Statistical Inference Problems with Applications to Computational Structural Biology

by

Parthan Kasarapu

Thesis

submitted for fulfillment of the requirements for the degree of

Doctor of Philosophy

Faculty of Information Technology
Monash University

April, 2016

Copyright Notice

Contents

List of Symbols & Abbreviations

$\approx$	Approximation
$\mathbf{\Phi}$	Set of mixture parameters
$\mathbf{\Theta}$	Vector of parameters
$\mathcal{D}$	Data set
$\mathbb{E}[.]$	Expectation of the quantity in [.]
$\mathcal{L}(\mathcal{D}\|\mathbf{\Theta})$	*Negative* log-likelihood
$\mathcal{H}$	Hypothesis
$\mathcal{F}$	Fisher information matrix
Pr	Probability
$\widehat{\mathbf{\Theta}}$	Estimate of the parameter vector
d	Dimensionality
K	Number of mixture components
3D	three-dimensional
FB_5	5-parameter Fisher-Bingham
AIC	Akaike Information Criterion
BIC	Bayesian Information Criterion
BVM	Bivariate von Mises
EM	Expectation-Maximization
KL	Kullback-Leibler
MAP	Maximum *a posteriori*
MDL	Minimum Description Length
ML	Maximum Likelihood
MML	Minimum Message Length
vMF	von Mises-Fisher

List of Tables

List of Figures

Statistical Inference Problems with Applications to Computational Structural Biology

Parthan Kasarapu
Monash University, 2016

Supervisors:
Arun Konagurthu & Maria Garcia de la Banda

Abstract

In this data pervasive world, the efficient and accurate modelling of data is crucial to support reliable analyses and to improve the solution to related problems. In order to describe the given data, the problem of selecting a suitable model has to be carefully addressed. Traditional approaches to the problem of optimal model selection have relied predominantly on the number of model parameters rather than the actual parameters themselves. This limits the ability of traditional methods to correctly distinguish among models that, while being of different type, have the same number of model parameters. In order to address the problem of model selection satisfactorily, this thesis explores the Bayesian information-theoretic principle of minimum message length (MML). The inference framework based on the MML principle enables the optimal selection of models by using the constituent parameters to better balance the trade-off between the *model's complexity* and its *goodness-of-fit* to the data. The core of this thesis explores the MML-based inference of some of the commonly used probability distributions whose parameters have not yet been characterized and of mixtures of these probability distributions. The models of these probability distributions allow for accurate modelling of data in the Euclidean space and data that is directional in nature. These probabilistic models and their mixtures have widespread uses in statistical machine learning tasks. In this context, we have developed a general purpose search method to determine the optimal number of mixture components and their parameters that describe the given data in a completely unsupervised setting. The use of the MML modelling paradigm and our proposed search method is explored in detail on a variety of real-world data, specifically on directional text data and on the spatial orientation data of protein three-dimensional structures. Further, mixtures of directional probability distributions have facilitated the design of reliable computational models for protein structural data. Furthermore, the inference framework has been used for concise representations of protein folding patterns using a combination of non-linear parametric curves. The results of this work have a wide-variety of important uses including direct applications in protein structural biology.

Statistical Inference Problems with Applications to Computational Structural Biology

Declaration

I declare that this thesis is my own work and has not been submitted in any form for another degree or diploma at any university or other institute of tertiary education. Information derived from the published and unpublished work of others has been acknowledged in the text and a list of references is given.

Parthan Kasarapu
April 21, 2016

Acknowledgements

The work carried out in my thesis would not have been possible without the generous support of my supervisors, Arun Konagurthu and Maria Garcia de la Banda. I will be forever grateful for their constant guidance and encouragement, which have been immensely uplifting. I am especially thankful for the confidence they had in me and the freedom they provided to pursue my research interests. Their extreme support and meticulous tutelage are invaluable and I consider myself fortunate to be supervised by them. I would like to extend my gratitude to Lloyd Allison, whose mentorship and insights have contributed to the shaping of my thesis.

I would also like to thank my family, friends, colleagues, and the staff at Faculty of Information Technology who have supported my time at Monash University. Finally, I am grateful to the Faculty of Information Technology for awarding a tuition-fee scholarship and National ICT Australia for granting me the living allowance stipend for the duration of my PhD candidature.

Parthan Kasarapu

Monash University
April 2016

Chapter 1

Introduction

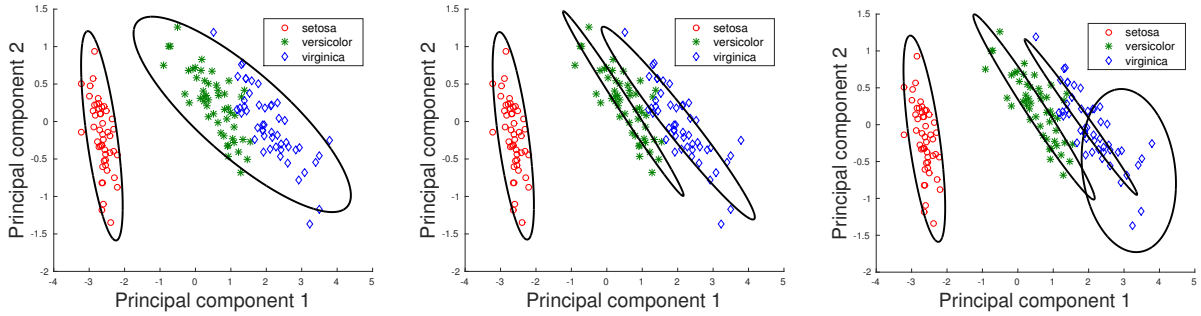
Data forms the basis of every aspect of our society. The number and direction of vehicles travelling on a particular road to the three-dimensional (3D) positions of the atoms in a protein structure are all potential sources of data. Analyzing this data can help us gain insights that, in turn, can enable us to improve the solution to related problems, like increasing the flow of vehicles along the road or finding similarities between two 3D protein structures. In order to support any analysis, data is usually described by a mathematical formula that is characterized by some parameters. This description of the data is referred to as a *model*. In general, the models adopted to describe the data are approximations of the, often unknown, true model.

As explained in Oliver and Baxter (1994), to describe some observed data, a set of models is usually considered. This set consists of *model classes*, where a model class is defined as a collection of models that have the same parametric form, including the same number of parameters. As per this definition, Oliver and Baxter (1994) consider, for example, the cubic polynomials to be of a different model class as compared to the quadratic polynomials. The union of all such model classes (in this case, polynomials of varying degree) make up the entire collection of models. A *model* is defined as a *particular* instance of this collection, that is, one whose parameters are fully specified.

In order to describe the data, a suitable model needs to be selected. This includes selecting a model class as well as the parameters that uniquely identify a model within that model class. The selection of a particular model refers to inductive inference or learning from observed data (Wallace, 2005). The model is an assertion on the nature of the observed data and inductive inference results in a probability value for the model given the data. The inductive inference methodology is usually based on statistical learning theory and, hence, referred to as statistical inference. The core of statistical inference relies on evaluating the probability of occurrence of a model and its associated parameters given the data, taking into account any prior knowledge that we may have about the data. When the data is described using probability distributions, model selection refers to the problem of inference of the parameters that characterize those distributions.

The notion of what constitutes a *suitable* model is dependent on the complexity of the model and the its ability to describe the data. The *model's complexity* is typically characterized in terms of a function involving the model's parameters (see below). The model's ability to describe the data is defined in terms of the error associated with fitting the model to the data. This is commonly referred to as the *goodness-of-fit*. A model that fits the observed data with minimal error has a high goodness-of-fit. A model with increased complexity generally improves the goodness-of-fit to the data. Naturally, there is a cost-benefit trade-off associated with selecting a suitable model.

In practice, there are often several candidate models that might be suitable and, therefore, their suitability, that is, their complexity and goodness-of-fit to the data needs to be carefully assessed and compared. In order to further motivate the problem of model selection, consider the modelling of the Iris data using Gaussian mixtures shown in Figure 1.1 (detailed analysis is presented in Section 5.5.6). A mixture model contains K component distributions, where each K-component mixture corresponds to a model class. Figure 1.1(a)-(c) shows three models, each from a different model class, corresponding



(a) (b) (c)

Figure 1.1: Example motivating the problem of model selection: which mixture model should be selected?

to the K-component mixtures for $K = 2, 3$, and 4 respectively. Other mixture models can be obtained by varying K. Each K-component mixture has an associated complexity and a goodness-of-fit value. From Figure 1.1, it is not immediately clear as to which model should be selected, as all the three models seem to be *suitable*. Hence, the problem of model selection is an important one as any future analysis will be based on the selected model.

Several criteria have been formulated to achieve a trustworthy balance between the complexity of the model and its goodness-of-fit to the data. In many applications, statistical hypothesis testing based on the likelihood ratio test is employed in model selection (see Section 2.3.1). The practical implementation of the likelihood ratio test involves its approximation by the chi-squared test (Wilks, 1938). This approximation is only valid for large sample sizes and for nested model classes (that is, when one of the models is a specific case of the other). These constraints limit the usage of the likelihood ratio test for testing varied model classes. This is why information-theoretic based criteria such as the Akaike information criterion (Akaike, 1974), Bayesian information criterion (Schwarz, 1978), and minimum description length (Rissanen, 1978). are widely used in model selection (see Chapter 2).

These information-theoretic criteria assign a score to a model based on its complexity and its goodness-of-fit to the data. Hence, model selection using these criteria is achieved by comparing the models based on their assigned score, and selecting the one that has the optimal score. However, these criteria are based on many simplifying assumptions that can be problematic (discussed in Section 2.3). The problems mainly arise from the fact that the model's complexity is determined by the *number* of parameters defining a model, rather than by the parameters themselves. As a result, all models within a model class have the same complexity. This would make it impossible to distinguish between the complexity of the models in Figure 1.1 and their counterparts with the same number of mixture components.

For a model under consideration, the notion of complexity should not be coarsely approximated solely by the number of model parameters. Indeed, the complexity of a model is a function of its probability distribution and its characterizing parameters. Any framework that quantifies the model complexity should be able to objectively account for these factors. This thesis attempts to address the problem of model selection from a Bayesian-information theoretic viewpoint using the minimum message length (MML) principle (Wallace and Boulton, 1968). By adopting the MML principle, the model's complexity is measured by giving importance to the constituent parameters, and not just their number (Wallace and Boulton, 1975; Wallace and Freeman, 1987; Wallace, 1997).

As per the MML model selection paradigm, an inference problem is decomposed as a two-part lossless message communicated between a hypothetical transmitter and receiver pair: the first part encodes the model's parameters and the second part encodes the data using those parameters. Hence,

different models result in different message lengths. The model that results in the shortest total message length is considered to be the best explanation of the data.

As part of model selection, the inference of model parameters is traditionally done by either *maximum likelihood* (ML) or by the Bayesian *maximum a posteriori* probability (MAP) based approaches. Parameter inference using these methods relies on maximizing the probability of the model given the data. This is called the posterior density. The computation of the posterior density depends on the prior knowledge of the parameters. This is expressed in terms of a prior probability defined over the parameter space (see Section 2.2). The ML estimation procedure ignores the prior probability of the parameters and, hence, does not include the model's parameters in determining the model's complexity. The MAP estimation procedure formulates the posterior density using a prior. However, the MAP estimates are not invariant to non-linear transformations of the parameter space and are, hence, inconsistent (Oliver and Baxter, 1994).

The MML method fundamentally differs from the traditional methods of statistical inference in the manner it handles the model's parameters. Every continuous value has a measurement precision and need not be inferred more precisely than required. As part of the MML-based inference, the optimal precision to which the parameters need to be stated is computed. As a result, every parameter has a non-zero probability value and consequently an associated encoding cost that is used to quantify the model's complexity. Unlike ML, MML accounts for the model's complexity by computing the precision of a model's parameters and unlike MAP, MML estimators are invariant to re-parameterization of data and parameters (Wallace, 2005).

Thus, the MML framework provides a rigorous means to evaluate competing hypotheses in their ability to explain some observed data. Unfortunately, the "strict" MML formulation proposed by Wallace and Boulton (1975) is computationally intractable for majority of cases (Farr and Wallace, 2002). As a means of overcoming this problem, Wallace and Freeman (1987) proposed a practical approximation for model selection and inference. This approximation leads to a systematic procedure to formulate the message length expression to model the data using probability distributions (see Section 2.4). This version of the MML formulation forms the foundation on which the model selection problems are investigated in this thesis.

The core of the thesis explores the MML-based parameter inference of some of the commonly used probability distributions whose parameters have not yet been characterized using the Wallace and Freeman (1987) approximation. These include the Laplace and multivariate Gaussian distributions that are useful for modelling the data in the Euclidean space. The Laplace distribution has diverse applications in areas such as signal processing (Eltoft et al., 2006), image denoising (Rabbani et al., 2006), gene expression studies (Bhowmick et al., 2006), and market risk prediction (Haas et al., 2006). The multivariate Gaussian distributions are important as they are widely used in statistical pattern recognition tasks (McLachlan and Basford, 1988; McLachlan and Peel, 2000).

The thesis also explores the MML-based inference using some commonly used directional probability distributions. As the Laplace and Gaussian distributions are not suitable to model data with inherent directional nature, the directional probability distributions are studied to model such data (Mardia and Jupp, 2000). The directional distributions considered in this thesis are the multivariate von Mises-Fisher (vMF) to model data distributed on the unit hypersphere, the Kent distribution to model data on the three-dimensional (3D) sphere, and the bivariate von Mises distribution to model data distributed on the surface of the 3D torus.

These distributions are important in statistical modelling tasks and have widespread applicability. For example, multivariate vMF distributions have been used in text clustering and protein modelling tasks (Banerjee et al., 2005). Kent distributions have applications in geology (Peel et al., 2001) and structural bioinformatics (Kent and Hamelryck, 2005; Boomsma et al., 2006; Hamelryck, 2009). Toroidal distributions are used to model protein dihedral angles (Mardia et al., 2007, 2008), which serve as an important step in understanding the structural properties of proteins and characterizing their folding patterns (Leach, 2001).

The modelling of real-world data using probability distributions often requires the consideration of mixture models. This is because the empirical distribution of data typically contains multiple modes, and, therefore, cannot be efficiently modelled by a single component. Hence, mixtures of probability distributions are considered as they facilitate the modelling of data that is potentially multimodal. In this thesis, the MML principle has been used to develop a framework to model mixtures of probability distributions. An important aspect of the mixture modelling problem is the determination of an optimal number of mixture components. As stated previously, this corresponds to identifying the model class, and selecting a mixture model corresponds to estimating the mixture parameters. In this context, we designed a generalized search method to infer an optimal mixture that best models the given data.

In this thesis, we consider mixtures of multivariate Gaussian distributions to model data in the Euclidean space, and mixtures of multivariate vMF, 3D Kent, and bivariate von Mises distributions to model directional data. We have tested our mixture modelling approach using Gaussian mixtures and demonstrated it performs better compared to the state of the art (see Chapter 5). We have also tested multivariate vMF mixtures to model high-dimensional text data (which is directional) and demonstrated the effectiveness of the modelling approach in selecting optimal mixtures.

We have also used directional distributions to model the directional data arising out of protein chain conformations. The protein chain is comprised of a sequence of amino acid residues, each containing a central carbon atom. The directional data corresponds to the spatial orientation of these carbon atoms with respect to the preceding carbon atoms. This data lies on the surface of a three-dimensional (3D) unit sphere and is ideally modelled using mixtures of 3D vMF and Kent distributions.

The vMF distribution is a specific case of a Kent distribution with fewer number of parameters. Hence, the vMF and Kent mixtures correspond to different model classes. In this context, we demonstrate the ability of the MML principle in distinguishing competing mixtures belonging to different model classes, and selecting the optimal mixture model. Furthermore, mixtures of bivariate von Mises distributions have been employed to model protein dihedral angles. Unlike the directional data used in the case of vMF and Kent mixtures, the dihedral angles are distributed on the toroidal surface and hence, bivariate von Mises mixtures are appropriate. The resulting mixture models of protein directional data serve as fundamental tools in structural biology tasks such as generating random protein chain conformations, three-dimensional protein structure alignment, secondary structure assignment, among others (Hamelryck et al., 2006; Konagurthu et al., 2012, 2013; Collier et al., 2014).

In addition to MML-based parameter inference of Euclidean and directional probability distributions, and the modelling and search for optimal mixtures, this thesis also explores the MML-based model selection in the context of function approximation. Specifically, the problem considered is the approximation of a periodic function by truncating its infinite series expansion using a (finite) linear combination of orthogonal basis functions. Each truncation using (say) K basis functions corresponds to a model class. A model within the model class is specified using the weights for each basis function. In this case, model selection refers to the statistical inference of these weights associated with the K basis functions. The motivating foundations of this problem lies in protein structure elucidation using X-ray crystallography, where the intensity values of the diffracted X-rays of crystallized protein molecules are converted to 3D structures using an inverse Fourier transform. The cost-benefit trade-off of approximating using a longer Fourier expansion (model's complexity) versus the smaller error rate (goodness-of-fit) is resolved using the MML principle.

This thesis further explores the MML-based model selection when the underlying data is not modelled using probability distributions. The previously discussed mixture modelling and function approximation problems rely on probability distributions to model the given data. When using probability distributions, the Wallace and Freeman (1987) based MML inference allows us to formulate the message length expression in a closed form by encoding the model's parameters and the data given those parameters. However, in cases where the models are not defined in terms of probability distributions, we need to develop *ad hoc* schemes to explicitly encode the model and the data given that model.

As a case in point, we consider the problem of representing a sequence of 3D points using a piecewise ensemble of Bézier curves (linear, quadratic, and cubic). A Bézier curve is a polynomial whose degree is determined by the number of control points. The control points give the flexibility to control the 3D trace of the curve and they uniquely determine the mathematical form of the Bézier curve (see Section 8.2). A given collection of curves constitute a model (first part). Each model is specified by the degree and control points of each of the Bézier curves in the ensemble. The ability of the model to fit the 3D points with minimum error constitutes the second part of the message. We develop methods to construct the message lengths corresponding to the two parts and minimize the total message length to obtain an optimal ensemble. An example application includes the modelling of protein structural data leading to their concise representations using Bézier curves. These representations are central to the analyses of protein conformations, including effective structural searches on large protein structural databases.

The potential implications of my research outcomes are in the construction of rigorous statistical models that could be employed to model data in the Euclidean space and data that is directional. The thesis aims to develop novel computational models for modelling data and we have highlighted the applications to modelling of protein spatial orientation data. The resulting models are key to fundamental tasks in structural biology. The models developed in conjunction with the inference framework have demonstrable applications in many other domains. From a theoretical standpoint, I have characterized the MML estimators of the parameters of a variety of probability distributions as part of my research. In particular, I have advanced the state of the art with respect to modelling using directional distributions. The effectiveness of the MML principle to objectively assess models is the key theme that formed the foundations of my thesis.

1.1 Contributions

In summary, the research contributions as part of my thesis are the following:

- Derivation of the MML estimates of the parameters of the following distributions, thus allowing for the accurate modelling of data in the Euclidean space:
  - Laplace distribution, and
  - Multivariate Gaussian distribution
- Derivation of the MML estimates of the parameters of the following directional probability distributions:
  - Multivariate von Mises-Fisher to model data on the unit hypersphere,
  - Kent distribution to model data on the surface of a 3D sphere, and
  - Bivariate von Mises to model data distributed on the surface of a 3D torus
- Mixture modelling
  - Conceptualizing and implementing a generic search method to infer the optimal number of mixture components that best describe the given data.
  - Modelling multivariate directional data using vMF mixtures; applying it to high-dimensional text clustering, and modelling three-dimensional directional data using Kent mixtures.
  - Designing efficient descriptors of protein data that can aid in modelling structural biology tasks.
  - Using mixtures of bivariate von Mises distributions to model protein main chain dihedral angles that can aid in studying the structural properties of proteins.
- Function approximation

  - Determining the optimal number of terms to approximate the infinite series expansion of a periodic function using orthogonal basis functions, that has potential applications in protein structure determination.

- Modelling using curvilinear representations
  - Designing an inference framework to model 3D data with non-linear geometry using a combination of parametric Bézier curves.
  - Generating concise representations of protein folding patterns using piecewise Bézier curves, and applying to database search and retrieval.

1.2 Thesis Outline

The structure of the thesis is as follows: **Chapter 2** describes the traditional methods for parameter inference as part of model selection. These are the maximum likelihood (ML) and maximum *a posteriori* probability (MAP) approaches. We explain how they are traditionally used along with the different model selection criteria. These include statistical hypothesis testing and information-theoretic criteria such as minimum description length (MDL), Akaike and Bayesian information criteria. In this context, we explain the usual framework of model selection using the minimum message length (MML) criterion and how it differs from the other criteria. In particular, we focus on the well-known Wallace and Freeman (1987) method, which is a practical approximation to the generalized MML criterion (Wallace and Boulton, 1975). The Wallace and Freeman (1987) based inference framework is used in the subsequent chapters to estimate the parameters of probability distributions that had not been previously characterized. In order to evaluate the new MML parameter estimates against the traditional ML and MAP estimates, the chapter includes a discussion of using the bias and mean squared error (MSE) of the estimates and the Kullback-Leibler (KL) distance to compare the different estimators.

Chapter 3 focuses on distributions used to model data distributed in the Euclidean space. In particular, it presents my derivations of the MML estimates of the Laplace and of the multivariate Gaussian distribution using the Wallace and Freeman (1987) method. The resulting MML estimators of both distributions are demonstrated to have lower bias as compared to the traditionally used ML estimators of Laplace (Norton, 1984) and Gaussian (Barton, 1961) distributions. This unbiased estimator of the multivariate Gaussian distribution is used in mixture modelling in Chapter 5.

Since the Laplace distribution is univariate, we consider its analysis against the univariate Gaussian distribution in Chapter 3. In modelling some observed data using the Laplace and (univariate) Gaussian distributions, we use the derived estimators to determine the preference of a Laplace or a Gaussian as per the MML framework. This is done by comparing the message lengths obtained by the two distributions. The distribution with the least total message length is selected as the best model to describe the data. We study the behaviour of the difference in their message lengths in order to validate that the MML framework is able to discriminate between the Laplace and Gaussian distributions and select the correct model under varying amount of data. We then show some applications of MML-based model selection in modelling real-world datasets using the Laplace and univariate Gaussian distributions.

Chapter 4 focuses on distributions suitable to model data with inherent directional nature, that is, directional probability distributions. In particular, we consider three such distributions: the multivariate von Mises-Fisher (vMF), the 3D Kent distribution, and the bivariate von Mises (BVM) distribution. The vMF distribution is the spherical analogue of the symmetric Gaussian wrapped on the unit hypersphere. The 3D Kent distribution is a generalization of the vMF distribution and has ellipse-like contours on the spherical surface. It is the spherical analogue of the asymmetric bivariate Gaussian distribution. The BVM distribution is the toroidal analogue of the asymmetric bivariate Gaussian distribution, and has ellipse-like contours on the surface of the 3D torus.

The chapter starts with a description of the existing estimators of the parameters of a multivariate vMF distribution. The MML estimates of the parameters are derived and are compared with the

contemporary estimators based on ML estimation (Banerjee et al., 2005; Tanabe et al., 2007; Sra, 2012; Song et al., 2012). The MML estimators are demonstrated to outperform the contemporary estimators with respect to metrics such as bias, MSE, KL distance, and likelihood ratio based statistical hypothesis testing.

The chapter then details the inference using Kent distributions defined on the surface of a 3D sphere. We explain the intuition behind modelling using the Kent distributions and discuss the traditional methods of parameter estimation. These are based on moment, (Kent, 1982), ML and MAP-based estimation. In this context, we discuss different parameterizations of the distribution in order to demonstrate how the MAP estimates vary and consequently result in inconsistent estimates. We outline the derivation of the MML paramter estimates, and compare them with the exising estimators. We empirically evaluate the MML estimators against the traditional estimators by comparing their bias and MSE with respect to the true distribution parameters. We also analyze the KL distance due to the estimated parameters and perform statistical hypothesis testing to demonstrate the superior performance of the MML estimators.

This is followed by a discussion of the inference using BVM distributions defined on the surface of a 3D torus. Similar to the Kent distribution, we discuss the existing methods of parameter estimation and subsequently derive the MML estimates. The derived estimators are shown to perform better than the traditional estimators based on the experimental evaluation, as it was done for Kent distributions.

Chapter 5 explores the MML-based model selection in the context of mixture modelling using probability distributions. In addition to the selection of a representative component distribution, it is also essential to infer an optimal number of mixture components. This chapter describes some of the existing methods that are often used to determine the number of mixture components. We highlight the limitations of these methods and propose an MML-based search method in order to rectify those drawbacks. The chapter details how the MML framework, in conjunction with the search method, is used to evaluate competing mixtures and select the one that has the least total message length.

The proposed search method begins with a one-component mixture and iteratively updates the mixture components through a series of perturbations depending on the improvement to the total message length. These perturbations, which are termed as Split, Delete, and Merge operations, alter the number of mixture components. We provide an example illustrating the search process and describe the various steps involved. The chapter details the design of this mixture modelling apparatus in the case of multivariate Gaussian distributions. The superior performance of the search and inference method is demonstrated against the widely used method of Figueiredo and Jain (2002). The proposed method is generalizable to accommodate models of probability distributions whose parameters can be estimated using the Wallace and Freeman (1987) method.

Chapter 6 focuses on mixture modelling using the previously discussed directional distributions, namely the multivariate vMF, 3D Kent, and the bivariate von Mises. Our proposed mixture modelling method, formulated in the case of Gaussian distributions in Chapter 5, is extended to model data using vMF mixtures. The vMF distributions have been previously used in clustering text documents (Banerjee et al., 2005), and protein dihedral angles (Dowe et al., 1996a). As example applications, we employed mixtures of vMF distributions in the context of high-dimensional text clustering and modelling directional data resulting from protein conformations. Compared to the related work, the results obtained due to the MML-based search method demonstrate the ability of MML-based inference to infer optimal vMF mixtures in a completely unsupervised setting (Kasarapu and Allison, 2015).

We then adapted our search method to handle mixtures of Kent distributions. As a specific example, Kent mixtures were employed in modelling protein directional data. The resulting vMF and Kent mixtures indeed serve as efficient descriptors of protein structural data. These are shown to be better alternatives as compared to the uniform distribution on the sphere which was previously considered by Konagurthu et al. (2012) and Collier et al. (2014) in protein modelling tasks. The results following the application of Kent mixtures to model protein directional data demonstrate that they supersede the vMF mixture models. The ability of Kent distributions to model asymmetrical data

leads to improved description of the protein structural data. Hence, they serve as natural successors to the vMF mixture models to describe protein structural data.

Both the vMF and Kent probability distributions are defined on unit spheres and are used to model data distributed on the spherical surface. We explored a further extension in the thesis to model directional data distributed on the toroidal manifold using the MML framework. This is important to model data such as the protein main chain dihedral angles. The dihedral angle data is different to the data considered in protein structure modelling using vMF and Kent distributions, and are not distributed on the spherical surface. Therefore, we consider mixtures of bivariate von Mises distributions to model the dihedral angles of the protein main chain.

Chapter 7 focuses on the problem of model selection in the context of function approximation, where the infinite series expansion of a periodic function is truncated using a linear combination of orthogonal basis functions. In this chapter, for an expansion involving a finite number of orthogonal basis functions, the inference of parameters of the linear combination is done using MML-based regression analysis. We consider two types of orthogonal basis functions: (1) the Fourier series comprising of trignometric basis functions such as alternating sines and cosines, and (2) Legendre polynomials. We demonstrate how the MML-based model selection paradigm is able to compute the total message length corresponding to different series expansions with different number of orthogonal functions, and select the expansion that has the least total message length. The approximation of a Fourier series is an important problem as it has direct implications in protein structure determination using X-ray crystallography, where the 2D diffraction pattern of protein crystals is transformed to its corresponding 3D structure using a Fourier transform.

Chapter 8 explores the problem of model selection when the model does not correspond to a probability distribution. In particular, we consider the problem of modelling a given set of 3D points using a piecewise combination of Bézier curves of varying degree. We discuss our strategies involved in explicitly encoding the model's parameters and the data using those parameters. We demonstrate how the MML-based inference framework is used in determining the suitable number and the degree of the Bézier curves that can model the given sequence of 3D points. An example application includes the modelling of protein folding patterns leading to their concise representations. We discuss case studies involving some protein structures and show how our proposed method is able to generate concise representations of the 3D trace of the coordinates of protein structures. We detail how the representations can be used as unique signatures corresponding to each protein and how these are central to the analyses of protein conformations, including effective structural searches on large protein databases (Kasarapu et al., 2014).

Chapter 9 concludes the thesis by outlining the key results of the work. We provide some potential research ideas that can be carried out as extensions of the current thesis.

Chapter 2

Model selection and inference

2.1 Introduction

This chapter outlines the methods that are commonly used in model selection and inference, thus providing the background for the rest of the thesis. As mentioned in Chapter 1, a set of models is usually considered when describing some observed data. This *model set* consists of several *model classes*, where a model class is defined as a collection of models that have the same parametric form and, thus, the same number of parameters.

A *model* is defined as an instance of the model class whose parameters are fully specified. For example, consider a graph with K nodes wherein every pair of nodes are connected to each other. The parameters of the graph, in this case, could be the weights of the connections between any pair of nodes. A model in this case corresponds to a graph with K nodes but with a specified set of weights for the connections. All graphs with K number of nodes but with varying weights constitute a model class. Naturally, there are countably infinite model classes corresponding to varying number of nodes. Similarly, as explained in Chapter 1, mixture distributions containing K components constitute a model class. Varying K leads to different model classes, and within each class, a model corresponds to a mixture whose parameters are completely specified.

The problem of model selection refers to the consideration of *one* particular model that belongs to one of the infinite model classes. When modelling using probability distributions, model selection refers to the problem of inference of the parameters that characterize those distributions. Some traditional methods of parameter estimation include the maximum likelihood (ML) and the maximum *a posteriori* probability (MAP) (Section 2.2). A model obtained in this way can be used for future analysis by predicting the probability of occurrence of unseen data.

As mentioned in Chapter 1, a model that optimally balances the trade-off due to its complexity and goodness-of-fit should be selected for modelling the data. Any model selection paradigm should be able to balance the trade-off arising due to these two conflicting objectives. Generally speaking, models with a higher number of parameters fit the data better. However, the model's complexity should not be a sole function of the *number* of parameters. In this context, we discuss some commonly used model selection frameworks that account for these two factors in Section 2.3. These include the likelihood ratio test and information-theoretic criteria such as minimum message length (MML) and minimum description length (MDL).

This chapter focuses on the MML-based model selection framework and explains the key differences with respect to other model selection frameworks. In particular, we focus on the Wallace and Freeman (1987) method which, as mentioned in Chapter 1, is a practical approximation to the generalized MML criterion (Wallace and Boulton, 1975). In this thesis, the Wallace and Freeman (1987) based inference framework is used to estimate the parameters of the probability distributions. In order to evaluate the MML parameter estimates against the traditional ML and MAP estimates, this chapter includes a discussion of using the bias and mean squared error (MSE) of the estimates and the Kullback-Leibler (KL) distance to compare the different estimators.

This chapter is organized as follows: Section 2.2 describes the traditional ML and MAP-based approaches used to infer the parameters of a model. Section 2.3 describes the frameworks commonly used in the model selection. The details of the likelihood ratio test are presented first followed by a discussion on the information-theoretic model selection criteria. This is followed by a discussion of the MML criterion in Section 2.4, This section outlines the Wallace and Freeman (1987) approximation. An example outlining the Wallace and Freeman (1987) approximation is shown in the case of inference of parameters of a univariate Gaussian distribution. This is followed by a proof establishing the statistical consistency of the MML parameter estimators. Section 2.5 outlines the comparison methodologies that are employed to evaluate the quality of the inferred parameter estimates against the traditional estimators.

2.2 Traditional methods of parameter estimation

A model is characterized by a set of parameters $\boldsymbol{\Theta}$. Thus, model selection corresponds to inference of the model's parameters. In other words, estimating the parameters of a model is fundamental to selecting a model from a model class. Traditional methods of parameter estimation are based on a Bayesian theoretic interpretation of the data and an assumed hypothesis on the data. A *hypothesis* is defined as an assertion on the underlying distribution of the data, with a model then being a mathematical realization of the hypothesis. The hypothesis corresponds to a model class that is parameterized by $\boldsymbol{\Theta}$. According to the well-known Bayes's theorem, we have

$$\Pr(\mathcal{H} \,\&\, \mathcal{D}) = \Pr(\mathcal{H}) \times \Pr(\mathcal{D}|\mathcal{H}) = \Pr(\mathcal{D}) \times \Pr(\mathcal{H}|\mathcal{D}) \tag{2.1}$$

where $\mathcal{D}$ denotes the observed data, $\mathcal{H}$ some hypothesis about that data, and $\Pr(X)$ denotes the probability of X. Further, $\Pr(\mathcal{H} \,\&\, \mathcal{D})$ is referred to as the joint probability, $\Pr(\mathcal{H})$ and $\Pr(\mathcal{D})$ are called the prior densities of the hypothesis $\mathcal{H}$ and data $\mathcal{D}$, respectively, $\Pr(\mathcal{H}|\mathcal{D})$ is referred to as the posterior probability, and $\Pr(\mathcal{D}|\mathcal{H})$ is referred to as the likelihood. In terms of the model parameters, we have

$$\Pr(\boldsymbol{\Theta}|\mathcal{D}) = \frac{\Pr(\mathcal{D}|\boldsymbol{\Theta})\Pr(\boldsymbol{\Theta})}{\Pr(\mathcal{D})} \quad \text{where} \quad \Pr(\mathcal{D}) = \int_{\boldsymbol{\Theta}} \Pr(\mathcal{D}|\boldsymbol{\Theta})\Pr(\boldsymbol{\Theta})\, d\boldsymbol{\Theta} \tag{2.2}$$

Note the dependence on $\Pr(\boldsymbol{\Theta})$, which is a prior density defined on the parameters. As mentioned before, there are two commonly used methods of parameter estimation: These are the maximum likelihood (ML) and the maximum *a posteriori* probability (MAP). They differ in their treatment of the prior density and consequent approximations of the posterior density $\Pr(\boldsymbol{\Theta}|\mathcal{D})$. In both these methods, the optimal parameters correspond to those that maximize the posterior density, and will be referred to as $\widehat{\boldsymbol{\Theta}}_{\text{ML}}$ and $\widehat{\boldsymbol{\Theta}}_{\text{MAP}}$ respectively. The ML and MAP-based estimation methods are summarized in the next two sub-sections.

2.2.1 Maximum Likelihood (ML) estimation

The ML estimation procedure considers the use of a *uniform* prior density $\Pr(\boldsymbol{\Theta})$ on the parameters (Murphy, 2012). As such, the prior density is essentially ignored, while maximizing the posterior density $\Pr(\boldsymbol{\Theta}|\mathcal{D})$. This is appropriate when there is no additional information available about the nature of the parameters and, thus, no prior knowledge. Based on this assumption, maximizing the posterior density $\Pr(\boldsymbol{\Theta}|\mathcal{D})$ is equivalent to maximizing the likelihood function $\Pr(\mathcal{D}|\boldsymbol{\Theta})$. The ML estimator is, thereby, given by

$$\widehat{\boldsymbol{\Theta}}_{\text{ML}} = \arg\max_{\boldsymbol{\Theta}} \Pr(\mathcal{D}|\boldsymbol{\Theta})$$

where $\arg\max_{\boldsymbol{\Theta}}$ refers to the $\boldsymbol{\Theta}$ (argument) that maximizes the likelihood function $\Pr(\mathcal{D}|\boldsymbol{\Theta})$.

Further, the likelihood term is calculated by assuming the data consists of independent and identically distributed (i.i.d.) sample points. For $\mathbf{x}_i \in \mathcal{D}$ and a probability model $f(\mathbf{x}; \boldsymbol{\Theta})$, assuming an

i.i.d. sample leads to the likelihood function being computed as

$$\Pr(\mathcal{D}|\boldsymbol{\Theta}) \approx \prod_{\mathbf{x}_i \in \mathcal{D}} f(\mathbf{x}_i; \boldsymbol{\Theta})$$

where $\approx$ denotes the approximation symbol, and $\prod$ denotes the product expression.

2.2.2 Maximum *a posteriori* probability (MAP) estimation

The MAP estimation considers a non-uniform prior defined on the parameter vector $\boldsymbol{\Theta}$. This is appropriate when some prior knowledge is available about the nature of the parameters, as it is usually the case. The resultant posterior density $\Pr(\boldsymbol{\Theta}|\mathcal{D})$ is then directly proportional to the product of the prior density of the parameters $\Pr(\boldsymbol{\Theta})$ and the likelihood of the data given those parameters $\Pr(\mathcal{D}|\boldsymbol{\Theta})$. The likelihood term is computed assuming again an i.i.d. sample. The MAP estimator is then obtained by maximizing the posterior density and is given by (Murphy, 2012)

$$\widehat{\boldsymbol{\Theta}}_{\text{MAP}} = \arg\max_{\boldsymbol{\Theta}} \Pr(\boldsymbol{\Theta}) \Pr(\mathcal{D}|\boldsymbol{\Theta})$$

Note that the MAP estimate depends on the parameterization of the probability density function $f(\mathbf{x}; \boldsymbol{\Theta})$. As a result, the MAP estimator is *not* invariant under some non-linear transformations of the parameter space. If $T(\boldsymbol{\Theta}) = \boldsymbol{\Theta}'$ denotes a transformation on the parameter vector $\boldsymbol{\Theta}$, then for invariance, the parameter estimates $\widehat{\boldsymbol{\Theta}}_{\text{MAP}}$ and $\widehat{\boldsymbol{\Theta}}'_{\text{MAP}}$ in both the parameterizations should be affected by the same transformation. In this setup, the two versions of the posterior density will be

$$\begin{aligned} \text{Posterior}(\boldsymbol{\Theta}|\mathcal{D}) &\propto \Pr(\boldsymbol{\Theta}) \times \prod_{\mathbf{x}_i \in \mathcal{D}} f(\mathbf{x}_i; \boldsymbol{\Theta}) \\ \text{Posterior}(\boldsymbol{\Theta}'|\mathcal{D}) &\propto \Pr(\boldsymbol{\Theta}') \times \prod_{\mathbf{x}_i \in \mathcal{D}} f(\mathbf{x}_i; \boldsymbol{\Theta}') \end{aligned} \tag{2.3}$$

where $\Pr(\boldsymbol{\Theta}')$ is the prior density in the alternative parameter space. If $\boldsymbol{J}$ is the Jacobian matrix of the transformation given by $\boldsymbol{J} = \dfrac{\partial \boldsymbol{\Theta}'}{\partial \boldsymbol{\Theta}}$ and $|\boldsymbol{J}|$ is its determinant, then $\Pr(\boldsymbol{\Theta}') = \dfrac{\Pr(\boldsymbol{\Theta})}{|\boldsymbol{J}|}$. The two versions of the MAP estimates should be related as $T(\widehat{\boldsymbol{\Theta}}_{\text{MAP}}) = \widehat{\boldsymbol{\Theta}}'_{\text{MAP}}$. However, this is not usually the case in MAP-based estimation. This is because the MAP estimator corresponds to the mode of the posterior density and the mode is not invariant under non-linear transformations (Oliver and Baxter, 1994). This is a drawback of MAP estimation because statistical invariance is a required property for any estimation method. Any estimator that is invariant under non-linear transformations of the parameter space is robust as it is not dependent on how the model is parameterized. Examples demonstrating this behaviour are discussed in Sections 4.3.3 and 4.4.2.

2.3 Commonly used model selection paradigms

The estimation of the parameters of a model allows the computation of the likelihood function, which corresponds to the goodness-of-fit of the model to the data. In order to determine the suitability of a model in describing the given data, the model's complexity needs to be accounted for. Here, we look into some commonly used frameworks that determine the model's complexity. One of them is the statistical hypothesis testing method, which examines the asymptotic behaviour of the likelihood ratio corresponding to competing models. The other frameworks are based on information-theoretic criteria and are used to quantify the model's complexity as a function of the model's parameters. These are the minimum description length (MDL) and the minimum message length (MML).

2.3.1 Likelihood Ratio

The likelihood ratio test is used to determine the preference of a null hypothesis $\mathcal{H}$ over an alternate hypothesis $\mathcal{H}'$. An alternate hypothesis is defined as a general model class with a certain number of free parameters. A null hypothesis is a simpler or nested model class, which is defined as a specific case of the general model class. The nested class is obtained by constraining some of the parameters of the alternate hypothesis. The nested model class therefore, has fewer free parameters compared to the alternate hypothesis and is fully contained within the general model class. Every model in the nested model class is consequently a member of the general model class. For example, a quadratic polynomial is a nested model class of a cubic polynomial. As mentioned previously, a model is a mathematical characterization of a hypothesis and hence, each hypothesis will have a parametric form governed by the parameter vector $\boldsymbol{\Theta}$.

Given the data $\mathcal{D}$, the likelihood ratio test test involves evaluating the ratio of the maximized likelihood of the data under both the hypotheses. Let the ratio be denoted by

$$\lambda = \frac{\max\limits_{\mathcal{H}} \text{ likelihood}(\mathcal{D}|\mathcal{H})}{\max\limits_{\mathcal{H}'} \text{ likelihood}(\mathcal{D}|\mathcal{H}')} = \frac{\Pr(\mathcal{D}|\widehat{\boldsymbol{\Theta}}_{\text{ML}})}{\Pr(\mathcal{D}|\widehat{\boldsymbol{\Theta}}'_{\text{ML}})}$$

In the above equation, $\widehat{\boldsymbol{\Theta}}_{\text{ML}}$ and $\widehat{\boldsymbol{\Theta}}'_{\text{ML}}$ are the maximum likelihood estimates of the parameters. In other words, $\widehat{\boldsymbol{\Theta}}_{\text{ML}}$ and $\widehat{\boldsymbol{\Theta}}'_{\text{ML}}$ correspond to the optimal models in both the model classes $\mathcal{H}$ and $\mathcal{H}'$ respectively. The test essentially compares the ratio of likelihoods if it were modelled using the *best* representative models from the two model classes.

The *test statistic* based on λ is defined as the negative logarithm of the likelihood ratio and is given by $\Lambda = -2\log\lambda$. As the likelihood due to the null hypothesis decreases, then λ decreases leading to an increase in Λ. Thus, the test statistic can be seen as a qualitative measure of the divergence from the null hypothesis. The likelihood ratio test involves computing the ratio Λ given the data $\mathcal{D}$. This is then compared with a pre-defined critical value that corresponds to the *significance* of the test. If the obtained test statistic exceeds the critical value, then it is considered that there is *significant* evidence in rejecting the null hypothesis, that is, the model class with fewer number of parameters is rejected in favour of the alternate hypothesis.

The significance of the test is pre-set to some reasonable value, say $\alpha < 1$. The value α corresponds to the minimum allowed probability of observing a likelihood ratio by chance, assuming the data is sampled from the null hypothesis. The value of α can be chosen depending on the required statistical significance (a value of 0.5 is typically chosen). This is used to compute the critical value which defines a confidence region in which the test statistic should lie if the null hypothesis were to be accepted. To determine this confidence region, it is required to formulate the probability distribution of Λ. This presents a difficulty as it is often not possible to determine the exact form of the probability distribution. In such cases, the distribution of the statistic Λ is *asymptotically* approximated as a χ^2 distribution with degrees of freedom equal to the difference in the number of free parameters between the alternate and the null hypotheses (Wilks, 1938). In summary, if λ is sufficiently small, it would lead to a rejection of the null hypothesis. Conversely, if Λ exceeds some confidence threshold, $\mathcal{H}$ is rejected.

The likelihood ratio test is appealing because its simplistic approximation allows its employability in a wide variety of scenarios. However, it is to be noted that the approximation suggested by Wilks (1938) is only valid for nested model classes. This restricts its direct application in comparing model classes which do not satisfy this property. For example, the likelihood ratio test is not applicable to differentiate mixture models as formulation of a nested mixture hypothesis is not unique (Smith, 1989). This has led to researchers using several other approximations to the problem of hypothesis testing of mixtures (Chen and Cheng, 1997; Lo et al., 2001; Chen et al., 2001; Chen and Li, 2009). Also, much of the research has been focused on univariate Gaussian mixtures and a corresponding null hypothesis with one mixture component. In this regard, Li and Chen (2010) aim to improve

on these approximations. However, the natural generalization of these approximating approaches to accommodate multivariate data and to mixtures of any probability distribution is not available to the best of our knowledge. Furthermore, these approximations are valid only in the asymptotic case which further restricts its utility, as most practical examples deal with finite amounts of data.

2.3.2 Information-theoretic criteria

In this section, some of the commonly used information-theoretic model evaluation criteria are outlined following their chronological order.

Strict Minimum Message Length (Wallace and Boulton, 1968, 1975)

As mentioned before, Wallace and Boulton (1968) developed the first criterion for model selection based on information theory. The work is based on that of Shannon (1948), which showed how given an event E with probability $\Pr(E)$, the length of the optimal lossless code to represent that event requires $I(E) = -\log_2(\Pr(E))$ bits. Applying Shannon's insight to Bayes's theorem (Equation 2.1), Wallace and Boulton (1968) got the following relationship:

$$I(\mathcal{H} \,\&\, \mathcal{D}) = I(\mathcal{H}) + I(\mathcal{D}|\mathcal{H}) = I(\mathcal{D}) + I(\mathcal{H}|\mathcal{D}) \tag{2.4}$$

As a result, given two competing hypotheses $\mathcal{H}$ and $\mathcal{H}'$, the difference in message lengths ΔI, is given as

$$\Delta I = I(\mathcal{H} \,\&\, \mathcal{D}) - I(\mathcal{H}' \,\&\, \mathcal{D}) = I(\mathcal{H}|\mathcal{D}) - I(\mathcal{H}'|\mathcal{D}) \,\text{bits}.$$

If the difference $\Delta I > 0$, it gives a measure of the extra compression achieved by $\mathcal{H}'$ over $\mathcal{H}$. The value of ΔI can be used to compute the posterior log-odds ratio between the two hypotheses. Using the fact that $I(\mathcal{H}|\mathcal{D}) = -\log_2 \Pr(\mathcal{H}|\mathcal{D})$, the above equation ΔI can be expressed as

$$\begin{aligned} \Delta I = I(\mathcal{H}|\mathcal{D}) - I(\mathcal{H}'|\mathcal{D}) &= -\log_2 \Pr(\mathcal{H}|\mathcal{D}) + \log_2 \Pr(\mathcal{H}'|\mathcal{D}) \\ \Pr(\mathcal{H}'|\mathcal{D}) &= 2^{\Delta I} \Pr(\mathcal{H}|\mathcal{D}) \end{aligned} \tag{2.5}$$

This relation is important because depending on ΔI, one can determine the odds of using a particular hypothesis to model the given data.

$I(\mathcal{H} \,\&\, \mathcal{D})$ can be interpreted as the *total* cost to encode a message comprising of the following two parts:

1. the hypothesis $\mathcal{H}$, which takes $I(\mathcal{H})$ bits, and
2. the observed data $\mathcal{D}$ using knowledge of $\mathcal{H}$, which takes $I(\mathcal{D}|\mathcal{H})$ bits.

A more complex $\mathcal{H}$ may explain $\mathcal{D}$ better thus resulting in a smaller $I(\mathcal{D}|\mathcal{H})$, but may also take more bits to be stated resulting in a larger $I(\mathcal{H})$. The trade-off comes from the fact that (hypothetically) transmitting the message requires the encoding of both the hypothesis and the data given the hypothesis, that is, the model complexity $I(\mathcal{H})$ and the goodness-of-fit $I(\mathcal{D}|\mathcal{H})$. The model that results in the shortest *total* message length is considered to be the best model to describe the data. Thus, the framework provides a better means to objectively compare two competing hypotheses, as it accounts for the complexity of the hypothesis itself.

For a hypothesis $\mathcal{H}$ that corresponds to a model with parameter vector $\boldsymbol{\Theta}$, the MML framework accounts for the contribution made by $\boldsymbol{\Theta}$ in quantifying the model complexity. In the "strict" MML formulation (Wallace and Boulton, 1975), given the data $\mathcal{D}$, one is required to find a partition of the data space such that the data within each part is explained by a different model $\boldsymbol{\Theta}$. The partition that minimizes the total message length corresponds to the best description of the data. This problem of partitioning the data and mapping each part to an appropriate model has been shown to be

NP-hard (Farr and Wallace, 2002). As such, it becomes practically difficult to compute the model complexity. As a means of overcoming this problem, Wallace and Freeman (1987) proposed a practical approximation, whereby it becomes possible to compute the model complexity in most cases. This version of the MML formulation forms the foundation on which the model selection problems are investigated in this thesis. The message length formulation and parameter estimation using the Wallace and Freeman (1987) approach is elaborated in Section 2.4.

Minimum Description Length (Rissanen, 1978)

The minimum description length (MDL) principle was proposed by Rissanen (1978). According to the MDL, the model that leads to the best compression of the data should be preferred. It is important to note that in MDL parlance, this postulate is employed in the consideration of a model class, and not a model. Once an optimal model class is inferred, the optimal model within the model class is obtained by employing a maximum likelihood based estimation method.

As described in Oliver and Baxter (1994), the MDL principle selects model classes by choosing the model class with minimum stochastic complexity (Rissanen, 1989). The *stochastic complexity (SC)* for the given observed data $\mathcal{D}$, and a model class with a fixed parametric form $\mathbf{\Theta}$ is defined as

$$SC = -\log \Pr(\mathcal{D}) = -\log \left(\int_{\mathbf{\Theta}} \Pr(\mathcal{D}|\mathbf{\Theta}) \Pr(\mathbf{\Theta}) \, d\mathbf{\Theta} \right)$$

where $\Pr(\mathcal{D})$ is the probability of the observed data (Equation 2.2). Thus, the model class that minimizes the stochastic complexity corresponds to the case when the best compression of the data is achieved.

It is to be noted that each model class has a distinct parametric form and, consequently, the likelihood term $\Pr(\mathcal{D}|\mathbf{\Theta})$ varies for each model class. Further, the above integral can be understood as the expected value (the average) of the likelihood term over all possible models within the model class. Thus, the above formulation enables the selection of a model class that is uniquely identified by a parametric form $\mathbf{\Theta}$.

The feasibility of the explicit computation of the integral relies on the mathematical forms of the prior density $\Pr(\mathbf{\Theta})$ and the likelihood function $\Pr(\mathcal{D}|\mathbf{\Theta})$. However, approximations to minimizing the stochastic complexity as a means of practical MDL formulation have been proposed by Schwarz (1978). This is popularly known as the Bayesian Information Criterion (BIC) and, for large sample sizes N, is given by the following simple-to-use expression:

$$\text{BIC}(p) = \frac{p}{2} \log N + \mathcal{L}(\mathcal{D}|\mathbf{\Theta}) \tag{2.6}$$

where p denotes the number of free parameters in the model class, N is the amount of data, and $\mathcal{L}(\mathcal{D}|\mathbf{\Theta})$ is the *negative* log-likelihood of the data.

The factor $(p/2) \log N$ can be treated as a penalty associated with a model class. The factor can be rearranged such that the probability of each free parameter in the model is given by $\Pr(\theta_i) = 1/\sqrt{N}$, where $\theta_i \in \mathbf{\Theta}$ and $1 \leq i \leq p$, so that its information content is $-\log \Pr(\theta_i) = (1/2) \log N$. For p free parameters, the cumulative information content is, therefore, $(p/2) \log N$. The above expression for BIC is a result of an asymptotic approximation of the integral appearing in the definition of stochastic complexity.

The above formulation suggests that models with greater number of free parameters have greater penalties. Each model class has different number of free parameters, and an associated negative log-likelihood value. Increasing the number of free parameters leads to generalized models that have greater penalties but which also minimize the negative log-likelihood value. This trade-off can be thought of as attempting to balance the model complexity (given by the number of free parameters within a model class) and the goodness-of-fit (given by the negative log-likelihood) in modelling the given data.

Another related information-theoretic criterion proposed by Akaike (1974) called the Akaike Information Criterion (AIC) has a similar formulation as BIC and is given by

$$\text{AIC}(p) = p + \mathcal{L}(\mathcal{D}|\boldsymbol{\Theta}) \tag{2.7}$$

The penalty factor here is given by the number of free parameters p in the model class. Similar to BIC, AIC is also limited to the selection of model classes and not models.

Note that all models within a model class have the same value of p and hence, selecting a model within the model class is equivalent to minimizing the negative log-likelihood expression. Essentially, all models within a model class have the same importance and the selection of the optimal model is done by minimizing the negative log-likelihood expression. As an alternative approach to give unequal importance to individual models, this thesis explores the MML-based inference which not only distinguishes model classes but also provides a unifying framework to differentiate the models.

Minimum Message Length and Algorithmic Complexity

Here, we will explore the strong connection between the MML framework and the concepts from algorithmic complexity research. The formal study and applications of the complexity of the information contained in a body of data, represented by random binary strings, have roots in three distinct streams of thought (Wallace and Dowe, 1999). The first stream that developed the theory of algorithmic probability was conceived by Solomonoff (1964). The second stream conceptualizes the notion of algorithmic complexity and was initiated by Kolmogorov (1965) with important contributions by Chaitin (1966). The third and practically motivated notion of algorithmic complexity was formulated and developed as the minimum message length framework (Wallace and Boulton, 1968; Boulton and Wallace, 1970, 1973; Wallace and Boulton, 1975). These streams have been developed independently of each other and have been conceptualized around the same time, although with different motivating foundations (Wallace and Dowe, 1999).

The different notions of algorithmic complexity are explained by representing a body of information in terms of a binary data string I of finite length $|I|$ and by considering a universal Turing machine (UTM), denoted by T. According to the first stream proposed by Solomonoff (1964), the focus is to model the real-world observed data $\mathcal{D}$ by developing a probability distribution over the set of all possible binary strings $I \in \{I_1, I_2, \ldots\}$. Solomonoff's version considers all strings I that will cause T to produce an output having $\mathcal{D}$ as a prefix. It then defines a probability distributions for $\mathcal{D}$ as

$$P_T(\mathcal{D}) = \sum_{I \in \{I_1, I_2, \ldots\}} 2^{-|I|}$$

In the second stream, Kolmogorov (1965) defines algorithmic complexity of a binary string $\mathcal{D}$ as the length of the *shortest* string I which when given as input to T produces $\mathcal{D}$ as the output. Let the Kolomogorov complexity of $\mathcal{D}$ with respect to T be denoted by $K_T(\mathcal{D})$. As noted in Wallace and Dowe (1999), the above probability sum is dominated by the term for the shortest input string. Consequently, there is a close correspondence between the versions of algorithmic complexity proposed by Kolomogorov and Solomonoff and hence, the probability $P_T(\mathcal{D})$ is often approximated as $2^{-K_T(\mathcal{D})}$.

In the MML stream of thought, the notion of algorithmic complexity is based on Shannon's theory of information rather than with respect to the input to a UTM. According to the MML framework, $\mathcal{D}$ represents some data in the real-world and the goal is to find a *shortest* string I composed of two sub-strings $\mathcal{H}$ and A. The string I corresponds to the total message, the *first part* $\mathcal{H}$ is a hypothesis about $\mathcal{D}$, and the *second part* A is an encoding of the string $\mathcal{D}$ assuming the hypothesis about $\mathcal{D}$ were true. As described previously and highlighted through Equations 2.1 and 2.4, the MML notion integrates Bayesian theory and Shannon's insight to compute the optimal length of the string I, that is,

$$|I| = |\mathcal{H}| + |A| = -\log_2 \Pr(\mathcal{H}) - \log_2 \Pr(\mathcal{D}|\mathcal{H})$$

A major difference between Solomonoff's theoretical construct and that of Kolmogorov and MML's focus is the emphasis on prediction rather on inductive inference. Solomonoff's pursuit of the theory of algorithmic probability is largely motivated by the prediction of future data and thus, requires a probability model $P_T(\mathcal{D})$. On the other hand, Kolmogorov's development of the theory of algorithmic complexity and Wallace and Boulton (1968)'s MML framework were largely driven by model selection and inductive inference. Wallace and Dowe (1999) establish a formal connection between Kolmogorov complexity and the MML framework. They argue that as applied to the inference of models, Kolmogorov complexity and MML are essentially the same. They differ only in the choice of the reference Turing machines. In the first stream, a UTM is considered while in the MML stream, the theory is restricted to a non-universal form in the interest of computational feasibility. While practical realizations of the ideal Solomonoff and Kolomogorov concepts are difficult, the MML paradigm offers a readily computable formalization of the principle of algorithmic complexity. The practical utility of MML has been demonstrated by its applications in machine learning and statistical modelling of real-world data (Wallace, 2005).

It is also worth mentioning that the minimum description length (MDL) framework of Rissanen (1978) is another widely used approach for inductive inference and is closely related to MML in practice. In fact, its relationship with Kolomogorov complexity has been adequately discussed by Li and Vitányi (1997) and Grünwald (2007). However, as described previously, the MDL formulation (Equation 2.6) is a simplified approximation and has limitations in model selection and inference. We now proceed to give an overview of the MML framework, specifically the Wallace and Freeman (1987) approximation, and illustrate the procedure to estimate the parameters of a statistical model under this framework.

2.4 Minimum Message Length Framework (Wallace and Freeman, 1987)

This section describes the formulation of the total message length expression, in terms of encoding the model parameters and the data using those parameters. This is followed by the Wallace and Freeman (1987) approximation which is used as a means to realize the practical computation of the parameter estimates.

2.4.1 Message length formulation

In the MML framework, both the parameters and data need to be communicated between a hypothetical transmitter and a receiver. As a result, the framework takes into account the message length associated with encoding both the parameters and the data using those parameters. Hence, depending on the parameters chosen, the message length varies. The parameters that minimize the total message length are the MML estimates. Let $\boldsymbol{\Theta}$ be the parameter vector that defines a hypothesis and $\mathcal{D} = \{\mathbf{x}_i\}\ \forall\ 1 \leq i \leq N$ such that $\mathbf{x}_i \in \mathbb{R}^d$. The total message length to encode the parameters and the data using the parameters is then given as: $I(\boldsymbol{\Theta}\ \&\ \mathcal{D}) = I(\boldsymbol{\Theta}) + I(\mathcal{D}|\boldsymbol{\Theta})$

Encoding the parameters $I(\boldsymbol{\Theta})$

As noted by Oliver and Baxter (1994), the parameters and the data are limited by their measurement accuracy and can only be stated to a finite precision. The parameter space is partitioned into cells of volume $V(\boldsymbol{\Theta})$ corresponding to the *region of uncertainty* around $\boldsymbol{\Theta}$. Each cell is assigned a unique index and the cell index corresponding to a $\boldsymbol{\Theta}$ is communicated by the transmitter. The receiver decodes the cell index and interprets the parameter as being the center of the cell. This mechanism ensures that parameters are decoded to the precision given by the volume of region of uncertainty. If $h(\boldsymbol{\Theta})$ defines a prior density on the parameter $\boldsymbol{\Theta}$, its probability is approximated as $\Pr(\boldsymbol{\Theta}) \approx h(\boldsymbol{\Theta}) \times V(\boldsymbol{\Theta})$.

Hence, the message length to encode the parameter $\boldsymbol{\Theta}$ is given as

$$I(\boldsymbol{\Theta}) = -\log(h(\boldsymbol{\Theta})V(\boldsymbol{\Theta}))$$

As part of the MML inference, we need to determine the optimal probability of the parameters, which is a function of the prior density $h(\boldsymbol{\Theta})$ and the accuracy of the parameter vector $V(\boldsymbol{\Theta})$. In Bayesian inference problems, $h(\boldsymbol{\Theta})$ is usually assumed based on some domain knowledge. However, determination of an optimal $V(\boldsymbol{\Theta})$ is not straightforward. In Section 2.4.2, we show the computation of $V(\boldsymbol{\Theta})$ based on the Wallace and Freeman (1987) approximation method.

Encoding the data given the parameters $I(\mathcal{D}|\boldsymbol{\Theta})$

If ϵ denotes the accuracy of measurement (AOM) of a datum along each dimension, then its volume of uncertainty is given as ϵ^d. Unlike the volume of uncertainty of the parameters $V(\boldsymbol{\Theta})$ which depends on $\boldsymbol{\Theta}$, the volume of uncertainty of a datum is assumed to be constant and it is known *a priori*. To encode the data, the domain from which the data is sampled is partitioned into cells of equal volume. As with encoding the parameters, each cell is identified by a unique index, and instead of transmitting the actual datum, the transmitter sends the corresponding cell index. If a datum belongs to a particular cell, the probability associated with that cell is computed which is then used to calculate its code length. On the other end, the receiver interprets the datum as being the center of this cell. The mechanism ensures that data is decoded to the precision given by AOM. Assuming the data are sampled independently from a probability density function given by $f(\mathbf{x};\boldsymbol{\Theta})$, the probability of data given the parameters is approximated as $\Pr(\mathcal{D}|\boldsymbol{\Theta}) \approx \prod_{\mathbf{x}_i \in \mathcal{D}} f(\mathbf{x}_i;\boldsymbol{\Theta}) \times \epsilon^d$. Hence, the message length to encode the data given the parameter vector is

$$I(\mathcal{D}|\boldsymbol{\Theta}) = -\log \Pr(\mathcal{D}|\boldsymbol{\Theta}) = -Nd\log\epsilon \underbrace{- \sum_{\mathbf{x}_i \in \mathcal{D}} \log f(\mathbf{x}_i;\boldsymbol{\Theta})}_{\mathcal{L}(\boldsymbol{\Theta})}$$

Note that the expression for $I(\mathcal{D}|\boldsymbol{\Theta})$ is similar to the negative log-likelihood expression $\mathcal{L}(\mathcal{D}|\boldsymbol{\Theta})$ (represented as $\mathcal{L}(\boldsymbol{\Theta})$ in the above equation for easy handling of the subsequent derivation in this section) without the constant term[1].

MML rationalized as a communication framework

The transmitter initially selects a parameter vector $\boldsymbol{\Theta}$, encodes its cell index, and communicates it to the receiver. This constitutes the *first part* of the message. The transmitter then encodes the data using the selected parameter vector and communicates it to the receiver. This forms the *second part* of the message. The receiver first decodes the cell index and uses the center of the cell as the value of the parameter vector, following which the receiver decodes the transmitted data using this parameter vector. Thus, a chosen parameter vector has an associated message length to encode the data.

As the transmitter-receiver pair are concerned with the encoding/decoding of the cell indexes, any change to their respective centers can impact the message lengths significantly (Oliver and Hand, 1994). Such sensitive changes can be due to encoding subtleties that depend on the boundaries of the parameter space, and to changes to the partitioning scheme of the parameter/data space. For this reason, given a parameter vector $\boldsymbol{\Theta}$, the expected value of the message length in the vicinity of $\boldsymbol{\Theta}$ is determined (Wallace and Boulton, 1975) as discussed in the next section. The expected value of the message length computed within the volume $V(\boldsymbol{\Theta})$ is then approximated as the second part of the message, that is, $I(\mathcal{D}|\boldsymbol{\Theta}) \approx \mathbb{E}[I(\mathcal{D}|\boldsymbol{\Theta})]$.

[1]We use a slightly different notation just in this case, where the negative log-likelihood $\mathcal{L}(\mathcal{D}|\boldsymbol{\Theta})$ is represented by $\mathcal{L}(\boldsymbol{\Theta})$ for convenience.

2.4.2 The Wallace-Freeman approximation

In order to decode the message losslessly, the transmitter and receiver must mutually agree upon the encoding details of the parameter space. For this, they should decide on $V(\mathbf{\Theta})$ corresponding to a cell. However, note that encoding $V(\mathbf{\Theta})$ again requires a prior on $V(\mathbf{\Theta})$ and determining the volume of region of uncertainty of $V(\mathbf{\Theta})$, that is, $V(V(\mathbf{\Theta}))$. This leads to an infinite regress and is computationally intractable (Oliver and Baxter, 1994). There are few approximations proposed that are used in practice. Of particular interest is the Wallace and Freeman (1987) approximation. A characteristic feature of this approximation is in the mechanics involved in computing the optimal $V(\mathbf{\Theta})$, and subsequently inferring $\mathbf{\Theta}$ that minimizes the total message length.

We now proceed to formulate the message length expression as per Wallace and Freeman (1987). The following derivation is reproduced from Oliver and Baxter (1994). The expected value of the second part of the message is obtained by integrating $I(\mathcal{D}|\mathbf{\Theta})$ in the region corresponding to $V(\mathbf{\Theta})$, that is,

$$\mathbb{E}[I(\mathcal{D}|\mathbf{\Theta})] = \frac{1}{V(\mathbf{\Theta})} \int_{V(\mathbf{\Theta})} \mathcal{L}(\mathbf{\Theta} + \mathbf{z})\ dV$$

where $(\mathbf{\Theta} + \mathbf{z})$ corresponds to a parameter vector within the area of interest and $\mathcal{L}(\cdot)$ is the negative log-likelihood of the data. Considering up to the quadratic terms in the Taylor series expansion for $\mathcal{L}(\mathbf{\Theta} + \mathbf{z})$, the total message length is approximated as

$$I(\mathbf{\Theta}\ \&\ \mathcal{D}) \approx -\log(V(\mathbf{\Theta})h(\mathbf{\Theta})) + \frac{1}{V(\mathbf{\Theta})} \int_{V(\mathbf{\Theta})} \left(\mathcal{L}(\mathbf{\Theta}) + \mathbf{z}^T \frac{\partial \mathcal{L}(\mathbf{\Theta})}{\partial \mathbf{\Theta}} + \frac{1}{2} \mathbf{z}^T \frac{\partial^2 \mathcal{L}(\mathbf{\Theta})}{\partial \mathbf{\Theta} \partial \mathbf{\Theta}^T} \mathbf{z} \right) dV$$

where $\dfrac{\partial^2 \mathcal{L}(\mathbf{\Theta})}{\partial \mathbf{\Theta} \partial \mathbf{\Theta}^T}$ is the matrix of second-order differentials of $\mathcal{L}(\mathbf{\Theta})$, called the *Hessian* matrix. The *observed Fisher* information matrix is the Hessian matrix of the negative log-likelihood evaluated at the ML estimate. On simplifying, we get

$$I(\mathbf{\Theta}\ \&\ \mathcal{D}) \approx -\log(V(\mathbf{\Theta})h(\mathbf{\Theta})) + \mathcal{L}(\mathbf{\Theta}) + \frac{1}{2V(\mathbf{\Theta})} \int_{V(\mathbf{\Theta})} \mathbf{z}^T \frac{\partial^2 \mathcal{L}(\mathbf{\Theta})}{\partial \mathbf{\Theta} \partial \mathbf{\Theta}^T} \mathbf{z}\ dV$$

The Hessian is dependent on the observed data. However, the receiver does not have knowledge of the data and therefore, the transmitter has to communicate this information. Hence, to make the message decodeable, the *observed* Fisher is replaced by the *expected* Fisher $\mathcal{F}(\mathbf{\Theta})$, that is, $\dfrac{\partial^2 \mathcal{L}(\mathbf{\Theta})}{\partial \mathbf{\Theta} \partial \mathbf{\Theta}^T} \approx \mathbb{E}\left[\dfrac{\partial^2 \mathcal{L}(\mathbf{\Theta})}{\partial \mathbf{\Theta} \partial \mathbf{\Theta}^T} \right]$. This eliminates the infinite regress problem as the precision to which the parameters need to be encoded, that is, $V(\mathbf{\Theta})$ will be independent of the data sample. For a p-dimensional parameter vector $\mathbf{\Theta}$, the elements of the Fisher matrix are given by $\mathcal{F}_{\theta_i \theta_j} = \displaystyle\int_{\mathbf{x}} \frac{\partial^2 \mathcal{L}(\mathbf{\Theta})}{\partial \theta_i \partial \theta_j} f(\mathbf{x}; \mathbf{\Theta})\ d\mathbf{x}$ where θ_i and θ_j are the i^{th} and j^{th} components of the parameter vector. Hence,

$$I(\mathbf{\Theta}\ \&\ \mathcal{D}) \approx -\log(V(\mathbf{\Theta})h(\mathbf{\Theta})) + \mathcal{L}(\mathbf{\Theta}) + \frac{1}{2V(\mathbf{\Theta})} \int_{V(\mathbf{\Theta})} \mathbf{z}^T \mathcal{F}(\mathbf{\Theta}) \mathbf{z}\ dV$$

To simplify the integral, define the transformation $\mathbf{y} = B^{-1}\mathbf{z}$ such that $\mathbf{z}^T \mathcal{F}(\mathbf{\Theta}) \mathbf{z} = \mathbf{y}^T \mathbf{y}$. Let $g(\mathbf{\Phi})$ be the transformed prior density of $h(\mathbf{\Theta})$, and let $U(\mathbf{\Phi})$ be the volume of uncertainty corresponding to $V(\mathbf{\Theta})$. Then,

$$g(\mathbf{\Phi}) = h(\mathbf{\Theta}) \frac{dV(\mathbf{\Theta})}{dU(\mathbf{\Phi})} = \frac{h(\mathbf{\Theta})}{\text{Jacobian}(B^{-1})}$$

As $\mathcal{F}(\mathbf{\Theta})$ is a symmetric and positive semi-definite matrix, $\text{Jacobian}(B^{-1}) = \sqrt{|\mathcal{F}(\mathbf{\Theta})|}$, where $|\cdot|$ is the determinant operator. The reparameterization does not have any effect on the total message length,

as it is shown to be invariant under non-linear transformations (Oliver and Baxter, 1994). Hence,

$$I(\boldsymbol{\Phi} \,\&\, \mathcal{D}) \approx -\log(U(\boldsymbol{\Phi})g(\boldsymbol{\Phi})) + \mathcal{L}(\boldsymbol{\Phi}) + \frac{1}{2}\mathbb{E}\left[\mathbf{y}^T\mathbf{y}\right]$$

Further, $\mathbb{E}\left[\mathbf{y}^T\mathbf{y}\right] = pq_pU(\boldsymbol{\Phi})^{\frac{2}{p}}$, where is q_p is the p-dimensional optimal quantizing lattice constant (Conway and Sloane, 1984), and hence,

$$I(\boldsymbol{\Phi} \,\&\, \mathcal{D}) \approx -\log(U(\boldsymbol{\Phi})g(\boldsymbol{\Phi})) + \mathcal{L}(\boldsymbol{\Phi}) + \frac{p}{2}\kappa_pU(\boldsymbol{\Phi})^{\frac{2}{p}}$$

To find the optimal partitioning which minimizes the total message length, we have

$$\frac{\partial\, I(\boldsymbol{\Phi} \,\&\, \mathcal{D})}{\partial U(\boldsymbol{\Phi})} = 0 \implies U(\boldsymbol{\Phi}) = q_p^{-\frac{p}{2}} \quad \text{and} \quad V(\boldsymbol{\Theta}) = \frac{q_p^{-\frac{p}{2}}}{\sqrt{|\mathcal{F}(\boldsymbol{\Theta})|}}$$

Consequently, $\mathbb{E}\ [\mathbf{y}^T\mathbf{y}] = p$ and $I(\boldsymbol{\Phi} \,\&\, \mathcal{D}) = \frac{p}{2}\log q_p - \log g(\boldsymbol{\Phi}) + \mathcal{L}(\boldsymbol{\Phi}) + \frac{p}{2}$. After substituting the expression for $g(\boldsymbol{\Phi})$ in the above equation, the resulting expression for the total message length becomes

$$I(\boldsymbol{\Theta} \,\&\, \mathcal{D}) = \underbrace{\frac{p}{2}\log q_p - \log\frac{h(\boldsymbol{\Theta})}{\sqrt{|\mathcal{F}(\boldsymbol{\Theta})|}}}_{\text{first part: } I(\boldsymbol{\Theta})} + \underbrace{\mathcal{L}(\boldsymbol{\Theta}) + \frac{p}{2}}_{\text{second part:} I(\mathcal{D}|\boldsymbol{\Theta})} \tag{2.8}$$

Equation 2.8 is the message length formulation according to the Wallace and Freeman (1987) approximation. Minimizing the above equation gives the MML estimate of $\boldsymbol{\Theta}$ as

$$\widehat{\boldsymbol{\Theta}}_{\text{MML}} = \arg\min_{\boldsymbol{\Theta}} I(\boldsymbol{\Theta} \,\&\, \mathcal{D})$$

This thesis explores the practical use of this approximation in inference problems of various probability distributions. More generally, the computation of the MML estimates requires a suitable prior $h(\boldsymbol{\Theta})$ and the computation of the determinant of the Fisher information matrix $|\mathcal{F}(\boldsymbol{\Theta})|$. Based on the final form of $I(\boldsymbol{\Theta} \,\&\, \mathcal{D})$, there might be analytical solutions for the MML estimates, or in the absence of closed-form solutions, some numerical optimization techniques need to be employed to infer the estimates. As an example application of the Wallace and Freeman (1987) approximation that leads to a closed-form analytical estimate, the MML-based inference of the parameters of the univariate Gaussian distribution using the Wallace and Freeman (1987) approximation is discussed next.

2.4.3 MML estimators of the univariate Gaussian distribution

The probability density function of the univariate Gaussian distribution is

$$f(x;\mu,\sigma) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left\{-\frac{(x-\mu)^2}{2\sigma^2}\right\},$$

where $x \in \mathbb{R}$, and $\boldsymbol{\Theta} = (\mu, \sigma)$ are the parameters denoting the mean and standard deviation respectively. Given data $\mathcal{D} = \{x_1, \ldots, x_N\}$, the negative log-likelihood $\mathcal{L}$ is

$$\mathcal{L}(\mathcal{D}|\boldsymbol{\Theta}) = \frac{N}{2}\log(2\pi) + N\log\sigma + \frac{1}{2\sigma^2}\sum_{i=1}^{N}(x_i-\mu)^2$$

The maximum likelihood estimates are obtained by minimizing the negative log-likelihood expression and are given by

$$\hat{\mu}_{\text{ML}} = \frac{1}{N}\sum_{i=1}^{N} x_i \quad \text{and} \quad \hat{\sigma}^2_{\text{ML}} = \frac{1}{N}\sum_{i=1}^{N}(x_i - \hat{\mu}_{\text{ML}})^2$$

Prior density of the parameters: To formulate the message length expression based on the Wallace and Freeman (1987) method, a prior $h(\mu, \sigma)$ should be defined and the determinant of the Fisher information $|\mathcal{F}(\mu, \sigma)|$ needs to be computed. In the context of MML-based inference, Wallace and Boulton (1968) assume that μ and $\log \sigma$ are drawn from a uniform distribution with prespecified ranges R_μ and R_σ respectively. This results in the joint prior density to be $h(\mu, \sigma) = \dfrac{1}{\text{R}_\mu} \times \dfrac{1}{\sigma \text{R}_\sigma}$ (assuming the individual priors are independent).

Fisher information: To compute the determinant of the Fisher matrix, the expectation of the second order partial derivatives of the negative log-likelihood with respect to μ and σ needs to be evaluated. The first order partial derivatives are given by

$$\frac{\partial \mathcal{L}}{\partial \mu} = -\frac{1}{\sigma^2}\sum_{i=1}^{N}(x_i - \mu) \quad \text{and} \quad \frac{\partial \mathcal{L}}{\partial \sigma} = \frac{N}{\sigma} - \frac{1}{\sigma^3}\sum_{i=1}^{N}(x_i - \mu)^2$$

The second order derivatives are, therefore,

$$\frac{\partial^2 \mathcal{L}}{\partial \mu^2} = \frac{N}{\sigma^2} \quad \text{and} \quad \frac{\partial^2 \mathcal{L}}{\partial \sigma^2} = -\frac{N}{\sigma^2} + \frac{3}{\sigma^4}\sum_{i=1}^{N}(x_i - \mu)^2 \quad \text{and} \quad \frac{\partial^2 \mathcal{L}}{\partial \mu \partial \sigma} = \frac{2}{\sigma^3}\sum_{i=1}^{N}(x_i - \mu)$$

It is to be noted that for a random variable X sampled from the Gaussian distribution, the expectations are given by $\text{E}[X] = \mu$ and $\text{E}[(X - \mu)^2] = \sigma^2$. Hence, we have

$$\mathbb{E}\left[\frac{\partial^2 \mathcal{L}}{\partial \mu^2}\right] = \frac{N}{\sigma^2} \quad \text{and} \quad \mathbb{E}\left[\frac{\partial^2 \mathcal{L}}{\partial \sigma^2}\right] = \frac{2N}{\sigma^2} \quad \text{and} \quad \mathbb{E}\left[\frac{\partial^2 \mathcal{L}}{\partial \mu \partial \sigma}\right] = 0$$

$$\text{so that} \quad |\mathcal{F}(\mu, \sigma)| = \begin{vmatrix} \frac{N}{\sigma^2} & 0 \\ 0 & \frac{2N}{\sigma^2} \end{vmatrix} = \frac{2N^2}{\sigma^4}$$

Message length formulation: On substituting the prior $h(\mu, \sigma)$, the determinant of the Fisher $|\mathcal{F}(\mu, \sigma)|$, and the negative log-likelihood $\mathcal{L}$ in Equation 2.8, the total message length expression to encode the parameters $\boldsymbol{\Theta} = (\mu, \sigma)$ and the data $\mathcal{D}$ is given by

$$I(\boldsymbol{\Theta}\ \&\ \mathcal{D}) = (N-1)\log\sigma + \frac{1}{2\sigma^2}\sum_{i=1}^{N}(x_i - \mu)^2 + \text{constant} \tag{2.9}$$

The MML estimates of μ and σ that minimize I correspond to solutions of $\dfrac{\partial I}{\partial \mu} = 0$ and $\dfrac{\partial I}{\partial \sigma} = 0$, and are as follows

$$\hat{\mu}_{\text{MML}} = \frac{1}{N}\sum_{i=1}^{N} x_i \quad \text{and} \quad \hat{\sigma}^2_{\text{MML}} = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \hat{\mu})^2 \tag{2.10}$$

Note that the MML estimate of σ has $(N-1)$ in the denominator, which is the unbiased version of the maximum likelihood estimate.

2.4.4 Statistical consistency of the MML estimator

In this section, a proof is provided for the statistical consistency of the MML parameter estimator derived using the Wallace and Freeman (1987) approximation. Given a parameter space $\boldsymbol{\Theta}_S \subset \mathbb{R}^d$,

let $\boldsymbol{\Theta}_0 \in \boldsymbol{\Theta}_S$ be the true parameter of a probability distribution characterized by $f(\mathbf{x}; \boldsymbol{\Theta})$. Let $\mathcal{D} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ be a sample of size N randomly drawn from the distribution. Then, an estimator $\widehat{\boldsymbol{\Theta}} \in \boldsymbol{\Theta}_S$ is said to be consistent if $\widehat{\boldsymbol{\Theta}} \xrightarrow{\text{a.s.}} \boldsymbol{\Theta}$ as $N \to \infty$, that is, $\widehat{\boldsymbol{\Theta}}$ *almost surely* (a.s.) converges to $\boldsymbol{\Theta}$. Alternatively, if $\widehat{\boldsymbol{\Theta}}$ is obtained by maximizing an objective function, say $M(\mathcal{D}, \boldsymbol{\Theta})$, then for consistency, we need that $\sup_{\boldsymbol{\Theta} \in \boldsymbol{\Theta}_S} \|M(\mathcal{D}, \boldsymbol{\Theta}) - M(\mathcal{D}, \boldsymbol{\Theta}_0)\| \xrightarrow{\text{a.s.}} 0$, where $M(\mathcal{D}, \boldsymbol{\Theta})$ is continuous and has a unique maximum at $\boldsymbol{\Theta} = \boldsymbol{\Theta}_0$ (Newey and McFadden, 1994). So, we need to establish that

$$\Pr\left(\lim_{N \to \infty} \sup_{\boldsymbol{\Theta} \in \boldsymbol{\Theta}_S} \|M(\mathcal{D}, \boldsymbol{\Theta}) - M(\mathcal{D}, \boldsymbol{\Theta}_0)\| = 0\right) = 1 \tag{2.11}$$

In our proof that follows, we invoke the *uniform strong law of large numbers* (Newey and McFadden, 1994), which states that, for a compact parameter space $\boldsymbol{\Theta}_S$ and an upper-continuous function $F(\mathbf{x}, \boldsymbol{\Theta})$, if there exists a function $\phi(\mathbf{x})$ such that $\mathbb{E}[\phi(\mathbf{x})] < \infty$ and $F(\mathbf{x}, \boldsymbol{\Theta}) \leq \phi(\mathbf{x}) \, \forall \, \mathbf{x}, \boldsymbol{\Theta}$, then $\frac{1}{N} \sum_{i=1}^{N} F(\mathbf{x}_i, \boldsymbol{\Theta}) \xrightarrow{\text{a.s.}} \mathbb{E}[F(\mathbf{x}, \boldsymbol{\Theta})]$ uniformly.

Proof. We outline the proof of the convergence of the MML estimator based on the argument of the proof of consistency of the maximum likelihood estimator (Wald, 1949; Lebanon, 2008). The MML estimator $\widehat{\boldsymbol{\Theta}}$ minimizes the message length expression $I(\boldsymbol{\Theta}, \mathcal{D})$ given by Equation 2.8. If we consider $M(\mathcal{D}, \boldsymbol{\Theta}) = -\frac{I(\boldsymbol{\Theta}, \mathcal{D})}{N}$, then $\widehat{\boldsymbol{\Theta}}$ satisfies $M(\mathcal{D}, \widehat{\boldsymbol{\Theta}}) = \sup_{\boldsymbol{\Theta} \in \boldsymbol{\Theta}_S} M(\mathcal{D}, \boldsymbol{\Theta})$. On rearranging the terms in Equation 2.8 and expressing $|\mathcal{F}(\boldsymbol{\Theta})| = N^d |\mathcal{F}_1(\boldsymbol{\Theta})|$, where $|\mathcal{F}_1(\boldsymbol{\Theta})|$ is the determinant of the Fisher information matrix corresponding to a single datum, we have

$$M(\mathcal{D}, \boldsymbol{\Theta}) = -\frac{d}{2N}(1 + \log q_d) + \frac{1}{N} \log \frac{h(\boldsymbol{\Theta})}{\sqrt{|\mathcal{F}_1(\boldsymbol{\Theta})|}} - \frac{d}{2N} \log N + \frac{1}{N} \sum_{i=1}^{N} \log f(\mathbf{x}_i, \boldsymbol{\Theta})$$

The MML estimator equivalently maximizes the following expression for a known $\boldsymbol{\Theta}_0$:

$$M(\mathcal{D}, \boldsymbol{\Theta}) - M(\mathcal{D}, \boldsymbol{\Theta}_0) = \frac{1}{N}\left(\log \frac{h(\boldsymbol{\Theta})}{\sqrt{|\mathcal{F}_1(\boldsymbol{\Theta})|}} - \log \frac{h(\boldsymbol{\Theta}_0)}{\sqrt{|\mathcal{F}_1(\boldsymbol{\Theta}_0)|}}\right) + \frac{1}{N} \sum_{i=1}^{N} (\log f(\mathbf{x}_i, \boldsymbol{\Theta}) - \log f(\mathbf{x}_i, \boldsymbol{\Theta}_0))$$

Let $M(\mathbf{x}, \boldsymbol{\Theta}) = \frac{1}{N} \log \frac{h(\boldsymbol{\Theta})}{\sqrt{|\mathcal{F}_1(\boldsymbol{\Theta})|}} + \log f(\mathbf{x}, \boldsymbol{\Theta})$ and let $F(\mathbf{x}, \boldsymbol{\Theta}) = M(\mathbf{x}, \boldsymbol{\Theta}) - M(\mathbf{x}, \boldsymbol{\Theta}_0)$. Then,

$$M(\mathcal{D}, \boldsymbol{\Theta}) = \frac{1}{N} \sum_{i=1}^{N} M(\mathbf{x}_i, \boldsymbol{\Theta}) + \text{constant, and } \frac{1}{N} \sum_{i=1}^{N} F(\mathbf{x}_i, \boldsymbol{\Theta}) = M(\mathcal{D}, \boldsymbol{\Theta}) - M(\mathcal{D}, \boldsymbol{\Theta}_0) \tag{2.12}$$

Now, if we assume the existence of a suitable $\phi(\mathbf{x})$ such that $\mathbb{E}[\phi(\mathbf{x})]$ is finite and $F(\mathbf{x}, \boldsymbol{\Theta}) \leq \phi(\mathbf{x}) \, \forall \, \mathbf{x}, \boldsymbol{\Theta}$ (Wald, 1949), then $F(\mathbf{x}, \boldsymbol{\Theta})$ satisfies the conditions required for invoking the uniform strong law of large numbers. Further, we have

$$\mathbb{E}[F(\mathbf{x}, \boldsymbol{\Theta})] = 0 + \lim_{N \to \infty} \frac{1}{N} \sum_{i=1}^{N} \log \frac{f(\mathbf{x}_i, \boldsymbol{\Theta})}{f(\mathbf{x}_i, \boldsymbol{\Theta}_0)} = \mathbb{E}_{\boldsymbol{\Theta}_0}\left[\log \frac{f(\mathbf{x}, \boldsymbol{\Theta})}{f(\mathbf{x}, \boldsymbol{\Theta}_0)}\right] = -D_{KL}(f_{\boldsymbol{\Theta}_0} || f_{\boldsymbol{\Theta}})$$

where $D_{KL}(f_{\boldsymbol{\Theta}_0} || f_{\boldsymbol{\Theta}})$ is the KL distance (Section 2.5.2) between the two distributions $f(\mathbf{x}; \boldsymbol{\Theta}_0)$ and $f(\mathbf{x}; \boldsymbol{\Theta})$. As $D_{KL}(f_{\boldsymbol{\Theta}_0} || f_{\boldsymbol{\Theta}}) \geq 0$, we have $\mathbb{E}[F(\mathbf{x}, \boldsymbol{\Theta})] \leq 0$ (equality *only* when $\boldsymbol{\Theta} = \boldsymbol{\Theta}_0$).

For some $\epsilon > 0$, consider the parameter space $S_\epsilon = \{\boldsymbol{\Theta} \in \boldsymbol{\Theta}_S : \|\boldsymbol{\Theta} - \boldsymbol{\Theta}_0\| \geq \epsilon\}$. We note that $\mathbb{E}[F(\mathbf{x}, \boldsymbol{\Theta})]$ is strictly negative in S_ϵ and consequently, the maximum value or $\sup_{\boldsymbol{\Theta} \in S_\epsilon} \mathbb{E}[F(\mathbf{x}_i, \boldsymbol{\Theta})] < 0$.

Thus, by the uniform strong law of large numbers, we can assert that there exists N_ϵ such that $\forall N > N_\epsilon$, we have $\sup_{\boldsymbol{\Theta} \in S_\epsilon} \frac{1}{N} \sum_{i=1}^{N} F(\mathbf{x}_i, \boldsymbol{\Theta}) < 0$ with probability 1. However, since the MML estimator $\widehat{\boldsymbol{\Theta}}$ ensures that Equation 2.12 is maximized, we have $\frac{1}{N} \sum_{i=1}^{N} F(\mathbf{x}_i, \widehat{\boldsymbol{\Theta}}) \geq 0$. This implies that $\widehat{\boldsymbol{\Theta}} \notin S_\epsilon$. Since $\epsilon > 0$ can be made arbitrarily small, we have $\frac{1}{N} \sum_{i=1}^{N} F(\mathbf{x}_i, \widehat{\boldsymbol{\Theta}}) \xrightarrow{\text{a.s.}} 0$, that is, $\Pr\left(\lim_{N \to \infty} \sup_{\boldsymbol{\Theta} \in \boldsymbol{\Theta}_S} \frac{1}{N} \sum_{i=1}^{N} F(\mathbf{x}_i, \boldsymbol{\Theta}) = 0 \right) = 1$. As $F(\mathbf{x}, \boldsymbol{\Theta})$ relates to the message length expression (Equation 2.12), we have showed that Equation 2.11 holds thus, establishing the consistency of the MML estimator. □

In the MML framework, as $N \to \infty$, for a given parameter $\boldsymbol{\Theta}$, the second part of the message corresponding to the negative log-likelihood keeps on increasing while there is only a miniscule increase in the first part. Thus, the contribution to the total message length will be mainly due to the second part. Hence, asymptotically, the MML estimator converges to the ML estimator.

A note on the statistical invariance of the MML estimator: Another important property of the MML estimator is that it is invariant to non-linear invertible transformations of the parameter space (Oliver and Baxter, 1994). If $T(\boldsymbol{\Theta}) = \boldsymbol{\Theta}'$ defines some transformation, then we have $I(\boldsymbol{\Theta}', \mathcal{D}) = I(\boldsymbol{\Theta}, \mathcal{D})$. This is because both the prior density $h(\boldsymbol{\Theta})$ and the Fisher matrix $\mathcal{F}(\boldsymbol{\Theta})$ are also transformed accordingly thus, maintaining the same message length expression. If $\boldsymbol{J}$ denotes the Jacobian matrix of the transformation, that is, $\boldsymbol{J} = \frac{\partial \boldsymbol{\Theta}'}{\partial \boldsymbol{\Theta}}$ and $|\boldsymbol{J}|$ its determinant, then the transformed prior density is given by $h(\boldsymbol{\Theta}') = \frac{h(\boldsymbol{\Theta})}{|\boldsymbol{J}|}$ and the transformed Fisher is $\mathcal{F}(\boldsymbol{\Theta}) = \boldsymbol{J}^T \mathcal{F}(\boldsymbol{\Theta}') \boldsymbol{J}$. The determinant of the transformed Fisher is thus, $|\mathcal{F}(\boldsymbol{\Theta})| = |\boldsymbol{J}|^2 |\mathcal{F}(\boldsymbol{\Theta}')|$, so that $\frac{h(\boldsymbol{\Theta}')}{\sqrt{|\mathcal{F}(\boldsymbol{\Theta}')|}} = \frac{h(\boldsymbol{\Theta})}{\sqrt{|\mathcal{F}(\boldsymbol{\Theta})|}}$. Further, as the negative log-likelihood is also invariant, the total message length expression (Equation 2.8) in the transformed parameter space remains unchanged, that is, $I(\boldsymbol{\Theta}', \mathcal{D}) = I(\boldsymbol{\Theta}, \mathcal{D})$.

2.5 Comparing parameter estimators of different methods

The different estimation methods discussed before, namely the ML, MAP, and MML result in estimators that optimize different objectives. If the negative log-likelihood is used as the comparison criterion, the ML estimators will have a lower value compared to the others. Similarly, the MML estimators will have a lower message length. As each estimation technique optimizes a different objective function, we need to find a metric that impartially evaluates the different estimates. The mean squared error of the estimates and Kullback-Leibler (KL) distance (Kullback and Leibler, 1951) are, therefore, used to compare the various estimates. These two metrics serve as neutral metrics because they are not directly optimized using any of the estimation methods and thus, provide for a fair comparison of the various estimators. The estimators are also compared using statistical hypothesis testing in order to test if the estimators diverge from the true parameter values.

2.5.1 Mean squared error of the estimates

For a parameter vector $\boldsymbol{\Theta}$, and its estimate $\widehat{\boldsymbol{\Theta}}$, the mean squared error (MSE) is given by $\mathbb{E}[(\widehat{\boldsymbol{\Theta}} - \boldsymbol{\Theta})^2]$. Further, the MSE can be decomposed into bias and variance terms, as given below (Lebanon, 2010; Taboga, 2012)

$$\text{MSE} = \mathbb{E}[(\widehat{\boldsymbol{\Theta}} - \boldsymbol{\Theta})^2] = \text{Bias}^2(\widehat{\boldsymbol{\Theta}}) + \text{trace}(\text{Var}(\widehat{\boldsymbol{\Theta}})) \tag{2.13}$$

where $\text{Bias}^2(\widehat{\boldsymbol{\Theta}}) = \|\mathbb{E}[\widehat{\boldsymbol{\Theta}}] - \boldsymbol{\Theta}\|^2$ and $\text{Var}(\widehat{\boldsymbol{\Theta}})$ is the covariance matrix of the estimator.

In the specific case of a parameter vector that contains only one free parameter θ, the expression for MSE, given the parameter estimate $\hat{\theta}$, decomposes as follows

$$\text{MSE} = \mathbb{E}[(\hat{\theta} - \theta)^2] = \underbrace{(\mathbb{E}[\hat{\theta}] - \theta)^2}_{\text{Bias}^2(\hat{\theta})} + \underbrace{\mathbb{E}[(\hat{\theta} - \mathbb{E}[\hat{\theta}])^2]}_{\text{Variance}(\hat{\theta})} \tag{2.14}$$

Ideally, it is expected that the estimates result in low MSE values, which depends on the bias and variance of the parameter estimates. An estimate that results in lower values of MSE is usually preferred over the other estimates.

2.5.2 Kullback-Leibler distance

The Kullback-Leibler (KL) distance (Kullback and Leibler, 1951) is a similarity measure used to determine the "distance" between the true distribution and the distribution with the estimated parameters. Let $f(\mathbf{x}; \boldsymbol{\Theta}_f)$ and $g(\mathbf{x}; \boldsymbol{\Theta}_g)$ be two probability distributions with parameters $\boldsymbol{\Theta}_f$ and $\boldsymbol{\Theta}_g$ respectively. The KL distance between any two distributions is given by

$$D_{KL}(f||g) = \int_{\mathbf{x}} f(\mathbf{x}; \boldsymbol{\Theta}_f) \log \frac{f(\mathbf{x}; \boldsymbol{\Theta}_f)}{g(\mathbf{x}; \boldsymbol{\Theta}_g)} d\mathbf{x} = \text{E}_f \left[\log \frac{f(\mathbf{x}; \boldsymbol{\Theta}_f)}{g(\mathbf{x}; \boldsymbol{\Theta}_g)} \right] \tag{2.15}$$

An estimate that results in a lower KL distance is considered a better estimate. In order to compare the different parameter estimators, the KL distance can be used if the true distribution is known. For a distribution $f(\mathbf{x}; \boldsymbol{\Theta})$, if $\widehat{\boldsymbol{\Theta}}$ corresponds to an estimator, then the KL distance between the true and the distribution with estimated parameters will be

$$D_{KL} = \text{E}_{f(\mathbf{x}, \boldsymbol{\Theta})} \left[\log \frac{f(\mathbf{x}; \boldsymbol{\Theta})}{f(\mathbf{x}; \widehat{\boldsymbol{\Theta}})} \right]$$

2.6 Summary

This chapter discussed the various methods employed in model selection and inference. To select an optimal model to describe the given data, the likelihood ratio test and information-theoretic criteria based MDL, BIC, AIC, and MML are outlined. The scope and limitations of the approaches based on these criteria are elaborated. In summary, the likelihood ratio test is employed in the case of evaluating *nested* models, and fails to generalize well for models (such as mixtures) that do not satisfy this property. Further, the criteria for model selection, such as MDL, BIC, and AIC lead to simplifying assumptions, where the model complexity is treated as a function of just the number of model parameters. The formulations resulting from these criteria indicate that the model parameters do not contribute to quantifying the model complexity. As such, these criteria have some utility in distinguishing among model classes but not among the constituent models.

In order to rigorously account for the model complexity, this chapter described the generalized MML framework, which formulates the message length expression to jointly describe the hypothesis and the data given the hypothesis. The MML framework will be used in the subsequent chapters to estimate the model parameters and to select the model classes. The ability of the MML framework to distinguish among model classes is demonstrated through applications such as mixture modelling (Chapters 5 and 6), selection of optimal number of terms in the decomposition of orthogonal basis functions (Chapter 7), and evaluation of competing representations of protein folding patterns (Chapter 8).

It should be noted that inference using the MML framework does not require the actual code to be constructed, as it is sufficient to know the associated message length. As described in the

previous section, the Wallace and Freeman (1987) method can be used to compute the code length by decomposing the inference problem into two parts. The method involves three steps:

1. Define the prior density of the parameters.
2. Computation of the determinant of the *expected* Fisher information matrix. This requires calculating the expectations of the second order partial derivatives of the negative log-likelihood function of the data with respect to the parameters.
3. Formulation of the two-part message length expression, and its minimization to obtain the MML estimates.

The above approach is used to estimate the parameters of the probability distributions discussed in the following chapters.

Chapter 3

MML inference of Laplace & multivariate Gaussian distributions

3.1 Introduction

In this chapter, we consider the modelling of data in the Euclidean space. The probability distributions that are commonly used in this context are the Laplace and the Gaussian distributions, which are important for statistical inference and for modelling symmetrically distributed data. The multivariate Laplace distributions (Kotz et al., 2001; Eltoft et al., 2006) have not received much attention and, hence, we restrict to the univariate case in this chapter. In contrast, the multivariate Gaussian distributions are widely used and, hence, we consider their multivariate forms.

The Laplace distribution is the model of choice in areas as diverse as signal processing (Eltoft et al., 2006), image denoising (Rabbani et al., 2006), gene expression studies (Bhowmick et al., 2006), market risk prediction (Haas et al., 2006), and machine learning (Cord et al., 2006). This is because the inference of parameters of this distribution is not severely affected by outliers in the data. This is due to the fact that the negative logarithm of the probability density of the Laplace distribution varies *linearly* with the absolute deviation of the data from the mean.

The Gaussian distribution and its mixtures are used in varied statistical pattern recognition tasks such as classification and unsupervised learning of data (Jain and Dubes, 1988). While the Gaussian distribution is sensitive to outliers because of the *quadratic* nature of the contributions of the individual deviations of the data, its computational tractability has motivated its use in several research disciplines (McLachlan and Peel, 2000). The wide applicability of these two distributions motivates their study in this chapter.

Traditionally, in models that use these distributions, the parameters are inferred based on the ML estimation method. However, the ML estimates are known to be biased and issues related with their use in modelling Euclidean data, especially with Gaussian distributions, have been previously documented (Gray, 1994; Lo, 2011). Given the ubiquitous nature of the Laplace and Gaussian distributions, it is important to have estimators that are unbiased and are statistically invariant. To this end, we consider the MML-based inference of the parameters of these distributions.

The Wallace and Freeman (1987) method of parameter estimation has been previously used in the inference of parameters of several probability distributions (Wallace, 2005). We discussed an example in the case of inference of the parameters of the univariate Gaussian distribution in Section 2.4. Others include the multinomial (Boulton and Wallace, 1969), Poisson (Wallace and Dowe, 1994a), von Mises distributions (Wallace and Dowe, 1994b; Dowe et al., 1996c), Student's t-distribution (Agusta and Dowe, 2002), Gamma (Ziou and Bouguila, 2004), generalized Dirichlet (Bouguila and Ziou, 2007), and inverse Gaussian (Schmidt and Makalic, 2012). However, this has not been used for the Laplace or the multivariate Gaussian distributions.

This chapter is organized as follows: we present a derivation of the MML estimators of the parameters of the Laplace distribution in Section 3.2. In this context, we explain the steps involved

in parameter estimation using the Wallace and Freeman (1987) method for the Laplace distribution. The resulting MML estimators are compared with the ML counterparts in Section 3.3. The derived estimators are used in the analysis of the variation in the message lengths and the appropriate selection of a univariate Gaussian or a Laplace distribution depending on the amount of observed data in Section 3.4. This section also examines the behaviour of these distributions when modelling simulated and real-world data. Section 3.5 outlines the MML-based inference of the parameters of the multivariate Gaussian distributions. This section also demonstrates that the resulting MML estimator corresponds to the unbiased estimator that is well known.

3.2 Laplace distribution

The probability density of a Laplace distribution is given by

$$f(x;\mu,b) = \frac{1}{2b}\exp\left\{-\frac{|x-\mu|}{b}\right\},$$

where $x \in \mathbb{R}$ and the parameters $\boldsymbol{\Theta} = (\mu, b)$ are the *location* (mean) and *scale* of the distribution respectively. Given data $\mathcal{D} = \{x_1, \ldots, x_N\}$, the negative log-likelihood $\mathcal{L}$ of the data is

$$\mathcal{L}(\mathcal{D}|\boldsymbol{\Theta}) = N\log(2b) + \frac{1}{b}\sum_{i=1}^{N}|x_i - \mu|$$

The maximum likelihood estimates are obtained by minimizing the negative log-likelihood expression $\mathcal{L}(\mathcal{D}|\boldsymbol{\Theta})$ and are given by

$$\hat{\mu}_{\mathrm{ML}} = \mathrm{median}\{x_1, \ldots, x_N\} \quad \text{and} \quad \hat{b}_{\mathrm{ML}} = \frac{1}{N}\sum_{i=1}^{N}|x_i - \hat{\mu}|$$

Prior density of the parameters: To derive the MML-based estimates using the Wallace and Freeman (1987) method, a prior density on the parameters is chosen in a similar manner as the univariate Gaussian case. Under the assumption that μ and $\log b$ follow a uniform distribution in the intervals R_μ and R_b respectively, and are independent of each other, their joint prior density is given by $h(\mu, b) = \dfrac{1}{\mathrm{R}_\mu} \times \dfrac{1}{b\mathrm{R}_b}$.

Fisher information: While computing the Fisher information associated with the parameters of the Laplace distribution, it is to be noted that the negative log-likelihood function is continuous but not everywhere differentiable with respect to μ. The procedure to compute the Fisher in such a case[1] as described in Daniels (1961) is outlined here. Consider the negative log-likelihood associated with a datum $x \in \mathcal{D}$ given by $\ell(x|\mu, b) = \log(2b) + \dfrac{|x-\mu|}{b}$.

The first derivative of ℓ with respect to μ is $\dfrac{\partial \ell}{\partial \mu} = -\dfrac{1}{b}$ if $x < \mu$ and $\dfrac{\partial \ell}{\partial \mu} = \dfrac{1}{b}$ if $x > \mu$. Therefore, the Fisher information for the location parameter μ is

$$\mathcal{F}(\mu) = \int_{-\infty}^{\mu}\left(\frac{\partial \ell}{\partial \mu}\right)^2 f(x;\mu,b)dx + \int_{\mu}^{\infty}\left(\frac{\partial \ell}{\partial \mu}\right)^2 f(x;\mu,b)dx = \frac{1}{b^2}$$

The first and second derivatives of ℓ with respect to b are

$$\frac{\partial \ell}{\partial b} = \frac{1}{b} - \frac{1}{b^2}|x-\mu| \quad \text{and} \quad \frac{\partial^2 \ell}{\partial b^2} = -\frac{1}{b^2} + \frac{2}{b^3}|x-\mu|$$

[1] `http://www.emakalic.org/blog/?p=71`

For a random variable X sampled from a Laplace distribution, we have $\mathbb{E}\{|X-\mu|\} = b$. Hence, the Fisher information for the scale parameter is $\mathcal{F}(b) = \mathbb{E}\left[\dfrac{\partial^2 \ell}{\partial b^2}\right] = \dfrac{1}{b^2}$. Further, the second order partial derivatives of ℓ with respect to μ and b are $\dfrac{\partial^2 \ell}{\partial \mu \partial b} = -\dfrac{1}{b^2}$ if $x < \mu$ and $\dfrac{\partial^2 \ell}{\partial \mu \partial b} = \dfrac{1}{b^2}$ if $x > \mu$. Using the definition of the expected Fisher and the symmetric nature of the Laplace distribution about μ, we have

$$\mathbb{E}\left[\frac{\partial^2 \ell}{\partial \mu \partial b}\right] = \int_{-\infty}^{\mu} -\frac{1}{b^2}\ f(x;\mu,b)dx + \int_{\mu}^{\infty} \frac{1}{b^2}\ f(x;\mu,b)dx = 0$$

Hence, the determinant of the expected Fisher associated with the entire data $\mathcal{D}$ is

$$|\mathcal{F}(\mu,b)| = \begin{vmatrix} \frac{N}{b^2} & 0 \\ 0 & \frac{N}{b^2} \end{vmatrix} = \frac{N^2}{b^4}$$

Message length formulation: On substituting the prior $h(\mu, b)$, the determinant of the Fisher $|\mathcal{F}(\mu,b)|$, and the negative log-likelihood $\mathcal{L}$ in Equation 2.8, the total message length expression to encode the parameters and the data $\mathcal{D}$ is given by

$$I(\mu, b, \mathcal{D}) = (N-1)\log b + \frac{1}{b}\sum_{i=1}^{N}|x_i - \mu| + \text{constant} \tag{3.1}$$

To obtain the MML estimates $\hat{\mu}_{\text{MML}}$ and $\hat{b}_{\text{MML}}$ that minimize I, we must solve $\dfrac{\partial I}{\partial \mu} = 0$ and $\dfrac{\partial I}{\partial b} = 0$. The MML estimates are, therefore, given by

$$\hat{\mu}_{\text{MML}} = \text{median}\{x_1, \ldots, x_N\} \quad \text{and} \quad \hat{b}_{\text{MML}} = \frac{1}{N-1}\sum_{i=1}^{N}|x_i - \hat{\mu}_{\text{MML}}| \tag{3.2}$$

Note that the MML estimate of b has $(N-1)$ in the denominator, similar to the MML estimate of σ in the case of the Gaussian distribution (Section 2.4.3, Equation 2.10).

3.3 Experimental evaluation of the parameter estimates

The MML estimates are evaluated here by comparing the bias and mean squared error (MSE) values to those obtained with the ML estimates of the Laplace parameters. Additionally, empirical analyses using the MML framework are presented.

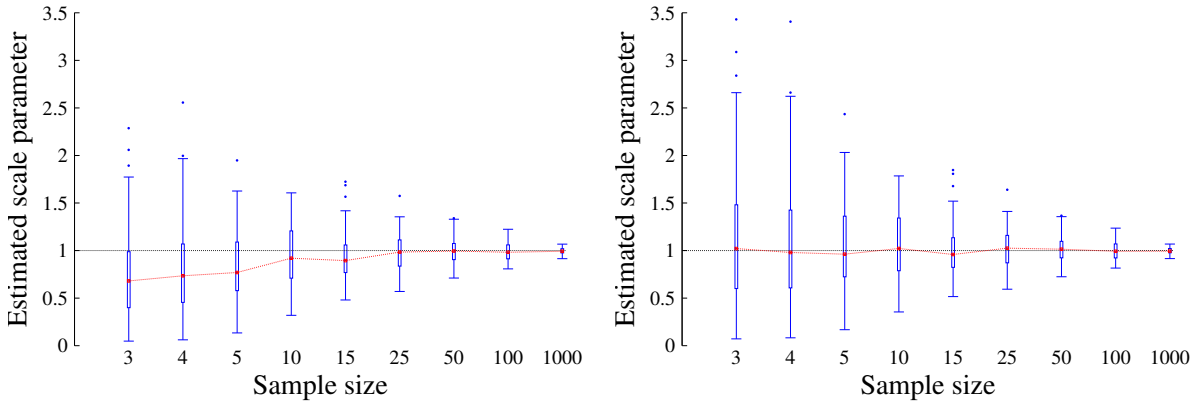
3.3.1 Bias and Mean Squared Error

The MSE of a parameter θ can be decomposed into the sum of bias squared and variance terms (Equation 2.14). The ML estimate of the mean μ is the same as the respective MML estimate for the Laplace distribution. However, the ML and MML estimates of the scale parameter b are different.

Experimental setup: In this section, it is empirically demonstrated that the MML estimator for the Laplace scale parameter minimizes the bias when compared to its ML estimator. Random samples of size $N = \{3, 4, 5, 10, 15, 25, 50, 100, 1000\}$ are generated from a Laplace distribution with mean 0 and scale 1. For each value of N, the ML and MML estimators of the scale parameter are computed and the process is repeated 100 times. The resulting estimators in these 100 iterations are compared using box-whisker plots.

The red curve in Figure 3.1 connects the medians of the estimate values for varying values of N. The ML estimators in Figure 3.1(a) are distant from the true scale value (of 1) when compared with

the MML estimators in Figure 3.1(b). This difference is particularly evident at smaller sample sizes, and becomes insignificant as the sample size increases. Hence, it is observed that the bias in the case of ML estimators is more pronounced, as compared to the MML estimators where the bias is negligible. The MML estimator for the Laplace scale parameter, therefore, serves as an unbiased estimator.



(a) ML estimates (b) MML estimates

Figure 3.1: Comparison of the ML and MML estimators of the Laplace scale parameter.

While bias measures the difference between the expected value of estimator and its actual value, the MSE captures the net average squared error in estimation. Let $\hat{b}_{\text{ML}}$ and $\hat{b}_{\text{MML}}$ be the ML and MML estimates of the scale parameter b, respectively. Table 3.1 shows both the bias squared and the MSE in the estimation of scale parameter for varying values of N averaged over 100 iterations (bold values indicate the best values, that is, smallest bias and MSE). Although the Laplace MML estimator has the smallest bias in comparison to that of the ML estimator, it does not minimize the expected squared error. This is due to the high variability of the estimated scale parameter values. In Figure 3.1(b), for $N = \{3, 4, 5\}$, one can see the relatively large variance in the estimated scale parameter. Since the MSE is a combination of bias and variance, it is greater for the Laplace MML estimator. This is reflected in the MSE values for smaller samples. However, as the sample size increases, both the ML and MML estimators converge to their theoretical values, which for a large N are practically the same.

Table 3.1: Comparison of the ML & MML estimates of the Laplace scale parameter.

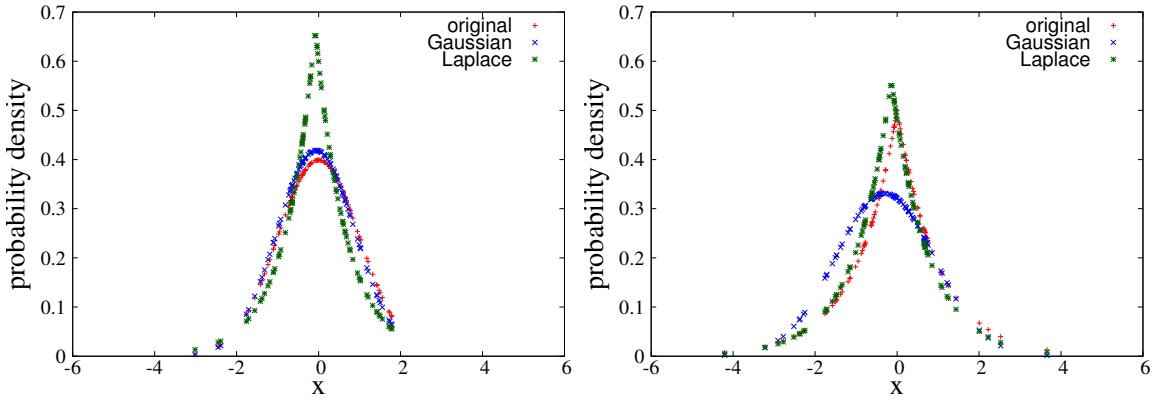
Sample size (N)	Bias squared		Mean squared error	
	$\hat{b}_{\text{ML}}$	$\hat{b}_{\text{MML}}$	$\hat{b}_{\text{ML}}$	$\hat{b}_{\text{MML}}$
3	5.888e-2	**1.851e-2**	**2.925e-1**	5.442e-1
4	3.020e-2	**1.034e-2**	**2.588e-1**	4.166e-1
5	2.767e-2	**1.777e-3**	**1.563e-1**	2.027e-1
10	4.586e-3	**1.530e-3**	**9.640e-2**	1.217e-1
15	5.050e-3	**2.209e-5**	**6.349e-2**	6.711e-2
25	4.558e-4	**3.818e-4**	**3.573e-2**	3.878e-2
50	8.446e-5	**1.208e-4**	**1.500e-2**	1.566e-2
100	1.332e-4	**2.624e-6**	**9.091e-3**	9.148e-3
1000	1.129e-5	**5.617e-6**	9.915e-4	**9.881e-4**

3.3.2 Evaluating the distributions using message length

In this section, the message lengths obtained due to modelling using Gaussian and Laplace distributions are compared. An experiment is conducted as follows: data is randomly sampled from a known

Gaussian or a Laplace distribution, and the parameters of both distributions are inferred separately. It is conceivable that if the true distribution is a Gaussian/Laplace, then the fit to the data would be better if the same distribution type is used. In other words, the compression in message length is better when it is modelled using the same type of distribution. This is expected and is done as a validation check to ensure that the derived MML formulation for the Laplace is consistent with the observation. The optimal code lengths are computed by substituting the MML estimates of Gaussian and Laplace distributions (Equations 2.10 and 3.2) in their respective message length expressions (Equations 2.9 and 3.1).

As an example, a random sample of 100 data points is generated from the two distributions. The true values of the mean μ and the *spread* parameter (standard deviation σ for a Gaussian and scale parameter b for a Laplace) are taken to be 0 and 1 respectively. In Figure 3.2, the probability density of the true distribution corresponding to the sampled data is plotted as a red curve. Using this empirical data, the parameters of the Gaussian and Laplace distributions are estimated using the MML framework. As expected, in Figure 3.2(a), which represents the case when the data is sampled from the Gaussian distribution, the Gaussian fit (blue) overlaps almost entirely with the original distribution (red) and is indicative of a good fit. The Laplace density (green) deviates from the original distribution. The same behaviour is observed in Figure 3.2(b), where the true distribution is Laplace, and naturally, a corresponding Laplace (green) models the data better.



(a) Original distribution: Gaussian

(b) Original distribution: Laplace

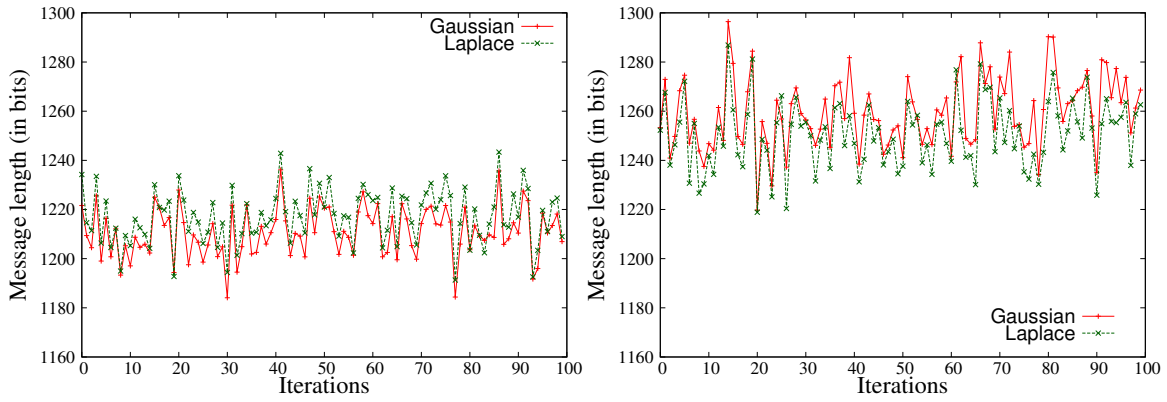
Figure 3.2: Modelling data using both the Gaussian and Laplace distributions.

Table 3.2 provides a comparison of the estimates of the two distributions. For this example, when the true distribution is Gaussian, the corresponding Gaussian fit results in a total message length of ~ 1204 bits, while the Laplace approximation has a total message length of ~ 1208 bits, that is, an additional compression of ~ 4 bits is achieved when modelled using the Gaussian distribution. Thus, as per Equation 2.5, the Gaussian hypothesis is 2^4 times more likely than the Laplace. Similarly, when the true distribution is Laplace, the Laplace model results in a gain of ~ 5 bits in compression over the corresponding Gaussian fit. Thus, in this case, the Laplace hypothesis is 2^5 times more likely compared to the Gaussian model. This example demonstrates how the MML criterion is able to compare two competing hypotheses on some given data.

Table 3.2: Comparison of the MML-based estimates (corresponding to Figure 3.2)

True distribution	True mean	True spread	Gaussian approximation			Laplace approximation		
			$\hat{\mu}$	$\hat{\sigma}$	$I(\hat{\mu},\hat{\sigma},\mathcal{D})$	$\hat{\mu}$	$\hat{b}$	$I(\hat{\mu},\hat{b},\mathcal{D})$
Gaussian	0	1	-0.0618	0.9533	**1204.20**	-0.0785	0.7512	1208.52
Laplace	0	1	-0.2987	1.2047	1237.64	-0.1390	0.8900	**1232.74**

The above experiment is repeated 100 times, and the message lengths corresponding to each simulation are shown in Figure 3.3. The results demonstrate that, the total message length used to encode the observed data is lower when the data is modelled by the same distribution as the underlying one. In some cases, it might happen that a Gaussian distribution would better model data that was generated from a Laplace and vice-versa, as the data is being randomly sampled. However, this is explained in terms of the total message length. The model that results in the least total message length is selected from a set of competing models.



(a) Original distribution: Gaussian

(b) Original distribution: Laplace

Figure 3.3: Comparison of message lengths over 100 iterations

3.4 Discrimination of Gaussian & Laplace distributions

Given data $\mathcal{D} = \{x_1, \ldots, x_N\}$ with no knowledge of the underlying distribution, one needs to determine whether the Laplace or the univariate Gaussian is best suited to model the data. In this context, we demonstrate how the difference in total message lengths due to the two distributions can be used to select a suitable model. For this, we analyze the variation in the total message lengths with increasing amount of data.

For a given data $\mathcal{D}$, let $(\hat{\mu}, \hat{\sigma})$ and $(\hat{\mu}, \hat{b})$ be the MML estimates of the two distributions. Let $I_G(\hat{\mu}, \hat{\sigma}, \mathcal{D})$ and $I_L(\hat{\mu}, \hat{b}, \mathcal{D})$ be the optimal message lengths corresponding to the Gaussian and Laplace distributions respectively. The message length if the data is encoded using the Gaussian distribution is given by Equation 2.9 and the minimum message length expression if the data is encoded using the Laplace distribution is given by Equation 3.1. The optimal message lengths are obtained by substituting the MML estimates of the parameters in these expressions. The difference in their optimal message lengths is

$$\Delta I_{\min} = I_L(\hat{\mu}, \hat{b}, \mathcal{D}) - I_G(\hat{\mu}, \hat{\sigma}, \mathcal{D}) = \frac{N}{2} \log \frac{2}{\pi} - \frac{1}{2} \log 2 + \frac{N-1}{2} + (N-1) \log \hat{b} - (N-1) \log \hat{\sigma}$$

If the Gaussian distribution has a lower message length compared to the Laplace, then $\Delta I_{\min} > 0$. In such a case, as per the MML framework, the Gaussian distribution results in better compression and is preferred over the Laplace to model the given data.

3.4.1 Asymptotic properties of the difference in message lengths

The derivation of the asymptotic distributions for the *logarithm of ratio of maximum likelihood (RML)* is presented in Kundu (2005). A similar approach is adopted here to derive the asymptotic distributions of $\Delta I_{\min}$. The proofs are similar except that the random variable in the current context is $\Delta I_{\min}$ instead

of the logarithm of RML. The theoretical asymptotic results of $\Delta I_{\min}$ are derived here followed by an empirical evaluation of these results.

Case 1: Data follow Gaussian distribution

Based on Theorem 1 of Kundu (2005), it can be concluded that under the assumption that the data follow a Gaussian distribution, the distribution of $\Delta I_{\min}$ is asymptotically normally distributed with mean $\mathbb{E}_G[\Delta I_{\min}]$ and variance $V_G[\Delta I_{\min}]$. Further, as the sample size $N \to \infty$, let the asymptotic MML estimates be $\tilde{\mu}$ and $\tilde{b}$, that is, $\hat{\mu} \to \tilde{\mu}$ and $\hat{b} \to \tilde{b}$, where

$$\mathbb{E}_G[I_L(\tilde{\mu}, \tilde{b}, \mathcal{D})] = \min_{\mu,b} \mathbb{E}_G[I_L(\mu, b, \mathcal{D})]$$

The expressions for $\tilde{\mu}$, $\tilde{b}$, $\mathbb{E}_G[\Delta I]$, and $V_N[\Delta I]$ are derived using Equation 3.1. We have

$$\mathbb{E}_G[I_L(\mu, b, \mathcal{D})] = c_L + (N-1)\log b + \frac{N}{b}\mathbb{E}_G[|X-\mu|]$$

where the constant $c_L = 1 + \log \kappa_2 + \log(\mathrm{R}_\mu \mathrm{R}_b) + \log N + N\log(2/\epsilon)$. The values of $\tilde{\mu}$ and $\tilde{b}$ correspond to those that minimize $\mathbb{E}_G[I_L(\mu, b, \mathcal{D})]$. Since $X \sim \text{Gaussian}(\mu, \sigma)$, therefore,

$$\tilde{\mu} = \mu \quad \text{and} \quad \tilde{b} = \left(\frac{N}{N-1}\right)\sqrt{\frac{2}{\pi}}\,\sigma$$

Since the data is sampled from a Gaussian distribution, the distribution of $\Delta I_{\min}$ will be independent of μ and σ, therefore, without loss of generality, we assume $\mu = 0$ and $\sigma = 1$.

$$\begin{aligned}
\mathbb{E}_G[\Delta I_{\min}] &\approx \mathbb{E}_G[I_L(\tilde{\mu}, \tilde{b}, \mathcal{D}) - I_G(0, 1, \mathcal{D})] \\
&= \mathbb{E}_G\left[\frac{N}{2}\log\left(\frac{2}{\pi}\right) - \frac{1}{2}\sum_{i=1}^{N} x_i^2 + (N-1)\log\tilde{b} + \frac{1}{\tilde{b}}\sum_{i=1}^{N}|x_i - \tilde{\mu}|\right] \\
&= \frac{N}{2}\log\left(\frac{2}{\pi}\right) - \frac{N}{2}\mathbb{E}_G\left[x^2\right] + (N-1)\log\tilde{b} + \frac{N}{\tilde{b}}\mathbb{E}_G[|x|] \\
\frac{\mathbb{E}_G[\Delta I_{\min}]}{N} &= \left(1 - \frac{1}{2N}\right)\log\left(\frac{2}{\pi}\right) + \frac{1}{2} - \frac{1}{N} - \left(\frac{N-1}{N}\right)\log\left(\frac{N-1}{N}\right) \\
\lim_{N\to\infty} \frac{\mathbb{E}_G[\Delta I_{\min}]}{N} &= \log\left(\frac{2}{\pi}\right) + \frac{1}{2} = 0.0484
\end{aligned}$$

$$\begin{aligned}
\text{and} \quad V_G[\Delta I_{\min}] &\approx V_G[I_L(\tilde{\mu}, \tilde{b}, \mathcal{D}) - I_G(0, 1, \mathcal{D})] \\
\frac{V_G[\Delta I_{\min}]}{N} &= V_G\left(-\frac{1}{2}x^2 + \frac{N-1}{N}\sqrt{\frac{\pi}{2}}|x|\right) \\
&= \frac{1}{4}V_G(x^2) + \left(\frac{N-1}{N}\right)^2\frac{\pi}{2}V_G(|x|) - \left(\frac{N-1}{N}\right)\sqrt{\frac{\pi}{2}}\,\mathrm{cov}_G(x^2, |x|) \\
&= \frac{1}{2} + \left(\frac{N-1}{N}\right)^2\frac{\pi}{2}\left(1 - \frac{2}{\pi}\right) - \left(\frac{N-1}{N}\right) \\
\lim_{N\to\infty} \frac{V_G[\Delta I_{\min}]}{N} &= \frac{1}{2} + \frac{\pi}{2}\left(1 - \frac{2}{\pi}\right) - 1 = 0.0708
\end{aligned}$$

Case 2: Data follow Laplace distribution

Based on Theorem 2 of Kundu (2005), it is established that under the assumption that the data is sampled from a Laplace distribution, the distribution of $\Delta I_{\min}$ is asymptotically normally distributed with mean $\mathbb{E}_L[\Delta I_{\min}]$ and variance $V_L[\Delta I_{\min}]$. Also, as the sample size $N \to \infty$, $\hat{\mu} \to \tilde{\mu}$ and $\hat{\sigma} \to \tilde{\sigma}$, where

$$\mathbb{E}_L[I_G(\tilde{\mu}, \tilde{\sigma}, \mathcal{D})] = \min_{\mu,\sigma} \mathbb{E}_L[I_G(\mu, \sigma, \mathcal{D})]$$

To compute $\tilde{\mu}$, $\tilde{\sigma}$, $\mathbb{E}_L[\Delta I_{\min}]$, and $V_L[\Delta I_{\min}]$, we use the message length expression in Equation 2.9 based on which we have

$$\mathbb{E}_L[I_G(\mu, \sigma, \mathcal{D})] = c_G + (N-1)\log\sigma + \frac{N}{2\sigma^2}\mathbb{E}_L[(x-\mu)^2]$$

where the constant $c_G = 1 + \log\kappa_2 + \log(\mathrm{R}_\mu \mathrm{R}_\sigma) + (1/2)\log(2N^2) + (N/2)\log(2\pi/\epsilon^2)$. The expressions for $\tilde{\mu}$ and $\tilde{\sigma}$ correspond to those that minimize $\mathbb{E}_L[I_G(\mu, \sigma, \mathcal{D})]$. Since $X \sim \text{Laplace}(\mu, b)$, we obtain the following result:

$$\tilde{\mu} = \mu \quad \text{and} \quad \tilde{\sigma} = \sqrt{\frac{2N}{N-1}}\, b$$

Since the data follow a Laplace distribution, the distribution of $\Delta I_{\min}$ will be independent of μ and b. Hence, without loss of generality, we can assume $\mu = 0$ and $b = 1$, obtaining

$$\begin{aligned}
\mathbb{E}_L[\Delta I_{\min}] &\approx \mathbb{E}_L[I_L(0,1,\mathcal{D}) - I_G(\tilde{\mu}, \tilde{\sigma}, \mathcal{D})] \\
&= \mathbb{E}_L\left[\frac{N}{2}\log\left(\frac{2}{\pi}\right) - (N-1)\log\tilde{\sigma} - \frac{1}{2\tilde{\sigma}^2}\sum_{i=1}^{N} x_i^2 + \sum_{i=1}^{N}|x_i|\right] \\
&= \frac{N}{2}\log\left(\frac{2}{\pi}\right) - (N-1)\log\tilde{\sigma} - \frac{N}{2\tilde{\sigma}^2}\mathbb{E}_L[x^2] + N\mathbb{E}_L[|x|] \\
\frac{\mathbb{E}_L[\Delta I_{\min}]}{N} &= \frac{1}{2}\log\left(\frac{2}{\pi}\right) - \frac{N-1}{2N}\log\left(\frac{2N}{N-1}\right) + \frac{N+1}{2N} \\
\lim_{N\to\infty} \frac{\mathbb{E}_L[\Delta I_{\min}]}{N} &= -\frac{1}{2}\log\pi + \frac{1}{2} = -0.0724
\end{aligned}$$

$$\begin{aligned}
\text{and} \quad V_L[\Delta I_{\min}] &\approx V_L[I_L(0,1,\mathcal{D}) - I_G(\tilde{\mu}, \tilde{\sigma}, \mathcal{D})] \\
&= V_L\left(-\frac{1}{2\tilde{\sigma}^2}\sum_{i=1}^{N} x_i^2 + \sum_{i=1}^{N}|x_i|\right) \\
\frac{V_L[\Delta I_{\min}]}{N} &= V_L\left(-\frac{N-1}{4N}x^2 + |x|\right) \\
&= \frac{1}{16}\left(\frac{N-1}{N}\right)^2 V_L(x^2) + V_L(|x|) - \left(\frac{N-1}{2N}\right)\,\mathrm{cov}_L(x^2, |x|) \\
&= \frac{5}{4}\left(\frac{N-1}{N}\right)^2 + 1 - 2\left(\frac{N-1}{N}\right) \\
\lim_{N\to\infty} \frac{V_L[\Delta I_{\min}]}{N} &= 0.25
\end{aligned}$$

Discussion about the asymptotic results: In Case 1, the original distribution is Gaussian, and therefore, it is expected that a corresponding Gaussian distribution would better model the data. This is reflected by the fact that $\mathbb{E}_G[\Delta I_{\min}] > 0$. Recall that $\Delta I_{\min}$ corresponds to the difference in optimal message lengths of Laplace and Gaussian. Hence, the observation is valid with what one would expect. Based

Table 3.3: Case 2: Distribution of the difference in minimum message lengths $\Delta I_{\min}$

Sample size (N)	Empirical results		Theoretical results	
	$\frac{E[\Delta I_{\min}]}{N}$	$\frac{V[\Delta I_{\min}]}{N}$	$\frac{E[\Delta I_{\min}]}{N}$	$\frac{V[\Delta I_{\min}]}{N}$
2	0.0243	0.0000	0.1776	0.3125
3	-0.0399	0.0026	0.0747	0.2222
4	0.0078	0.0369	0.0314	0.2031
5	-0.0208	0.0352	0.0077	0.2000
10	-0.0223	0.0993	-0.0351	0.2125
15	-0.0346	0.1086	-0.0481	0.2222
25	-0.0512	0.1497	-0.0581	0.2320
50	-0.0577	0.1885	-0.0653	0.2405
100	-0.0651	0.2024	-0.0689	0.2451
1000	-0.0722	0.2390	-0.0720	0.2495

Table 3.4: A comparison of data compression using ML and MML methods.

Sample size (N)	Encoding using ML	Encoding using MML
2	1059.90	**1000.98**
3	**987.55**	988.24
4	1000.52	**982.631**
5	958.36	**957.52**
10	872.19	**868.61**
15	767.47	**766.52**
25	587.25	**584.27**
50	329.15	**323.14**
100	122.80	**118.34**
1000	3.29e-13	**3.25e-13**

on Equation 2.5, a Gaussian model is $2^{\Delta I_{\min}}$ more likely than a Laplace distribution. In Case 2, the original distribution is Laplace, and hence, the theoretical result that $\mathbb{E}_G[\Delta I_{\min}] < 0$ is in agreement. Again, the log-odds posterior ratio of the Laplace and the Gaussian models is given by $2^{\Delta I_{\min}}$.

Further, for a large sample size N, the distributions of *logarithm of RML* (Kundu, 2005) and $\Delta I_{\min}$ (shown above) converge to the same distribution. This is expected as for large N, the ML and MML estimators converge to the same value. For very large amounts of data, the contribution of the negative log-likelihood to the second part of the total message length outweighs the first part of the message. The theoretical results are validated empirically with an experiment demonstrating the scenario in Case 2. From a known Laplace distribution, samples of varying sizes are generated which are then modelled using both Laplace and Gaussian distributions. The expectation and variance of the optimal message lengths for both distributions are computed over 1000 simulations. The results are tabulated in Table 3.3. The theoretical limits (corresponding to Case 2) are computed using the expressions derived above. It is observed that for larger samples, the experimentally determined expectation and variance of $\Delta I_{\min}$ asymptotically converge to their theoretical limits.

3.4.2 Data Compression (ML vs. MML)

As per the MML framework described in Section 2.4, the posterior log-odds ratio of any two competing hypotheses is related to the difference in message lengths, that is, the additional compression due to the better hypothesis (Equation 2.5). If ML estimators as used, the better hypothesis is determined as the one which results in the maximum value of the likelihood function.

Given data $\mathcal{D}$ and two hypotheses, namely Gaussian and Laplace, one can randomly guess (with equal probability) one of the two distributions that can best model the data. Let this correspond to a null model description of the data. It requires 1 bit to encode a randomly selected distribution. If we were to better the null model, the encoding method, on an average, should be conservative and be less than 1 bit. Since both ML and MML can be used to infer the model probabilities, these can then be converted into information content. This insight is used to measure the average information content associated with using the ML and MML methods. Consider the following experiment:

- Randomly choose a distribution (call it hypothesis $\mathcal{H}$, either Gaussian or Laplace) and select parameters for the distribution from a prespecified range. Generate data from this underlying distribution. The null model information content will be 1 bit.

- Estimate the ML and MML-based parameters of the data using both the distributions and infer $\mathcal{H}'$ as per the ML and MML evaluation criteria. The inferred model will have a probability

higher than its counterpart. Let the inferred model be $\mathcal{H}'$. The information content associated with this will be $I = -\log_2(\Pr(\mathcal{H}'|D))$ bits.

- Since $\mathcal{H}$ is the true distribution, one would expect that the optimal length of encoding to losslessly state the data would be $I = -\log_2(\Pr(\mathcal{H}|D))$ bits (Shannon, 1948). If $\mathcal{H}'$ is not same as $\mathcal{H}$, that would mean $\Pr(\mathcal{H}|D) < \Pr(\mathcal{H}'|D)$ and, therefore, $I = -\log_2(\Pr(\mathcal{H}|D)) > 1$ bit. In such a case, stating the data using a null model would be better. On the contrary, if the inferred model $\mathcal{H}'$ is same as $\mathcal{H}$, then $I < 1$ bit and there is a saving in information (efficient compression).

- Repeat this experiment several times and compute the average value of I. Both ML and MML are expected to perform better than the null model. It is to be noted that this procedure can be used as a basis to judge the degree of compression obtained using ML and MML estimators.

The above experiment is conducted for different values of N and each simulation is repeated 1000 times. Therefore, the null model would result in a total 1000 bits corresponding to each iteration. Both ML and MML are expected to perform better than the null model. The results shown in Table 3.4 are consistent with this expectation except for when the sample size is 2.

When $N = 2$, both ML and MML perform worse than the null model. It is interesting to note that MML is very close to the null. The null model is better on average by 0.98 bits over 1000 iterations. It is small but it demonstrates the difficulty associated with inferring a model using just two data points. When $N = 3$, ML performs marginally better than MML (987.55 vs. 988.24 bits) – a difference of 0.1% is probably not significant. As the sample size increases, the ability to compress data increases for both ML and MML. However, MML consistently performs better than ML. At larger sample sizes ($N = 1000$), the probability of inferring the correct model using both ML and MML is close to 1.

3.4.3 Example applications

This section demonstrates the applicability of MML criterion in choosing one of the Gaussian and Laplace distributions to model some real-world data. The tests of fit for the Laplace distribution using real world datasets were discussed by Puig and Stephens (2000). Their test statistics are based on the empirical distribution function (EDF) and include the families of Cramér-von Mises and Kolmogorov-Smirnov. We also tested the MML approach on a data set that was used by Kundu (2005). We use MML as a differentiating metric and our results are consistent with the results in the sources mentioned.

Example 1: The first data set analyzed by Puig and Stephens (2000) is *breaking strengths of yarn*, which contains 100 samples and was originally presented by Duncan (1974). Puig and Stephens (2000) conclude that the Laplace distribution is a better fit to the data than the Gaussian distribution. They compute EDF statistics and use tables of significance values to reject the Gaussian assumption. This behaviour is consistent if we use the MML formulation to encode the data using Gaussian and Laplace distributions. The respective estimates and the corresponding message lengths are presented in Table 3.5. It is observed that the total message length using Laplace distribution (552.006 bits) is smaller than the message length using the Gaussian distribution (569.736). Hence, the Laplace distribution is favoured as a better fit to the data.

Example 2: The second data set considered is the *flood levels* data (Bain and Engelhardt, 1973), which was discussed by Puig and Stephens (2000). They, however, compared Laplace with a Logistic distribution and concluded that the Laplace model would just be rejected. For our discussion, we will compare the Laplace model with the Gaussian. From the minimum message length in Table 3.5, the Gaussian model would be preferred compared to the Laplace one as there is a difference of ~ 5 bits. This demonstrates that Laplace is not a model of choice to fit this data as per the MML criterion.

Example 3: The third data set considered is the *ball bearings* data (Lawless, 1982) and was analyzed by Kundu (2005). They concluded that Gaussian was a better fit to the data. The message lengths in Table 3.5 confirm this. The minimum message length when the data is encoded using Gaussian distribution is 340.370 bits whereas the length of encoding using a Laplace distribution is 341.869 bits. Hence, Gaussian would be the preferred model. It is interesting to note that the Gaussian model fares better than the Laplace model by mere 1.5 bits.

Data set	Gaussian MML estimates			Laplace MML estimates		
	$\hat{\mu}$	$\hat{\sigma}$	$I(\hat{\mu}, \hat{\sigma}, \mathcal{D})$	$\hat{\mu}$	$\hat{b}$	$I(\hat{\mu}, \hat{b}, \mathcal{D})$
Breaking strengths	99.43	12.46	574.737	99.00	8.414	**557.007**
Flood levels	9.35	4.02	**359.175**	10.13	3.46	364.157
Ball bearings	4.15	0.53	**340.370**	4.21	0.44	341.869

Table 3.5: Selection of Gaussian & Laplace distributions based on their message lengths.

3.5 Multivariate Gaussian distribution

The probability density function of a d-variate Gaussian distribution is given by

$$f(\mathbf{x}; \boldsymbol{\mu}, \mathbf{C}) = \frac{1}{(2\pi)^{\frac{d}{2}} |\mathbf{C}|^{\frac{1}{2}}} \exp\left\{ -\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu})^T \mathbf{C}^{-1} (\mathbf{x} - \boldsymbol{\mu}) \right\}$$

where $\mathbf{x}_i \in \mathbb{R}^d$, $\boldsymbol{\mu}$, $\mathbf{C}$ are the respective mean, covariance matrix, and $|\mathbf{C}|$ is the determinant of the covariance matrix.

3.5.1 Maximum likelihood estimates

Given data $\mathcal{D} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$, the negative log-likelihood $\mathcal{L}$ is given by Equation 3.3. To compute the traditional maximum likelihood estimates, $\mathcal{L}$ needs to be *minimized*. This is achieved by computing the gradient of the negative log-likelihood function with respect to the parameters and solving the resultant equations.

$$\mathcal{L}(\mathcal{D}|\boldsymbol{\mu}, \mathbf{C}) = \frac{Nd}{2} \log(2\pi) + \frac{N}{2} \log|\mathbf{C}| + \frac{1}{2} \sum_{i=1}^{N} (\mathbf{x}_i - \boldsymbol{\mu})^T \mathbf{C}^{-1} (\mathbf{x}_i - \boldsymbol{\mu}) \tag{3.3}$$

The *gradient vector* of $\mathcal{L}$ with respect to $\boldsymbol{\mu}$ and the *gradient matrix* of $\mathcal{L}$ with respect to $\mathbf{C}$ are

$$\nabla_{\boldsymbol{\mu}} \mathcal{L} = \frac{\partial \mathcal{L}}{\partial \boldsymbol{\mu}} = -\sum_{i=1}^{N} \mathbf{C}^{-1} (\mathbf{x}_i - \boldsymbol{\mu}) \tag{3.4}$$

$$\nabla_{\mathbf{C}} \mathcal{L} = \frac{\partial \mathcal{L}}{\partial \mathbf{C}} = \frac{N}{2} \mathbf{C}^{-1} - \frac{1}{2} \sum_{i=1}^{N} \mathbf{C}^{-1} (\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^T \mathbf{C}^{-1}$$

The maximum likelihood estimates are obtained by solving $\nabla_{\boldsymbol{\mu}} \mathcal{L} = 0$ and $\nabla_{\mathbf{C}} \mathcal{L} = 0$, and are

$$\widehat{\boldsymbol{\mu}}_{\mathrm{ML}} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{x}_i \quad \text{and} \quad \widehat{\mathbf{C}}_{\mathrm{ML}} = \frac{1}{N} \sum_{i=1}^{N} (\mathbf{x}_i - \widehat{\boldsymbol{\mu}}_{\mathrm{ML}})(\mathbf{x}_i - \widehat{\boldsymbol{\mu}}_{\mathrm{ML}})^T$$

$\widehat{\mathbf{C}}_{\mathrm{ML}}$ is known to be a biased estimate of the covariance matrix (Fisher, 1925; Rao, 1945; Blackwell, 1947; Barton, 1961; Basu, 1964; Eaton and Morris, 1970; White, 1982). An unbiased estimator of $\mathbf{C}$

was proposed by Barton (1961) and is given as

$$\widehat{\mathbf{C}}_{\text{unbiased}} = \frac{1}{N-1}\sum_{i=1}^{N}(\mathbf{x}_i - \widehat{\boldsymbol{\mu}}_{\text{ML}})(\mathbf{x}_i - \widehat{\boldsymbol{\mu}}_{\text{ML}})^T$$

In addition to the maximum likelihood estimates, Bayesian inference of Gaussian parameters involving conjugate priors over the parameters has also been dealt with in the literature (Bishop, 2006). However, the unbiased estimate of the covariance matrix, as determined by the sample covariance, is typically used in the analysis of Gaussian distributions.

3.5.2 Minimum message length estimation

The MML estimation of parameters of the multivariate Gaussian distribution has been previously attempted by Agusta and Dowe (2003). However, citing mathematical challenges, they used the MMLD approximation (Lam, 2000; Fitzgibbon et al., 2002). Wallace (2005) explored the multivariate Gaussian case using an "informative" conjugate prior.

As described in Section 2.4.2, the message length formulation based on the generalized framework introduced by Wallace and Freeman (1987) requires a reasonable prior $h(\boldsymbol{\Theta})$ on the hypothesis and the computation of the *determinant* of the Fisher information matrix $|\mathcal{F}(\boldsymbol{\Theta})|$ of the *expected* second-order partial derivatives of the negative log-likelihood function, $\mathcal{L}(\mathcal{D}|\boldsymbol{\Theta})$. The MML framework requires the statement of parameters to a finite precision. The optimal precision is related to the Fisher information and in conjunction with a reasonable prior, the probability of parameters is computed.

Prior density of the parameters: A flat prior is usually chosen on each of the d dimensions of $\boldsymbol{\mu}$ (Roberts et al., 1998; Oliver et al., 1996) and a conjugate inverted Wishart prior is chosen for the covariance matrix $\mathbf{C}$ (Gauvain and Lee, 1994; Agusta and Dowe, 2003; Bishop, 2006). The joint prior density of the parameters is then given as $h(\boldsymbol{\mu}, \mathbf{C}) \propto |\mathbf{C}|^{-\frac{d+1}{2}}$. Wallace (2005) provides an alternative version of the prior density which requires the knowledge of previously seen data. In the absence of such information, a non-informative prior is considered.

Fisher information of the parameters: The computation of the Fisher information requires the evaluation of the second order partial derivatives of $\mathcal{L}(\mathcal{D}|\boldsymbol{\mu}, \mathbf{C})$. Let $|\mathcal{F}(\boldsymbol{\mu}, \mathbf{C})|$ represent the determinant of the Fisher information matrix. This is equal to the product of $|\mathcal{F}(\boldsymbol{\mu})|$ and $|\mathcal{F}(\mathbf{C})|$ (Oliver et al., 1996; Roberts et al., 1998), where $|\mathcal{F}(\boldsymbol{\mu})|$ and $|\mathcal{F}(\mathbf{C})|$ are the respective determinants of Fisher information matrices due to the parameters $\boldsymbol{\mu}$ and $\mathbf{C}$.

On differentiating the gradient vector in Equation 3.4 with respect to $\boldsymbol{\mu}$, we get $\nabla^2_{\boldsymbol{\mu}}\mathcal{L} = N\,\mathbf{C}^{-1}$. Consequently, $|\mathcal{F}(\boldsymbol{\mu})| = N^d|\mathbf{C}|^{-1}$. To compute $|\mathcal{F}(\mathbf{C})|$, Magnus and Neudecker (1988) derived an analytical expression using the theory of matrix derivatives based on matrix vectorization (Dwyer, 1967). Let $\mathbf{C} = [c_{ij}] \,\forall\, 1 \leq i, j \leq d$ where c_{ij} is the element corresponding to the i^{th} row and j^{th} column of the matrix. Let $v(\mathbf{C}) = (c_{11}, \ldots, c_{1d}, c_{22}, \ldots, c_{2d}, \ldots, c_{dd})$ be the vector containing the $d(d+1)/2$ free parameters that completely describe the symmetric matrix $\mathbf{C}$. Then, the Fisher information due to the vector of parameters $v(\mathbf{C})$ is equal to $|\mathcal{F}(\mathbf{C})|$ and is given as $N^{\frac{d(d+1)}{2}}2^{-d}|\mathbf{C}|^{-(d+1)}$ (Magnus and Neudecker, 1988; Bozdogan, 1990). Multiplying the Fisher expressions for $\boldsymbol{\mu}$ and $\mathbf{C}$, we get

$$|\mathcal{F}(\boldsymbol{\mu}, \mathbf{C})| = N^{\frac{d(d+3)}{2}}2^{-d}|\mathbf{C}|^{-(d+2)}$$

Message length formulation: To derive the message length expression to encode data using certain $\boldsymbol{\mu}, \mathbf{C}$, we need to substitute the expressions for $h(\boldsymbol{\mu}, \mathbf{C})$, $|\mathcal{F}(\boldsymbol{\mu}, \mathbf{C})|$, and the negative log-likelihood (Equation 3.3) in Equation 2.8 with the number of free parameters as $p = d(d+3)/2$. The message

length expression is then given by

$$I(\boldsymbol{\mu}, \mathbf{C}, \mathcal{D}) = \frac{(N-1)}{2} \log|\mathbf{C}| + \frac{1}{2} \sum_{i=1}^{N} (\mathbf{x}_i - \boldsymbol{\mu})^T \mathbf{C}^{-1} (\mathbf{x}_i - \boldsymbol{\mu}) + \text{constant}$$

To obtain the MML estimates of $\boldsymbol{\mu}$ and $\mathbf{C}$, $I(\boldsymbol{\mu}, \mathbf{C}, \mathcal{D})$ needs to be minimized. The MML estimate of $\boldsymbol{\mu}$ is same as the maximum likelihood estimate. To compute the MML estimate of $\mathbf{C}$, we need to compute the gradient matrix of $I(\boldsymbol{\mu}, \mathbf{C}, \mathcal{D})$ with respect to $\mathbf{C}$. On solving the equation $\nabla_{\mathbf{C}} I = 0$, the MML estimate of $\mathbf{C}$ is obtained as

$$\nabla_{\mathbf{C}} I = \frac{(N-1)}{2} \mathbf{C}^{-1} - \frac{1}{2} \sum_{i=1}^{N} \mathbf{C}^{-1} (\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^T \mathbf{C}^{-1}$$

$$\text{Hence, } \widehat{\mathbf{C}}_{\text{MML}} = \frac{1}{N-1} \sum_{i=1}^{N} (\mathbf{x}_i - \widehat{\boldsymbol{\mu}}_{\text{MML}})(\mathbf{x}_i - \widehat{\boldsymbol{\mu}}_{\text{MML}})^T \quad \text{where} \quad \widehat{\boldsymbol{\mu}}_{\text{MML}} = \widehat{\boldsymbol{\mu}}_{\text{ML}}$$

It is observed that the MML estimate $\widehat{\mathbf{C}}_{\text{MML}}$ is equivalent to the *unbiased* estimate of the covariance matrix $\mathbf{C}$, thus, lending credibility for its preference to model Euclidean data over the traditional maximum likelihood estimate.

3.6 Summary

In this chapter, we considered the modelling of Euclidean data using the Laplace and multivariate Gaussian distributions. These distributions are useful to model symmetrically distributed data. The analysis using these distributions is further facilitated by the computationally tractabe forms of their density functions. To support modelling using these distributions, we must estimate the parameters of the respective distributions. This chapter presented the derivations of the MML estimates of the parameters of of the Laplace and multivariate Gaussian distributions using the Wallace and Freeman (1987) method. The MML estimator of the scale parameter of the Laplace distribution is empirically demonstrated to have lower bias as compared to the ML estimate.

For modelling observed data when the competing models are the Gaussian and Laplace distributions, we have shown how the MML framework can be used in selecting the optimal distribution. We considered the total message lengths due to the MML estimators corresponding to the two distributions and determined the optimal model as the one that results in the least total message length. This was demonstrated when evaluating the suitability of modelling using Laplace and Gaussian distributions in both simulated and real-world applications.

Furthermore, we analyzed the asymptotic variation of the difference in message lengths due to the two distributions. This is useful to get insights into selecting an optimal Gaussian or Laplace depending on the amount of available data. We examined the behaviour of the message lengths in cases corresponding to when the true data distributions are Gaussian and Laplace. It was observed that for very large amounts of data, the limiting distributions of the ML-based logarithm of the ratio of maximized likelihood and the difference in message lengths are the same. In other words, the ML and MML estimates converge asymptotically. The theoretical results were validated through experimental evaluation.

As in the case of the Laplace distribution, we showed that the MML estimator of the covariance matrix of the multivariate Gaussian distribution corresponds to the unbiased estimator. The multivariate Gaussian distribution and the accompanying MML framework will be discussed in Chapter 5 when considering mixture models of the Gaussian distributions.

Chapter 4

MML inference of directional distributions

4.1 Introduction

The previous chapter detailed the MML inference of the Laplace and the multivariate Gaussian distributions. As noted in Chapter 3, these distributions are commonly used for modelling the data in the Euclidean space. However, when modelling data with inherent directionality, the Laplace and the Gaussian distributions are not suitable. Many areas such as earth sciences, meteorology, physics, and biology have data where the *direction* of the constituent vectors is important. This chapter focuses on modelling data using some commonly used directional probability distributions.

The modelling of directional data has been explored using several types of distributions described on the surfaces of compact manifolds, especially spheres and tori (Fisher, 1953, 1993; Mardia and Jupp, 2000). The directional distributions considered in this chapter include the von Mises-Fisher (vMF) defined on any arbitrary d-dimesional unit hypersphere. The vMF distribution is useful for modelling symmetrically distributed data on the spherical surface. Special cases are considered for data distributed on three-dimensional (3D) compact surfaces – the FB_5 or the 5-parameter Fisher-Bingham distribution (also referred to as the Kent distribution) to model data distributed on the surface of a 3D unit sphere, and the bivariate von Mises (BVM) distribution to model data distributed on the surface of a 3D torus.

The vMF, FB_5 and the BVM distributions have practical significance. The models of vMF distributions have been previously used in applications such as text clustering (Banerjee et al., 2005; Gopal and Yang, 2014). The directional data in this case correspond to the normalized representations of text documents which are points on the unit hypersphere. This has motivated the use of multivariate vMF distributions in this context. Other applications include modelling protein data where the directional data correspond to the spatial orientations of the constituent atoms (Dowe et al., 1996a; Banerjee et al., 2005). The modelling of protein directional data has also been done using the FB_5 (Kent and Hamelryck, 2005; Hamelryck et al., 2006) and the BVM distributions (Mardia et al., 2007).

The traditional methods of parameter inference using these distributions are based on ML estimation. However, unlike the Gaussian and the Laplace distributions, the ML parameter estimators of these directional distributions do not have closed-form expressions. The probability density functions have intricate mathematical forms which leads to approximating their parameter estimates. Traditionally, the modelling of the directional data using these probability distributions is based on such approximations. Further, the ML estimators of these directional distributions are known to be biased (Dryden and Mardia, 1998; Dore et al., forthcoming). To address these limitations in this chapter, we consider parameter estimation using the MML framework.

The chapter is organized as follows: the multivariate vMF distribution is discussed in Section 4.2. For the vMF distribution, the commonly used approximations of its concentration parameter κ are reviewed first. This is followed by the derivation of the MML estimate of κ using the Wallace and

Freeman (1987) approach. Dowe et al. (1996c) have demonstrated the superior performance of the MML estimator in the case of a vMF defined on the surface of a 3D sphere. Their work is extended here to derive the MML estimators for a generic d-dimensional vMF distribution.

Section 4.3 details the inference mechanism using the FB_5 distribution, which is a natural extension of the vMF distribution on a 3D sphere, as it is used to model directional data asymmetrically distributed with respect to a mean direction. We explain an intuitive parameterization of the distribution by a geometrical construction of the axes of the distribution. We then outline the procedure to compute the traditionally used moment and ML parameter estimators. For the FB_5 distribution, we also discuss the alternative forms of the probability density function to highlight the problems associated with MAP-based estimation under varying parameterizations. We then describe the procedure to derive the MML estimators. This also includes the methods for numerical computation of some essential derivatives of the normalization constant, that appear in the derivation.

Section 4.4 pertains to the BVM distribution and, in particular, to a specific case of the BVM distribution called the Sine variant. The BVM Sine distribution allows for modelling data that is correlated and thereby serves as a natural analogue of the asymmetric Gaussian distribution but on the surface of a 3D torus. We discuss the traditional ML and MAP-based estimators and proceed to derive the MML estimators. As in the case of the FB_5 distribution, we outline the methods to compute the intricate expressions of normalization constant and its associated derivatives in a numerically stable form.

In all these distributions, we compare our derived MML estimators with the traditionally used estimators. The MML estimators are analyzed by empirically evaluating the bias, mean squared error, the Kullback-Leibler distance and likelihood ratio tests (see Section 2.5). We demonstrate the superior and robust performance of the MML estimators across all these distributions.

4.2 Multivariate von Mises-Fisher distribution

The von Mises-Fisher (vMF) is the most fundamental directional distribution as it is analogous to a *symmetric* Gaussian distribution, wrapped around a unit hypersphere (Watson and Williams, 1956). It is useful for modelling directional data that is symmetrically distributed with respect to a mean direction. Previous studies have established the importance of vMF distributions in mixture modelling and their applications to clustering of protein dihedral angles (Dowe et al., 1996a; Mardia et al., 2007), large-scale text clustering (Banerjee et al., 2003), and gene expression analyses (Banerjee et al., 2005).

The probability density function of a vMF distribution with parameters $\mathbf{\Theta} = (\boldsymbol{\mu}, \kappa) \equiv$ (mean direction, concentration parameter) for a random unit vector $\mathbf{x} \in \mathbb{R}^d$ on a d-dimensional hyperspherical surface $\mathbb{S}^{d-1}$ is given by

$$f(\mathbf{x}; \boldsymbol{\mu}, \kappa) = C_d(\kappa) \exp\{\kappa \boldsymbol{\mu}^T \mathbf{x}\} \tag{4.1}$$

where $C_d(\kappa) = \dfrac{\kappa^{d/2-1}}{(2\pi)^{d/2} I_{d/2-1}(\kappa)}$ is the normalization constant and I_v is a modified Bessel function of the first kind and order v.

A commonly used method of estimation of the parameters of the vMF distribution is the maximum likelihood approach. However, the complex nature of the mathematical form makes it difficult to estimate the concentration parameter κ. This has lead to researchers using many different approximations, as discussed below in Section 4.2.1. Most of these methods perform well when the amount of data is large. At smaller sample sizes, they result in inaccurate estimates of κ and are, thus, unreliable. This is demonstrated by the experiments conducted on a range of sample sizes. The problem is particularly evident when the dimensionality of the data is large. We will rectify this issue by using MML estimates for κ, which for other distributions is shown to be statistically robust. The experiments in Section 4.2.3 demonstrate that the MML estimate of κ provides a more reliable answer and is an improvement on the current state of the art.

4.2.1 Existing methods of parameter estimation

The estimate of the concentration parameter κ is often approximated as discussed below. Given data $\mathcal{D} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$, such that for $i \in \{1, N\}$, $\mathbf{x}_i$ is a d-dimensional datum and $\|\mathbf{x}_i\| = 1$, the negative log-likelihood $\mathcal{L}$ of the data using a vMF distribution is given by

$$\mathcal{L}(\mathcal{D}|\boldsymbol{\mu}, \kappa) = -N \log C_d(\kappa) - \kappa \boldsymbol{\mu}^T \mathbf{R} \tag{4.2}$$

where N is the sample size and $\mathbf{R} = \sum_{i=1}^{N} \mathbf{x}_i$ (the vector sum) such that $\|\mathbf{R}\|$ denotes the magnitude of the resultant vector $\mathbf{R}$. Let $\widehat{\boldsymbol{\mu}}_{\mathrm{ML}}$ and $\widehat{\kappa}_{\mathrm{ML}}$ be the maximum likelihood estimators of $\boldsymbol{\mu}$ and κ respectively. Under the condition that $\widehat{\boldsymbol{\mu}}_{\mathrm{ML}}$ is a unit vector, the negative log-likelihood $\mathcal{L}$ will be minimum when

$$\widehat{\boldsymbol{\mu}}_{\mathrm{ML}} = \frac{\mathbf{R}}{\|\mathbf{R}\|}$$

The maximum likelihood estimator of κ is the solution of the equation $\dfrac{\partial \mathcal{L}}{\partial \kappa} = 0$.

$$\frac{\partial \mathcal{L}}{\partial \kappa} = -N \frac{C'_d(\kappa)}{C_d(\kappa)} - \boldsymbol{\mu}^T \mathbf{R}$$

$$\text{Hence, } \widehat{\kappa}_{\mathrm{ML}} = A_d^{-1}(\bar{R}) \quad \text{where} \quad A_d(\widehat{\kappa}_{\mathrm{ML}}) = -\frac{C'_d(\widehat{\kappa}_{\mathrm{ML}})}{C_d(\widehat{\kappa}_{\mathrm{ML}})} = \frac{\|\mathbf{R}\|}{N} = \bar{R} \tag{4.3}$$

In the above equation, $C'_d(\kappa)$ is the derivative of the normalization constant (see Equation 4.1) with respect to κ. Solving the non-linear equation: $F(\kappa) \equiv A_d(\kappa) - \bar{R} = 0$ yields the corresponding maximum likelihood estimate. For a given κ, $A_d(\kappa)$ can further be simplified in terms of the ratio of Bessel functions and can be represented as

$$A_d(\kappa) = \frac{I_{d/2}(\kappa)}{I_{d/2-1}(\kappa)} \tag{4.4}$$

Because of the difficulties in analytically solving Equation 4.3, there have been several approaches to approximate $\widehat{\kappa}_{\mathrm{ML}}$ (Mardia and Jupp, 2000). Each of these methods is an improvement over their respective predecessors. Tanabe et al. (2007) is an improvement over the estimate proposed by Banerjee et al. (2005). Sra (2012) is an improvement over Tanabe et al. (2007) and Song et al. (2012) fares better when compared to Sra (2012). The common theme in all these methods is that they approximate the maximum likelihood estimate governed by Equation 4.3. The methods are summarized below.

Approximation of Banerjee et al. (2005)

The approximation due to Banerjee et al. (2005) provides an easy to use expression for $\widehat{\kappa}_{\mathrm{ML}}$. The formula is very appealing as it eliminates the need to evaluate complex Bessel functions. Banerjee et al. (2005) demonstrated that this approximation yields better results compared to the ones suggested in Mardia and Jupp (2000). It is an empirical approximation which can be used as a starting point when estimating the root of Equation 4.3.

$$\tilde{\kappa}_{\mathrm{ML}}^{\mathrm{B}} = \frac{\bar{R}(d - \bar{R}^2)}{1 - \bar{R}^2} \tag{4.5}$$

Approximation of Tanabe et al. (2007)

The approximation of Tanabe et al. (2007) utilizes the properties of Bessel functions to determine the lower and upper bounds for $\widehat{\kappa}_{\mathrm{ML}}$. The approach uses a fixed point iteration function in conjunction

with linear interpolation to approximate $\widehat{\kappa}_{\mathrm{ML}}$. The bounds for $\widehat{\kappa}_{\mathrm{ML}}$ are given by

$$\kappa_l = \frac{\bar{R}(d-2)}{1-\bar{R}^2} \leq \widehat{\kappa}_{\mathrm{ML}} \leq \kappa_u = \frac{\bar{R}d}{1-\bar{R}^2}$$

Tanabe et al. (2007) proposed to use a fixed point iteration function defined as $\phi_{2d}(\kappa) = \bar{R}\kappa A_d(\kappa)^{-1}$ and used this to approximate $\widehat{\kappa}_{\mathrm{ML}}$ as

$$\tilde{\kappa}^{\mathrm{T}}_{\mathrm{ML}} = \frac{\kappa_l \phi_{2d}(\kappa_u) - \kappa_u \phi_{2d}(\kappa_l)}{(\phi_{2d}(\kappa_u) - \phi_{2d}(\kappa_l)) - (\kappa_u - \kappa_l)} \tag{4.6}$$

Approximation of Sra (2012)

The heuristic approximation of Sra (2012) involves refining the approximation given by Banerjee et al. (2005) by performing two iterations of Newton's method. Sra (2012) demonstrate that this approximation fares well when compared to the approximation proposed by Tanabe et al. (2007). The following two iterations result in $\tilde{\kappa}^{\mathrm{N}}_{\mathrm{ML}}$, the approximation proposed by Sra (2012):

$$\kappa_1 = \tilde{\kappa}^{\mathrm{B}}_{\mathrm{ML}} - \frac{F(\tilde{\kappa}^{\mathrm{B}}_{\mathrm{ML}})}{F'(\tilde{\kappa}^{\mathrm{B}}_{\mathrm{ML}})} \quad \text{and} \quad \tilde{\kappa}^{\mathrm{N}}_{\mathrm{ML}} = \kappa_1 - \frac{F(\kappa_1)}{F'(\kappa_1)} \tag{4.7}$$

$$\text{where } F'(\kappa) = A'_d(\kappa) = 1 - A_d(\kappa)^2 - \frac{(d-1)}{\kappa} A_d(\kappa) \tag{4.8}$$

Approximation of Song et al. (2012)

This approximation provided by Song et al. (2012) uses Halley's method, which is the second order expansion of Taylor's series of a given function $F(\kappa)$. The higher order approximation results in a more accurate estimate as demonstrated by Song et al. (2012). Halley's method is truncated after iterating through two steps of the root finding algorithm, similar to the method of Sra (2012). The following two iterations result in $\tilde{\kappa}^{\mathrm{H}}_{\mathrm{ML}}$, the approximation proposed by Song et al. (2012):

$$\kappa_1 = \tilde{\kappa}^{\mathrm{B}}_{\mathrm{ML}} - \frac{2F(\tilde{\kappa}^{\mathrm{B}}_{\mathrm{ML}})F'(\tilde{\kappa}^{\mathrm{B}}_{\mathrm{ML}})}{2F'(\tilde{\kappa}^{\mathrm{B}}_{\mathrm{ML}})^2 - F(\tilde{\kappa}^{\mathrm{B}}_{\mathrm{ML}})F''(\tilde{\kappa}^{\mathrm{B}}_{\mathrm{ML}})} \quad \text{and} \quad \tilde{\kappa}^{\mathrm{H}}_{\mathrm{ML}} = \kappa_1 - \frac{2F(\kappa_1)F'(\kappa_1)}{2F'(\kappa_1)^2 - F(\kappa_1)F''(\kappa_1)} \tag{4.9}$$

$$\text{where } F''(\kappa) = A''_d(\kappa) = 2A_d(\kappa)^3 + \frac{3(d-1)}{\kappa} A_d(\kappa)^2 + \frac{(d^2-d-2\kappa^2)}{\kappa^2} A_d(\kappa) - \frac{(d-1)}{\kappa} \tag{4.10}$$

It is to be noted that the ML estimators (of concentration parameter κ) have considerable bias (Schou, 1978; Best and Fisher, 1981; Cordeiro and Vasconcellos, 1999). To address this, the minimum message length based estimation procedure is explored next.

4.2.2 MML-based parameter estimation of d-dimensional vMF

The MML method for parameter estimation has been previously explored for the vMF distributions defined on the 2D and 3D spherical surfaces (Wallace and Dowe, 1994b; Dowe et al., 1996b,c). In this section, we extend the work of Dowe et al. (1996c) for a d-dimensional vMF in conjunction with the Wallace and Freeman (1987) approach to derive the MML estimators.

Prior density of the parameters: Regarding a reasonable prior for the parameters $\boldsymbol{\Theta} = (\boldsymbol{\mu}, \kappa)$ of a vMF distribution, in the absence of any supporting evidence, Wallace and Dowe (1994b) and Dowe et al. (1996c) suggest the use of the following informative prior that is normalizable and locally uniform at the Cartesian origin in κ.

$$h(\boldsymbol{\mu}, \kappa) = h(\boldsymbol{\mu})\, h(\kappa) \propto \frac{\kappa^{d-1}}{(1+\kappa^2)^{\frac{d+1}{2}}}$$

Fisher information of the parameters: Regarding the evaluation of the determinant of the Fisher information, Dowe et al. (1996c) argue that in the general d-dimensional case, the determinant of the Fisher information is

$$|\mathcal{F}(\boldsymbol{\mu},\kappa)| = (N\kappa A_d(\kappa))^{d-1} \times N A'_d(\kappa)$$

where $A_d(\kappa)$ and $A'_d(\kappa)$ are described by Equations 4.4 and 4.8 respectively. The result was intuitively described based on the asymptotic expressions for the variance of the ML estimators of $\widehat{\boldsymbol{\mu}}_{\mathrm{ML}}$ and $\widehat{\kappa}_{\mathrm{ML}}$ (Mardia, 1975a).

Message length formulation: Substituting the negative log-likelihood expression (Equation 4.2), and the expressions for the joint prior density of the parameters and the corresponding Fisher information in Equation 2.8, where the number of free parameters $p = d$, the total message length expression is derived as:

$$I(\boldsymbol{\mu},\kappa,\mathcal{D}) = \frac{(d-1)}{2}\log\frac{A_d(\kappa)}{\kappa} + \frac{1}{2}\log A'_d(\kappa) + \frac{(d+1)}{2}\log(1+\kappa^2) - N\log C_d(\kappa) - \kappa\boldsymbol{\mu}^T\mathbf{R} + \text{constant}$$

To obtain the MML estimates of $\boldsymbol{\mu}$ and κ, the above equation needs to be minimized. The MML estimate for $\boldsymbol{\mu}$ is same as the ML estimate.

$$\widehat{\boldsymbol{\mu}}_{\mathrm{MML}} = \frac{\mathbf{R}}{\|\mathbf{R}\|} \tag{4.11}$$

The resultant equation in κ that needs to be minimized is then given by:

$$I(\kappa) = \frac{(d-1)}{2}\log\frac{A_d(\kappa)}{\kappa} + \frac{1}{2}\log A'_d(\kappa) + \frac{(d+1)}{2}\log(1+\kappa^2) - N\log C_d(\kappa) - \kappa R + \text{constant} \tag{4.12}$$

The MML estimate of κ is obtained by differentiating $I(\kappa)$ with respect to κ and solving the resulting non-linear equation, as follows:

$$\text{Let}\quad G(\kappa) \equiv \frac{\partial I}{\partial \kappa} = -\frac{(d-1)}{2\kappa} + \frac{(d+1)\kappa}{1+\kappa^2} + \frac{(d-1)}{2}\frac{A'_d(\kappa)}{A_d(\kappa)} + \frac{1}{2}\frac{A''_d(\kappa)}{A'_d(\kappa)} + NA_d(\kappa) - R \tag{4.13}$$

The MML estimate is the root of the non-linear equation $G(\widehat{\kappa}_{\mathrm{MML}}) = 0$. However, no closed form solution for $\widehat{\kappa}_{\mathrm{MML}}$ exists. The first and second order approximations of the Taylor series of $G(\widehat{\kappa}_{\mathrm{MML}})$, namely the Newton and Halley method are used to approximate the root of $G(\widehat{\kappa}_{\mathrm{MML}}) = 0$. Both these variants and the effects of the two approximations are discussed in the experimental results below. To be fair and consistent with Sra (2012) and Song et al. (2012), the initial guess of the root is taken as $\tilde{\kappa}^{\mathrm{B}}_{\mathrm{ML}}$ (Equation 4.5) and the methods are iterated twice to obtain the root, that is, the MML estimate $\widehat{\kappa}_{\mathrm{MML}}$.

1. *Approximation using Newton's method:*

$$\kappa_1 = \tilde{\kappa}^{\mathrm{B}}_{\mathrm{ML}} - \frac{G(\tilde{\kappa}^{\mathrm{B}}_{\mathrm{ML}})}{G'(\tilde{\kappa}^{\mathrm{B}}_{\mathrm{ML}})} \quad \text{and} \quad \tilde{\kappa}^{\mathrm{N}}_{\mathrm{MML}} = \kappa_1 - \frac{G(\kappa_1)}{G'(\kappa_1)} \tag{4.14}$$

2. *Approximation using Halley's method:*

$$\kappa_1 = \tilde{\kappa}^{\mathrm{B}}_{\mathrm{ML}} - \frac{2G(\tilde{\kappa}^{\mathrm{B}}_{\mathrm{ML}})G'(\tilde{\kappa}^{\mathrm{B}}_{\mathrm{ML}})}{2G'(\tilde{\kappa}^{\mathrm{B}}_{\mathrm{ML}})^2 - G(\tilde{\kappa}^{\mathrm{B}}_{\mathrm{ML}})G''(\tilde{\kappa}^{\mathrm{B}}_{\mathrm{ML}})} \quad \text{and} \quad \tilde{\kappa}^{\mathrm{H}}_{\mathrm{MML}} = \kappa_1 - \frac{2G(\kappa_1)G'(\kappa_1)}{2G'(\kappa_1)^2 - G(\kappa_1)G''(\kappa_1)} \tag{4.15}$$

The details of evaluating $G'(\kappa)$ and $G''(\kappa)$ are discussed in Appendix A.1.

Equation 4.14 gives the MML estimate $\tilde{\kappa}^{\mathrm{N}}_{\mathrm{MML}}$ using Newton's method, while Equation 4.15 gives the MML estimate $\tilde{\kappa}^{\mathrm{H}}_{\mathrm{MML}}$ using Halley's method.

4.2.3 Evaluation of the MML estimates

In this section, the approximation of the MML estimate using Newton's method $\tilde{\kappa}^{\mathrm{N}}_{\mathrm{MML}}$ and using Halley's method $\tilde{\kappa}^{\mathrm{H}}_{\mathrm{MML}}$ (Equations 4.14 and 4.15), are compared against the traditionally used approximations (discussed in Section 4.2.1). While the estimation of the vMF mean direction is the same across all these methods, the estimation of κ differs and, hence, the corresponding results are presented. Through these experiments, the better performance of the MML estimates as compared to its competitors is demonstrated.

For different values of dimensionality d and concentration parameter κ, random vMF samples of size N are generated using the simulation method of Wood (1994). The parameters of a vMF distribution are estimated using the previously mentioned approximations. Let $\widehat{\kappa} = \{\tilde{\kappa}^{\mathrm{T}}_{\mathrm{ML}}, \tilde{\kappa}^{\mathrm{N}}_{\mathrm{ML}}, \tilde{\kappa}^{\mathrm{H}}_{\mathrm{ML}}, \tilde{\kappa}^{\mathrm{N}}_{\mathrm{MML}}, \tilde{\kappa}^{\mathrm{H}}_{\mathrm{MML}}\}$ denote the estimates of κ corresponding to Tanabe's, truncated Newton (Sra), truncated Halley (Song), MML-Newton, and MML-Halley's approximations, respectively.

Errors in κ estimation: The errors in κ estimation are reported by calculating the absolute error $|\widehat{\kappa} - \kappa|$ and the squared error $(\widehat{\kappa} - \kappa)^2$ averaged over 1000 simulations. The relative error $\frac{|\widehat{\kappa}-\kappa|}{\kappa}$ can be used to measure the percentage error in κ estimation. The following observations are made based on the results shown in Table 4.1.

Table 4.1: Errors in κ estimation. The averages are reported over 1000 simulations for each (N, d, κ) triple. (As before, bold font indicates the best result.)

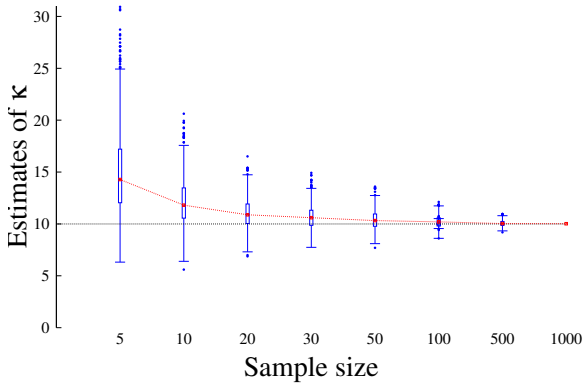
(N, d, κ)	Mean absolute error					Mean squared error				
	Tanabe	Sra	Song	MML		Tanabe	Sra	Song	MML	
	$\tilde{\kappa}^{\mathrm{T}}_{\mathrm{ML}}$	$\tilde{\kappa}^{\mathrm{N}}_{\mathrm{ML}}$	$\tilde{\kappa}^{\mathrm{H}}_{\mathrm{ML}}$	$\tilde{\kappa}^{\mathrm{N}}_{\mathrm{MML}}$	$\tilde{\kappa}^{\mathrm{H}}_{\mathrm{MML}}$	$\tilde{\kappa}^{\mathrm{T}}_{\mathrm{ML}}$	$\tilde{\kappa}^{\mathrm{N}}_{\mathrm{ML}}$	$\tilde{\kappa}^{\mathrm{H}}_{\mathrm{ML}}$	$\tilde{\kappa}^{\mathrm{N}}_{\mathrm{MML}}$	$\tilde{\kappa}^{\mathrm{H}}_{\mathrm{MML}}$
10,10,10	2.501e+0	2.486e+0	2.486e+0	**2.008e+0**	2.012e+0	1.009e+1	9.984e+0	9.984e+0	**5.811e+0**	5.850e+0
10,10,100	1.879e+1	1.877e+1	1.877e+1	**1.316e+1**	**1.316e+1**	5.930e+2	5.920e+2	5.920e+2	**2.800e+2**	2.802e+2
10,10,1000	1.838e+2	1.838e+2	1.838e+2	**1.289e+2**	**1.289e+2**	5.688e+4	5.687e+4	5.687e+4	**2.721e+4**	2.724e+4
10,100,10	2.716e+1	2.716e+1	2.716e+1	2.708e+1	**1.728e+1**	7.464e+2	7.464e+2	7.464e+2	7.414e+2	**4.102e+2**
10,100,100	2.014e+1	2.014e+1	2.014e+1	1.274e+1	**1.265e+1**	4.543e+2	4.543e+2	4.543e+2	2.069e+2	**2.049e+2**
10,100,1000	1.215e+2	1.215e+2	1.215e+2	3.873e+1	**3.870e+1**	1.760e+4	1.760e+4	1.760e+4	2.338e+3	**2.337e+3**
10,1000,10	3.415e+2	3.415e+2	3.415e+2	3.415e+2	**1.386e+2**	1.167e+5	1.167e+5	1.167e+5	1.167e+5	**2.220e+4**
10,1000,100	2.702e+2	2.702e+2	2.702e+2	2.702e+2	**1.652e+2**	7.309e+4	7.309e+4	7.309e+4	7.309e+4	**3.101e+4**
10,1000,1000	1.991e+2	1.991e+2	1.991e+2	1.232e+2	**1.222e+2**	4.014e+4	4.014e+4	4.014e+4	1.570e+4	**1.547e+4**
100,10,10	5.092e-1	5.047e-1	5.047e-1	**4.906e-1**	**4.906e-1**	4.097e-1	4.022e-1	4.022e-1	**3.717e-1**	**3.717e-1**
100,10,100	3.921e+0	3.915e+0	3.915e+0	**3.813e+0**	**3.813e+0**	2.457e+1	2.450e+1	2.450e+1	**2.278e+1**	**2.278e+1**
100,10,1000	3.748e+1	3.747e+1	3.747e+1	**3.669e+1**	**3.669e+1**	2.320e+3	2.319e+3	2.319e+3	**2.174e+3**	**2.174e+3**
100,100,10	4.223e+0	4.223e+0	4.223e+0	3.674e+0	**3.414e+0**	1.862e+1	1.862e+1	1.862e+1	**1.403e+1**	1.420e+1
100,100,100	2.187e+0	2.186e+0	2.186e+0	**1.683e+0**	**1.683e+0**	7.071e+0	7.067e+0	7.067e+0	**4.395e+0**	**4.395e+0**
100,100,1000	1.447e+1	1.447e+1	1.447e+1	**1.129e+1**	**1.129e+1**	3.226e+2	3.226e+2	3.226e+2	**2.027e+2**	**2.027e+2**
100,1000,10	9.150e+1	9.150e+1	9.150e+1	9.146e+1	**8.251e+1**	8.377e+3	8.377e+3	8.377e+3	8.370e+3	**6.970e+3**
100,1000,100	4.299e+1	4.299e+1	4.299e+1	4.882e+1	**4.080e+1**	1.856e+3	1.856e+3	1.856e+3	2.659e+3	**1.738e+3**
100,1000,1000	1.833e+1	1.833e+1	1.833e+1	**8.821e+0**	**8.821e+0**	3.728e+2	3.728e+2	3.728e+2	**1.060e+2**	**1.060e+2**

- For $N = 10, d = 10, \kappa = 10$, the average relative error of $\tilde{\kappa}^{\mathrm{T}}_{\mathrm{ML}}, \tilde{\kappa}^{\mathrm{N}}_{\mathrm{ML}}, \tilde{\kappa}^{\mathrm{H}}_{\mathrm{ML}}$ is $\sim 25\%$, while for $\tilde{\kappa}^{\mathrm{N}}_{\mathrm{MML}}$ and $\tilde{\kappa}^{\mathrm{H}}_{\mathrm{MML}}$, it is $\sim 20\%$. When N is increased to 100, the average relative error of $\tilde{\kappa}^{\mathrm{T}}_{\mathrm{ML}}$ is 5.09%, of $\tilde{\kappa}^{\mathrm{N}}_{\mathrm{ML}}$ and $\tilde{\kappa}^{\mathrm{H}}_{\mathrm{ML}}$ is 5.05%, and of $\tilde{\kappa}^{\mathrm{N}}_{\mathrm{MML}}$ and $\tilde{\kappa}^{\mathrm{H}}_{\mathrm{MML}}$ is 4.9%. We note that increasing N while holding d and κ reduces the error rate across all estimation methods and for all tested combinations of d, κ. This is expected because as more data becomes available, the accuracy of the inference increases. The plots shown in Figure 4.1 reflect this behaviour: while the mean error at lower values of $N = 5, 10, 20, 30$ is noticeable, as N is increased to 1000, there is a drastic drop in the error.

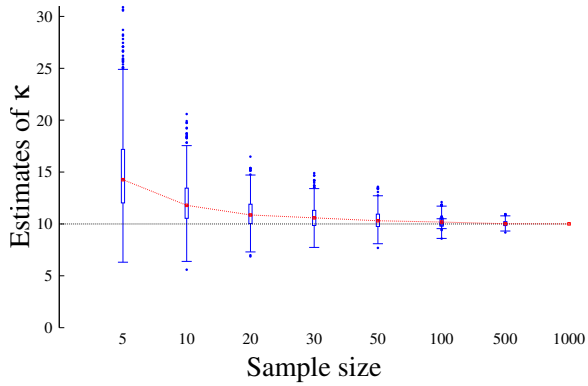
- For a fixed N and d, increasing κ increases the mean absolute error but decreases the average relative error. As an example, for $N = 100, d = 100, \kappa = 10$, the average relative error of $\tilde{\kappa}^{\mathrm{T}}_{\mathrm{ML}}, \tilde{\kappa}^{\mathrm{N}}_{\mathrm{ML}}$, and $\tilde{\kappa}^{\mathrm{H}}_{\mathrm{ML}}$ is $\sim 42\%$, while for $\tilde{\kappa}^{\mathrm{N}}_{\mathrm{MML}}$ and $\tilde{\kappa}^{\mathrm{H}}_{\mathrm{MML}}$ is 36.7% and 34%, respectively. When κ is increased to 100, the error rate for $\tilde{\kappa}^{\mathrm{T}}_{\mathrm{ML}}, \tilde{\kappa}^{\mathrm{N}}_{\mathrm{ML}}$, and $\tilde{\kappa}^{\mathrm{H}}_{\mathrm{ML}}$ drops to 2.18% and for $\tilde{\kappa}^{\mathrm{N}}_{\mathrm{MML}}$ and $\tilde{\kappa}^{\mathrm{H}}_{\mathrm{MML}}$, it drops to 1.68%. Further increasing κ by an order of magnitude to 1000 results in

average relative errors of 1.4% for $\tilde{\kappa}^{\mathrm{T}}_{\mathrm{ML}}, \tilde{\kappa}^{\mathrm{N}}_{\mathrm{ML}}$, and $\tilde{\kappa}^{\mathrm{H}}_{\mathrm{ML}}$, while it is 1.1% for $\tilde{\kappa}^{\mathrm{N}}_{\mathrm{MML}}$ and $\tilde{\kappa}^{\mathrm{H}}_{\mathrm{MML}}$. This indicates that as the data becomes more concentrated, the errors in parameter estimation decrease.

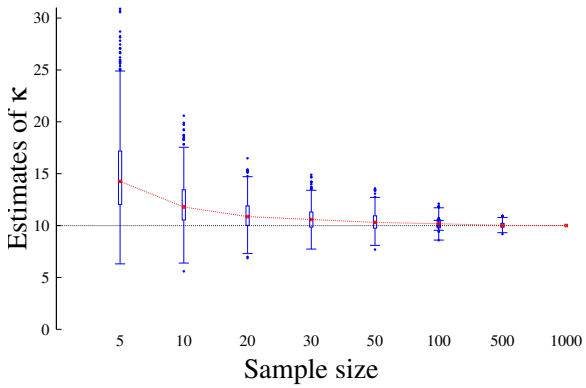
- There does not appear to be a clear pattern of the variation in error rates when d is changed keeping N and κ fixed. However, in any case, MML-based approximations have the least mean absolute and MSE.



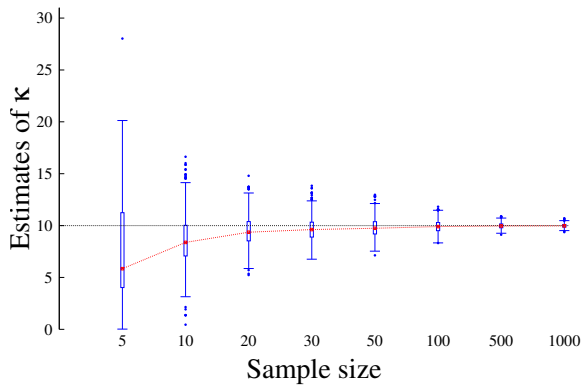
(a) Tanabe estimate ($\tilde{\kappa}^{\mathrm{T}}_{\mathrm{ML}}$)



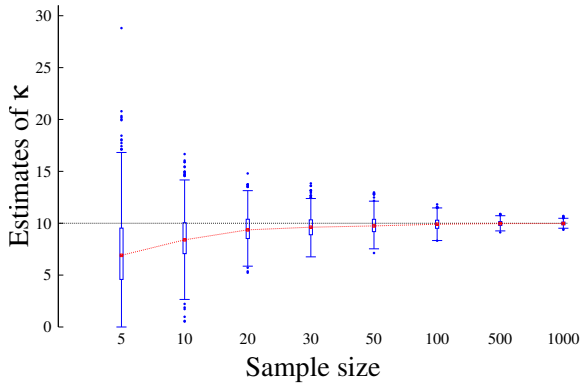
(b) Sra estimate ($\tilde{\kappa}^{\mathrm{N}}_{\mathrm{ML}}$)



(c) Song estimate ($\tilde{\kappa}^{\mathrm{H}}_{\mathrm{ML}}$)



(d) MML (Newton) ($\tilde{\kappa}^{\mathrm{N}}_{\mathrm{MML}}$)



(e) MML (Halley) ($\tilde{\kappa}^{\mathrm{H}}_{\mathrm{MML}}$)

Figure 4.1: Box-whisker plots illustrating the κ estimates as the sample size is gradually increased. True distribution is a 10-dimensional vMF with $\kappa = 10$. The plots are also indicative of the bias due to the estimates.

KL distance and message lengths of the estimates: The quality of parameter inference is further determined by computing the KL distance and the message lengths associated with the parameter estimates. The analytical expression to calculate the KL distance of any two vMF distributions is derived in Appendix A.2. The KL distance is computed between the estimated parameters and the true vMF parameters. The minimum message length expression for encoding data using a vMF distribution is previously derived in Equation 4.12. The average values of both these metrics are listed in Table 4.2. It is observed that the MML estimates of κ result in the least value of KL distance across all combinations of N, d, κ. Also, the message lengths associated with the MML based estimates are the least. From Table 4.2, note that when $N = 10$, $\tilde{\kappa}^{\mathrm{N}}_{\mathrm{MML}}$ and $\tilde{\kappa}^{\mathrm{H}}_{\mathrm{MML}}$ have lower message lengths. For $N = 10, d = 10$, and $\kappa = 10$, $\tilde{\kappa}^{\mathrm{N}}_{\mathrm{MML}}$ and $\tilde{\kappa}^{\mathrm{H}}_{\mathrm{MML}}$ result in extra compression of ~ 1.5 bits over $\tilde{\kappa}^{\mathrm{T}}_{\mathrm{ML}}, \tilde{\kappa}^{\mathrm{N}}_{\mathrm{ML}}$, and $\tilde{\kappa}^{\mathrm{H}}_{\mathrm{ML}}$, which makes the MML estimates $2^{1.5}$ times more likely than the others (as per Equation 2.5) to model the same empirical data.

Table 4.2: Comparison of the κ estimates using KL distance and message length formulation (both metrics are measured in bits).

(N, d, κ)	Average KL distance					Average message length				
	Tanabe	Sra	Song	MML		Tanabe	Sra	Song	MML	
	$\tilde{\kappa}^{\mathrm{T}}_{\mathrm{ML}}$	$\tilde{\kappa}^{\mathrm{N}}_{\mathrm{ML}}$	$\tilde{\kappa}^{\mathrm{H}}_{\mathrm{ML}}$	$\tilde{\kappa}^{\mathrm{N}}_{\mathrm{MML}}$	$\tilde{\kappa}^{\mathrm{H}}_{\mathrm{MML}}$	$\tilde{\kappa}^{\mathrm{T}}_{\mathrm{ML}}$	$\tilde{\kappa}^{\mathrm{N}}_{\mathrm{ML}}$	$\tilde{\kappa}^{\mathrm{H}}_{\mathrm{ML}}$	$\tilde{\kappa}^{\mathrm{N}}_{\mathrm{MML}}$	$\tilde{\kappa}^{\mathrm{H}}_{\mathrm{MML}}$
10,10,10	8.777e-1	8.750e-1	8.750e-1	**6.428e-1**	6.445e-1	9.285e+2	9.285e+2	9.285e+2	**9.269e+2**	**9.269e+2**
10,10,100	8.803e-1	8.798e-1	8.798e-1	**7.196e-1**	7.199e-1	8.214e+2	8.214e+2	8.214e+2	**8.208e+2**	**8.208e+2**
10,10,1000	9.006e-1	9.005e-1	9.005e-1	**7.443e-1**	7.446e-1	6.925e+2	6.925e+2	6.925e+2	**6.919e+2**	**6.919e+2**
10,100,10	8.517e+0	8.517e+0	8.517e+0	8.479e+0	**5.321e+0**	8.633e+3	8.633e+3	8.633e+3	8.633e+3	**8.585e+3**
10,100,100	8.444e+0	8.444e+0	8.444e+0	**6.007e+0**	6.009e+0	8.428e+3	8.428e+3	8.428e+3	**8.414e+3**	**8.414e+3**
10,100,1000	8.472e+0	8.472e+0	8.472e+0	**7.118e+0**	7.120e+0	7.274e+3	7.274e+3	7.274e+3	**7.269e+3**	**7.269e+3**
10,1000,10	8.433e+1	8.433e+1	8.433e+1	8.433e+1	**1.777e+1**	7.030e+4	7.030e+4	7.030e+4	7.030e+4	**6.925e+4**
10,1000,100	8.430e+1	8.430e+1	8.430e+1	8.430e+1	**4.697e+1**	7.030e+4	7.030e+4	7.030e+4	7.030e+4	**6.989e+4**
10,1000,1000	8.451e+1	8.451e+1	8.451e+1	**5.976e+1**	5.977e+1	6.825e+4	6.825e+4	6.825e+4	**6.811e+4**	**6.811e+4**
100,10,10	7.409e-2	7.385e-2	7.385e-2	**7.173e-2**	**7.173e-2**	**9.115e+3**	**9.115e+3**	**9.115e+3**	**9.115e+3**	**9.115e+3**
100,10,100	7.539e-2	7.535e-2	7.535e-2	**7.411e-2**	**7.411e-2**	**7.858e+3**	**7.858e+3**	**7.858e+3**	**7.858e+3**	**7.858e+3**
100 10,1000	7.271e-2	7.271e-2	7.271e-2	**7.161e-2**	**7.161e-2**	**6.403e+3**	**6.403e+3**	**6.403e+3**	**6.403e+3**	**6.403e+3**
100,100,10	7.270e-1	7.270e-1	7.270e-1	**6.146e-1**	6.208e-1	8.615e+4	8.615e+4	8.615e+4	**8.614e+4**	**8.614e+4**
100,100,100	7.357e-1	7.357e-1	7.357e-1	**7.117e-1**	**7.117e-1**	**8.299e+4**	**8.299e+4**	**8.299e+4**	**8.299e+4**	**8.299e+4**
100,100,1000	7.330e-1	7.330e-1	7.330e-1	**7.210e-1**	**7.210e-1**	**6.976e+4**	**6.976e+4**	**6.976e+4**	**6.976e+4**	**6.976e+4**
100,1000,10	7.324e+0	7.324e+0	7.324e+0	7.318e+0	**6.201e+0**	7.024e+5	7.024e+5	7.024e+5	7.024e+5	**7.023e+5**
100,1000,100	7.302e+0	7.302e+0	7.302e+0	**7.045e+0**	7.106e+0	7.022e+5	7.022e+5	7.022e+5	**7.019e+5**	7.022e+5
100,1000,1000	7.340e+0	7.340e+0	7.340e+0	**7.097e+0**	**7.097e+0**	**6.707e+5**	**6.707e+5**	**6.707e+5**	**6.707e+5**	**6.707e+5**

Bias of the parameter estimates: The maximum likelihood estimate of κ is known to have significant bias (Schou, 1978; Best and Fisher, 1981; Cordeiro and Vasconcellos, 1999). The goal here is to demonstrate that MML-based parameter approximations result in estimates with reduced bias. The MSE in Table 4.1 can be decomposed into the sum of bias and variance terms shown in Equation 2.14 (Taboga, 2012). Table 4.3 shows the bias-variance decomposition of the errors when estimating the concentration parameter in the above simulations. The bias of $\tilde{\kappa}^{\mathrm{N}}_{\mathrm{MML}}$ and $\tilde{\kappa}^{\mathrm{H}}_{\mathrm{MML}}$ is lower compared to the other estimates. The variance of the MML estimates, however, is not always the smallest, as observed in Table 4.3. The combination of bias and variance, which is the mean squared error (MSE), is empirically demonstrated to be the least for the MML estimates.

Statistical hypothesis testing: There have been several goodness-of-fit methods proposed in the literature to test the null hypothesis of a vMF distribution against some alternative hypothesis (Kent, 1982; Mardia et al., 1984; Mardia and Jupp, 2000). Recently, Figueiredo (2012) suggested tests for the specific case of concentrated vMF distributions. Here, the behaviour of κ estimates for generic vMF distributions as proposed by Mardia et al. (1984) is examined. They derived a likelihood ratio test for the null hypothesis of a vMF distribution ($\mathcal{H}_0$) against the alternative of a Fisher-Bingham distribution ($\mathcal{H}_a$). The asymptotically equivalent Rao's score statistic (Rao, 1973) was used to test the hypothesis.

The score statistic $\mathcal{W}$, in this case, is a function of the concentration parameter. It has an asymptotic $\chi^2(p)$ distribution (with degrees of freedom $p = \frac{1}{2}d(d+1) - 1$) under $\mathcal{H}_0$ as the sample size

Table 4.3: Bias-variance decomposition of the squared error $(\widehat{\kappa} - \kappa)^2$.

(N, d, κ)	Bias squared					Variance				
	Tanabe	Sra	Song	MML		Tanabe	Sra	Song	MML	
	$\tilde{\kappa}^{\mathrm{T}}_{\mathrm{ML}}$	$\tilde{\kappa}^{\mathrm{N}}_{\mathrm{ML}}$	$\tilde{\kappa}^{\mathrm{H}}_{\mathrm{ML}}$	$\tilde{\kappa}^{\mathrm{N}}_{\mathrm{MML}}$	$\tilde{\kappa}^{\mathrm{H}}_{\mathrm{MML}}$	$\tilde{\kappa}^{\mathrm{T}}_{\mathrm{ML}}$	$\tilde{\kappa}^{\mathrm{N}}_{\mathrm{ML}}$	$\tilde{\kappa}^{\mathrm{H}}_{\mathrm{ML}}$	$\tilde{\kappa}^{\mathrm{N}}_{\mathrm{MML}}$	$\tilde{\kappa}^{\mathrm{H}}_{\mathrm{MML}}$
10,10,10	5.609e+0	5.520e+0	5.520e+0	1.299e+0	**1.269e+0**	4.476e+0	**4.464e+0**	**4.464e+0**	4.512e+0	4.581e+0
10,10,100	2.298e+2	2.288e+2	2.288e+2	4.986e-3	**2.577e-4**	3.632e+2	3.632e+2	3.632e+2	**2.800e+2**	2.802e+2
10,10,1000	2.157e+4	2.156e+4	2.156e+4	**2.764e+1**	3.193e+1	3.531e+4	3.531e+4	3.531e+4	**2.718e+4**	2.720e+4
10,100,10	7.378e+2	7.378e+2	7.378e+2	7.333e+2	**2.875e+2**	8.660e+0	8.660e+0	8.660e+0	**8.066e+0**	1.226e+2
10,100,100	4.054e+2	4.053e+2	4.053e+2	1.546e+2	**1.522e+2**	**4.894e+1**	**4.894e+1**	**4.894e+1**	5.231e+1	5.273e+1
10,100,1000	1.473e+4	1.473e+4	1.473e+4	2.207e+1	**1.994e+1**	2.870e+3	2.870e+3	2.870e+3	**2.316e+3**	2.317e+3
10,1000,10	1.166e+5	1.166e+5	1.166e+5	1.166e+5	**1.921e+4**	**8.090e+1**	**8.090e+1**	**8.090e+1**	**8.090e+1**	2.983e+3
10,1000,100	7.301e+4	7.301e+4	7.301e+4	7.300e+4	**2.728e+4**	8.685e+1	8.685e+1	8.685e+1	**8.635e+1**	3.735e+3
10,1000,1000	3.964e+4	3.964e+4	3.964e+4	1.517e+4	**1.493e+4**	**4.969e+2**	**4.969e+2**	**4.969e+2**	5.306e+2	5.342e+2
100,10,10	4.129e-2	3.528e-2	3.528e-2	8.132e-3	**8.129e-3**	3.684e-1	3.669e-1	3.669e-1	**3.636e-1**	**3.636e-1**
100,10,100	1.280e+0	1.206e+0	1.206e+0	5.505e-2	**5.504e-2**	2.329e+1	2.329e+1	2.329e+1	**2.273e+1**	**2.273e+1**
100,10,1000	9.796e+1	9.728e+1	9.728e+1	6.620e+0	**6.619e+0**	2.222e+3	2.222e+3	2.222e+3	2.168e+3	**2.168e+3**
100,100,10	1.783e+1	1.783e+1	1.783e+1	**4.661e+0**	6.202e+0	**7.807e-1**	**7.807e-1**	**7.807e-1**	9.369e+0	8.003e+0
100,100,100	3.371e+0	3.367e+0	3.367e+0	7.147e-1	**7.146e-1**	3.700e+0	3.700e+0	3.700e+0	**3.681e+0**	**3.681e+0**
100,100,1000	1.161e+2	1.161e+2	1.161e+2	**3.504e-1**	**3.504e-1**	2.065e+2	2.065e+2	2.065e+2	**2.023e+2**	**2.023e+2**
100,1000,10	8.372e+3	8.372e+3	8.372e+3	8.364e+3	**6.809e+3**	5.385e+0	5.385e+0	5.385e+0	**5.200e+0**	1.614e+2
100,1000,100	1.848e+3	1.848e+3	1.848e+3	**5.143e+2**	1.628e+3	**7.656e+0**	**7.656e+0**	**7.656e+0**	2.145e+3	1.099e+2
100,1000,1000	3.359e+2	3.359e+2	3.359e+2	6.926e+1	**6.925e+1**	3.692e+1	3.692e+1	3.692e+1	**3.674e+1**	**3.674e+1**

Table 4.4: Goodness-of-fit tests for the null hypothesis $\mathcal{H}_0$: vMF distribution and the alternate hypothesis $\mathcal{H}_a$: Fisher-Bingham distribution. Critical values of the test statistic correspond to a significance of 5%.

(d, κ)	Critical value	Test statistic					p-value of the test				
		Tanabe	Sra	Song	MML		Tanabe	Sra	Song	MML	
	$\chi^2(p)$	$\tilde{\kappa}^{\mathrm{T}}_{\mathrm{ML}}$	$\tilde{\kappa}^{\mathrm{N}}_{\mathrm{ML}}$	$\tilde{\kappa}^{\mathrm{H}}_{\mathrm{ML}}$	$\tilde{\kappa}^{\mathrm{N}}_{\mathrm{MML}}$	$\tilde{\kappa}^{\mathrm{H}}_{\mathrm{MML}}$	$\tilde{\kappa}^{\mathrm{T}}_{\mathrm{ML}}$	$\tilde{\kappa}^{\mathrm{N}}_{\mathrm{ML}}$	$\tilde{\kappa}^{\mathrm{H}}_{\mathrm{ML}}$	$\tilde{\kappa}^{\mathrm{N}}_{\mathrm{MML}}$	$\tilde{\kappa}^{\mathrm{H}}_{\mathrm{MML}}$
10,10	7.215e+1	1.850e+2	5.353e+1	5.353e+1	5.353e+1	5.353e+1	0.000e+0	5.258e-1	5.258e-1	5.260e-1	5.260e-1
10,100	7.215e+1	1.698e+3	4.949e+1	4.949e+1	4.945e+1	4.945e+1	0.000e+0	6.247e-1	6.247e-1	6.267e-1	6.267e-1
10,1000	7.215e+1	1.950e+3	4.811e+1	4.811e+1	5.060e+1	5.060e+1	0.000e+0	6.571e-1	6.571e-1	5.724e-1	5.724e-1
100,10	5.215e+3	5.090e+3	5.090e+3	5.090e+3	5.090e+3	5.090e+3	3.739e-1	3.739e-1	3.739e-1	3.741e-1	3.741e-1
100,100	5.215e+3	5.010e+3	5.010e+3	5.010e+3	5.010e+3	5.010e+3	6.103e-1	6.127e-1	6.127e-1	6.125e-1	6.125e-1
100,1000	5.215e+3	5.025e+3	5.018e+3	5.018e+3	5.022e+3	5.022e+3	5.427e-1	5.597e-1	5.597e-1	5.517e-1	5.517e-1
1000,10	5.021e+5	5.006e+5	5.006e+5	5.006e+5	5.006e+5	5.006e+5	4.682e-1	4.682e-1	4.682e-1	4.687e-1	4.687e-1
1000,100	5.021e+5	5.005e+5	5.005e+5	5.005e+5	5.005e+5	5.005e+5	5.050e-1	5.050e-1	5.050e-1	5.057e-1	5.057e-1
1000,1000	5.021e+5	5.007e+5	5.007e+5	5.007e+5	5.007e+5	5.007e+5	4.283e-1	4.283e-1	4.283e-1	4.196e-1	4.196e-1

$N \to \infty$. For $d = \{10, 100, 1000\}$, the critical values at 5% significance level are given in Table 4.4. If the computed test statistic exceeds the critical value, then the null hypothesis of a vMF distribution is rejected. A simulation study is conducted where samples of size $N = 1$ million are generated from a vMF distribution with known mean and $\kappa = \{10, 100, 1000\}$. For each inferred estimate $\widehat{\kappa}$, the test statistic is compared with the corresponding critical value.

For $d = 10$, the approximation $\tilde{\kappa}^{\mathrm{T}}_{\mathrm{ML}}$ has a significant effect as its test statistic exceeds the critical value, and consequently the p-value is close to zero. This implies that the null hypothesis of a vMF distribution is rejected by using the estimate $\tilde{\kappa}^{\mathrm{T}}_{\mathrm{ML}}$. However, this is incorrect as the data was generated from a vMF distribution. The p-values due to the estimates $\tilde{\kappa}^{\mathrm{N}}_{\mathrm{ML}}, \tilde{\kappa}^{\mathrm{H}}_{\mathrm{ML}}, \tilde{\kappa}^{\mathrm{N}}_{\mathrm{MML}}$, and $\tilde{\kappa}^{\mathrm{H}}_{\mathrm{MML}}$ are all greater than 0.05 (the significance level) which implies that the null hypothesis is accepted. For $d = \{100, 1000\}$, the p-values corresponding to the different estimates are greater than 0.05. In these cases, the use of all the estimates lead to the same conclusion, that is, accepting the null hypothesis of a vMF distribution. As the amount of data increases, the error due to all the estimates decreases. This is further exemplified below.

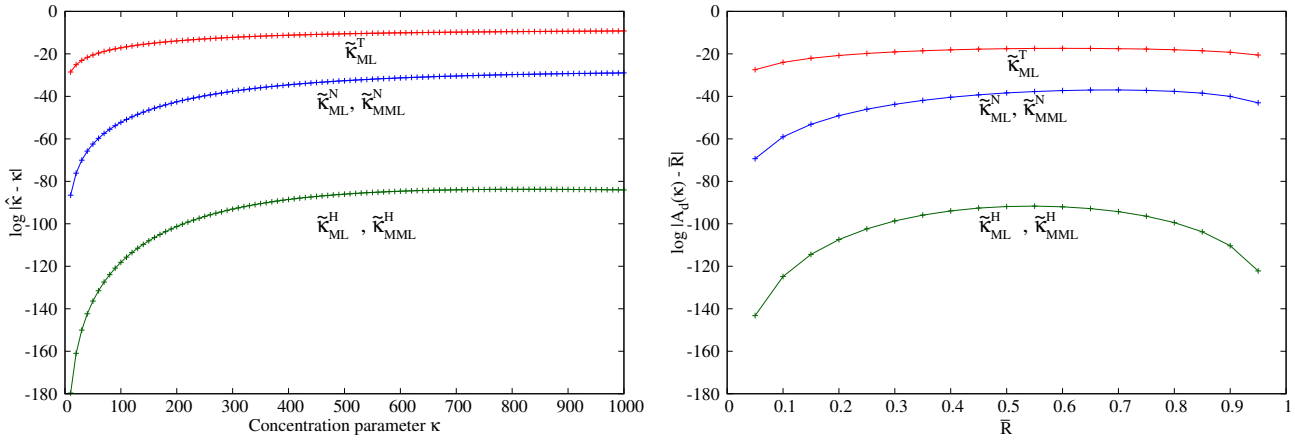
Asymptotic behaviour of the MML estimates: Based on the empirical tests, we have so far seen that MML estimates fare better when compared to the other approximations. The limiting case behaviour of the MML estimates is now discussed. For large sample sizes ($N \to \infty$), we plot the errors in κ estimation. Song et al. (2012) demonstrated that their approximation $\tilde{\kappa}^{\mathrm{H}}_{\mathrm{ML}}$ results in the lowest error

in the limiting case, as compared to Sra (2012). The variation in error is computed in two scenarios with fixed $d = 1000$ and

1. *increasing* κ: Figure 4.2(a) illustrates the behaviour of the absolute error with increasing κ. The first observation is that irrespective of the estimation procedure, the error continues to increase with increasing κ values (which corroborates our observations in the empirical tests) and then saturates. According to Song et al. (2012), their estimate $\tilde{\kappa}^{\mathrm{H}}_{\mathrm{ML}}$ produces the lowest error which we can see in the figure. Further, our MML-Newton approximation $\tilde{\kappa}^{\mathrm{N}}_{\mathrm{MML}}$ actually performs worse than Song's approximation $\tilde{\kappa}^{\mathrm{H}}_{\mathrm{ML}}$. However, we note that the errors due to MML Halley's approximation $\tilde{\kappa}^{\mathrm{H}}_{\mathrm{MML}}$ are identical to those produced by $\tilde{\kappa}^{\mathrm{H}}_{\mathrm{ML}}$. This suggests that asymptotically, the approximations achieved by $\tilde{\kappa}^{\mathrm{H}}_{\mathrm{ML}}$ and $\tilde{\kappa}^{\mathrm{H}}_{\mathrm{MML}}$ are more accurate (note that the errors in the limiting case are extremely low).

2. *increasing* $\bar{R}$: The maximum likelihood estimate of κ aims to achieve $F(\widehat{\kappa}) \equiv A_d(\widehat{\kappa}) - \bar{R} = 0$ (Equation 4.3). Hence, $\log|A_d(\kappa) - \bar{R}|$ gives a measure of the error corresponding to an estimate of κ. Figure 4.2(b) depicts the variation of this error with increasing $\bar{R}$. It is observed that $\tilde{\kappa}^{\mathrm{H}}_{\mathrm{ML}}$ and $\tilde{\kappa}^{\mathrm{H}}_{\mathrm{MML}}$ produce the least error. Also note that the error produced due to $\tilde{\kappa}^{\mathrm{N}}_{\mathrm{MML}}$ is greater than that produced by $\tilde{\kappa}^{\mathrm{H}}_{\mathrm{MML}}$. However, we highlight the fact that MML-based parameter inference aims to achieve $G(\widehat{\kappa}) \equiv 0$ (Equation 4.13), a fundamentally different objective function compared to the maximum likelihood based one.

The asymptotic results are shown here by assuming a value of $N = 10^{200}$ (note the corresponding extremely low error rates). In the limiting case, the MML estimate $\tilde{\kappa}^{\mathrm{H}}_{\mathrm{MML}}$ coincides with the ML estimate $\tilde{\kappa}^{\mathrm{H}}_{\mathrm{ML}}$. However, the performance of Halley's approximation $\tilde{\kappa}^{\mathrm{H}}_{\mathrm{ML}}$ is better compared to the MML-Newton's approximation $\tilde{\kappa}^{\mathrm{N}}_{\mathrm{MML}}$. The same behaviour is observed for when κ is fixed and the dimensionality is increased. For *enormous* amounts of data, the ML approximations converge to the MML ones.



(a) Variation in error with increasing κ

(b) Variation in error with increasing $\bar{R}$

Figure 4.2: Errors in κ estimation for $d = 1000$ as the sample size $N \to \infty$.

For higher dimensional vMF distributions, the above empirical analyses demonstrate that the approximations of the MML estimate of κ perform better in terms of the neutral evaluation metrics such as bias, MSE, and KL distance. The analyses was done under many combinations of sample sizes, dimensionality, and the true distribution parameters. In all the cases, as observed above, the MML estimates outperform the ML estimates.

4.3 Kent distribution on a 3D sphere

The vMF distribution discussed in the previous section is useful for modelling directional data that is symmetrically distributed with respect to a mean direction. The modelling of asymmetrically distributed directional data, however, requires distributions which generalize the vMF distribution. In this section, we focus on the specific case where the data is distributed on the surface of a sphere in the three-dimensional space. A generalization of the vMF in 3D, called the Fisher-Bingham distribution (Mardia, 1975b). has the form

$$f(\mathbf{x};\boldsymbol{\Theta}) \propto \exp\{\kappa\boldsymbol{\gamma}_1^T\mathbf{x} \ + \beta_2(\boldsymbol{\gamma}_2^T\mathbf{x})^2 + \beta_3(\boldsymbol{\gamma}_3^T\mathbf{x})^2\} \tag{4.16}$$

where the parameters $\boldsymbol{\gamma}_1, \boldsymbol{\gamma}_2, \boldsymbol{\gamma}_3$ are unit vectors with $\boldsymbol{\gamma}_2$ and $\boldsymbol{\gamma}_3$ being orthogonal to each other, the parameters β_2 and β_3 are real values with $\beta_2 \geq \beta_3$. As the distribution is characterized using an 8 real valued parameter vector $\boldsymbol{\Theta}$ (two for $\boldsymbol{\gamma}_1$, three for $\boldsymbol{\gamma}_2$ and $\boldsymbol{\gamma}_3$, and three scalars κ, β_2, β_3), it is also referred to as the FB_8 distribution. Notice that compared to the vMF, the FB_8 distribution has an exponential factor with additional quadratic terms.

The FB_8 distribution is a better choice compared to the vMF distribution when modelling real world 3D asymmetrical directional data, because the distribution has more free parameters. However, the use of the FB_8 distribution in directional statistics poses difficulties owing to its complex mathematical form and the lack of a natural understanding of its parameters (Kent, 1982). In order to achieve a balance between the highly simplified vMF model and the complex FB_8 distribution, Kent (1982) suggested an alternative form that is relatively easy to work with and whose parameters have natural interpretations. This distribution, referred to as the *Kent* distribution, is obtained from Equation 4.16 by assuming $\boldsymbol{\gamma}_1, \boldsymbol{\gamma}_2$, and $\boldsymbol{\gamma}_3$ form an orthogonal system of vectors and are subject to the constraint $\beta_2 = -\beta_3 = \beta$. The probability density function is then given by

$$f(\mathbf{x};\boldsymbol{\Theta}) = c(\kappa,\beta)^{-1} \exp\{\kappa\boldsymbol{\gamma}_1^T\mathbf{x} + \beta[(\boldsymbol{\gamma}_2^T\mathbf{x})^2 - (\boldsymbol{\gamma}_3^T\mathbf{x})^2]\} \tag{4.17}$$

where $\boldsymbol{\gamma}_1, \boldsymbol{\gamma}_2$, and $\boldsymbol{\gamma}_3$ form an orthogonal system and represent the *mean*, *major*, and *minor* axes, respectively; κ, as before, measures the concentration, and $0 \leq \beta < \kappa/2$ describes the *ovalness*.

The Kent distribution is a spherical analogue of the *general* Gaussian distribution and serves as a natural extension to the vMF distribution. The distribution has *ellipse-like contours* of constant probability density on the spherical surface. Kent (1982) argued that by imposing the constraint $\beta < \kappa/2$ and requiring $\boldsymbol{\gamma}_1, \boldsymbol{\gamma}_2$, and $\boldsymbol{\gamma}_3$ to be an orthogonal system, the distribution would be unimodal and have a behaviour similar to the Gaussian distribution but on a spherical surface.

As the Kent distribution is characterized using a five real valued parameter vector $\boldsymbol{\Theta}$ (three for $\boldsymbol{\gamma}_1, \boldsymbol{\gamma}_2$, and $\boldsymbol{\gamma}_3$ because they are orthogonal and unit vectors, and two for the scalar entities κ, β), it is popularly referred to as the FB_5 distribution. The 5-parameter Fisher-Bingham distribution is denoted as $FB_5(\mathbf{Q}, \kappa, \beta)$, where $\mathbf{Q} = (\boldsymbol{\gamma}_1, \boldsymbol{\gamma}_2, \boldsymbol{\gamma}_3)$ is a 3×3 orthogonal matrix. The normalization constant $c(\kappa, \beta)$ of the distribution is derived as an infinite series

$$c(\kappa,\beta) = 2\pi \sum_{j=0}^{\infty} \frac{\Gamma(j+\frac{1}{2})}{\Gamma(j+1)} \beta^{2j} \left(\frac{2}{\kappa}\right)^{2j+\frac{1}{2}} I_{2j+\frac{1}{2}}(\kappa) \tag{4.18}$$

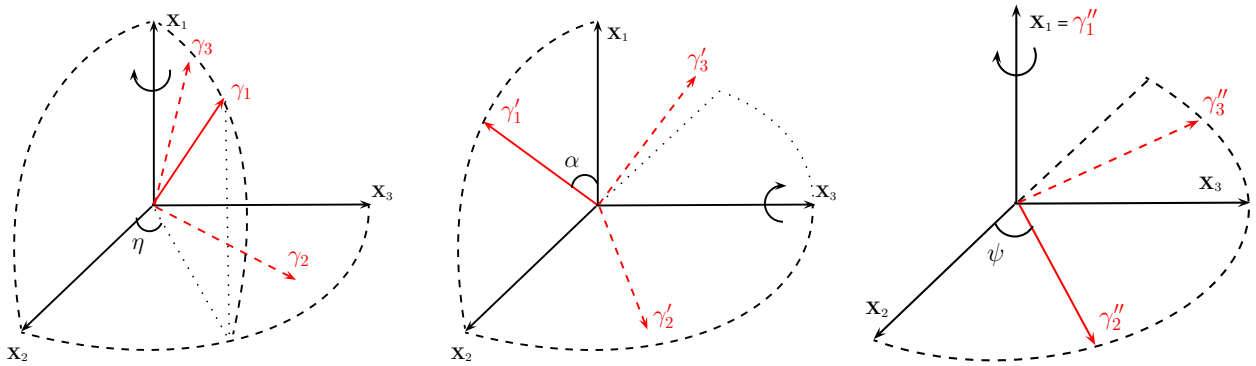
that depends on the Gamma function Γ and the modified Bessel function I_v of the first kind and order v (Abramowitz and Stegun, 1965; Kent, 1982).

The analysis of data using FB_5 distributions requires estimating the corresponding parameters. Due to the complex mathematical form of the density function, these estimates are approximated. Kent (1982) derived the *moment* estimates and suggested limiting case approximations. However, the use of such simplified approximations can have considerable effects from a practical standpoint. To overcome this, Bayesian estimation using the MML principle has been explored, as it has before

resulted in reliable estimators. In the case of the FB_5 distribution, this is demonstrated by the experiments in Section 4.3.6.

4.3.1 An intuitive parameterization of the distribution

The FB_5 distribution defined by Equation 4.17 comprises of three directional parameters and two scalar parameters. An intuitive way to parameterize the distribution that is relatively easy to comprehend is presented here. Let $\mathbf{X}_1 = (1\ 0\ 0)^T, \mathbf{X}_2 = (0\ 1\ 0)^T, \mathbf{X}_3 = (0\ 0\ 1)^T$ be the unit vectors along the standard coordinate axes. Let $\mathbf{R}$ be the rotation matrix that transforms the orientation axes $(\boldsymbol{\gamma}_1, \boldsymbol{\gamma}_2, \boldsymbol{\gamma}_3)$ supporting an FB_5 distribution to align with the standard coordinate axes. Then, $\mathbf{R}^T = (\boldsymbol{\gamma}_1, \boldsymbol{\gamma}_2, \boldsymbol{\gamma}_3) = \mathbf{Q}$ based on the following reasoning. Let $\alpha \in [0, \pi]$ and $\eta \in [0, 2\pi]$ be the co-latitude and longitude that determine the mean axis $\boldsymbol{\gamma}_1$ (shown in Figure 4.3a). A clockwise rotation by an angle η about $\mathbf{X}_1$ brings $\boldsymbol{\gamma}_1$ into the $\mathbf{X}_1\mathbf{X}_2$ plane. This operation transforms the axes to $\boldsymbol{\gamma}'_1, \boldsymbol{\gamma}'_2, \boldsymbol{\gamma}'_3$ respectively (Figure 4.3b). A subsequent clockwise rotation by an angle α about $\mathbf{X}_3$ aligns $\boldsymbol{\gamma}'_1$ with $\mathbf{X}_1$. This rotation brings the major and minor axes into the $\mathbf{X}_2\mathbf{X}_3$ plane (as orthogonality should be preserved). In this orientation (Figure 4.3c), let $\psi \in [0, \pi]$ be the angle between the transformed axis $\boldsymbol{\gamma}''_2$ and $\mathbf{X}_2$. A clockwise rotation by ψ about $\mathbf{X}_1$ aligns $\boldsymbol{\gamma}''_2$ with $\mathbf{X}_2$ and $\boldsymbol{\gamma}''_3$ with $\mathbf{X}_3$.



(a) R_η (b) R_α (c) R_ψ

Figure 4.3: The series of rotations to orient $\boldsymbol{\gamma}_1, \boldsymbol{\gamma}_2, \boldsymbol{\gamma}_3$ with the standard coordinate axes. The *red dashed* lines indicate the axis that is not in the first octant. For example, in (b), $\boldsymbol{\gamma}'_2$ is below the $\mathbf{X}_2\mathbf{X}_3$ plane whereas $\boldsymbol{\gamma}'_3$ is above the $\mathbf{X}_2\mathbf{X}_3$ plane but behind the $\mathbf{X}_1\mathbf{X}_3$ plane.

If $\mathbf{R}_\eta, \mathbf{R}_\alpha$, and $\mathbf{R}_\psi$ denote the respective rotation matrices given by

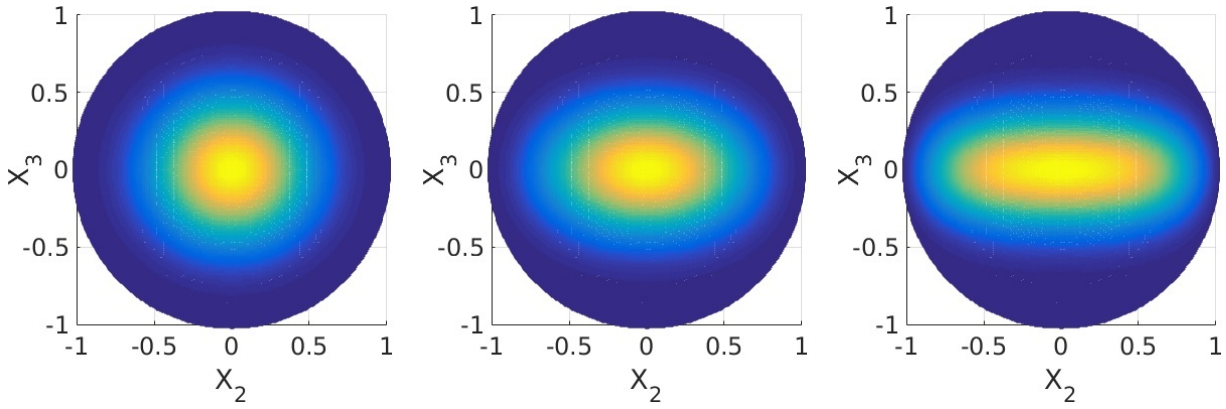
$$\mathbf{R}_\eta = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\eta & \sin\eta \\ 0 & -\sin\eta & \cos\eta \end{bmatrix}, \mathbf{R}_\alpha = \begin{bmatrix} \cos\alpha & \sin\alpha & 0 \\ -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}, \mathbf{R}_\psi = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\psi & \sin\psi \\ 0 & -\sin\psi & \cos\psi \end{bmatrix}$$

then the complete rotation matrix that effects the transformation from $(\boldsymbol{\gamma}_1, \boldsymbol{\gamma}_2, \boldsymbol{\gamma}_3)$ to $(\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3)$ is given by their product $\mathbf{R} = \mathbf{R}_\psi\mathbf{R}_\alpha\mathbf{R}_\eta$. By construction, any $\mathbf{X}_i = \mathbf{R}\boldsymbol{\gamma}_i, (i = 1, 2, 3)$ and, consequently, $\mathbf{Q} = \mathbf{R}^\mathsf{T}$. Hence, the three orthogonal axes $\boldsymbol{\gamma}_i$ of an FB_5 distribution can effectively be described using the three *angular* parameters ψ, α, and η as follows

$$\begin{aligned} \boldsymbol{\gamma}_1 &= (\cos\alpha,\ \sin\alpha\cos\eta,\ \sin\alpha\sin\eta)^T \\ \boldsymbol{\gamma}_2 &= (-\cos\psi\sin\alpha,\ \cos\psi\cos\alpha\cos\eta - \sin\psi\sin\eta,\ \cos\psi\cos\alpha\sin\eta + \sin\psi\cos\eta)^T \\ \boldsymbol{\gamma}_3 &= (\sin\psi\sin\alpha,\ -\sin\psi\cos\alpha\cos\eta - \cos\psi\sin\eta,\ -\sin\psi\cos\alpha\sin\eta + \cos\psi\cos\eta)^T \end{aligned} \tag{4.19}$$

The parameters κ and β are interpreted as scalars controlling the concentration and ovalness of the distribution. Also, since the distribution has ellipse-shaped contours on the spherical surface, it is easier to visualize the distribution and relate κ and β terms using eccentricity. Kent (1982) defined the

eccentricity[1] as $2\beta/\kappa$, which is constrained to be less than 1 (by definition), allowing correspondence between a specific Kent distribution and its elliptical nature. In order to better understand the interaction of κ and the eccentricity terms, we provide examples in Figure 4.4.



(a) eccentricity = 0.1 (b) eccentricity = 0.5 (c) eccentricity = 0.9

Figure 4.4: An example of an FB_5 distribution with varying eccentricities for $\kappa = 10$.

For a given $\kappa = 10$, an eccentricity of 0.1 results in (almost) spherical contours (Figure 4.4a) (reminiscent of a vMF distribution); an eccentricity of 0.5 results in contours which are moderately eccentric (Figure 4.4b); an eccentricity of 0.9 further disperses the data along the major axis (Figure 4.4c).

4.3.2 Existing methods of parameter estimation

The traditional methods of maximum likelihood (ML) estimation or maximum a priori (MAP) based estimation require the optimization of negative log-likelihood or the posterior density functions respectively. They, however, do not result in closed form solutions and present difficulties because of the complex form of the probability distribution. Hence, the widely used method of estimating the parameters of an FB_5 distribution is done using moment estimation. Kent (1982) formulated a procedure to obtain these estimates that may be subsequently used as starting points to obtain the ML or MAP estimates. Kent (1982) derived the moment estimates and suggested approximations based on these estimates.

Moment estimation

The moment estimates were proposed as an alternative to the maximum likelihood estimates. The approach adopted by Kent (1982) is described here: let data $\mathcal{D} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ be a random sample from $FB_5(\mathbf{Q}, \kappa, \beta)$. The *sample mean* $\bar{\mathbf{x}}$ and *sample dispersion* 3×3 *matrix* $\mathbf{S}$ of the data are then given as

$$\bar{\mathbf{x}} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_i \quad \text{and} \quad \mathbf{S} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_i\mathbf{x}_i^T$$

Let $\widetilde{\kappa}, \widetilde{\beta}$, and $\widetilde{\mathbf{Q}}$, where $\widetilde{\mathbf{Q}} = (\widetilde{\boldsymbol{\gamma}}_1, \widetilde{\boldsymbol{\gamma}}_2, \widetilde{\boldsymbol{\gamma}}_3)$ be the respective moment estimates of κ, β, and $\mathbf{Q}$. Then the moment estimate $\widetilde{\boldsymbol{\gamma}}_1$ of the unit mean vector is obtained by normalizing $\bar{\mathbf{x}}$, while the moment estimates $\widetilde{\boldsymbol{\gamma}}_2$ and $\widetilde{\boldsymbol{\gamma}}_3$ are obtained by diagonalizing $\mathbf{S}$. The matrix $\widetilde{\mathbf{Q}}$ is obtained using the following two steps

1. Choose an orthogonal matrix $\mathbf{H}$ to rotate $\bar{\mathbf{x}}$ to align with the $\mathbf{X}_1 = (1\ 0\ 0)^T$ axis (based on the discussion in Section 4.3.1, $\mathbf{H} = \mathbf{R}_\alpha\mathbf{R}_\eta$, where α and η are the co-latitude and longitude of $\bar{\mathbf{x}}$

[1]The definition of eccentricity in this context differs from the traditional definition of eccentricity for a conic section such as a parabola, an ellipse, or a hyperbola defined in the Euclidean plane.

respectively). Let $\mathbf{B} = \mathbf{H}^T\mathbf{S}\mathbf{H}$, so that $\mathbf{B}$ is the dispersion matrix in the transformed frame of reference.

2. The moment estimates of the major and minor axis correspond to the respective directions of maximum and minimum variance of the data in this transformed reference frame. If the angle between the direction of maximum variance and the $\mathbf{X}_2 = (0\ 1\ 0)^T$ axis is ψ, then a rotation defined by the orthogonal matrix $\mathbf{K}$ about $\mathbf{X}_1$ by ψ, aligns the maximum and minimum variance directions with the $\mathbf{X}_2$ and $\mathbf{X}_3 = (0\ 0\ 1)^T$ axes respectively. To compute these directions, it is required to diagonalize $\mathbf{B}_L$, the lower 2×2 submatrix of $\mathbf{B}$. The eigenvalue decomposition of $\mathbf{B}_L$ gives the angle ψ between the maximum variance direction and $\mathbf{X}_2$, which can be subsequently used to determine $\mathbf{K}$. If the 3×3 dispersion matrix $\mathbf{B} = [b_{ij}], 1 \leq i, j \leq 3$, the expression for ψ is

$$\tan 2\psi = \frac{2b_{23}}{b_{22} - b_{33}} \quad \text{where} \quad \mathbf{B}_L = \begin{bmatrix} b_{22} & b_{23} \\ b_{23} & b_{33} \end{bmatrix} \tag{4.20}$$

The two rotations defined by the orthogonal transformations $\mathbf{H}$ followed by $\mathbf{K}$ transform the axes of an FB_5 distribution to align with the standard coordinate axes. In effect, the original data $\mathcal{D}$ is transformed to $\mathcal{D}' = \{\mathbf{y}_1, \ldots, \mathbf{y}_N\}$ such that $\mathcal{D}'$ corresponds to a random sample drawn from $\text{FB}_5(\mathbf{I}, \kappa, \beta)$, where $\mathbf{I}$ is the identity matrix. Hence, an inverse transformation of the coordinate axes yields the moment estimates $\widetilde{\mathbf{Q}}$ of the axes of the FB_5 distribution. Further, for $\mathbf{y} = (y_1, y_2, y_3)^T$, Kent (1982) provided the first and second order moment expressions as follows

$$\mathbb{E}[y_1] = c_\kappa/c, \quad \mathbb{E}[y_2^2 - y_3^2] = c_\beta/c, \text{ where } c = c(\kappa, \beta), c_\kappa = \partial c/\partial\kappa, c_\beta = \partial c/\partial\beta \tag{4.21}$$

For data $\mathcal{D}$, if $\|\bar{\mathbf{x}}\|$ is the magnitude of the sample mean $\bar{\mathbf{x}}$ and $l_1 > l_2$ are the eigenvalues of $\mathbf{B}_L$, then Kent (1982) defines the *shape* and *size* and quantities as r_1 and r_2 respectively and are given as

$$r_1 = \mathbb{E}[y_1] = \|\bar{\mathbf{x}}\| \quad \text{and} \quad r_2 = \mathbb{E}[y_2^2 - y_3^2] = l_1 - l_2 \tag{4.22}$$

Solving these two equations in conjunction with Equation 4.21 results in the moment estimates $\widetilde{\kappa}$ and $\widetilde{\beta}$. As the expressions of the partial derivatives c_κ and c_β are difficult to work with, the following limiting case approximations of $\widetilde{\kappa}$ and $\widetilde{\beta}$ are often used (Kent, 1982).

$$\begin{aligned} \widetilde{\kappa} &\approx (2 - 2r_1 - r_2)^{-1} + (2 - 2r_1 + r_2)^{-1} \\ \widetilde{\beta} &\approx \frac{1}{2}\{(2 - 2r_1 - r_2)^{-1} - (2 - 2r_1 + r_2)^{-1}\} \end{aligned} \tag{4.23}$$

These asymptotic approximations can also be used as starting points to accurately determine $\widetilde{\kappa}$ and $\widetilde{\beta}$ using an optimization library.

Maximum likelihood estimation

To obtain the maximum likelihood estimates, the negative log-likelihood function $\mathcal{L}(\mathcal{D}|\mathbf{\Theta})$ of the data $\mathcal{D}$ needs to be minimized. It is to be noted that $\boldsymbol{\gamma}_1, \boldsymbol{\gamma}_2$, and $\boldsymbol{\gamma}_3$ are expressed in terms of ψ, α, and η (Equation 4.19), so that $\mathbf{\Theta} = \{\psi, \alpha, \eta, \kappa, \beta\}$ is a vector of parameters.

$$\mathcal{L}(\mathcal{D}|\mathbf{\Theta}) = N \log c(\kappa, \beta) - \kappa \boldsymbol{\gamma}_1^T \sum_{i=1}^{N} \mathbf{x}_i - \beta \boldsymbol{\gamma}_2^T \left(\sum_{i=1}^{N} \mathbf{x}_i \mathbf{x}_i^\mathsf{T} \right) \boldsymbol{\gamma}_2 + \beta \boldsymbol{\gamma}_3^T \left(\sum_{i=1}^{N} \mathbf{x}_i \mathbf{x}_i^\mathsf{T} \right) \boldsymbol{\gamma}_3 \tag{4.24}$$

The maximum likelihood estimates are given as solutions to the equation $\dfrac{\partial \mathcal{L}}{\partial \mathbf{\Theta}} = 0$. These estimates are obtained through numerical optimization as the solution cannot be written in an analytical form. The optimization routine often requires some initial values of the roots. These starting points are taken to be the moment estimates discussed previously.

4.3.3 Maximum *a posteriori* probability (MAP) estimation

The moment estimates of an FB_5 distribution are typically used in a variety of applications (Peel et al., 2001; Kent and Hamelryck, 2005; Boomsma et al., 2006; Hamelryck et al., 2006). In this section, we explore the MAP-based parameter estimation of the FB_5 distribution and use it when comparing the various estimators (see Section 4.3.6). The estimation procedure requires the maximization of the posterior density given some observed data $\mathcal{D}$.

As discussed in Section 2.2.2, if $\Pr(\boldsymbol{\Theta})$ is an appropriate prior density of the parameters and $\Pr(\mathcal{D}|\boldsymbol{\Theta})$ is the likelihood of data, then the posterior density $\Pr(\boldsymbol{\Theta}|\mathcal{D})$ is given as

$$\Pr(\boldsymbol{\Theta}|\mathcal{D}) \propto \Pr(\boldsymbol{\Theta}) \times \Pr(\mathcal{D}|\boldsymbol{\Theta})$$

The MAP estimator corresponds to the mode of the posterior distribution. The mode is, however, *not* invariant under some non-linear transformation of the parameter space (Murphy, 2012). This drawback is demonstrated in the context of estimating the parameters of a FB_5 distribution in the following discussion. A prior $\Pr(\boldsymbol{\Theta})$ is described on the parameter vector $\boldsymbol{\Theta}$ and it is formulated based on the choice of priors for the individual elements of the parameter vector.

Prior density of the parameters

The formulation of the prior density of the 5-parameter vector $\boldsymbol{\Theta}$ is derived as a product of the priors of the three angular parameters ψ, α, and η and two scalar parameters κ and β. Hence, the prior density of the complete set of parameters is given by $\Pr(\boldsymbol{\Theta}) = \Pr(\psi, \alpha, \eta, \kappa, \beta) = \Pr(\psi, \alpha, \eta) \times \Pr(\kappa, \beta)$.

Prior density on the angular parameters ψ, α, η: By construction (see Section 4.3.1), the pair α, η uniquely defines the mean direction $\boldsymbol{\gamma}_1$ of an FB_5 distribution. The mean may be considered to be uniformly distributed on the spherical surface and, hence, its prior density is $\dfrac{\sin\alpha}{4\pi}$. The angle ψ which determines the orientation of the major and minor axis in a plane perpendicular to $\boldsymbol{\gamma}_1$ is treated to be uniformly distributed in the range $[0, \pi]$. The joint prior of the angular parameters is, therefore, given by $\Pr(\psi, \alpha, \eta) = \dfrac{\sin\alpha}{4\pi^2}$.

Prior density on the scale parameters κ, β: The prior of the concentration parameter κ corresponds to the one used by Dowe et al. (1996c) in their analysis of vMF distributions defined on the 3D sphere and is given as $\Pr(\kappa) = \dfrac{4\kappa^2}{\pi(1+\kappa^2)^2}$. For a given κ, as per the definition of an FB_5 distribution, the parameter β will be in the range $[0, \kappa/2)$. A uniform prior is considered for β within this range, that is, the conditional density $\Pr(\beta|\kappa) = 2/\kappa$. Therefore, the joint prior density of the scalar parameters is $\Pr(\kappa, \beta) = (2/\kappa)\Pr(\kappa)$. The joint prior density $\Pr(\boldsymbol{\Theta})$ is, hence, given as:

$$\Pr(\psi, \alpha, \eta, \kappa, \beta) = \frac{2\kappa \sin\alpha}{\pi^3(1+\kappa^2)^2} \tag{4.25}$$

Non-linear transformations of the parameter space

With the help of an example, it is demonstrated here that the invariance property is not a characteristic of MAP-based estimation, thus making it a statistically less robust estimator. As discussed in Section 2.2.2, if $T(\boldsymbol{\Theta}) = \boldsymbol{\Theta}'$ denotes a transformation T on the parameter vector $\boldsymbol{\Theta}$ then, for invariance, the parameter estimates in both the parameterizations should be affected by the same transformation. The parameter estimate $\widehat{\boldsymbol{\Theta}}'$ in the transformed space and the estimate $\widehat{\boldsymbol{\Theta}}$ should be related as $T(\widehat{\boldsymbol{\Theta}}) = \widehat{\boldsymbol{\Theta}}'$.

An alternative parameterization involving β: Another parameterization is considered where the *eccentricity* $e = 2\beta/\kappa$ (see Section 4.3.1) is used instead of β. This is an example of a non-linear

transformation of the parameter β. The prior density $\Pr(\boldsymbol{\Theta}')$ of the modified parameter vector $\boldsymbol{\Theta}' = (\psi, \alpha, \eta, \kappa, e)$ is obtained by dividing the prior density $\Pr(\boldsymbol{\Theta})$ by the Jacobian of the transformation given by $J = \partial e/\partial\beta = 2/\kappa$. The prior density $\Pr(\boldsymbol{\Theta}')$ (after reparameterization) is

$$\Pr(\boldsymbol{\Theta}') = \Pr(\psi, \alpha, \eta, \kappa, e) = \frac{\Pr(\psi, \alpha, \eta, \kappa, \beta)}{J} = \frac{\kappa^2 \sin\alpha}{\pi^3(1+\kappa^2)^2} \tag{4.26}$$

Based on the definitions of prior densities in varying parameter spaces, one can estimate the parameters by maximizing the posterior density in the corresponding parameterization. The different expressions for the posterior density are given in Equation 2.3. The expression for $f(\mathbf{x}, \boldsymbol{\Theta}')$ is obtained by substituting $\beta = \kappa\, e/2$ in the FB_5 probability density function $f(\mathbf{x}, \boldsymbol{\Theta})$ given by Equation 4.17. It should be noted that the value of likelihood expression is the same across different parameterizations.

An example demonstrating the effects of alternative parameterizations

An example of estimating parameters using the various posterior distributions for a given dataset is shown here. A random sample of size $N = 10$ is generated from an FB_5 distribution (Kent et al., 2013). The true parameters of the distribution are $\{\psi, \alpha, \eta\} = \pi/2$ each, $\kappa = 10$, and $\beta = 2.5$ (eccentricity $= 0.5$). To obtain the MAP estimates, the objective functions corresponding to the posterior density (Equation 2.3) need to be maximized. To solve for the parameter estimates, the non-linear optimization library NLopt[2] (Johnson, 2014) in conjunction with derivative-free optimization (Powell, 1994) is used. Maximizing the two versions of the posterior density results in the following MAP estimates of ψ, α, and η

$$\widehat{\psi} = 2.071,\ \widehat{\alpha} = 1.493,\ \widehat{\eta} = 1.522 \quad \text{using } \Pr(\boldsymbol{\Theta})$$
$$\widehat{\psi} = 2.071,\ \widehat{\alpha} = 1.493,\ \widehat{\eta} = 1.522 \quad \text{using } \Pr(\boldsymbol{\Theta}')$$

It is observed that the MAP estimates of ψ, α, and η are not different from their counterparts obtained using the two variations of the posterior density. However, the estimates $\widehat{\kappa}$ and $\widehat{\beta}$ under the parameterizations $\boldsymbol{\Theta}$ and $\boldsymbol{\Theta}'$ do not correspond to each other, as illustrated in the results below.

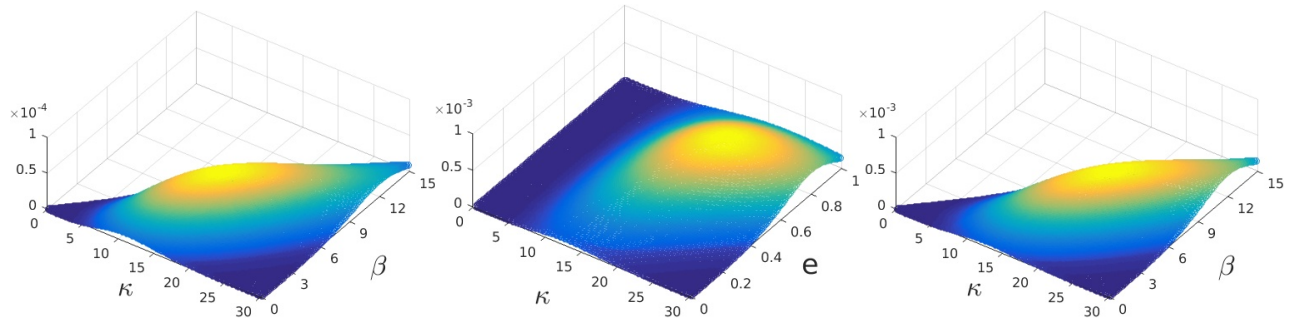
$$\widehat{\kappa} = 17.023,\ \widehat{\beta} = 5.493 \quad \text{using } \Pr(\boldsymbol{\Theta})$$
$$\widehat{\kappa} = 20.549,\ \widehat{e} = 0.701 \implies \widehat{\beta} = \widehat{\kappa}\,\widehat{e}/2 = 7.199 \quad \text{using } \Pr(\boldsymbol{\Theta}')$$

Ideally, the values of $\widehat{\kappa}$ and $\widehat{\beta}$ obtained through the use of $\Pr(\boldsymbol{\Theta}')$ should be the same as that obtained when the posterior density is maximized using $\Pr(\boldsymbol{\Theta})$ prior. Clearly, with MAP-based estimation, the end results are different for the two cases.

The modes of the posterior in the $\kappa,\ \beta$ and $\kappa,\ e$ parameterizations are shown in Figures 4.5(a) and (b), respectively. It is expected that the modes of the posterior shift as per the parameter space. However, they should be invariant regardless of the transformation affecting the two parameter spaces. It is observed that the mode in κ, e space, when mapped back to the κ, β space, results in a posterior density as shown in Figure 4.5(c). This is different from the posterior density shown in Figure 4.5(a), as the modes are at different locations. It is to be noted that the invariance property of parameter estimates is central to inductive inference. The example considered here corroborates that MAP estimation of the parameters of an FB_5 distribution does not satisfy the invariance property, thus resulting in unreliable estimators.

The aforementioned eccentricity transform is a straightforward transformation involving β. The remaining four parameters are left unchanged in this case. Another parameterization involving all five parameters of the FB_5 distribution is outlined in Appendix B.1.

[2] `http://ab-initio.mit.edu/nlopt`



(a) $\Pr(\mathbf{\Theta})$ (b) $\Pr(\mathbf{\Theta}')$ (c) $\Pr(\mathbf{\Theta}')$

Figure 4.5: Heat maps depicting the modes (MAP estimate) of the posterior density as a function of κ, β and κ, e parameterizations. The Z-axis denotes the posterior density value in the respective parameterization.

4.3.4 MML-based parameter estimation

In this section, the derivation of the MML-based parameter estimates of an FB_5 distribution is described. As explained in Section 2.4.2, the derivation of the MML estimates requires the formulation of the message length expression (Equation 2.8) for encoding some observed data using the FB_5 distribution. The formulation requires the use of a suitable prior density on the parameters (see Section 4.3.3). The prior for κ is taken as $h(\kappa)$, the prior of κ for the 3D vMF distribution on the two-sphere (Dowe et al., 1996c). This results in the joint prior density $\Pr(\psi, \alpha, \eta, \kappa, \beta)$ (Equation 4.25). The main bottleneck involved in the MML-based parameter estimation is, however, the evaluation of the Fisher information matrix. As shown later, its computation involves the first and second order moments corresponding to an FB_5 distribution.

Notations: Before describing the MML-based approach, the following notations are defined to be used subsequently. The partial derivatives of the normalization constant $c(\kappa, \beta)$ of the FB_5 distribution would be required later on. The following are the notations adopted to represent them.

$$c(\kappa, \beta) = c, \quad c_\kappa = \partial c/\partial \kappa, \quad c_\beta = \partial c/\partial \beta$$
$$c_{\kappa\kappa} = \partial^2 c/\partial \kappa^2, \quad c_{\beta\beta} = \partial^2 c/\partial \beta^2, \quad c_{\kappa\beta} = \partial^2 c/\partial \kappa \partial \beta$$

Derivation of the moments of a general FB_5 distribution

Kent (1982) provided the moment expressions in the case of an FB_5 distribution whose mean, major and minor axes are aligned with the standard coordinate axes. In this setup, consider a random vector $\mathbf{y} \sim \text{FB}_5(\mathbf{I}, \kappa, \beta)$, where $\mathbf{I}$ is the identity matrix. Then, Kent (1982) provided the following moments:

$$\mathbb{E}[\mathbf{y}] = \begin{pmatrix} c_\kappa/c & 0 & 0 \end{pmatrix}^T, \text{ and}$$
$$\mathbb{E}[\mathbf{y}\mathbf{y}^T] = \mathbf{\Lambda} = \begin{pmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{pmatrix}, \text{where}$$
$$\lambda_1 = \frac{c_\kappa}{c}, \lambda_2 = \frac{c - c_{\kappa\kappa} + c_\beta}{2c}, \lambda_3 = \frac{c - c_{\kappa\kappa} - c_\beta}{2c} \tag{4.27}$$

Based on these, the moments in the case of a general FB_5 distribution, that is, whose three mutually orthogonal axes can be oriented in any fashion, can be derived. Let $\mathbf{x} \sim \text{FB}_5(\mathbf{Q}, \kappa, \beta)$ be a generic distribution whose axes are not aligned with the coordinate axes. Recall, from Section 4.3.1, that $\mathbf{Q}$ is the rotation matrix that aligns the standard coordinate axes with the axes of an FB_5 distribution.

Based on the parameterization of the FB_5 distribution, it can be deduced that $\forall \mathbf{x}, \exists\, \mathbf{y} \sim \text{FB}_5(\mathbf{I}, \kappa, \beta)$ such that $\mathbf{x} = \mathbf{Q}\mathbf{y}$, and hence, $\mathbf{x}\mathbf{x}^\mathsf{T} = \mathbf{Q}\mathbf{y}\mathbf{y}^T\mathbf{Q}^T$. Using the results from Equation 4.27, we have

$$\begin{aligned} \mathbb{E}[\mathbf{x}] = \mathbf{Q}\,\mathbb{E}[\mathbf{y}] = \begin{pmatrix}\boldsymbol{\gamma}_1 & \boldsymbol{\gamma}_2 & \boldsymbol{\gamma}_3\end{pmatrix} \begin{pmatrix} c_\kappa/c & 0 & 0\end{pmatrix}^T = c_\kappa/c\,\boldsymbol{\gamma}_1 \\ \text{and } \mathbb{E}[\mathbf{x}\mathbf{x}^\mathsf{T}] = \mathbf{Q}\,\mathbb{E}[\mathbf{y}\mathbf{y}^T]\,\mathbf{Q}^T = \mathbf{Q}\boldsymbol{\Lambda}\mathbf{Q}^T \end{aligned} \tag{4.28}$$

Computation of the Fisher information

As described in Section 2.4.2, the computation of the *determinant* of the Fisher information matrix requires the evaluation of the second order partial derivatives of the negative log-likelihood function with respect to the parameters of the distribution. As per the density function (Equation 4.17), the negative log-likelihood of a datum $\mathbf{x}$ is given by

$$\mathcal{L}(\mathbf{x}|\boldsymbol{\Theta}) = \log c(\kappa, \beta) - \kappa\boldsymbol{\gamma}_1^T\mathbf{x} - \beta\,\boldsymbol{\gamma}_2^T\mathbf{x}\mathbf{x}^\mathsf{T}\boldsymbol{\gamma}_2 + \beta\,\boldsymbol{\gamma}_3^T\mathbf{x}\mathbf{x}^\mathsf{T}\boldsymbol{\gamma}_3 \tag{4.29}$$

where $\boldsymbol{\Theta} = \{\psi, \alpha, \eta, \kappa, \beta\}$. Let $\mathcal{F}_1(\boldsymbol{\Theta})$ denote the Fisher information for a *single* observation. the Fisher information matrix $\mathcal{F}_1(\boldsymbol{\Theta})$ in the case of an FB_5 distribution is a 5×5 *symmetric* matrix. Further, as explained later, the determinant $|\mathcal{F}_1(\boldsymbol{\Theta})|$ is decomposed as a product of $|\mathcal{F}_A|$ and $|\mathcal{F}_S|$, where $\mathcal{F}_A$ is the Fisher matrix associated with the angular parameters ψ, α, and η, and $\mathcal{F}_S$ is the Fisher matrix associated with the scale parameters κ and β.

Fisher matrix ($\mathcal{F}_A$) associated with ψ, α, η: $\mathcal{F}_A$ is a 3×3 symmetric matrix whose elements are the expected values of the second order partial derivatives of $\mathcal{L}$ with respect to $\theta_i, \theta_j \in \{\psi, \alpha, \eta\}$. Let the *expectation* be given as

$$\mathbb{E}\left[\frac{\partial^2\mathcal{L}}{\partial\theta_i\partial\theta_j}\right] = -\kappa\,T(\boldsymbol{\gamma}_1) - \beta\,T(\boldsymbol{\gamma}_2) + \beta\,T(\boldsymbol{\gamma}_3) \tag{4.30}$$

where the individual terms $T(\boldsymbol{\gamma}_m)$, $m \in \{1, 2, 3\}$ are comprised of the expectations of the corresponding partial differentials of $\boldsymbol{\gamma}_m$. They are computed using the following identities:

$$\begin{aligned} T(\boldsymbol{\gamma}_1) &= \mathbb{E}\left[\frac{\partial^2(\boldsymbol{\gamma}_1^T\mathbf{x})}{\partial\theta_i\partial\theta_j}\right] = \mathbb{E}\left[\mathbf{x}\right]^T \frac{\partial^2\boldsymbol{\gamma}_1}{\partial\theta_i\partial\theta_j}, \text{ and} \\ T(\boldsymbol{\gamma}_m) &= \mathbb{E}\left[\frac{\partial^2(\boldsymbol{\gamma}_m^T\mathbf{x}\mathbf{x}^\mathsf{T}\boldsymbol{\gamma}_m)}{\partial\theta_i\partial\theta_j}\right] \quad (\text{for } m = 2, 3) \\ &= 2\left(\boldsymbol{\gamma}_m^T\mathbb{E}\left[\mathbf{x}\mathbf{x}^\mathsf{T}\right]\frac{\partial^2\boldsymbol{\gamma}_m}{\partial\theta_i\partial\theta_j} + \left(\frac{\partial\boldsymbol{\gamma}_m}{\partial\theta_i}\right)^T\mathbb{E}\left[\mathbf{x}\mathbf{x}^\mathsf{T}\right]\left(\frac{\partial\boldsymbol{\gamma}_m}{\partial\theta_j}\right)\right) \end{aligned} \tag{4.31}$$

The terms $T(\boldsymbol{\gamma}_m)$ depend on the expressions for the constituent first and second order partial differentials of $\boldsymbol{\gamma}_m$, which are provided in Appendix B.2. Using Equations 4.28, 4.30 and 4.31, the elements of the Fisher information matrix $\mathcal{F}_A$ are derived as follows

$$\begin{aligned} \mathcal{F}_{\psi\psi} &= 4\,\beta\,c_\beta/c\,; \quad \mathcal{F}_{\alpha\psi} = 0; \quad \mathcal{F}_{\eta\psi} = (\cos\alpha)\,4\,\beta\,c_\beta/c \\ \mathcal{F}_{\alpha\alpha} &= \kappa\,c_\kappa/c + 2\beta\left\{(\lambda_1 - \lambda_3)\sin^2\psi - (\lambda_1 - \lambda_2)\cos^2\psi\right\} \\ \mathcal{F}_{\eta\alpha} &= \beta(1 - 3\lambda_1)\sin 2\psi\,\sin\alpha \\ \mathcal{F}_{\eta\eta} &= (\sin^2\alpha)\,\kappa\,c_\kappa/c \\ &\quad +2\beta\left\{\begin{array}{l} \lambda_2(\cos^2\psi\,\cos^2\alpha + \sin^2\psi) + (\lambda_2 - \lambda_3)\cos^2\alpha \\ -\lambda_3(\sin^2\psi\,\cos^2\alpha + \cos^2\psi) + \lambda_1\sin^2\alpha\cos 2\psi \end{array}\right\} \end{aligned} \tag{4.32}$$

Fisher matrix ($\mathcal{F}_S$) associated with κ, β: $\mathcal{F}_S$ is a 2×2 symmetric matrix whose elements are the *expectations* of the second order partial derivatives of $\mathcal{L}$ with respect to κ and β. Differentiating

Equation 4.29 with respect to κ and β, we get

$$\frac{\partial \mathcal{L}}{\partial \kappa} = \frac{c_\kappa}{c} - \boldsymbol{\gamma}_1^T \mathbf{x} \quad \text{and} \quad \frac{\partial \mathcal{L}}{\partial \beta} = \frac{c_\beta}{c} - \boldsymbol{\gamma}_2^T \mathbf{x}\mathbf{x}^\mathsf{T}\boldsymbol{\gamma}_2 + \boldsymbol{\gamma}_3^T \mathbf{x}\mathbf{x}^\mathsf{T}\boldsymbol{\gamma}_3$$

$$\frac{\partial^2 \mathcal{L}}{\partial \kappa^2} = \frac{c c_{\kappa\kappa} - c_\kappa^2}{c^2} = \mathcal{F}_{\kappa\kappa}$$

$$\frac{\partial^2 \mathcal{L}}{\partial \beta^2} = \frac{c c_{\beta\beta} - c_\beta^2}{c^2} = \mathcal{F}_{\beta\beta}, \quad \text{and}$$

$$\frac{\partial^2 \mathcal{L}}{\partial \kappa \partial \beta} = \frac{c c_{\kappa\beta} - c_\kappa c_\beta}{c^2} = \mathcal{F}_{\kappa\beta} \tag{4.33}$$

Fisher matrix $\mathcal{F}(\boldsymbol{\Theta})$ associated with the 5-parameter vector $\boldsymbol{\Theta}$: It is to be noted that for $\theta_i \in \{\kappa, \beta\}$ and $\theta_j \in \{\psi, \alpha, \eta\}$, $T(\boldsymbol{\gamma}_m) = 0$ as $\dfrac{\partial \boldsymbol{\gamma}_m}{\partial \theta_i} = 0$ ($\boldsymbol{\gamma}_m$ given by Equation 4.19 are independent of κ, β). Consequently, $\mathcal{F}_{\theta_i\theta_j} = 0$. This allows for the computation of $|\mathcal{F}_1(\boldsymbol{\Theta})|$ as the product of $|\mathcal{F}_A|$ and $|\mathcal{F}_S|$, that is,

$$|\mathcal{F}_1(\boldsymbol{\Theta})| = \begin{vmatrix} \mathcal{F}_{\psi\psi} & \mathcal{F}_{\psi\alpha} & \mathcal{F}_{\psi\eta} & 0 & 0 \\ \mathcal{F}_{\alpha\psi} & \mathcal{F}_{\alpha\alpha} & \mathcal{F}_{\alpha\eta} & 0 & 0 \\ \mathcal{F}_{\eta\psi} & \mathcal{F}_{\eta\alpha} & \mathcal{F}_{\eta\eta} & 0 & 0 \\ 0 & 0 & 0 & \mathcal{F}_{\kappa\kappa} & \mathcal{F}_{\kappa\beta} \\ 0 & 0 & 0 & \mathcal{F}_{\beta\kappa} & \mathcal{F}_{\beta\beta} \end{vmatrix} = |\mathcal{F}_A||\mathcal{F}_S|$$

Then, the Fisher information for some observed data $\mathcal{D} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ is given by

$$|\mathcal{F}(\boldsymbol{\Theta})| = N^5 |\mathcal{F}_1(\boldsymbol{\Theta})| \tag{4.34}$$

as each element in $|\mathcal{F}_1(\boldsymbol{\Theta})|$ is multiplied by the sample size N.

Message length formulation

The message length to encode some observed data $\mathcal{D}$ can now be formulated by substituting the prior density $\Pr(\boldsymbol{\Theta})$ (Equation 4.25), the Fisher information $|\mathcal{F}(\boldsymbol{\Theta})|$ and the negative log-likelihood of the data (Equation 4.24) in the message length expression (Equation 2.8). The MML parameter estimates are the ones that minimize the total message length. As there is no analytical form of the MML estimates, the solution is obtained, as for the maximum likelihood and MAP cases, by using the NLopt optimization library (Johnson, 2014). At each stage of the optimization routine, the Fisher information needs to be calculated. However, this involves the computation of complex entities such as the normalization constant $c(\kappa, \beta)$ and its partial derivatives. The computation of these intricate mathematical forms using numerical methods is discussed in Section 4.3.5.

4.3.5 Computation of the normalization constant and its derivatives

The computation of the negative log-likelihood function and the message length is hindered because of the presence of the normalization constant and its associated derivatives. Kent (1982) provided an asymptotic formula for $c(\kappa, \beta)$ as $2\pi \exp(\kappa)[(\kappa^2 - 4\beta^2)]^{-1/2}$. However, this approximation is valid for large κ and when $2\beta/\kappa$ is sufficiently small. This section describes new methods that can be employed to efficiently compute these complex functions.

Computing $\log c(\kappa, \beta)$ and the logarithm of the partial derivatives: c_κ and $c_{\kappa\kappa}$

The expressions of $c, c_\kappa, c_{\kappa\kappa}$ are related to each other. These are explained by defining the quantity $S_1^{(m)}$, a logarithm sum,

$$S_1^{(m)} = \log \delta_1 + \log \sum_{j=0}^{\infty} \underbrace{\frac{\Gamma(j+\frac{1}{2})}{\Gamma(j+1)} e^{2j} I_{p+m}(\kappa)}_{f_j} \tag{4.35}$$

where $m \in \{0, 1, 2\}$, $p = 2j + \frac{1}{2}$, $\delta_1 = 2\pi\sqrt{\frac{2}{\kappa}}$, and $e = \frac{2\beta}{\kappa} < 1$ (by definition).

Computation of the series $S_1^{(m)}$: We first establish that $f_{j+1} < f_j, \forall j \geq 0$ and show that $S_1^{(m)}$ converges to a finite sum as $j \to \infty$. Consider the logarithm of the ratio of consecutive terms f_j and f_{j+1} in $S_1^{(m)}$, that is,

$$\log \frac{f_{j+1}}{f_j} = \log \frac{j+\frac{1}{2}}{j+1} + 2\log e + \log \frac{I_{p+m+2}(\kappa)}{I_{p+m}(\kappa)} \tag{4.36}$$

For $p, v > 0$, $I_{p+v} < I_p$, and the ratio $\frac{I_{p+v}}{I_p} \to 0$ for large v (Amos, 1974). Further, $e < 1$ implies the above equation is the sum of negative terms. Hence, $\log \frac{f_{j+1}}{f_j} < 0$, which means $f_{j+1} < f_j$. Also,

$$\lim_{j\to\infty} \log \frac{f_{j+1}}{f_j} = 0 + 2\log e + \lim_{j\to\infty} \log \frac{I_{2j+\frac{1}{2}+2}(\kappa)}{I_{2j+\frac{1}{2}}(\kappa)} = -\infty$$

Hence, as $\lim_{j\to\infty} \frac{f_{j+1}}{f_j} = 0$, $S_1^{(m)}$ is a convergent series.

For a practical implementation of the sum, we express $S_1^{(m)}$ as the modified summation

$$S_1^{(m)} = \log \delta_1 + \log f_0 + \log \sum_{j=0}^{\infty} t_j \tag{4.37}$$

where each f_j is divided by the *maximum* term f_0. For each $j > 0, \log f_j$ is calculated using the previous term $\log f_{j-1}$ (Equation 4.36). The new term $t_j = f_j/f_0$ is then computed[3] as $\exp(\log f_j - \log f_0)$. This is because computing the difference with the maximum value and then taking the exponent ensures numerical stability. The summation is terminated when the ratio $\frac{t_j}{\sum_{k=0}^{j} t_k} < \epsilon$ (a small threshold $\sim 10^{-6}$).

- Let $S(c) = \log c(\kappa, \beta)$: Substituting $m = 0$ in Equation 4.35 gives the logarithm of the normalization constant (given in Equation 4.18). Hence, $S(c) = S_1^{(0)}$.

- Let the j^{th} term dependent on κ in Equation 4.18 be represented as $g_j(\kappa) = I_p/\kappa^p$, where I_p implicitly refers to $I_p(\kappa)$. Based on the relationship between the Bessel functions I_p, I_{p+1}, and the derivative I_p' in Equation 4.38 (Abramowitz and Stegun, 1965), the expressions for the first and second derivatives of $g_j(\kappa)$ (Equation 4.39) are derived as

$$\kappa I_p' = p I_p + \kappa I_{p+1} \tag{4.38}$$

$$g_j'(\kappa) = \frac{I_{p+1}}{\kappa^p} \quad \text{and} \quad g_j''(\kappa) = \frac{I_{p+2}}{\kappa^p} + \frac{1}{\kappa}.\frac{I_{p+1}}{\kappa^p} \tag{4.39}$$

[3]Because of the nature of Bessel functions, $\log f_j$ can get very large and can result in overflow when calculating the exponent $\exp(\log f_j)$. However, dividing by f_0 results in $f_j/f_0 < 1$.

- Let $S(c_\kappa) = \log c_\kappa$: Because of the similar forms of $g_j(\kappa)$ and $g'_j(\kappa)$, the expression for $S(c_\kappa)$ will be similar to that of $S(c)$ with a change in *order* of the Bessel functions from $m = 0$ in Equation 4.35 to $m = 1$. Hence, $S(c_\kappa) = S_1^{(1)}$ and an expression similar to Equation 4.37 can be derived for $S(c_\kappa)$.

- Let $S(c_{\kappa\kappa}) = \log c_{\kappa\kappa}$: Substituting $m = 2$ in Equation 4.35 gives the logarithm sum $S_1^{(2)}$ corresponding to the series with terms $\frac{I_{p+2}}{\kappa^p}$. Based on the nature of $g''_j(\kappa)$ (Equation 4.39), and noting that $S(c_\kappa) > S_1^{(2)}$ (as $I_{p+1} > I_{p+2}\,\forall\, p \geq 0$), $S(c_{\kappa\kappa})$ is formulated as

$$S(c_{\kappa\kappa}) = S(c_\kappa) + \log\left(\exp(S_1^{(2)} - S(c_\kappa)) + \frac{1}{\kappa}\right)$$

The logarithm of the partial derivatives: c_β, $c_{\kappa\beta}$, and $c_{\beta\beta}$

The expressions of c_β and $c_{\kappa\beta}$ are related and are explained using the logarithm sum $S_2^{(n)}$

$$S_2^{(n)} = \log \delta_2 + \log \sum_{j=1}^{\infty} \underbrace{\frac{\Gamma(j+\frac{1}{2})}{\Gamma(j)} e^{2j} I_{p+n}(\kappa)}_{f_j} \tag{4.40}$$

where $n \in \{0, 1\}$, $\delta_2 = \frac{4\pi}{\beta}\sqrt{\frac{2}{\kappa}}$, $p = 2j + \frac{1}{2}$, and $e = \frac{2\beta}{\kappa}$. Note that $S_2^{(n)}$ is a convergent series (the proof is based on the same reasoning as for $S_1^{(m)}$).

Let the j^{th} term dependent on β, κ in Equation 4.18 be represented as $g_j(\beta, \kappa) = \beta^{2j}\frac{I_p}{\kappa^p}$. Its partial derivatives are given below. These derivatives are the terms in the series $S_2^{(n)}$ (after factoring out the common elements as δ_2).

$$\frac{\partial g_j}{\partial \beta} = 2j\beta^{2j-1}\frac{I_p}{\kappa^p} \quad \text{and} \quad \frac{\partial^2 g_j}{\partial \kappa \partial \beta} = 2j\beta^{2j-1}\frac{I_{p+1}}{\kappa^p}$$

- Let $S(c_\beta) = \log c_\beta$: this is obtained by substituting $n = 0$ in Equation 4.40. Hence, $S(c_\beta) = S_2^{(0)}$.

- Similarly, $S(c_{\kappa\beta}) = \log c_{\kappa\beta} = S_2^{(1)}$.

- The expression to compute $S(c_{\beta\beta}) = \log c_{\beta\beta}$ is given by

$$S(c_{\beta\beta}) = \log\left(\frac{\delta_2}{\beta}\right) + \log \sum_{j=1}^{\infty} \underbrace{\frac{\Gamma(j+\frac{1}{2})}{\Gamma(j)}(2j-1)e^{2j} I_p(\kappa)}_{f_j}$$

The practical implementation of $S_2^{(n)}$ and $S(c_{\beta\beta})$ is similar to that of $S_1^{(m)}$ given by Equation 4.37. However, in these cases, the expressions of f_j and consequently t_j, are modified depending on their individual forms. Also, the series begin from $j = 1$, and hence, the maximum terms will correspond to f_1.

4.3.6 Evaluation of the MML estimates

For a given FB_5 distribution characterized by concentration κ and eccentricity e, a random sample of size N is generated using the method proposed by Kent et al. (2013). The angular parameters of the true distribution are set to $\{\psi, \alpha, \eta\} = \pi/2$. The scale parameters κ and e are varied to obtain

different FB_5 distributions and corresponding random samples. The parameters are estimated using the sampled data and the different estimation methods. The procedure is repeated 1000 times for each combination of N, κ, and e.

Methods of comparison

The moment, ML, MAP, and MML estimates of the data generated in these simulations are compared with each other. The results include the two versions of MAP estimates resulting from the two forms of the posterior distributions (Equations 4.25 and 4.26): *MAP1* corresponds to the posterior with parameterization κ, β, and *MAP2* corresponds to the posterior with parameterization κ, e. The MML estimates are obtained by minimizing the message length expression. Naturally, the estimates due to other methods do not result in lower message lengths. Similarly, if the negative log-likelihood is used as the comparison criterion, the maximum likelihood estimates have a lower value compared to the others. As each estimation technique optimizes a different objective function, we need to use a metric that impartially evaluates the different estimates.

Comparison using mean squared error (MSE) and Kullback-Leibler (KL) distance: As discussed in Section 2.5, the mean squared error (MSE) and Kullback-Leibler (KL) distance (Kullback and Leibler, 1951) are, therefore, used to compare the various estimates. For a parameter vector $\mathbf{\Theta}$, and its estimate $\widehat{\mathbf{\Theta}}$, the MSE is given by Equation 2.13. It is desirable for an estimator to have a lower MSE.

With respect to KL distance, an estimate *wins* if it results in a lower value of the KL distance, and it is considered a better estimate. The analytical form of the KL distance between two FB_5 distributions is derived in Appendix B.3. The percentage of times (out of 1000 random simulations) that the KL distance of a particular estimator is lower than that of others is reported. When the KL distance of different estimates is compared, because of two different versions of MAP estimation, two separate frequency plots are presented. The KL distance of the moment, ML, and MML estimates is contrasted with the KL distance of the MAP1 or MAP2 estimates.

Comparison using statistical hypothesis testing: In order to compare the various estimators, we consider the likelihood ratio test statistic Λ (see Section 2.3.1), which is is asymptotically approximated as an χ^2 distribution with five degrees of freedom. In the current analysis of the various estimators, the likelihood ratio resulting from the use of a particular estimate $\widehat{\mathbf{\Theta}}$ (that is, moment, MAP or MML) is compared against that of a general FB_5 distribution. This is equivalent to testing the null hypothesis $\mathcal{H}_0 : \mathbf{\Theta} = \widehat{\mathbf{\Theta}}$ (explicit parameters) against the alternate hypothesis $\mathcal{H}_A : \mathbf{\Theta} \neq \widehat{\mathbf{\Theta}}$ (with five free parameters). Assuming a statistical significance of the test as 1%, $\mathcal{H}_0$ is rejected when $\Lambda > \tau$, where $\tau = 13.086$ corresponds to the 99^{th} percentile of an χ^2 distribution with 5 degrees of freedom. Alternatively, the test statistic can be used to evaluate the p-value, which if less than 1% (significance of the test) amounts to rejection of $\mathcal{H}_0$.

For the various parameter estimates compared here, it is expected that at especially large sample sizes, the estimates are close to the maximum likelihood estimate as determined by the corresponding test statistic. In other words, the empirically determined test statistic is expected to be lower than the critical value τ, which implies it has a corresponding p-value greater than 0.01.

Empirical analyses

The estimates are analyzed here in two controlled cases: (1) fixing sample size N while varying κ and e, and (2) varying N while fixing κ and e.

Fixed sample size, varying concentration κ and eccentricity e: The results are presented when a random sample of size $N = 10$ is generated from the FB_5 distribution for a κ that is increased by an order of magnitude starting from 1 to 100. The behaviour of the estimates is analyzed below.

For $\kappa = 1$: The performance of the various estimates using the above described comparison methodologies is shown in Figure 4.6. It is observed that the bias and MSE of moment and ML estimates is greater than that of the MAP and MML estimates. The two versions of the MAP estimates also have a greater bias and MSE as compared to the MML estimates shown in Figure 4.6(a) and (b).

It is also observed that the MML estimates result in a lower KL distance, more than 80% of time as compared to other estimates when MAP1 is used (see Figure 4.6c). With MAP2, the frequency of wins for the MML estimates increases to more than 90% (see Figure 4.6d). This suggests that transforming the parameter space greatly impacts the MAP estimates. The ML estimates win less than 5% of the time. This is in agreement with the relatively greater MSE observed for the ML estimates.

The boxplots shown in Figures 4.6(e) and (f) show the variation of the test statistics and the corresponding p-values. There is a greater variation for the MML estimates. However, across all values of eccentricity, the test statistic Λ is less than the threshold $\tau = 13.086$ and the smallest p-value is greater than 0.01. This is true across all estimation methods, thus, suggesting that the null hypothesis of modelling data using a particular estimate (moment, MAP or MML) is accepted at the 1% significance level.

For $\kappa = 10$: The comparison results are presented in Figure 4.7. Similar to the previous case ($\kappa = 1$), the moment and ML estimates have greater bias and MSE. It is interesting to note that MAP2 has greater bias and MSE compared to the MML estimates (see Figures 4.7(a) and (b) respectively). However, MAP1 estimates are in close competition with those of MML. The bias and MSE are lower for MML estimates until $e \leq 0.5$ and greater compared to MAP1 estimates for $e > 0.5$.

The number of times the KL distance is lower for the MML estimates decreases with increasing eccentricity (for both versions of MAP estimates). For $e \leq 0.5$, the percentage of wins for the MML estimates is greater than for all other estimates. However, for $e > 0.5$, MAP1 wins most of the time (Figure 4.7c). When comparing to MAP2, the percentage of wins for MML estimates continuously decreases. However, the number of wins for MML estimates is still always in the majority (Figure 4.7d).

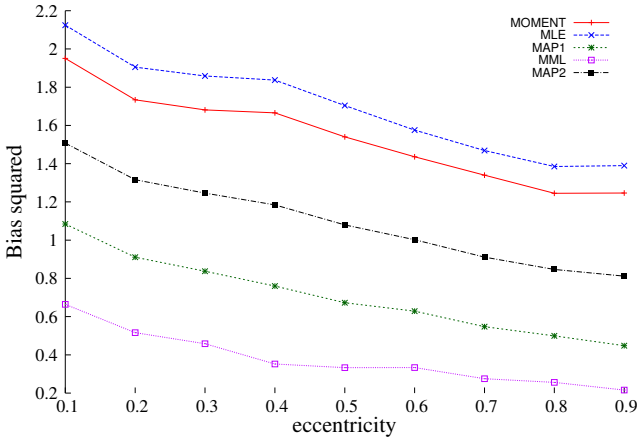
These observations are in contrast to what was observed in the case of $\kappa = 1$ where MML estimates emerged as consistently better estimates. In terms of statistical hypothesis testing, the null hypotheses corresponding to modelling using moment, ML, MAP or MML estimates are accepted at the 1% significance level.

For $\kappa = 100$: The comparison results in this case follow the same pattern as that of $\kappa = 10$ (not illustrated here as they are similar to Figure 4.7).

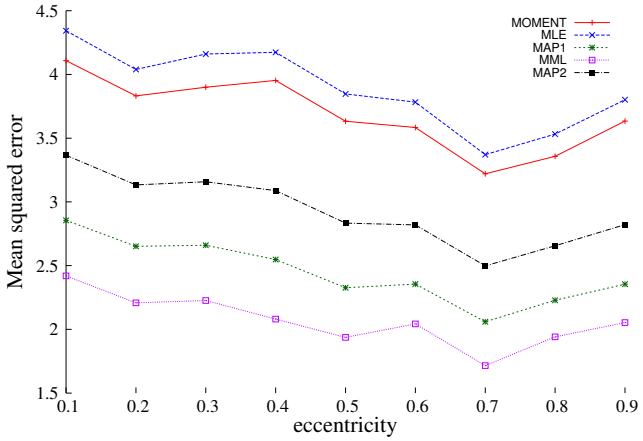
When $\kappa = 10$ and 100, the MAP1 estimates perform competitively compared to the MML estimates (with respect to bias and MSE). Further, the proportion of times MAP1 estimates win with respect to KL distance progressively increases as the eccentricity increases. In general, similar results are observed for $\kappa > 10$. However, as discussed previously, the MAP estimation is subjective to the parameterization of the distribution as shown by the stark contrast between MAP1 and MAP2 estimates by the two parameterizations, even though they are both reasonable. The moment, ML and MML estimates, on the other hand, are not affected by parameterization. Amongst these, MML estimates outperform with respect to all objective metrics as described here.

Varying sample size N, fixed concentration κ and eccentricity e: The behaviour of different estimates with increasing values of sample size from $N = 10$ to $N = 50$ is also explored. We present the results for when $\kappa = 10$. The other values of $\kappa = 1$ and 100 follow the same trend as $\kappa = 10$. The results are discussed for three specific eccentricity values, ranging from low eccentricity ($e = 0.1$), to moderate ($e = 0.5$) to high ($e = 0.9$).

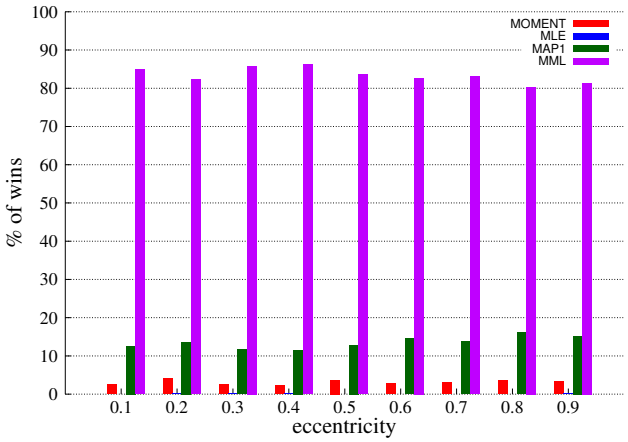
For $e = 0.1$: The comparison results are presented in Figure 4.8 and clearly show how, across all estimators, the bias and MSE decrease as N increases. This is expected since as more data becomes



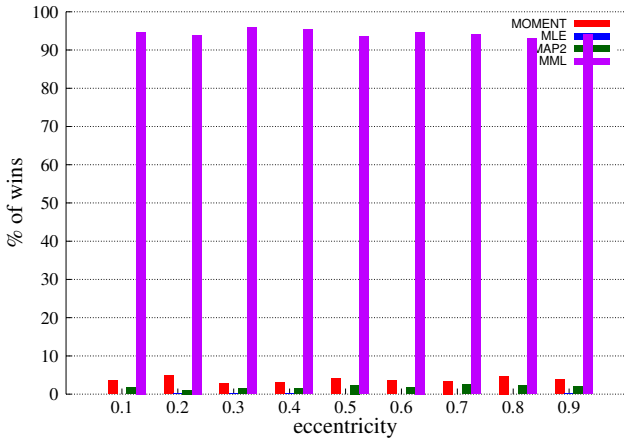
(a) Bias-squared



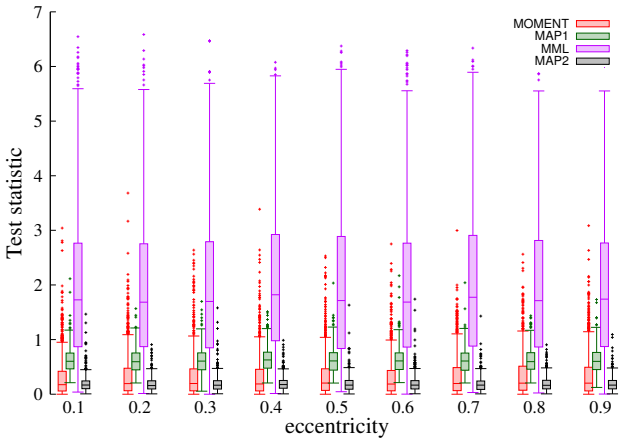
(b) Mean squared error



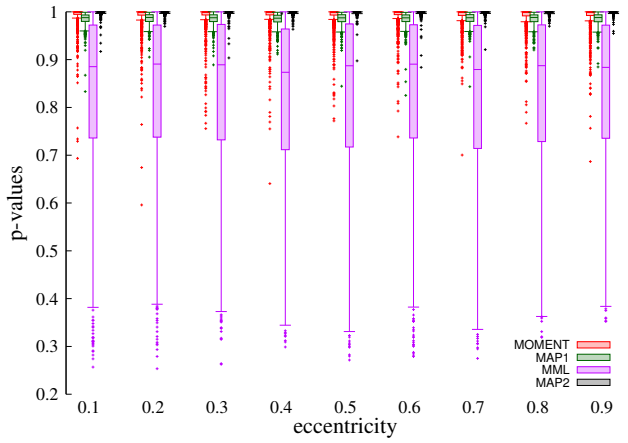
(c) KL distance (MAP version 1)



(d) KL distance (MAP version 2)



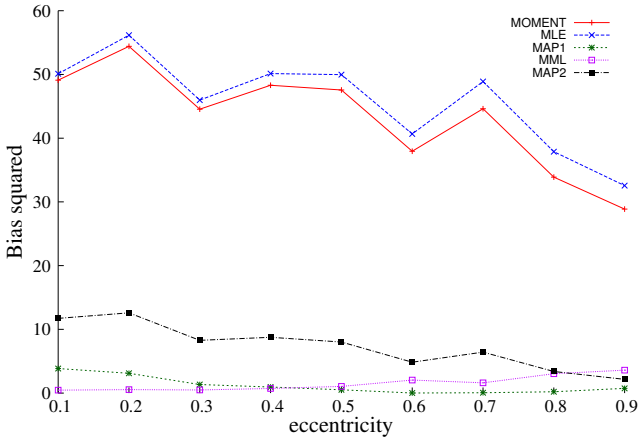
(e) Variation of test statistics



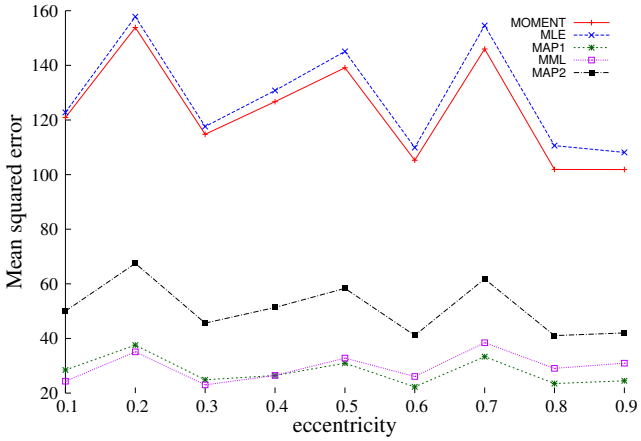
(f) Variation of p-values

Figure 4.6: Comparison of the FB_5 parameter estimates when $N = 10, \kappa = 1$.

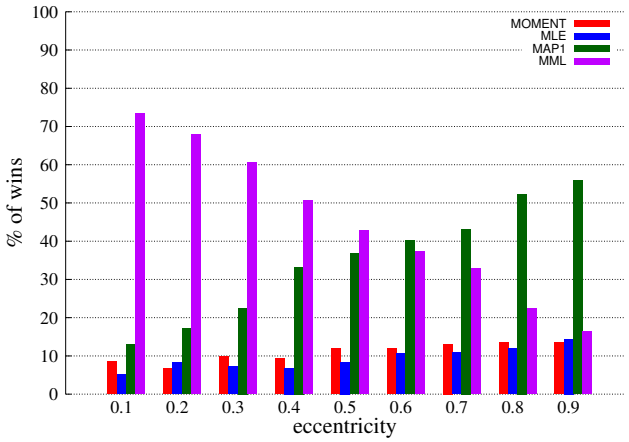
available, the accuracy of estimation increases. Figures 4.8(a) and (b) illustrate that the bias and MSE are prominent for the moment and ML estimators. The bias of MML estimates is close to zero and convincingly lower than both versions of MAP estimates, especially when $N < 25$. The MSE of MML estimates is smaller but close to that of MAP1 estimate.



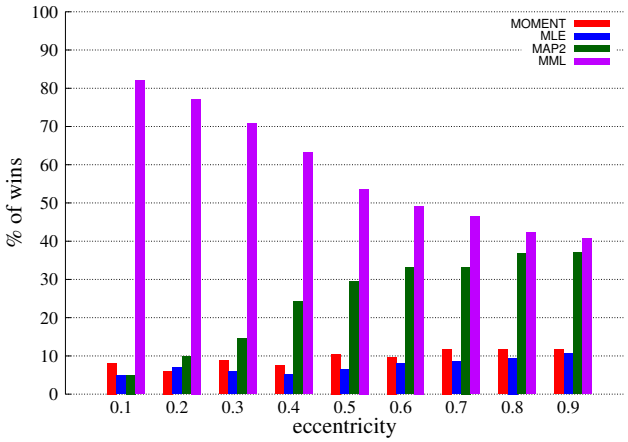
(a) Bias-squared



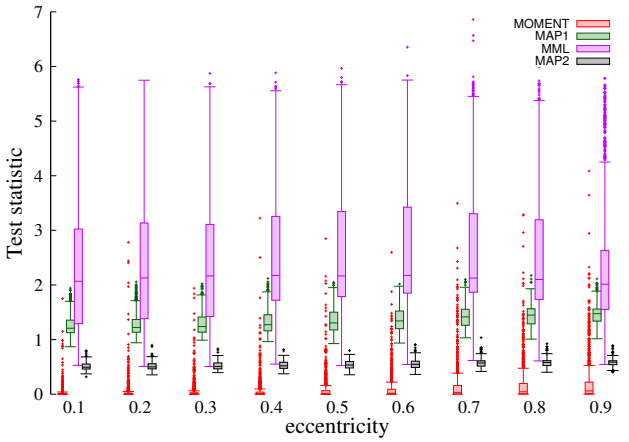
(b) Mean squared error



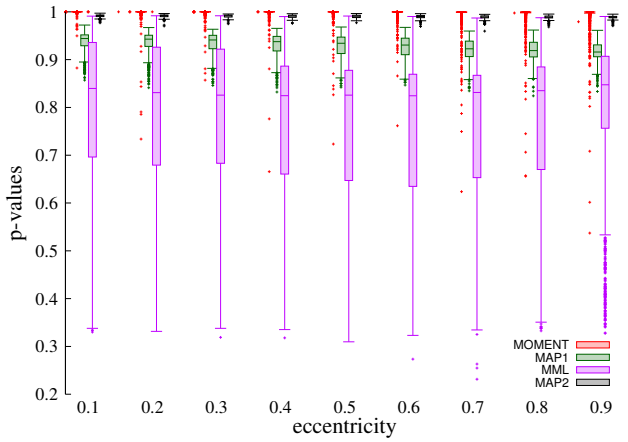
(c) KL distance (MAP version 1)



(d) KL distance (MAP version 2)



(e) Variation of test statistics

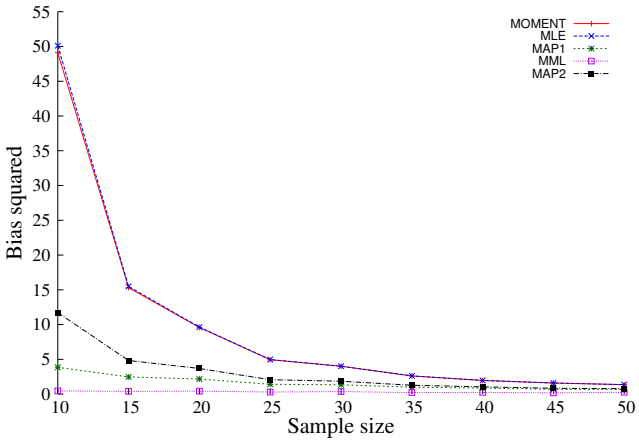


(f) Variation of p-values

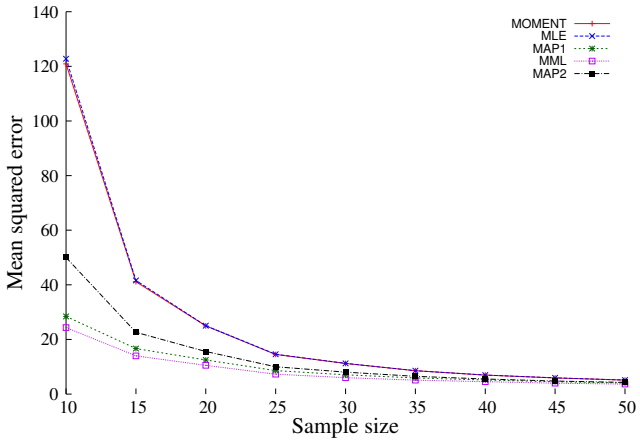
Figure 4.7: Comparison of the FB_5 parameter estimates when $N = 10, \kappa = 10$.

The proportion of wins of MML estimates with respect to KL distance is the highest with values of at least 70% and 80% when compared with MAP1 and MAP2, respectively (see Figures 4.8c,d). Also, hypothesis testing results indicate that the respective estimates constituting the null hypothesis are accepted at the 1% significance level, as observed from the boxplots of test statistics and p-values

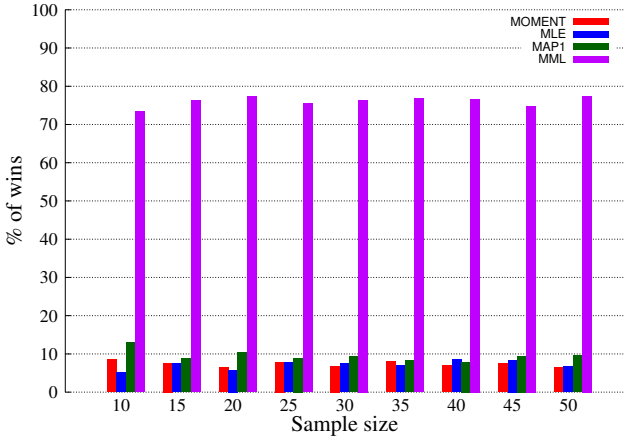
in Figures 4.8(e) and (f).



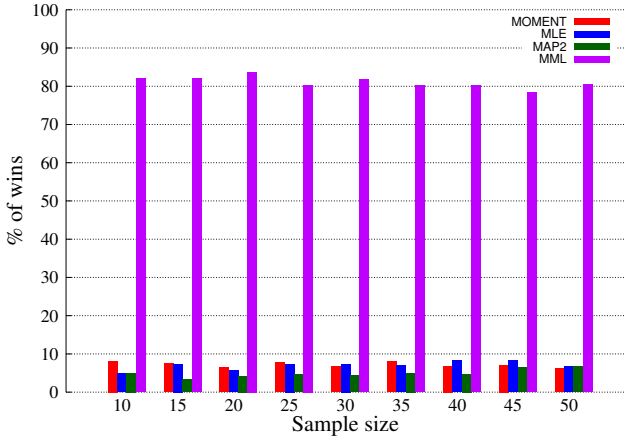
(a) Bias-squared



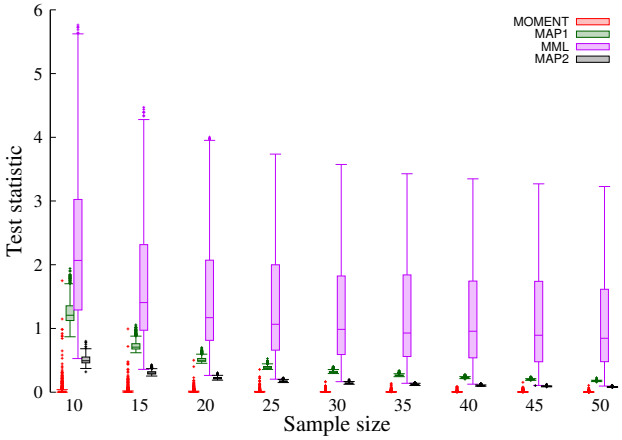
(b) Mean squared error



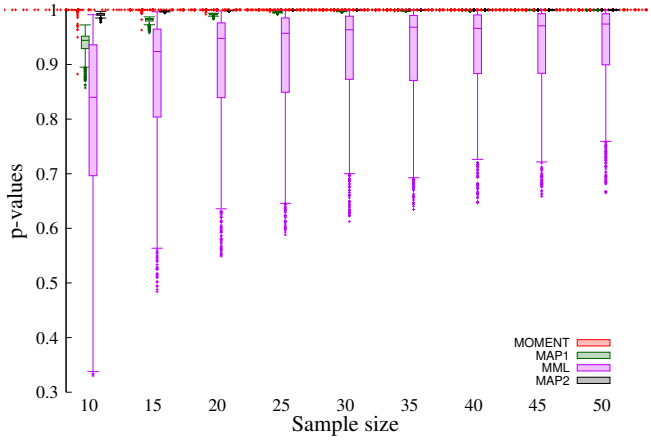
(c) KL distance (MAP version 1)



(d) KL distance (MAP version 2)



(e) Variation of test statistics



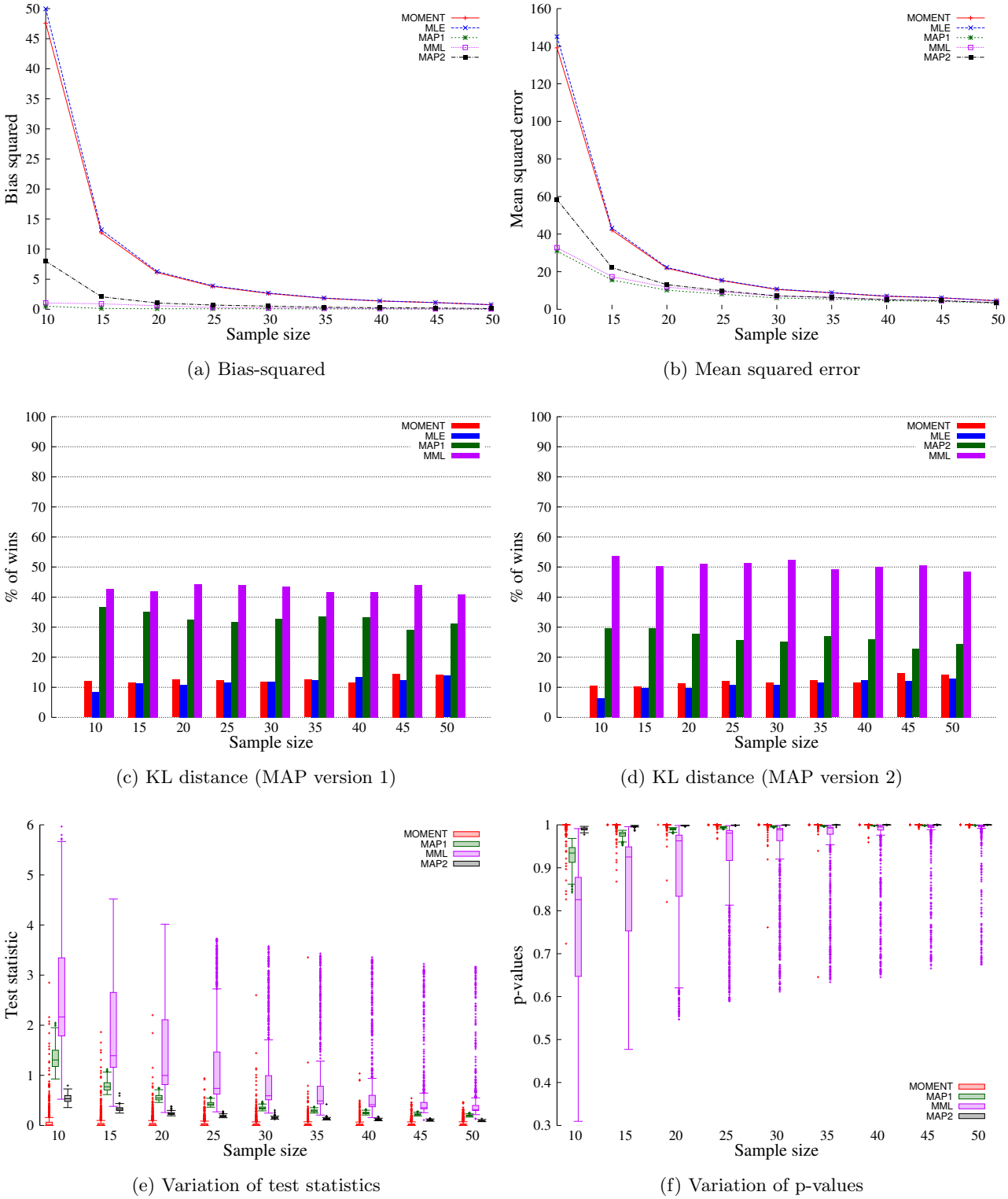
(f) Variation of p-values

Figure 4.8: Comparison of the FB_5 parameter estimates when $\kappa = 10$, eccentricity $= 0.1$.

For $e = 0.5$: The comparison results are presented in Figure 4.9. Similar to the previous case, the bias and MSE of the moment and ML estimates are considerably high compared to those of the MAP and

MML estimates. Also, MAP1 estimates have greater bias and MSE as compared to MAP2 estimates. In this case, the bias and MSE of MAP2 and MML are close to zero.

The proportion of wins of MML estimates with respect to KL distance is higher with about 40% and 50% when compared against MAP1 and MAP2 estimates, respectively. The proportion of wins are, however, lower compared to the previous case when $e = 0.1$ as shown in Figure 4.9(c) and (d).



(a) Bias-squared

(b) Mean squared error

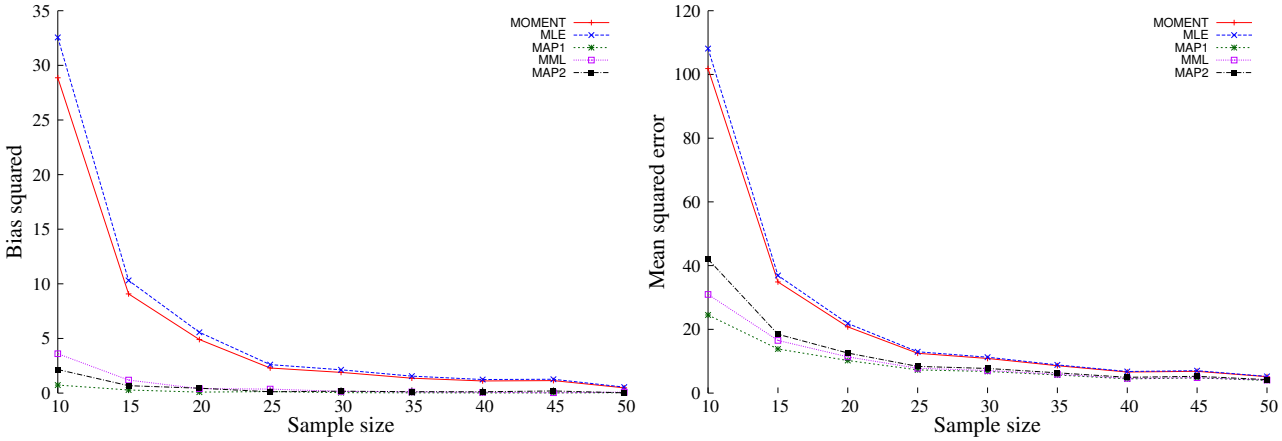
(c) KL distance (MAP version 1)

(d) KL distance (MAP version 2)

(e) Variation of test statistics

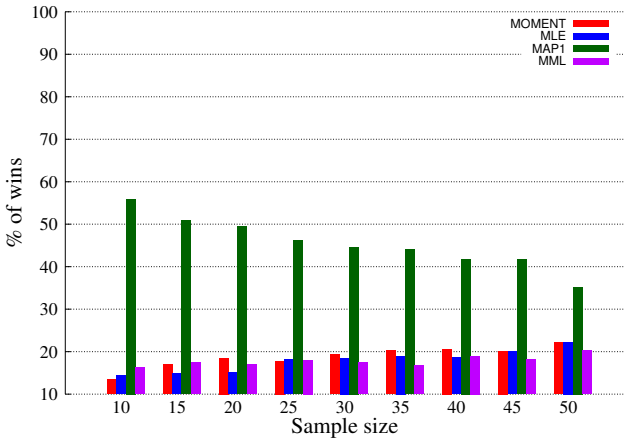
(f) Variation of p-values

Figure 4.9: Comparison of the FB_5 parameter estimates when $\kappa = 10$, eccentricity $= 0.5$.

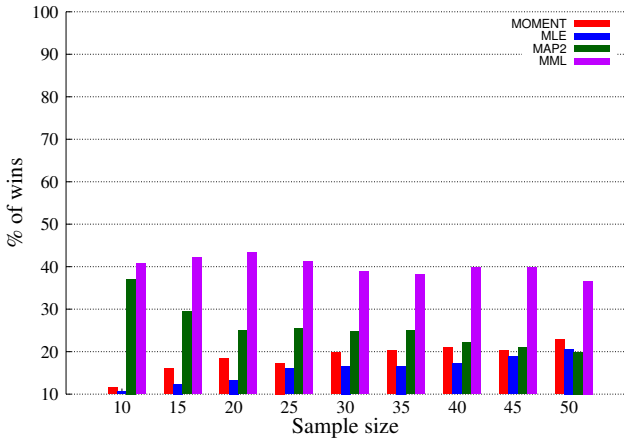


(a) Bias-squared

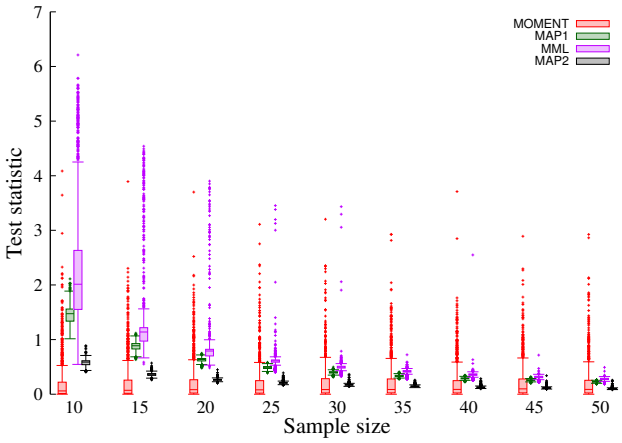
(b) Mean squared error



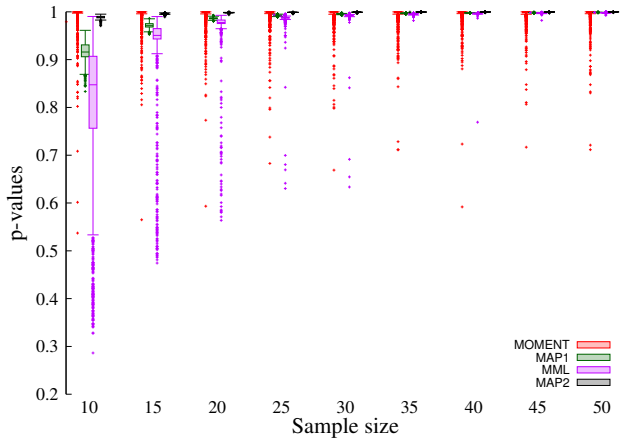
(c) KL distance (MAP version 1)



(d) KL distance (MAP version 2)



(e) Variation of test statistics



(f) Variation of p-values

Figure 4.10: Comparison of the FB_5 parameter estimates when $\kappa = 10$, eccentricity $= 0.9$.

For $e = 0.9$: The comparison results are presented in Figure 4.10. In this case, again, the bias and MSE of the moment and ML estimates are greater compared to others. For $N < 20$, the bias of the MML estimates is greater when compared to those of MAP1 and MAP2 (Figure 4.10a). Further, the MSE of the MML estimates is greater than that of MAP1 and lower than that of MAP2. As the MSE combines the bias and variance terms, there is a trade-off that leads to this result. When $N > 25$, there is almost no difference in the bias and MSE of the MAP and MML estimates.

Also, the proportion of wins of the KL distance for MAP1 is greater than all others (Figure 4.10c). This corresponds to the proportion of wins as illustrated through Figure 4.7(c), similar to the $N = 10, \kappa = 10, e = 0.9$ case. However, when compared with MAP2, the MML estimates have greater proportion of wins $\sim 40\%$ (Figure 4.10d).

The traditional ML estimators are known to have considerable bias, especially at lower sample sizes (Dryden and Mardia, 1998; Dore et al., forthcoming). The ML estimates of κ in the case of a vMF distribution are known to be biased (Schou, 1978; Best and Fisher, 1981; Cordeiro and Vasconcellos, 1999). The MML estimates have been shown to be effective in reducing bias in the case of a vMF distribution (Kasarapu and Allison, 2015). Similarly, the ML estimates of a Bingham distribution, which is a special case of an FB_5 distribution, are also shown to be biased and corrections have been proposed (Cordeiro and Klein, 1994; Kume and Wood, 2007; Dore et al., forthcoming).

As an extension, the empirical tests discussed above demonstrate that for an FB_5 distribution, in comparison to the moment and ML estimates, the MML estimates have lower bias and MSE. Further, when compared to MAP estimates, MML estimates are competitive, particularly considering that MAP estimates are dependent on the parameterization. As a result, the MAP estimates are inconsistent and should therefore be avoided. In contrast, the MML estimates are invariant to alternative parameterizations (Oliver and Baxter, 1994; Wallace, 2005). In this regard, another parameterization is discussed in Appendix B.1 involving all parameters of an FB_5 distribution to further strengthen the case.

4.4 Bivariate von Mises on a 3D torus

The class of bivariate von Mises (BVM) distributions was introduced by Mardia (1975b,c) to model data distributed on the surface of a 3D torus. The study of these distributions has been partly motivated by biologcal research, where it is required to model the protein dihedral angles (see Section 6.4.2). The probability density function of the BVM distribution has the general form

$$f(\mathbf{x};\mathbf{\Theta}) \propto \exp\{\kappa_1 \cos(\theta_1 - \mu_1) + \kappa_2 \cos(\theta_2 - \mu_2) + (\cos\theta_1, \sin\theta_1)\mathbf{A}(\cos\theta_2, \sin\theta_2)^T\} \tag{4.41}$$

where $\mathbf{x} = (\theta_1, \theta_2)$, such that $\theta_1, \theta_2 \in [-\pi, \pi)$ and the parameter vector $\mathbf{\Theta} = (\mu_1, \mu_2, \kappa_1, \kappa_2, \mathbf{A})$, such that $\mu_1, \mu_2 \in [-\pi, \pi)$ are the mean angles, $\kappa_1 \geq 0$ and $\kappa_2 \geq 0$ are the concentration parameters, and $\mathbf{A}$ is a 2×2 real-valued matrix. The term $\exp\{\kappa_1 \cos(\theta_1 - \mu_1)\}$ corresponds to a vMF distribution on a circle ($d = 2$ in Equation 4.1) characterized by the parameters μ_1 and κ_1, Hence, the BVM distribution (Equation 4.41) can be explained as a product of two von Mises circular distributions, with an additional exponential term involving $\mathbf{A}$, that accounts for the correlation.

The general form of the BVM distribution has 8 free parameters. In order to draw an analogy to the bivariate Gaussian distribution (with 5 free parameters), sub-models of the BVM distribution have been proposed by restricting the values that $\mathbf{A}$ can take (Jupp and Mardia, 1980). A 6-parameter version was explored by Rivest (1988) and has the form

$$\begin{aligned} f(\mathbf{x};\mathbf{\Theta}) \propto \exp\{&\kappa_1 \cos(\theta_1 - \mu_1) + \kappa_2 \cos(\theta_2 - \mu_2) + \alpha \cos(\theta_1 - \mu_1)\cos(\theta_2 - \mu_2) \\ &+ \beta \sin(\theta_1 - \mu_1)\sin(\theta_2 - \mu_2)\} \end{aligned} \tag{4.42}$$

In particular, when $\alpha = 0$ and $\beta = \lambda$, the above density reduces to the following 5-parameter version, which is called the BVM *Sine* model (Singh et al., 2002).

$$f(\mathbf{x};\mathbf{\Theta}) = c(\kappa_1, \kappa_2, \lambda)^{-1} \exp\{\kappa_1 \cos(\theta_1 - \mu_1) + \kappa_2 \cos(\theta_2 - \mu_2) + \lambda \sin(\theta_1 - \mu_1)\sin(\theta_2 - \mu_2)\} \tag{4.43}$$

where $c(\kappa_1, \kappa_2, \lambda)$ is the normalization constant of the distribution defined as

$$c(\kappa_1, \kappa_2, \lambda) = 4\pi^2 \sum_{j=0}^{\infty} \binom{2j}{j} \left(\frac{\lambda^2}{4\kappa_1\kappa_2}\right)^j I_j(\kappa_1) I_j(\kappa_2) \tag{4.44}$$

and I_v is the modified Bessel function of first kind and order v. The 5-parameter vector will be $\mathbf{\Theta} = (\mu_1, \mu_2, \kappa_1, \kappa_2, \lambda)$ where λ is a real number. If $\lambda = 0$, the probability density function (Equation 4.43) will just be the product of two independent von Mises circular distributions, and corresponds to the case when there is no correlation between the two variables θ_1 and θ_2. The probability density function in such a case is given as

$$f(\mathbf{x}; \mathbf{\Theta}) = c(\kappa_1, \kappa_2)^{-1} \exp\{\kappa_1 \cos(\theta_1 - \mu_1) + \kappa_2 \cos(\theta_2 - \mu_2)\} \tag{4.45}$$

where $c(\kappa_1, \kappa_2)$ is the normalization constant defined as $c(\kappa_1, \kappa_2) = \frac{1}{2\pi I_0(\kappa_1)} \frac{1}{2\pi I_0(\kappa_2)}$. The normalization constant corresponds to the product of $C_2(\kappa_1)$ and $C_2(\kappa_2)$, which are the normalization constants for the respective von Mises Fisher distributions with $d = 2$ (see Equation 4.1).

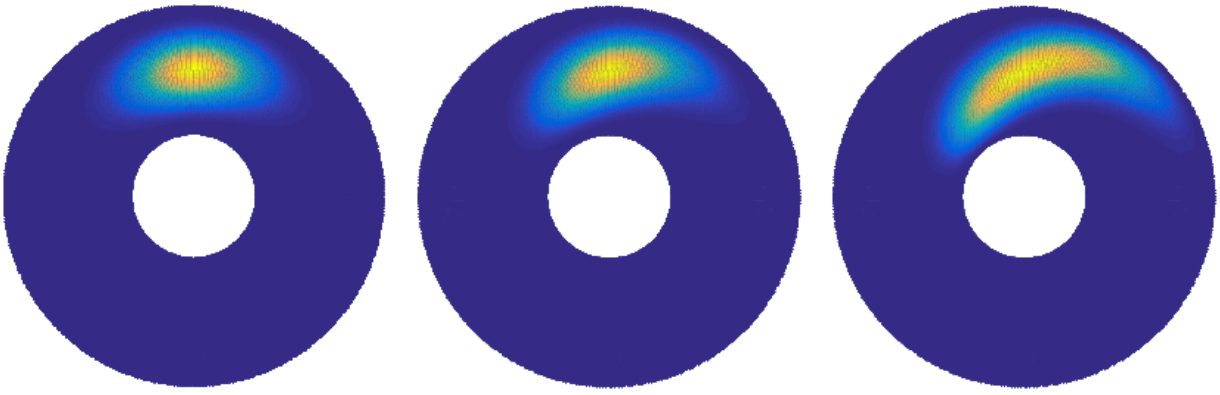
Alternatively, when $\alpha = -\beta$, the form of Equation 4.42 results in a different reduced form called the BVM *Cosine* model (Mardia et al., 2007). The Sine and the Cosine models serve as natural analogues of the bivariate Gaussian distribution on the 3D torus. In fact, for huge concentrations, Singh et al. (2002) approximate the Sine model to a bivariate Gaussian distribution with the 2×2 covariance matrix $\mathbf{C} = [c_{ij}], i, j \in \{1, 2\}$, whose elements are given by

$$c_{11} = \frac{\kappa_2}{\kappa_1\kappa_2 - \lambda^2}, \quad c_{22} = \frac{\kappa_1}{\kappa_1\kappa_2 - \lambda^2}, \quad c_{12} = c_{21} = \frac{\lambda}{\kappa_1\kappa_2 - \lambda^2}$$

The limiting case approximation is valid when $\kappa_1\kappa_2 > \lambda^2$. Also, from the covariance matrix, the correlation coefficient ρ can be determined as (Pearson, 1895):

$$\rho = \frac{c_{12}}{\sqrt{c_{11}c_{22}}} = \frac{\lambda}{\sqrt{\kappa_1\kappa_2}} \quad \text{such that} \quad |\rho| < 1 \tag{4.46}$$

In order to better understand the interaction of κ_1, κ_2, and the correlation coefficient ρ, we provide an example in Figure 4.11, where the distribution is shown for values of $\rho = 0.1$ (low correlation), $\rho = 0.5$ (moderate correlation), and $\rho = 0.9$ (high correlation). Note that ρ can take negative values, in which case the resultant distribution will just be a reflection in some axis (Mardia et al., 2007).



(a) $\rho = 0.1$ (b) $\rho = 0.5$ (c) $\rho = 0.9$

Figure 4.11: BVM Sine model showing different correlations. The distribution has $\mu_1 = \mu_2 = \frac{\pi}{2}$ and $\kappa_1 = \kappa_2 = 10$. For each value of ρ, the corresponding value of $\lambda = \rho\sqrt{\kappa_1\kappa_2}$.

The modelling of directional data using the BVM Sine and Cosine models has been previously explored by Mardia et al. (2007). In this section, we discuss the MML inference using the BVM Sine distributions. This involves deriving the MML estimates of the parameters of the distribution and subsequently using them to model the protein dihedral angles (see Section 6.4.2).

4.4.1 Maximum likelihood parameter estimation

In applications involving modelling directional data using the BVM Sine distributions, the maximum likelihood (ML) estimates are typically used (Boomsma et al., 2006; Mardia et al., 2007, 2008). In the case of the FB_5 distribution (Section 4.3.2), we explored the moment, ML, and MAP-based estimation. For BVM Sine distributions, the moment and ML estimates are the same, as the BVM Sine distribution belongs to the exponential family of distributions (Mardia et al., 2008).

Given data $\mathcal{D} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$, where $\mathbf{x}_i = (\theta_{i1}, \theta_{i2})$, the ML estimates of the parameter vector $\mathbf{\Theta} = (\mu_1, \mu_2, \kappa_1, \kappa_2, \lambda)$ are obtained by minimizing the negative log-likelihood expression of the data given by

$$\begin{aligned}\mathcal{L}(\mathcal{D}|\mathbf{\Theta}) =& N \log c(\kappa_1, \kappa_2, \lambda) - \kappa_1 \sum_{i=1}^{N} \cos(\theta_{i1} - \mu_1) - \kappa_2 \sum_{i=1}^{N} \cos(\theta_{i2} - \mu_2) \\ & - \lambda \sum_{i=1}^{N} \sin(\theta_{i1} - \mu_1) \sin(\theta_{i2} - \mu_2) \end{aligned} \tag{4.47}$$

The ML estimates satisfy $\dfrac{\partial \mathcal{L}}{\partial \mathbf{\Theta}} = 0$. However, as no closed form soultions exist because of the complicated form of $c(\kappa_1, \kappa_2, \lambda)$, an optimization library is used. In this thesis, NLopt[4] is used to numerically compute the estimates.

4.4.2 Maximum *a posteriori* probability (MAP) estimation

For an independent and identically distributed sample $\mathcal{D}$, the MAP estimates are obtained by maximizing the posterior density $\Pr(\mathbf{\Theta}|\mathcal{D})$. The procedure for MAP estimation is described in Section 2.2.2. This requires the definition of a reasonable prior $\Pr(\mathbf{\Theta})$ on the parameter space. As shown in the case of the FB_5 distribution (Section 4.3.3), the MAP estimates vary depending on the parameterization of the probability distribution. We consider two alternative parameterizations in the case of the BVM Sine distribution.

Prior on the angular parameters μ_1 and μ_2: Since $\mu_1, \mu_2 \in [-\pi, \pi)$, a uniform prior can be assumed in this range for each of the means. Further, assuming μ_1 and μ_2 to be independent of each other, their joint prior will be $\Pr(\mu_1, \mu_2) = \dfrac{1}{4\pi^2}$.

Prior on the scale parameters κ_1, κ_2, and λ: As discussed for Equation 4.41, the BVM density function can be regarded as a product of two von Mises circular distributions with an additional term that captures the correlation. In the Bayesian analysis of the von Mises circular distribution, Wallace and Dowe (1994b) used the prior on the concentration parameter κ as $\Pr(\kappa) = \dfrac{\kappa}{(1+\kappa^2)^{3/2}}$. In the current context of defining priors on κ_1 and κ_2 for a BVM distribution, we use the prior $\Pr(\kappa)$. Assuming κ_1 and κ_2 to be independent of each other, the joint prior is given by

$$\Pr(\kappa_1, \kappa_2) = \frac{\kappa_1 \kappa_2}{(1+\kappa_1^2)^{3/2}(1+\kappa_2^2)^{3/2}}$$

[4] `http://ab-initio.mit.edu/nlopt`

In order to define a reasonable prior on λ, we use the fact that $\lambda^2 < \kappa_1\kappa_2$ (see Equation 4.46). Hence, the conditional probability density of λ is given as: $\Pr(\lambda|\kappa_1,\kappa_2) = \dfrac{1}{2\sqrt{\kappa_1\kappa_2}}$. Therefore, the joint prior density of the scalar parameters κ_1, κ_2 and λ is

$$\Pr(\kappa_1,\kappa_2,\lambda) = \Pr(\kappa_1,\kappa_2)\Pr(\lambda|\kappa_1,\kappa_2) = \frac{\sqrt{\kappa_1\kappa_2}}{2(1+\kappa_1^2)^{3/2}(1+\kappa_2^2)^{3/2}}$$

Using the product of the priors for the angular and the scale parameters, that is, $\Pr(\mu_1,\mu_2)$ and $\Pr(\kappa_1,\kappa_2,\lambda)$, the joint prior of the parameter vector $\mathbf{\Theta}$, is given by

$$\Pr(\mathbf{\Theta}) = \Pr(\mu_1,\mu_2,\kappa_1,\kappa_2,\lambda) = \frac{\sqrt{\kappa_1\kappa_2}}{8\pi^2(1+\kappa_1^2)^{3/2}(1+\kappa_2^2)^{3/2}} \tag{4.48}$$

The prior density $\Pr(\mathbf{\Theta})$ can be used along with the likelihood function to formulate the posterior density $\Pr(\mathbf{\Theta}|\mathcal{D})$ (see Section 2.2.2). The MAP estimates correspond to the maximized value of the posterior $\Pr(\mathbf{\Theta}|\mathcal{D})$.

Non-linear transformations of the parameter space

We consider non-linear transformations of the parameter space, in order to demonstrate that the MAP estimates are not invariant in different parameterizations of the probability distribution. For the FB_5 distribution, we explained a simple non-linear transformation involving the eccentricity parameter (Section 4.3.3). In the current context of a BVM Sine distribution, we discuss a similar non-linear transformation of the parameter space involving the correlation parameter λ. Additionally, we also describe a parameterization that transforms all the five parameters.

An alternative parameterization involving λ: The BVM Sine probability density function (Equation 4.43) can be reparameterized in terms of the correlation coefficient ρ, instead of λ, by using the relationship $\lambda = \rho\sqrt{\kappa_1\kappa_2}$ (as per Equation 4.46). If $\mathbf{\Theta}' = (\mu_1,\mu_2,\kappa_1,\kappa_2,\rho)$ denotes the modified vector of parameters, the modified prior density $\Pr(\mathbf{\Theta}')$ is obtained by dividing $\Pr(\mathbf{\Theta})$ with the Jacobian of the transformation $J = \dfrac{\partial\rho}{\partial\lambda} = \dfrac{1}{\sqrt{\kappa_1\kappa_2}}$ as follows

$$\Pr(\mathbf{\Theta}') = \frac{\Pr(\mu_1,\mu_2,\kappa_1,\kappa_2,\lambda)}{J} = \frac{\kappa_1\kappa_2}{8\pi^2(1+\kappa_1^2)^{3/2}(1+\kappa_2^2)^{3/2}} \tag{4.49}$$

With this transformation, the posterior density $\Pr(\mathbf{\Theta}'|\mathcal{D})$ can be computed, and subsequently used to determine the MAP estimates (see Equation 2.3).

An alternative parameterization involving $\mathbf{\Theta}$: In addition to the transformation of the correlation parameter λ, we study another transformation that was proposed by Rosenblatt (1952). The method transforms a given continuous k-variate probability distribution into the uniform distribution on the k-dimensional *unit* hypercube. Such a transformation applied on the prior density of the parameter vector $\mathbf{\Theta}$ results in the prior transforming to a uniform distribution. Hence, estimation in this transformed parameter space is equivalent to the corresponding maximum likelihood estimation.

For the 5-parameter vector $\mathbf{\Theta} = (\mu_1, \mu_2, \kappa_1, \kappa_2, \lambda)$, the Rosenblatt (1952) transformation to $\mathbf{\Theta}'' = (z_1, z_2, z_3, z_4, z_5)$ involves computing the cumulative densities F_i, $\forall\, i \in \{1, \ldots, 5\}$ as follows

$$
\begin{aligned}
z_1 &= \Pr(X_1 \leq \mu_1) = F_1(\mu_1) \\
z_2 &= \Pr(X_2 \leq \mu_2 | X_1 = \mu_1) = F_2(\mu_2|\mu_1) \\
z_3 &= \Pr(X_3 \leq \kappa_1 | X_2 = \mu_2, X_1 = \mu_1) = F_3(\kappa_1|\mu_2, \mu_1) \\
z_4 &= \Pr(X_4 \leq \kappa_2 | X_3 = \kappa_1, X_2 = \mu_2, X_1 = \mu_1) = F_4(\kappa_2|\kappa_1, \mu_2, \mu_1) \\
z_5 &= \Pr(X_5 \leq \lambda | X_4 = \kappa_2, X_3 = \kappa_1, X_2 = \mu_2, X_1 = \mu_1) = F_5(\lambda|\kappa_2, \kappa_1, \mu_2, \mu_1)
\end{aligned}
$$

As the cumulative densities are bounded by 1, the above transformation results in $0 \leq z_i \leq 1$, $i = \{1, \ldots, 5\}$. Further, Rosenblatt (1952) argue that each z_i is uniformly and independently distributed on $[0, 1]$, so that the prior density in this transformed parameter space is

$$\Pr(\mathbf{\Theta}'') = \Pr(z_1, z_2, z_3, z_4, z_5) = 1 \tag{4.50}$$

In order to achieve such a transformation, we need to express z_i in terms of the original parameters. Based on the assumptions made in the formulation of the prior $\Pr(\mathbf{\Theta})$, we derive the following relationships:

$$z_1 = \int_{-\pi}^{\mu_1} \frac{1}{2\pi}\, d\mu_1 = \frac{\mu_1 + \pi}{2\pi} \implies \mu_1 = \pi(2z_1 - 1) \quad \text{and} \quad z_2 = \frac{\mu_2 + \pi}{2\pi} \implies \mu_2 = \pi(2z_2 - 1)$$

Based on the independence assumption in the formulation of priors of angular and scale parameters, we have $z_3 = F_3(\kappa_1|\mu_2, \mu_1) = F_3(\kappa_1)$, and therefore we have

$$
\begin{gathered}
z_3 = \int_0^{\kappa_1} \Pr(\kappa)\, d\kappa = \int_0^{\kappa_1} \frac{\kappa}{(1+\kappa^2)^{3/2}} d\kappa = 1 - \cos(\arctan \kappa_1) \\
\text{Hence, } \kappa_1 = \tan(\arccos(1 - z_3)) \quad \text{and} \quad \kappa_2 = \tan(\arccos(1 - z_4))
\end{gathered}
$$

Further, $F_5(\lambda|\kappa_2, \kappa_1, \mu_2, \mu_1) = F_5(\lambda|\kappa_2, \kappa_1)$, as λ is independent of μ_1 and μ_2. Hence, the invertible transformation corresponding to λ is as follows

$$z_5 = F_5(\lambda|\kappa_2, \kappa_1) = \int_{-\sqrt{\kappa_1\kappa_2}}^{\lambda} \frac{1}{2\sqrt{\kappa_1\kappa_2}}\, d\lambda = \frac{1}{2}\left(\frac{\lambda}{\sqrt{\kappa_1\kappa_2}} + 1\right)$$

so that λ can be expressed as a function of z_3, z_4, and z_5. The transformed BVM Sine probability density function $f(\mathbf{x}, \mathbf{\Theta}'')$ is obtained by substituting the expressions of $\mathbf{\Theta}$ in terms of $z_i, 1 \leq i \leq 5$ in $f(\mathbf{x}, \mathbf{\Theta})$ (Equation 4.43).

In summary, we consider two additional parameterizations of the BVM Sine probability density. For statistical invariance, the estimates of the parameters should be affected by the same transformation in alternative parameterizations. The MAP estimation does not satisy this property, as illustrated in the context of estimating the parameters of a FB_5 distribution (see Section 4.3.3). The same behaviour is exhibited in the current context of estimating the parameters of a BVM Sine distribution. We describe this behaviour through an example.

An example demonstrating the effects of alternative parameterizations

An example of estimating parameters using the posterior distributions resulting from the various prior densities (Equations 4.48 - 4.50) is described here. A random sample of size $N = 10$ is generated from a BVM Sine distribution (Singh et al., 2002). The true parameters of the distribution are $\mu_1 = \mu_2 = \pi/2$, $\kappa_1 = \kappa_2 = 10$, and $\lambda = 9$ (corresponding to a correlation coefficient of $\rho = 0.9$).

The MAP estimators are obtained by maximizing the posterior densities using the non-linear optimization library NLopt (Johnson, 2014) in conjunction with derivative-free optimization (Powell, 1994). The differences in the estimates are explained below.

We observe that the estimates of the angular parameters, μ_1 and μ_2, are similar across the different parameterizations, with values close to 1.730 and 1.695 radians respectively. In the case of using $\boldsymbol{\Theta}''$, the estimated values $\widehat{z}_1$ and $\widehat{z}_2$ are transformed back into $\widehat{\mu}_1$ and $\widehat{\mu}_2$ to allow comparison of similar quantities.

$$\widehat{\mu}_1 = 1.730,\ \widehat{\mu}_2 = 1.695 \text{ using } \Pr(\boldsymbol{\Theta})$$
$$\widehat{\mu}_1 = 1.731,\ \widehat{\mu}_2 = 1.696 \text{ using } \Pr(\boldsymbol{\Theta}')$$
$$\widehat{z}_1 = 0.276, \widehat{z}_2 = 0.270 \implies \widehat{\mu}_1 = 1.735,\ \widehat{\mu}_2 = 1.698 \text{ using } \Pr(\boldsymbol{\Theta}'')$$

The estimation of the scale parameters, κ_1, κ_2, and λ however, results in different values. We observe that, in the case of $\Pr(\boldsymbol{\Theta}')$, $\widehat{\rho} = 0.684$, which translates to $\widehat{\lambda} = 6.565$. This is different from the estimated value of $\widehat{\lambda} = 5.017$ using $\Pr(\boldsymbol{\Theta})$. The values of $\widehat{\kappa}_1$ and $\widehat{\kappa}_2$ are also different. Further, with $\Pr(\boldsymbol{\Theta}'')$, the transformation of estimated z_i into the $\boldsymbol{\Theta}$ parameter space result in different estimates.

$$\widehat{\kappa}_1 = 4.451,\ \widehat{\kappa}_2 = 14.158,\ \widehat{\lambda} = 5.017 \text{ using } \Pr(\boldsymbol{\Theta})$$
$$\widehat{\kappa}_1 = 5.311,\ \widehat{\kappa}_2 = 17.338,\ \widehat{\rho} = 0.684 \implies \widehat{\lambda} = 6.565 \text{ using } \Pr(\boldsymbol{\Theta}')$$
$$\widehat{z}_3 = 0.900, \widehat{z}_4 = 0.970, \widehat{z}_5 = 0.924 \implies \widehat{\kappa}_1 = 9.998,\ \widehat{\kappa}_2 = 33.931,\ \widehat{\lambda} = 15.628 \text{ using } \Pr(\boldsymbol{\Theta}'')$$

The above example demonstrates a drawback of the MAP-based estimation with respect to parameter invariance. As discussed in Section 2.2.2, the MAP estimator corresponds to the mode of the posterior distribution. The mode is, however, not invariant under varying parameterizations. We use the above parameterizations in analyzing the behaviour of the various estimators in the experiments section (Section 4.4.5).

4.4.3 MML-based parameter estimation

In this section, the derivation of the MML-based parameter estimates of a BVM Sine distribution is described. As explained in Section 2.4.2, the derivation of the MML estimates requires the formulation of the message length expression (Equation 2.8) for encoding some observed data using the BVM Sine distribution.

The formulation requires the use of a suitable prior density on the parameters. We use the parameterization $\boldsymbol{\Theta}$ and the corresponding prior $\Pr(\boldsymbol{\Theta})$ that was formulated in the MAP analyses in Section 4.4.2. It is to be noted that the MML estimation is invariant to the parameterization used (Oliver and Baxter, 1994).

Notations: Before describing the MML approach, the following notations are defined as these are used in the following discussion. The partial derivatives of the normalization constant $c(\kappa_1, \kappa_2, \lambda)$ of the BVM Sine distribution would be required later on. The following are the notations adopted to represent them.

$$c(\kappa_1, \kappa_2, \lambda) = c, \quad c_{\kappa_1} = \partial c/\partial \kappa_1, \quad c_{\kappa_2} = \partial c/\partial \kappa_2, \quad c_{\lambda} = \partial c/\partial \lambda$$
$$c_{\kappa_1\kappa_1} = \partial^2 c/\partial \kappa_1^2, \quad c_{\kappa_2\kappa_2} = \partial^2 c/\partial \kappa_2^2, \quad c_{\lambda\lambda} = \partial^2 c/\partial \lambda^2,$$
$$c_{\kappa_1\kappa_2} = \partial^2 c/\partial \kappa_1 \partial \kappa_2, \quad c_{\kappa_1\lambda} = \partial^2 c/\partial \kappa_1 \partial \lambda, \quad c_{\kappa_2\lambda} = \partial^2 c/\partial \kappa_2 \partial \lambda$$

We also require the determinant of the Fisher information for the MML estimation of parameters. We use the above notations in the following computation of the Fisher information. The computation of these partial derivatives is explained in Section 4.4.4.

Computation of Expectations

In order to proceed with the derivation of the Fisher information, we first outline the derivation of some of the required *expectation* quantities. For random variables θ_1, θ_2 sampled from the BVM Sine distribution (Equation 4.43), we compute the following quantities: $\mathbb{E}[\cos(\theta_1 - \mu_1)]$, $\mathbb{E}[\cos(\theta_2 - \mu_2)]$, $\mathbb{E}[\cos(\theta_1 - \mu_1)\cos(\theta_2 - \mu_2)]$, and $\mathbb{E}[\sin(\theta_1 - \mu_1)\sin(\theta_2 - \mu_2)]$.

Singh et al. (2002) derived the normalization constant as an infinite series expansion given by Equation 4.44. We use the following *integral* form of the normalization constant to derive the above mentioned expectations, as a function of κ_1, κ_2, and λ.

$$c(\kappa_1, \kappa_2, \lambda) = \int_{-\pi}^{\pi}\int_{-\pi}^{\pi} \exp\{\kappa_1 \cos(\theta_1 - \mu_1) + \kappa_2 \cos(\theta_2 - \mu_2) + \lambda \sin(\theta_1 - \mu_1)\sin(\theta_2 - \mu_2)\}\, d\theta_2\, d\theta_1$$

On differentiating the above integral with respect to κ_1, we get

$$\begin{aligned}\frac{\partial}{\partial \kappa_1} c(\kappa_1, \kappa_2, \lambda) &= \int_{-\pi}^{\pi}\int_{-\pi}^{\pi} \cos(\theta_1 - \mu_1) \exp\{\kappa_1 \cos(\theta_1 - \mu_1) + \kappa_2 \cos(\theta_2 - \mu_2) \\ &\qquad + \lambda \sin(\theta_1 - \mu_1)\sin(\theta_2 - \mu_2)\}\, d\theta_2\, d\theta_1 \\ &= c(\kappa_1, \kappa_2, \lambda)\ \mathbb{E}[\cos(\theta_1 - \mu_1)]\end{aligned}$$

Hence, the expectation can be represented using the above defined notation as

$$\begin{aligned}\mathbb{E}[\sin(\theta_1 - \mu_1)] = 0 = \mathbb{E}[\sin(\theta_2 - \mu_2)] \\ \mathbb{E}[\cos(\theta_1 - \mu_1)] = \frac{1}{c(\kappa_1, \kappa_2, \lambda)} \frac{\partial c(\kappa_1, \kappa_2, \lambda)}{\partial \kappa_1} = \frac{c_{\kappa_1}}{c} \\ \text{Similarly,}\quad \mathbb{E}[\cos(\theta_2 - \mu_2)] = \frac{c_{\kappa_2}}{c} \quad \text{and} \quad \mathbb{E}[\sin(\theta_1 - \mu_1)\sin(\theta_2 - \mu_2)] = \frac{c_\lambda}{c}\end{aligned} \tag{4.51}$$

On differentiating twice the integral form of $c(\kappa_1, \kappa_2, \lambda)$ with respect to κ_1, κ_2, and λ, we get the following relationships

$$\begin{aligned}\mathbb{E}[\cos(\theta_1 - \mu_1)\cos(\theta_2 - \mu_2)] = \frac{c_{\kappa_1\kappa_2}}{c}, \\ \mathbb{E}[\cos(\theta_1 - \mu_1)\sin(\theta_2 - \mu_2)] = 0 = \mathbb{E}[\sin(\theta_1 - \mu_1)\cos(\theta_2 - \mu_2)]\end{aligned} \tag{4.52}$$

Computation of the Fisher information

As described in Section 2.4.2, the computation of the *determinant* of the Fisher information matrix requires the evaluation of the second order partial derivatives of the negative log-likelihood function with respect to the parameters of the distribution. As per the density function (Equation 4.43), the negative log-likelihood of a datum $\mathbf{x} = (\theta_1, \theta_2)$ is given by

$$\mathcal{L}(\mathbf{x}|\mathbf{\Theta}) = \log c(\kappa_1, \kappa_2, \lambda) - \kappa_1 \cos(\theta_1 - \mu_1) - \kappa_2 \cos(\theta_2 - \mu_2) - \lambda \sin(\theta_1 - \mu_1)\sin(\theta_2 - \mu_2) \tag{4.53}$$

where $\mathbf{\Theta} = (\mu_1, \mu_2, \kappa_1, \kappa_2, \lambda)$ as indicated before. Let $\mathcal{F}_1(\mathbf{\Theta})$ denote the Fisher information for a *single* observation. the Fisher information matrix $\mathcal{F}_1(\mathbf{\Theta})$ in the case of an FB_5 distribution is a 5×5 *symmetric* matrix. Further, the determinant $|\mathcal{F}_1(\mathbf{\Theta})|$ is decomposed as a product of $|\mathcal{F}_A|$ and $|\mathcal{F}_S|$, where $\mathcal{F}_A$ is the Fisher matrix associated with the angular parameters μ_1 and μ_2, and $\mathcal{F}_S$ is the Fisher matrix associated with the scale parameters κ_1, κ_2, and λ.

Fisher matrix ($\mathcal{F}_A$) associated with μ_1, μ_2: $\mathcal{F}_A$ is a 2×2 symmetric matrix whose elements are the expected values of the second order partial derivatives of $\mathcal{L}$ with respect to μ_1 and μ_2. On

differentiating Equation 4.53 with respect to μ_1, we get

$$\frac{\partial \mathcal{L}}{\partial \mu_1} = -\kappa_1 \sin(\theta_1 - \mu_1) + \lambda \cos(\theta_1 - \mu_1) \sin(\theta_2 - \mu_2) \tag{4.54}$$

$$\text{and} \quad \frac{\partial^2 \mathcal{L}}{\partial \mu_1^2} = \kappa_1 \cos(\theta_1 - \mu_1) + \lambda \sin(\theta_1 - \mu_1) \sin(\theta_2 - \mu_2)$$

$$\text{Hence,} \quad \mathcal{F}_{\mu_1\mu_1} = \mathbb{E}\left[\frac{\partial^2 \mathcal{L}}{\partial \mu_1^2}\right] = \kappa_1 \, \mathbb{E}[\cos(\theta_1 - \mu_1)] + \lambda \, \mathbb{E}[\sin(\theta_1 - \mu_1) \sin(\theta_2 - \mu_2)]$$

$$= \kappa_1 \frac{c_{\kappa_1}}{c} + \lambda \frac{c_\lambda}{c}$$

$$\text{Similarly,} \quad \mathcal{F}_{\mu_2\mu_2} = \mathbb{E}\left[\frac{\partial^2 \mathcal{L}}{\partial \mu_2^2}\right] = = \kappa_2 \frac{c_{\kappa_2}}{c} + \lambda \frac{c_\lambda}{c} \tag{4.55}$$

On taking the derivative of Equation 4.54 with respect to μ_2, we get

$$\frac{\partial^2 \mathcal{L}}{\partial \mu_2 \partial \mu_1} = -\lambda \cos(\theta_1 - \mu_1) \cos(\theta_2 - \mu_2)$$

$$\text{so that,} \quad \mathcal{F}_{\mu_2\mu_1} = \mathbb{E}\left[\frac{\partial^2 \mathcal{L}}{\partial \mu_2 \partial \mu_1}\right] = -\lambda \mathbb{E}[\cos(\theta_1 - \mu_1) \cos(\theta_2 - \mu_2)] = -\lambda \frac{c_{\kappa_1\kappa_2}}{c} \tag{4.56}$$

Fisher matrix ($\mathcal{F}_S$) associated with $\kappa_1, \kappa_2, \lambda$: $\mathcal{F}_S$ is a 3×3 symmetric matrix whose elements are the expected values of the second order partial derivatives of $\mathcal{L}$ with respect to κ_1, κ_2, and λ. On differentiating Equation 4.53 with respect to κ_1, κ_2, and λ, we get

$$\frac{\partial \mathcal{L}}{\partial \kappa_1} = \frac{c_{\kappa_1}}{c} - \cos(\theta_1 - \mu_1) \quad \text{and} \quad \frac{\partial \mathcal{L}}{\partial \lambda} = \frac{c_\lambda}{c} - \sin(\theta_1 - \mu_1) \sin(\theta_2 - \mu_2)$$

$$\frac{\partial^2 \mathcal{L}}{\partial \kappa_1^2} = \frac{c c_{\kappa_1\kappa_1} - c_{\kappa_1}^2}{c^2} = \mathcal{F}_{\kappa_1\kappa_1}$$

$$\frac{\partial^2 \mathcal{L}}{\partial \kappa_2^2} = \frac{c c_{\kappa_2\kappa_2} - c_{\kappa_2}^2}{c^2} = \mathcal{F}_{\kappa_2\kappa_2}$$

$$\frac{\partial^2 \mathcal{L}}{\partial \lambda^2} = \frac{c c_{\lambda\lambda} - c_\lambda^2}{c^2} = \mathcal{F}_{\lambda\lambda}$$

$$\frac{\partial^2 \mathcal{L}}{\partial \kappa_1 \partial \kappa_2} = \frac{c c_{\kappa_1\kappa_2} - c_{\kappa_1} c_{\kappa_2}}{c^2} = \mathcal{F}_{\kappa_1\kappa_2}$$

$$\frac{\partial^2 \mathcal{L}}{\partial \lambda \partial \kappa_1} = \frac{c c_{\lambda\kappa_1} - c_\lambda c_{\kappa_1}}{c^2} = \mathcal{F}_{\lambda\kappa_1}$$

$$\frac{\partial^2 \mathcal{L}}{\partial \lambda \partial \kappa_2} = \frac{c c_{\lambda\kappa_2} - c_\lambda c_{\kappa_2}}{c^2} = \mathcal{F}_{\lambda\kappa_2} \tag{4.57}$$

Fisher matrix $\mathcal{F}(\boldsymbol{\Theta})$ associated with the 5-parameter vector $\boldsymbol{\Theta}$: On differentiating Equation 4.54 with respect to κ_1 and computing the expectation of the differential, we get

$$\frac{\partial^2 \mathcal{L}}{\partial \kappa_1 \partial \mu_1} = -\sin(\theta_1 - \mu_1) \quad \text{and} \quad \frac{\partial^2 \mathcal{L}}{\partial \lambda \partial \mu_1} = \cos(\theta_1 - \mu_1) \sin(\theta_2 - \mu_2)$$

$$\text{Hence,} \quad \mathbb{E}\left[\frac{\partial^2 \mathcal{L}}{\partial \kappa_1 \partial \mu_1}\right] = 0 = \mathcal{F}_{\kappa_1\mu_1} \quad \text{and} \quad \mathbb{E}\left[\frac{\partial^2 \mathcal{L}}{\partial \lambda \partial \mu_1}\right] = 0 = \mathcal{F}_{\lambda\mu_1}$$

This allows for the computation of $|\mathcal{F}_1(\boldsymbol{\Theta})|$ as the product of $|\mathcal{F}_A|$ and $|\mathcal{F}_S|$, that is,

$$|\mathcal{F}_1(\boldsymbol{\Theta})| = \begin{vmatrix} \mathcal{F}_{\mu_1\mu_1} & \mathcal{F}_{\mu_1\mu_2} & 0 & 0 & 0 \\ \mathcal{F}_{\mu_2\mu_1} & \mathcal{F}_{\mu_2\mu_2} & 0 & 0 & 0 \\ 0 & 0 & \mathcal{F}_{\kappa_1\kappa_1} & \mathcal{F}_{\kappa_1\kappa_2} & \mathcal{F}_{\kappa_1\lambda} \\ 0 & 0 & \mathcal{F}_{\kappa_2\kappa_1} & \mathcal{F}_{\kappa_2\kappa_2} & \mathcal{F}_{\kappa_2\lambda} \\ 0 & 0 & \mathcal{F}_{\lambda\kappa_1} & \mathcal{F}_{\lambda\kappa_2} & \mathcal{F}_{\lambda\lambda} \end{vmatrix} = |\mathcal{F}_A||\mathcal{F}_S|$$

Then, the Fisher information for some observed data $\mathcal{D} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ is given by

$$|\mathcal{F}(\boldsymbol{\Theta})| = N^5 |\mathcal{F}_1(\boldsymbol{\Theta})| \tag{4.58}$$

as each element in $|\mathcal{F}_1(\boldsymbol{\Theta})|$ is multiplied by the sample size N.

Message length formulation

The message length to encode some observed data $\mathcal{D}$ can now be formulated by substituting the prior density $\Pr(\boldsymbol{\Theta})$ (Equation 4.48), the Fisher information $|\mathcal{F}(\boldsymbol{\Theta})|$ and the negative log-likelihood of the data (Equation 4.47) in the message length expression (Equation 2.8). The MML parameter estimates are the ones that minimize the total message length. As there is no analytical form of the MML estimates, the solution is obtained, as for the maximum likelihood and MAP cases, by using the NLopt optimization library (Johnson, 2014). At each stage of the optimization routine, the Fisher information needs to be calculated. However, this involves the computation of complex entities such as the normalization constant $c(\kappa, \beta)$ and its partial derivatives. The computation of these intricate mathematical forms using numerical methods is discussed next in Section 4.4.4.

4.4.4 Computation of the normalization constant and its derivatives

The computation of the negative log-likelihood and the message length requires the normalization constant and its associated derivatives. In this section, the description of the methods that can be employed to efficiently compute these complex functions is explored. Recall that in Section 4.3.5, in the case of the FB_5 distribution, we showed how to implement these partial derivatives as infinite summations. In the following discussion, we will take the same approach to implement the necessary partial derivatives as summations for the BVM Sine distribution.

Computing $\log c(\boldsymbol{\kappa_1, \kappa_2, \lambda})$ and the logarithm of the partial derivatives: $c_{\boldsymbol{\kappa_1}}, c_{\boldsymbol{\kappa_2}}, c_{\boldsymbol{\kappa_1\kappa_1}}, c_{\boldsymbol{\kappa_2\kappa_2}}$ and $c_{\boldsymbol{\kappa_1\kappa_2}}$

The expressions of $c, c_{\kappa_1}, c_{\kappa_2}, c_{\kappa_1\kappa_1}, c_{\kappa_2\kappa_2}$, and $c_{\kappa_1\kappa_2}$ are related to each other. These expressions are explained by defining the quantity $S_1^{(m,n)}$, a logarithm sum,

$$S_1^{(m,n)} = \log \delta_1 + \log \sum_{j=0}^{\infty} \underbrace{\binom{2j}{j} e^j I_{j+m}(\kappa_1) I_{j+n}(\kappa_2)}_{f_j} \tag{4.59}$$

where $m, n \in \{0, 1, 2\}$, $\delta_1 = 4\pi^2$, and $e = \dfrac{\lambda^2}{4\kappa_1\kappa_2} < 1$ (by definition).

Computation of the series $S_1^{(m,n)}$: We first establish that $f_{j+1} < f_j \, \forall j \geq 0$ and show that $S_1^{(m,n)}$ converges to a finite sum as $j \to \infty$. Consider the logarithm of the ratio of consecutive terms f_j and

f_{j+1} in $S_1^{(m,n)}$

$$\log \frac{f_{j+1}}{f_j} = \log \frac{\binom{2j+2}{j+1}}{\binom{2j}{j}} + \log e + \log \frac{I_{j+m+1}(\kappa_1)}{I_{j+m}(\kappa_1)} + \log \frac{I_{j+n+1}(\kappa_2)}{I_{j+n}(\kappa_2)} \tag{4.60}$$

for $p, v > 0$, $I_{p+v} < I_p$, and the ratio $\frac{I_{p+v}}{I_p} \to 0$ as $p \to \infty$ (Amos, 1974). Further, $e < 1$ implies the above equation is the sum of negative terms. Hence, $\log \frac{f_{j+1}}{f_j} < 0$, which means $f_{j+1} < f_j$. Also,

$$\lim_{j\to\infty} \log \frac{f_{j+1}}{f_j} = \log 4 + \log e + \lim_{j\to\infty} \log \frac{I_{j+m+1}(\kappa_1)}{I_{j+m}(\kappa_1)} + \lim_{j\to\infty} \log \frac{I_{j+n+1}(\kappa_2)}{I_{j+n}(\kappa_2)} = -\infty$$

Hence, as $\lim_{j\to\infty} \frac{f_{j+1}}{f_j} = 0$, $S_1^{(m,n)}$ is a convergent series.

For a practical implementation of the sum, we need to express $S_1^{(m,n)}$ as the modified summation

$$S_1^{(m,n)} = \log \delta_1 + \log f_0 + \log \sum_{j=0}^{\infty} t_j \tag{4.61}$$

where each f_j is divided by the *maximum* term f_0. For each $j > 0, \log f_j$ is calculated using the previous term $\log f_{j-1}$ (Equation 4.60). The new term $t_j = f_j/f_0$ is then computed[5] as $\exp(\log f_j - \log f_0)$. This is because computing the difference with the maximum value and then taking the exponent ensures numerical stability. The summation is terminated when the ratio $\frac{t_j}{\sum_{k=0}^{j} t_k} < \epsilon$ (a small threshold $\sim 10^{-6}$).

- Let $S(c) = \log c(\kappa_1, \kappa_2, \lambda)$: Substituting $m = 0$ and $n = 0$ in Equation 4.59 gives the logarithm of the normalization constant (given in Equation 4.44). Hence, $S(c) = S_1^{(0,0)}$.

- Let the j^{th} term dependent on κ_1 in Equation 4.44 be represented as $g_j(\kappa_1) = I_j/\kappa_1^j$, where I_j implicitly refers to $I_j(\kappa_1)$. Based on the relationship between the Bessel functions I_j, I_{j+1}, and the derivative I_j' in Equation 4.62 (Abramowitz and Stegun, 1965), the expressions for the first and second derivatives of $g_j(\kappa_1)$ (Equation 4.63) are derived as

$$\kappa_1 I_j' = j I_j + \kappa_1 I_{j+1} \tag{4.62}$$

$$g_j'(\kappa_1) = \frac{I_{j+1}}{\kappa_1^j} \quad \text{and} \quad g_j''(\kappa_1) = \frac{I_{j+2}}{\kappa_1^j} + \frac{1}{\kappa_1}.\frac{I_{j+1}}{\kappa_1^j} \tag{4.63}$$

- Let $S(c_{\kappa_1}) = \log c_{\kappa_1}$: Because of the similar forms of $g_j(\kappa_1)$ and $g_j'(\kappa_1)$, the expression for $S(c_{\kappa_1})$ will be similar to that of $S(c)$ with a change in *order* of the Bessel functions from $m = 0$ in Equation 4.59 to $m = 1$. Hence, $S(c_{\kappa_1}) = S_1^{(1,0)}$ and an expression similar to Equation 4.61 can be derived for $S(c_{\kappa_1})$.

- Let $S(c_{\kappa_2}) = \log c_{\kappa_2}$: Similar to the computation of $S(c_{\kappa_1})$ above, if we substitute $m = 0, n = 1$ in Equation 4.61, we obtain the expression for $S(c_{\kappa_2}) = S_1^{(0,1)}$.

- Let $S(c_{\kappa_1\kappa_2}) = \log c_{\kappa_1\kappa_2}$: Similar to the above computations of $S(c_{\kappa_1})$ and $S(c_{\kappa_2})$, if we substitute $m = 1, n = 1$ in Equation 4.61, we obtain the expression for $S(c_{\kappa_1\kappa_2}) = S_1^{(1,1)}$.

[5]Because of the nature of Bessel functions, $\log f_j$ can get very large and can result in overflow when calculating the exponent $\exp(\log f_j)$. However, dividing by f_0 results in $f_j/f_0 < 1$.

- Let $S(c_{\kappa_1\kappa_1}) = \log c_{\kappa_1\kappa_1}$: Substituting $m = 2, n = 0$ in Equation 4.59 gives the logarithm sum $S_1^{(2,0)}$ corresponding to the series with terms $\frac{I_{j+2}}{\kappa_1^j}$. Based on the nature of $g_j''(\kappa_1)$ (Equation 4.63), and noting that $S(c_{\kappa_1}) > S_1^{(2,0)}$ (as $I_{j+1} > I_{j+2}\,\forall\, j \geq 0$), $S(c_{\kappa_1\kappa_1})$ is formulated as

$$S(c_{\kappa_1\kappa_1}) = S(c_{\kappa_1}) + \log\left(\exp(S_1^{(2,0)} - S(c_{\kappa_1})) + \frac{1}{\kappa_1}\right)$$

- Let $S(c_{\kappa_2\kappa_2}) = \log c_{\kappa_2\kappa_2}$: Based on the same reasoning as above, we have

$$S(c_{\kappa_2\kappa_2}) = S(c_{\kappa_2}) + \log\left(\exp(S_1^{(0,2)} - S(c_{\kappa_2})) + \frac{1}{\kappa_2}\right)$$

The logarithm of the partial derivatives: $c_{\boldsymbol{\lambda}}$, $c_{\boldsymbol{\kappa_1\lambda}}$, $c_{\boldsymbol{\kappa_2\lambda}}$, and $c_{\boldsymbol{\lambda\lambda}}$

The expressions of $c_\lambda, c_{\kappa_1\lambda}$, and $c_{\kappa_2\lambda}$ are related and are explained using the logarithm sum $S_2^{(m,n)}$

$$S_2^{(m,n)} = \log\delta_2 + \log\sum_{j=1}^{\infty}\underbrace{\binom{2j}{j} j e^j I_{j+m}(\kappa_1) I_{j+n}(\kappa_2)}_{f_j} \tag{4.64}$$

where $m, n \in \{0, 1\}$, $\delta_2 = \dfrac{8\pi^2}{\lambda}$, and $e = \dfrac{\lambda^2}{4\kappa_1\kappa_2}$. Note that $S_2^{(m,n)}$ is a convergent series (the proof is based on the same reasoning as for $S_1^{(m,n)}$).

Let the j^{th} term dependent on λ, κ_1 in Equation 4.44 be represented as $g_j(\lambda, \kappa_1) = \lambda^{2j}\dfrac{I_j}{\kappa_1^j}$. Its partial derivatives are given below. These derivatives are the terms in the series $S_2^{(m,n)}$ (after factoring out the common elements as δ_2).

$$\frac{\partial g_j}{\partial\lambda} = 2j\lambda^{2j-1}\frac{I_j}{\kappa_1^j} \quad \text{and} \quad \frac{\partial^2 g_j}{\partial\kappa_1\partial\lambda} = 2j\lambda^{2j-1}\frac{I_{j+1}}{\kappa_1^j}$$

- Let $S(c_\lambda) = \log c_\lambda$: this is obtained by substituting $m = 0$ and $n = 0$ in Equation 4.64. Hence, $S(c_\lambda) = S_2^{(0,0)}$.
- Similarly, $S(c_{\kappa_1\lambda}) = \log c_{\kappa_1\lambda} = S_2^{(1,0)}$ and $S(c_{\kappa_2\lambda}) = \log c_{\kappa_2\lambda} = S_2^{(0,1)}$.
- The expression to compute $S(c_{\lambda\lambda}) = \log c_{\lambda\lambda}$ is given by

$$S(c_{\lambda\lambda}) = \log\left(\frac{\delta_2}{\lambda}\right) + \log\sum_{j=1}^{\infty}\underbrace{\binom{2j}{j} j(2j-1) e^j I_j(\kappa_1) I_j(\kappa_2)}_{f_j}$$

The practical implementation of $S_2^{(m,n)}$ and $S(c_{\lambda\lambda})$ is similar to that of $S_1^{(m,n)}$ given by Equation 4.61. However, in these cases, the expressions of f_j and consequently t_j, are modified depending on their specific forms. Also, the series begin from $j = 1$ and, hence, the respective maximum terms will correspond to f_1.

4.4.5 Evaluation of the MML estimates

For a given BVM Sine distribution characterized by concentration parameters κ_1, κ_2 and correlation coefficient ρ, a random sample of size N is generated using the method proposed by Mardia et al.

(2007). The angular parameters of the true distribution are set to $\{\mu_1, \mu_2\} = \pi/2$. The scale parameters κ_1, κ_2, and ρ are varied to obtain different BVM Sine distributions and corresponding random samples. The parameters are estimated using the sampled data and the different estimation methods. The procedure is repeated 1000 times for each combination of N, κ_1, κ_2, and ρ.

Methods of comparison

The comparison methodology is similar to the one adopted in the case of the different estimators of the FB_5 distribution (Section 4.3.6). For the BVM Sine distribution, the ML, MAP, and MML estimates of the data generated in these simulations are compared with each other.

The results include the three versions of MAP estimates resulting from the three forms of the posterior distributions (Equations 4.48-4.50): *MAP1* corresponds to the posterior with parameterization $\mathbf{\Theta} = (\mu_1, \mu_2, \kappa_1, \kappa_2, \lambda)$, *MAP2* corresponds to the posterior with parameterization $\mathbf{\Theta'} = (\mu_1, \mu_2, \kappa_1, \kappa_2, \rho)$, and *MAP3* corresponds to the posterior with parameterization $\mathbf{\Theta''} = (z_1, z_2, z_3, z_4, z_5)$. As noted in Section 4.4.2, the MAP3 estimator will be the same as the ML estimator due to the Rosenblatt (1952) transformation of $\mathbf{\Theta}$ to $\mathbf{\Theta''}$.

In order to compare the various estimators, we use again the mean squared error (MSE) and Kullback-Leibler (KL) distance (Section 2.5). The estimates are also compared using statistical hypothesis testing. The empirical testing is as discussed in Section 4.3.6. For a parameter vector $\mathbf{\Theta}$ characterizing a true BVM Sine distribution, and its estimate $\widehat{\mathbf{\Theta}}$, we analyze the MSE and KL distance of $\widehat{\mathbf{\Theta}}$ with respect to the true parameter vector $\mathbf{\Theta}$.

The analytical form of the KL distance between two BVM distributions is derived in Appendix C.1. As in the case of the FB_5 experimental analyses (Section 4.3.6), we analyze the percentage of times (*wins*) the KL distance of a particular estimator is smaller than that of others. When the KL distance of different estimates is compared, because of three different versions of MAP estimation, three separate frequency plots are presented. corresponding to the MAP1, MAP2, and MAP3 estimators.

With respect to statistical hypothesis testing, the likelihood ratio test statistic is asymptotically approximated as an χ^2 distribution with five degrees of freedom (see Sections 2.3.1 and 4.3.6). For the various parameter estimates compared here, it is expected that at especially large sample sizes, the estimates are close to the ML estimate. In other words, the empirically determined test statistic is expected to be lower than the critical value $\tau = 13.086$, corresponding to a p-value greater than 0.01.

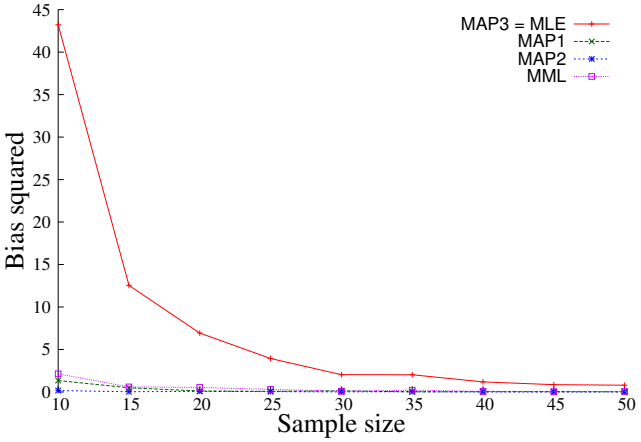
Empirical analyses

As per the experimental setup, we present the results for when the original distribution from which the data is sampled has $\kappa_1 = 1$ and $\kappa_2 = 10$. The correlation coefficient ρ is varied between 0 and 1, so that we obtain different values for the correlation parameter λ (Equation 4.46). We discuss the results for varying values of sample sizes N, and $\rho = 0.1, 0.5, 0.9$, corresponding to a low, moderate, and high correlation, respectively.

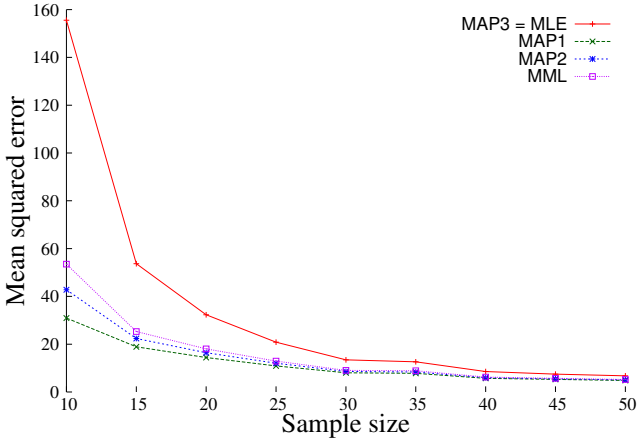
For $\rho = 0.1$: The results are presented in Figure 4.12. Compared to the ML estimators, the MAP and MML estimators result in lower bias and MSE for all values of N. Both the bias and MSE continue to decrease as the sample size increases, as the estimation improves with more evidence for all methods. When compared with MAP1 and MAP2, the MML estimators have greater bias and greater MSE. As with the FB_5 distribution, we observe that that MAP1 and MAP2 result in different estimators, and therefore, result in different bias and MSE values.

The KL distance with respect to MAP1 is in favour of the MAP1 estimators. The MAP1 estimates result in lower KL distance as compared to the other estimators almost 50% of the 1000 simulations for each N (Figure 4.12c). However, the MML estimators win when the MAP2 and MAP3 versions are used. When MAP3 is used, the MML estimators have a smaller KL distance in close to 70% of the simulations (Figure 4.12e). Further analysis using statistical hypothesis testing illustrates that the null hypotheses corresponding to the MAP and MML estimators are accepted (p-values greater

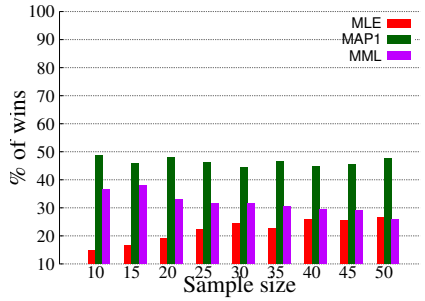
than 0.01 in Figure 4.12f). at the 1% significance level.



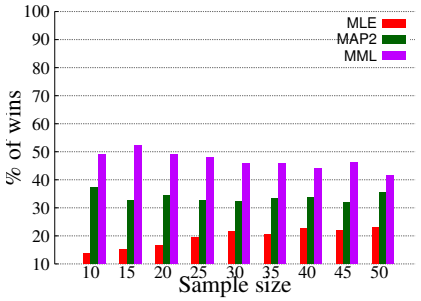
(a) Bias-squared



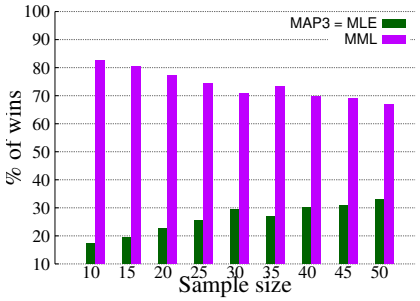
(b) Mean squared error



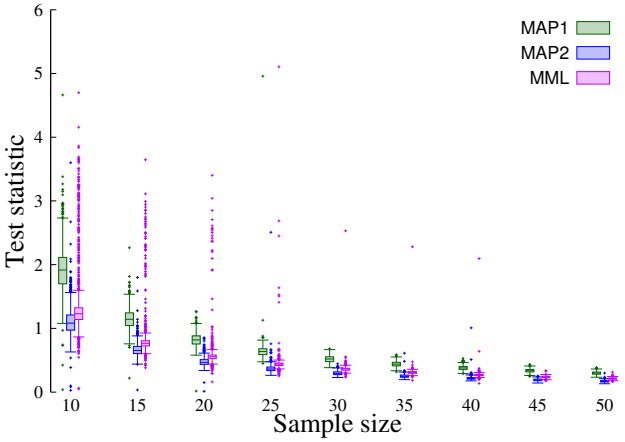
(c) KL distance (MAP version 1)



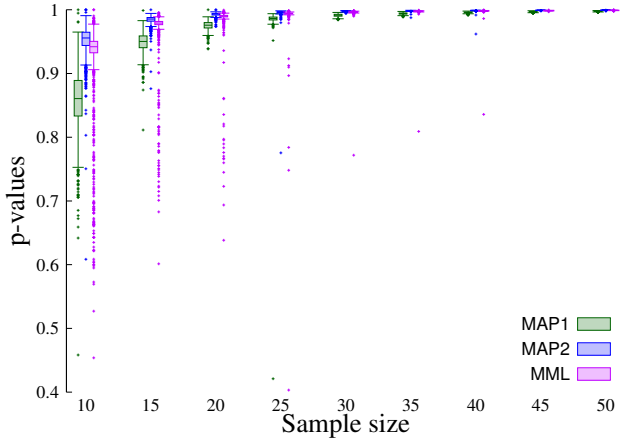
(d) KL distance (MAP version 2)



(e) KL distance (MAP version 3)



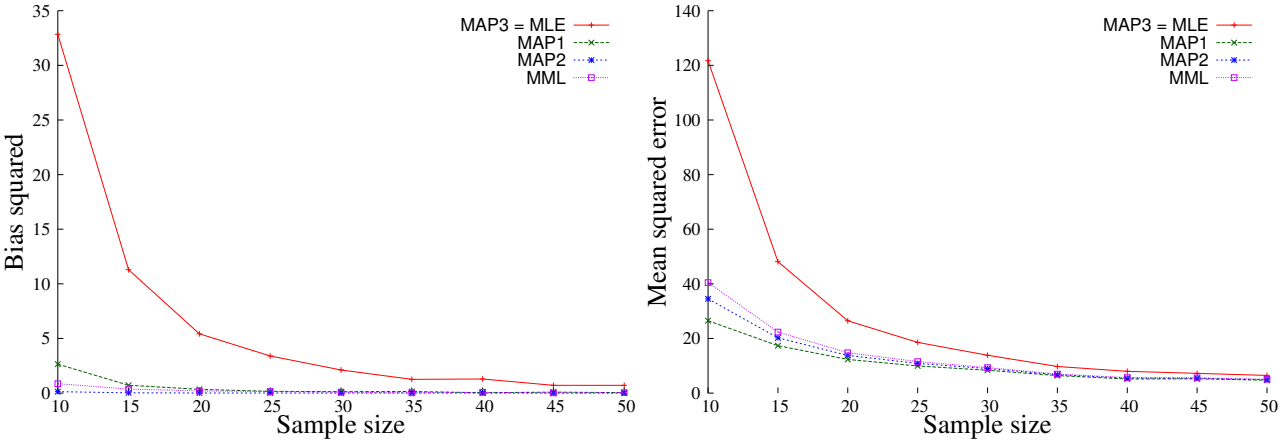
(f) Variation of test statistics



(g) Variation of p-values

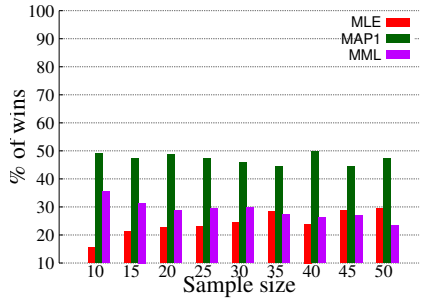
Figure 4.12: Comparison of the parameter estimates when $\kappa_1 = 1, \kappa_2 = 10, \rho = 0.1$.

For $\rho = 0.5$: Similar to when $\rho = 0.1$, we observe that the bias and MSE of the MAP and MML estimators are lower than the ML estimators for different values of N. In contrast to $\rho = 0.1$, the bias of the MML estimator is lower than the MAP1 estimator but higher than the MAP2 estimator (Figure 4.13a). As with the previous case, MAP-based estimation result in different estimators. Further analysis of the estimators using KL distance and statistical hypothesis testing follow the same pattern as when $\rho = 0.1$.

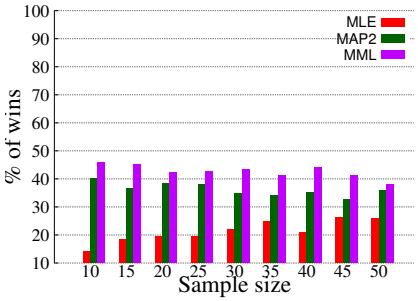


(a) Bias-squared

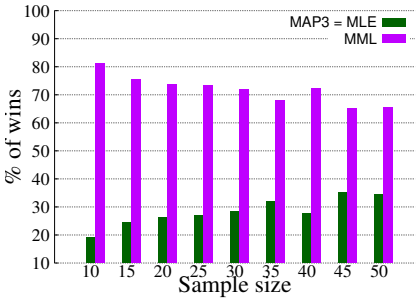
(b) Mean squared error



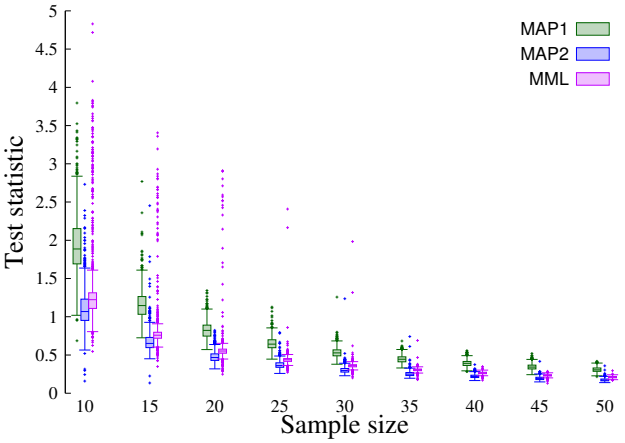
(c) KL distance (MAP version 1)



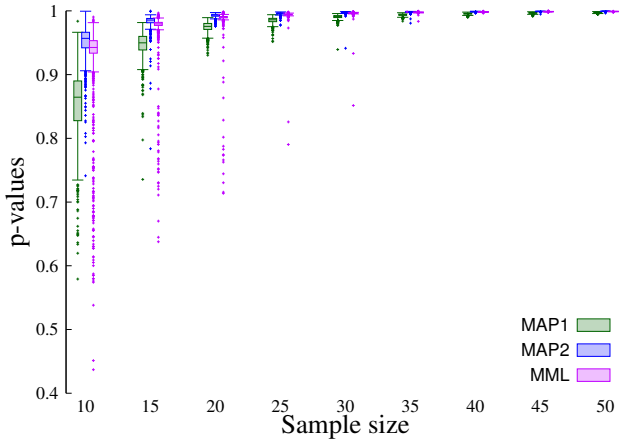
(d) KL distance (MAP version 2)



(e) KL distance (MAP version 3)



(f) Variation of test statistics

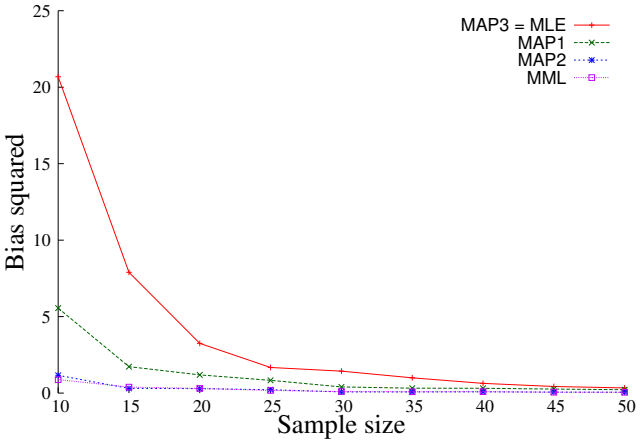


(g) Variation of p-values

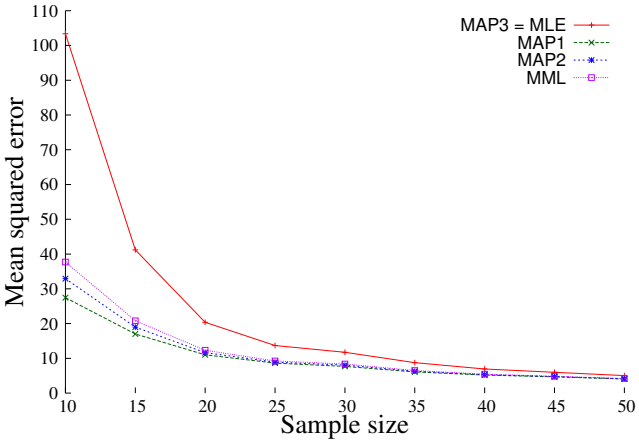
Figure 4.13: Comparison of the parameter estimates when $\kappa_1 = 1, \kappa_2 = 10, \rho = 0.5$.

For $\rho = 0.9$: The results are presented in Figure 4.14. As with the previous two cases, we observe that the ML estimators have the greatest bias and MSE for all values of N. The bias of the MML estimators is lower than all the MAP estimators. However, the MSE of the MML estimators is greater compared to the MAP1 or MAP2 estimators. Contrary to the previous two cases, we observe that the frequency of wins of KL distance for the MML estimators is lower when compared to MAP2 estimation (Figure 4.14e). Further, the results following the statistical hypothesis testing follow the same trend as the previous two cases. As the same size increases, the different estimators converge to the ML estimators as seen from the high p-values (Figure 4.14g).

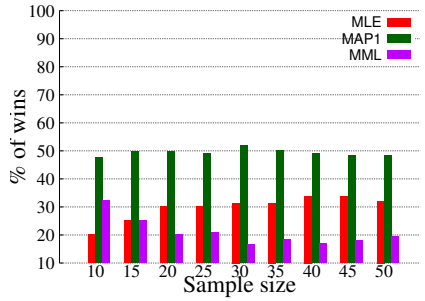
The empirical analyses of the controlled experiments discussed above indicate that the ML estimators are biased estimators. The same was observed with the vMF and FB_5 directional distributions.



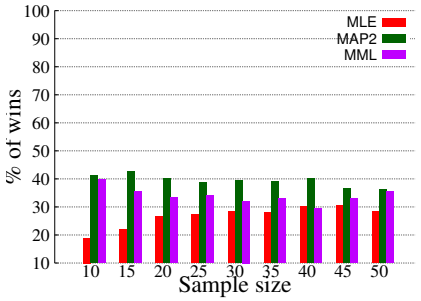
(a) Bias-squared



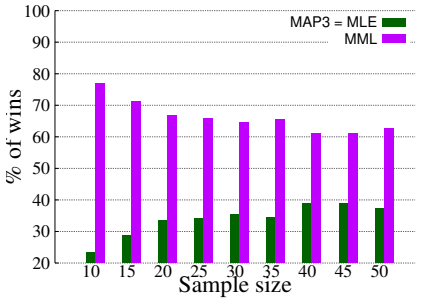
(b) Mean squared error



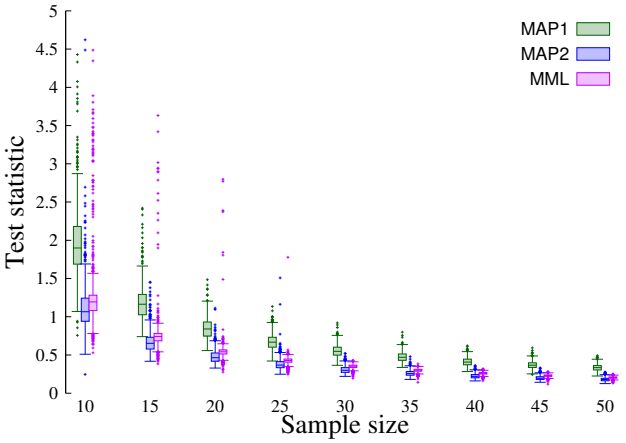
(c) KL distance (MAP version 1)



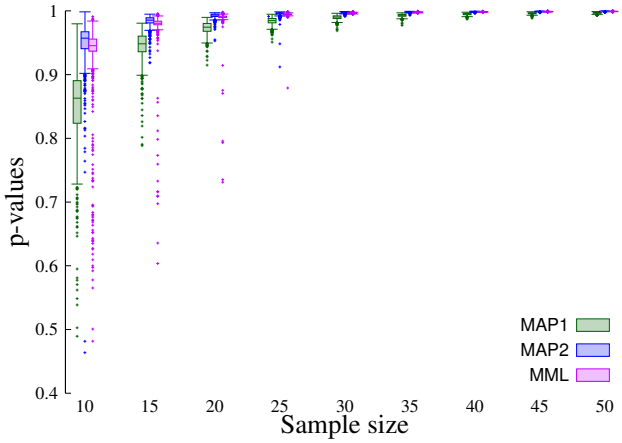
(d) KL distance (MAP version 2)



(e) KL distance (MAP version 3)



(f) Variation of test statistics



(g) Variation of p-values

Figure 4.14: Comparison of the parameter estimates when $\kappa_1 = 1, \kappa_2 = 10, \rho = 0.9$.

Also, we observe that the MAP estimation method result in different estimators depending on how the distribution is parameterized. As with the FB_5 distribution, the MAP estimators are shown to be not invariant. In this context, the MML estimators are empirically demonstrated to have lower bias than the traditional ML estimators and are invariant to alternative parameterizations.

4.5 Summary

This chapter presented a study of three commonly used directional probability distributions, namely, the multivariate von Mises-Fisher, the three-dimesional Kent, and the bivariate von Mises. In the case

of a vMF distribution, the ML estimate of the concentration parameter is approximated in several ways, as discussed in Section 4.2.1, and shown to result in a biased estimator. This is empirically demonstrated in Section 4.2.3. In comparison, the derived MML estimators are shown to have lower bias and mean squared error. They also fare better when evaluated using an objective metric such as the Kullback-Leibler distance. The MML estimates, therefore, serve as improvements to the traditionally used ML estimates. We published our MML-based inference of vMF distributions in Kasarapu and Allison (2015) .

The vMF distribution is widely preferred due to its simplicity and the relative ease of parameter estimation. The Kent distribution, on the other hand, is a generalization of the vMF distribution. Although it is more suitable for modelling directional data, its use is limited due to the complex mathematical expressions arising from formulating the negative log-likelihood function. In this chapter, these difficulties are addressed and a better estimator, in terms of the MML estimator is derived. The MML estimation involves computational challenges, more than those associated with the traditional moment, ML or MAP-based approaches. However, we resolved these challenges, and a superior estimator is provided. The estimator has lower bias and mean squared error compared to the moment and ML estimators. It is robust to parameterization, unlike the MAP estimator, and performs better in terms of the Kullback-Leibler distance (Kasarapu, 2015).

The above observations are in agreement with the BVM distribution as well. Even in this case, the MML estimators are shown to be superior compared to the traditional ML and MAP estimators. Therefore, the MML estimators provide a strong case to be used in modelling tasks involving directional data. We use the MML estimators in Chapter 6, where we study the inference of mixtures of directional probability distributions.

Chapter 5

Mixture modelling

5.1 Introduction

The previous chapters focused on the parameter estimation of single component probability distributions. In most real-world settings, however, the true data generating model is usually unknown and is potentially multimodal. To describe such multimodal data, mixtures of component probability distributions must be considered. Mixture models are common tools in statistical pattern recognition tasks (McLachlan and Basford, 1988). They offer a mathematical basis to explain data in several fields as diverse as astronomy, biology, ecology, engineering, and economics, amongst many others (McLachlan and Peel, 2000).

A mixture model is composed of component probabilistic models; a component may correspond to a subtype, kind, species, or sub-population of the observed data. These models aid in the identification of hidden patterns in the data through sound probabilistic formalisms. Mixture models have been extensively employed in machine learning tasks such as classification and unsupervised learning (Titterington et al., 1985; McLachlan and Peel, 2000; Jain et al., 2000).

Much of the literature on mixture modelling concerns the theory and application of mixtures of Gaussian distributions (McLachlan and Peel, 2000; Jain and Dubes, 1988). The importance of Gaussian mixtures in practical applications is well established, and their use in several research disciplines has been partly motivated by their computational tractability (McLachlan and Peel, 2000). Thus, we consider the Gaussian mixtures elaborately in this chapter and discuss mixture modelling using other probability distributions of interest in Chapter 6.

The general problem of mixture modelling involves determining the number of components, the probabilities of those components, and estimating the parameters of each of the component distributions. The simultaneous inference of the mixture parameters and the number of components involves the difficult problem of balancing the trade-off between two conflicting objectives: low *model complexity* as determined by the number of components and their respective parameters, versus *goodness-of-fit* to the observed data.

The generalized methodology for mixture modelling relies on the following elements:

1. An *estimator* of the parameters of each component of a mixture,

2. An *objective criterion*, that is a cost or a score, that can be used to compare two hypothetical mixtures and decide which is better, and

3. A *search strategy* for the best number of components.

Estimation of component parameters: For a *given* number of components, the conventional method of estimating the parameters of a mixture relies on the Expectation-Maximization (EM) algorithm (Dempster et al., 1977), where the parameters are iteratively updated using the estimates that optimize an objective function. Traditional optimization is based on maximizing the log-likelihood of the

data (ML-based) or maximizing the posterior probability density (in the case of MAP-based estimation). In contrast, as per the MML framework, the parameters are updated by their MML estimates that minimize the total message length. The standard EM algorithm is a local optimization method that is sensitive to initialization and, in certain cases, may converge to the boundary of the parameter space (Krishnan and McLachlan, 1997; Figueiredo and Jain, 2002).

Scoring function: In order to compare competing mixture models, there have been numerous scoring functions proposed in the literature that aim to evaluate a given mixture model. A review of these methods is presented in McLachlan and Peel (2000). These methods are motivated by a common theme to balance the complexity of a chosen mixture model and its goodness-of-fit to the data. The scoring functions that quantify the model complexity based on just the number of components are the Akaike Information Criterion (AIC) (Akaike, 1974), the Bayesian Information Criterion (BIC) (Schwarz, 1978; Rissanen, 1978), and the Integrated Completed Likelihood (ICL) criterion (Biernacki et al., 2000). It is to be noted that AIC and BIC are shown to be approximations of the general MML framework (Figueiredo and Jain, 2002).

The information-theoretic criteria that account for not just the number of components but also the components' parameters are the *information complexity criterion* (Bozdogan, 1993), the *Laplace empirical criterion* (LEC) (Roberts et al., 1998), and the *approximated MML criterion* (Oliver et al., 1996; Figueiredo and Jain, 2002). Further, these criteria are derived using an MML interpretation. However, as detailed in Section 5.3, these criteria are simplified versions of the generic MML framework, and are incomplete in objectively addressing the trade-off associated with selecting a suitable mixture model.

Previous attempts to infer Gaussian mixtures based on these MML-like criteria have been undertaken by using simplifying assumptions, such as the covariance matrices being diagonal (Oliver et al., 1996), or by coarsely approximating the probabilities of mixture parameters (Roberts et al., 1998; Figueiredo and Jain, 2002). In this work, these drawbacks are rectified by proposing a comprehensive MML formulation with no assumptions on the nature of the component distribution.

Search strategy: To determine the optimal number of mixture components using the aforementioned criteria (Akaike, 1974; Schwarz, 1978; Oliver et al., 1996; Roberts et al., 1998; Biernacki et al., 2000), mixtures are inferred for a varying number of components using the EM algorithm, and the mixture that has the smallest score is treated as the optimal one. As the EM only guarantees convergence to a local optimum, a few trials are conducted with different starting points in an effort to minimize the possibility of getting trapped in a local optimum (Krishnan and McLachlan, 1997; McLachlan and Peel, 2000).

In order to rectify the issues arising from the use of the EM algorithm, methods based on iteratively splitting and merging constituent mixture components have been proposed so as to enable the intermediate mixtures to escape from the local optima. The most notable amongst these are the *split-merge* based EM (SMEM) method proposed by Ueda et al. (2000) and component-deletion based unsupervised learning approach proposed by Figueiredo and Jain (2002). However, as explained in Section 5.3, these strategies have limited utility.

In this work, we propose a search method that selectively *splits, deletes*, or *merges* components depending on improvements to the MML scoring criterion. The operations, combined with the EM steps, result in a sensible redistribution of data between the mixture components. As an example, a component may be split into two children, and at a later stage, one of the children may be merged with another component. Our proposed method starts with a one-component mixture and alters the number of components in the subsequent iterations. This avoids the overhead of dealing with a large number of components unless required.

The rest of the chapter is organized as follows: Section 5.2 outlines the procedure to estimate the mixture parameters using the EM algorithm. Section 5.3 details the different strategies that are traditionally employed in determining the optimal number of mixture components. These methods

provide varied formulations to assess the mixture components and their ability to explain the data. Methods using the MML criterion have been proved to be effective in achieving a reliable balance between these conflicting aims (Wallace and Boulton, 1968; Oliver et al., 1996; Roberts et al., 1998; Figueiredo and Jain, 2002). This has motivated the use of MML-based inference in mixture modelling of probability distributions in this chapter. To this end, we propose a general purpose mixture modelling approach in Section 5.4. The method has been employed in the context of mixture modelling using multivariate Gaussian distributions. Section 5.5 compares the proposed methodology against the widely used method of Figueiredo and Jain (2002), and presents an analysis of the performance of our search method when modelling simulated and real-world data using multivariate Gaussian mixtures.

5.2 Parameter estimation of mixtures

Mixture modelling involves representing an observed distribution of data as a weighted sum of individual probability density functions. Specifically, the problem considered here is to model the mixture distribution $\mathcal{M}$ as defined below

$$f(\mathbf{x};\mathbf{\Phi}) = \sum_{j=1}^{K} w_j f_j(\mathbf{x};\mathbf{\Theta}_j) \tag{5.1}$$

where $\mathbf{x}$ is a d-dimensional datum, K is the number of mixture components, w_j and $f_j(\mathbf{x};\mathbf{\Theta}_j)$ are the weight and probability density of the j^{th} component respectively; the weights are positive and sum to one, and $\mathbf{\Phi}$ denotes the entire set of mixture parameters $\{w_1,\cdots,w_K,\mathbf{\Theta}_1,\cdots,\mathbf{\Theta}_K\}$.

The standard EM algorithm (Dempster et al., 1977) to estimate $\mathbf{\Phi}$ involves minimizing the objective function depending on the type of the estimation method. In the current context, the EM algorithm is explained to obtain the ML and MML estimates of the mixture parameters.

5.2.1 EM algorithm for ML estimation of mixture parameters

The EM algorithm to obtain the ML estimates is based on minimizing the negative log-likelihood function of the data. For some observed data $\mathcal{D} = \{\mathbf{x}_1,\ldots,\mathbf{x}_N\}$ and a mixture $\mathcal{M}$, the negative log-likelihood using the mixture distribution is as follows

$$\mathcal{L}(\mathcal{D}|\mathbf{\Phi}) = -\sum_{i=1}^{N}\log\sum_{j=1}^{K} w_j f_j(\mathbf{x}_i;\mathbf{\Theta}_j) \tag{5.2}$$

For a fixed K, the ML estimates are then given as $\widehat{\mathbf{\Phi}}_{\text{ML}} = \arg\min_{\mathbf{\Phi}} \mathcal{L}(\mathcal{D}|\mathbf{\Phi})$. Because of the absence of a closed form solution for $\widehat{\mathbf{\Phi}}_{\text{ML}}$, a gradient descent method is employed where the estimates are iteratively updated until convergence to some local minimum is achieved (Dempster et al., 1977; McLachlan and Basford, 1988; Xu and Jordan, 1996; Krishnan and McLachlan, 1997; McLachlan and Peel, 2000). The EM method consists of two steps:

- *E-step*: In the EM framework for mixture modelling, it is assumed that each datum $\mathbf{x}_i$ originates from one of the mixture components. However, as the component of origin is unknown, the membership of $\mathbf{x}_i$ is computed as its conditional expectation given the current value of the mixture parameters. These partial memberships of the data points to each of the components are defined using the *responsibility matrix*

$$r_{ij} = \frac{w_j f(\mathbf{x}_i;\mathbf{\Theta}_j)}{\sum_{k=1}^{K} w_k f(\mathbf{x}_i;\Theta_k)}, \quad \forall\, 1\le i\le N, 1\le j\le K \tag{5.3}$$

where r_{ij} denotes the conditional probability of a datum $\mathbf{x}_i$ belonging to the j^{th} component. The effective membership associated with the component is given by

$$n_j = \sum_{i=1}^{N} r_{ij} \quad \text{and} \quad \sum_{j=1}^{K} n_j = N \tag{5.4}$$

- *M-step*: Assuming $\mathbf{\Phi}(t)$ to be the estimates at some iteration t, the expectation of the negative log-likelihood using $\mathbf{\Phi}(t)$ and the partial memberships is then *minimized* which is tantamount to computing $\mathbf{\Phi}(t+1)$, the updated maximum likelihood estimates for the next iteration $(t+1)$. The weights are updated as $w_j(t+1) = \dfrac{n_j(t)}{N}$.

The above sequence of steps are repeated until a certain convergence criterion is satisfied. At some intermediate iteration t, the mixture parameters are updated using the corresponding ML estimates.

In the case of mixtures of *Gaussian* distributions, where the component parameters are $\mathbf{\Theta}_j = (\boldsymbol{\mu}_j, \mathbf{C}_j)$, the ML updates of the mean $(\boldsymbol{\mu}_j)$ and covariance matrix $(\mathbf{C}_j)$ are as follows (see Section 3.5.1)

$$\widehat{\boldsymbol{\mu}}_j(t+1) = \frac{1}{n_j(t)} \sum_{i=1}^{N} r_{ij}(t)\mathbf{x}_i \quad \text{and} \quad \widehat{\mathbf{C}}_j(t+1) = \frac{1}{n_j(t)} \sum_{i=1}^{N} r_{ij}(t) \left(\mathbf{x}_i - \widehat{\boldsymbol{\mu}}_j(t+1)\right) \left(\mathbf{x}_i - \widehat{\boldsymbol{\mu}}_j(t+1)\right)^T$$

5.2.2 EM algorithm for MML estimation of mixture parameters

The objective function that needs to be minimized using the EM algorithm corresponds to the total message length expression. To describe the EM algorithm in the context of MML framework, it is required to formulate the two-part message length corresponding to the mixture distribution and the data using the mixture. The computation of the total message length is explained below.

Encoding a mixture model using MML

The encoding of a mixture model using the MML framework requires the encoding of (1) the model parameters and then (2) the data using the parameters. The statement costs for encoding the mixture model and the data can be decomposed as follows (Wallace, 2005):

1. Encoding the *number of components* K: In order to encode the message losslessly, the information pertaining to the number of components should be communicated. In the absence of background knowledge, one would like to model the prior belief in such a way that the probability decreases for an increasing number of components. If $h(K) \propto 2^{-K}$, then $I(K) = K \log 2 + \text{constant}$.

   This prior reflects that there is a difference of one bit in encoding the *numbers* K and $K+1$. Alternatively, one could assume a uniform prior over K within some predefined range. The chosen prior has little effect as its contribution is minimal when compared to the magnitude of the total message length (Wallace, 2005).

2. Encoding the *weights* $w_1, \cdots, w_K$: The component weights are treated as parameters of a multinomial distribution with sample sizes $n_j, \forall\ 1 \leq j \leq K$, and N being the total sample size (Equation 5.4). The message length to encode all the weights is then given by the expression (Boulton and Wallace, 1969)

$$I(\mathbf{w}) = \frac{(K-1)}{2} \log N - \frac{1}{2} \sum_{j=1}^{K} \log w_j - (K-1)!$$

3. Encoding each of the *component parameters* $\mathbf{\Theta}_j$: This has been previously discussed in Section 2.4.2, and is given by $I(\mathbf{\Theta}_j) = -\log \dfrac{h(\mathbf{\Theta}_j)}{\sqrt{|\mathcal{F}(\mathbf{\Theta}_j)|}}$

4. Encoding the *data*: Each datum $\mathbf{x}_i$ can be stated to a finite precision ϵ.[1] If the precision to which each element of a d-dimensional vector can be stated is ϵ, then the *probability* of a datum $\mathbf{x}_i \in \mathbb{R}^d$ is given as $\Pr(\mathbf{x}_i) = \epsilon^d f(\mathbf{x}_i; \mathbf{\Phi})$ where $f(\mathbf{x}_i; \mathbf{\Phi})$ is the *probability density* given by Equation 5.1. Hence, the length of its encoding $\mathbf{x}_i$ is given by

$$I(\mathbf{x}_i) = -\log \Pr(\mathbf{x}_i) = -d \log \epsilon - \log \sum_{j=1}^{K} w_j f_j(\mathbf{x}_i; \mathbf{\Theta}_j)$$

The entire data $\mathcal{D}$ can now be encoded as:

$$I(\mathcal{D}|\mathbf{\Phi}) = -Nd \log \epsilon - \sum_{i=1}^{N} \log \sum_{j=1}^{K} w_j f_j(\mathbf{x}_i; \mathbf{\Theta}_j)$$

Thus, the total two-part message length of a K-component mixture is given by

$$I(\mathbf{\Phi}, \mathcal{D}) = I(\mathbf{\Phi}) + I(\mathcal{D}|\mathbf{\Phi})$$

$$\text{where } I(\mathbf{\Phi}) = I(K) + I(\mathbf{w}) + \left(-\sum_{j=1}^{K} \log\, h(\mathbf{\Theta}_j) + \frac{1}{2} \sum_{j=1}^{K} \log |\mathcal{F}(\mathbf{\Theta}_j)| \right) + \text{constant} \tag{5.5}$$

Note that the *constant* term includes the lattice quantization constant (resulting from stating all the model parameters) in a p-dimensional space, where p is equal to the number of free parameters in the mixture model (see Section 2.4.2).

Estimating the mixture parameters

The MML-based estimates of the parameters of the mixture model are those that *minimize* Equation 5.5. To achieve this, we use the EM algorithm (Section 5.2.1), where the parameters are iteratively updated using their respective *MML* estimates. The component weights are obtained by differentiating Equation 5.5 with respect to w_j under the constraint $\sum_{j=1}^{K} w_j = 1$. The derivation of the MML updates of the weights is shown in Appendix D.1 and are given as

$$w_j(t+1) = \frac{n_j(t) + \frac{1}{2}}{N + \frac{K}{2}} \tag{5.6}$$

The parameters of the j^{th} component are updated using the partial memberships, $r_{ij}(t)$ and $n_j(t)$ given by Equations 5.3 and 5.4 at an intermediate iteration t of the EM algorithm. In the case of Gaussian mixtures, the MML updates of the mean and covariance matrix of the j^{th} Gaussian component, as described in Section 3.5.2, are given by

$$\widehat{\boldsymbol{\mu}}_j(t+1) = \frac{1}{n_j(t)} \sum_{i=1}^{N} r_{ij}(t) \mathbf{x}_i \quad \text{and} \quad \widehat{\mathbf{C}}_j(t+1) = \frac{1}{n_j(t) - 1} \sum_{i=1}^{N} r_{ij}(t) \left(\mathbf{x}_i - \widehat{\boldsymbol{\mu}}_j(t+1)\right) \left(\mathbf{x}_i - \widehat{\boldsymbol{\mu}}_j(t+1)\right)^T \tag{5.7}$$

The EM algorithm terminates when the change in the total message length (improvement rate) between successive iterations falls below some predefined threshold. The difference between the two variants of standard EM algorithms discussed above is firstly, the objective function that is being optimized. In Section 5.2.1, the negative log-likelihood function is *minimized*. This corresponds to the $I(\mathcal{D}|\mathbf{\Phi})$ term in Section 5.2.2. The total message length expression (Equation 5.5) includes additional terms that correspond to the cost associated with stating the mixture parameters. Secondly,

[1] Note that ϵ is a constant value and has no effect on the overall inference process. It is used in order to maintain the theoretical validity when making the distinction between *probability* and *probability density*.

in the M-step, in Section 5.2.1, the components are updated using their ML estimates, whereas in Section 5.2.2 the components are updated using their MML estimates.

Limitations of the EM algorithms: The standard EM algorithms outlined above can be used only when the number of mixture components K is fixed or known *a priori*. Even when the number of components are fixed, the EM algorithm has potential pitfalls as the method is sensitive to the initialization conditions. To overcome this, some reasonable start state for the EM may be determined by initially clustering the data (Krishnan and McLachlan, 1997; McLachlan and Peel, 2000). Another strategy is to run the EM a few times and choose the best amongst all the trials. Figueiredo and Jain (2002) point out that, in the case of Gaussian mixture modelling, the EM algorithm can converge to the boundary of the parameter space when the corresponding covariance matrix is nearly singular or when there are few initial members assigned to that component.

5.3 Existing methods to infer the number of components

In a mixture modelling problem, for a completely unsupervised setting, it is required to determine the number of mixture components. However, inferring the "right" number of mixture components for unlabelled data is a difficult problem (McLachlan and Peel, 2000). There have been numerous approaches proposed that attempt to tackle this problem (Akaike, 1974; Schwarz, 1978; Rissanen, 1978; Bozdogan, 1993; Oliver et al., 1996; Roberts et al., 1998; Biernacki et al., 2000; Figueiredo and Jain, 2002; Baudry and Celeux, 2015).

Given some observed data, there are infinitely many mixtures that one can fit to the data. Any method that aims to selectively determine the optimal number of components should be able to factor the cost associated with the mixture parameters. To this end, several methods based on information theory have been proposed where there is some form of penalty associated with choosing a certain parameter value (Wallace and Boulton, 1968; Akaike, 1974; Schwarz, 1978; Wallace and Freeman, 1987; Rissanen, 1989). Some of these methods are reviewed here followed by a discussion of the widely used method of Figueiredo and Jain (2002).

5.3.1 Akaike and Bayesian information criteria

The Akaike Information Criterion (AIC) (Akaike, 1974) and Bayesian Information Criterion (BIC) serve as scoring functions to evaluate a mixture and its corresponding goodness-of-fit to the data.

Formulation of the scoring functions: As discussed in Section 2.3.2, the AIC (Akaike, 1974) in its simplest form adds the *number* of free parameters p to the negative log-likelihood expression (Equation 2.7). Some variants of AIC have been suggested (Bozdogan, 1983; Burnham and Anderson, 2002). However, in each of these variants, all the free parameters have the same penalty constants.

Similar to AIC, BIC (Schwarz, 1978) adds a constant multiple of $\log \sqrt{N}$ (N being the sample size), for each free parameter in the model (Equation 2.6). Rissanen (1978) proposed the minimum description length (MDL) criterion which formally coincides with BIC (Oliver et al., 1996; Figueiredo and Jain, 2002). The criteria can be interpreted under the MML framework, where the first part of the message is a constant multiplied by the number of free parameters. AIC and BIC formulations can be obtained as approximations to the two-part MML formulation governed by Equation 2.8 (Figueiredo and Jain, 2002).

Search method: To determine the optimal number of mixture components K, the AIC or BIC scores are computed for mixtures with varying values of K. The mixture model with the smallest score is selected as per these criteria.

Limitations: The formulations of the criteria suggest that the parameter cost associated with adopting a model depends only on the number of free parameters and *not* on the parameter values themselves. In other words, these criteria consider all models of a particular type (of probability distribution) to have the same statement cost associated with the parameters. For example, a generic d-dimensional Gaussian distribution has $p = d(d+3)/2$ free parameters. All such distributions will have the same parameter costs regardless of what precise values the means and covariance matrices take.

Furthermore, a d-variate Gaussian mixture with K number of components has $p = \frac{Kd(d+3)}{2} + (K-1)$ free parameters. All mixtures with a set number of components have the same cost associated with their parameters using these criteria. The mixture complexity is therefore treated as independent of the constituent mixture parameters. In contrast, the MML formulation incorporates the statement cost of losslessly encoding mixture parameters by calculating their relevant probabilities as discussed in Section 5.2.2. It has been argued that for tasks such as mixture modelling, where the number of free parameters potentially grows in proportion to the data, MML is known in theory to give consistent results as compared to AIC and BIC (Wallace, 1986; Wallace and Dowe, 1999).

5.3.2 Approximate MML criterion (Oliver et al., 1996)

Formulation of the scoring function: For modelling Gaussian mixtures, Oliver et al. (1996) proposed a MML-based scoring function akin to the one shown in Equation 5.5. Their proposed criterion is however, only for Gaussian mixtures.

Search method: The search method adopted by Oliver et al. (1996) to infer the number of components is to run the EM algorithm several times, and choose the K for which the message length is the least out of the several EM trials. For each K, the standard EM algorithm (Section 5.2.1) was used to attain local convergence.

Limitations: Oliver et al. (1996) developed their method only for mixtures of Gaussian distributions with diagonal covariance matrices and fail to provide a general method dealing with full covariance matrices. Their method cannot be generalized to different kinds of probability distributions. Further, the search method employed to infer the optimal number of mixture components lacks a rigorous treatment as they simply vary K and select the one that results in the best EM outcome.

5.3.3 Approximate Bayesian criterion (Roberts et al., 1998)

The approximate Bayesian method to model the Gaussian mixtures was proposed by Roberts et al. (1998). The method, also referred to as *Laplace-empirical criterion* (LEC) (McLachlan and Peel, 2000), uses a scoring function derived using Bayesian inference and serves to provide a trade-off between model complexity and the goodness-of-fit. The parameter estimates of $\mathbf{\Phi}$ are those that result in the minimum value of the following scoring function.

$$-\log \Pr(\mathcal{D}, \mathbf{\Phi}) = \mathcal{L}(\mathcal{D}|\mathbf{\Phi}) + Kd\log(2\alpha\beta\sigma_p^2) - \log(K-1)! - \frac{N_d}{2}\log(2\pi) + \frac{1}{2}\log(|H(\mathbf{\Phi})|)$$

where d is the dimensionality of the given data $\mathcal{D}$, $\mathcal{L}(\mathcal{D}|\mathbf{\Phi})$ is the negative log-likelihood, K is the number of mixture components, α and β are hyperparameters (which are set to 1 in their experiments), σ_p is a pre-defined constant or is pre-computed using the entire data, $H(\mathbf{\Phi})$ is the Hessian matrix which is equivalent to the empirical Fisher matrix for the set of component parameters, and p is the number of free parameters in the model.

Formulation of the scoring function: The scoring function formulated by Roberts et al. (1998) can be obtained as an approximation to the message length expression in Equation 5.5 by identifying the following related terms in both equations

1. $I(\mathbf{w})$ is approximated by $-\log(K-1)!$
2. For a d-variate Gaussian with mean $\boldsymbol{\mu}$ and covariance matrix $\mathbf{C}$, the joint prior $h(\boldsymbol{\mu}, \mathbf{C})$ is calculated as follows:
   - *Prior on $\boldsymbol{\mu}$:* Each of the d parameters of the mean direction are assumed to have uniform priors in the range $(-\alpha\sigma_p, \alpha\sigma_p)$, so that the prior density of the mean is $h(\boldsymbol{\mu}) = \frac{1}{(2\alpha\sigma_p)^d}$.
   - *Prior on $\mathbf{C}$:* It is assumed that the prior density is dependent only on the diagonal elements in $\mathbf{C}$. Each diagonal covariance element is assumed to have a prior in the range $(0, \beta\sigma_p)$ so that the prior on $\mathbf{C}$ becomes $h(\mathbf{C}) = \frac{1}{(\beta\sigma_p)^d}$

   The joint prior, is therefore, assumed to be $h(\boldsymbol{\mu}, \mathbf{C}) = \frac{1}{(2\alpha\beta\sigma_p^2)^d}$.
   Thus, $-\sum_{j=1}^{K} \log\, h(\boldsymbol{\Theta}_j)$ in Equation 5.5 is approximated by $Kd\log(2\alpha\beta\sigma_p^2)$
3. $\frac{1}{2}\sum_{j=1}^{K} \log|\mathcal{F}(\boldsymbol{\Theta}_j)|$ in Equation 5.5 is approximated by $\frac{1}{2}\log|H|$
4. The constant is approximated by $(p/2)\log(2\pi)$

Search method: The search method used to select the optimal number of components in this approach is similar to the one adopted by Oliver et al. (1996). The optimal number of mixture components is chosen by running the EM 10 times for every value of K within a given range. The optimal K is selected as the one for which the best of the 10 trials results in the least value of the scoring function.

Limitations: Although the formulation is an improvement over the previously discussed methods, there are some limitations due to the assumptions made while proposing the scoring function:

- While computing the prior density of the covariance matrix, the off-diagonal elements are ignored. Real-world data is often correlated and the prior density should reflect this. By ignoring the off-diagonal elements in the covariance matrix, it is being assumed that the different dimensions of the data points are not correlated.
- The computation of the determinant of the Fisher matrix is approximated by computing the Hessian $|H|$. It is to be noted that while the Hessian is the *observed information* (data dependent), the Fisher information is the *expectation* of the observed information. The MML formulation requires the use of the expected value because, when MML-based inference (Wallace and Freeman, 1987) is rationalized as a communication framework, the transmitter and receiver can encode and decode data losslessly only when the expected Fisher is used (see Section 2.4.1).
- The approximated Hessian was derived for Gaussians with diagonal covariances. For Gaussians with full covariance matrices, the Hessian was approximated by replacing the diagonal elements with the corresponding eigenvalues of the Hessian matrix. The empirical Fisher computed in this form does not guarantee the charactersitic invariance property of the classic MML method (Oliver and Baxter, 1994).

5.3.4 Integrated Complete Likelihood (Biernacki et al., 2000)

The Integrated Complete Likelihood (ICL) criterion was proposed by Biernacki et al. (2000). The criterion *maximizes* the *complete log-likelihood* (CL) given by

$$CL(\mathcal{D}, \boldsymbol{\Phi}) = -\mathcal{L}(\mathcal{D}|\boldsymbol{\Phi}) - \sum_{i=1}^{N}\sum_{j=1}^{K} z_{ij}\log r_{ij}$$

where $\mathcal{L}(\mathcal{D}|\mathbf{\Phi})$ is the negative log-likelihood of the data, r_{ij} is the responsibility term (Equation 5.3), and $z_{ij} = 1$ if $\mathbf{x}_i$ arises from component j and zero otherwise. The term $\sum_{i=1}^{N}\sum_{j=1}^{K} z_{ij} \log r_{ij}$ is explained as the estimated mean entropy.

Formulation of the scoring function: The ICL criterion is then defined as: $ICL(\mathcal{D}, \mathbf{\Phi}) = CL(\mathcal{D}, \mathbf{\Phi}) - \frac{p}{2} \log N$,where p is the number of free parameters in the model. As per the above definition, note that the scoring function has a form similar to that of BIC.

Search method: The search method adopted in conjunction with this scoring function is similar to the one used by Oliver et al. (1996) and Roberts et al. (1998). The EM algorithm is initiated 20 times for each value of K with random starting points and the best amongst those is chosen.

Limitations: As the formulation of the ICL scoring function is similar to the previously discussed information-theoretic criteria such as AIC and BIC, we note that the ICL scoring function penalizes each free parameter by a constant value and does not account for the actual model parameters. This restricts the ability of ICL to distinguish between mixture models containing the same number of components.

5.3.5 Split-Merge EM (Ueda et al., 2000)

While estimating the parameters of a K-component mixture, the traditional EM algorithm (Section 5.2.1) can potentially be trapped in a local optimum. The *split-merge* based EM (SMEM) method was proposed by Ueda et al. (2000) to improve on the EM solution, by perturbing the mixture components through a deterministic series of split and merge operations applied on the components.

Search method: Given a mixture with K components that is optimized using the EM algorithm, the SMEM method selects the top three candidates, merges two of them, and splits the other into two, thus, leaving the effective number of components unchanged. The potential candidates are chosen depending on the improvement to the complete data log-likelihood function (used to formulate the ICL criterion).

Limitations: The SMEM algorithm does not facilitate an increase or decrease in the mixture size. Once the three top candidates in a parent mixture that improve the ICL value are determined, the parent mixture is updated, and the algorithm is repeated with the updated parent. The authors note that such a heuristic doesn't necessarily guarantee the best selection of candidates for perturbing the parent mixture. They keep trying until they find the set of three candidates (a maximum of 5 random trials is used in their experiments) It is therefore, not exhaustive, as there could be a combination of three components that could be potentially left out.

5.3.6 Unsupervised Learning of Finite Mixtures (Figueiredo and Jain, 2002)

The mixture modelling method due to Figueiredo and Jain (2002) uses a simplified MML-like criterion to formulate the scoring function. This formulation can be intrepreted as a two-part message for encoding the model parameters and the observed data as follows

$$I(\mathcal{D}, \mathbf{\Phi}) = \underbrace{\frac{N_p}{2}\sum_{j=1}^{K} \log\left(\frac{N w_j}{12}\right) + \frac{K}{2}\log\frac{N}{12} + \frac{K(N_p+1)}{2}}_{\text{first part}} + \underbrace{\mathcal{L}(\mathcal{D}|\mathbf{\Phi})}_{\text{second part}} \tag{5.8}$$

where N_p is the *number* of free parameters per component and w_j is the component weight.

Formulation of the scoring function: This scoring function is derived from Equation 5.5 by assuming the prior density of the component parameters to be a Jeffreys prior. According to Jeffreys (1946), if $\mathbf{\Theta}_j$ is the vector of parameters describing the j^{th} component, then the prior density $h(\mathbf{\Theta}_j) \propto \sqrt{|\mathcal{F}(\mathbf{\Theta}_j)|}$. Applying this prior to the mixture modelling problem, the prior for weights would result in $h(w_1, \ldots, w_K) \propto (w_1 \ldots w_K)^{-1/2}$. These assumptions are used in the encoding of the parameters which correspond to the first part of the message.

Limitations: We note that the scoring function is consistent with the MML scheme of encoding parameters and the data using those parameters. However, the formulation can be improved by amending the assumptions as detailed in our approach in Section 3.5.2, for Gaussian mixtures. Further, the assumptions made in Figueiredo and Jain (2002) have the following side effects

- The value of $-\log \dfrac{h(\mathbf{\Theta}_j)}{\sqrt{|\mathcal{F}(\mathbf{\Theta}_j)|}}$ gives the cost of encoding the component parameters. By assuming $h(\mathbf{\Theta}_j) \propto \sqrt{|\mathcal{F}(\mathbf{\Theta}_j)|}$, the message length associated with using any vector of parameters $\mathbf{\Theta}_j$ is essentially treated the same. To avoid this, the use of independent uniform priors over non-informative Jeffreys's priors was advocated previously (Oliver et al., 1996; Lee, 1997; Roberts et al., 1998).

  The use of Jeffreys prior offers certain advantages, for example, not having to compute the Fisher information (Jeffreys, 1946). However, this is important and cannot be ignored as it dictates the *precision of encoding the parameter vector*. Wallace (2005) state that "Jeffreys, while noting the interesting properties of the prior formulation did not advocate its use as a genuine expression of prior knowledge."

  Therefore, by making this assumption, Figueiredo and Jain (2002) avoid the difficulty associated with explicitly computing the Fisher information. As a result, for encoding the parameters of the entire mixture, *only* the cost associated with encoding the component weights is considered.

- The code length to state each $\mathbf{\Theta}_j$ is, therefore, greatly simplified as $(N_p/2)\log(Nw_j)$ (notice the sole dependence on weight w_j). Figueiredo and Jain (2002) interpret this as being similar to an MDL/BIC formulation because Nw_j gives the expected number of data points generated by the j^{th} component. This is equivalent to the BIC criterion discussed earlier. We note that MDL/BIC are highly simplified versions of MML formulation and therefore, Equation 5.8 does not capture the entire essence of complexity and goodness-of-fit accurately.

Search method: The method begins by assuming a large number of components and updates the weights iteratively in the EM steps as

$$w_j = \frac{\max\left\{0, n_j - \frac{N_p}{2}\right\}}{\sum_{j=1}^{K} \max\left\{0, n_j - \frac{N_p}{2}\right\}} \tag{5.9}$$

where n_j is the effective membership of data points in j^{th} component (Equation 5.4). A component is eliminated when its weight becomes zero and, consequently, the number of mixture components decreases. We note that the search method proposed by Figueiredo and Jain (2002) using the MML criterion is an improvement over the methods they compare against.

Limitations: The search method of Figueiredo and Jain (2002) has the following associated problems

- The method updates the weights as given by Equation 5.9. During any iteration, if the amount of data allocated to a component is less than $N_p/2$, its weight is updated as zero and this component is ignored in subsequent iterations. This imposes a lower bound on the amount of

data that can be assigned to each component. As an example, for a 10-dimensional Gaussian mixture, the number of free parameters per component is $N_p = 65$ and, hence, the lower bound is 32.5. Thus, in this example, if a component has ~ 30 data points, the mixture size is reduced and these data are assigned to some other component(s). Consider a scenario where there are 50 observed 10 dimensional data points originally generated by a mixture with two components with equal mixing proportions. The method of Figueiredo and Jain (2002) would always infer that there is only one component regardless of the separation between the two components. This is clearly a wrong inference (see Section 5.5.4 for the relevant experiments).

- Once a component is discarded, the mixture size decreases by one, and it cannot be recovered. Because the memberships n_j are updated iteratively using an EM algorithm and because EM might not always lead to global optimum, it is conceivable that the updated values need not always be optimal. This might lead to situations where a component is deleted owing to its low prominence. There is no provision to increase the mixture size in the subsequent stages of the algorithm to account for such behaviour.

- The method assumes a large number of initial components in an attempt to be robust with respect to EM initialization. However, this places a significant overhead on the computation due to the need to handle several components.

5.3.7 MML-Snob mixture modelling (Wallace and Boulton, 1968)

The Snob program is an MML-based mixture modelling software for unsupervised learning (Wallace and Boulton, 1968; Boulton and Wallace, 1970). The method progressively modifies the number of mixture components by a series of reclassifying, splitting, merging, and swapping operations on the components. Each operation results in a modified mixture that is evaluated based on the reduction to the total message length.

There have been several versions of Snob since Boulton and Wallace (1970). The initial implementation of Snob dealt with complete assignments of data to the mixture components. Wallace (1986) suggested improvements to allow for partial assignments. The common theme across the different revisions of Snob has been the assumption that all attributes in the data are independent within each component (Jorgensen and McLachlan, 2008). This assumption has been used in modelling mixtures of multi-state, Poisson, von Mises circular and Gaussian distributions (Wallace and Dowe, 2000).

Limitations: The independent assumption of the data attributes limits the use of Snob when the data has correlated attributes. As an example, for mixture modelling using multivariate Gaussian mixtures, this assumption only facilitates the use of Gaussian distributions with diagonal covariance matrices.

Further, as noted by Jorgensen and McLachlan (2008), there are limited extensions of the methodology to incorporate multivariate distributions of different types. Agusta and Dowe (2002) developed an extension for mixtures of multivariate Gaussian distributions. We note that the although the strategies employed in changing the number of mixture components are generalizable, however, the implementation of Snob inhibits its extension to model multivariate data whose attributes are correlated.

Summary: We observe that while all these methods (and many more) work well within their defined scope, they are incomplete in achieving the true objective, that is, to rigorously score models and their ability to fit the data. The methods discussed above can be seen as different approximations to the MML framework. They adopted various simplifying assumptions and approximations. To avoid such limitations, we developed a classic MML formulation, giving the complete message length formulations for the various probability distributions.

Secondly, in most of these methods, the search for the optimal number of mixture components is achieved by selecting the mixture that results in the best EM outcome out of many trials (Akaike, 1974; Schwarz, 1978; Oliver et al., 1996; Roberts et al., 1998; Biernacki et al., 2000). This is not an elegant

solution and Figueiredo and Jain (2002) proposed a search heuristic which integrates estimation and model selection. A comparative study of these methods is presented in McLachlan and Peel (2000). Their analysis suggested the superior performance of ICL (Biernacki et al., 2000) and LEC (Roberts et al., 1998). Later, Figueiredo and Jain (2002) demonstrated that their proposed method outperforms the contemporary methods based on ICL and LEC and is regarded as the current state of the art. The proposed search method is, therefore, compared against that of Figueiredo and Jain (2002) to demonstrate its effectiveness.

With this background, we formulate an alternate search heuristic to infer the optimal number of mixture components that aims to address the above limitations.

5.4 Proposed search method to infer an optimal mixture

The space of candidate mixture models to explain the given data is infinitely large. As per the MML criterion (Equation 5.5), the goal is to search for the mixture that has the least total message length. If the number of mixture components are fixed, then the EM algorithm (Section 5.2.2) can be used to estimate the mixture parameters, namely the component weights and the parameters of each component. However, here, it is required to search for the optimal *number* of mixture components along with the corresponding mixture parameters.

The proposed search method extends the MML-based Snob program (Wallace and Boulton, 1968; Wallace, 1986; Jorgensen and McLachlan, 2008) for unsupervised learning. We define three operations, namely *split, delete,* and *merge* that can be applied to any component in the mixture.

5.4.1 The search algorithm

The pseudocode of the proposed method is presented in Algorithm 1. The basic idea behind the search strategy is to *perturb* a mixture from its current sub-optimal state to obtain a new state (if the perturbed mixture results in a smaller message length).

In general, if a (current) mixture has K components, it is perturbed using a series of *Split, Delete*, and *Merge* operations to check for improvement. Each component is split and the new $(K+1)$-component mixture is re-estimated. If there is an improvement (that is, if there is a decrease in message length with respect to the current mixture), the new $(K+1)$-component mixture is retained. There are K splits possible and the one that results in the greatest improvement is recorded (see lines 5 - 7 in Algorithm 1). A component is first split into two sub-components (children) which are locally optimized by the EM algorithm on the data that belongs to that sole component. The child components are then integrated with the others and the mixture is then optimized to generate a $(K+1)$-component mixture. The reason being it is better to start from some already optimized state to reach an improved state, rather than use random initial values in the EM algorithm.

Similarly, each of the components is then deleted, one after the other, and the $(K-1)$-component mixture is compared against the current mixture. There are K possible deletions and the best amongst these is recorded (see lines 8 - 11 in Algorithm 1). Finally, the components in the current mixture are merged with their closest matches (determined by calculating the KL-divergence) and each of the resultant $(K-1)$-component mixtures are evaluated against the K component mixture. The best among these merged mixtures is then retained (see lines 12 - 15 in Algorithm 1).

The algorithm starts by assuming a one-component mixture. This component is split into two children which are locally optimized. If the split results in a better model, it is retained. For any given K-component mixture, there might be improvement due to splitting, deleting and/or merging its components. We select the perturbation that best improves the current mixture. This process is repeated until there is no further improvement possible. The notion of *best* or improved mixture is based on the amount of reduction of message length that the perturbed mixture provides. In the current state, the observed data have partial memberships in each of the K components. Before the execution of each operation, these memberships need to be adjusted and EM steps are subsequently

Algorithm 1: Search for an optimal mixture model

```
1  current ← one-component-mixture
2  while true do
3      components ← current mixture components
4      K ← number of components
5      for i ← 1 to K do                          // exhaustively split all components
6          splits[i] ← SPLIT(current, components[i])
7      BestSplit ← best(splits)                   // remember the best split
8      if K > 1 then
9          for i ← 1 to K do                      // exhaustively delete all components
10             deletes[i] ← DELETE(current, components[i])
11         BestDelete ← best(deletes)             // remember the best deletion
12         for i ← 1 to K do                      // exhaustively merge all components
13             j ← closest-component(i)
14             merges[i] ← MERGE(current, i, j)
15         BestMerge ← best(merges)               // remember the best merge
                                                  /* select the best perturbation */
16     BestPerturbation ← best(BestSplit, BestDelete, BestMerge)
       ΔI ← message_length(BestPerturbation) − message_length(current)
17     if ΔI < 0                                  // check for improvement
18     then
19         current ← BestPerturbation
20         loop from line 2
21     else
22         stop
23 return current
```

carried out to achieve an optimum with a different number of components. Below, each of these operations is examined in detail including how the memberships are adjusted after each operation.

5.4.2 Strategic operations employed to determine an optimal mixture

Let $R = [r_{ij}]$ be the $N \times K$ responsibility (membership) matrix and w_j be the weight of j^{th} component in mixture $\mathcal{M}$.

SPLIT OPERATION (LINE 6 IN ALGORITHM 1): Assume a component with index $\alpha \in \{1, K\}$ and weight w_α in the current mixture $\mathcal{M}$ is being split to generate two child components. The goal is to find two distinct clusters amongst the data associated with component α. It is to be noted that the data have fractional memberships in component α. The EM is therefore, carried out *within* component α assuming a *two-component sub-mixture* with the data weighted as per their current memberships $r_{i\alpha}$. The remaining $(K-1)$ components are untouched. A sequence of EM steps are carried out to optimize the two-component sub-mixture. The initial state and the subsequent updates in the Maximization-step of the EM algorithm are described below.

Parameter initialization of the two-component sub-mixture: The goal is to identify two distinct clusters within component α. For example, for *Gaussian* mixtures, to provide a reasonable starting point, we can compute the direction of maximum variance of the parent component and locate two points that are one standard deviation away on either side of its mean (along this direction). These points serve as the initial means for the two children generated due to splitting the parent component. Selecting the initial means in this manner ensures they are reasonably apart from each other and serves as a

good starting point for optimizing the two-component sub-mixture. The memberships are initialized by allocating the data points to the closest of the two means. Once the means and the memberships are initialized, the covariance matrices of the two child components are computed.

There are conceivably several variations to how the two-component sub-mixture can be initialized. These include random initialization, selecting two data points as the initial component means, and many others. However, the reason for selecting the direction of maximum variance is to utilize the already available characteristic of the data, that is, the distribution within component α.

Once the parameters of the sub-mixture are initialized, an EM algorithm is carried out (just for the sub-mixture) with the following Maximization-step updates. Let $R^c = [r^c_{ik}]$ be the $N \times 2$ responsibility matrix for the two-component sub-mixture. For $k \in \{1, 2\}$, let $n^{(k)}_\alpha$ be the effective memberships of data belonging to the two child components, let $w^{(k)}_\alpha$ be the weights of the child components within the sub-mixture, and let $\boldsymbol{\Theta}^{(k)}_\alpha$ be the parameters describing the child components. The effective memberships are updated as given below

$$n^{(k)}_\alpha = \sum_{i=1}^{N} r^c_{ik} \quad \text{and} \quad n^{(1)}_\alpha + n^{(2)}_\alpha = N$$

As the sub-mixture comprises of two child components, substitute $K = 2$ in Equation 5.6 to obtain the updates for the weights. These are given as

$$w^{(k)}_\alpha = \frac{n^{(k)}_\alpha + \frac{1}{2}}{N + 1} \quad \text{and} \quad w^{(1)}_\alpha + w^{(2)}_\alpha = 1$$

For *Gaussian* mixtures, the component parameters $\boldsymbol{\Theta}^{(k)}_\alpha = (\widehat{\boldsymbol{\mu}}^{(k)}_\alpha, \widehat{\mathbf{C}}^{(k)}_\alpha)$ are updated as

$$\widehat{\boldsymbol{\mu}}^{(k)}_\alpha = \frac{\sum_{i=1}^{N} r_{i\alpha} r^c_{ik} \mathbf{x}_i}{\sum_{i=1}^{N} r_{i\alpha} r^c_{ik}} \quad \text{and} \quad \widehat{\mathbf{C}}^{(k)}_\alpha = \frac{\sum_{i=1}^{N} r_{i\alpha} r^c_{ik} (\mathbf{x}_i - \widehat{\boldsymbol{\mu}}^{(k)}_\alpha)(\mathbf{x}_i - \widehat{\boldsymbol{\mu}}^{(k)}_\alpha)^T}{\sum_{i=1}^{N} r_{i\alpha} r^c_{ik} - 1} \tag{5.10}$$

The difference between the EM updates in Equations 5.7 and 5.10 is the presence of the coefficient $r_{i\alpha} r^c_{ik}$ with each $\mathbf{x}_i$. Since the sub-mixture is considered, the original responsibility $r_{i\alpha}$ is multiplied by the responsibility within the sub-mixture r^c_{ik} to quantify the influence of datum $\mathbf{x}_i$ to each of the child components.

After the sub-mixture is locally optimized, it is integrated with the unperturbed $(K-1)$ components of $\mathcal{M}$ to result in a $(K+1)$-component mixture $\mathcal{M}'$. An EM is finally carried out on the combined $(K+1)$ components to estimate the parameters of $\mathcal{M}'$ and, thus, result in an optimized $(K+1)$-component mixture.

EM initialization for $\mathcal{M}'$: Usually, the EM is started by a random initialization of the members. However, because the two-component sub-mixture is now optimal and the $(K-1)$ components in $\mathcal{M}$ are also in an optimal state, we exploit this situation to initialize the EM (for $\mathcal{M}'$) with a reasonable starting point. As mentioned above, the component with index α and weight w_α is split. Upon integration, the (child) components that replaced component α will now correspond to indices α and $\alpha + 1$ in the new mixture $\mathcal{M}'$. Let $R' = [r'_{ij}]\ \forall 1 \leq i \leq N, 1 \leq j \leq K+1$ be the responsibility matrix for the new mixture $\mathcal{M}'$ and let w'_j be the component weights in $\mathcal{M}'$.

- *Component weights:* The weights are initialized as

$$w'_j = w_j \quad \text{if} \quad j < \alpha \quad \text{and} \quad w'_j = w_{j-1} \quad \text{if} \quad j > \alpha + 1$$
$$w'_\alpha = w_\alpha w^{(1)}_\alpha \quad \text{and} \quad w'_{\alpha+1} = w_\alpha w^{(2)}_\alpha$$

- *Memberships:* The responsibility matrix R' is initialized for all data $\mathbf{x}_i \, \forall 1 \leq i \leq N$ as

$$r'_{ij} = r_{ij} \quad \text{if} \quad j < \alpha \quad \text{and} \quad r'_{ij} = r_{i\,(j-1)} \quad \text{if} \quad j > \alpha + 1$$
$$r'_{i\alpha} = r_{i\alpha} r^c_{i1} \quad \text{and} \quad r'_{i\,\alpha+1} = r_{i\alpha} r^c_{i2}$$
$$n'_j = \sum_{i=1}^{N} r'_{ij} \quad \forall\, 1 \leq j \leq K+1$$

  where n'_j are the effective memberships of the components in $\mathcal{M}'$.

With these starting points, the parameters of $\mathcal{M}'$ are estimated using the traditional EM algorithm with updates in the Maximization-step given by Equations 5.6 and 5.7. The EM results in local convergence of the $(K+1)$-component mixture. If the resultant message length of encoding data using $\mathcal{M}'$ is lower than that due to $\mathcal{M}$, that means the perturbation of $\mathcal{M}$ due to splitting component α resulted in a new mixture $\mathcal{M}'$ that compresses the data better and, hence, is a better mixture model to explain the data.

Delete operation (Line 10 in Algorithm 1): The goal here is to remove a component from the current mixture and check whether it results in a better mixture model to explain the observed data. Assume the component with index α and weight w_α is to be deleted from $\mathcal{M}$ to generate a $(K-1)$-component mixture $\mathcal{M}'$. Once deleted, the data memberships of the component need to be redistributed between the remaining components. The redistribution of data results in a good starting point to employ the EM algorithm to estimate the parameters of $\mathcal{M}'$.

EM initialization for $\mathcal{M}'$: Let $R' = [r'_{ij}]$ be the $N \times (K-1)$ responsibility matrix for the new mixture $\mathcal{M}'$ and let w'_j be the weight of the j^{th} component in $\mathcal{M}'$.

- *Component weights:* The weights are initialized as

$$w'_j = \frac{w_j}{1 - w_\alpha} \quad \text{if} \quad j < \alpha \quad \text{and} \quad w'_j = \frac{w_{j+1}}{1 - w_\alpha} \quad \text{if} \quad j \geq \alpha$$

  It is to be noted that $w_\alpha \neq 1$ because the MML update expression in the M-step for the component weights always ensures non-zero weights during every iteration of the EM algorithm (see Equation 5.6).

- *Memberships:* The responsibility matrix R' is initialized for all data $\mathbf{x}_i \, \forall\, 1 \leq i \leq N$ as

$$r'_{ij} = \frac{r_{ij}}{1 - r_{i\alpha}} \quad \text{if} \quad j < \alpha \quad \text{and} \quad r'_{ij} = \frac{r_{i\,(j+1)}}{1 - r_{i\alpha}} \quad \text{if} \quad j \geq \alpha$$
$$n'_j = \sum_{i=1}^{N} r'_{ij} \quad \forall\, 1 \leq j \leq K-1$$

  where n'_j is the effective membership of the data in component j in $\mathcal{M}'$. It is possible for a datum $\mathbf{x}_i$ to have complete membership in component α (that is, $r_{i\alpha} = 1$), in which case, its membership is equally distributed among the other $(K-1)$ components (that is, $r'_{ij} = \frac{1}{K-1}, \forall\, j \in \{1, K-1\}$).

With these readjusted weights and memberships, and the constituent $(K-1)$ components, the traditional EM algorithm is used to estimate the parameters of the new mixture $\mathcal{M}'$. If the resultant message length of encoding data using $\mathcal{M}'$ is lower than that due to $\mathcal{M}$, that means the perturbation of $\mathcal{M}$ due to deleting component α results in a new mixture $\mathcal{M}'$ with improved explanatory power of the given data and, hence, is an improvement over the current mixture.

MERGE OPERATION (LINE 14 IN ALGORITHM 1): The goal is to join a pair of components in $\mathcal{M}$ and determine whether the resulting $(K-1)$-component mixture $\mathcal{M}'$ is any better than the current mixture $\mathcal{M}$. One strategy to identify an improved mixture model would be to consider merging all possible pairs of components and choose the one which results in the greatest improvement. This would, however, lead to a runtime complexity of $O(K^2)$, which could be significant for large values of K. Another strategy is to consider merging components which are "close" to each other.

For a given component, its *closest* component is identified by computing the Kullback-Leibler (KL) distance with all others and selecting the one with the least value. This would result in a linear runtime complexity of $O(K)$ as computation of KL-divergence is a constant time operation. For every component in $\mathcal{M}$, its closest match is identified and they are merged to obtain a $(K-1)$-component mixture $\mathcal{M}'$. Merging the pair involves reassigning the component weights and the memberships. An EM algorithm is then employed to optimize $\mathcal{M}'$.

Assume components with indices α and β are merged. Let their weights be w_α and w_β; and their responsibility terms be $r_{i\alpha}$ and $r_{i\beta}, 1 \leq i \leq N$ respectively. The component that is formed by merging the pair is determined first. It is then integrated with the $(K-2)$ remaining components of $\mathcal{M}$ to produce a $(K-1)$-component mixture $\mathcal{M}'$.

EM initialization for $\mathcal{M}'$: Let $w^{(m)}$ and $r_i^{(m)}$ be the weight and responsibility vector of the merged component m, respectively. They are given as

$$w^{(m)} = w_\alpha + w_\beta \quad \text{and} \quad r_i^{(m)} = r_{i\alpha} + r_{i\beta}, 1 \leq i \leq N$$

In the case of Gaussian mixtures, the parameters $\boldsymbol{\Theta}^{(m)} = (\widehat{\boldsymbol{\mu}}^{(m)}, \widehat{\mathbf{C}}^{(m)})$ of this merged component are estimated as

$$\widehat{\boldsymbol{\mu}}^{(m)} = \frac{\sum_{i=1}^{N} r_i^{(m)} \mathbf{x}_i}{\sum_{i=1}^{N} r_i^{(m)}} \quad \text{and} \quad \widehat{\mathbf{C}}^{(m)} = \frac{\sum_{i=1}^{N} r_i^{(m)} (\mathbf{x}_i - \widehat{\boldsymbol{\mu}}^{(m)})(\mathbf{x}_i - \widehat{\boldsymbol{\mu}}^{(m)})^T}{\sum_{i=1}^{N} r_i^{(m)} - 1}$$

The merged component m with weight $w^{(m)}$, responsibility vector $r_i^{(m)}$, and parameters $\boldsymbol{\Theta}^{(m)}$ is then integrated with the $(K-2)$ components. The merged component and its associated memberships along with the $(K-2)$ other components serve as the starting point for optimizing the new mixture $\mathcal{M}'$. If $\mathcal{M}'$ results in a lower message length compared to $\mathcal{M}$, it means the perturbation of $\mathcal{M}$ because of merging the pair of components resulted in an improvement to the current mixture.

5.4.3 Illustrative example of the search procedure

The proposed search and inference of the mixture components is explained through the following example that was also considered by Figueiredo and Jain (2002). Consider a bivariate Gaussian mixture shown in Figure 5.1 (the data points are coloured based on their density values). The mixture has three components with equal weights of 1/3 each and their means at (-2,0), (0,0), and (2,0). The covariance matrices of the three components are the same and are equal to $\text{diag}\{2, 0.2\}$. We simulate 900 data points from this mixture (as done by Figueiredo and Jain (2002)) and employ the proposed search strategy. The progression of the search method using various operations is detailed below.

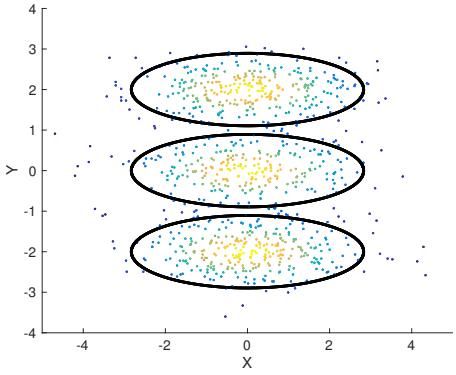
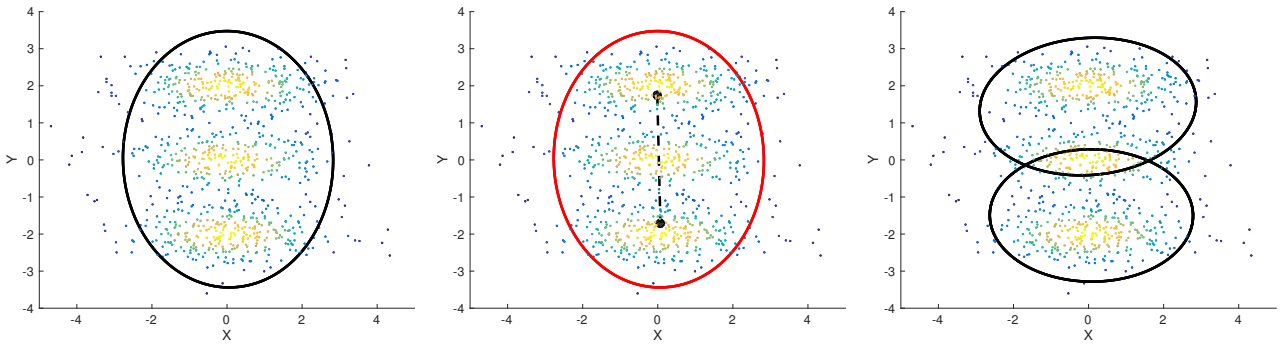


Figure 5.1: Original mixture

Search for the optimal mixture model

The method begins by inferring a one-component mixture P_1 (Figure 5.2a). It then splits the component (described in SPLIT step of Section 5.4.2) and checks whether there is an improvement in explanation. The red ellipse in Figure 5.2(b) depicts the component being split. The direction of maximum variance (dotted black line) is first identified, and the means (shown by black dots at the end of the dotted line) are initialized. An EM algorithm is then used to optimize the two children. This results in a mixture P_2 shown in Figure 5.2(c). Since the new mixture has a lower message length, the current is updated as P_2.



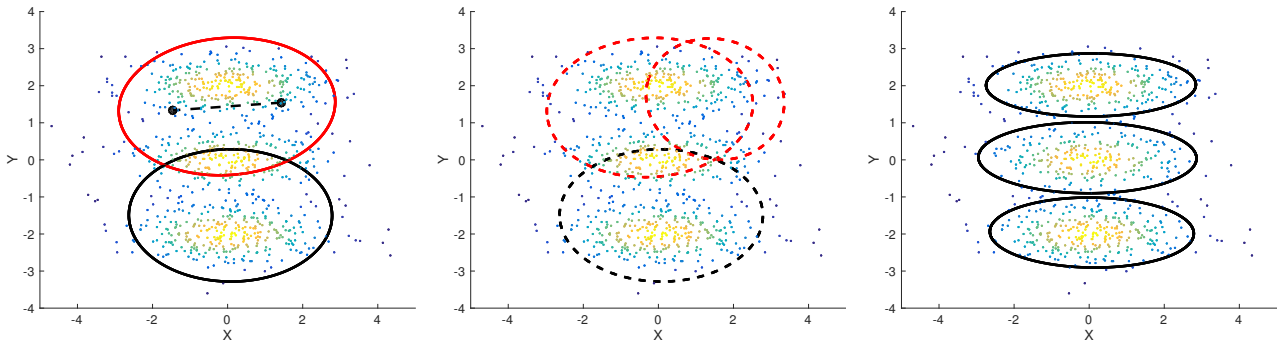
(a) (b) (c)

Figure 5.2: (a) P_1: the one-component mixture after the first iteration (message length $I = 22793$ bits) (b) Red colour denotes the component being split. The dotted line is the direction of maximum variance. The black dots represent the initial means of the two-component sub-mixture (c) P_2: optimized mixture post-EM phase ($I = 22673$ bits) results in an improvement.

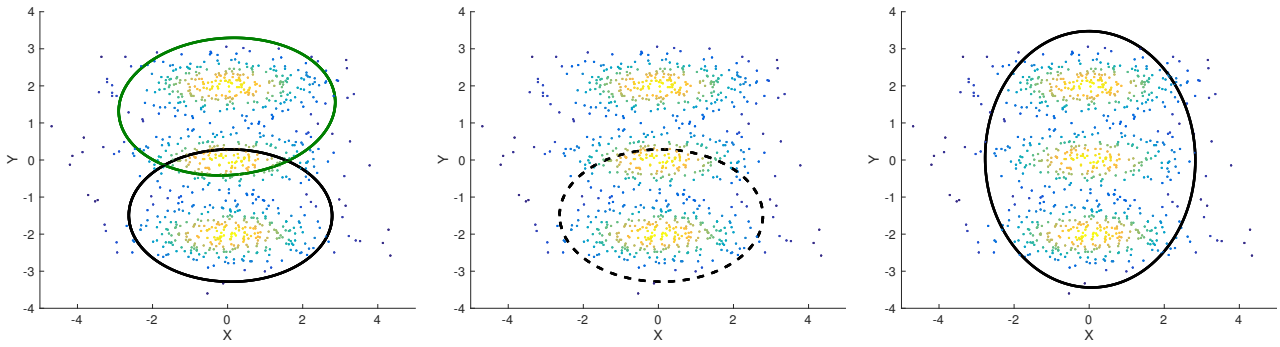
In the second iteration, each component in P_2 is iteratively split, deleted, and merged. Figure 5.3(a)-(c) shows the splitting of the first component (red). On splitting, the new mixture P_3 results in a lower message length. Deletion of the first component is shown in Figure 5.3(d)-(f). Before merging the first component, its closest component is identified (with the least KL divergence) (see Figure 5.3g). Deletion and merging operations, in this case, do not result in an improvement. These two operations have different intermediate EM initializations, shown in Figures 5.3(e) and (h) but result in the same optimized one-component mixture. The same set of operations are performed on the second component in P_2. In this particular case, splitting results in an improved mixture (same as P_3).

The new parent is updated as P_3 and the series of split, delete, and merge operations are carried out on all components in P_3. Figure 5.4 shows these operations on the first component. We see that splitting the first component in P_3 results in P_4 (see Figure 5.4c). However, P_4 is not an improvement over P_3 as seen by the message lengths and is, therfore, discarded. Similarly, deletion and merging of the components do not yield improvements to P_3. The operations are carried out on the remaining two components in P_3 (not shown in the figure) too. These perturbations do not produce improved mixtures in terms of the total message length. Since the third iteration does not result in any further improvement, the search terminates and the parent P_3 is considered to be the best mixture.

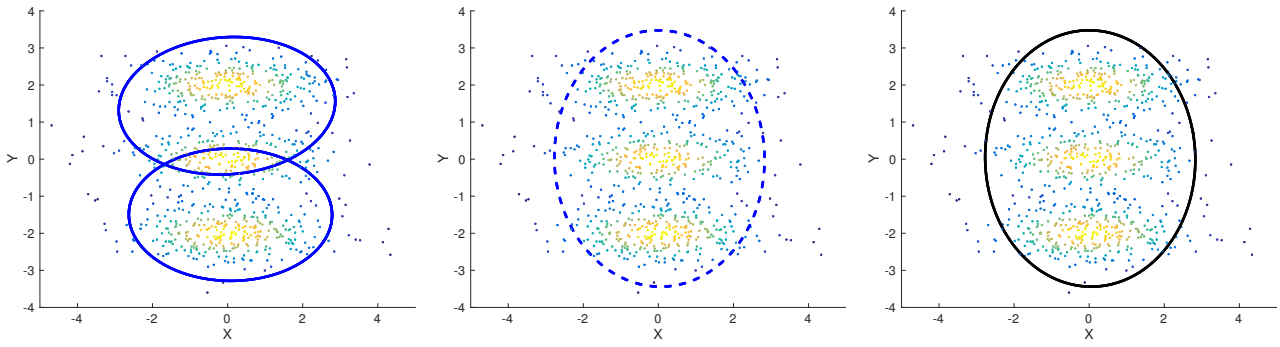
In different stages of the search method, there are different intermediate mixtures. The EM algorithm is a gradient descent technique and it can get trapped in a local optimum. By employing the suggested search, we are exhaustively considering the possible options, and aiming to reduce the possibility of the EM getting stuck in a local optimum. The proposed method infers a mixture by balancing the tradeoff due to model complexity and the fit to the data. This is particularly useful when there is no prior knowledge pertaining to the nature of the data. In such a case, this method provides an objective way to infer a mixture with suitable components that best explains the data through lossless compression.



(a) Initial means (black dots) (b) Optimized children ($I = 22691$ bits) (c) P_3: post-EM ($I = 22460$ bits)



(d) Green ellipse is deleted (e) EM initialization ($I = 25599$ bits) (f) post-EM ($I = 22793$ bits)



(g) Blue ellipses are merged (h) EM initialization (i) post-EM ($I = 22793$ bits)

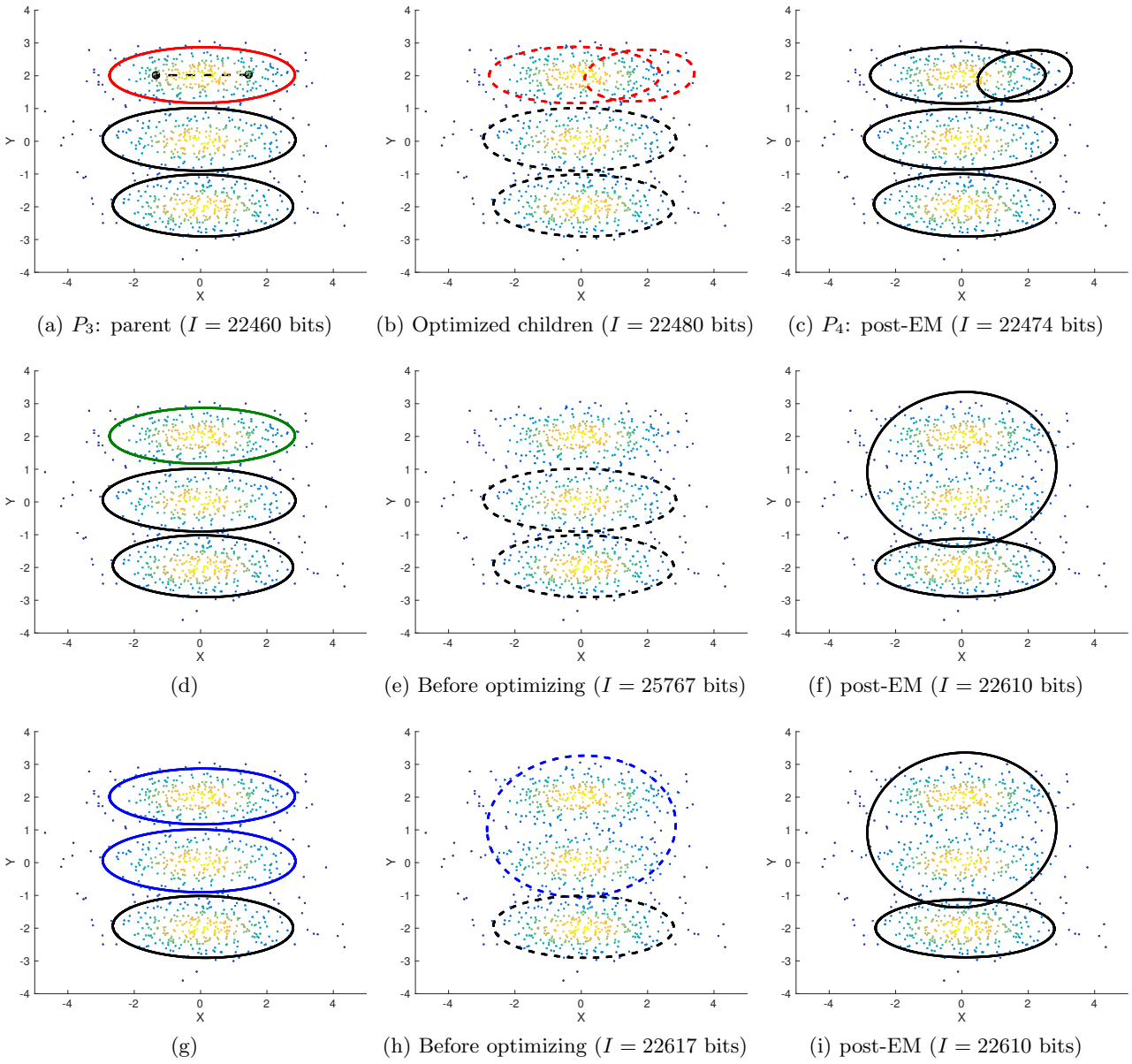
Figure 5.3: Second iteration (perturbations of the first component in P_2): (a)-(c) *Splitting* results in an improved mixture P_3 (d)-(f) *Deleting* – no improvement (g)-(i) *Merging* the two components – no improvement

Progression of the two-part message length during the search phase

The search method infers three components and terminates. In order to demonstrate that the inferred number of components is the optimum number, we infer mixtures with increasing number of components (until it reaches $K = 15$ as an example) and plot their resultant message lengths. For each $K > 3$, the standard EM algorithm (Section 5.2.2) is employed to infer the mixture parameters.

Figure 5.5 shows the total message lengths to which the EM algorithm converges for varying number of components M. As expected, the total message length (green curve) drastically decreases initially until $K = 3$ components are inferred. Starting from $K = 4$, the total message length gradually increases, clearly suggesting that the inferred models are over-fitting the data with increasing statement cost to encode the additional parameters of these (more complex) models.

The reason for the initial decrease and subsequent increase in the total message length is further elaborated. As per the MML criterion, the message length comprises of two parts – statement cost



(a) P_3: parent ($I = 22460$ bits) (b) Optimized children ($I = 22480$ bits) (c) P_4: post-EM ($I = 22474$ bits)

(d) (e) Before optimizing ($I = 25767$ bits) (f) post-EM ($I = 22610$ bits)

(g) (h) Before optimizing ($I = 22617$ bits) (i) post-EM ($I = 22610$ bits)

Figure 5.4: Third iteration: Operations involving the first component (a)-(c) denote the *splitting* process, (d)-(f) denote the *deletion* process, and (g)-(i) shows the *merging* of the first component with its closest component.

for the parameters and the cost for stating the data using those parameters. The model complexity (which corresponds to the mixture parameters) increases with increasing K. Therefore, the first part of the message to encode parameters increases with an increase in the number of parameters. This behaviour is illustrated by the red curve in Figure 5.5. The first part message lengths are shown in red on the right side Y-axis in the figure.

As the mixture model becomes increasingly more complex, the error of fitting the data decreases. This corresponds to the second part of the message in the MML encoding framework. This behaviour is consistent with what is observed in Figure 5.5 (blue curve). There is a sharp fall until $K = 3$; then onwards increasing the model complexity does not lower the error significantly. The error saturates and there is minimal gain with regards to encoding the data (the case of overfitting). However, the model complexity dominates after $K = 3$. The optimal trade-off is achieved when $K = 3$. We note that for a fixed number of mixture components, the EM algorithm for the MML measure is monotonically decreasing. However, while searching for the number of components using our proposed split, delete,

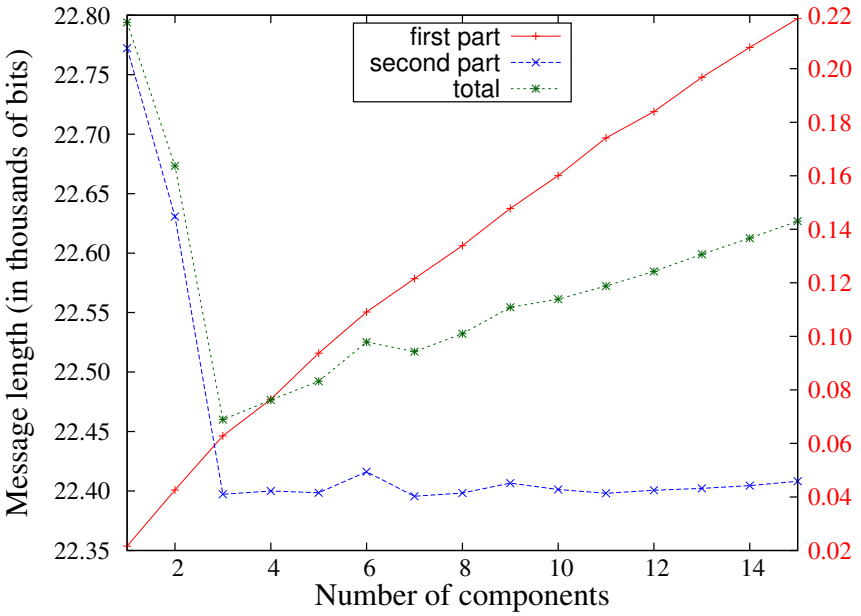


Figure 5.5: Variation of the individual parts of the total message length with increasing number of components (note the two Y-axes have different scales – the first part follows the right side Y-axis; the second part and total message lengths follow the left side Y-axis)

and merge operations, MML continues to decrease until some optimum is found and then steadily increases as illustrated through this example.

5.5 Experimental analyses of multivariate Gaussian mixtures

The search and inference methodology proposed here is compared against the widely cited method of Figueiredo and Jain (2002) using the Gaussian mixtures. The authors tested the performance of their method against that of Bayesian Information Criterion (BIC), the Integrated Complete Likelihood (ICL), and the approximate Bayesian (LEC) methods (discussed in Section 5.3). It was shown that the method of FJ[2] was superior than BIC, ICL and LEC (using Gaussian mixtures). In this section, a series of experiments are conducted to demonstrate that the proposed approach to infer mixtures fares better when compared with the method of FJ.

The experimental setup is as follows. We generate a random sample of data from a given Gaussian mixture $\mathcal{M}^t$ (true distribution). On this random sample, we use our proposed search method to infer the mixture. This is repeated 50 times and the performance of the proposed method is compared against that of FJ. The analyses include comparison of the number of inferred components as well as the quality of the inferred mixtures.

5.5.1 Methodologies used to compare the inferred mixtures

Comparing message lengths: The difference in message lengths gives the log-odds posterior ratio of any two mixtures (Equation 2.5). Given some observed data, and any two mixtures, one can determine which of the two best explains the data. The proposed methodology uses the scoring function (I_{MML}) defined in Equation 5.5. As elaborated in Section 5.3.6, FJ use an approximated MML-like scoring function (I_{FJ}) given by Equation 5.8.

[2]From here on, we will use the short form FJ to refer to Figueiredo and Jain (2002)

Both the proposed and FJ's search methods are employed to infer mixtures of the same data. Let these inferred mixtures be $\mathcal{M}^*$ and $\mathcal{M}^{FJ}$, respectively. We compute the following two quantities

$$\Delta I_{MML} = I_{MML}(\mathcal{M}^{FJ}) - I_{MML}(\mathcal{M}^*) \quad \text{and} \quad \Delta I_{FJ} = I_{FJ}(\mathcal{M}^{FJ}) - I_{FJ}(\mathcal{M}^*) \tag{5.11}$$

The two different scoring functions are used to compute the differences in message lengths of the resulting mixtures $\mathcal{M}^{FJ}$ and $\mathcal{M}^*$. Since the search method used to obtain $\mathcal{M}^*$ optimizes the scoring function I_{MML}, it is expected that $I_{MML}(\mathcal{M}^*) < I_{MML}(\mathcal{M}^{FJ})$ and consequently $\Delta I_{MML} > 0$. This implies that the proposed method is performing better using our defined objective function. However, if $I_{FJ}(\mathcal{M}^*) < I_{FJ}(\mathcal{M}^{FJ})$, this indicates that our inferred mixture $\mathcal{M}^*$ results in a lower value of the scoring function that is defined by FJ. Such an evaluation not only demonstrates the superior performance of the proposed search (leading to $\mathcal{M}^*$) using our defined scoring function but also proves it is better compared to the one defined by FJ.

Kullback Leibler (KL) distance: In addition to using the message length based evaluation criterion, the mixtures are also compared using the KL distance (Section 2.5.2) with respect to the original mixture distribution. For a mixture probability distribution, there is no analytical form to compute the metric. However, one can calculate its empirical value (which asymptotically converges to the actual KL distance). In experiments relating to mixture simulations, we know the true mixture $\mathcal{M}^t$ from which the data $\{\mathbf{x}_i\}, 1 \leq i \leq N$ is being sampled. The KL distance, denoted by $D_{KL}(\mathcal{M}^t \,||\, \mathcal{M})$, is then approximated by the following expression

$$D_{KL}(\mathcal{M}^t \,||\, \mathcal{M}) = E_{\mathcal{M}^t}\left[\log \frac{\Pr(\mathbf{x}, \mathcal{M}^t)}{\Pr(\mathbf{x}, \mathcal{M})}\right] \approx \frac{1}{N}\sum_{i=1}^{N} \log \frac{\Pr(\mathbf{x}_i, \mathcal{M}^t)}{\Pr(\mathbf{x}_i, \mathcal{M})} \tag{5.12}$$

where $\mathcal{M}$ is an inferred mixture distribution ($\mathcal{M}^*$ or $\mathcal{M}^{FJ}$) whose *closeness* to the true mixture $\mathcal{M}^t$ is to be determined.

5.5.2 Bivariate Gaussian mixture simulation

An experiment conducted by FJ was to randomly generate $N = 800$ data points from a two-component (with equal mixing proportions) bivariate mixture $\mathcal{M}^t$ whose means are at $\boldsymbol{\mu}_1 = (0, 0)^T$ and $\boldsymbol{\mu}_2 = (\delta, 0)^T$, have equal covariance matrices: $\mathbf{C}_1 = \mathbf{C}_2 = \mathbf{I}$ (the identity matrix), and compare the number of inferred components. The same experiment is repeated here and the results are compared with those of FJ. The separation δ between the means is gradually increased within the range of 1.8 to 2.6 (same as FJ) and the percentage of the correct selections (over 50 simulations) as determined by the two search methods is analyzed.

As expected, an increase in the separation between the component means leads to an increase in the number of correctly inferred components. In this case, for the mixtures inferred using both approaches, the differences in message lengths ΔI_{MML} and ΔI_{FJ} are close to zero. The KL distances for the inferred mixtures are also the same. Therefore, for this experimental setup, the performance of both methods is roughly similar.

As the difference between the two search methods is not apparent from this experiment, the behaviour of the methods was investigated for smaller samples. Hence, the experiment is repeated with $N = 100$. In this case, our search method results in a mean value (of the inferred components) close to 1 for different values of δ (see Figure 5.6a). The average value of the number of inferred components using FJ's method fluctuates between 2 and 3. However, there is significant variance in the number of inferred components as can be seen in Figure 5.6(a). There are many instances where the number of inferred components is more than 3. The results indicate that the FJ's method is overfitting the data.

The correctness of the mixtures inferred by the two search methods is further evaluated by comparisons using the message length formulations and KL distance values. Figure 5.6(b) shows the boxplot

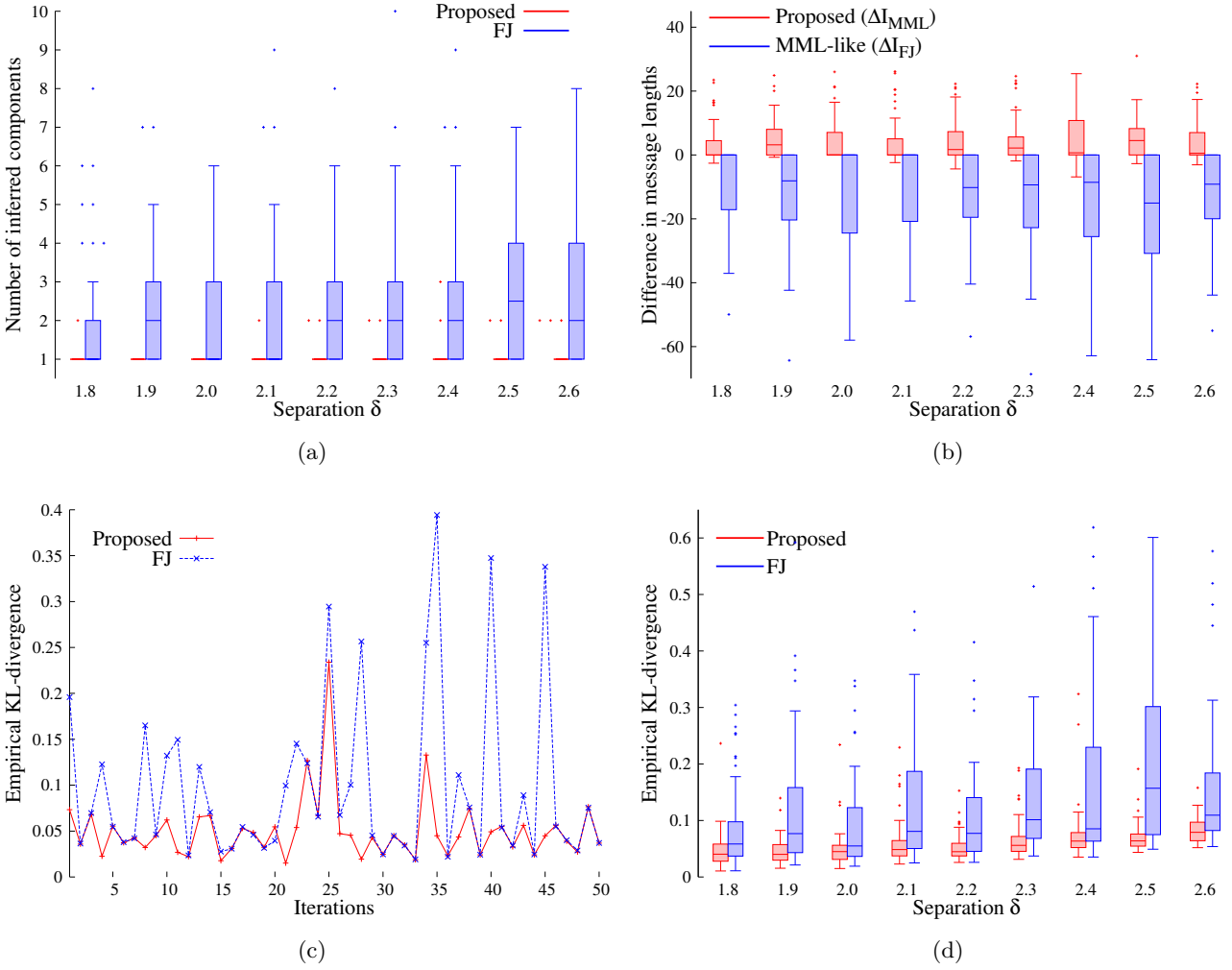


Figure 5.6: Bivariate mixture simulation ($N = 100$ and over 50 simulations). (a) Box-whisker plot showing the variability in the number of inferred components (b) Difference in message lengths computed using the two different scoring functions (see Equation 5.11) (c) KL distance of inferred mixtures when $\delta = 2.0$ (d) KL distance for all values of $\delta \in \{1.8, \ldots, 2.6\}$.

of the difference in message lengths of the mixtures $\mathcal{M}^*$ inferred using the proposed search method and the mixtures $\mathcal{M}^{FJ}$ inferred using FJ's method. It is observed that $\Delta I_{MML} > 0$ across all values of δ for the 50 simulations.

As per Equation 5.11, we have $I_{MML}(\mathcal{M}^*) < I_{MML}(\mathcal{M}^{FJ})$. This implies that $\mathcal{M}^*$ has a lower message length compared to $\mathcal{M}^{FJ}$ when evaluated using our scoring function. Similarly, we have $\Delta I_{FJ} < 0$, that is, $I_{FJ}(\mathcal{M}^*) > I_{FJ}(\mathcal{M}^{FJ})$. This implies that $\mathcal{M}^{FJ}$ has a lower message length compared to $\mathcal{M}^*$ when evaluated using FJ's scoring function. These results are not surprising as $\mathcal{M}^*$ and $\mathcal{M}^{FJ}$ are obtained using search methods that optimize the respective MML and MML-like scoring functions.

The KL distances of $\mathcal{M}^*$ and $\mathcal{M}^{FJ}$ are then analyzed with respect to the true bivariate mixture $\mathcal{M}^t$ over all 50 simulations and across all values of δ. Ideally, the KL distance should be close to zero. Figure 5.6(c) shows the KL distance of the mixtures inferred using the two search methods with respect to $\mathcal{M}^t$ when the separation is $\delta = 2.0$. The proposed search method infers mixtures whose KL distance (denoted by red lines) is close to zero, and more importantly less than the KL distance of mixtures inferred by FJ's search method (denoted by blue lines). The same type of behaviour is noticed with other values of δ. Figure 5.6(d) compares the KL distance for varying values of δ. The median value of the KL distance due to the proposed search method is close to zero with not much variation. FJ's

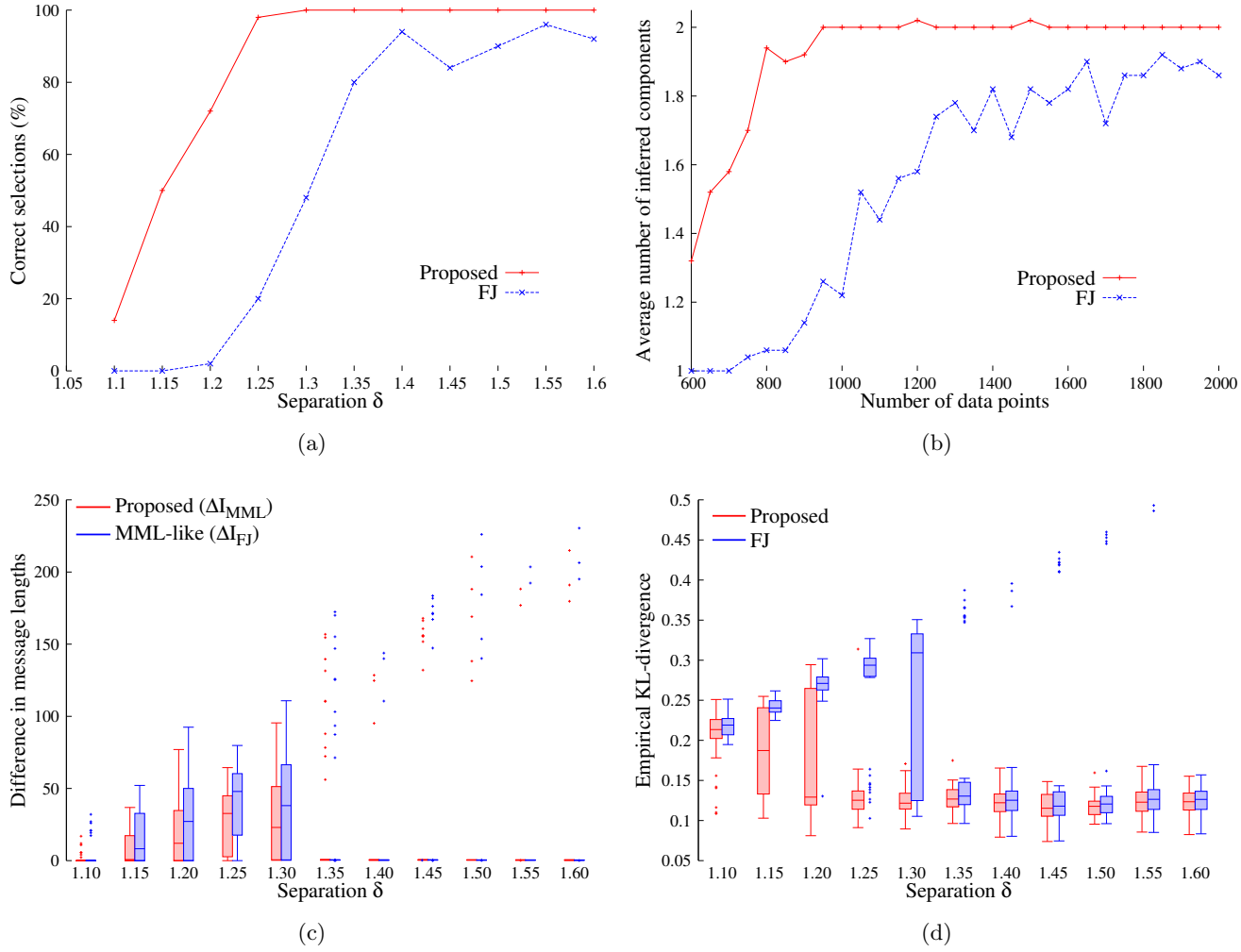


Figure 5.7: 10-dimensional mixture simulations (a) Percentage of correct selections with varying δ for a fixed sample size of $N = 800$ (b) Average number of inferred mixture components with different sample sizes and $\delta = 1.20$ between component means. (c) Difference in message lengths of inferred mixtures (d) Box-whisker plot of KL distance of inferred mixtures

search method always results in mixtures whose KL distance is higher. The results suggest that, in this case, mixtures $\mathcal{M}^{FJ}$ inferred by employing the FJ's search method deviate significantly from the true mixture distribution $\mathcal{M}^t$. This can also be explained by the fact that there is a wide spectrum of the number of inferred components (see Figure 5.6a). This suggests that the MML-like scoring function is failing in its objective to control the trade-off between complexity and goodness-of-fit and, hence, is selecting more complex mixture models than necessary to describe the data.

5.5.3 Simulation of 10-dimensional mixtures

Along the same lines as the previous setup, FJ conducted another experiment for a 10-variate two-component mixture $\mathcal{M}^t$ with equal mixing proportions. The means are at $\boldsymbol{\mu}_1 = (0, \ldots, 0)^T$ and $\boldsymbol{\mu}_2 = (\delta, \ldots, \delta)^T$ so that the Euclidean distance between them is $\delta\sqrt{10}$. The covariances of the two components are $\mathbf{C}_1 = \mathbf{C}_2 = \mathbf{I}$. Random samples of size $N = 800$ were generated from the mixture and the number of inferred components are plotted. The experiment is repeated for different values of δ and over 50 simulations.

Figure 5.7(a) shows the number of inferred components using the two search methods. At lower values of δ, the components are close to each other and, hence, it is relatively more difficult to correctly

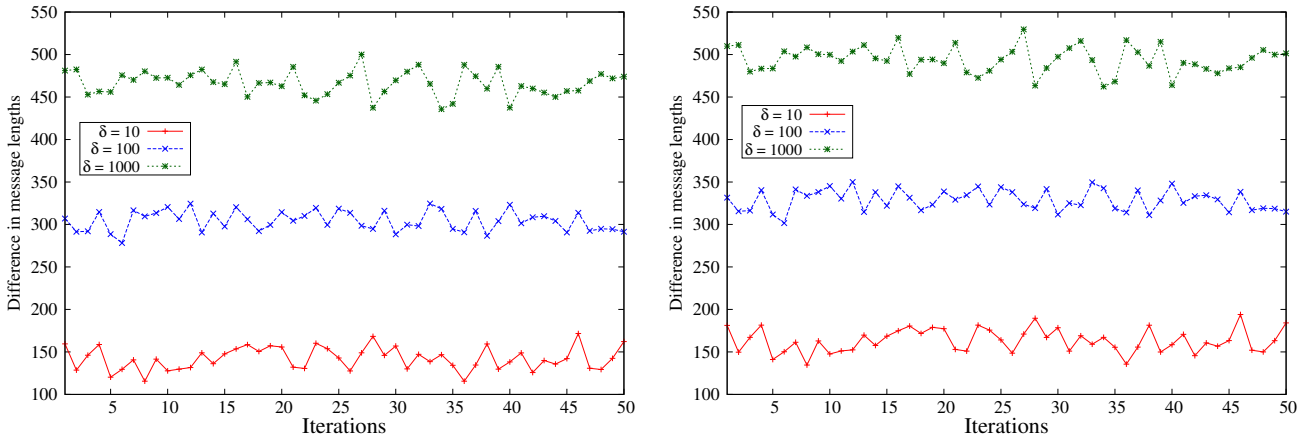
infer the true number of components. It is observed that our proposed method clearly performs better than that of Figueiredo and Jain (2002) across all values of δ. The quality of these inferred mixtures is also compared by calculating the difference in message lengths using the two scoring functions, and also using the KL distance with respect to $\mathcal{M}^t$. For all values of δ, $\Delta I_{MML} > 0$, that is, our inferred mixtures $\mathcal{M}^*$ have a lower message length compared to $\mathcal{M}^{FJ}$ when evaluated using our scoring function. More interestingly, we also note that $\Delta I_{FJ} > 0$ (see Figure 5.7c). This reflects that $\mathcal{M}^*$ have a lower message length compared to $\mathcal{M}^{FJ}$ when evaluated using the scoring function of Figueiredo and Jain (2002). This suggests that their search method results in a sub-optimal mixture $\mathcal{M}^{FJ}$ and fails to infer the better mixture $\mathcal{M}^*$.

In addition to the message lengths, the mixtures are analyzed using the KL distance. Similar to the bivariate example in Figure 5.6(c), the KL distance of our inferred mixtures $\mathcal{M}^*$ is lower than $\mathcal{M}^{FJ}$, the mixtures inferred by Figueiredo and Jain (2002). Figure 5.7(d) shows the boxplot of KL distance of the inferred mixtures $\mathcal{M}^*$ and $\mathcal{M}^{FJ}$. At higher values of $\delta >= 1.45$, the median value of KL distance is close to zero, as the number of correctly inferred components (Figure 5.7a) is more than 90%. However, the proposed method always infers mixtures $\mathcal{M}^*$ with lower KL distance compared to $\mathcal{M}^{FJ}$. These results demonstrate the superior performance of the proposed method.

Another experiment was carried out where $\delta = 1.20$ was held constant (that is, the components are extremely close), gradually increased the sample size N, and plotted the average number of inferred components by running 50 simulations for each N. Figure 5.7(b) shows the results for the average number of inferred components as the amount of data increases. The proposed search method, on average, infers the true mixture when the sample size is ~ 1000. However, FJ's search method requires larger amounts of data; even with a sample size of 2000, the average number of inferred components is ~ 1.9. In Figure 5.7(b), the red curve reaches the true number of 2 and saturates more rapidly than the blue curve.

5.5.4 Discussion of Figueiredo and Jain (2002)'s method

One of the drawbacks associated with FJ's search method is due to the form of the updating expression for the component weights (Equation 5.9). As discussed in Section 5.3, an incorrect inference is bound to happen when the net membership of a (valid) component is less than $N_p/2$, where N_p is the number of free parameters per component. In such a case, the component weight is updated as zero, and is eliminated, effectively reducing the mixture size by one.



(a) ΔI_{MML} (b) ΔI_{FJ}

Figure 5.8: Evaluation of the quality of inferred mixtures by comparing the difference in message lengths as computed using the two scoring functions. Positive difference indicates that the mixtures inferred by our search method have lower message lengths and are better than those of $\mathcal{M}^{FJ}$ (see Equation 5.11).

We conduct the following experiment to demonstrate this behaviour. Consider a two-component, 10-variate mixture, $\mathcal{M}^t$, as before and randomly generate samples of size 50 from the mixture. Since the constituent components of $\mathcal{M}^t$ have equal weights, on average, each component has a membership of about 25. We used the varying values of $\delta = \{10, 100, 1000\}$, so that the two components are well apart from each other in each case. For each δ, we run 50 simulations and analyze the number of inferred components. As expected, FJ's search method infers a mixture with one component regardless of the separation δ. In contrast, our method infers the correct number of components. In order to test the validity of mixtures inferred by our proposed method, we analyze the resultant mixtures as discussed in Section 5.5.1.

Figure 5.8(a) shows the difference in message lengths ΔI_{MML} given in Equation 5.11. We observe that $\Delta I_{MML} > 0$ for all δ. This demonstrates that our inferred mixtures $\mathcal{M}^*$ have lower message lengths than those of mixtures $\mathcal{M}^{FJ}$ using our scoring function. The same phenomenon is observed when using FJ's MML-like scoring function. In Figure 5.8(b), we observe that $\Delta I_{FJ} > 0$, which means our search based mixtures $\mathcal{M}^*$ have lower message lengths compared to mixtures $\mathcal{M}^{FJ}$ when evaluated using their scoring function. This demonstrates that $\mathcal{M}^*$ is a better mixture as compared to $\mathcal{M}^{FJ}$, irrespective of the scoring function used to evaluate the mixture, and FJ's search method is unable to infer it.

We also note that the differences in message lengths increases with increasing δ. This is because for the one-component inferred mixture $\mathcal{M}^{FJ}$, the second part of the message (see Equation 5.8) which corresponds to the negative log-likelihood term increases because of poorer fit to the data. The two modes in the data sampled from $\mathcal{M}^t$ become increasingly pronounced as the separation between the components in the true mixture increases and, hence, modelling such a distribution using a one-component mixture results in a fit that becomes progressively worse. This is clearly an incorrect inference. We further strengthen the case for the superiority of our proposed method by comparing the KL distance of the inferred mixtures $\mathcal{M}^*$ and $\mathcal{M}^{FJ}$ with respect to $\mathcal{M}^t$. Figure 5.9 illustrates these results. As δ increases, the blue coloured plots shift corresponding to the mixtures $\mathcal{M}^{FJ}$ inferred by FJ's method shift higher. The proposed search method, however, infers mixtures $\mathcal{M}^*$ with lower KL distance. The figure indicates that the inferred mixtures $\mathcal{M}^*$ are more similar to the true distribution as compared to the mixtures $\mathcal{M}^{FJ}$.

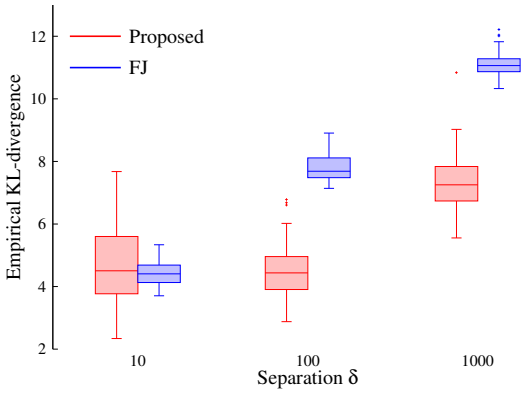
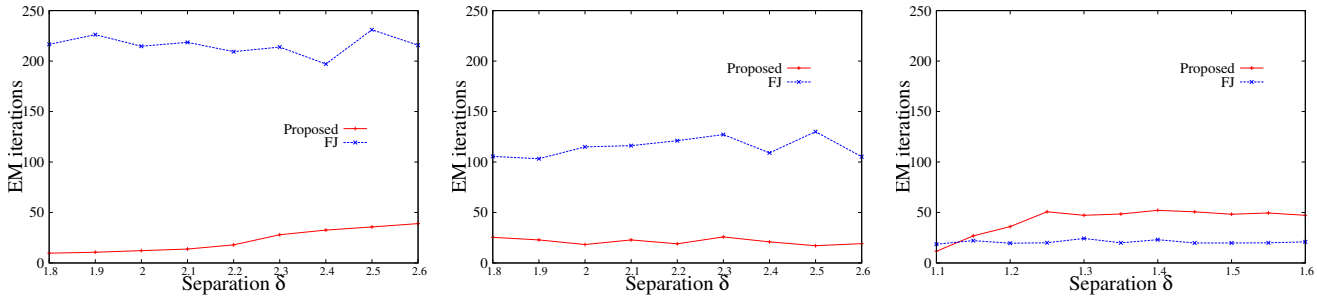


Figure 5.9: Box-whisker plot of KL distance of mixtures inferred by the two search methods. A random sample of size $N = 50$ is generated for each δ and this is repeated 50 times.

5.5.5 Analysis of the computational cost

At any intermediate stage of our search procedure, a *current* mixture with K components requires K number of split, delete, and merge operations before it is updated. Each of the perturbations involve performing an EM to optimize the corresponding mixture parameters. To determine the convergence of the EM algorithm, a threshold of 10^{-5} is used to terminate the EM method, which was the same as used by FJ in their experiments. The FJ's method also requires to start with a large number of components. We set this to 25 based on what was suggested in FJ. We investigate the number of times the EM routine is called and compare it with that of FJ's results.

The analysis corresponds to the simulations that were carried out previously. For the bivariate mixture discussed in Section 5.5.2, the number of resulting EM iterations when the sample sizes are $N = 800$ and $N = 100$ are shown in Figures 5.10(a) and (b), respectively. As per the discussion in Section 5.5.2, at $N = 800$, the average number of components inferred by the two methods are about the same. However, the number of EM iterations required by FJ's method is greater than 200 across



(a) Bivariate mixture ($N = 800$) (b) Bivariate mixture ($N = 100$) (c) 10-variate mixture ($N = 800$)

Figure 5.10: Number of EM iterations performed during the mixture simulations discussed in Sections 5.5.2 and 5.5.3.

all values of δ (see Figure 5.10a). In contrast, the proposed method, on average, requires fewer than 50 iterations. In this case, both methods infer similar mixtures with FJ's method requiring more number of EM iterations. When the bivariate mixture simulation is carried out using $N = 100$, the number of EM iterations required by FJ's method, on average, is greater than 100, while the proposed method requires fewer than 40 iterations (see Figure 5.10b). In this case, the proposed method not only infers better mixtures (as discussed in Section 5.5.2) but is also conservative with respect to the computational cost.

For the simulation results corresponding to the 10-variate mixtures (Section 5.5.3), our proposed method requires close to 50 iterations on average, while FJ's method requires about 20 (see Figure 5.10c). This is because our proposed method further perturbs the inferred two-component mixture through a series of split, delete, and merge operations and each of them internally uses the EM algorithm. However, the mixtures inferred by the proposed method fare better when compared to that of FJ (see Figure 5.7). Furthermore, for the simulation results (see Section 5.5.4), FJ's method stops after 3 EM iterations because their method terminates prematurely by inferring a one-component mixture. This is because their program does not accommodate components when the memberships are less than $N_p/2$. Our method requires 18 EM iterations on average and infers the correct mixture components. In these two cases, our method infers better quality mixtures, with no significant computational overhead.

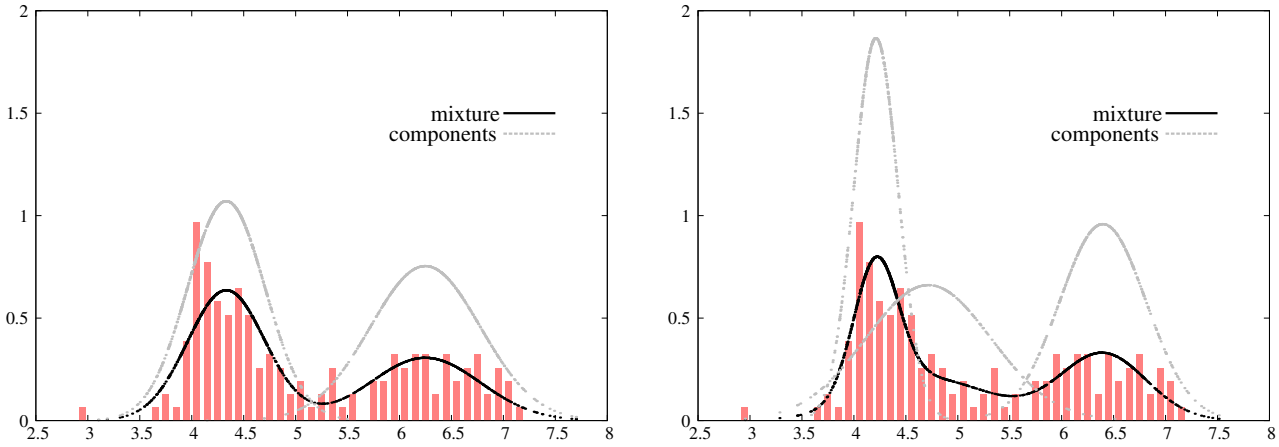
These experiments demonstrate that the performance of our proposed search is better than the widely used FJ's method. We compared the resulting mixtures using both the proposed MML formulation and FJ's MML-like formulation, showing the advantages of the former over the latter. We also used a neutral metric, KL distance, to establish the similarity of our inferred mixtures to the true distributions.

5.5.6 Applications to real-world data

Acidity data set (Richardson and Green, 1997; McLachlan and Peel, 1997)

The proposed search and inference methodology is applied on two real-world data sets. The first example is the univariate *acidity* data set which contains 155 points. The proposed search method infers a mixture $\mathcal{M}^*$ with 2 components whereas the search method of Figueiredo and Jain (2002) infers a mixture $\mathcal{M}^{FJ}$ with 3 components. The inferred mixtures are shown in Figure 5.11 and their corresponding parameter estimates are given in Table 5.1. In order to compare the mixtures inferred by the two search methods, the message lengths of the inferred mixtures are computed using the proposed complete MML and the approximated MML-like (FJ's) scoring functions.

When these mixtures are evaluated using the proposed MML scoring function, our inferred mixture results in a gain of about 4 bits (see Table 5.2). Based on the MML framework, our two-component mixture $\mathcal{M}^*$ is 2^4 times more likely than the three-component mixture $\mathcal{M}^{FJ}$ (as per Equation 2.5).



(a) $\mathcal{M}^*$ inferred by the proposed search method.

(b) $\mathcal{M}^{FJ}$ inferred by FJ's search method.

Figure 5.11: Mixtures inferred by the two search methods using the acidity data set. Y-axis shows the mixture probability density. (See Table 5.1 for the corresponding parameter estimates.)

Furthermore, when the inferred mixtures are evaluated using the MML-like (FJ's) scoring function, $\mathcal{M}^*$ is still evaluated as better ($\sim$ 298 bits) than $\mathcal{M}^{FJ}$ ($\sim$ 320 bits). Thus, as per both the scoring functions, $\mathcal{M}^*$ is the better mixture model of this data set.

Table 5.1: The parameters of the inferred mixtures shown in Figure 5.11

Component index	Weight	Parameters (μ, σ^2)
1	0.41	6.24, 0.28
2	0.59	4.33, 0.14

(a) Proposed

Component index	Weight	Parameters (μ, σ^2)
1	0.34	6.39, 0.17
2	0.35	4.21, 0.05
3	0.31	4.71, 0.36

(b) FJ

Table 5.2: Message lengths (measured in bits) of the mixtures (in Figure 5.11) as evaluated using the MML and MML-like scoring functions.

	Inferred mixtures	
Scoring functions	Proposed $(\mathcal{M}^*)$	FJ $(\mathcal{M}^{FJ})$
MML	**1837.61**	1841.69
MML-like	**298.68**	320.02

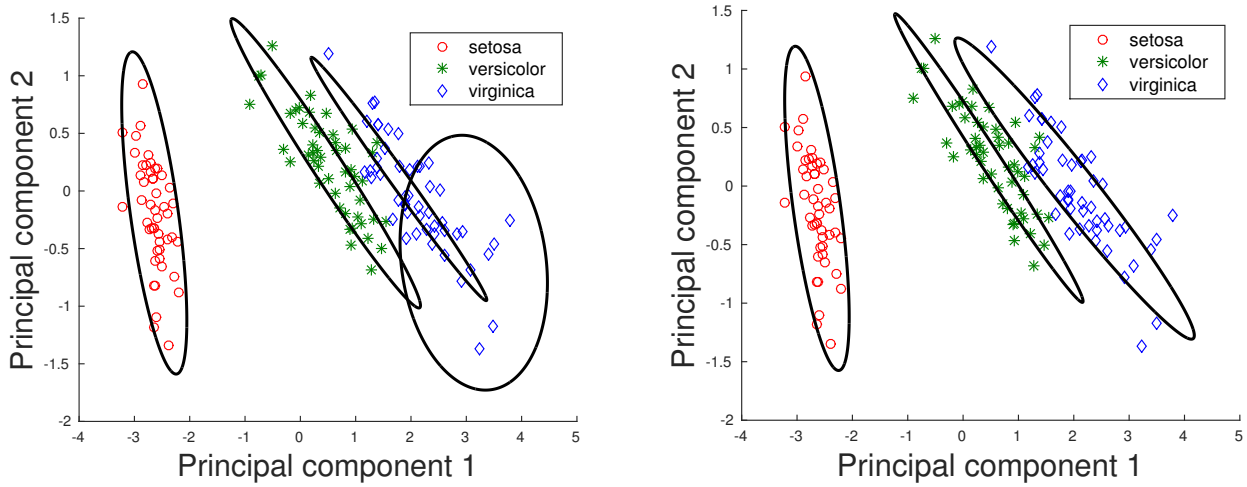
Iris data set (Anderson, 1935; Fisher, 1936)

The second example is the popular Iris data set. The data is 4-dimensional and comes from three Iris species namely, *Iris-setosa, Iris-versicolor,* and *Iris-virginica.* The data size is 150 with each class (species) comprising 50 representative elements.

The proposed search method infers a 4-component mixture $\mathcal{M}^*$ and the search method of Figueiredo and Jain (2002) infers a 3-component mixture $\mathcal{M}^{FJ}$ (see Figure 5.12). Table 5.3 shows the memberships of the 150 elements in each of the components in the inferred mixtures. We notice an additional component M4 in $\mathcal{M}^*$ which has a net membership of 9.51, that is, $\sim$ 6% of the entire data set. It

appears that the component M2 in $\mathcal{M}^{FJ}$ (Table 5.3b) is split into two components M2 and M4 in $\mathcal{M}^*$ (Table 5.3a).

The quality of the inferred mixtures is determined by comparing their message lengths using the MML and MML-like scoring functions. Table 5.4 shows the values obtained using the two formulations. When evaluated using our complete MML formulation, our inferred mixture $\mathcal{M}^*$ results in extra compression of about 1 bit, which makes it twice as likely as $\mathcal{M}^{FJ}$ – it is a closely competing model compared to ours. When evaluated using the FJ's MML-like scoring function, our inferred mixture continues to have a lower message length. In both cases, the mixture $\mathcal{M}^*$ inferred by our search method is preferred.



(a) $\mathcal{M}^*$ inferred by the proposed search method.

(b) $\mathcal{M}^{FJ}$ inferred by FJ's search method.

Figure 5.12: Mixtures inferred by the two search methods using the Iris data set. The data is projected onto the leading two principal components.

Table 5.3: Memberships of Iris data as using the inferred mixtures in Figure 5.12 (a) Distribution of data using $\mathcal{M}^*$ (b) Distribution of data using $\mathcal{M}^{FJ}$

Species	M1	M2	M3	M4
setosa	50	0	0	0
versicolor	0	5.64	44.36	0
virginica	0	40.29	0.20	9.51

(a) Data distribution using 4 components

Species	M1	M2	M3
setosa	50	0	0
versicolor	0	5.55	44.45
virginica	0	49.78	0.22

(b) Data distribution using 3 components

Table 5.4: Message lengths (measured in bits) of the mixtures (in Figure 5.12) as evaluated using the MML and MML-like scoring functions.

	Inferred mixtures	
Scoring functions	Proposed ($\mathcal{M}^*$)	FJ ($\mathcal{M}^{FJ}$)
MML	**6373.01**	6374.27
MML-like	**323.31**	342.57

5.6 Summary

In this chapter, we have proposed a search method to infer an optimal mixture in a completely unsupervised setting for modelling the given data. As noted in Section 5.3, inferring a suitable mixture requires balancing the trade-off due to the mixture model's complexity and its goodness-of-fit to the data. We discussed the various methods that aim to resolve this trade-off. However, the existing methods are based on approximations of the mixture's complexity and hence, do not address the trade-off satisfactorily (see Section 5.3). To rectify those drawbacks, we outlined a comprehensive MML formulation that is used to evaluate competing mixtures and select the better one.

Furthermore, unlike the previous MML-based methods, we proposed a search strategy that progressively alters the mixture size by Split, Delete, and Merge perturbations (see Section 5.4). Of all the perturbations, we select the one that results in the greatest improvement to the total message length. By doing so, we are making sure that the new mixture is an improved version of the current one. We perform an exhaustive search by considering the operations on all components in the mixture. These different operations give a mixture the best chance to escape a local optimum. The strategic operations minimize the possibility of the EM algorithm resulting in sub-optimal mixture parameters.

In order to better motivate the search method, we presented an example illustrating the inference of a Gaussian mixture. We explained how the different operations lead to different stages in the evolution of the mixture model. We obtain various intermediate mixtures that are sub-optimal as part of the search process. However, the search method perturbs these sub-optimal mixtures to eventually converge to an improved mixture model (see Section 5.4.3). We have tested the proposed approach against the widely cited method of Figueiredo and Jain (2002).

Figueiredo and Jain (2002) proposed a method that begins with a large number of components and iteratively eliminates components with no provision to recovering a component if it is deleted in error. As we highlighted in Section 5.3.6, their method is based on an approximated MML criterion that essentially ignores the encoding of the components' parameters and hence, leads to a simplifying expression to quantify the complexity of a mixture model. We have conducted a thorough analysis comparing our proposed method to that of Figueiredo and Jain (2002). We have demonstrated that our proposed method results in better mixtures in a variety of contexts, as explained in Section 5.5. The evaluation of the two search methods is carried on both simulated and real-world data. We published the search method to infer optimal mixtures in Kasarapu and Allison (2015).

Although we discussed the method in the context of inference of multivariate Gaussian mixtures, our method can be generalized to accommodate probability distributions whose parameters can be estimated using the Wallace and Freeman (1987) approximation. We demonstrate this in the context of inference of mixtures of directional probability distributions for modelling protein directional data in the next chapter.

Chapter 6

Mixture modelling of directional distributions

6.1 Introduction

The previous chapter presented a general purpose mixture modelling method to infer mixture components and their parameters. The method was explained in the context of modelling using Gaussian mixtures. While Gaussian mixtures are ubiquitous in nature, mixture modelling using other probability distributions have also been widely used (McLachlan and Peel, 2000). This chapter extends our search and method proposed in the previous chapter to directional probability distributions. In particular, we discuss mixtures of the multivariate vMF (mixture modelling on the unit hypersphere), 3D FB_5 (mixture modelling on the 3D unit sphere), and BVM (mixture modelling on the 3D torus) distributions.

The importance of vMF and FB_5 distributions in mixture modelling tasks has been well established: vMF mixtures have been used in large-scale text clustering (Banerjee et al., 2003; Gopal and Yang, 2014), clustering of protein dihedral angles (Dowe et al., 1996a; Mardia et al., 2007), and gene expression analyses (Banerjee et al., 2005). Mixtures of FB_5 distributions have been employed by Peel et al. (2001) to identify joint sets in rock masses, and by Hamelryck et al. (2006) to sample random protein conformations. The FB_5 distribution has increasingly found support in machine learning tasks for structural bioinformatics (Kent and Hamelryck, 2005; Boomsma et al., 2006; Hamelryck, 2009). The mixtures of BVM distributions have been used in modelling protein dihedral angles (Dowe et al., 1996a; Mardia et al., 2008).

For the directional distributions considered, we adapt our generalized search method (discussed in Section 5.4) by modifying the split operation to cater for the respective directional probability distribution. Note that since the directional distributions are defined on the surfaces of a sphere and a torus, the split operation explained in the case of Gaussian mixtures (defined in the Euclidean space) is not directly applicable.

The rest of the chapter is organized as follows: Section 6.2 first details the changes in our search method needed to infer multivariate vMF mixtures. It then studies the performance of our proposed search method by presenting empirical analyses of mixture modelling using vMF distributions under varying scenarios. It then applies the vMF mixtures in modelling high-dimensional text data that is normalized to lie on the unit hypersphere. We have demonstrated the ability of our search method to infer mixtures of text documents in a completely unsupervised setting as opposed to the existing work of Banerjee et al. (2005).

Section 6.3 outlines the mixture modelling apparatus using the FB_5 distributions. We illustrate the progression of the various stages of our search method through an example. We then apply the FB_5 mixtures in modelling protein directional data. For comparison, we also model the same data using vMF mixtures. In both cases, we demonstrate the ability our search method to infer components that are biologically meaningful. The resulting vMF and FB_5 mixtures are demonstrated to be superior

models as opposed to the naïve uniform model (Konagurthu et al., 2012) and, therefore, serve as efficient descriptors of protein directional data. Furthermore, we show that FB_5 mixtures supersede the vMF mixtures in their ability to better explain the protein directional data.

Section 6.4 discusses the search and inference of mixtures of bivariate von Mises (BVM) distributions. As a specific application, we employ the mixtures to model protein dihedral angles. As explained in Section 4.4, the BVM Sine distribution has the ability to model correlated data. In order to demonstrate the utility of the BVM Sine distributions, we provide a comparison with mixtures of BVM Independent distributions that are not ideal to model correlated data. As expected, the results of mixture modelling indicate the ability of the Sine mixtures to efficiently describe the empirical distribution of protein dihedral angles. We demonstrate that our search method is able to infer meaningful clusters that directly correspond to frequently occuring conformations in protein structures.

6.2 Mixtures of multivariate von Mises-Fisher distributions

In this section, we use our search method (proposed in Section 5.4) to infer mixtures of multivariate vMF distributions. To achieve this, that is, to infer the optimal number of mixture components, the mixture modelling apparatus that was explained in the case of multivariate Gaussian mixtures is modified to cater for the directional data distributed on the surface of a unit hypersphere.

The strategic split, delete, and merge operations that are part of the search method are tailored to vMF mixtures. In each operation, the EM algorithm is used to estimate the mixture parameters (Section 5.2.2). In the M-step of the EM algorithm, the parameters of the vMF components are updated using their respective MML estimates. Recall in Section 4.2.2, we described the MML estimation of the parameters of the vMF distribution. The update rule to estimate the parameters for Gaussian mixtures given by Equation 5.7 is modified to obtain the MML estimators of the vMF mean (Equation 4.11), and the vMF concentration parameter, which is obtained by minimizing the message length expression given by Equation 4.12.

For the j^{th} vMF component in the mixture, the resultant vector sum is updated as $\mathbf{R}_j(t+1) = \sum_{i=1}^{N} r_{ij}(t)\mathbf{x}_i$, where $r_{ij}(t)$ is an element of the responsibility matrix at iteration t (Section 5.2.2). If $R_j(t+1)$ represents the magnitude of vector $\mathbf{R}_j(t+1)$, then the updated mean is given by the following equation.

$$\widehat{\boldsymbol{\mu}}_j(t+1) = \frac{\mathbf{R}_j(t+1)}{R_j(t+1)}$$

The MML update of the concentration parameter $\widehat{\kappa}_j(t+1)$ is obtained by solving $G(\widehat{\kappa}_j(t+1)) = 0$ after substituting $N \to n_j(t)$ and $R \to R_j(t+1)$ in Equation 4.13.

As part of the search method, in the split operation detailed in Section 5.4.2, we need to initialize the component means of the two child components. Recall that in the case of splitting a Gaussian mixture (parent) component, we computed the direction of maximum variance and identified two points that are one standard deviation away on either side of the parent mean. These correspond to the initial means of the children. The computation of the children means is feasible in this manner for Gaussian components. However, in the case of a multivariate vMF parent component, this approach cannot be readily used as the directional data is on the surface of a hypersphere and not in the Euclidean plane. Hence, while splitting a multivariate vMF component, we simply assign the means randomly. In this context, we explain another splitting strategy tailored for data distributed on the *three-dimensional* unit sphere in Section 6.3.1.

The delete and merge operations are carried out in the same spirit as those described in Section 5.4.2. During merging vMF components, the closest pair is identified by computing the KL distance. In the case of vMF distributions, an analytical form for the KL distance is derived in Appendix B.3.

6.2.1 Experimental analyses of vMF mixtures

In this section, we use the search method described above to infer mixtures of vMF distributions and perform an empirical evaluation. The experimental setup involves generating data randomly from a pre-defined mixture distribution, and then employing our proposed search method to infer the optimal number of mixture components. The amount of data that is sampled is progressively increased in our experiments. For each sample size N, the simulation is repeated 50 times and the number of inferred components is plotted.

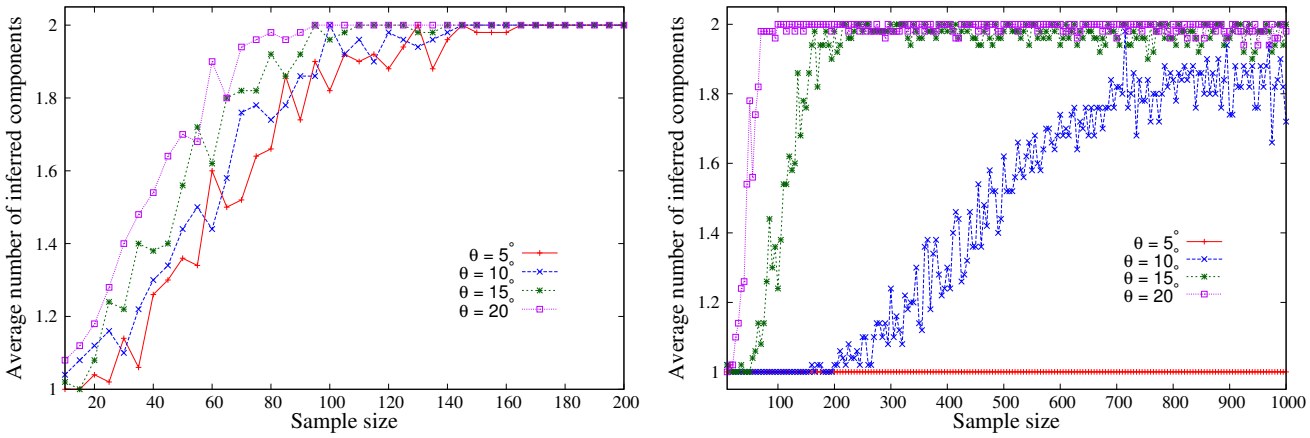
The search method is analyzed assuming that the original mixtures from which the data is generated have one of the following two characteristics

1. The true components in the mixture have *different* mean directions (separated by an angle θ).

2. The true components in the mixture have the *same* mean direction but different concentration parameters.

The MML estimates of the parameters of a vMF distribution have been derived and explained in Section 4.2.2. The vMF concentration parameter κ is estimated using the MML-Halley approximation (Equation 4.15). These estimates are used in the EM steps of the mixture modelling method. The various case studies are discussed below.

Components have different means: The true mixture has two components with equal mixing proportions. We consider the case when the dimensionality is $d = 3$. The mean of one of the vMF components is aligned with the Z-axis. The mean of the other component is chosen such that the angle between the two means is θ degrees. Figure 6.1(a) illustrates the scenario when the concentration parameters of the constituent components are different. Figure 6.1(b) shows the variation in the number of inferred components when the true vMF components have the same concentration parameter. In both scenarios, as the angular separation is progressively increased, the components become more distinguishable and, hence, lesser amount of data is required to accurately identify them.



(a) $\kappa_1 = 10$, $\kappa_2 = 100$ (b) $\kappa_1 = \kappa_2 = 100$

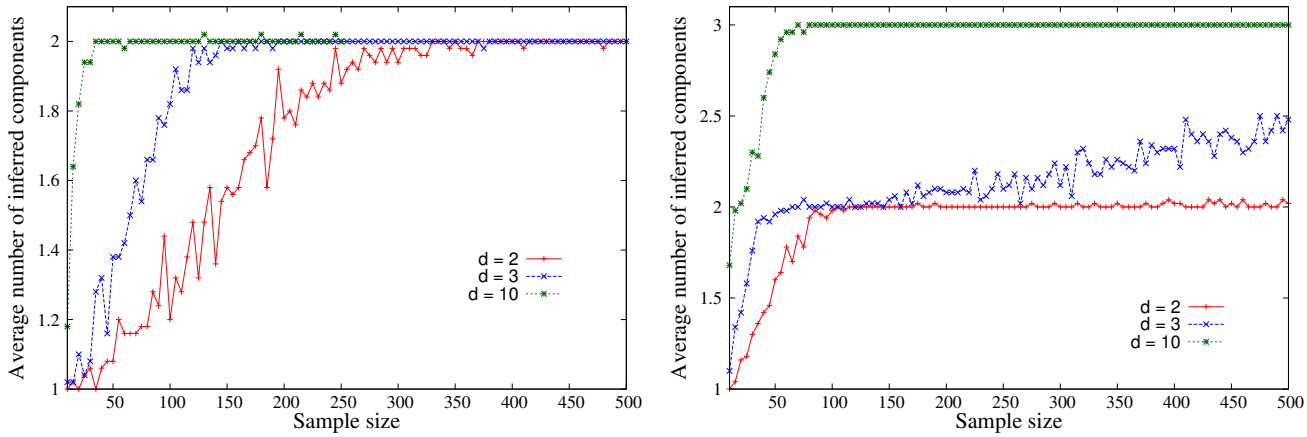
Figure 6.1: Average number of components inferred for the two-component mixture whose means are separated by θ degrees.

When the concentration parameters of the constituent components are different (Figure 6.1a), the inference of the mixture is relatively easier compared to the case when the concentration parameters are same (Figure 6.1b). In Figure 6.1(a), for all angular separations, the true number of components is correctly inferred at a sample size of $N = 200$. When $\theta = 20°$, the search method converges faster at $N \sim 100$ as compared to $\theta = 5°$, when the convergence is at $N \sim 160$. In Figure 6.1(b), when $\theta = 5°$,

the search method infers only one-component as the true mixture components are hardly distinguishable. When $\theta = 10°$, even at $N \sim 1000$, the average number of inferred components is ~ 1.8. When $\theta = 15°$, the search method converges at $N \sim 300$ as compared to $N \sim 120$ in Figure 6.1(a). Clearly, when the component means are different, it is easier to correctly infer mixtures whose components have different concentration parameters.

Components have the same mean but different concentration parameters: In this case, multivariate ($d = \{2, 3, 10\}$) vMF mixtures with equal mixing proportions and same component means are considered. The simulation results of true mixtures with two and three components are presented here.

Figure 6.2(a) shows the average number of components inferred for a two-component mixture whose concentration parameters are $\kappa_1 = 10$ and $\kappa_2 = 100$. For each value of d, as the sample size increases, the average number of inferred components saturates to the true value (2 in this case). Increasing the sample size beyond this does not impact the number of inferred mixture components.



(a) 2-component mixture

(b) 3-component mixture

Figure 6.2: Average number of components inferred when the true mixture has components with the same mean direction but different concentration parameters. (a) 2-component mixture ($\kappa_1 = 10, \kappa_2 = 100$) (b) 3-component mixture ($\kappa_1 = 10, \kappa_2 = 100, \kappa_3 = 1000$)

The results for a 3-component mixture with identical means but different concentration parameters $\kappa_1 = 10, \kappa_2 = 100$, and $\kappa_3 = 1000$ are shown in Figure 6.2(b). As expected, the average number of inferred components increases in the light of more data. However, it clearly requires greater amount of data to correctly infer the three mixture components as compared to the two-component case. In the two-component case (Figure 6.2a), at around $N = 450$, all three curves converge to the right number of components. For the three-component mixture in Figure 6.2(b), there is no convergence for $d = 2, 3$ noticeable until $N = 500$. For $d = 10$, the average number of inferred components converges much faster for the 2-component mixture ($N \sim 25$) than for the 3-component mixture ($N \sim 100$).

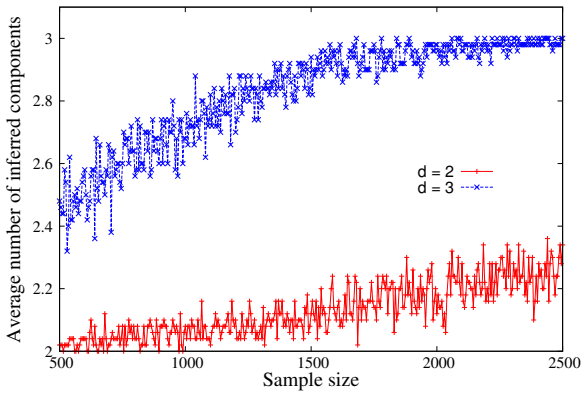


Figure 6.3: Gradual increase in the average inferred components for the 3-component mixture in Figure 6.2(b).

It is also interesting to note that for $d = 2$ in Figure 6.2(b), the average number of inferred components appears to saturate at 2, while the actual number of mixture components is 3. However, as the amount

of available data increases, the (almost) horizontal line shows signs of gradual increase in its slope. Figure 6.3 shows the increase in the average number for $d = 2, 3$ as the data is increased beyond $N = 500$. The blue curve representing $d = 3$ stabilizes at $N \sim 2500$. However, the red curve ($d = 2$) slowly starts to move up but the average is still well below the true number. This demonstrates the relative difficulty in estimating the true mixture when the means coincide, especially at lower dimensions.

As shown above, when the means coincide, greater amount of data is needed to accurately infer the true mixture components. Further, the amount of data needed also depends on the dimensionality in consideration. It appears that as the dimensionality increases, we need smaller amounts of data (as can be seen for the $d = 10$ case). In d-dimensional space, each datum comprises of d real values with the constraint that it should lie on the *unit* hypersphere. So, the estimation of the mean direction requires the estimation of $(d-1)$ values and one value for κ. When there is data of size N, we actually have $n_d = N \times (d-1)$ values available for estimating the d free parameters. For instance, given a sample of size N, for $d = 2, n_2 = N$ and for $d = 10, n_{10} = 9N$. We conjecture that this could be a possible reason for faster convergence in high dimensional space. Through these experiments done so far, the ability of the proposed search method to infer appropriate mixtures is demonstrated in situations with varying difficulty levels.

6.2.2 Application to text clustering

The use of vMF mixtures in modelling high dimensional text data has been investigated by Banerjee et al. (2005). Computing the similarity between text documents requires their representation in some vector form. The elements of the vectors are typically a function of the word and document frequencies in a given collection. These vector representations are commonly used in clustering textual data with cosine based similarity metrics being central to such analyses (Strehl et al., 2000). There is a strong argument for transforming the vectors into points on a unit hypersphere (Salton and McGill, 1986; Salton and Buckley, 1988). Such a normalized representation of text data (that compensates for different document lengths) motivates their modelling using vMF mixture distributions.

Banerjee et al. (2005) proposed an approximation (Equation 4.5) to estimate the parameters of a mixture with a *known* number of components. They did not, however, propose a method to search for the optimal number of mixture components. In contrast, we not only derive the MML estimates and show they fare better than the previous approximations (see Chapter 4), but also use them to devise a search method that infers the optimal mixtures (see Section 5.4). Ideally, the search is continued until there is no further improvement in the message length (Algorithm 1). For practial purposes, the search is terminated when the improvement due to the intermediate split, delete and merge operations during the search process is less than 0.01%.

In order to determine the optimal number of components, Banerjee et al. (2005) vary the number of mixture components and compute the mutual information (MI) to assess the quality of clustering for each mixture. For given cluster assignments X and the (known) class labels Y, MI is defined as $E\left[\log \dfrac{\Pr(X,Y)}{\Pr(X)\Pr(Y)}\right]$. In addition to the message length criterion, we also use the MI criterion to compare the quality of the inferred mixtures.

Further, the actual class labels of the data can be compared against the labels predicted as per the cluster assignment. This is done using a confusion matrix, which is a tabulation of the number of true/false positives/negatives. Using the tabulated numbers, we can compute the classification error in terms of precision and recall. Precision is the proportion of the correctly classified cases in the total data. Recall is the proportion of the correctly classified points in each class. The error of classification is then computed using the F-measure, which is the harmonic mean of the precision and recall of the classification model, where a good model is one with a high F-measure. The average F-measure is calculated when the number of inferred clusters is the same as the number of actual classes.

For each of the data sets, in the preprocessing step, feature vectors are generated for each document in the collection by computing a score for each word. The score is a function of the frequency of

occurrence of the word in the document (term frequency, TF) and the inverse document frequency (IDF), that is, the number of documents which contain the word. One such scoring function that combines the TF and IDF values is the Okapi BM25 score (Robertson and Zaragoza, 2009), which is typically used in quantifying the relevance of a word in a document, in the context of information retrieval. These feature vectors are then normalized to generate unit vectors in some d-dimensional space. Using this as directional data on a hypersphere, a suitable mixture model was inferred using the search algorithm proposed in Section 5.4.

Classic3 dataset

This dataset[1] contains documents from three distinct categories: 1398 Cranfield (aeronautical related), 1033 Medline (medical journals) and 1460 Cisi (information retrieval related) documents. The processed data has $d = 4358$ features.

Optimal number of clusters: While this data set is known to have three distinct categories, this information is not usually known in real world setting (and we do not know if they are from three vMF distributions). Assuming no knowledge of the nature of the data, the search method infers a mixture with 16 components. The corresponding assignments are shown in Table 6.1. A closer look at the generated assignments illustrates that each category of documents is represented by more than one component. The three categories are split to possibly represent specialized sub-categories. The Cisi category is distributed among 6 main components (M4 - M9). The Cranfield documents are distributed among M6, M10 - M15 components and the Medline category is split into M0 - M3, and M6 components. It is observed that all but three components are non-overlapping; only M6 has representative documents from all three categories.

Table 6.1: Confusion matrix for 16-component assignment (MML-Halley).

	M0	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12	M13	M14	M15
cisi	0	0	4	0	288	133	28	555	197	255	0	0	0	0	0	0
cran	0	0	0	0	2	0	362	1	0	0	58	144	135	175	223	298
med	9	249	376	138	2	0	9	0	0	0	0	0	0	0	0	0

The 16-component mixture inferred by the search method is a finer segregation of the data when compared to modelling using a 3-component mixture. The parameters of the 3-component mixture are estimated using the EM algorithm of Section 5.2.2. Table 6.2 shows the confusion matrices obtained for the cluster assignments using the various estimation methods. We see that all the estimates perform comparably with each other; there is not much difference in the assignments of data to the individual mixture components.

Table 6.2: Confusion matrices for 3-cluster assignment. (Sra's confusion matrix is omitted as it is same as that of Tanabe)

	cisi	cran	med
cisi	1441	0	19
cran	22	1293	83
med	8	0	**1025**

(a) Banerjee

	cisi	cran	med
cisi	1449	0	11
cran	24	1331	43
med	13	0	1020

(b) Tanabe

	cisi	cran	med
cisi	**1450**	0	10
cran	24	**1339**	35
med	14	0	1019

(c) Song

	cisi	cran	med
cisi	**1450**	0	10
cran	24	1331	43
med	13	0	1020

(d) MML (Halley)

The collection is comprised of documents that belong to dissimilar categories and, hence, the clusters obtained are wide apart. This can be seen from the extremely high F-measure scores (Table 6.3). For the 3-component mixture, all five different estimates result in high F-measure values with Song being the best with an average F-measure of 0.978 and an MI of 0.982. MML-Halley's estimate is

[1] `http://www.dataminingresearch.com/index.php/2010/09/classic3-classic4-datasets/`

close with an average F-measure of 0.976 and an MI of 0.976. However, based on the message length criterion, the MML estimate results in the least message length ($\sim$ 190 bits less than Song's). The mutual information score using MML estimate is 1.04 (for 16 components) compared to 0.976 for 3 components. Also, the message length is lower for the 16-component case. However, Song's estimator results in a MI score of 1.043, very close to the score of 1.040 obtained using MML estimates.

Table 6.3: Clustering performance on Classic3 dataset.

Number of clusters	Evaluation metric	Banerjee	Tanabe	Sra	Song	MML (Halley)
3	Message length (bits)	100678069	100677085	100677087	100677080	**100676891**
	Avg. F-measure	0.9644	0.9758	0.9758	**0.9780**	0.9761
	Mutual Information	0.944	0.975	0.975	**0.982**	0.976
16	Message length (bits)	100458153	100452893	100439983	100444649	**100427178**
	Mutual Information	1.029	1.036	0.978	**1.043**	1.040

For the Classic3 dataset, Banerjee et al. (2005) analyzed mixtures with greater numbers of components than the "natural" number of clusters. They report that a 3-component mixture is not necessarily a good model and more number of clusters may be preferred for this example. As part of their observations, they suggest to "generate greater number of clusters and combine them appropriately". However, this is subjective and requires some background information about the likely number of clusters. The proposed search method, in conjunction with the inference framework, is able to resolve this dilemma and determine the optimal number of mixture components in a completely unsupervised setting.

CMU_Newsgroup

This dataset[2] contains documents from 20 different news categories each containing 1000 documents. Preprocessing of the data, as discussed above, resulted in feature vectors of dimensionality $d = 6448$. The data is first modelled using a mixture containing 20 components. The evaluation metrics are shown in Table 6.4. The average F-measure is 0.509 for MML-based estimation, slightly better than Banerjee's score of 0.502. The low F-measure values are indicative of the difficulty in accurately distinguishing the news categories. The mutual information score for the MML case is 1.379, which is lower than that of Sra's. However, the total message length is lower for the MML mixture when compared to that of others.

Optimal number of clusters: The proposed search method when applied to this dataset infers a mixture with 21 components. This is close to the "true" number of 20 (although there is no strong reason to believe that each category corresponds to a vMF component). The mutual information for the 21-cluster assignment is highest for Sra's mixture with a score of 1.396 and for MML mixture, it is 1.375 (Table 6.4). However, the total message length is the least for the MML-based mixture.

Table 6.4: Clustering performance on CMU_Newsgroup dataset.

Number of clusters	Evaluation metric	Banerjee	Tanabe	Sra	Song	MML (Halley)
20	Message length (bits)	728666702	728545471	728585441	728536451	**728523254**
	Avg. F-measure	0.502	0.470	0.487	0.435	**0.509**
	Mutual Information	1.391	1.383	**1.417**	1.244	1.379
21	Message length (bits)	728497453	728498076	728432625	728374429	**728273820**
	Mutual Information	1.313	1.229	**1.396**	1.377	1.375

The analysis of vMF mixtures by Banerjee et al. (2005) for both the datasets considered here shows a continued increase in the MI scores even beyond the true number of clusters. As such, using the mutual information score for different number of mixture components does not aid in the inference

[2] `http://archive.ics.uci.edu/ml/datasets/Twenty+Newsgroups`

of an optimal mixture model. Our proposed search method balances the trade-off between using a certain mixture and its ability to explain the observed data and, thus, objectively aids in inferring mixtures to model the normalized vector representations of a given collection of text documents.

Potential extensions of the search method: A mixture modelling problem of the kind where there is some information available regarding the nature of the data can be studied by altering our proposed generic search method. In such scenarios, some alternate strategies where the mixture modelling can be done in a semi-supervised setting are provided below.

- The priors on the number of components and their parameters can be modelled based on the background knowledge.
- If the true number of clusters are known, only splits may be carried out until we near the true number (each split being the best one given the current mixture). As the mixture size approaches the true number, all the three operations (split, delete, and merge) can be resumed until convergence. This increases the chance of the inferred mixture having about the same number of components as the true model.
- Another variant could be to start from a number close to the true number and prefer delete/merge operations over the splits. The split operations cannot be ignored completely because a component after splitting may be merged at a later iteration if there would be an improvement to the message length.
- Another strategy could be to employ the EM algorithm and infer a mixture with the true number of components. This mixture can then be perturbed using split, delete, and merge operations until the perturbations do not improve the message length.

6.3 Search and inference of mixtures of 3D Kent distributions

Let us now extend the search method described in Section 5.4 to infer mixtures of Kent distributions. To infer the optimal number of mixture components, the mixture modelling apparatus is modified to cater to the directional data distributed on the surface of a *three-dimensional* (3D) sphere. Specifically, the split operation detailed in Section 5.4.2 is modified as described below. The procedure for deleting and merging the components is same as that in the case of Gaussian or vMF mixtures. During merging FB_5 components, the KL distance is evaluated as shown in Appendix B.3. Further, in all the operations, the MML estimators of the FB_5 distribution are used in the update step of the EM algorithm (Section 5.2.2).

6.3.1 Splitting a mixture component of a directional distribution

As part of the split operation, we need to generate two child components from a parent component. This is done in order to check the suitability of modelling the data using a mixture with a greater number of components. The children are obtained by an EM algorithm carried on the data distribution corresponding to the parent. In order to ensure that the child components are distinct, during the EM initialization of the children, their initial components are chosen to lie on either side of the parent mean along the direction of maximum variance. This gives the children the best chance to escape from the local optimum (corresponding to the parent component). This splitting strategy is, however, specific to Gaussian mixtures.

A strategy akin to splitting Gaussian components is not directly applicable to directional distributions. For multivariate vMF distributions, the initial means are selected randomly, an approach used in vMF mixture modelling (Section 6.2). However, an alternative strategy can be proposed as explained below.

Selection of initial means of the two child components: The procedure is explained in the case of distributions defined on the surface of a 3D sphere, but can be generalized to higher dimensions as well (for example, multivariate vMF mixtures). The procedure for moment estimation of the major and minor axes of an FB_5 distribution involves the eigenvalue decomposition of the matrix $\mathbf{B}_L$, the submatrix derived from the dispersion matrix $\mathbf{B}$ (see Section 4.3.2). If l_1 and l_2 are the eigenvalues of $\mathbf{B}_L$ (see Equation 4.20), then l_1, l_2 are roots of the characteristic equation

$$l^2-(b_{22}+b_{33})l+b_{22}b_{33}-b_{23}^2=0 \quad \text{so that} \quad l_1+l_2=b_{22}+b_{33}$$

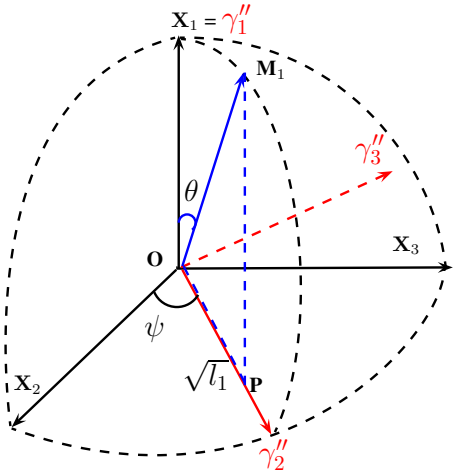


Figure 6.4: Initializing the means of the child FB_5 components while splitting the parent.

According to Equation 4.22, we have $l_1 - l_2 = r_2$, and hence, $l_1 = (b_{22}+b_{33}+r_2)/2$. The maximum variance is along the direction of major axis and is equal to the eigenvalue l_1. Hence, one standard deviation would correspond to $\sqrt{l_1}$. It is to be noted that these calculations are done in the $\mathbf{X}_2\mathbf{X}_3$ plane (see Section 4.3.1) which contains the major and minor axes (that is, after the mean of the parent, as part of moment estimation, is aligned with $\mathbf{X}_1$). However, it is now required to map this point back onto the *unit* sphere.

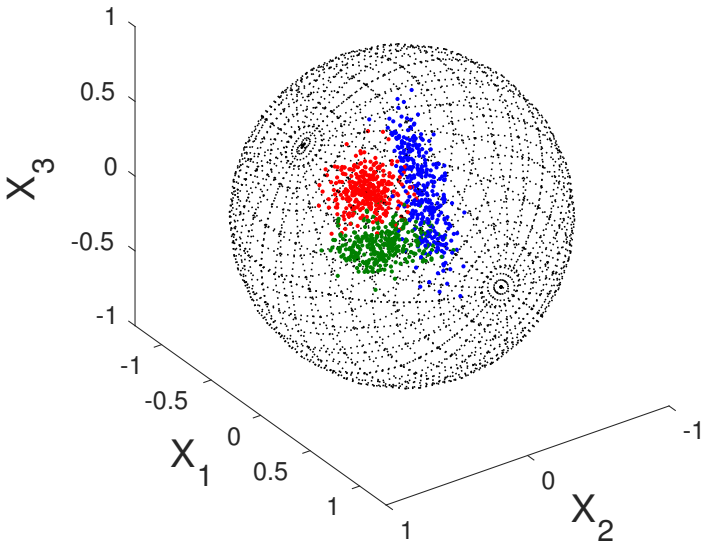
Consider Figure 6.4 where $\boldsymbol{\gamma}_2''$ and $\boldsymbol{\gamma}_3''$ are the major and minor axes in the $\mathbf{X}_2\mathbf{X}_3$ plane respectively (see Figure 4.3). The mean axis $\boldsymbol{\gamma}_1''$ of the parent component being split is aligned with $\mathbf{X}_1$. The segment OP has length $\sqrt{l_1}$ corresponding to a unit standard deviation along $\boldsymbol{\gamma}_2''$. Let M_1 be the mean of one of the children. Then, for M_1 such that M_1P is perpendicular to the $\mathbf{X}_2\mathbf{X}_3$ plane, we have $M_1P = \sqrt{1-l_1}$ (as $OM_1 = 1$ is the radius of the sphere). If $\theta \in [0, 180°]$ measures the co-latitude of the mean M_1 as shown, we have $\theta = \arccos\sqrt{1-l_1}$. The mean M_2 (not shown in the figure) of the second child component lies in the plane containing OM_1P such that the angle between OM_2 and OX_1 is θ. The two means are then transformed in order to conform with the axes of the parent FB_5 component. With these as starting points for the EM algorithm, the two child components are locally optimized. The children along with the untouched $(K-1)$-components, then, serve as a starting point for estimating the parameters of the $(K+1)$-component mixture using the EM algorithm.

6.3.2 Illustrative example of the search procedure

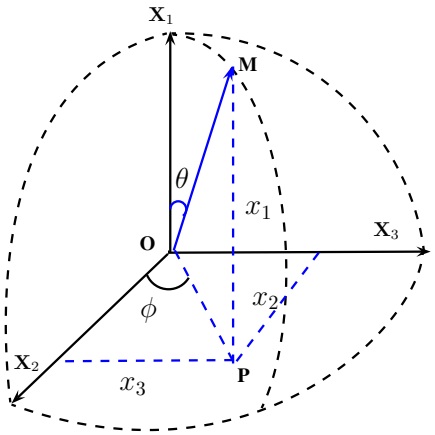
The mechanics of the inference of a suitable FB_5 mixture model hves been explained previously in Sections 5.4 and 6.3.1. To better illustrate the search process, in the case of FB_5 distributions, this section presents a detailed example.

Consider a mixture with three FB_5 components (Figure 6.5a) that have equal mixing proportions, the same concentration parameter $\kappa = 100$ but different eccentricities. The red component has eccentricity $e = 0.1$ and the angular parameters defining its axes are $(\psi, \alpha, \eta) = (0, 60°, 45°)$. The green component has $e = 0.5$ and $(\psi, \alpha, \eta) = (150°, 45°, 30°)$. The blue component has $e = 0.9$ and $(\psi, \alpha, \eta) = (30°, 45°, 60°)$. The parameters are chosen such that the components are close to each other. A sample of size $N = 1000$ was generated from the mixture using the method of Kent et al. (2013).

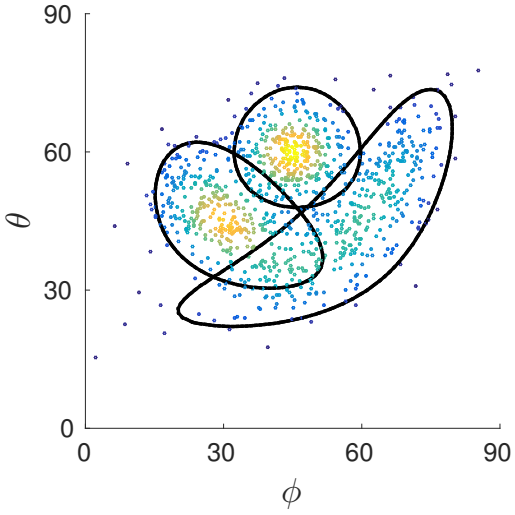
For ease of visualization, the density is represented in the (θ, ϕ)-space, where θ is the co-latitude and ϕ is the longitude (Figure 6.5b). The Cartesian coordinates of each datum M given by $\mathbf{x} = (x_1, x_2, x_3)^T$ in the sampled data are transformed into the spherical coordinates defined by unit radius, co-latitude,



(a)



(b)



(c)

Figure 6.5: Original mixture with equal component weights and $\kappa = 100$. (a) individual components with varying eccentricities: $e = 0.1$ (red), 0.5 (green), and 0.9 (blue), (b) transformation between the spherical and Cartesian coordinate system, and (c) data plotted in degrees in the (θ, ϕ)-space (contours encompass 90% of the data distribution).

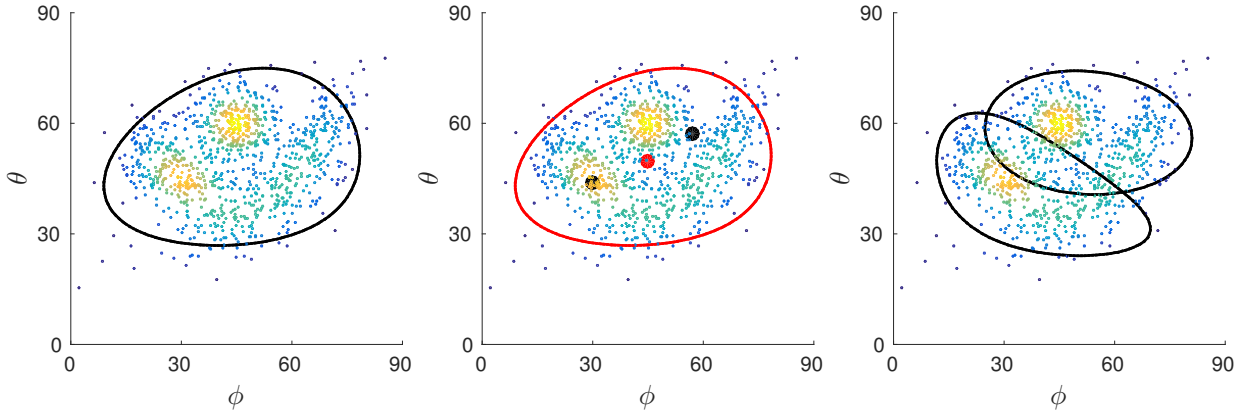
and longitude:[3] $x_1 = \cos\theta, \quad x_2 = \sin\theta\cos\phi, \quad x_3 = \sin\theta\sin\phi$. The resulting mixture density is shown as a heat map in Figure 6.5(c).

The search method explained

The search begins by inferring a one-component mixture $\mathcal{M}_1$ (Figure 6.6a). It has an associated message length of $I = 19364$ bits. Before splitting the component, the means of the children are initialized as shown in Figure 6.6(b). These means are determined as explained in Section 6.3.1. The children are optimized using the EM algorithm to generate the two-component mixture $\mathcal{M}_2$ (Figure 6.6c). $\mathcal{M}_2$ has a message length of $I = 19319$ bits and, hence, improves $\mathcal{M}_1$ by 45 bits.

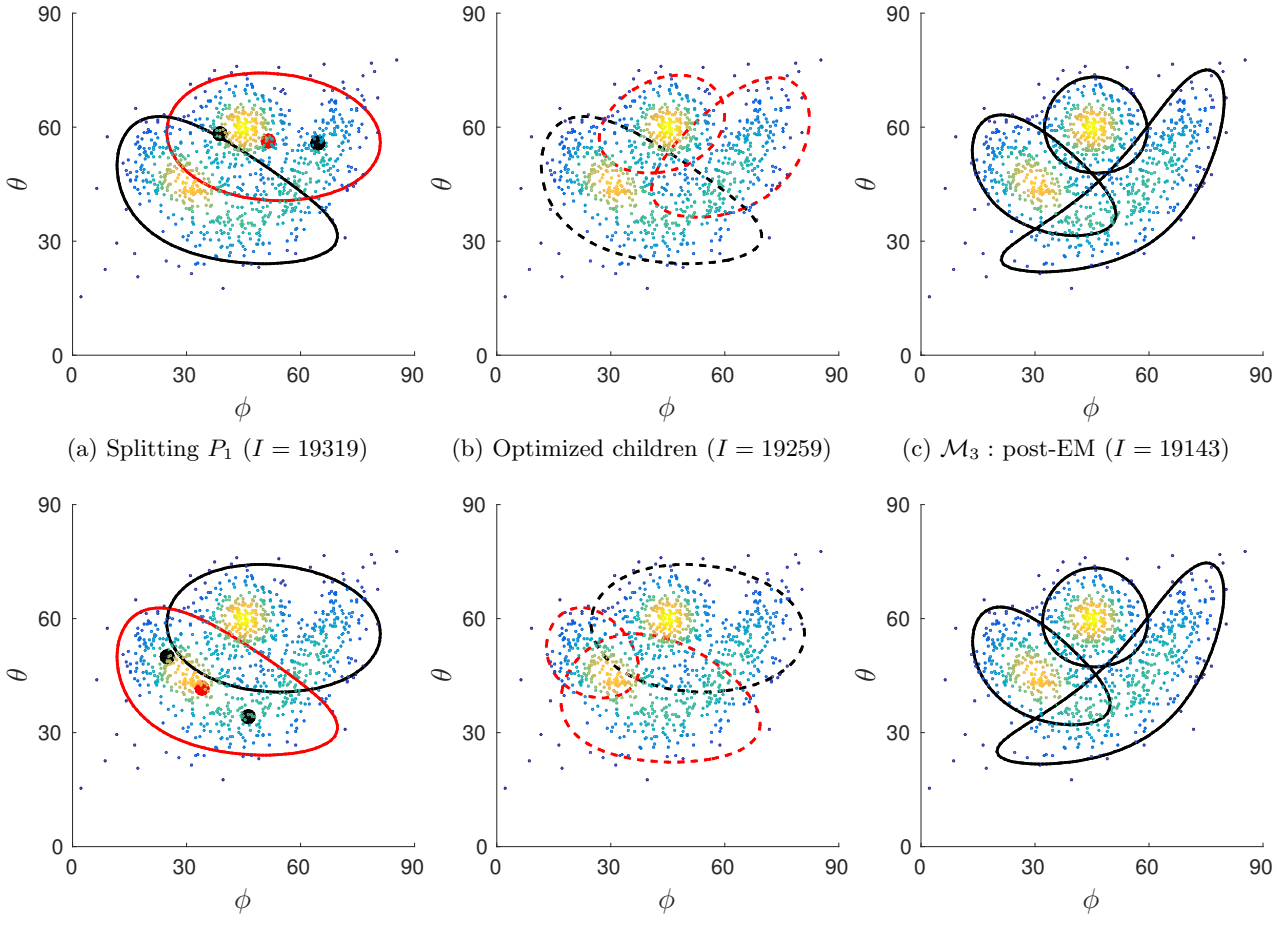
[3]It is to be noted that transforming data generated from an FB_5 distribution (that has elliptical contours on the spherical surface) and representing in the (θ, ϕ)-space produces shapes that do not have any decipherable pattern as can be seen through this example.

In the second iteration, each of the two components in $\mathcal{M}_2$ are split, deleted, and merged. Figure 6.7(a)-(c) illustrates the splitting of component P_1. After integrating the optimized children and



(a) $\mathcal{M}_1$ ($I = 19364$ bits) (b) Initialization of means (c) $\mathcal{M}_2$ ($I = 19319$ bits)

Figure 6.6: Iteration 1 (a) one-component mixture, (b) Red colour denotes the parent component being split, the red dot indicates the mean of the parent and the black dots (on either side) indicate the initial means of the children, (c) improved mixture.



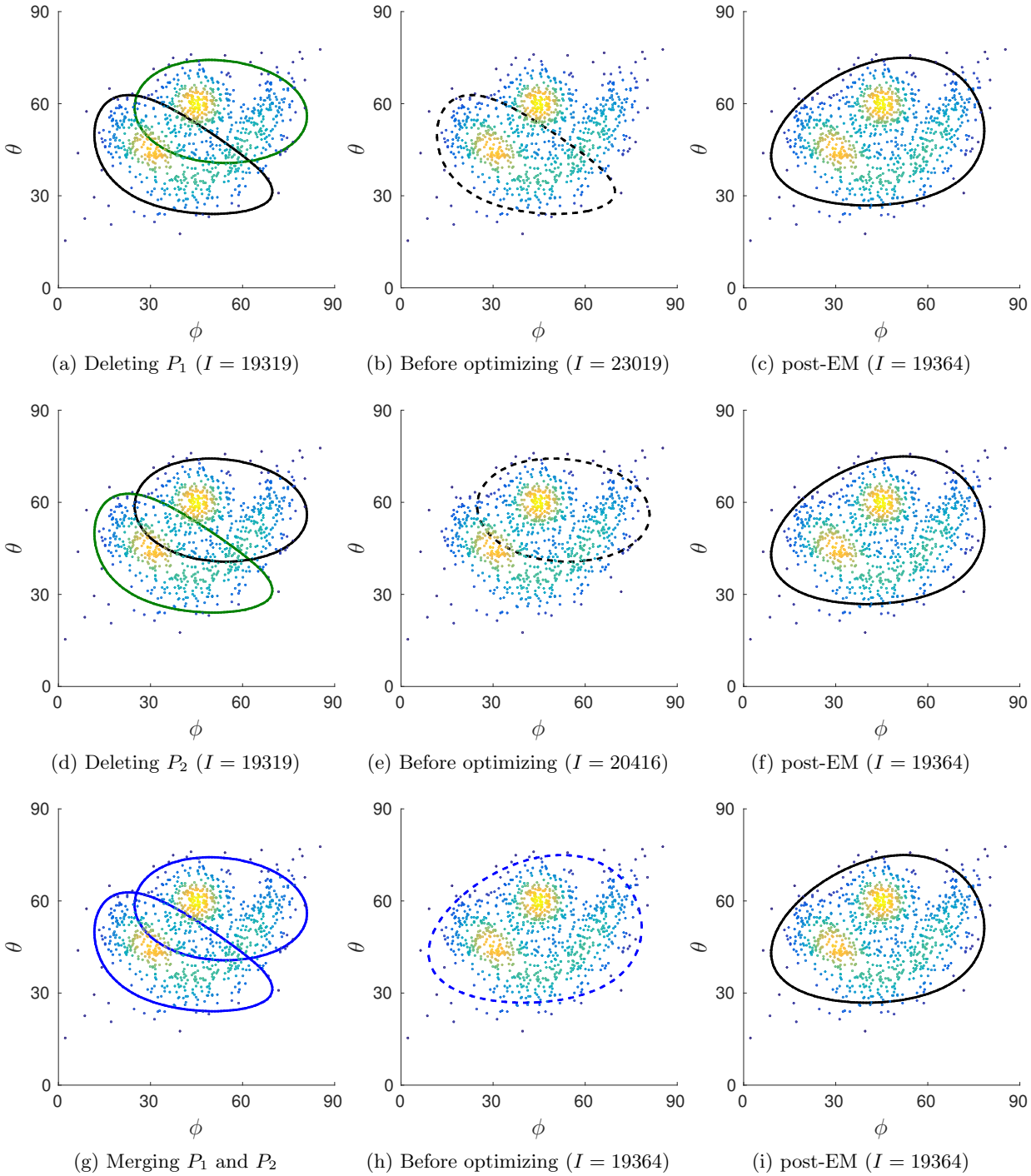
(a) Splitting P_1 ($I = 19319$) (b) Optimized children ($I = 19259$) (c) $\mathcal{M}_3$: post-EM ($I = 19143$)

(d) Splitting P_2 ($I = 19319$) (e) Optimized children ($I = 19333$) (f) $\mathcal{M}_3$: post-EM ($I = 19143$)

Figure 6.7: Iteration 2 – *Split* operations. (a)-(c) splitting the first component P_1 in $\mathcal{M}_2$, and (d)-(f) splitting the second component P_2 in $\mathcal{M}_2$. The red dashed lines in (b),(e) represent the optimized children (prior to integration), and black dashed lines in (b),(e) represent the unchanged components.

subsequently optimizing the resulting 3-component mixture using an EM algorithm, an improved mixture $\mathcal{M}_3$ is obtained. Figure 6.7(d)-(f) illustrates the splitting of component P_2. In this case, splitting P_2 results in the same 3-component mixture $\mathcal{M}_3$. It is to be noted that while splitting P_1 and P_2 produce different intermediate states, as shown in Figure 6.7(b) and (e), the EM converges to the same optimal state in these cases.

Figure 6.8(a)-(f) illustrate the deletion of P_1 and P_2. While their deletions also have different intermediate starting points, as shown in Figure 6.8(b) and (e), the EM algorithm results in the same



(a) Deleting P_1 ($I = 19319$) (b) Before optimizing ($I = 23019$) (c) post-EM ($I = 19364$)

(d) Deleting P_2 ($I = 19319$) (e) Before optimizing ($I = 20416$) (f) post-EM ($I = 19364$)

(g) Merging P_1 and P_2 (h) Before optimizing ($I = 19364$) (i) post-EM ($I = 19364$)

Figure 6.8: Iteration 2 – *Deletions* and *Merging* (green colour represents the component being deleted and blue the pair being merged).

sub-optimal state (same as $\mathcal{M}_1$). As this one-component mixture has a greater message length than that of $\mathcal{M}_2$, the deletion operations do not result in improved mixtures. The merging of P_1 and P_2 components, as shown in Figure 6.8(g)-(i), also does not improve on $\mathcal{M}_2$. Hence, after the second iteration, it is observed that amongst all perturbations, the splitting of P_1 or P_2 results in an improved mixture $\mathcal{M}_3$.

In the third iteration, all perturbations are carried out exhaustively. Figure D.1 (in the Appendix) depicts the splitting, deletion, and merging of one of the three components (P_1) in $\mathcal{M}_3$. Observe, during splitting, the initial selection of means of the child components. The procedure outlined in Section 6.3.1 faithfully separates the two children and results in a mixture with a greater number of components. However, in this case, the optimized mixture $\mathcal{M}_4$ (Figure D.1c) does not improve the message length. Similarly, the deletion of P_1 does not lead to an improved mixture (Figure D.1f). While merging P_1, KL divergence is used to determine an appropriate candidate that is closest. Accordingly, the pair is selected (Figure D.1g) which also does not result in an improved mixture (Figure D.1i). The other two components in $\mathcal{M}_3$ are also perturbed similarly. However, the operations do not result in an improvement.

Progression of the two-part message length

The progression of the various states of the mixture model during the search process is explained in terms of the two-part message length (Equation 5.5). While increasing the number of mixture components leads to increased mixture complexity, the goodness-of-fit to the data improves. The first part of the message corresponds to the encoding of the mixture parameters (number of components, weights, and constituent components' parameters). The second part mainly corresponds to the description of the data using the given mixture model.

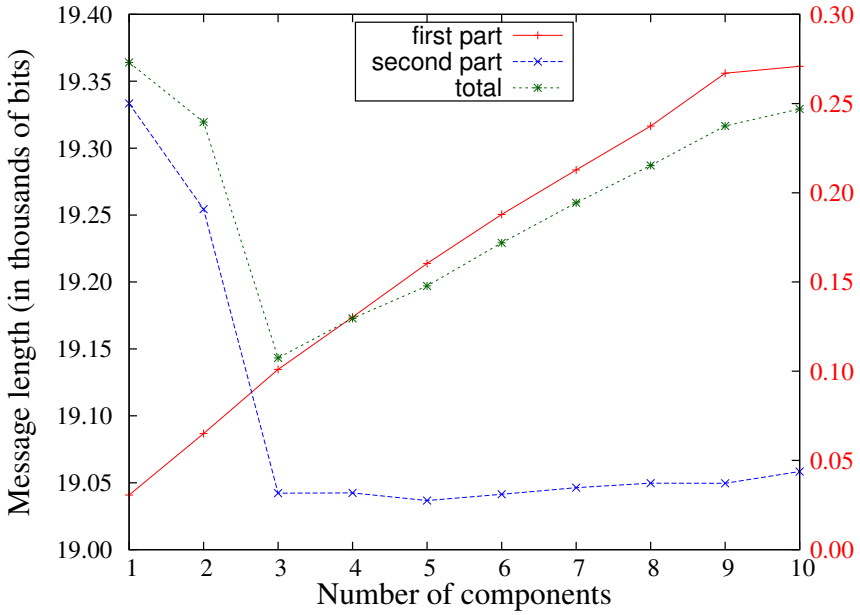


Figure 6.9: Variation of the individual parts of the total message length with increasing number of components (note that the two Y-axes have different scales: the first part of the message follows the right side Y-axis; while the second part of the message and total message lengths follow the left side Y-axis)

In the previous example, the search method infers three components and terminates thereafter. The message lengths corresponding to the optimal mixtures obtained during the associated search process are plotted in Figure 6.9. It is observed that, until $K = 3$, the total message length (green curve) decreases. The variation of the message length is examined beyond the inferred number of

components. For this, starting from $K = 4$ until $K = 10$, we estimated the mixture parameters using the EM algorithm (Section 5.2.2) for each value of $K > 3$. The results indicate that the total message length steadily increases beyond $K = 3$. The reason is that although the negative log-likelihood of the data decreases with increasing K, the second part of the message (the blue curve) only changes marginally, while the first part continues to increase. Thus, as mixtures become overly complex, there is a greater cost associated with encoding their parameters. This affects the total message length as the minimal gain in negative log-likelihood is overshadowed by the increase in the first part of the message. Hence, this example demonstrates the effectiveness of the search method in the context of FB_5 distributions. Furthermore, it also demonstrates the ability of the MML criterion to balance the tradeoff between the model complexity and the goodness-of-fit to the data.

6.3.3 Mixture modelling of protein coordinate data

The applicability of high-dimensional vMF mixtures in real-world context has been previously demonstrated in Section 6.2.2. In this section, the applicability of mixtures of the vMF and FB_5 distributions is demonstrated in the context of modelling experimental 3D protein structural data.

A protein structure consists of a chain of amino acid residues that forms the protein backbone. We consider the protein structure so that the arrangement of atoms along the protein main chain begins with a Nitrogen (N) atom and ends with a Carbon (C) atom (referred to as N- to C-terminus). Each residue along this backbone is identified by a central carbon atom denoted by C_α (see Figure 6.10a). The structures that proteins adopt are largely dictated by the chemical interactions between these constituent atoms. These chemical interactions impose constraints on the orientation of atoms with respect to one another due to which the distance between consecutive C_α atoms is highly constrained to be 3.8 Å. Due to this property, the coordinates of any given C_α can be expressed in a canonical fashion in the context of its preceding C_α atoms.

The almost constant distance between the main chain carbon atoms motivates their modelling using directional probability distributions. The directional data under consideration correspond to the orientations of C_α atoms in the corpus of known protein structures. Models of this directional data have applications to structural modelling tasks such as generating random protein chain conformations, three-dimensional protein structure alignment, secondary structure assignment, and representing protein folding patterns using concise protein fragments. These tasks rely on efficient encoding of protein data (Konagurthu et al., 2012, 2013; Collier et al., 2014). As part of the results, and compared to the vMF mixtures, it is demonstrated that FB_5 mixtures offer a better means of encoding and can potentially serve as strong candidate models to be used in such varied tasks.

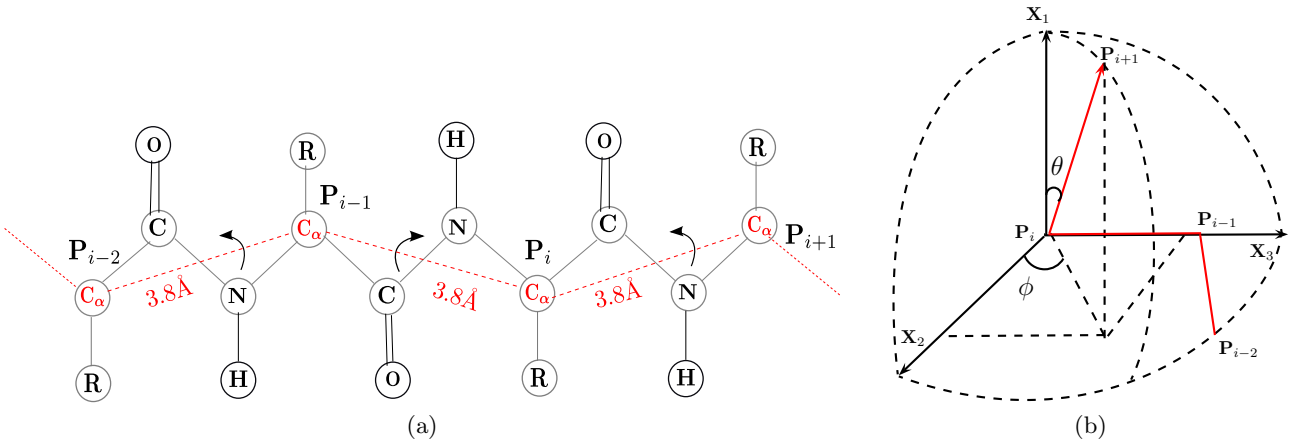


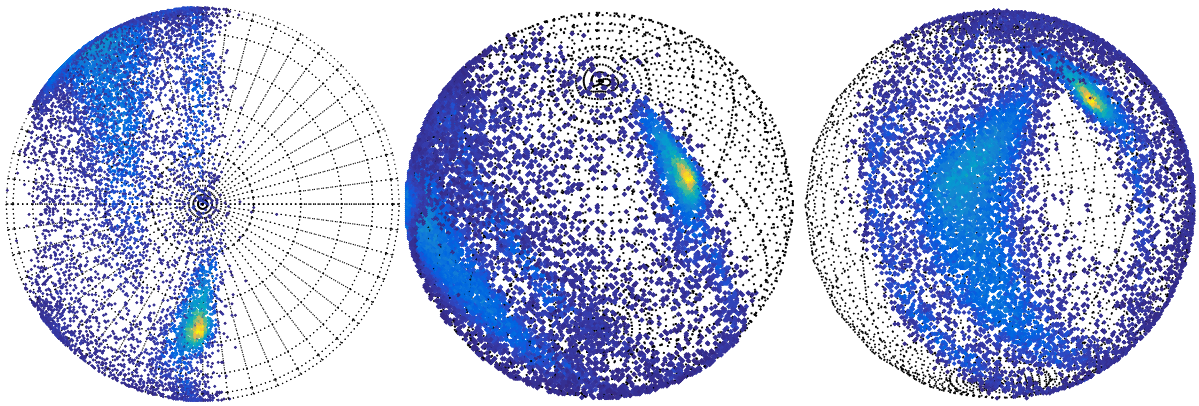
Figure 6.10: Canonical orientation used to generate the directional data (θ, ϕ) corresponding to the protein C_α coordinates.

The data set considered here is a collection of 1802 non-redundant experimentally determined protein structures from the popular and publicly available ASTRAL SCOP-40 (version 1.75) database (Murzin et al., 1995) representing the "β class" proteins. For each protein structure, the coordinates of the central carbon $C_\alpha$of successive residues are considered. Protein coordinate data is transformed into directional data and each direction vector is characterized by (θ, ϕ) = (co-latitude, longitude), where $\theta \in [0, 180°]$ and $\phi \in [0, 360°)$. Note that these (θ, ϕ) values have to be measured in a consistent, canonical manner. To compute (θ, ϕ) corresponding to the point P_{i+1} associated to residue $i+1$, this point is considered in the context of the three preceding points, forming a 4-mer comprising of the points $\mathbf{P}_{i-2}, \mathbf{P}_{i-1}, \mathbf{P}_i$, and $\mathbf{P}_{i+1}$. This 4-mer is orthogonally transformed into a canonical orientation (Figure 6.10b) in the following steps

- translate the 4-mer so that P_i is at the origin.
- rotate the resultant 4-mer so that P_{i-1} lies on the $\mathbf{X}_3$ axis.
- rotate further, so that P_{i-2} lies in the $\mathbf{X}_2\mathbf{X}_3$ plane and the angle between the vector $\mathbf{P_{i-2} - P_{i-1}}$ and the $\mathbf{X}_2$ axis is acute.

The transformation yields a canonical orientation for $\mathbf{P}_{i+1}$ with respect to its previous three coordinates. Using the transformed coordinates of $\mathbf{P}_{i+1}$, the direction (θ, ϕ) of $\mathbf{P}_{i+1}$ is computed. This transformation is repeated for every successive set of 4-mers in the protein chain, over all possible source structures in the collection. The empirical data collected in this way consists of 251,346 (θ, ϕ) pairs and both vMF and FB_5 mixtures are inferred on this directional data using the proposed search method.

A random sample of 10,000 (θ, ϕ) pairs generated from this empirical distribution is shown in Figure 6.11. The sphere is shown from different perspectives to illustrate the distribution of the data on its surface. The colour gradient represents the heat map corresponding to the data distribution with regions of high density coloured yellow (hot). In Figure 6.11(a) and (b), we see a concentrated mass of data (in yellow) that corresponds to helical regions in protein structures. Further in Figure 6.11(c), we observe the ellipse-like blue regions roughly corresponding to the β-strands in proteins. This illustrates the multimodal nature of the data distribution on the spherical surface. We therefore consider mixtures of vMF and FB_5 probability distributions to model this data.



(a) (b) (c)

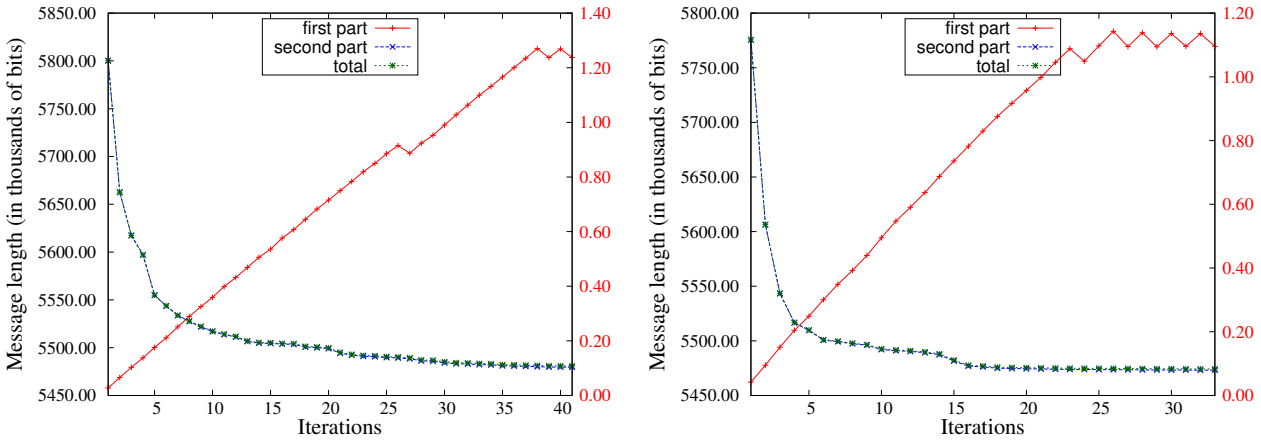
Figure 6.11: A sample of 10,000 points randomly generated from the empirical distribution of (θ, ϕ) pairs. The figure shows the random sample from different viewpoints.

Search of vMF and FB_5 mixtures

The search method inferred a 35-component vMF mixture and terminated after 41 iterations involving split, delete, and merge operations. When modelled using FB_5 distributions, the search method

inferred 23 components and terminated after 33 iterations. In each of these iterations, for every intermediate K-component mixture, each constituent component is split, deleted, and merged (with an appropriate component) to generate improved mixtures. As usual, the method terminates when these perturbations do not result in an improvement.

In the case of a vMF mixture, the search method begins with a one-component mixture, continuously favours splits over delete and merge operations until a 26-component mixture is inferred. This corresponds to the steady increase in the first part of the message length as observed by the red curve in Figure 6.12(a) until the 26^{th} iteration. Thereafter, a series of deletions and splits result in an intermediate sub-optimal 27-component mixture at the end of the 29^{th} iteration. This is characterized by the step-like behaviour of the red curve between the 26^{th} and the 29^{th} iteration.



(a) vMF mixture

(b) FB_5 mixture

Figure 6.12: Progression of the quality of the vMF and FB_5 mixtures inferred by our proposed search method. Note there are two Y-axes in both (a) and (b) with different scales: the first part of the message follows the right side Y-axis (red); while the second part and total message lengths follow the left side Y-axis (black).

The first part of the message is dependent on the number of components (model complexity) and an increase in number of mixture components leads to an increase in the encoding cost of the parameters. From the 29^{th} iteration, the method continues to split the constituent components until a 36-component mixture is inferred after 38 iterations. This is reflected in the continuous rise of the red curve in Figure 6.12(a) between the 29^{th} and 38^{th} iterations. Thereafter, through a series of perturbations, the final resultant mixture has 35 components at the end of 41 iterations, characterized by a step-like behaviour towards the end between 38^{th} and 41^{st} iterations.

In the case of the FB_5 mixture, the search method infers a 23-component mixture at the end of 23 iterations by continuous splitting. This corresponds to the steady increase in the first part of the message length denoted by the red curve in Figure 6.12(b). From here on, after a series of perturbations, the final mixture stabilizes at the end of 33^{rd} iteration thereby resulting in a 23-component mixture. This is characterized by the step-like behaviour corresponding to intermediate reduction and increase in the number of mixture components between the 24^{th} and 33^{rd} iterations.

In both cases, the second part of the message length continues to decrease as the number of mixture components increases. An initial sharp decrease is observed in both mixture types because the data is not effectively described using fewer number of components (less than 5) given that the data is multimodal. The search method terminates when the increase in the first part dominates the reduction in the second part leading to an increase in total message length.

6.3.4 Comparison of vMF and FB_5 mixture models of protein data

The resulting vMF and FB_5 mixtures of protein directional data are shown in Figure 6.13. In order to effectively visualize the individual mixture components, the illustration includes the contours of the components such that they encompass 80% of the probability corresponding to each component. The data plotted is a random sample of $10,000$ (θ, ϕ) pairs drawn from the empirical distribution of β class of proteins. The regions in Figure 6.13 are coloured based on the empirical distribution (heat map). There are two distinguishable regions of the distribution of θ and ϕ values. At $(\theta, \phi) \sim (90°, 60°)$, there is a concentrated mass which corresponds to the *helical* region in a typical protein. The area characterized by $\theta \in (40°, 80°)$, $\phi \in (180°, 270°)$ roughly corresponds to the *strand* region in proteins.

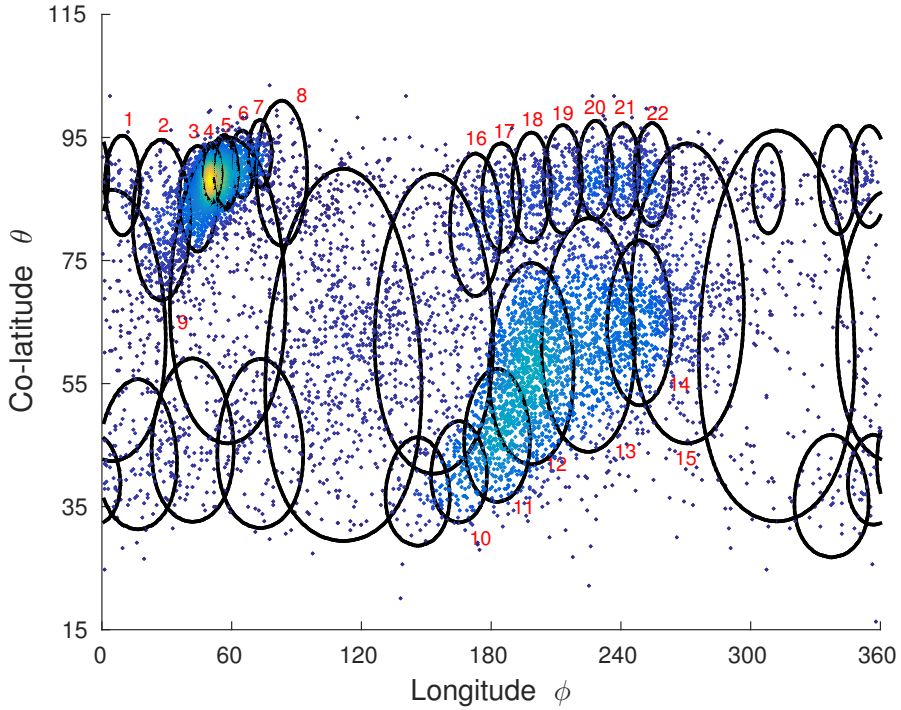
The search method inferred a 35-component vMF mixture and a 23-component FB_5 mixture. It is observed that the number of components used to model the entire collection of 251,346 (θ, ϕ)-pairs using a FB_5 mixture model is fewer compared to a vMF mixture. This is expected as a vMF distribution is a specific case of a FB_5 distribution and, hence, a vMF mixture requires more components to model data that is asymmetrically distributed. In Figure 6.13(a), the vMF mixture components 1-9 are used to model the helical region (approximately), whereas in Figure 6.13(b), the same region is modelled using FB_5 mixture components 1-6. Similarly, the strand region in the proteins is modelled by components 10-15 in the vMF case, whereas, it is modelled by components 7-11 using the FB_5 mixture. Further, components 16-22 in the vMF mixture and components 12, 13 in the FB_5 mixture model the same region. The other regions in the protein directional data space follow the same modelling pattern, that is, with fewer FB_5 components. These observations reflect the better explanatory power of FB_5 mixtures compared to vMF mixtures.

Compared to a singleton vMF distribution, the encoding cost of the parameters of a FB_5 distribution would be greater as it is a complex model with more number of parameters. As shown in Table 6.5, the encoding cost of the parameters of the inferred 23-component FB_5 mixture is 1095 bits. A vMF mixture with the same number of components has a first part equal to 819 bits (a difference of 276 bits). However, the second part of the message (the fit to the data) is lower for the FB_5 mixture (a difference of $\sim 18,000$ bits). Hence, the gain in the second part outweighs the greater cost of encoding the more complex FB_5 mixture. Thus, the total message length is lower for the FB_5 mixture and serves as a better model to explain the data. If the 23-component vMF mixture is compared with the 35-component vMF mixture inferred by our search method, the vMF mixture with 23 components has smaller first part and greater second part (Table 6.5), which is expected. The 35-component mixture has a first part equal to 1237 bits compared to 819 bits in the 23-component case (a difference of 418 bits). However, there is a gain of 12,000 bits in the second part, thus, resulting in a significantly lower total message length in the 35-component case. Through this analysis, it is shown how the trade-off of choosing a complex model and the quality of fit is addressed using the MML framework.

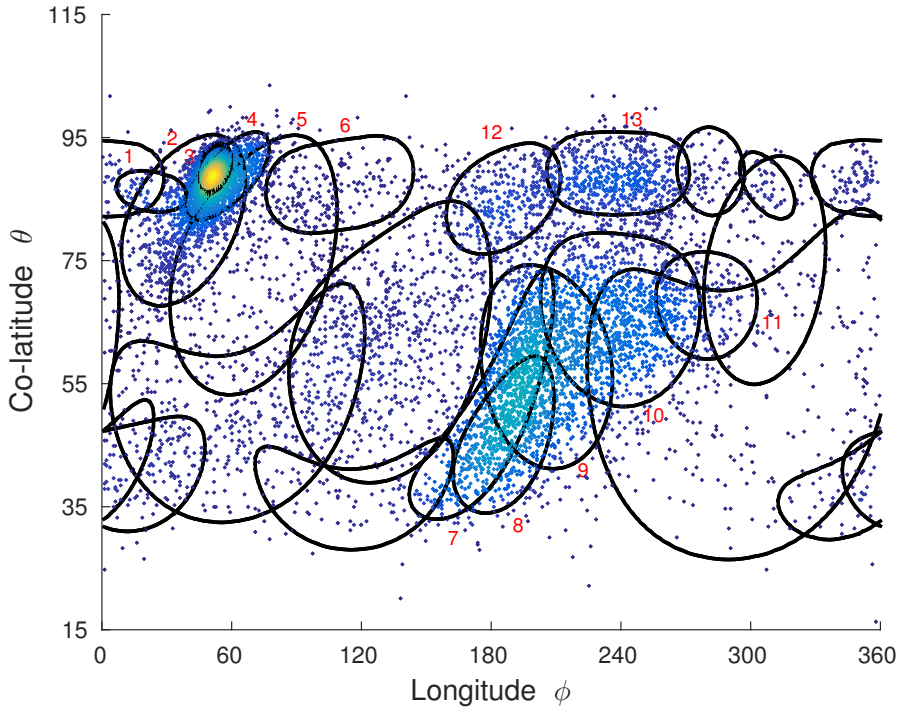
Table 6.5: Message lengths of the mixtures inferred on the protein directional data.

Mixture model	Number of components	First part (thousands of bits)	Second part (millions of bits)	Total message length (millions of bits)
vMF	23	0.819	5.491	5.492
vMF	35	1.237	5.479	5.481
FB_5	23	1.095	5.473	**5.474**

It is also interesting to note the shape of the contours generated by both vMF and FB_5 mixtures. A vMF distribution caters to symmetrically distributed data and has circular contours of constant probability on a spherical surface. Hence, in the (θ, ϕ)-space, we see regular oval-shaped contours as shown in Figure 6.13(a). In contrast, an FB_5 distribution has ellipse-like contours on a spherical surface (Figure 4.4). Thus, when projected onto the (θ, ϕ)-space, it results in a myriad of contour shapes (Figure 6.13b) depending on the parameters defining an FB_5 distribution.



(a) vMF MML mixture (35 components)



(b) FB_5 MML mixture (23 components)

Figure 6.13: Mixtures inferred on the β-class proteins (θ and ϕ are in degrees).

Utility of the inferred mixture models in structural bioinformatics

The better explanatory power of FB_5 mixtures over vMF mixtures leads to enhanced data compression (demonstrated through Table 6.5) and, hence, serves as an efficient descriptor to model directional data. In the context of proteins, the previous null model description is based on the naïve uniform distribution on the sphere (Konagurthu et al., 2012). The null model description provides a baseline for encoding protein coordinate data in modelling tasks in structural bioinformatics (Konagurthu et al.,

2012, 2013; Collier et al., 2014). In this regard, the use of vMF and FB_5 mixtures offers a better alternative as opposed to an encoding that uses the uniform distribution.

The message length expressions to encode the directional data using uniform, vMF, and FB_5 null models are given by Equation 6.1 (Konagurthu et al., 2012; Kasarapu and Allison, 2015), where $\mathbf{x}$ corresponds to a unit vector described by (θ, ϕ) on the surface of the sphere, ϵ is the precision[4] to which each coordinate is measured, and r denotes the distance between successive C_α atoms. In Equation 6.1, for the vMF mixture, $K = 35$ and the null model corresponding to the FB_5 mixture has $K = 23$ components.

$$\begin{aligned} \text{Uniform Null} &= -\log_2\left(\frac{\epsilon^2}{4\pi r^2}\right) = \log_2(4\pi) - 2\log_2\left(\frac{\epsilon}{r}\right) \quad \text{bits.} \\ \text{vMF \& } FB_5 \text{ Null} &= -\log_2\left(\sum_{j=1}^{K} w_j f_j(\mathbf{x};\Theta_j)\right) - 2\log_2\left(\frac{\epsilon}{r}\right) \quad \text{bits.} \end{aligned} \tag{6.1}$$

The inferred vMF and FB_5 mixture models are then used to encode the entire protein data. After accounting for the distances between the successive C_α atoms ($r \sim 3.8$ as in Figure 6.10a), the resulting total message lengths are given in Table 6.6. The uniform distribution is clearly not an appropriate descriptor and this can be reasoned from the fact the data is concentrated only in about one-half of the (θ, ϕ)-space (Figure 6.11a).

Further, the empirical distribution has multiple modes and is best modelled using mixtures (Figure 6.13). The inferred vMF mixture has better explanatory power over the uniform distribution as it has a corresponding saving of 446,000 bits over 251,346 data points (residues). This translates to an enhanced compression of 1.778 bits per residue (on average). The inferred FB_5 mixture, however, encodes the same amount of data with a saving of 7,000 bits against the vMF mixture (an average of 0.026 bits extra compression per residue). The results following the application of FB_5 mixtures to modelling protein directional data demonstrate that that they supersede the vMF mixture models (Table 6.6). The ability of FB_5 distributions to model asymmetrical data leads to an improved encoding of the protein data. Hence, they serve as natural successors to the vMF null model descriptors.

Table 6.6: Comparison of the null model encoding lengths based on uniform distribution, vMF mixture (35 components), and FB_5 mixture (23 components).

Null model	Message length (millions of bits)	Bits per residue
Uniform	6.895	27.434
vMF mixture	6.449	25.656
FB_5 mixture	**6.442**	**25.630**

Comparison of the MML criterion with AIC and BIC

The MML criterion is used to compute the score associated with a mixture model by separately encoding the parameters (first part) and the data given those parameters (second part). This yields the total message length (Equation 5.5) which is used to find improved mixtures during the search process. In addition to the MML criterion, as discussed in Section 5.3, the traditional information-theoretic criteria used are AIC and BIC/MDL (see Chapter 2).

AIC and BIC introduce constant term penalties depending on the number of free parameters in the mixture model. These are given by Equations 2.7 and 2.6 respectively. For a vMF mixture with K components, the number of free parameters is $p = 3K + (K-1) = 4K - 1$ (3 free parameters per component plus $K - 1$ mixture weights). Similarly, for an FB_5 mixture, $p = 5K + (K-1) = 6K - 1$.

[4]Protein coordinate data is measured to an accuracy of $\epsilon = 0.001$Å.

Mixture modelling of some observed data based on these criteria can be done as follows:

- *Exhaustive search:* The search heuristic (Section 5.4) to determine the optimal mixture can be used alongside any objective function and not necessarily the MML criterion. The series of perturbations are carried out as described and the improvement to mixtures is determined based on the criterion in use.

- *Traditional search:* As discussed in Section 5.3.1, the traditional search method using AIC/BIC involves estimating the mixture parameters using the EM algorithm (Section 5.2.1) for varying number of components K and choosing the one which results in minimum criterion value.

It is to be noted that with the MML criterion, the EM algorithm in Section 5.2.2 is used to obtain the MML estimates of the mixture parameters. However, with AIC and BIC, the EM algorithm in Section 5.2.1 results in the maximum likelihood (ML) estimates for a given K. For FB_5 mixtures, the ML estimates are often approximated by the moment estimates which are used in the M-step of the EM algorithm (Peel et al., 2001; Kent and Hamelryck, 2005; Hamelryck et al., 2006). The results for mixtures obtained using the ML estimates and their approximations are compared against those obtained using the MML-based estimates.

The results pertaining to the *exhaustive* search method are shown in Table 6.7. It is observed that when the search is based on AIC, the mixtures resulting due to moment and ML estimation have 37 and 34 components respectively. The moment and the ML mixtures have the same AIC values in this case. With BIC, the mixture resulting from the ML estimation has the lower BIC value. In this case, the moment and the ML mixtures have 23 and 24 components respectively. This number resembles the one obtained by the exhaustive search but MML-based parameter estimation. The ML mixture has the lowest BIC score.

Table 6.7: FB_5 mixtures inferred by employing the *exhaustive search* method and changing the evaluation criteria and methods to estimate mixture parameters.

Criterion	Moment mixtures			Maximum likelihood mixtures		
	K	Criterion score ($\times 10^5$ bits)	Message length ($\times 10^6$ bits)	K	Criterion score ($\times 10^5$ bits)	Message length ($\times 10^6$ bits)
AIC	37	**2.313**	5.474	34	**2.313**	5.474
BIC	23	4.647	5.475	24	**4.645**	5.474

The results pertaining to the traditional search method are shown in Figure 6.14. As the number of components K is increased, it is expected that the AIC and BIC scores decrease until some minimum is reached and then increase thereafter. The value of K at which this behaviour happens is treated to be the optimal mixture that models the data. It is observed that initially, both criteria decrease and after $K = 30$, the values do not change dramatically. By increasing K, the linear increase in penalty factors and the associated increase in log-likelihood are of the same magnitude and, hence, the difference in criteria is not apparent. Thus, using the traditional search, it is difficult to decide on an appropriate number of mixture components.

The trend observed in Figure 6.14 is the same for mixtures obtained using both moment and ML estimates. The expressions for AIC and BIC do not help in distinguishing the moment and ML mixtures because for different types of estimates and a given K, the penalty terms are the same. Also, the log-likelihood is approximately the same because for huge amounts of data, as is the case here, all the estimates converge to the same value.

In contrast, if we compute the first part message lengths corresponding to the moment and ML mixtures for a given K, the differences in their encoding lengths become apparent. The variation in the first part message lengths for the moment and ML mixtures resulting from the traditional search are shown in Figure 6.16. It is observed that until $K = 30$, the first part message lengths of moment

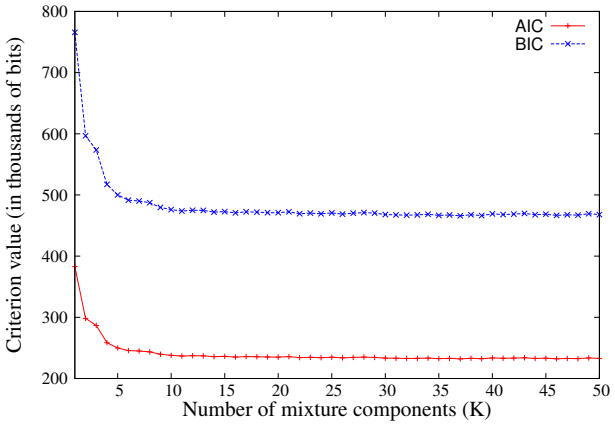


Figure 6.14: Comparison of the criteria computed for maximum likelihood mixtures (moment mixtures have the same behaviour and are, hence, not shown)

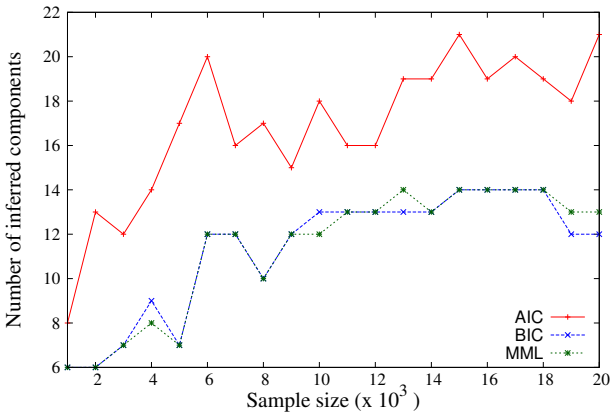
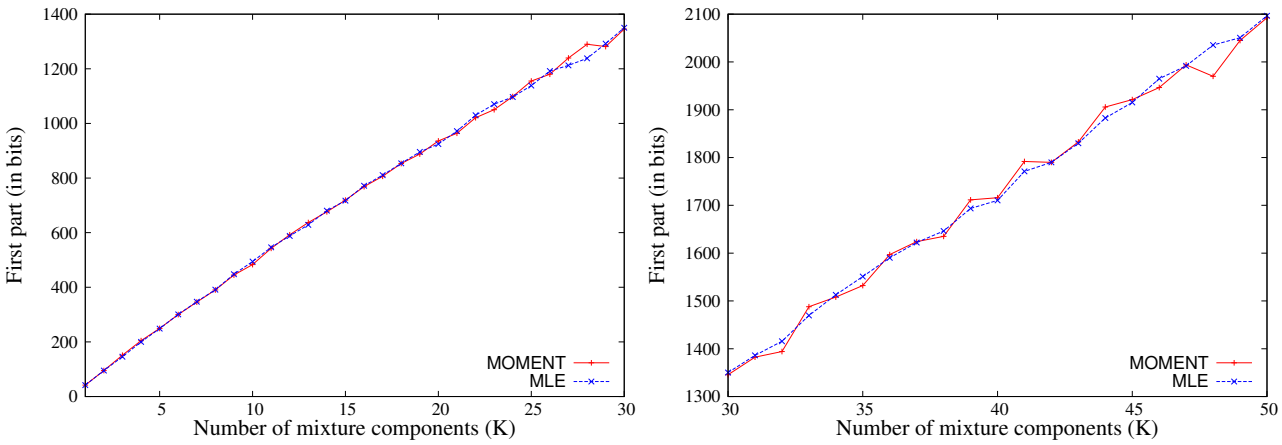


Figure 6.15: Variation of the number of inferred FB_5 components using the search method based on exhaustive perturbations.

and ML mixtures are close to each other. In Figure 6.16(b), when $K > 30$, there are minute differences between encoding lengths of mixture parameters obtained using moment and ML estimates. Thus, unlike AIC/BIC, the MML criterion is able to distinguish mixtures with equal number of components. The first part corresponds to the model complexity and is dependent on not just the number of components K but also on the components' parameters themselves according to the MML framework.



(a) (b)

Figure 6.16: First part message length corresponding to mixtures evaluated using AIC. The results for BIC display the same pattern and are, hence, not shown. (the range of $K \in [1, 50]$ is split into two sub-figures (a) and (b) in order to highlight the differences in the message lengths).

The above discussion is aimed at projecting the limitations of the traditional search method and also the use of AIC and BIC as the evaluation criteria. We find in MML an objective way to assess mixtures and, in conjunction with the search method, offers a better alternative to determine reliable mixture models. We further illustrate the behaviour of the search method with smaller amount of data. The previous discussion pertains to the entire empirical data set containing $N = 251,346$ (θ, ϕ) pairs. In the current context, from the empirical distribution, we randomly sample varying amounts of data ranging from $N = 1000$ to $N = 20,000$. The experiment is conducted by fixing the search method

(exhaustive) but changing the evaluation criteria to infer suitable FB_5 mixtures. It is observed that the mixtures based on AIC have greater number of components as compared to BIC and MML (see Figure 6.15). The mixtures corresponding to BIC and MML have the same number of components in most of the experimental trials. These results are in agreement with what was observed on the complete protein data (Table 6.7), where AIC resulted in greater number of components.

6.4 Mixtures of bivariate von Mises distributions

We consider two kinds of bivariate von Mises (BVM) distributions in mixture modelling. In addition to the Sine variant (Equation 4.43) that has the correlation parameter λ, we also consider the independent variant obtained when $\lambda = 0$. The independent version assumes zero correlation between the data distributed on the torus (see Equation 4.45). We provide a comparison for the mixture models obtained using both versions of the BVM distributions.

Previous work on MML-based modelling of protein dihedral angles used independent BVM distributions (Dowe et al., 1996a). Their work used the Snob mixture modelling software (see Section 5.3.7). As pointed out by Dowe et al. (1996a), Snob does not have the functionality to account for the correlation between the data. We therefore study the BVM Sine distributions and demonstrate how they can be integrated with our generalized MML-based mixture modelling method.

6.4.1 Approach for BVM distributions

We extend the search method described in Section 5.4 to infer mixtures of BVM distributions. To infer the optimal number of mixture components, the mixture modelling apparatus is now modified to handle the directional data distributed on the surface of a *three-dimensional* (3D) torus.

As in the case of the 3D vMF and FB_5 distributions, the split operation detailed in Section 5.4.2 is tailored for the BVM mixtures. The basic idea behind splitting a parent component is to identify the means of the child components so that they are on either side of the parent mean and are reasonably apart from each other. Recall that for a Gaussian parent component, we computed the direction of maximum variance and selected the initial means, along this direction, that are one standard deviation away on either side of the parent mean. We employ the same strategy for BVM distributions. For data $\mathcal{D} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$, where $\mathbf{x}_i = (\phi_i, \psi_i)$ such that $\phi_i, \psi_i \in [-\pi, \pi)$, we compute the direction of maximum variance in the (ϕ, ψ)-space. This allows us to compute the initial means of the child components.

The delete and merge operations are carried out in the same spirit. During merging BVM components, the KL distance is evaluated to determine the closest pair. We derive the KL distance for BVM Sine and BVM Independent distributions as shown in Appendix C.1. Further, in all the operations, the MML estimators of the BVM Sine distribution, derived in Section 4.4.3 are used in the update step of the EM algorithm (Section 5.2.2). We adopt the search strategy as described in Algorithm 1.

6.4.2 Mixture modelling of protein main chain dihedral angles

Recall that in Section 6.3.3 we explained the arrangement of the central carbon C_α atoms along the protein backbone. We previously considered the directional data arising from the spatial orientations of these C_α atoms. Because of the almost constant C_α-C_α distance, the data was generated using the spherical coordinate system and, therefore, we modelled the data using mixtures of vMF and FB_5 directional probability distributions.

In this section, we consider the spatial orientations resulting from not just the C_α atoms but also the C and N atoms that link the C_α atoms. A protein main chain is characterized by a sequence of ϕ, ψ, and ω angles. These angles uniquely determine the 3D geometry of the protein backbone structure (Richardson, 1981). However, in a majority of protein structures, $\omega = 180^\circ$ and, hence the

sequence of C_α-C-N-C_α atoms lie in a plane (see the dotted planar representation in Figure 6.17a). As a result, the angles ϕ and ψ are typically analyzed[5] (Ramachandran et al., 1963).

The angles ϕ and ψ are called the dihedral angle pair corresponding to an amino acid residue with a central carbon atom C_α along the protein main chain. Geometrically, a dihedral angle is the angle between any two planes defined using four non-collinear points. In Figure 6.17(a), ϕ is the angle between the two planes formed by C-N-C_α and N-C_α-C. Similarly, ψ is the angle between the two planes formed by N-C_α-C and C_α-C-N.

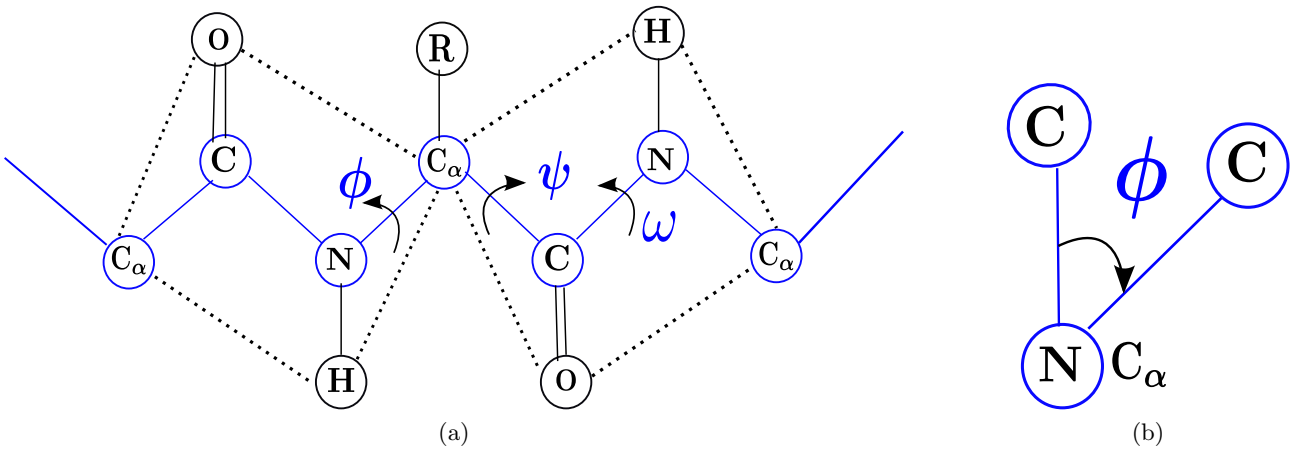


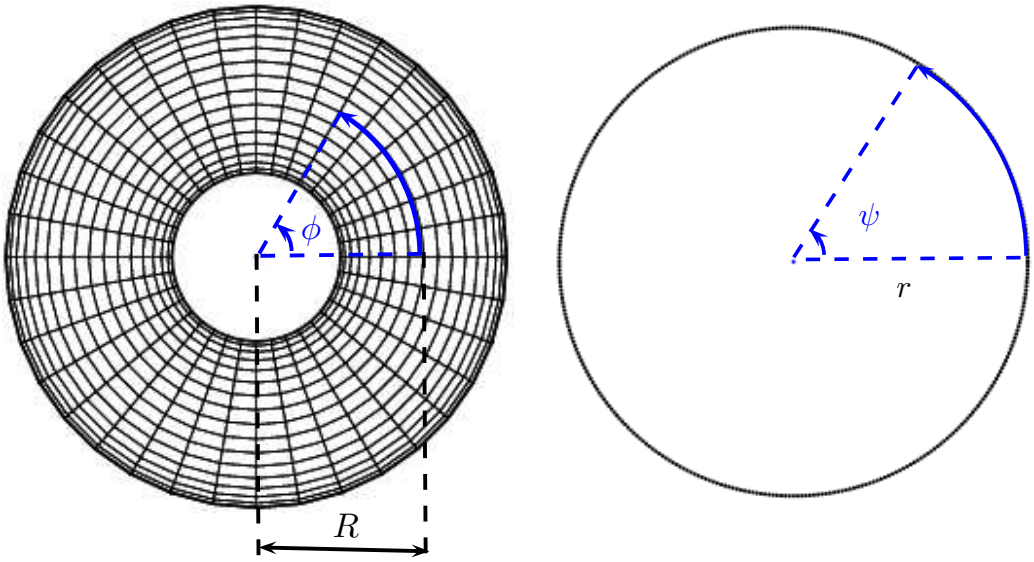
Figure 6.17: Protein main chain dihedral angles denoted by (ϕ, ψ).

The dihedral angles ϕ and ψ are measured in a consistent manner. For example, in order to measure ϕ, the four atoms C-N-C_α-C are arranged such that ϕ is calculated as the deviation between N-C and C_α-C when viewed in some consistent orientation. As an illustration, in Figure 6.17(b), view the arrangement of the four atoms through the N-$C_\alpha$bond such that C_α is behind the plane of the paper and N eclipses the C_α atom. Also, the C atom directly attached to N is at the 12 o' clock position. In this orientation, ϕ is given as the angle of rotation required to align the N-C bond with the C_α-C bond in the plane of the paper. Further, if it is a clockwise rotation, it is considered a positive value. This ensures that $\phi \in [-\pi, \pi)$. The dihedral angle ψ is measured by following the same convention with the four atoms being N-C_α-C-N. The (ϕ, ψ) pair measured in this way can be plotted on the surface of a 3D torus. Each (ϕ, ψ) pair corresponds to a point on the toroidal surface. The angle ϕ is used to identify a particular cross-section (circle) of a torus, while ψ locates a point on this circle (see Figure 6.18).

As in Section 6.3.3, we generate the directional data (dihedral angles in this case) from the previously considered 1802 experimentally determined protein structures from the ASTRAL SCOP-40 (version 1.75) database (Murzin et al., 1995) representing the "β class" proteins. The number of (ϕ, ψ) dihedral angle pairs resulting from this data set is 253,165. We model this generated set of dihedral angles using BVM Sine distributions.

A random sample from this empirical distribution consisting of 10,000 points is shown in Figure 6.19. The plot is a heat map showing the density of the data distribution on the toroidal surface. Note that there are regions on the torus which are highly concentrated (yellow), corresponding to the helical regions in the protein. The ellipse-like patches (mostly in blue) roughly correspond to the β strands in proteins. Furthermore, the data is multimodal which motivates its modelling using mixtures of BVM distributions. We consider the effects of using the BVM Sine distribution as compared to the BVM Independent variant in this context.

[5]Note that the angle ϕ is different from the one considered in modelling using vMF and FB_5 distributions in Section 6.3.3.



(a) Identifying the cross-section at ϕ (b) Circular cross-section at ϕ

Figure 6.18: Representing a (ϕ, ψ) point on the torus.

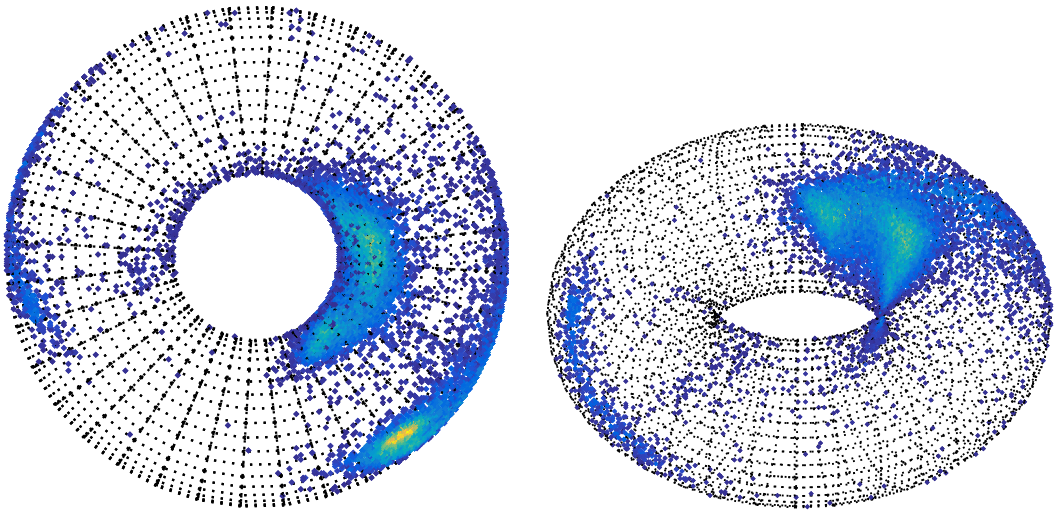


Figure 6.19: A sample of 10,000 points randomly generated from the empirical distribution of (ϕ, ψ) pairs. The figure shows the random sample from different viewpoints.

Search of BVM Independent and BVM Sine mixtures

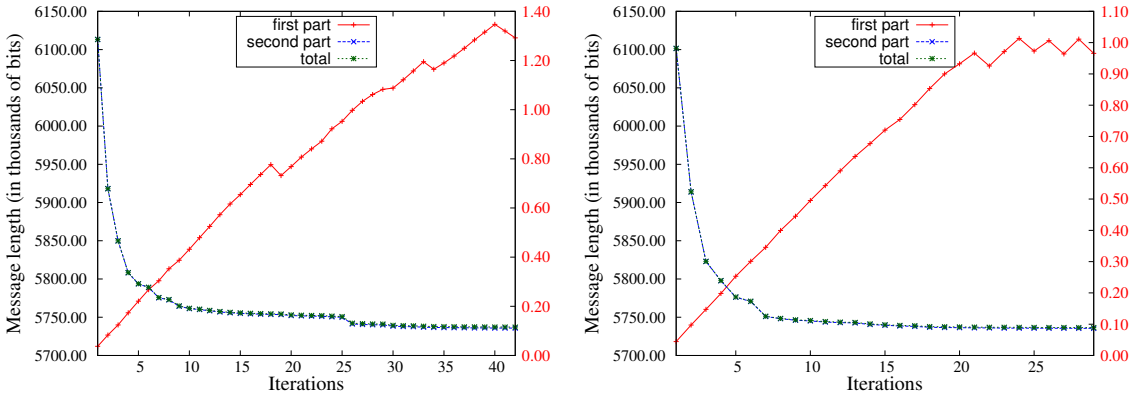
The search method inferred a 32-component BVM Independent mixture and terminated after 42 iterations involving split, delete, and merge operations. In the case of modelling using BVM Sine distributions, our search method inferred 21 components and terminated after 29 iterations. In each of these iterations, for every intermediate K-component mixture, each constituent component is split, deleted, and merged (with an appropriate component) to generate improved mixtures.

The progression of the search method for the optimal BVM Independent mixture begins with a single component. The search method results in continuous split operations until the 17^{th} iteration when a 17-component mixture is inferred (see Figure 6.20a). This corresponds to a progressive increase in the first part of the message (red curve). Between the 17^{th} and the 21^{st} iterations, we observe a series of delete/merge and split operations leading to a stable 19-component mixture. The search method again continues to favour the split operations until the 28^{th} iteration when a 26-component mixture is inferred. Thereafter, a series of deletions and splits yield a stable 29-component mixture at

the end of the 35$^{\text{th}}$ iteration. The search method eventually terminates when a 32-component mixture is inferred with a characteristic step-like behaviour towards the end indicating perturbations involving split and delete/merge operations (see Figure 6.20a).

In the case of searching for the optimal BVM Sine mixture, our proposed search method continues to split the components thereby increasing the mixture size. This occurs until 21 iterations. At this stage, there are 21 mixture components. This can be observed in Figure 6.20(b), when the first part of the message (red curve) continually increases until the 21$^{\text{st}}$ iteration. During this period, observe that the second part (blue) and the total message length (green) continually decrease signifying an improvement to the mixtures.

After the 21$^{\text{st}}$ iteration, we observe a step-like behaviour as in the case of mixture modelling using the BVM Independent distributions. The behaviour characterizes the reduction or increase in the number of mixture components corresponding to a decrease or increase to the first part of the message. After the 24$^{\text{th}}$ iteration, we observe that the mixture has 22 components. However, the final mixture stabilizes in the subsequent iterations to a 21-component mixture. After the 29$^{\text{th}}$ iteration, there is no further improvement to the total message length and the search method terminates.



(a) BVM Independent

(b) BVM Sine

Figure 6.20: Progression of the quality of the BVM mixtures inferred by our proposed search method. Note there are two Y-axes in both (a) and (b) with different scales: the first part of the message follows the right side Y-axis (red); while the second part and total message lengths follow the left side Y-axis (black).

There are striking similarities to the progression of the search method when compared with that used for the inference of vMF and FB$_5$ mixture models. We observe the same characteristic increase in the mixture size initially followed by some perturbations stabilizing the intermediate mixture (step-like behaviour), and eventually resulting in an optimal mixture (see Figure 6.12). Similar to the case of vMF and FB$_5$ mixtures, there is an initial sharp decrease in the total message length until about 7 iterations for BVM mixtures. Because of the multimodal nature of the directional data (see Figure 6.19), the initial increase in the number of components would explain the data distribution corresponding to those modes that are clearly distinguishable. This leads to a substantial improvement to the total message length as the minimal increase in the first part is dominated by the gain in the second part. However, towards the end of the search, when the increase in first part dominates the reduction in second part, the method stops. Thus, we see the trade-off of model complexity (as a function of the number of components and their parameters), and the goodness-of-fit being balanced using the search based on the MML inference framework.

6.4.3 Comparison of BVM mixture models of protein data

The existing work of MML-based mixture modelling of protein dihedral angles by Dowe et al. (1996a) inferred 27 clusters using the BVM Independent distributions. In contrast, our search method inferred

32 clusters. However, their data consists of only 41,731 (ϕ, ψ) pairs generated from the protein structures known at that time. In contrast, we have used 253,165 pairs of dihedral angles along with a different search method as explained previously (see Section 6.4.2). So, there is some consensus on the rough number of component distributions if the protein dihedral angles were modelled using BVM distributions assuming no correlation between ϕ and ψ.

The visualization of the dihedral angles is commonly done by the Ramachandran plot (Ramachandran et al., 1963) who first analyzed the various possible protein configurations and represented them as a two-dimensional plot. An example of one such plot is provided in Lovell et al. (2003) and reproduced here (Figure 6.21). Such a plot is indicative of the allowed conformations that protein structures can adopt. There are vast spaces in the dihedral angle space where few data are present. The conformations corresponding to those regions are not possible. We consider the plot to explain the similarities between our inferred mixture models and the one that is traditionally used.

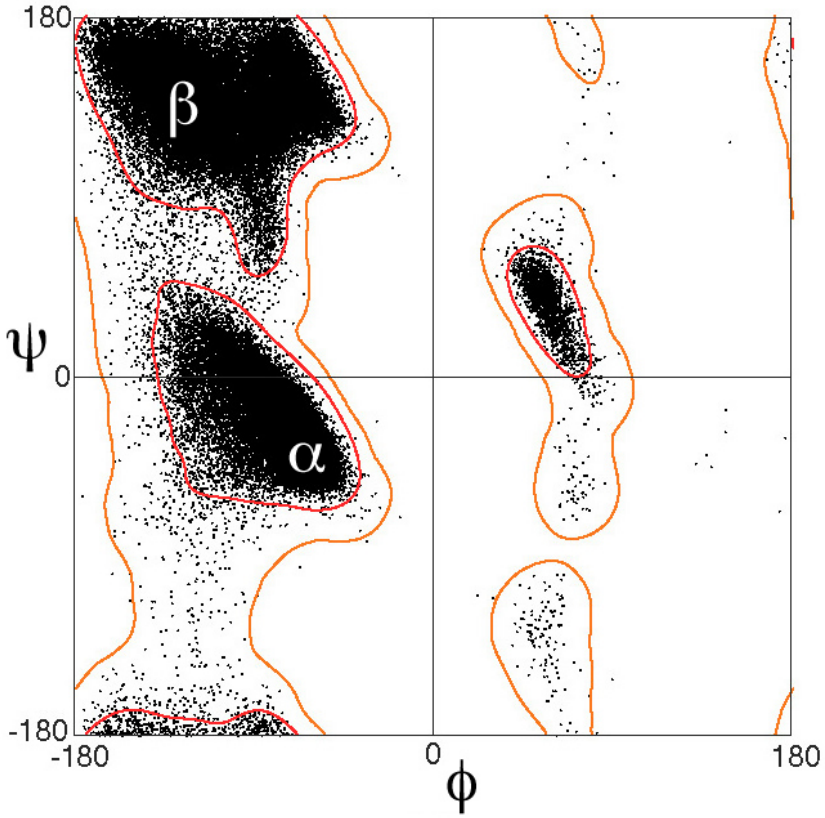
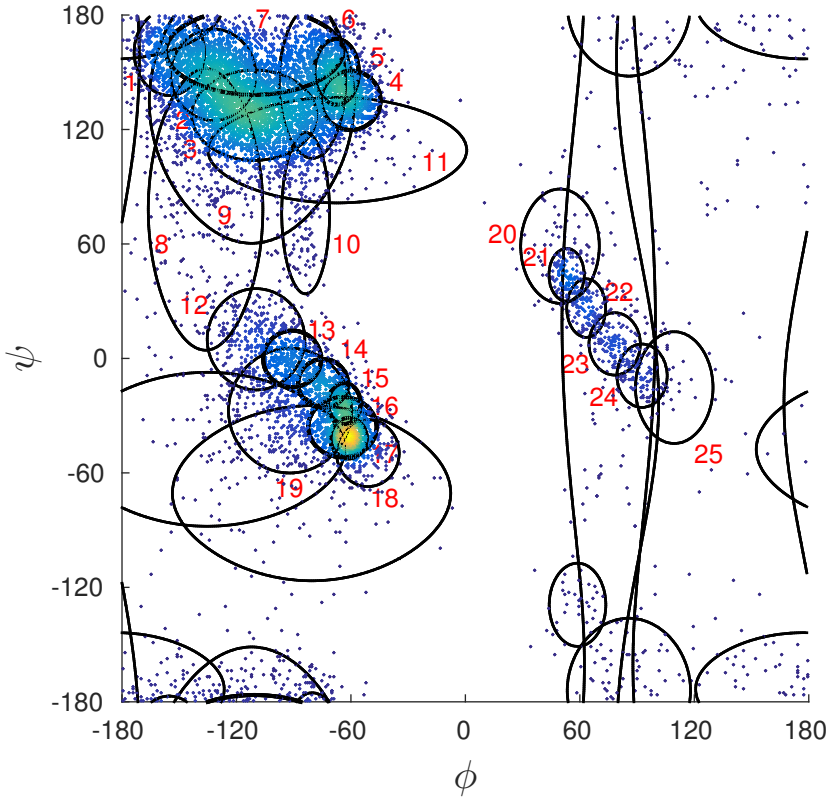


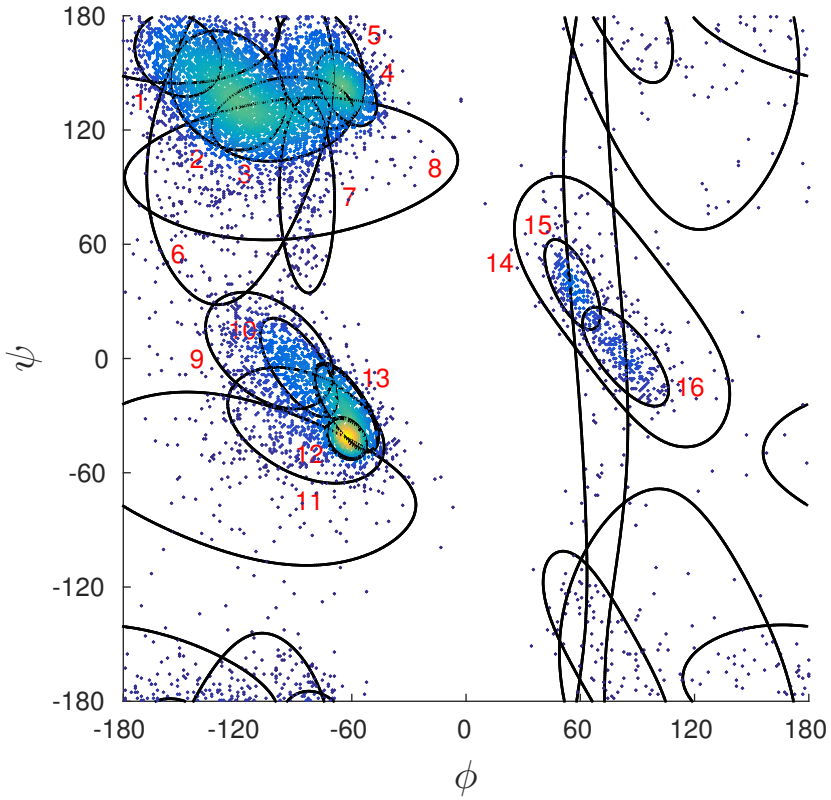
Figure 6.21: Models of the protein main chain dihedral angles (ϕ and ψ are in degrees). Plot taken from Lovell et al. (2003).

Our resulting mixtures of BVM Independent and the Sine variants are shown in Figure 6.22. The contours of the constituent components plotted in the (ϕ, ψ)-space can be seen in the diagram. For visualization purposes, we display the contour of each component that corresponds to 80% of the data distribution. The data in Figure 6.22 corresponds to a random sample drawn from the empirical distribution (same as in Figure 6.19) visualized in the (ϕ, ψ)-space.

In Figure 6.21, we observe that the top-left region corresponds to the β strands in protein structures. The empirical distribution of dihedral angles we generated also has this characterstic. We observe a concentrated mass in the top-left in Figure 6.22. Furthermore, our inferred mixtures are able to model this region using the appropriate components. Note that smaller or highly compact contours correspond to BVM distributions that have greater concentration parameters (κ_1 and κ_2 in Equation 4.43).



(a) BVM Independent MML mixture (32 components)



(b) BVM Sine MML mixture (21 components)

Figure 6.22: Models of the protein main chain dihedral angles (ϕ and ψ are in degrees).

We note that components numbered 1-11 (Figure 6.22a) and components 1-8 (Figure 6.22b) are used to describe this region. These correspond to the components of the BVM Independent and BVM Sine models respectively. Clearly, more number of components are required to model roughly the same amount of data (corresponding to the β strands) using the BVM Independent mixture.

Similarly, in Figure 6.21, we observe another concentration of mass in the middle-left portion of the figure. This corresponds mainly to *right-handed* α-helices, which are very frequent in protein structures. In Figure 6.22, we have the corresponding mass and also note the dense region (bright yellow). As per our inferred mixtures, component 17 (Figure 6.22a) and component 12 (Figure 6.22b) are used to predominantly describe this dense region. The other surrounding regions in the dihedral angle space of the right-handed helices are described by components 12-19 (Figure 6.22a) and by components 9-13 (Figure 6.22b). Again, we observe that the similar data is described using 8 components by the BVM Independent mixture as opposed to 5 components by the BVM Sine mixture.

Lovell et al. (2003) display another region of concentrated mass in the middle-right of Figure 6.21. This region corresponds to the infrequent *left-handed* helices in protein structures. We see a corresponding mass in the empirical distribution in Figure 6.22. The components 20-25 of our inferred BVM Indenpendent mixture describe this region (Figure 6.22a). The same region is described by components 14-16 of our inferred BVM Sine mixture (Figure 6.22b). Notice how this region is described by components 15 and 16. These two components describe the dense mass within this region while component 14 is responsible for mainly modelling the data that is further away from this clustered mass. We again observe that the same region is modelled by greater number of components when using BVM Independent distributions.

The remaining mixture components describe the insignificant mass present in other regions of the dihedral angle space. The ability of our inferred mixtures to identify and describe specific regions of the protein conformational space in a completely unsupervised setting is remarkable. Further, we have qualified the effects of using the BVM distributions which do not account for the correlation between the dihedral angle pairs. In this regard, the BVM Sine mixtures fare better when compared to mixtures of BVM Independent distributions. We now quantify these effects in terms of the total message length.

As described in Section 5.4, our proposed search method to infer an optimal mixture involves evaluating the encoding cost of the mixture parameters or the first part (model complexity), and encoding the data using those parameters or the second part (goodness-of-fit). The progression of the search method continues until there is no improvement to the total message length. We observe that the resulting 21-component BVM Sine mixture has a first part of 966 bits and a corresponding second part of 5.735 million bits (see Table 6.8). A BVM Independent mixture with the same number of components has a first part of 872 bits and a corresponding second part of 5.751 million bits. Although the model complexity is lower for the BVM Independent mixture (difference of ~ 94 bits), the BVM Sine mixture has an additional compression of $\sim 16,000$ bits in its goodness-of-fit. Thus, the significant gain in the second part dominates the minimal increase in the first part of the BVM Sine mixture.

Further, if we compare the 21-component BVM Independent mixture with the inferred 32-component BVM Independent mixture, we observe that the first part is more in the 32-component case. This is expected because there are more number of mixture parameters to encode in the 32-component mixture. There is a difference of $1292 - 872 = 420$ bits (see Table 6.8). However, the 32-component mixture results in an extra compression of $\sim 15,000$ bits. So, the total message length is lower for the 32-component mixture, and is therefore, preferred to the 21-component BVM Independent mixture.

When the inferred 32-component BVM Independent and the 21-component BVM Sine mixtures are compared, we observe that the total message length is lower for the BVM Sine mixture. In this case, both the first and second parts are lower for the Sine mixture leading to an overall gain of about ~ 1000 bits. Thus, the BVM Sine mixture is more appropriate as compared to the BVM Independent mixture in describing the protein dihedral angles. This exercise shows how an optimal mixture model

Table 6.8: Message lengths of the BVM mixtures inferred on the protein dihedral angles.

Mixture model	Number of components	First part (thousands of bits)	Second part	Total message length
			(millions of bits)	
Independent	21	0.872	5.751	5.752
Independent	32	1.292	5.736	5.737
Sine	21	0.966	5.735	**5.736**

is selected by achieving a balance between the trade-off due to the complexity and the goodness-of-fit to the data.

Furthermore, as in the case of the vMF and FB_5 distributions, we can devise null model descriptions of protein dihedral angles based on the BVM mixtures. For comparison, we consider a uniform distribution on the torus, which is referred to as the uniform null model in the equation below.

$$\text{Uniform Null} = -\log_2\left(\frac{\epsilon^2}{4\pi^2 Rr}\right) = 2\log_2(2\pi) - \log_2\left(\frac{\epsilon^2}{Rr}\right) \quad \text{bits.}$$

where R and r are the radii that define the size of the torus (see Figure 6.18). When $R = r = 1$, the surface area of the torus is $1/4\pi^2$. The null models based on the BVM mixtures have the same form as the vMF and FB_5 mixtures given in Equation 6.1 with $K = 32$ and $K = 21$ corresponding to the Independent and the Sine variants respectively.

Compared to the uniform model, both the BVM mixtures result in additional compression (see Table 6.9). The message length to encode the entire collection of 253,165 dihedral angle pairs using the uniform null model is 6.388 million bits which amounts to 25.234 bits per residue. In comparison, the BVM Independent mixture results in a compression of 5.735 million bits which amounts to 22.656 bits per residue. The additional compression is therefore, close to 2.58 bits per residue (on average). The BVM Sine mixture further leads to an additional compression of 323 bits over the BVM Independent mixture. This is equivalent to an additional saving of 0.0013 bits per residue (on average).

Table 6.9: Comparison of the null model encoding lengths based on the uniform distribution on the torus, the 32-component BVM Independent and the 21-component BVM Sine mixtures.

Null model	Message length (in bits)	Bits per residue
Uniform	6,388,508	25.2346
BVM Independent mixture	5,735,711	22.6560
BVM Sine mixture	**5,735,388**	**22.6547**

These results indicate that the BVM mixtures are superior compared to the uniform model. This can be argued from the fact that the empirical distribution (see Figure 6.19) has empty regions in the dihedral angle space. This is also confirmed from the Ramachandran plot (Figure 6.21). However, the BVM Independent and the BVM Sine variants are in close competition with each other. Noting that we need more mixture components in the Independent case and because the Sine mixture can describe the data more effectively, we conclude that the BVM Sine mixture supersedes the BVM Independent mixture. The ability of the BVM Sine mixture to model correlated data leads to improved description of the protein dihedral angles.

6.5 Summary

In this chapter, we have extended our proposed search method (described in Chapter 5) to infer mixtures of directional probability distributions. The generalized mixture modelling apparatus is tailored to cater to specific probability distributions. Based on the nature of the probability distributions, the

Split operation discussed in Section 5.4 needs to be fine-tuned. Additionally, in order to carry out the Merge operation that requires identification of a closest component based on KL distance, we derived the expressions to compute the KL distance for each individual probability distribution.

We have considered the modelling of directional data that is distributed on the unit hypersphere (by multivariate vMF mixtures), on the 3D sphere (by FB_5 mixtures), and on the 3D torus (by BVM mixtures). We have validated the performance of the search method by modelling multivariate data on the unit hypersphere in varying scenarios. We have employed the vMF mixtures to determine clusters of text documents. We have shown how our search method is able to determine an optimal number of vMF clusters in an unsupervised manner without requiring to have any background knowledge of the distribution of the data. We published the results of mixture modelling using vMF distributions (Kasarapu and Allison, 2015).

The FB_5 mixtures have been employed for modelling asymmetrically distributed data on the unit 3D sphere. We have generated the directional data corresponding to the coordinates of C_α atoms along the protein backbone (see Section 6.3.3). The resulting FB_5 mixtures are able to identify conformationally relevant regions in the directional angle space corresponding to protein structures. We observe that the mixtures clearly describe the frequently occuring helical and β-strand regions in protein structures using suitable mixture components.

We contrasted the FB_5 mixtures with vMF mixtures of the protein directional data. The inferred vMF and FB_5 mixtures have 35 and 23 components respectively. Because the vMF is a special case of a FB_5 distribution, we observe that the same data is modelled using a greater number of vMF components. The vMF distributions are best suited to model data that is symmetrically distributed around a mean direction. However, the empirical distribution of protein directional data is predominantly asymmetrical. Hence, more number of vMF distributions are required to model an asymmetrical region. In comparison, the FB_5 distributions can effectively model asymmetrically distributed data. Consequently, we obtain fewer number of FB_5 mixture components. Furthermore, we note that both the vMF and FB_5 mixtures are superior to a uniform description of the protein directional data while we demonstrated that FB_5 mixtures supersede the vMF mixtures in terms of effectively describing the data (Kasarapu, 2015).

We finally considered the modelling of protein dihedral angles using mixtures of BVM distributions (see Section 6.4.2). The empirical distribution of the pairs of dihedral angles represented on a toroidal surface clearly suggests correlation between the angle pairs. As such, the BVM Sine mixtures are appropriate. The search method to infer the optimal number of mixture components resulted in a 21-component BVM mixture. In comparison, we use the BVM Independent distributions to model the same correlated data. The search method inferred a 32-component BVM Independent mixture. As in the case of vMF and FB_5 mixtures, we notice similarities in the description of the correlated dihedral angle data with greater number of components using the BVM Independent distributions. This is because the Independent variant is a specific case of the BVM Sine distribution. Naturally, an improved description of the dihedral angles would be achieved due to the Sine variant.

Both the BVM Independent and the Sine mixtures effectively model the dihedral angle space. The ability of the search method to correctly identify components corresponding to the regions of possible protein configurations is remarkable. This is more so because our search method does not rely on anything other than the MML framework to evaluate competing mixture models. In all the above considered directional probability distributions, the underpinning idea is based on model selection using the MML framework.

Chapter 7

MML model selection applied to function approximation

7.1 Introduction

The previous chapters have focused on extending and using the MML framework for model selection using probability distributions. Mixtures of probability distributions were used to model data distributed in the Euclidean space (Chapter 5) as well as data distributed on compact manifolds such as spheres and tori (Chapter 6). In the context of mixture modelling, the model selection problem refers to the inference of an optimal mixture that best describes the data. This entailed inferring the number of mixture components as well as the mixture parameters. In this chapter, the MML inference framework is applied to the problem of determining the optimal number of terms used in the infinite series approximation of a periodic function.

Function approximation involves representing a target function using a combination of *simpler* functions, that is, those whose mathematical forms are inexpensive to evaluate. The target function could have a mathematical form which may not be trivially computed. Some well known examples include the standard trigonometric functions, the exponential and the logarithm functions. These are expressed as an infinite summation of simple terms. Functions such as the Bessel functions or the normalization constant of an FB_5 distribution (Equation 4.18) are relatively more complex. However, even these are decomposed as simpler algebraic forms. Some examples of using finite series expansions can be found in applications such as regression analysis and X-ray crystallography, where the number of terms in the underlying mathematical functions has to be selected.

The target functions that generate the observed data are often unknown, in which case it is required to approximate the function using the available data. The function that is considered a suitable fit to the given data needs to be determined. In this chapter, the specific case of fitting the data using regression analysis is discussed (Section 7.2). The method assumes an underlying data generation process and minimizes the error to optimize for that process. As an example, it can be assumed that the process corresponds to a polynomial whose coefficients are determined by *optimally* fitting the known data to the polynomial function of some degree.

In this chapter, the regression analysis is carried out alongside function approximation using *orthogonal basis* functions, as discussed in Section 7.3. Specifically, the problem of linear regression is considered where the target function is approximated by a linear combination of orthogonal functions. Any *periodic* function may be decomposed as a linear combination of orthogonal basis functions. This approximation leads to an infinite summation, and for practical purposes requires truncation to a certain number of terms. Clearly, a longer series expansion results in improved goodness-of-fit. At the same time, a longer series could be unduly complex. The trade-off is similar to the trade-off associated with selecting an optimal number of components in the context of mixture modelling.

The type of orthogonal functions considered in this chapter are the sines/cosines and Legendre polynomials. These trigonometric functions correspond to the Fourier series decomposition of a periodic function. The consideration of Fourier analysis is motivated by its widespread use in a number of applications in sciences and engineering[1]. In particular, it plays an important role in X-ray crystallographic studies, where the intensity values of the diffracted X-rays of crystallized protein molecules are converted to 3D models using an inverse Fourier transform (Sherwood and Cooper, 2010), thus, having direct implications in protein structural biology.

The problem of selecting the optimal number of terms in the series expansion of a function has motivated the use of the MML framework. The encoding of the number of terms and their coefficients in the expansion corresponds to the first part of the message. The second part of the message corresponds to the error resulting due to the use of this expansion. According to the MML framework, the two-part message length is minimized, and the optimal number of terms in the expansion are determined.

The previous literature on MML-based function approximation is limited to univariate polynomial model selection. The MML framework to infer polynomials of suitable degree was first employed by Wallace (1997). They compared the performance of the method with the contemporary state of art (Cherkassky et al., 1997), and demonstrated that the MML-based method fares better. Further research on the applicability of MML to polynomial regression was carried out by Viswanathan and Wallace (1999) and by Fitzgibbon et al. (2002). A general linear regression analysis problem using the MML framework was previously proposed (Wallace, 2005; Makalic and Schmidt, 2006; Schmidt and Makalic, 2009). In this chapter, the linear regression analysis is employed to the problem of function approximation using orthogonal basis functions.

The rest of the chapter is organized as follows: Section 7.2 introduces the regression analysis using the MML framework. Section 7.3 briefly discusses the orthogonal basis functions, and the different forms considered in this chapter. Finally, Section 7.4 presents an experimental evaluation where we demonstrate how the regression analysis in conjunction with the MML framework framework is able to select the optimal number of terms to be used in approximating a periodic function.

7.2 Regression analysis

7.2.1 Problem framework

Regression analysis is a statistical technique to study the relationship among variables. Given data $\mathcal{D} = \{(x_1, y_1), \ldots, (x_N, y_N)\}$, comprising of N observations, the goal is to model the relationship between the *dependent* variable y and the *independent* variable x. A regression problem refers to the approximation (with minimal error) of the data with an assumed mathematical function.

Modelling these observations using a regression model is based on the belief that there is a target function f that maps each x_i to a corresponding function value $f(x_i)$ (Cherkassky et al., 1997). However, in an experiment where the function values are recorded, there is often a measurement error associated with each value. Hence, what are supposed to be exact function values are usually function values with added *noise*. Each x_i, therefore, has a corresponding noise altered y_i value. Although it is desirable to know the actual mathematical form that describes the data, in the absence of any extra information and also because of the random noise involved, one can only approximate the given data with an assumed function.

Given a set of data points, each observed y_i is treated as a deviation from the true function value $f(x_i)$, that is,

$$y_i = f(x_i) + \delta_i \tag{7.1}$$

where δ_i is the noise added to the i^{th} function value. The noise is modelled as a random variable sampled from a univariate Gaussian distribution with zero mean and standard deviation σ, that is, $\delta_i \sim \mathcal{N}(0, \sigma^2)$.

[1] `https://en.wikipedia.org/wiki/Fourier_analysis`

The target function $f(x_i)$ is modelled by an assumed series expansion of the form $g(x_i, \mathbf{w})$ given by

$$f(x_i) \approx g(x_i, \mathbf{w}) = \sum_{j=1}^{K} w_j \phi_j(x_i) \tag{7.2}$$

where $\mathbf{w} = \{w_1, \ldots, w_K\}$ is the vector of weights, and ϕ_j, $1 \leq j \leq K$ are the *basis functions*. In this chapter, ϕ_j are taken as orthogonal functions (Section 7.3), and, hence, f is modelled by their *linear* combination. Such a formulation is referred to as the *linear regression model*. It is important to note that ϕ_j itself could be a non-linear function. Hence, Equation 7.1 becomes

$$y_i \approx \hat{y}_i = \sum_{j=1}^{K} w_j \phi_j(x_i) + \delta_i, \quad i = 1, \ldots, N \tag{7.3}$$

As δ_i is assumed to be a Gaussian random variable with zero mean, the error term $y_i - \hat{y}_i$ is also normally distributed. This property is used to formulate the likelihood expression (Equation E.2).

The *problem* at hand is the determination of the optimal K for approximating a function using the linear model for regression. The approximation by a linear combination of K basis functions yields "a fit" to the data. The root mean squared error (RMSE) is a measure of the goodness-of-fit and is computed as

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2} \tag{7.4}$$

The RMSE decreases as the number of terms K in the approximating function increases. A greater value of K would fit a model with reduced RMSE at the cost of increased complexity. This leads to the problem of *overfitting* and, hence, the optimal balance that resolves the trade-off of model complexity and the goodness-of-fit needs to be achieved. A related discussion about the variation in error while approximating a function with increased number of orthogonal basis terms is available[2].

7.2.2 MML approach

The MML framework is applied in the current context of function approximation. For a given number of terms K in the series expansion of a target function (Equation 7.2), the message length formulation comprises of two parts.

- *First part*, corresponding to the encoding of the parameters, namely, the weights $\mathbf{w}$ and the standard deviation σ of the noise.
- *Second part*, corresponding to the encoding of the data, that is, the estimated values $\hat{y}_i$ using the parameters.

The formulation of the message length expression followed by a complete derivation of the MML estimates of the parameters, based on the Wallace and Freeman (1987) approximation (Section 2.4.2), is given by Wallace (2005). The derivation is included in Appendix E.1.

The minimization of the total message length over all possible values of K gives the optimal value of the number of terms to be used in the approximation (Equation 7.2). It is expected that the total message length would continuously decrease as K increases, until a minimum value is obtained, and subsequently increases with increasing K. The value of K corresponding to the minimum total message length is the optimal value at which the infinite series expansion should be truncated. This problem is explored using orthogonal basis functions and, as specific examples, discussed for when the basis functions are: (1) sines and cosines (Fourier decomposition), and (2) Legendre polynomials.

[2] `http://allisons.org/ll/Maths/Orthogonal/`

7.3 Orthogonal basis functions

The orthogonal basis functions are real-valued functions defined on the domain $[a, b]$ and which satisfy the orthogonal property as discussed here. For functions ϕ_i and ϕ_j, let their inner product be defined as

$$< \phi_i, \phi_j >= \int_a^b \phi_i(x)\phi_j(x)dx$$

and let the *norm* be denoted by $\|\phi_j\|=< \phi_j, \phi_j >$. The set of functions ϕ_j, $\{j = 1, 2, 3, \ldots\}$ are said to be *orthogonal* if $< \phi_i, \phi_j >= 0, \forall\, i, j$ such that $i \neq j$, and *orthonormal* if they are orthogonal $\|\phi_j\|= 1\, \forall\, j$.

The advantage of using orthogonal bases is that they facilitate the expression of functions as linear combinations of the elements in the orthogonal bases set. As part of the regression problem description, the unknown target function f is expressed as a series in terms of a set of simple functions (Equation 7.2). The computation of the corresponding weights w_j requires the solution to the resulting linear regression problem with an objective to minimize the sum of squared errors (Equation 7.4). Further, as detailed in Appendix E.1, the computation of the Fisher information becomes tractable owing to the properties of the orthonormal basis functions (see Equation E.8).

Two separate orthogonal basis sets, corresponding to the decomposition of a function using (1) trignometric functions (Fourier series), and (2) Legendre polynomials are considered in this section.

7.3.1 Fourier series

Fourier (1822) observed that any periodic function, possibly discontinuous, can be decomposed as an infinite summation of a series of relatively simple functions of sines and cosines. Such a decomposition using the sequence of sines and cosines refers to the *Fourier series* decomposition (Weisstein, a). Any periodic function $g(x)$ with a time period T can be represented using a Fourier series as

$$g(x) = a_0 + \sum_{j=1}^{\infty} \left(a_j \cos \frac{2j\pi x}{T} + b_j \sin \frac{2j\pi x}{T} \right) \tag{7.5}$$

where a_j and b_j are Fourier coefficients determined using the orthonormal property of the sines and cosines, and are

$$a_j = \frac{2}{T} \int_0^T g(x) \cos \frac{2j\pi x}{T} dx \quad \text{and} \quad b_j = \frac{2}{T} \int_0^T g(x) \sin \frac{2j\pi x}{T} dx \tag{7.6}$$

A Fourier decomposition can be treated as a specific case of the regression problem (Section 7.2.1), where a periodic function is approximated by a linear combination of sines and cosines. Drawing correspondences between the linear model for regression (Equation 7.2) and the Fourier decomposition (Equation 7.5), note that the unknown target function $f(x)$ is represented using the Fourier series. Comparing the approximation of the target function $f(x) \approx g(x, \mathbf{w})$, given by Equation 7.2, with the Fourier decomposition (Equation 7.5), the corresponding orthonormal basis set is

$$\phi_j(x) = \begin{cases} 1 & \text{if } j = 1 \\ \sqrt{\frac{2}{T}} \sin \left(\frac{j\pi x}{T} \right) & \text{if } j \text{ is even} \\ \sqrt{\frac{2}{T}} \cos \left(\frac{(j-1)\pi x}{T} \right) & \text{if } j > 1 \text{ and is odd} \end{cases}$$

The approximating function $g(x, \mathbf{w})$ is the linear combination of orthogonal functions $\phi_j\, \forall\, j$, which in this case correspond to sines and cosines. The weights of the regression model $\mathbf{w}$ are related to the sequence of coefficients $\{a_j, b_j\}$, $\forall\, j$ in the Fourier expansion.

As the Fourier expansion includes an infinite series, the number of terms K at which the series needs to be truncated is determined previously as discussed in Section 7.2.2. The weights inferred as part of the regression analysis, will differ from the true Fourier coefficients because the infinite series is truncated using the first K terms, and also because of measurement error in the form of random noise.

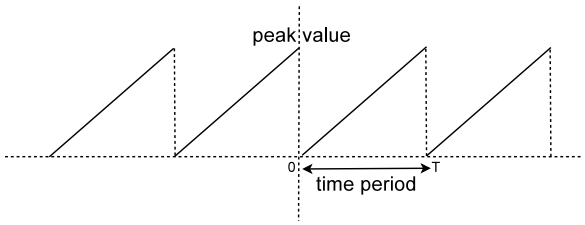
The current analysis pertains to the decomposition of a periodic function using the above mentioned orthogonal basis functions. Some periodic functions considered herein are the sawtooth, square, triangle, and parabolic functions. Note that the sawtooth and square functions are discontinuous. However, their analysis is still possible for piecewise-continuous functions, within the domain where the individual functions are continuous. This domain is characterized by the *time period* T of each of the functions (also referred to as *waveforms*).

The following examples are provided to illustrate the various forms of the periodic functions considered in our analysis; their infinite Fourier expansions are also given alongside, with the true coefficients computed using Equation 7.6. The periodic functions, namely, sawtooth, square, triangle, and parabola are considered here with their characterizing peak values at 1 and a time period of $T = 2$.

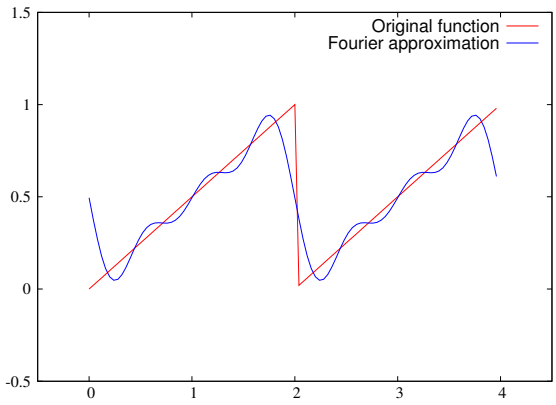
In these examples, the true periodic functions and their infinite series expansions are approximated to a finite number of terms in the Fourier expansion as demonstrated below. Their respective functional forms and *one* of their Fourier approximations, with a fixed j are shown below. In these examples, the series is truncated at $j = 3$, corresponding to $K = 7$ terms (as per Equation 7.5).

1. *Sawtooth*

$$g(x) = \begin{cases} \frac{x}{T} & \text{if } 0 \le x < T \\ g(x-T) & \text{if } x \ge T \\ f(x+T) & \text{if } x < 0 \end{cases} \qquad g(x) = \frac{1}{2} - \frac{1}{\pi}\sum_{j=1}^{\infty}\frac{1}{j}\sin\left(\frac{2j\pi x}{T}\right) \tag{7.7}$$



(a) Waveform

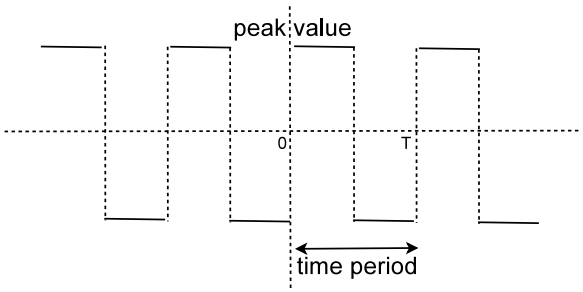


(b) A Fourier approximation with j=3

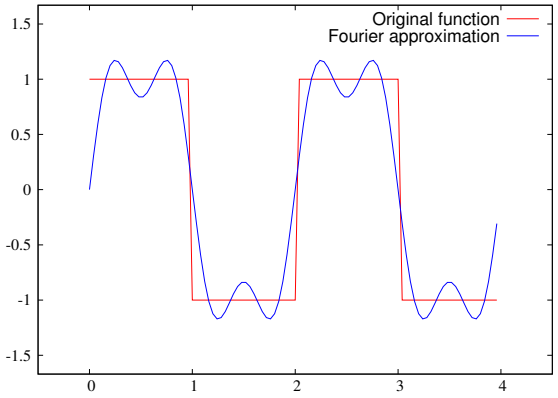
Figure 7.1: Sawtooth function & its Fourier approximation

2. *Square*

$$g(x) = \begin{cases} +1 & \text{if } 0 \le x < \frac{T}{2} \\ -1 & \text{if } \frac{T}{2} \le x < T \\ g(x-T) & \text{if } x \ge T \\ g(x+T) & \text{if } x < 0 \end{cases} \qquad g(x) = \frac{4}{\pi}\sum_{j=1,3,5,\ldots}^{\infty}\frac{1}{j}\sin\left(\frac{2j\pi x}{T}\right) \tag{7.8}$$



(a) Waveform

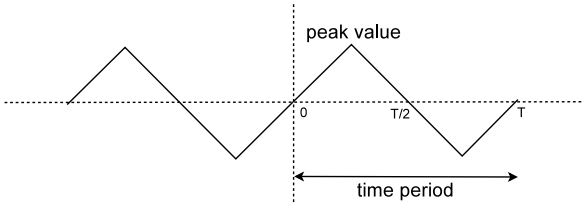


(b) A Fourier approximation with j=3

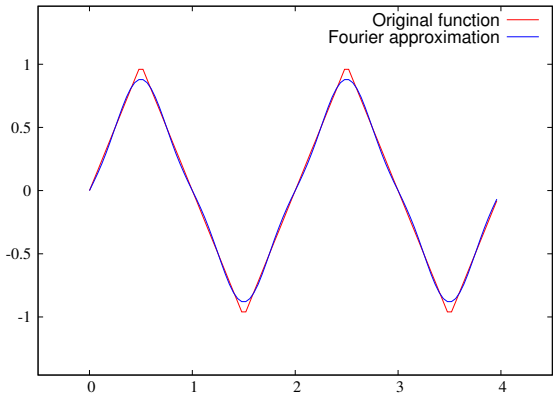
Figure 7.2: Square function & its Fourier approximation

3. *Triangle*

$$g(x) = \begin{cases} \frac{4x}{T} & \text{if } 0 \le x \le \frac{T}{4} \\ \frac{4}{T}\left(\frac{T}{2} - x\right) & \text{if } \frac{T}{4} < x < \frac{3T}{4} \\ 4\left(\frac{x}{T} - 1\right) & \text{if } \frac{3T}{4} \le x \le T \\ g(x - T) & \text{if } x \ge T \\ g(x + T) & \text{if } x < 0 \end{cases} \qquad g(x) = \frac{8}{\pi^2} \sum_{j=1,3,5,\ldots}^{\infty} \frac{(-1)^{\frac{j-1}{2}}}{j^2} \sin\left(\frac{2j\pi x}{T}\right) \quad (7.9)$$



(a) Waveform

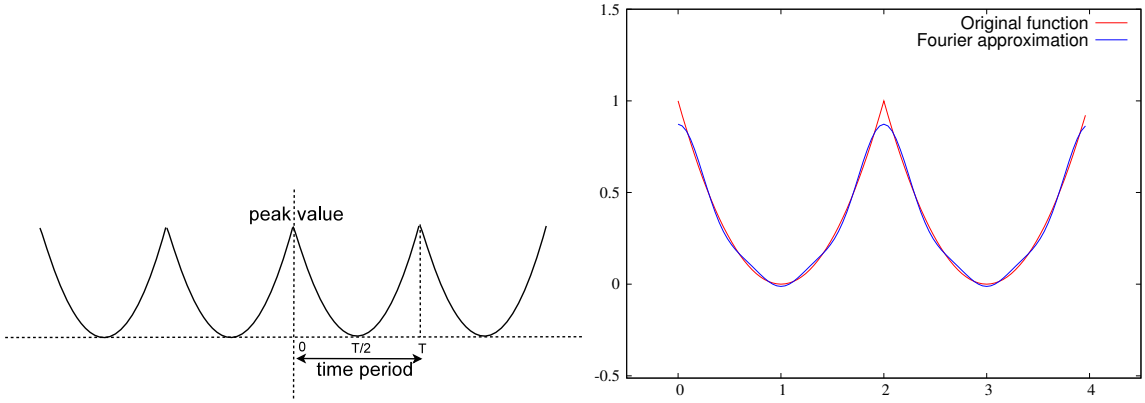


(b) A Fourier approximation with j=3

Figure 7.3: Triangle function & its Fourier approximation

4. *Parabolic*

$$g(x) = \begin{cases} \frac{4}{T}\left(x - \frac{T}{2}\right)^2 & \text{if } 0 \le x \le T \\ g(x - T) & \text{if } x > T \\ g(x + T) & \text{if } x < 0 \end{cases} \qquad g(x) = \frac{1}{3} + \frac{4}{\pi^2} \sum_{j=1}^{\infty} \frac{1}{j^2} \cos\left(\frac{2j\pi x}{T}\right) \quad (7.10)$$



(a) Waveform (b) A Fourier approximation with j=3

Figure 7.4: Parabolic function & its Fourier approximation

7.3.2 Legendre polynomials

The Legendre polynomials were first introduced by Legendre (1785). The series of functions, including the recurrence relation to obtain higher order basis functions, are given below (Weisstein, b).

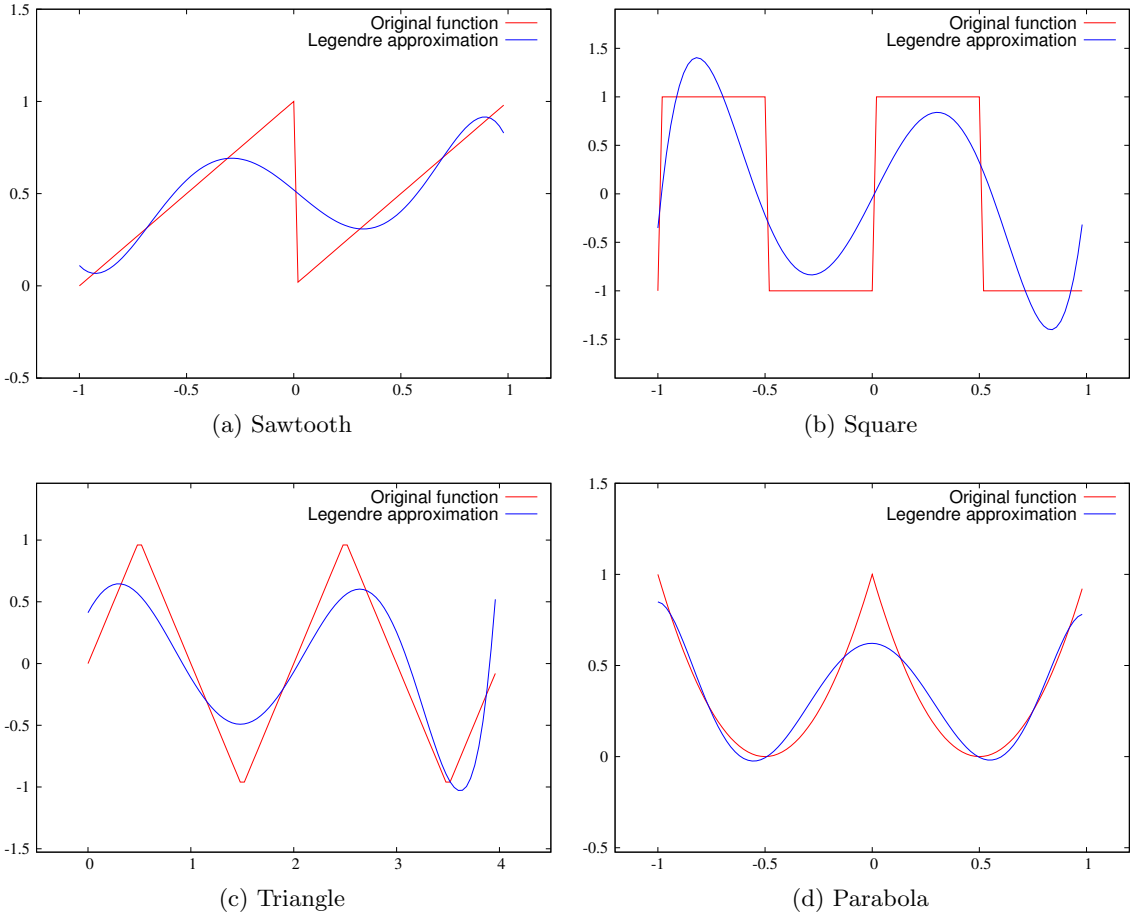
$$\phi_j(x) = \begin{cases} 1 & \text{if } j = 1 \\ x & \text{if } j = 2 \\ \dfrac{3x^2 - 1}{2} & \text{if } j = 3 \\ \dfrac{(2j-3)\, x\, \phi_{j-2}(x) - (j-2)\, \phi_{j-3}(x)}{j-1} & \text{if } j > 3 \end{cases}$$

The Legendre polynomials are considered here in order to provide a contrast to the orthogonal basis of sines and cosines. The Legendre polynomials are orthogonal in the range $[-1, 1]$. The determination of the optimal number of terms to approximate a target function using Legendre polynomials is done under the MML framework. The approximations are discussed with respect to the previously considered periodic functions, that is, the sawtooth, square, triangle, and parabola waveforms. As Legendre polynomials are orthogonal in the range $[-1, 1]$, the domain of the periodic functions is also between -1 and 1. The functions considered have a peak value of 1 and a time period $T = 1$. The respective functional forms and their approximations using the first 7 terms in the sequence are shown in Figure 7.5.

7.4 Experimental evaluation

The following set of experiments demonstrate the ability of the MML framework to identify the number of orthogonal basis functions in the approximation of a periodic function. The function approximation problem is treated as a linear regression problem involving K terms, whose parameters are inferred by MML estimation. Further, the MML criterion is used in determining an optimal K by formulating a two-part message length that includes the encoding cost for the regression parameters and the cost of encoding the data using those parameters (corresponding to the error in regression analysis).

For each of the periodic functions, there are an infinite number of Fourier approximations possible depending on the terms in their corresponding Fourier series. As discussed previously, a higher number of terms (K) results in a better fit (least approximation error) but at the cost of increased complexity of the fit. For each periodic function, data is generated randomly and, to each data sample, random noise is added (see Equation 7.1). The experiment is run for different values of the number of data samples N and varying values of the standard deviation σ of the noise. Each set



(a) Sawtooth

(b) Square

(c) Triangle

(d) Parabola

Figure 7.5: Approximations of the periodic functions using the first 7 Legendre polynomials

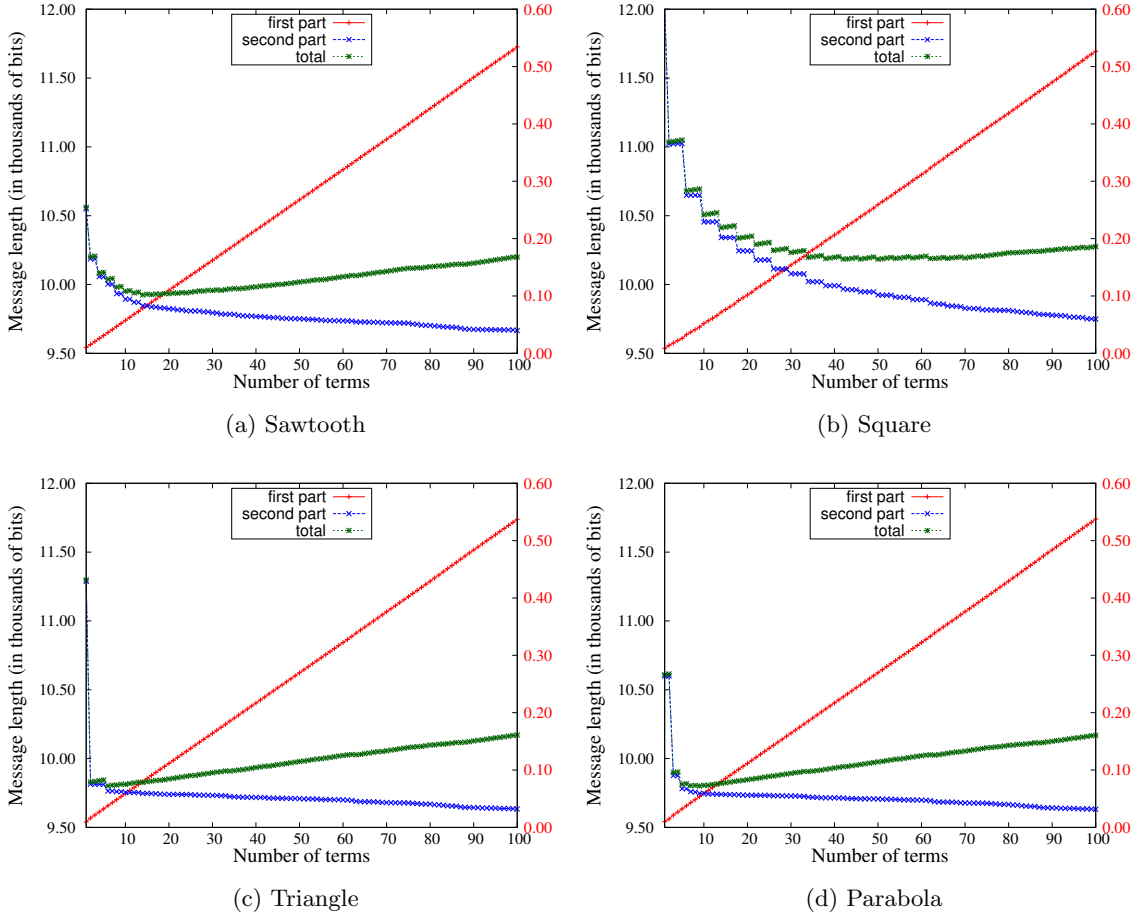
of (N, σ) constitutes an experimental setup. The values of N considered are $\{100, 1000, 10000\}$ and $\sigma \in \{0, 0.1, 0.2, 0.3, 0.4, 0.5\}$.

For each experimental setup, the two part message length is computed for different number of terms K. Each value of K requires the estimation of parameters $(\mathbf{w}, \sigma)$. The parameters and the data given those parameters are encoded as discussed in Appendix E.1. The combined message length is computed and minimized to give the MML estimates $\hat{\mathbf{w}}, \hat{\sigma}$ (Equation E.7). Using these estimates, the two parts of the message are computed using Equations E.5 and E.6 respectively. These are plotted individually along with the total message.

7.4.1 Regression fit using sines & cosines as orthogonal basis

The results for regression analysis and subsequent periodic function approximation using sines and cosines are illustrated in Figure 7.6. The results are plotted for $N = 1000$ and a standard deviation of noise $\sigma = 0.2$. The plots show the comparison of message lengths (part 1 & part 2) for K ranging from 1 to 100.

A common trend that can be observed in the plots corresponding to the various periodic functions in Figure 7.6 is that as the K increases, the statement cost of the parameters (part 1: red curve) steadily increases. Also, the statement cost of the data given the parameters (part 2: blue curve) decreases. The first part corresponds to the model complexity (explanation of the parameters) and the second part corresponds to the fit (error in approximation). The total message length (green curve), however, first decreases, reaches a minimum and then steadily increases. This shows that a complex model (with large K) need not be the best choice to model a set of observations. This addresses the tradeoff that arises from choosing a model against the associated error.



(a) Sawtooth

(b) Square

(c) Triangle

(d) Parabola

Figure 7.6: Regression fit using Fourier series

The drop in the total message length is abrupt for the triangle and parabolic waveforms as compared to the sawtooth and square functions. This is because, the weights of the orthogonal functions that are inferred correspond to the Fourier coefficients of the respective waveforms. For the sawtooth and square waves, the Fourier coefficients are inversely related to the order j (Equations 7.7 and 7.8). The Fourier coefficients of the triangle and parabolic functions are, however, inversely related to j^2 (Equations 7.9 and 7.10). Hence, the magnitude of the coefficients of the corresponding orthogonal functions decreases rapidly for the triangle and parabolic waveforms. This leads to a steep fall in the prediction error. As it can be seen, the encoding length of the first part is similar for all the four functions but the second part encoding has a sharp decrease early on.

The coefficients corresponding to the sine terms are non-zero in the case of Fourier decomposition of the sawtooth function. The corresponding coefficients of the cosine terms are zero (Equation 7.7). Hence, the weights inferred of the cosine terms, as part of the regression analysis, will be close to zero. Therefore, the addition of a cosine term does not cause much reduction in approximation error as the weight inferred of the additional term is nearly zero and, thus, the second part of the message remains almost unaltered. The first part of the message, however, increases because this additional weight needs to be communicated. As a result, the total message length remains almost the same over successive values of the number of terms. A similar argument holds for the parabolic function whose Fourier decomposition has the coefficients corresponding to the sine terms as zero.

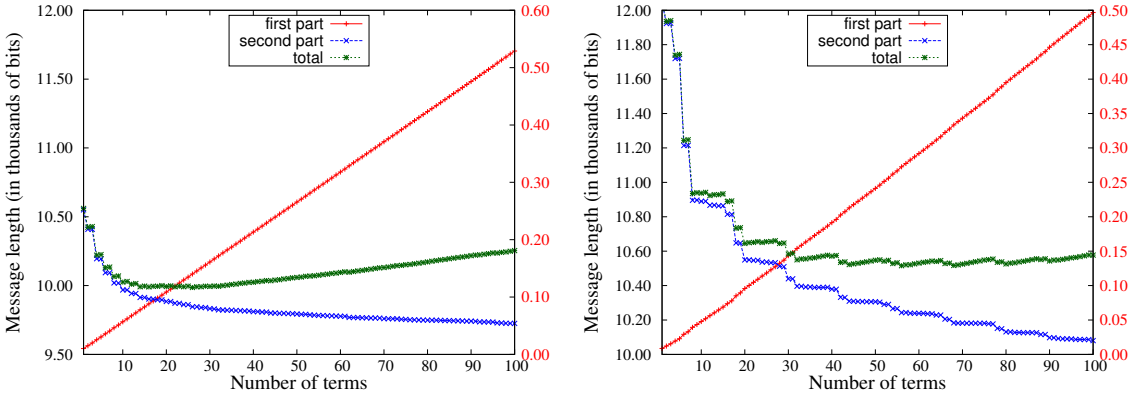
The square wave, however, is approximated by sine terms of odd order (Equation 7.8). This means that if the coefficient corresponding to K (an odd sine term is non-zero), the coefficients for the next three terms in the sequence will be zeroes. Therefore, the second part of the message length does not change significantly from K to $K + 3$. This behaviour is clearly visible for the square wave, in

Figure 7.6(b). The characteristic zig-zag pattern (blue curve) implies the message length is constant for a series of three consecutive K values. A similar argument holds for the triangle wave although it is not clearly visible from the plot.

A similar trend in the plots is observed for different number of data samples $N \in \{100, 10000\}$ and for varying levels of standard deviations of noise $\sigma \in \{0, 0.1, 0.3, 0.4, 0.5\}$.

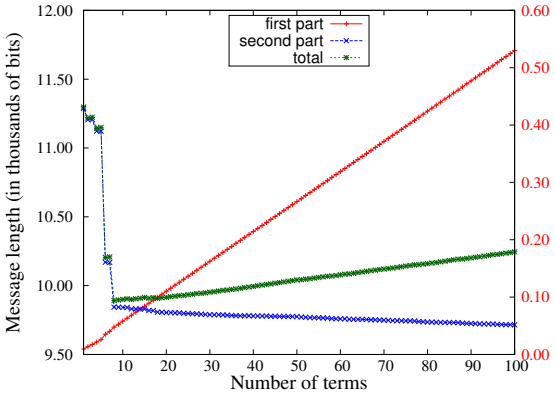
7.4.2 Regression fit using Legendre polynomials as orthogonal basis

The experimental setup for approximating using Legendre polynomials is similar to the previous case of regression fit using sines & cosines as orthogonal basis. The results are discussed for a sample of $N = 1000$ generated from each of the periodic functions. The regression analysis involves approximating the data using a series of Legendre polynomials. A comparison of the message lengths for the first 100 Legendre polynomials is shown in Figure 7.7. As before, depending on the number of terms considered, the fit to the data varies. The optimal fit is determined using the MML criterion. The total message length first decreases and then gradually increases. The lowest point refers to the value of the number of terms at which the message length is globally minimum.

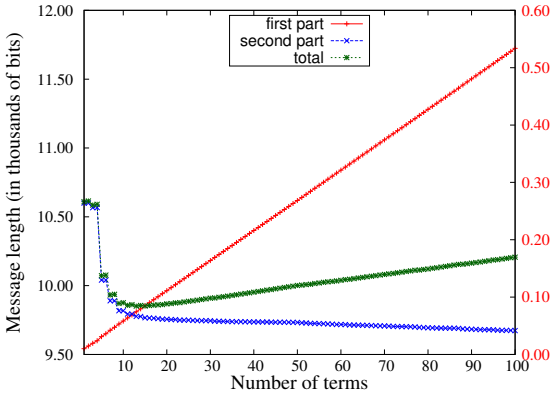


(a) Sawtooth

(b) Square



(c) Triangle



(d) Parabola

Figure 7.7: Regression fit using Legendre polynomials

7.5 Summary

In this chapter, we showed how the MML inference framework can be used in function approximation. Specifically, the problem considered is identifying the number of terms at which an infinite series expansion of a periodic function can be truncated. For data that is generated from a periodic

function, MML-based analysis was used to estimate the parameters of the linear regression model. With increasing number of parameters in the regression model, the model complexity increases due to a proportional increase in the encoding cost of the parameters. Consequently, the fit to the data improves. This corresponds to the second part of the message in the MML framework.

The application of the MML framework in regression analysis using orthogonal basis functions demonstrates the ability of the inference framework to objectively address the trade-off due to model complexity and the goodness-of-fit to the data. In the experiments carried out, it is observed that the MML inference process is able to identify a truncation point that corresponds to the least total message length.

Chapter 8

Modelling protein folding patterns using Bézier curves

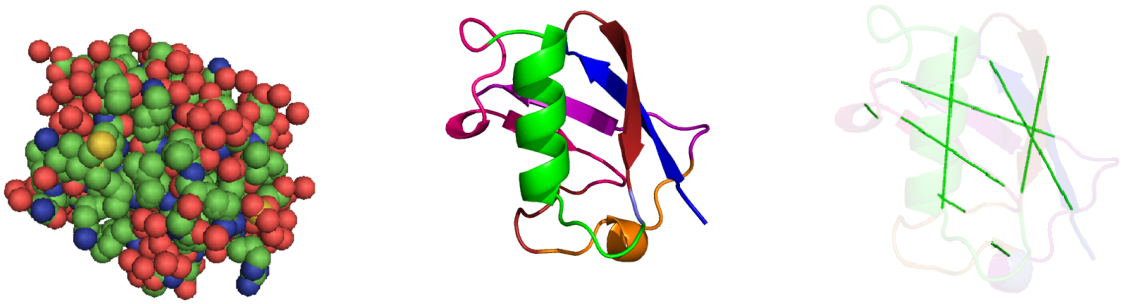
8.1 Introduction

The focus of all previous chapters was on extending and using the MML framework for its use in the statistical inference of probability distributions. In this chapter, the inference framework is used in the abstract representation of a collection of three-dimensional (3D) points by a series of non-linear parametric curves. This problem is discussed here in order to provide a contrast between how the generalized MML framework can be used in different scenarios: (1) statistical inference of probability distributions, their parameters and the optimal number (in the case of mixture models), and (2) modelling some observed data with abstract representations.

In both these scenarios, the goal is to distinguish between several competing hypotheses and to select the optimal one. In the first case, the hypotheses were probabilistic models of either the same type (for example, Gaussian) with different parameters, or of mixtures with different number of components. In the second scenario, there are numerous combinations of parametric curves that can be used as representations to model the given data. The MML criterion is used to select an optimal representation, in this case, by formulating a two-part message, where the first part corresponds to the cost associated with selecting a certain representation (model complexity), and the second part corresponds to the goodness of fit to the data.

The data being modelled can be generic and may correspond to a collection of 3D points with some observable non-linear geometry. As an application, protein structural data is considered because the interactions between the constituent atoms result in proteins folding into complex 3D shapes (Lesk, 2004). Additionally, modelling the protein coordinate data is a good case study as the resulting representations can be leveraged in applications involving the identification of proteins with similar structural properties.

Recall from Section 6.3.3, where we explained the basic backbone structure of a protein as a chain of amino acid residues. These residues form a detailed representation of the protein structures at an atomic level (see Figure 8.1a). However, such a representation does not yield much information about the underlying geometric structure of proteins. It is a well established fact that proteins are described using their secondary structure and are predominantly made up of α helices and β strands (Edsall et al., 1966), shown in Figure 8.1(b). In literature studying the principles of protein architecture, the focus has been mainly on studying arrangements of secondary structure elements (Louie and Somorjai, 1983; Barlow and Thornton, 1988; Chothia et al., 1981; Richardson, 1977) in folds. The essence of protein folding patterns is captured by the geometry of their standard secondary structural elements: helices and strands of sheet (Chothia and Finkelstein, 1990; Lesk, 1995). Therefore, most of the methods exploring protein folding patterns involve abstraction of their protein 3D structures at the level of secondary structure.



(a) (b) (c)

Figure 8.1: Example representations of a protein structure at (a) the residue level, (b) secondary structure level, and (c) an abstraction based on the secondary structure.

Simplified representations of folding patterns, like the ones at the level of secondary structures, are tremendously useful to understand the architecture and topology of protein structures. They are commonly achieved by replacing each secondary structure segment by straight lines (Chothia et al., 1981; Cohen et al., 1981; Lesk, 1995). This has resulted in very useful and compact summaries of protein folding patterns (see Figure 8.1c). With the rapidly growing world wide protein data bank (wwPDB), simplified representations allow for the definition of efficient and effective methods for large-scale search, alignment and classification, handling the whole corpus of structures.

Almost universally, current methods to represent protein folding patterns rely on secondary structural elements (Chothia et al., 1981; Cohen et al., 1981; Lesk, 1995). Although the representations at the level of secondary structural elements are very useful and compact, they remain inconsistent and lossy in terms of the structural information they capture (Levitt and Greer, 1977; Cuff and Barton, 1999). On average, the standard secondary structures account for 60-70% of globular protein chains, leaving the structural information in the remaining 30-40% of the chain entirely ignored in such representations. Thus is further exacerbated due to the lack of consensus in defining secondary structures (Colloc'h et al., 1993).

As a result, the current state-of-the-art abstractions of protein structures at a secondary structural level do not sufficiently capture the geometry of interactions within a protein structure. While a few rigorous attemps have been made, none of them fully explain the geometric patterns satisfactorily (Taylor et al., 1983; Richards and Kundrot, 1988; Sklenar et al., 1989; Dupuis et al., 2004; Majumdar et al., 2005; Ranganathan et al., 2009).

A few methods have been proposed to abstract protein folding patterns that capture their essence without discarding structural information (Taylor, 2001; Konagurthu et al., 2011). Mainly, these methods summarize a given protein backbone using *piecewise* line segments. These representations are independent of the notion of secondary structures and, hence, allow for the consideration of regions that do not belong to the conventional classes of helices and strands (Taylor, 2001), the information that is lost in abstractions at the secondary structure level. However, the main disadvantage of using piecewise linear abstractions relates to the fact that structures are flexible and undergo plastic deformations in protein evolution. Representing the protein backbone using rigid lines does not accommodate for the flexibility and plasticity required to describe protein folding patterns concisely. The current work addresses and rectifies this limitation.

The primary goal of any abstraction should be to maximize the economy of description of a protein structure while ensuring that the essence of its folding pattern is retained (Taylor, 2001; Konagurthu et al., 2011). In this chapter, a mathematically rigorous approach is proposed to represent any protein structure using arbitrary order non-linear parametric curves. The approach relies on the information theoretic based MML inference.

We employed the MML principle to select the assortment of non-linear curves to best compress the given protein coordinate data. While the method is applicable to any class of parametric curves, this work specifically focuses on Bézier curves for their robustness in modelling regions of protein structures effectively.

Bézier curves are smooth and continuous, and model the plasticity commonly observed in proteins. These curves can be scaled to any order and are simply defined by their *control points*. The number of control points determines the degree (or order) of the curve. A linear Bézier curve has two control points and is of degree 1, a quadratic Bézier curve has three control points and is of degree 2, a cubic Bézier curve is of degree 3 containing four control points and so on. In general, the first and last control points of Bézier curves always lie on their respective curve. The intermediate control points (where they exist) lie away from the curve. The freedom to move the intermediate control points give Bézier curves their flexibility and, making them ideal to describe the observed plasticity in proteins.

Our method relies on *dissecting* (or segmenting) any given protein structure into non-overlapping, variable-length regions (or segments), each of which is assigned to a Bézier curve of an arbitrary degree (or complexity) consistent with the MML framework. Each segmented region in the dissection has a statement cost corresponding to the amount of information (measurable in *bits*) required to losslessly encode the coordinate information in that region, using its assigned Bézier curve as the model for compression. (Hence, we use the term *code length* to define the information cost associated with each dissected region.) Furthermore, a dynamic programming based search strategy is designed to find the optimal dissection (and its corresponding Bézier curve assignment) that results in the shortest lossless encoding of protein coordinates using an ensemble of Bézier curves, where each curve in the ensemble defines a particular dissected region in the protein.

The details of the approach are elaborated in the following sections. Section 8.2 describes the problem of representing the given data using Bézier curves in terms of the MML framework. The section explains the mechanics involved in losslessly encoding the Bézier abstractions and subsequently formulating a two-part message length to encode the data. Section 8.3 describes the method used to determine the optimal Bézier abstraction by constructing code length matrices used to explain the different protein segments. Section 8.4 evaluates the proposed approach. A case study is first presented where a protein is segmented using Bézier curves based on the proposed method. In this example, segmentations obtained by using different forms of Bézier curves (linear to cubic) are contrasted with each other. These are also compared with traditionally used segmentation methods based on secondary structure assignment. The experiments section further builds methods to compare protein structures based on the resulting Bézier abstractions. These methods are evaluated against the commonly used techniques to compare proteins.

8.2 Problem formulation using MML

In this problem, the data considered are the three-dimensional coordinates of a given protein $\mathcal{P}$. Recall from Section 6.3.3 that any protein is represented as the sequence (ordered list) of N three-dimensional points $\{P_1, \ldots, P_N\}$, representing the coordinates of the C_α atoms along the N- to C-terminus of the protein chain.

A non-linear abstraction of $\mathcal{P}$ results in a dissection that is defined as a subsequence containing $k < N$ points from $\mathcal{P}$ denoted as $\mathcal{Q} = \{Q_1 \equiv P_{i_1}, Q_2 \equiv P_{i_2} \ldots, Q_k \equiv P_{i_k}\}$ such that $1 = i_1 < i_2 < \cdots < i_k = N$. Any successive pair of points $(Q_r, Q_{r+1})_{1 \leq r < k} \in \mathcal{Q}$ defines a *contiguous* region in the protein structure whose end points are $Q_r = P_{i_r}$ and $Q_{r+1} = P_{i_{r+1}}$. The term *dissection* is used to indicate the collection of regions defined by $\mathcal{Q}$. Associated with each region $Q_r \cdots Q_{r+1}$ in the dissection is a Bézier curve of some degree θ_r with (Q_r, Q_{r+1}) acting as the start and end control points of that curve. The remaining $(\theta_r - 1)$ control points are determined analytically by minimizing the total least squares errors of the set of points in that region with respect to the Bézier curve.

Translating the problem to the MML context, any dissection $\mathcal{Q}$ (and its corresponding Bézier curve assignment) denotes a *model*, that aims to concisely summarize a given protein structure. The best

dissection in this framework is the one that gives the shortest encoding (that is, the best compression) of the entire set of coordinates in $\mathcal{P}$, over all possible models to describe $\mathcal{P}$. However, unlike in the previous Chapters, we cannot directly employ the Wallace and Freeman (1987) method. We previously modelled the data using probability distributions and their mixtures. In those cases, we were able to explicitly compute the Fisher information that determines the precision of the model's parameters, which is integral to inference as per Wallace and Freeman (1987)'s method (see Section 2.4.2).

In the current context, we have a model defined as a sequence of Bézier curves of varying order. The parameters of the model include the number of curves, the degree and control points of each curve in the sequence. The Fisher information in this context is not explicitly computable, and thus, we cannot use Wallace and Freeman (1987)'s method. However, the general idea of the MML principle rationalized as a communication framework can still be applied. As described in Section 2.4, the communication of data between a hypothetical transmitter and receiver pair involves encoding the model and then the data using the model. The total message length can vary depending on the complexity of $\mathcal{Q}$ and how well it can explain $\mathcal{P}$. A more complex $\mathcal{Q}$ may fit $\mathcal{P}$ better but takes more number of bits to be stated itself. The trade-off comes from the fact that the transmission process requires the encoding of both the model $\mathcal{Q}$ and the data given $\mathcal{Q}$, that is, the model complexity $I(\mathcal{Q})$ and the goodness-of-fit $I(\mathcal{P}|\mathcal{Q})$.

8.2.1 Encoding the model: First part in the MML framework

The first part of the transmission process involves communicating the model, that is, the dissection $\mathcal{Q}$ containing k segments and its corresponding Bézier curve assignment. This is achieved using the following steps

1. *Transmit the number of segments:* A variable length integer code, namely the Elias omega code (Elias, 1975), is used to encode k. This takes $\log^* k = \log k + \log\log k + \cdots$ (over all positive terms) bits to encode k.

2. *Transmit the end points:* The end point of the previous segment in any dissection is also the start point of the current one. (The first segment is a special case where the transmission of the start point can be avoided if the coordinates in $\mathcal{P}$ are translated such that P_1 is always the origin.) The coordinates of the end point of each segment are three real numbers of the form (x, y, z). To transmit these coordinates a bounding box is specified using $(x_{\min}, y_{\min}, z_{\min})$ and $(x_{\max}, y_{\max}, z_{\max})$ which can be determined from the set of all coordinates. Each end point can be stated in $\log V$ bits, where $V = (x_{\max} - x_{\min}) \times (y_{\max} - y_{\min}) \times (z_{\max} - z_{\min})$ is the volume of the box. Hence, to encode the k end points requires a statement cost of $k \log V$ bits.

3. *Transmit intermediate control points:* Associated with any segment $Q_r \equiv P_i \cdots Q_{r+1} \equiv P_j$ is a Bézier curve of degree θ_r, containing $\theta_r - 1$ intermediate control points. The degree θ_r is an integer and takes $\log^* \theta_r$ bits to encode. The spatial positions of the control points are stated using the bounding box approach described above. The length of encoding the $(\theta_r - 1)$ intermediate control points associated with each segment is $(\theta_r - 1) \log V$ bits.

Adding each of the above contributions to the message length required for the first part, the encoding cost of the model is given as

$$I(\mathcal{Q}) = \log^*(k) + k \log V + \sum_{r=1}^{k} (\log^* \theta_r + (\theta_r - 1) \log V) \tag{8.1}$$

8.2.2 Encoding the data given the model: Second part in the MML framework

The second part of the message consists of transmitting the remaining protein coordinates, given the dissection (and assigned Bézier curves) as the model. This is achieved using the following steps

1. *Transmit the number of internal points within a region:* For segment r between P_i and P_j, there are $j - i - 1$ *internal* points that need to be encoded using the assigned Bézier curve as their model of compression. Again, the same integer encoding described previously is used taking $\log^*(j - i - 1)$ bits.

2. *Transmit the coordinates of the internal points:* Given the receiver knows the dissection and its assigned Bézier curves, the internal points corresponding to any segment can be explained as a set of three spatial deviations with respect to its curve. Using these deviations, the receiver can precisely locate (to the stated precision) the position of the internal protein residue. The three sets of spatial deviations are transmitted over a probability distribution, the parameters of which need to be encoded as part of the transmission.

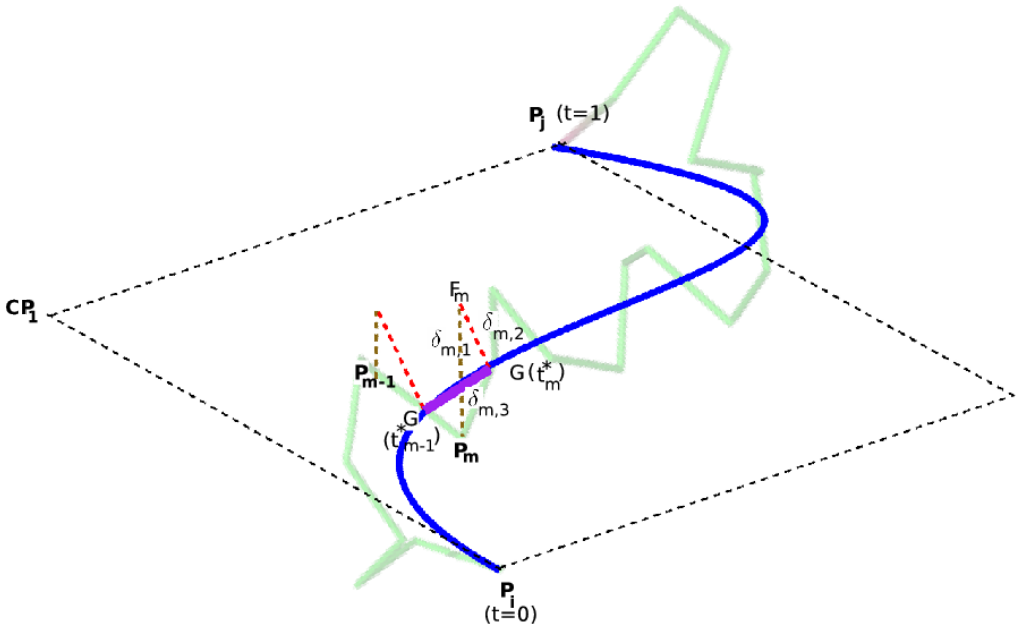


Figure 8.2: Deviations of the internal points of a region with respect to the assigned (cubic) Bézier curve.

Figure 8.2 illustrates the encoding of this part. Let C_{ij} be the curve that abstracts the protein segment between P_i and P_j. This curve is used in conjunction with a plane to explain the internal points that lie between P_i and P_j. To build the plane, consider three non-collinear points P_i, P_j and the first intermediate control point (if it exists) of the assigned Bézier curve. If no such control point exists (that is, when the assigned Bézier curve is of degree 1), use P_{i+1} (and transmit it using the bounding box approach).

Computation of deviations

Figure 8.2 shows two internal points P_{m-1} and P_m within a segment defined by P_i and P_j as its end points. A plane is defined for this segment using P_i, P_j, and CP_1. Each internal point is associated with three spatial deviations that are computed as

1. *Deviation 1:* The first deviation is the orthogonal projection of the internal point onto the defined plane. The foot of the perpendicular line from P_m onto the plane is denoted by F_m. The length of this projection is denoted by a signed deviation $\delta_{m,1}$. The sign is determined by the orientation of the P_m with respect to the plane (that is, above or below defined by the normal vector $\hat{n}$ of that plane).

2. *Deviation 2:* The second deviation denotes the shortest distance from the foot of the perpendicular F_m, to the Bézier curve C_{ij} assigned to this region.[1] Let G_m denote the projection on the curve which results in the shortest distance. The length of this projection is given by the signed deviation $\delta_{m,2}$. The sign is determined by the orientation of F_m with respect to the plane formed by the tangent at G_m and the normal $\hat{n}$.

3. *Deviation 3:* Every point on the Bézier curve, including G_m, is parameterized using t in the range $[0, 1]$. Let t^*_m be the parameter of G_m. Similar to the projection G_m of P_m, the previous intermediate point P_{m-1} has an associated projection G_{m-1}. Let t^*_{m-1} be the parameter corresponding to G_{m-1}. The third deviation is the offset of G_m from G_{m-1}, whose sign is given by the position of t^*_m with respect to t^*_{m-1}.

Encoding the deviations

For the segment r between P_{i_r} and P_{j_r} in the dissection, let the number of points whose deviations needs to be transmitted be $n_r = j_r - i_r$. Each of these n_r points has three spatial deviations. Let δ^r_1, δ^r_2, and δ^r_3 correspond to the set of the first, second, and third spatial deviations respectively, that is, $\delta^r_p = \{\delta^r_{(i+1,p)}, \delta^r_{(i+2,p)}, \ldots, \delta^r_{(n_r-1,p)}\}$ for $p = \{1, 2, 3\}$. These sets of deviations are explained using a probability distribution. The communication of these sets of deviations over this distribution will require the transmission of the parameters of the distribution, followed by the encoding of the observed deviations using that distribution.

In the current context, each set of deviations is encoded using a univariate Gaussian distribution. The parameters of the distribution which result in the minimum message length to encode the data are inferred using the Wallace and Freeman (1987) method and the resultant expression for message length to encode any set of deviations (δ^r_p) is given by the following formula (corresponding to Equation 2.9):

$$I_{\delta^r_p} = 1 + \log \kappa_2 + \log(\mathrm{R}_\mu \mathrm{R}_\sigma) + \frac{1}{2}\log(2n_r^2) + \frac{n_r}{2}\log\left(\frac{2\pi}{\epsilon^2}\right) + \frac{n_r - 1}{2}\log\left(\frac{\sum_{i=1}^{n_r}(\delta^r_{(i,p)} - \hat{\mu}_p)^2}{n_r - 1}\right) + \frac{n_r - 1}{2}$$

where $\hat{\mu}_p = \dfrac{1}{n_r}\displaystyle\sum_{i=0}^{n_r} \delta_{(i,p)}$ is the MML estimate of the mean of the distribution, and $\kappa_2, \mathrm{R}_\mu\mathrm{R}_\sigma, \epsilon$ are hyperparameters used in MML inference[2] (see Section 2.4.3). Hence, the statement cost to encode the coordinates of any segment r between points P_{i_r} and P_{j_r} is given by $I_{\delta^r} = \log^*(n_r) + \displaystyle\sum_{p=1}^{3} I_{\delta^r_p}$.

8.2.3 Total cost of communicating the coordinates using Bézier curves

Given the dissection $\mathcal{Q}$ (model) of the coordinates $\mathcal{P}$ (data), let the total message length required to explain the data be denoted as $I(\mathcal{Q}\,\&\,\mathcal{P})$. Combining the code lengths to state the two part message described in Sections 8.2.1 and 8.2.2, the total message length expression is given as

$$I(\mathcal{Q}\,\&\,\mathcal{P}) = \underbrace{\log^*(k) + k\log V + \sum_{r=1}^{k}(\log^*\theta_r + (\theta_r - 1)\log V)}_{\text{first part: } I(\mathcal{Q})} + \underbrace{\sum_{r=1}^{k} I_{\delta^r}}_{\text{second part: } I(\mathcal{P}|\mathcal{Q})}$$

This equation serves as the objective function that must be minimized to obtain the optimal representation. Noting that $\log^*(k)$ is very small and is significantly less than $k \log V$, this term is ignored to construct a slightly modified objective. The advantage of this modified objective is that it is strictly

[1] The details of this computation are described in Appendix F.1.

[2] Recall that κ_2 is the lattice quantization constant, R_μ, R_σ appear in the prior density of the Gaussian parameters, and ϵ is the accuracy of measurement (for protein coordinate data, $\epsilon = 0.001$ Å).

additive in terms of the code lengths corresponding to each individual segment in any dissection of the proteins into Bézier curves. More specifically,

$$I(\mathcal{Q}\,\&\,\mathcal{P}) \approx \tilde{I}(\mathcal{Q}\,\&\,\mathcal{P}) = \sum_{r=1}^{k} \ell_{ij}^{\theta_r}$$
$$\text{where } \ell_{ij}^{\theta_r} = \theta_r \log V + \log^*(\theta_r) + I_{\delta^r} \tag{8.2}$$

and $\ell_{ij}^{\theta_r}$ denotes the component code length required to explain the coordinates in the region between P_i^r and P_j^r in the dissection, using a Bézier curve of degree θ_r.

8.3 Optimal Bézier segmentation using dynamic programming

The total message length expression formulated in Section 8.2.3 sets up the search problem of finding the dissection $\mathcal{Q}^*$ of a set of coordinates of protein $\mathcal{P}$ such that the encoding length of $\mathcal{P}$ using $\mathcal{Q}^*$, that is, $I(\mathcal{Q}^*\,\&\,\mathcal{P})$ is the minimum over the entire space of possible dissections. This section will describe the procedure to compute the optimal dissection $\mathcal{Q}^*$ and its associated Bézier curve assignment for the given protein coordinate data $\mathcal{P} = \{P_1, \ldots, P_N\}$.

Potentially, every pair of points P_i and P_j, $1 \leq i < j \leq N$ defines a candidate region that the optimal dissection $\mathcal{Q}^*$ could include. Also, associated with each candidate region is the assignment of a specific Bézier curve of arbitrary degree θ.

In order to search for the best dissection, an ensemble of code length matrices $\ell^\theta = (\ell_{ij}^\theta)\ \forall\ 1 \leq i < j \leq N$, one for each degree $\theta = \{1, 2, 3, \cdots\}$ of the family of possible Bézier curves, is constructed.[3] Each ℓ_{ij}^θ contains the code length to encode the coordinates of the region $P_i \cdots P_j$, using a Bézier curve of degree θ.

The ensemble of code length matrices ℓ^θ is then used to search for the best dissection $\mathcal{Q}^*$ and its corresponding Bézier curve assignment. As described in Section 8.2, the goal is to find the dissection that yields a *globally* minimum message length, $I(\mathcal{Q}^*\,\&\,\mathcal{P})$, to encode the coordinate data. Given the property that the code lengths to encode individual segments of any dissection are strictly additive (see Equation 8.2), the best dissection can, therefore, be derived using a one-dimensional dynamic program (Bellman, 1957) with the following recurrence relationship (starting from the boundary value $C_1 = 0$)

$$C_j = \min_{i=1}^{j-1} \begin{cases} \min_\theta \ell_{1j}^\theta \\ (C_i + \min_\theta \ell_{ij}^\theta) \end{cases}, \forall\ 1 \leq j \leq N$$

where any C_j gives the optimal dissection from P_1 up to some intermediate point P_j $(1 < j \leq N)$. Note that the above recurrence ensures that the optimal dissection of coordinates P_1 to P_j is built incrementally using the optimal dissection of its sub-problems defined by $P_1, \cdots, P_i, \forall\ 1 \leq i < j < N$, using the ensemble of code length matrices which are precomputed before the search begins. At the end of the recurrence, the value C_N corresponds to the minimum message length $I(\mathcal{Q}^*\,\&\,\mathcal{P})$ corresponding to the optimal dissection of $P_1, \ldots, P_N$. This dissection and the corresponding Bézier curve assignment (which are *memoized* in order to speed up the process when computing each C_j) can be computed by *backtracking* (Bellman, 1957) along the array of dynamic programming history values given by C.

[3] In practice, only three matrices are considered, by restricting the degree of Bézier curves to at most *three*. Note that the ensemble of linear, quadratic and cubic Bézier curves are sufficiently powerful to explain the observed plasticity in protein structures.

8.4 Experimental analyses

8.4.1 Case study: dissection of regions of Ubiquitin-like domain of human homologue A of Rad23 protein

The approach described above to obtain the dissections of a given protein structure is illustrated using the (randomly chosen) Ubiquitin-like domain of Human homologue A of Rad23 protein (Chen et al., 2011) with wwPDB code 2WYQ. The generated representations derived from the above approach are compared and contrasted with the traditional representations at the level of secondary structural elements.

Figure 8.3(a) shows the structure of 2WYQ with standard secondary structures – helices and strands of sheet, assigned using DSSP (Kabsch and Sander, 1983). In the traditional approach to abstracting protein folding patterns, the secondary structural elements are replaced by line segments (Shi et al., 2009; Konagurthu et al., 2008). This is shown in Figure 8.3(b). Notice that in excess of 50% of the structure is omitted as they do not partake in forming local secondary structural features. As a result, the representation remains lossy.

Figure 8.3(c) shows three dissections of 2WYQ produced by our approach described above, by varying the maximum degree of Bézier curves used to dissect the protein. In the left frame of the figure, we constrain the approach to utilize only the linear order (degree=1) Bézier curves. This results in a piecewise linear abstraction of the protein folding pattern. This representation is equivalent to those produced by STICKS (Taylor, 2001) and PMML (Konagurthu et al., 2011). However the strength of the methodology described in this paper is that our method can be generalized to higher order curves, with the piecewise linear approximation being the limiting case. The middle frame of Figure 8.3(c) shows the dissection when the available models include both linear (degree 1) and quadratic (degree 2) Bézier curves. The right frame of the figure further extends this model set to include cubic Bézier curves (degree 3).

Discussion on the generated Bézier abstractions

To understand the quality of the dissections produced under varying constraints on the degree of Bézier curves, the dissections corresponding to maximum degree 1 and maximum degree 3 (that is, each dissected region in the protein can be of either degree 1, 2 or 3) reported in Figure 8.3(c) are compared with each other. These two dissections will be henceforth referred to as *linear* and *non-linear* Bézier abstractions respectively.

The linear Bézier abstraction yields 8 segments while the non-linear Bézier abstraction results in 7 segments. These segments are individually shown in Figures 8.4 and 8.5 respectively. (There are two more segments but as they are only two residues long; these will be ignored in the discussion below). Noteworthy aspects of the two abstractions are

- Segment 1 corresponds to a strand of a four-stranded anti-parallel β-sheet in 2WYQ. This strand shows a mild curvature (see Figure 8.4a). The non-linear abstraction suitably captures this curvature using a quadratic Bézier curve (Figure 8.5a).

- Segment 2 corresponds to the anti-parallel strand with respect to the first, and shows a moderate curvature. The linear Bézier abstraction introduces a poorly fitting line segment (see Figure 8.4b), while the non-linear Bézier abstraction chooses a well fitting quadratic Bézier curve to explain that region (see Figure 8.5b).

- Segment 3 comprises mainly of a helical region whose terminal regions are flanked by loops. Demonstrating the economy of abstraction, the non-linear Bézier abstraction models this region using a cubic curve (see Figure 8.5c). In stark contrast, the linear Bézier abstraction models the region using a poorly fitting line segment (see Figure 8.4c), thereby losing the information of the curvature in that region.

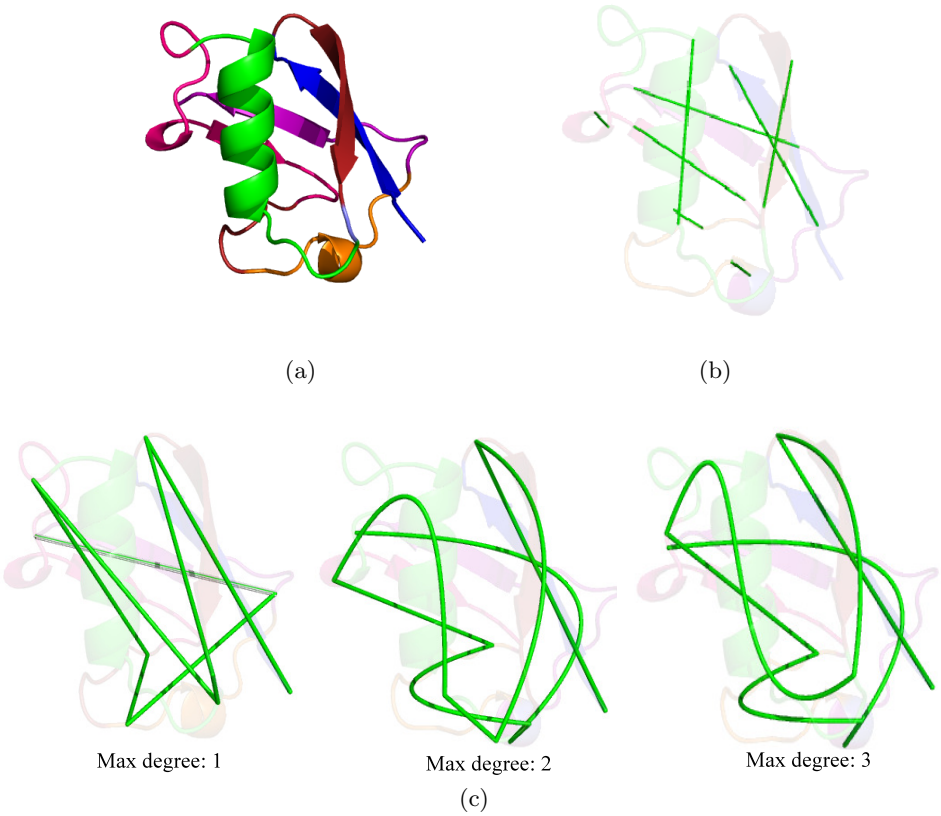


Figure 8.3: (a) Structure of Ubiquitin-like domain of human homologue A of RAD23 (wwPDB code 2WYQ) shown with standard secondary structures – helices and strands of sheet assigned using the DSSP software tool (Kabsch and Sander, 1983). (b) Traditional representation of the folding pattern which replaces each secondary structural element with line segments. (c) The representations produced by the MML approach by constraining the Bézier curves to a maximum degree of 1 – Linear (left frame), maximum degree of 2 – Linear and Quadratic (middle frame), and maximum degree of 3 – Linear through to Cubic (right frame).

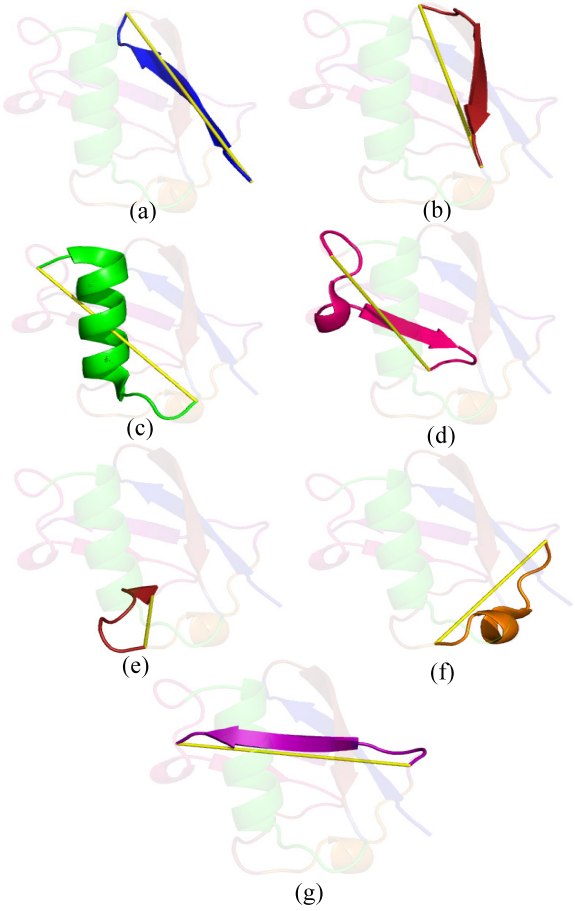


Figure 8.4: Linear Bézier curve abstractions of 2WYQ.

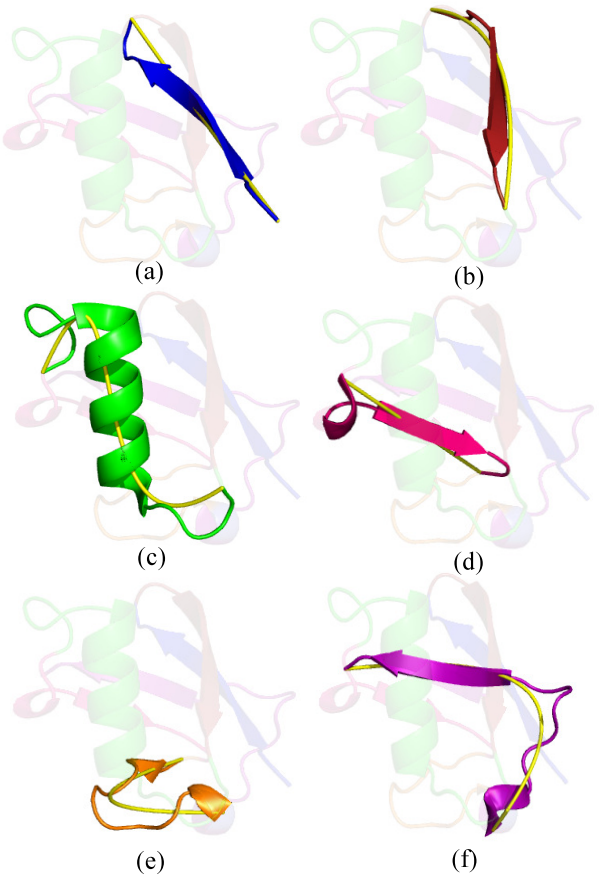


Figure 8.5: Non-linear Bézier curve abstractions of 2WYQ.

- Segment 4 has a linear trend and, hence, both dissections model this region using a line segment (see Figures 8.4d and 8.5d).

- Segment 5 is a coil and this is approximated using a quadratic Bézier curve in the non-linear abstraction (Figure 8.5e). The linear abstraction, however, models this region partially using two line segments (see Figure 8.4).

- Segment 6 in the non-linear abstraction (Figure 8.5f) comprises of a coil and a β strand. This combination is described in the non-linear abstraction using a single quadratic Bézier curve. The linear abstraction for this segment results in two segments – a part of the coil belongs to segments 6 and the beta sheet is represented as a line, as shown in Figures 8.4(f) and (g).

Comparison of the Bézier abstractions with secondary structural segmentations

The non-linear Bézier abstraction is now compared with the representation of folding patterns using secondary structures. Several comparative studies have highlighted the poor consensus among popular secondary structural assignment programs (Colloc'h et al., 1993). We consider the two radically different programs to assign secondary structures to protein coordinate data: DSSP (Kabsch and Sander, 1983) and SST (Konagurthu et al., 2012). The coordinates of 2WYQ were passed through DSSP and SST and the resulting segmentations are used to represent the backbone of this structure as a set of vectors (by replacing the secondary structural elements with line segments; see Figures 8.6 and 8.7). Using both these methods a major portion of the protein folding pattern goes unrepresented.

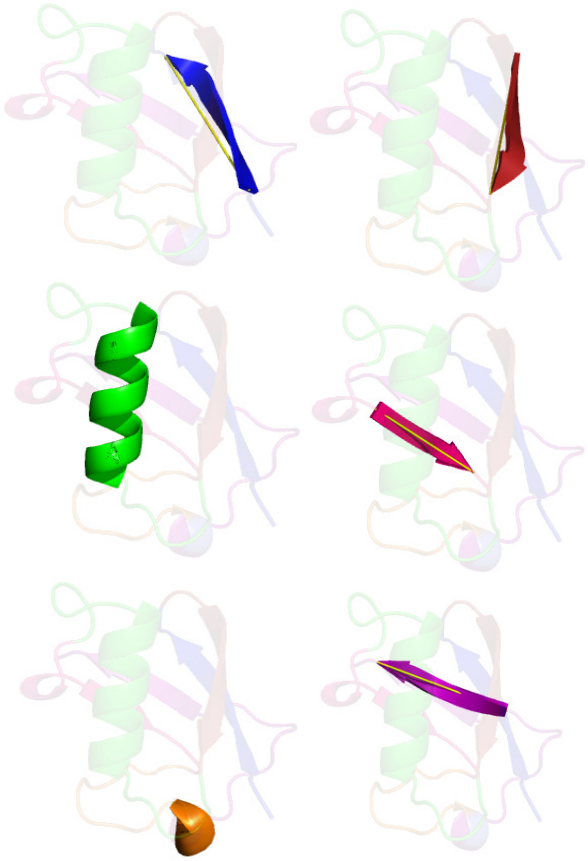


Figure 8.6: DSSP segmentation of 2WYQ.

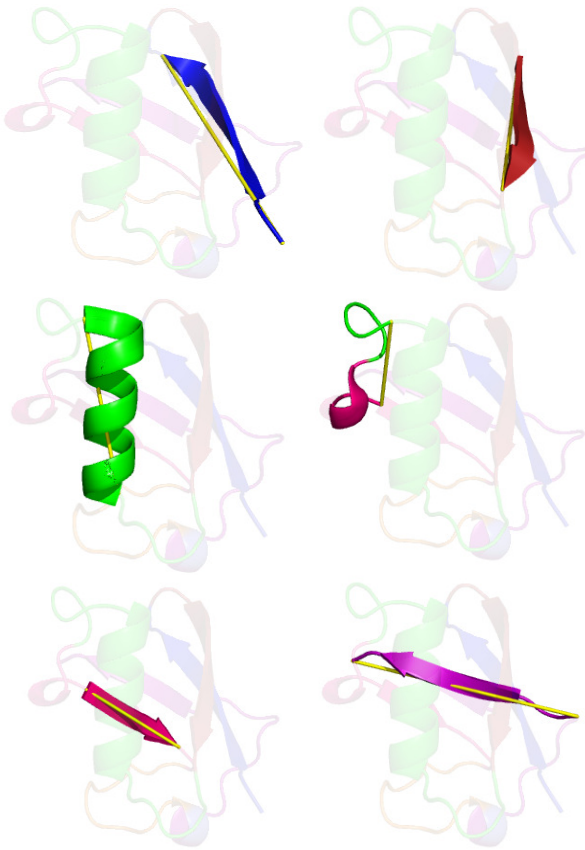


Figure 8.7: SST segmentation of 2WYQ.

Of the 77 residues, the segmentations generated using DSSP and SST result in representing 37 and 44 residues respectively, which is only $\sim 48\%$ and $\sim 57\%$ of the whole region.

Visual inspection of the Figure 8.3 reveals that the secondary structural elements in 2WYQ depart considerably from their ideal geometries and hence the line segments approximating these regions, for both the approaches, show significant errors. In contrast, our non-linear abstraction results in a well defined abstraction that closely approximates the folding pattern of 2WYQ. This case study and many others that were carefully examined manually (two of which are presented in Appendix F.2) validates the effectiveness of the non-linear Bézier abstractions produced by the proposed approach. It uses an elegant and mathematically rigorous approach to achieve the objective of maximizing the economy of description, while at the same time minimizing the loss of structural information.

8.4.2 Fold identification using Bézier abstractions

As a proof of concept, a simple-minded strategy is designed here to compare and score the similarity between any two abstractions generated by the proposed approach.

Given any dissection of a protein structure, a *geometric profile* is created, and is used in comparison. The basic idea here is that the sequences of geometric profiles of two proteins sharing similarity in their folding patterns would also be similar. This allows the use of the rich repertoire of sequence comparison methods to efficiently undertake large scale comparisons of protein folding patterns.

For a given Bézier curve dissection, the geometric profile is generated as follows: for every region in the dissection, each intermediate control point of the Bézier curve is projected onto the curve. The projection is the point on the curve that is closest to the control point. These projections are connected to the end points, thus, forming a series of line segments. Since any two skew lines have a mutually perpendicular vector, the angle required to rotate one line about this perpendicular axis so

that it eclipses the other line gives an orientation angle in the range $[0°, 360°]$. A table of orientation angles are recorded for each pair of line segments. In addition to the orientation angles, the distances between the mid-points of each pair of lines are also recorded. The row-wise concatenation, of all possible orientation angles and the corresponding distances between lines results in a sequence of 2-tuples representing a simple geometric sequence profile of the protein.

For comparing two dissections, their representative geometric profiles are compared by their alignment. Any two geometric profiles are aligned by implementing the standard affine-gap version of dynamic programming algorithm to compare pairs of sequences. An *ad hoc* scoring function is used to score matches in the alignment. Specifically, the score of aligning any two angles w_i and w_j and the corresponding distances r_1 and r_2 between the midpoints of the interacting line segments is given by

$$\text{score}(w_i, w_j) = (45° - \Delta w) \exp\left(-\frac{\Delta r^2}{c^2}\right)$$

where $\Delta w = \min\{|w_i - w_j|, 360° - |w_i - w_j|\}$, $\Delta r = |r_1 - r_2|$ is difference of the two distances, and c is a constant which is set to 20 Å. With this new scoring function, the alignments are recomputed. The resultant alignment score derived by aligning any two geometric profiles is normalized as

$$\hat{S}(A, B) = \frac{S(A, B)}{0.5 \times (S(A, A) + S(B, B))} \tag{8.3}$$

where $S(A, B)$ is the optimal score; $S(A, A)$ and $S(B, B)$ are the respective self-alignment scores. In summary, in order to compare any two Bézier segmentations, their geometric profiles are generated and their sequence alignment is carried out.

Using the Bézier segmentation profiles for fast database search

The applicability of Bézier segmentations is demonstrated by using them to achieve efficient and accurate database search and retrieval. To do so, Bézier segmentations are used initially as a screening filter to identify a small candidate set of proteins from the database, that are most likely to be structurally similar to the query. We then run sophisticated (and time consuming) structure alignment programs to accurately match the query with each of the candidate structures in this small set. Clearly, such an approach can save significant amounts of computation time without losing accuracy, *if* the filtering method is effective. The following experiments assess the effectiveness of a filtering method based on Bézier segmentations.

The dataset considered is the entire ASTRAL-SCOP 40 (version 1.75) database containing 11,146 domain structures (Murzin et al., 1995). The structures are categorized into 1,188 distinct folds and 7 distinct classes. It is to be noted that that no two domains in this dataset share more than 40% amino acid sequence identity. SCOP is a classification hierarchy of the protein domains. The hierarchy consists categories of protein domains that are categorized at the family, superfamily, fold, and the class levels. The domains belonging to the same family are closely related protein structures. Given a query domain, we wish to identify other domains with the same fold type as that of the query. This refers to searching for strutures with similar folding patterns. If we perform the search at a class level, we are searching for structures that belong to the same class.

The experiment proceeded as follows: five structures *(queries)* were selected initially by identifying the most common five folds types in the database (that is, those present in more structures) and then randomly selecting a structure for each fold type. The queries used in this experiment are listed in Table 8.1. Once the five queries were selected, the geometric profile of each query was aligned with the geometric profile of each of the 11,146 structures in the database, and the resulting alignment scores were used to rank the structures.

The structures at the top of the ranked list are expected to be the most similar to the query structure. The top k % (cutoff point) structures are filtered for varying values of k (0, 5, 10, ..., 100) and are considered to have a fold identical to that of the query. The *false positive rate (FPR)* and

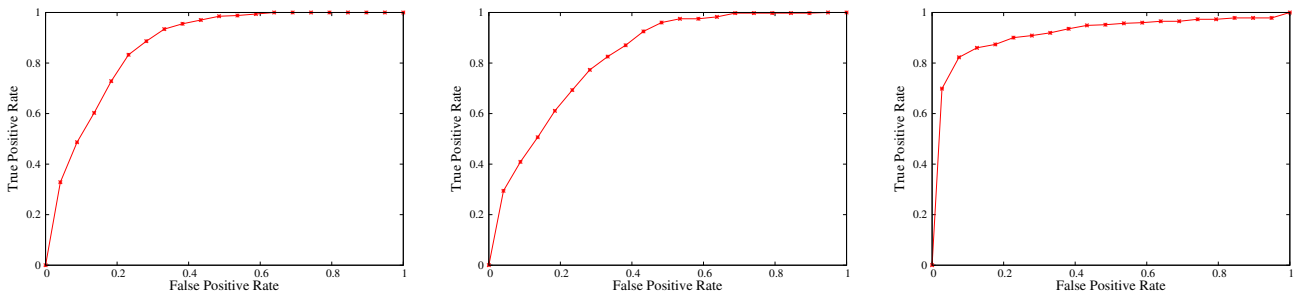
true positive rate (TPR) at each of these cutoff points are computed. These values are plotted and a Receiver Operating Characteristic (ROC) curve is generated which is used to establish the accuracy of the filtering method. The closer the curve is to the diagonal, the less accurate is the test. The area under the ROC curve (AUC) is, therefore, a direct measure of this accuracy.

Table 8.1: Description of the five queries selected.

Fold type	Fold description	Query	Query domain
a.4	DNA/RNA binding-3-helical bundle	d2hosa_	a.4.1.1
b.1	Immunoglobulin-like β-sandwich	d1l6za1	b.1.1.1
c.1	TIM β/α-barrel	d1w0ma_	c.1.1.1
c.2	NAD(P)-binding Rossman-fold	d1gu7a2	c.2.1.1
d.58	Ferredoxin-like	d2fdna_	d.58.1.1

Table 8.2: Area under the curve (AUC) values when evaluated at the *fold* and *class* levels.

Query	AUC (fold level)	AUC (class level)
d2hosa_	0.87	0.74
d1l6za1	0.83	0.67
d1w0ma_	0.92	0.80
d1gu7a2	0.81	0.78
d2fdna_	0.82	0.58



(a) d2hosa_ (b) d1l6za1 (c) d1w0ma_

Figure 8.8: ROC curves for the fold level evaluation. The corresponding AUC values are tabulated in Table 8.2. The red dots in each ROC curve denotes the (TPR,FPR) value at a cutoff points (k=0,5,10,...,100).

The TPR and FPR metrics are computed by evaluating the ranked list on the basis of fold similarity or class similarity as described previously. The folds are considered to be similar when the fold type of the query matches those in the ranked list. The class level similarity is when both the query and each of the domains in the ranked list belong to the same class. The values of TPR and FPR are calculated for both the cases and the corresponding results are shown in Table 8.2. The ROC curves for the query domains are shown in Figure 8.8.

It is observed that the accuracy of the test is greater when the discrimination between structures is at the fold level. The fold level discriminative ability provided by the Bézier segmentation profiles suggest that this method can be used to identify similar folds and hence, can be used in the initial screening of a database of structures. Once the candidate structures are filtered out, these can be scrutinized more thoroughly to find the structure with identical folding pattern as the query.

8.4.3 Comparison of our fold-identification method with others

Comparison with protein alignment methods

As discussed in Section 8.4.2, the geometric profiles constructed using the Bézier dissections can be used to compare the protein structures. It is expected that proteins that are structurally similar would have similar Bézier dissections and consequently similar geometric profiles. As a result, when their geometric profiles are aligned, they would have a relatively higher alignment score (Equation 8.3). This can be validated by using the existing protein alignment methods. The resulting alignments of protein structures using the existing methods are also expected to have a higher alignment score. The similar pattern is observed when structurally dissimilar proteins are considered, in which case, the alignment scores would be relatively lower. We demonstrate this behaviour here.

A quantitative assessment of the Bézier abstractions generated by our approach is demonstrated by undertaking a large scale comparison between protein structures abstracted using our proposed approach. These structures are chosen such that they vary across the entire spectrum of structural relationships. Specifically, we randomly choose a data set containing 500 domains from the ASTRAL SCOP 40 (version 1.75) database (Murzin et al., 1995). Let these selected domains be referred to as *pivots*. Corresponding to each pivot domain, five distinct domains (differing in length by no more than 50 residues with respect to the pivot) are randomly chosen, such that the first belongs to the same SCOP family as the pivot, the second belongs to the same SCOP superfamily (but not family), the third belongs to the same SCOP fold (but not family or superfamily), the fourth belongs the same SCOP class (but not any better) and the fifth belongs to an entirely different SCOP class (the *decoy*).

Geometric profiles of the non-linear Bézier curve abstractions (see Section 8.4.2), are generated for each of the 2500 ($= 5 \times 500$) SCOP domains in this data set. The geometric profiles of the 500 pivot domains are aligned separately with each of its 5 associated domains. A box-whisker plot is constructed to show the variance in the normalized alignment scores at each level in the SCOP hierarchy.

To serve as a gold standard, the same data set is aligned using the commonly used protein alignment methods, such as DALI (Holm and Sander, 1993), Matt (Menke et al., 2008), and TM-align (Zhang and Skolnick, 2005). The value of the scoring function for each method is used to generate a Box-Whisker plot showing the variability under this scoring function. The plots are shown in Figure 8.9.

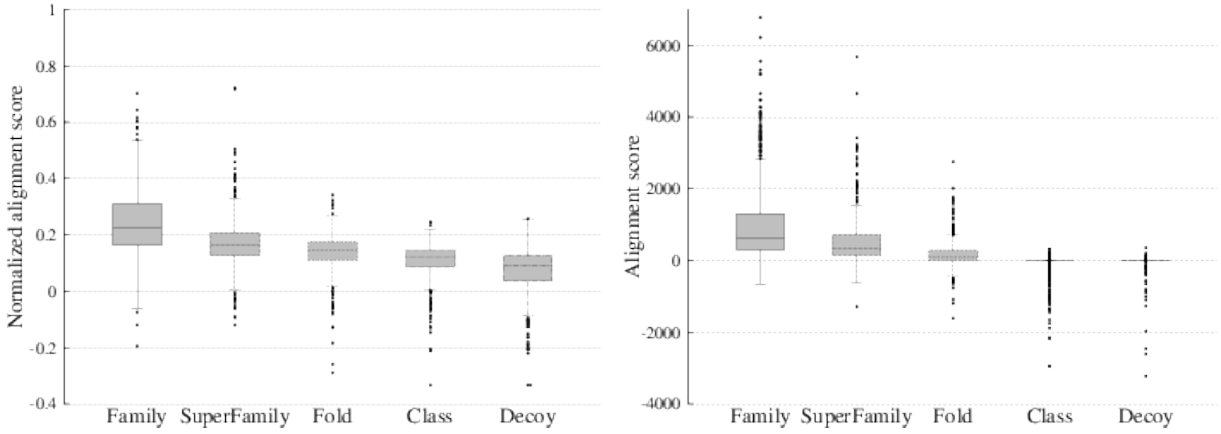
A good discriminative scoring function should result in a box-whisker plot where the median values of alignment scores decreases monotonically as the structures being compared diverge in their evolutionary (structural) distance along the SCOP hierarchy: family, superfamily, fold, class, and decoy. Examining these plots, all the alignment methods show this behaviour. At the levels of SCOP family, superfamily and fold, the discrimination achieved using DALI and our proposed approach remains comparable. However, as expected of residue-residue comparisons, DALI is significantly better at the levels of Class and Decoy. This accuracy of DALI comes at a heavy computational cost.

DALI takes about ~ 12 hours to compute all the 2500 alignments in the data set we considered. In contrast, it only takes us ~ 15 minutes to compare the geometric profiles. This excludes the one-time preprocessing cost of computing the geometric profiles of the source structures in the collection.

Furthermore, the standard scores of the normal distribution (z-scores) for the DALI alignments are compared with the z-scores for the raw alignment scores produced by the proposed method. To compute the z-scores corresponding to our alignment scores, the pairwise alignment scores at the class level are considered to be the population values. The mean and standard deviation of these values are then calculated. These population parameters are then used in the computation of the z-score corresponding to a given alignment score. The correlation coefficient of the z-scores at different levels of the SCOP hierarchy is then computed. The results are tabulated in Table 8.3.

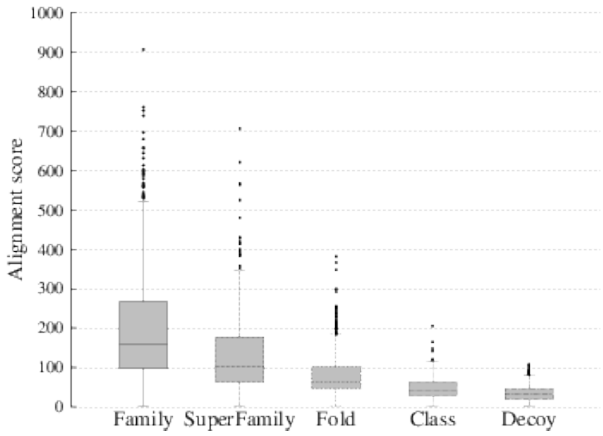
Table 8.3: Correlation of z-scores of alignments obtained using our method and DALI.

Family	Superfamily	Fold	Class	Decoy
0.82	0.75	0.79	0.18	0.05

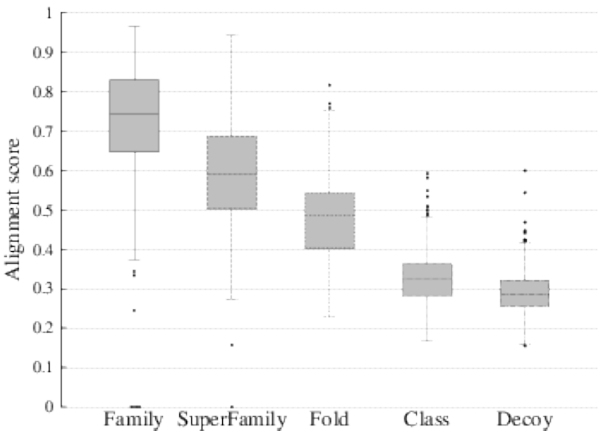


(a) Non-linear Bézier curve abstraction

(b) DALI



(c) Matt



(d) TM-align

Figure 8.9: Box-whisker plots of comparisons on a structural set derived from SCOP using the geometric profiles of our non-linear abstraction and other popular alignment methods *viz.* DALI, Matt, TM-align. (Note, the scale on the Y-axis differ between plots. A comparison of the widths of the boxes across two boxplots has no meaning. Each boxplot is used to assess the discriminative power across the SCOP hierarchy using an alignment method.)

From the correlation coefficients, it can be deduced that there is a significant linear dependence between the alignment z-scores produced by the two methods. This confirms the effectiveness of the alignments generated by our method at the family, superfamily, and fold level. However, at the class and decoy level, the degree of correlation is not substantial. The alignments are also compared against the ones generated using TM-align and Matt. TM-align also provides a better discrimination at the Class and Decoy level. However, the variance of the alignment scores at each level of the SCOP hierarchy is greater compared to all other alignment methods. The alignment results of Matt and ours are also comparable as can be seen from the boxplots.

The results suggest that the proposed abstraction mechanism allows for accurate fold-level discrimination using the simplest of the search strategies. The method of comparison is however used here only as a proof of concept. The proposed non-linear abstraction can be used as the basis of more rigorous methods for efficient large scale structure comparison. Towards this goal, the use of Knot invariants from algebraic topology is explored.

Knot Invariants based comparison

Røgen and Fain (2003) introduced an approach to compare protein structures by representing each structure as a 30-dimensional vector of topological *knot invariants*. The similarity between any two structures is then given by the Euclidean distance between the two vectors. This forms an alignment-free methodology for structural comparison. The components of the vector correspond to the knot invariants computed by approximating the protein backbone as a polygon. Knot invariants are generic and provide a framework to compare non-linear curves. But because of the lack of closed form solutions in computing the individual invariants, the non-linear curves are approximated as polygons. There exists an analytical way to compute the invariants when the curve is comprised of line segments. The details are outlined in Røgen and Fain (2003).

We adapt Røgen and Fain (2003)'s idea to our approach. Each Bézier abstraction is approximated as a polygon by the method described in Section 8.4.2 (that constructs geometric profiles for comparison). Knot invariants are then computed using this polygon. The discriminative ability of the knot inavriants based methodology is tested on our data set selected from SCOP. The results are shown in Figure 8.10. We notice that the discriminative ability using the knot invariant alignment-free approach is significantly poorer than the one achieved using simple geometric profiles (see Figure 8.9a). This might be because in order to achieve a better discrimination using knot invariants, the Bézier

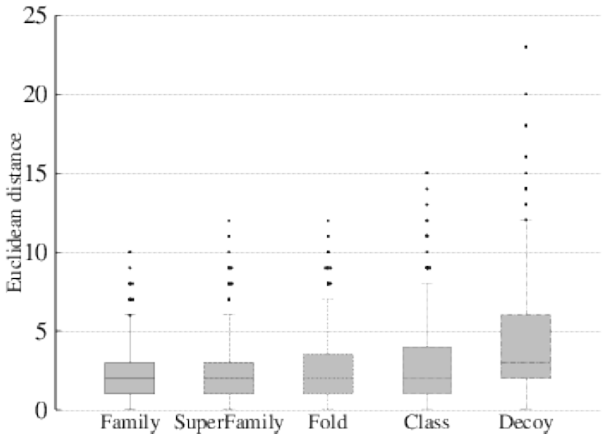


Figure 8.10: Box-whisker plots for the non-linear Bézier curve abstractions compared using knot invariants (smaller value on the ordinate axis implies better structural relationships).

curves should be considered raw (without abstracting them as polygons). This involves solving the Gaussian integrals which do not have closed form solutions (unlike the polygonal case considered in this exercise).

8.5 Summary

This chapter describes the MML inference of a set of piecewise non-linear curves that optimally model 3D data with inherent curvilinear nature. As a demonstration of the inference process, protein coordinate data is discussed as a case in point. The data is modelled using segments of non-linear Bézier curves. The ability of the MML criterion to objectively evaluate competing segmentations and determine the optimal combination of curves is practically demonstrated through relevant case studies.

The proposed Bézier abstraction mechanism does not rely on secondary structure assignment of proteins to generate concise representations. This preserves the underlying geometry which would otherwise be partially lost due to secondary structure assignment. Only about half of the protein is made of standard secondary structural elements. Hence, abstract representations requiring an initial assignment do not span the entire protein region. This drawback has been highlighted when studying

an example in Section 8.4.1. In contrast, the proposed mechanism represents the whole protein region and is able to model both the regular (helices and strands) and irregular portions (coils) of proteins using Bézier curves. This allows complete coverage without compromising the features essential in the modelling of proteins.

Furthermore, in the context of protein modelling, the practical utility of the method is carefully analyzed. The Bézier segmentation of proteins can be used as a starting point for investigations into other challenging computational problems in structural biology. As an example application, comparisons of protein structures is described. The generated Bézier segmentations are used to construct geometric profiles of the proteins. A methodology to compare any two protein structures by sequence alignment of their geometric profiles is described. The designed methodology is employed in the important task of identifying proteins with similar folding patterns. While searching a huge database of proteins, the comparison method can be employed to quickly filter out candidate proteins that are similar to the protein under consideration. The filtered candidates can then be carefully scrutinized to determine their similarity. This work has since been published in Kasarapu et al. (2014).

Chapter 9

Conclusion

9.1 Summary

This thesis addresses some of the inference problems that are typically encountered when modelling data distributed in the Euclidean space and data that is directional in nature. In doing this, it highlights the importance of correctly accounting for the model's complexity when selecting an optimal model. To achieve this, it uses an inference framework based on the MML principle which has been shown to be able to appropriately resolve the complexity-fit trade-off of using a certain model. In particular, it uses the Wallace and Freeman (1987) approximation of the strict MML principle to infer the parameters of the probability distributions.

The core of this thesis focused on deriving the MML estimators of some commonly used probability distributions whose parameters have not been previously characterized. These include the Laplace and multivariate Gaussian, which allow for modelling of data in the Euclidean space, and also the MML estimators of directional probability distributions such as the multivariate vMF to model data on the unit hypersphere, the Kent distribution to model data on the surface of a 3D sphere, and the BVM distribution to model data distributed on the surface of a 3D torus. We demonstrated that the derived estimators fare better in comparison to the traditionally used ML and MAP estimators, in terms of bias, MSE, and KL distance. The statistical consistency of the estimators has been established using the likelihood ratio based hypothesis testing.

Furthermore, for modelling real-world empirical data with multiple modes, we described a mixture modelling method that simultaneously infers an optimal number of mixture components along with the mixture parameters. We developed the search and inference modelling apparatus for the case of multivariate Gaussian mixtures and demonstrated that it is superior to the state-of-the-art. The generalizability of our proposed search method was demonstrated by adapting it to mixtures of directional probability distributions.

We then demonstrated the applicability of our mixture modelling method in describing real-world directional data that is of practical significance. To achieve this, we employed high-dimensional vMF mixtures in text clustering, where the normalized representations of the text documents correspond to data on the unit hypersphere. We also employed 3D vMF and Kent distributions in modelling 3D protein directional data generated from the spatial orientations of the central carbon atoms along the protein main chain. We observed that the Kent mixtures have better explanatory power compared to the vMF mixtures. These improved mixture models can be used as fundamental tools in protein modelling tasks such as generating random protein conformations and 3D protein structure alignments. We also considered mixtures of BVM distributions to model the protein main chain dihedral angle pairs. These BVM mixture models can facilitate the understanding of the important structural properties of the proteins such as their folds. The resulting BVM mixtures are shown to supersede the models that assume no correlation between the pairs of dihedral angles.

The inferred mixtures of the directional probability distributions considered in this thesis are shown to closely emulate the empirical distribution of the data. Our search method which was used to infer

these mixtures is able to identify regions of importance, namely the α-helices and β-strands, in the protein conformational space. These regions are important because they correspond to the secondary structure of proteins. Importantly, our MML-based mixture modelling method is carried out in a completely unsupervised setting without requiring any knowledge about the underlying distribution of the data.

This thesis also explored the problem of function approximation. Specifically, we showed the applicability of the MML framework in approximating the infinite series expansion of a periodic function using a finite number of orthogonal basis functions. Such an approach is useful in the context of protein structure determination, where one is required to approximate the infinite Fourier expansion correponding to the diffraction pattern of the 3D structures of crystallized protein molecules.

Finally, we also investigated the modelling of data that is not described using probability distributions. The Wallace and Freeman (1987) method is computationally tractable only for models of probability distributions. Hence, when there is no probabilistic interpretation of the models of the data, one has to develop the schemes to explicitly encode the model parameters and the data using those parameters in accordance with the general MML framework. We specifically explored the problem of representing a set of 3D points by using a piecewise combination of Bézier curves of varying degrees, rather than by probability distributions. We have demonstrated that such representations of protein structural data are useful in modelling their folding patterns, and serve as unique signatures that facilitate rapid searches for similar structures in large protein databases.

9.2 Further work

A clear direction of further work is to develop alternative strategies of modelling Euclidean and directional data. This may be facilitated by modifying the mixture modelling apparatus discussed in Chapter 5, where we described a generalized framework of mixture modelling using split, delete, and merge operations. It may be useful to define alternative versions of these operations. For example, in the split operation, we initialized the means of the child components to be at a distance equal to the maximum eigenvalue on either side of the parent mean along the direction of maximum variance. Such an approach is deterministic in nature and it is easy to develop specific examples where such an approach might not be useful. To make the split operation non-deterministic, a strategy, such as using random initializations along the maximum variance direction within some reasonable distance from the parent mean, may be used. Similarly, the deletion step can be modified by incorporating the effective number of data points within each components and selectively eliminating those with fewer data memberships in some reliable way. Further, while merging a pair of components, in addition to using the KL distance, we could take into account the empirical distribution of the data within those components. There are conceivably several variations of these operations that have to be carefully studied.

In addition to updating the individual operations employed in the search process, the current implementation of the search method can itself be modified. Our proposed search method perturbs an intermediate mixture exhaustively. An interesting modification is to include only some of the perturbations for selected components. This could be done by determining the propensity of the mixture components to improve the total message length by considering their previous history (Wallace and Dowe, 1994a). The different perturbations can also be carried out probabilistically, following a similar approach to simulated annealing. However, such an approach should be robust in terms of recovering components if they are deleted by chance.

Further interesting extensions of the mixture modelling method arise in the context of partitioning the observed data into groups using the widely used K-means clustering algorithm. In the traditional K-means method, the number of clusters, denoted by K, has to be specified. Alternatively, the best K is determined by trying various values of K and selecting the one that minimizes a given objective function. However, as pointed out in Section 5.3, such an approach only leads to a sub-optimal solution and is not elegant. A search method similar to the one proposed for mixture modelling in Section 5.4

can be designed in this regard. Also, the cluster memberships are computed using an Euclidean metric as the distribution of data within the clusters is not modelled by probability distributions. This poses another challenge in terms of using the MML framework to do mixture modelling when the component models are not probability distributions.

The MML-based search and inference method can further be developed for hierarchical clustering and subsequent selection of the optimal number of clusters. Hierarchical clustering methods do not yield distinct partitions of data and so, the data may belong to several clusters. Hierarchical clustering is of practical signifance as it has the potential to result in meaningful grouping of data that exhibit a clear taxonomical structure. Boulton and Wallace (1973, 1975) have previously attempted hierarchical clustering using the MML framework. Their MML-based hierarchical clustering was employed in the context of identifying the various vegetation types by Wallace and Dale (2005). There is a scope for generalizing the hierarchical clustering method to apply to wide variety of problems, an example of which is the identification of the evolutionary tree of different plant and animal species. It also has an important biological application in identifying various levels of supersecondary structures. The most common secondary structural elements in protein structures are the α-helices and β-strands. However, it is conceivable that these fundamental secondary structural elements can be grouped in a biologically meaningful way leading to the discovery of frequently occuring patterns in protein structures.

Another interesting extension of this thesis is to investigate the utility of other MML approximations (Lam, 2000). Recall from Chapter 2 that the Wallace and Freeman (1987) approximation is one practical way of inference using the MML framework. Parameter estimation based on this approach involves a quadratic approximation of the Taylor series expansion of the negative log-likelihood function in the derivation of the minimum message length expression (see Section 2.4.2). The Wallace and Freeman (1987) approximation relies on the explicit computation of the determinant of the expected Fisher information matrix. As we have seen in the context of parameter estimation of the probability distributions discussed in this thesis (Chapters 3, 4 and 5), the computation of the expected Fisher requires the complete mathematical forms of the probability distributions including explicit expressions of their normalization constants.

There are several probability distributions that do not have closed form expressions of their normalization constants or for which the normalization constants are not computationally tractable. Some examples include directional probability distributions from the Fisher-Bingham family (Mardia, 1975b). The MMLD approximation (Dowe et al., 1998; Wallace, 2005) is useful in such cases. Compared to the Wallace and Freeman (1987) approximation, the MMLD variant is computationally intensive. However, this approximation has been previously employed in univariate polynomial inference (Fitzgibbon et al., 2002), mixture modelling of multivariate Gaussian distributions (Agusta and Dowe, 2002), and inference of multilayer perceptrons (Makalic and Allison, 2013).

There can be a strong case made for using the MMLD approximation in modelling directional data. Recall that in Chapter 6, we used the vMF and the FB_5 probability distributions to model directional data on the unit sphere. The parameters of the vMF and FB_5 distributions have natural interpretations and are specific cases of the general Fisher-Bingham (FB_8) distribution (Mardia, 1975b; Bingham and Mardia, 1978). Further, these distributions are particularly useful for modelling data that are symmetrically distributed. In order to model data that has no obvious symmetry, the FB_8 distributions offer a better alternative (Wood, 1988). However, inference using the FB_8 distribution poses difficulties because of its complex mathematical form. As a result, the Wallace and Freeman (1987) approximation cannot be readily used. As an alternative, the MMLD approximation can be useful. Once the MML parameter estimators are derived for the FB_8 distribution, it can be used in conjunction with our mixture modelling apparatus. As an example application, mixtures of FB_8 distributions can be used to model protein directional data (as in Section 6.3.3). The models resulting from the use of FB_8 mixtures are expected to be superior null model descriptors of protein directional data. Recall that the FB_5 distribution is a generalization of the vMF distribution and we demonstrated that FB_5 mixtures better describe the protein directional data as compared to the vMF mixtures. In the same vein, because the FB_8 distribution is the most general one, the mixture models

with FB_8 component distributions would lead to additional compression of the protein directional data and, consequently, serve as efficient descriptors.

Finally, another interesting extension of this thesis is in the domain of modelling the dihedral angles along the entire protein backbone, rather than focusing only on the protein main chain. Recall from Section 6.4 that the bivariate von Mises (BVM) mixtures were used in the modelling of protein main chain dihedral angles. The main chain dihedrals form a collection of angular pairs that are represented as points on the surface of a 3D torus. The protein backbone is made up of side chain atoms in addition to the main chain atoms. In Figure 9.1, the main chain dihedral (ϕ, ψ) pairs are shown in blue and the side chain dihedrals (χ_1, χ_2, χ_3) are shown in green. The modelling of protein side chains is particularly important in applications that involve the sampling of protein conformational space to construct reliable models of protein structures (Bower et al., 1997). Finding an initial reliable sample set of conformations is the starting point for *ab initio* structure prediction tasks such as homology modelling, protein design and engineering (Dunbrack and Cohen, 1997).

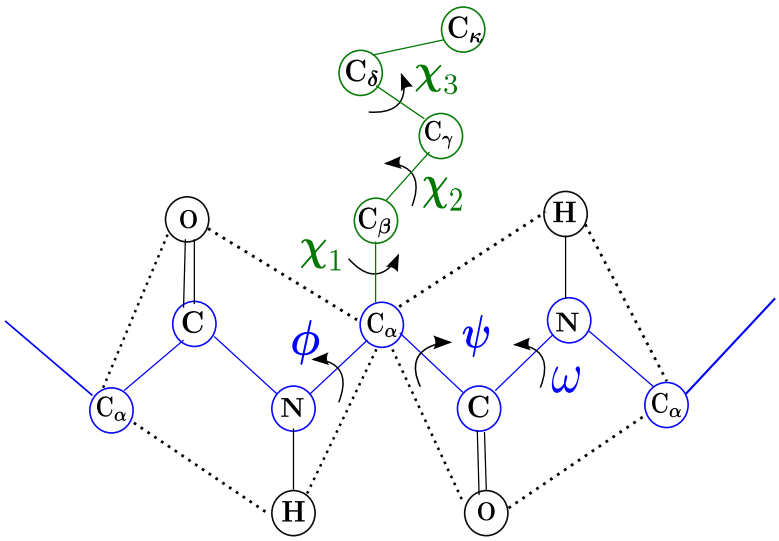


Figure 9.1: Protein side chain dihedral angles denoted by (χ_1, χ_2, χ_3).

The number of dihedral angles that are part of the side chain are variable (can be upto four) and depends on the amino acid type. The side chain dihedrals are represented as points on the surface of a multidimensional torus. A directional probability distribution that models the joint distribution of such data is the multivariate von Mises distribution. The explicit form of the normalization constant in the general multivariate case is not known for this distribution (Mardia et al., 2008). Hence, inference using this probability distribution is a difficult problem and requires the challenging task of estimating its parameters and employing it in the context of mixture modelling. Mardia et al. (2008) derived the parameter estimators using the methods of moments and pseudo-likelihood. The traditional estimation methods can be improved by employing MML-based estimation. The multivariate von Mises is a classic example where the MMLD approximation can be used to estimate the parameters of the distribution. The MML framework can, therefore, be extended to the modelling of the protein side chain and generating models that closely emulate the empirical distribution of the dihedral angles.

Publications

The publications arising from my thesis are

- **P. Kasarapu**, L. Allison, Minimum message length estimation of mixtures of multivariate Gaussian and von Mises–Fisher distributions, *Machine Learning*, Vol. 100, No. 2-3, Pages 333-378. Springer US, 2015.
- **P. Kasarapu**, M. G. de la Banda, A. S. Konagurthu, On representing protein folding patterns using non-linear parametric curves, *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, Vol. 11, No. 6, Pages 1218-1228. IEEE Computer Society, 2014

The manuscripts under communication and preparation include

- **P. Kasarapu**, Modelling of directional data using Kent distributions, preprint at `http://arxiv.org/abs/1506.08105` (*in communication*)
- **P. Kasarapu**, Modelling of directional data on the toroidal surface using bivariate von Mises distributions *(under preparation)*.

Bibliography

M. Abramowitz and I. A. Stegun. *Handbook of Mathematical Functions*. Dover, New York, 1965.

Y. Agusta and D. L. Dowe. MML clustering of continuous-valued data using Gaussian and t distributions. In *Proceedings of the 15th Australian Joint Conference on Artificial Intelligence: Advances in Artificial Intelligence*, pages 143–154. Springer-Verlag, 2002.

Y. Agusta and D. L. Dowe. Unsupervised learning of correlated multivariate Gaussian mixture models using MML. In *Advances in Artificial Intelligence*, pages 477–489. Springer, Heidelberg, 2003.

H. Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6):716–723, 1974.

D. E. Amos. Computation of modified Bessel functions and their ratios. *Mathematics of Computation*, 28(125):239–251, 1974.

E. Anderson. The Irises of the Gaspé Peninsula. *Bulletin of the American Iris Society*, 59:2–5, 1935.

L. J. Bain and M. Engelhardt. Interval estimation for the two-parameter double exponential distribution. *Technometrics*, 15(4):875–887, 1973.

A. Banerjee, I. Dhillon, J. Ghosh, and S. Sra. Generative model-based clustering of directional data. In *Proceedings of the Ninth International Conference on Knowledge Discovery and Data Mining*, pages 19–28, New York, 2003.

A. Banerjee, I. Dhillon, J. Ghosh, and S. Sra. Clustering on the unit hypersphere using von Mises-Fisher distributions. *Journal of Machine Learning Research*, 6:1345–1382, 2005.

D. J. Barlow and J. M. Thornton. Helix geometry in proteins. *Journal of Molecular Biology*, 201(3): 601–619, 1988.

D. E. Barton. Unbiased estimation of a set of probabilities. *Biometrika*, 48(1-2):227–229, 1961.

A. P. Basu. Estimates of reliability for some distributions useful in life testing. *Technometrics*, 6(2): 215–219, 1964.

J.-P. Baudry and G. Celeux. EM for mixtures. *Statistics and Computing*, 25(4):713–726, 2015.

R. E. Bellman. *Dynamic Programming*. Princeton University Press, NJ, USA, 1957.

D. J. Best and N. I. Fisher. The bias of the maximum likelihood estimators of the von Mises-Fisher concentration parameters. *Communications in Statistics-Simulation and Computation*, 10(5):493–502, 1981.

D. Bhowmick, A. C. Davison, D. R. Goldstein, and Y. Ruffieux. A Laplace mixture model for identification of differential expression in microarray experiments. *Biostatistics*, 7(4):630–641, 2006.

C. Biernacki, G. Celeux, and G. Govaert. Assessing a mixture model for clustering with the integrated completed likelihood. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(7):719–725, 2000.

C. Bingham and K. V. Mardia. A small circle distribution on the sphere. *Biometrika*, 65(2):379–389, 1978.

C. M. Bishop. *Pattern Recognition and Machine Learning*, volume 1. Springer, New York, 2006.

D. Blackwell. Conditional expectation and unbiased sequential estimation. *The Annals of Mathematical Statistics*, 18:105–110, 1947.

W. Boomsma, J. T. Kent, K. V. Mardia, C. C. Taylor, and T. Hamelryck. Graphical models and directional statistics capture protein structure. *Interdisciplinary Statistics and Bioinformatics*, 25: 91–94, 2006.

N. Bouguila and D. Ziou. High-dimensional unsupervised selection and estimation of a finite generalized Dirichlet mixture model based on minimum message length. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(10):1716–1731, 2007.

D. M. Boulton and C. S. Wallace. The information content of a multistate distribution. *Journal of Theoretical Biology*, 23:269–278, 1969.

D. M. Boulton and C. S. Wallace. A program for numerical classification. *The Computer Journal*, 13 (1):63–69, 1970.

D. M. Boulton and C. S. Wallace. An information measure for hierarchic classification. *The Computer Journal*, 16(3):254–261, 1973.

D. M. Boulton and C. S. Wallace. An information measure for single link classification. *The Computer Journal*, 18(3):236–238, 1975.

M. J. Bower, F. E. Cohen, and R. L. Dunbrack. Prediction of protein side-chain rotamers from a backbone-dependent rotamer library: a new homology modeling tool. *Journal of Molecular Biology*, 267(5):1268–1282, 1997.

H. Bozdogan. Determining the number of component clusters in the standard multivariate normal mixture model using model-selection criteria. Technical report, DTIC Document, 1983.

H. Bozdogan. On the information-based measure of covariance complexity and its application to the evaluation of multivariate linear models. *Communications in Statistics-Theory and Methods*, 19(1): 221–278, 1990.

H. Bozdogan. Choosing the number of component clusters in the mixture-model using a new informational complexity criterion of the inverse-fisher information matrix. In *Information and Classification*, Studies in Classification, Data Analysis and Knowledge Organization, pages 40–54. Springer Berlin Heidelberg, 1993.

K. P. Burnham and D. R. Anderson. *Model selection and multimodel inference: a practical information-theoretic approach*. Springer, New York, 2002.

G. J. Chaitin. On the length of programs for computing finite binary sequences. *Journal of the Association for Computing Machinery*, 13(4):547–569, 1966.

H. Chen, J. Chen, and J. D. Kalbfleisch. A modified likelihood ratio test for homogeneity in finite mixture models. *Journal of the Royal Statistical Society: Series B (Methodological)*, 63(1):19–29, 2001.

J. Chen and P. Cheng. On testing the number of components in finite mixture models with known relevant component distributions. *Canadian Journal of Statistics*, 25(3):389–400, 1997.

J. Chen and P. Li. Hypothesis test for normal mixture models: The EM approach. *The Annals of Statistics*, 37(5A):2523–2542, 2009.

Y. W. Chen, T. Tajima, and S. Agrawal. The crystal structure of the Ubiquitin-like (UbL) domain of human homologue A of Rad23 (hHR23A) protein. *Protein Engineering Design and Selection*, 24 (1-2):131–138, 2011.

V. Cherkassky, F. Mulier, and V. Vapnik. Comparison of VC-method with classical methods for model selection. In *Proceedings of World Congress on Neural Networks*, pages 957–962, 1997.

C. Chothia and A. V. Finkelstein. The classification and origins of protein folding patterns. *Annual Review of Biochemistry*, 59(1):1007–1035, 1990.

C. Chothia, M. Levitt, and D. Richardson. Helix to helix packing in proteins. *Journal of Molecular Biology*, 145(1):215–250, 1981.

F. E. Cohen, M. J. E. Sternberg, and W. R. Taylor. Analysis of the tertiary structure of protein β-sheet sandwiches. *Journal of Molecular Biology*, 148(3):253–272, 1981.

J. H. Collier, L. Allison, A. M. Lesk, M. G. de la Banda, and A. S. Konagurthu. A new statistical framework to assess structural alignment quality using information compression. *Bioinformatics*, 30(17):i512–i518, 2014.

N. Colloc'h, C. Etchebest, E. Thoreau, B. Henrissat, and J.-P. Mornon. Comparison of three algorithms for the assignment of secondary structure in proteins: the advantages of a consensus assignment. *Protein Engineering*, 6(4):377–382, 1993.

J. H. Conway and N. J. A. Sloane. On the Voronoi regions of certain lattices. *SIAM Journal on Algebraic and Discrete Methods*, 5:294–305, 1984.

A. Cord, C. Ambroise, and J. P. Cocquerez. Feature selection in robust clustering based on Laplace mixture. *Pattern Recognition Letters*, 27(6):627–635, 2006.

G. M. Cordeiro and R. Klein. Bias correction in ARMA models. *Statistics & Probability Letters*, 19 (3):169–176, 1994.

G. M. Cordeiro and K. L. P. Vasconcellos. Theory & Methods: Second-order biases of the maximum likelihood estimates in von Mises regression models. *Australian & New Zealand Journal of Statistics*, 41(2):189–198, 1999.

J. A. Cuff and G. J. Barton. Evaluation and improvement of multiple sequence methods for protein secondary structure prediction. *Proteins*, 34(4):508–519, 1999.

H. E. Daniels. The asymptotic efficiency of a maximum likelihood estimator. In *Fourth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1. University of California Press Berkeley, 1961.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–38, 1977.

L. H. G. Dore, G. J. A. Amaral, J. T. M. Cruz, and A. T. A. Wood. Bias-corrected maximum likelihood estimation of the parameters of the complex Bingham distribution. *Brazilian Journal of Probability and Statistics*, forthcoming.

D. L. Dowe, L. Allison, T. I. Dix, L. Hunter, C. S. Wallace, and T. Edgoose. Circular clustering of protein dihedral angles by minimum message length. In *Pacific Symposium on Biocomputing*, volume 96, pages 242–255, 1996a.

D. L. Dowe, J. J. Oliver, R. A. Baxter, and C. S. Wallace. Bayesian estimation of the von Mises concentration parameter. In *Maximum Entropy and Bayesian Methods*, pages 51–60. Springer, Netherlands, 1996b.

D. L. Dowe, J. J. Oliver, and C. S. Wallace. MML estimation of the parameters of the spherical Fisher distribution. In *Proceedings of the Seventh International Workshop on Algorithmic Learning Theory*, pages 213–227. Springer, Heidelberg, 1996c.

D. L. Dowe, R. A. Baxter, J. J. Oliver, and C. S. Wallace. Point estimation using the Kullback-Leibler loss function and MML. In *Research and Development in Knowledge Discovery and Data Mining*, pages 87–95. Springer, 1998.

I. L. Dryden and K. V. Mardia. *Statistical Shape Analysis*, volume 4. Wiley Chichester, 1998.

R. L. Dunbrack and F. E. Cohen. Bayesian statistical analysis of protein side-chain rotamer preferences. *Protein Science*, 6(8):1661–1681, 1997.

A. J. Duncan. *Quality Control and Industrial Statistics.* Richard D. Irwin Publishers, Homewood, IL, USA, 1974.

F. Dupuis, J. F. Sadoc, and J. P. Mornon. Protein secondary structure assignment through Voronoi tessellation. *Proteins*, 55:519–528, 2004.

P. S. Dwyer. Some applications of matrix derivatives in multivariate analysis. *Journal of the American Statistical Association*, 62(318):607–625, 1967.

M. L. Eaton and C. N. Morris. The application of invariance to unbiased estimation. *The Annals of Mathematical Statistics*, 41(5):1708–1716, 1970.

J. T. Edsall, P. J. Flory, J. C. Kendrew, A. M. Liquori, G. Némethy, G. N. Ramachandran, and H. A. Scheraga. A proposal of standard conventions and nomenclature for the description of polypeptide conformation. *The Journal of Biological Chemistry*, 241(4):1004, 1966.

P. Elias. Universal codeword sets and representations of the integers. *IEEE Transactions on Information Theory*, 21(2):194–203, 1975.

P. R. Elliott, X. Y. Pei, T. R. Dafforn, and D. A. Lomas. Topography of a 2.0 Å structure of α1-antitrypsin reveals targets for rational drug design to prevent conformational disease. *Protein Science*, 9(7):1274–1281, 2000.

T. Eltoft, T. Kim, and T. W. Lee. On the multivariate Laplace distribution. *IEEE Signal Processing Letters*, 13(5):300–303, 2006.

G. E. Farr and C. S. Wallace. The complexity of strict minimum message length inference. *The Computer Journal*, 45(3):285–292, 2002.

A. M. S. Figueiredo. Goodness-of-fit for a concentrated von Mises-Fisher distribution. *Computational Statistics*, 27(1):69–82, 2012.

M. A. T. Figueiredo and A. K. Jain. Unsupervised learning of finite mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(3):381–396, 2002.

N. I. Fisher. *Statistical Analysis of Spherical Data.* Cambridge University Press, Cambridge, 1993.

R. Fisher. Dispersion on a sphere. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 217(1130):295–305, 1953.

R. A. Fisher. Theory of statistical estimation. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 22, pages 700–725. Cambridge Univ Press, 1925.

R. A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7(2): 179–188, 1936.

L. J. Fitzgibbon, D. L. Dowe, and L. Allison. Univariate polynomial inference by Monte Carlo message length approximation. In *Proceedings of the Nineteenth International Conference on Machine Learning*, pages 147–154. Morgan Kaufmann Publishers, 2002.

J. Fourier. *Theorie analytique de la chaleur, par M. Fourier.* Chez Firmin Didot, père et fils, 1822.

J. Gauvain and C. H. Lee. Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains. *IEEE Transactions on Speech and Audio Processing*, 2(2):291–298, 1994.

S. Gopal and Y. Yang. Von Mises-Fisher clustering models. In *Proceedings of The 31st International Conference on Machine Learning*, pages 154–162, 2014.

G. Gray. Bias in misspecified mixtures. *Biometrics*, 50(2):457–470, 1994.

P. D. Grünwald. *The Minimum Description Length Principle.* MIT Press, Cambridge, MA, USA, 2007.

M. Haas, S. Mittnik, and M. S. Paolella. Modelling and predicting market risk with Laplace–Gaussian mixture distributions. *Applied Financial Economics*, 16(15):1145–1162, 2006.

T. Hamelryck. Probabilistic models and machine learning in structural bioinformatics. *Statistical Methods in Medical Research*, 18(5):505–526, 2009.

T. Hamelryck, J. T. Kent, and A. Krogh. Sampling realistic protein conformations using local structural bias. *PLoS Computational Biology*, 2(9):e131, 2006.

L. Holm and C. Sander. Protein structure comparison by alignment of distance matrices. *Journal of Molecular Biology*, 233(1):123–138, 1993.

A. K. Jain and R. C. Dubes. *Algorithms for Clustering Data.* Prentice-Hall Inc., Upper Saddle River, NJ, USA, 1988.

A. K. Jain, R. P. W. Duin, and J. Mao. Statistical Pattern Recognition: A Review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):4–37, 2000.

H. Jeffreys. An invariant form for the prior probability in estimation problems. *Proceedings of the Royal Society of London: Series A. (Mathematical and Physical Sciences)*, 186(1007):453–461, 1946.

S. G. Johnson. The NLopt nonlinear-optimization package. `http://ab-initio.mit.edu/nlopt`. 2014.

M. A. Jorgensen and G. J. McLachlan. Wallace's approach to unsupervised learning: the Snob program. *The Computer Journal*, 51(5):571–578, 2008.

P. E. Jupp and K. V. Mardia. A general correlation coefficient for directional data and related regression problems. *Biometrika*, 67(1):163–173, 1980.

W. Kabsch and C. Sander. Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers*, 22:2577–637, 1983.

P. Kasarapu. Modelling of directional data using Kent distributions. `http://arxiv.org/abs/1506.08105`. 2015.

P. Kasarapu and L. Allison. Minimum message length estimation of mixtures of multivariate Gaussian and von Mises-Fisher distributions. *Machine Learning*, 100(2-3):333–378, 2015.

P. Kasarapu, M. G. de la Banda, and A. S. Konagurthu. On representing protein folding patterns using non-linear parametric curves. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 11(6):1218–1228, 2014.

J. T. Kent. The Fisher-Bingham distribution on the sphere. *Journal of the Royal Statistical Society: Series B (Methodological)*, 44(1):71–80, 1982.

J. T. Kent and T. Hamelryck. Using the Fisher-Bingham distribution in stochastic models for protein structure. *Quantitative Biology, Shape Analysis, and Wavelets*, 24:57–60, 2005.

J. T. Kent, A. M. Ganeiber, and K. V. Mardia. A new method to simulate the Bingham and related distributions in directional data analysis with applications. `arXiv:1310.8110[math.ST]`. 2013.

A. N. Kolmogorov. Three approaches to the quantitative definition of information. *Problems of Information Transmission*, 1(1):1–7, 1965.

A. S. Konagurthu, P. J. Stuckey, and A. M. Lesk. Structural search and retrieval using a tableau representation of protein folding patterns. *Bioinformatics*, 24(5):645–651, 2008.

A. S. Konagurthu, L. Allison, P. J. Stuckey, and A. M. Lesk. Piecewise linear approximation of protein structures using the principle of minimum message length. *Bioinformatics*, 27(13):i43–i51, 2011.

A. S. Konagurthu, A. M. Lesk, and L. Allison. Minimum message length inference of secondary structure from protein coordinate data. *Bioinformatics*, 28(12):i97–i105, 2012.

A. S. Konagurthu, L. Allison, D. Abramson, P. J. Stuckey, and A. M. Lesk. Statistical inference of protein "LEGO bricks". In *Thirteenth International Conference on Data Mining (ICDM)*, pages 1091–1096. IEEE, 2013.

S. Kotz, T. J. Kozubowski, and K. Podgórski. Asymmetric multivariate Laplace distribution. In *The Laplace Distribution and Generalizations*, pages 239–272. Springer, 2001.

T. Krishnan and G. J. McLachlan. *The EM Algorithm and Extensions*. Wiley, New York, 1997.

S. Kullback and R. A. Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, 1951.

A. Kume and A. T. A. Wood. On the derivatives of the normalising constant of the Bingham distribution. *Statistics & Probability Letters*, 77(8):832–837, 2007.

D. Kundu. Discriminating between Normal and Laplace distributions. In *Advances in Ranking and Selection, Multiple Comparisons, and Reliability*, Statistics for Industry and Technology, pages 65–79. Birkhäuser Boston, 2005.

E. Lam. *Improved approximations in MML*. Honours thesis, School of Computer Science and Software Engineering, Monash University, Clayton, Australia, 2000.

J. F. Lawless. *Statistical Models and Methods for Lifetime Data*. Wiley, New York, 1982.

A. R. Leach. *Molecular Modelling: Principles and Applications*. Pearson Education, 2001.

G. Lebanon. Consistency of the maximum likelihood estimator. *http://www.cc.gatech.edu/~lebanon/notes/mleconsistency.pdf*, 2008.

G. Lebanon. Bias, Variance, and MSE of estimators, 2010.

P. Lee. *Bayesian Statistics: An Introduction.* Arnold, London., 1997.

A. M. Legendre. Recherches sur l'attraction des sphéroïdes homogènes. *Mémoires de Mathématiques et de Physique, présentés à l'Académie Royale des Sciences, par divers savans, et lus dans ses Assemblées*, pages 411–435, 1785.

A. M. Lesk. Systematic representation of protein folding patterns. *Journal of Molecular Graphics*, 13 (3):159–164, 1995.

A. M. Lesk. *Introduction to Protein Science: Architecture, Function, and Genomics.* Oxford University Press, Oxford, UK, 2004.

M. Levitt and J. Greer. Automatic identification of secondary structure in globular proteins. *Journal of Molecular Biology*, 114(2):181–239, 1977.

M. Li and P. M. B. Vitányi. *An Introduction to Kolmogorov Complexity and its Applications.* Springer Verlag, New York, 1997.

P. Li and J. Chen. Testing the order of a finite mixture. *Journal of the American Statistical Association*, 105(491):1084–1092, 2010.

Y. Lo. Bias from misspecification of the component variances in a normal mixture. *Computational Statistics and Data Analysis*, 55(9):2739–2747, 2011.

Y. Lo, N. R. Mendell, and D. B. Rubin. Testing the number of components in a normal mixture. *Biometrika*, 88(3):767–778, 2001.

A. H. Louie and R. L. Somorjai. Differential geometry of proteins: Helical approximations. *Journal of Molecular Biology*, 168(1):143 – 162, 1983.

S. C. Lovell, I. W. Davis, W. B. Arendall, P. I. W. de Bakker, J. M. Word, M. G. Prisant, J. S. Richardson, and D. C. Richardson. Structure validation by Cα geometry: ϕ, ψ and Cβ deviation. *Proteins: Structure, Function, and Genetics*, 50(3):437–450, 2003.

M. L. Ludwig, K. A. Pattridge, A. L. Metzger, M. M. Dixon, M. Eren, Y. Feng, and R. P. Swenson. Control of oxidation-reduction potentials in flavodoxin from Clostridium beijerinckii: the role of conformation changes. *Biochemistry*, 36(6):1259–1280, 1997.

J. R. Magnus and H. Neudecker. *Matrix Differential Calculus with Applications in Statistics and Econometrics.* Wiley, New York, 1988.

I. Majumdar, S. S. Krishna, and N. V. Grishin. PALSSE: A program to delineate linear secondary structural elements from protein structures. *BMC Bioinformatics*, 6:202, 2005.

E. Makalic and L. Allison. MMLD inference of multilayer perceptrons. In *Algorithmic Probability and Friends. Bayesian Prediction and Artificial Intelligence*, pages 261–272. Springer, 2013.

E. Makalic and D. F. Schmidt. Efficient linear regression by minimum message length. Technical report, Monash University, 2006.

K. V. Mardia. Distribution theory for the von Mises-Fisher distribution and its application. In *A Modern Course on Statistical Distributions in Scientific Work*, pages 113–130. Springer, Netherlands, 1975a.

K. V. Mardia. Statistics of directional data (with discussion). *Journal of the Royal Statistical Society: Series B (Methodological)*, 37:349–393, 1975b.

K. V. Mardia. Characterizations of Directional Distributions. In *A Modern Course on Statistical Distributions in Scientific Work*, volume 17, pages 365–385. Springer, Netherlands, 1975c.

K. V. Mardia and P. E. Jupp. *Directional Statistics.* Wiley, Hoboken, NJ, USA, 2000.

K. V. Mardia, D. Holmes, and J. T. Kent. A goodness-of-fit test for the von Mises-Fisher distribution. *Journal of the Royal Statistical Society: Series B (Methodological)*, 46(1):72–78, 1984.

K. V. Mardia, C. C. Taylor, and G. K. Subramaniam. Protein bioinformatics and mixtures of bivariate von Mises distributions for angular data. *Biometrics*, 63(2):505–512, 2007.

K. V. Mardia, G. Hughes, C. C. Taylor, and H. Singh. A multivariate von Mises distribution with applications to bioinformatics. *Canadian Journal of Statistics*, 36(1):99–109, 2008.

G. J. McLachlan and K. E. Basford. *Mixture models: Inference and Applications to Clustering (Statistics: Textbooks and Monographs).* Dekker, New York, 1988.

G. J. McLachlan and D. Peel. Contribution to the discussion of paper by S. Richardson and P. J. Green. *Journal of the Royal Statistical Society: Series B (Methodological)*, 59:779–780, 1997.

G. J. McLachlan and D. Peel. *Finite Mixture Models.* Wiley, New York, 2000.

M. Menke, B. Berger, and L. Cowen. Matt: local flexibility aids protein multiple structure alignment. *PLoS Computational Biology*, 4(1):e10, 2008.

K. P. Murphy. *Machine Learning: A Probabilistic Perspective.* MIT Press, Cambridge, MA, USA, 2012.

A. G. Murzin, S. E. Brenner, T. Hubbard, and C. Chothia. SCOP: a structural classification of proteins database for the investigation of sequences and structures. *Journal of Molecular Biology*, 247(4):536–540, 1995.

W. K. Newey and D. McFadden. Large sample estimation and hypothesis testing. *Handbook of Econometrics*, 4:2111–2245, 1994.

R. M. Norton. The double exponential distribution: Using calculus to find a maximum likelihood estimator. *The American Statistician*, 38(2):135–136, 1984.

J. J. Oliver and R. A. Baxter. MDL and MML: Similarities and differences (introduction to minimum encoding inference). Technical report, Monash University, 1994.

J. J. Oliver and D. Hand. Introduction to minimum encoding inference. Technical report, Dept. of Statistics, Open University, Walton Hall, Milton Keynes, UK, 1994.

J. J. Oliver, R. A. Baxter, and C. S. Wallace. Unsupervised learning using MML. In *Proceedings of the Thirteenthth International Conference on Machine Learning*, pages 364–372. Morgan Kaufmann Publishers, 1996.

K. Pearson. Note on Regression and Inheritance in the Case of Two Parents. *Proceedings of the Royal Society of London*, 58(347-352):240–242, 1895.

D. Peel, W. J. Whiten, and G. J. McLachlan. Fitting mixtures of Kent distributions to aid in joint set identification. *Journal of the American Statistical Association*, 96(453):56–63, 2001.

M. J. D. Powell. A direct search optimization method that models the objective and constraint functions by linear interpolation. In *Advances in Optimization and Numerical Analysis*, pages 51–67. Kluwer Academic Publishers, Dordrecht, Netherlands, 1994.

W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C++: The Art of Scientific Computing*. Cambridge University Press, 2002.

P. Puig and M. A. Stephens. Tests of fit for the Laplace distribution, with applications. *Technometrics*, 42(4):417–424, 2000.

H. Rabbani, M. Vafadust, and S. Gazor. Image denoising based on a mixture of Laplace distributions with local parameters in complex wavelet domain. In *IEEE International Conference on Image Processing*, pages 2597–2600. IEEE, 2006.

G. N. Ramachandran, C. T. Ramakrishnan, and V. Sasisekharan. Stereochemistry of polypeptide chain configurations. *Journal of Molecular Biology*, 7(1):95–99, 1963.

S. Ranganathan, D. Izotov, E. Kraka, and D. Cremer. Description and recognition of regular and distorted secondary structures in proteins using the automated protein structure analysis method. *Proteins*, 76(2):418–438, 2009.

C. R. Rao. Information and accuracy attainable in the estimation of statistical parameters. *Bulletin of Calcutta Mathematical Society*, 37:81–91, 1945.

C. R. Rao. *Linear Statistical Inference and its Applications*. Wiley, New York, 1973.

F. M. Richards and C. E. Kundrot. Identification of structural motifs from protein coordinate data: Secondary structure and first-level supersecondary structure. *Proteins*, 3(2):71–84, 1988.

J. S. Richardson. Beta-sheet topology and the relatedness of proteins. *Nature*, 268:495–500, 1977.

J. S. Richardson. The anatomy and taxonomy of protein structure. volume 34 of *Advances in Protein Chemistry*, pages 167 – 339. 1981.

S. Richardson and P. J. Green. On Bayesian analysis of mixtures with an unknown number of components. *Journal of the Royal Statistical Society: Series B (Methodological)*, 59(4):731–792, 1997.

J. Rissanen. Modeling by shortest data description. *Automatica*, 14(5):465–471, 1978.

J. Rissanen. *Stochastic Complexity in Statistical Inquiry Theory*. World Scientific, River Edge, NJ, USA, 1989.

L. P. Rivest. A distribution for dependent unit vectors. *Communications in Statistics-Theory and Methods*, 17(2):461–483, 1988.

S. J. Roberts, D. Husmeier, I. Rezek, and W. Penny. Bayesian approaches to Gaussian mixture modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(11):1133–1142, 1998.

S. Robertson and H. Zaragoza. *The probabilistic relevance framework: BM25 and beyond*. Now Publishers Inc., Hanover, MA, USA, 2009.

P. Røgen and B. Fain. Automatic classification of protein structure by using Gauss integrals. *Proceedings of the National Academy of Sciences*, 100(1):119–124, 2003.

M. Rosenblatt. Remarks on a multivariate transformation. *The Annals of Mathematical Statistics*, 23 (3):470–472, 1952.

G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5):513–523, 1988.

G. Salton and M. J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, Inc., New York, NY, USA, 1986.

D. F. Schmidt and E. Makalic. MML invariant linear regression. In *Advances in Artificial Intelligence*, pages 312–321. Springer, 2009.

D. F. Schmidt and E. Makalic. Minimum message length inference and mixture modelling of Inverse Gaussian distributions. In *Proceedings of the Twenty fifth Australasian Joint Conference on Advances in Artificial Intelligence*, pages 672–682. Springer-Verlag, 2012.

G. Schou. Estimation of the concentration parameter in von Mises–Fisher distributions. *Biometrika*, 65(2):369–377, 1978.

G. Schwarz. Estimating the dimension of a model. *The Annals of Statistics*, 6(2):461–464, 1978.

C. E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27: 379–423, 1948.

D. Sherwood and J. Cooper. *Crystals, X-Rays and Proteins: Comprehensive Protein Crystallography*. Oxford University Press, Oxford, UK, 2010.

S. Shi, B. Chitturi, and N. V. Grishin. ProSMoS server: a pattern-based search using interaction matrix representation of protein structures. *Nucleic Acids Research*, 37(suppl 2):W526–W531, 2009.

H. Singh, V. Hnizdo, and E. Demchuk. Probabilistic model for two dependent circular variables. *Biometrika*, 89(3):719–723, 2002.

H. Sklenar, C. Etchebest, and R. Lavery. Describing protein structure: a general algorithm yielding complete helicoidal parameters and a unique overall axis. *Proteins*, 6:46–60, 1989.

R. L. Smith. A survey of nonregular problems. In *Proceedings of International Statistical Institute*, volume 47, pages 353–372, 1989.

R. J. Solomonoff. A formal theory of inductive inference I, II. *Information and Control*, 7(1):1–22, 224–254, 1964.

H. Song, J. Liu, and G. Wang. High-order parameter approximation for von Mises–Fisher distributions. *Applied Mathematics and Computation*, 218(24):11880–11890, 2012.

S. Sra. A short note on parameter approximation for von Mises-Fisher distributions: and a fast implementation of $I_s(x)$. *Computational Statistics*, 27(1):177–190, 2012.

A. Strehl, J. Ghosh, and R. Mooney. Impact of similarity measures on web-page clustering. In *Workshop on Artificial Intelligence for Web Search (AAAI 2000)*, pages 58–64, 2000.

M. Taboga. *Lectures on Probability Theory and Mathematical Statistics*. CreateSpace Independent Pub., 2012.

A. Tanabe, K. Fukumizu, S. Oba, T. Takenouchi, and S. Ishii. Parameter estimation for von Mises–Fisher distributions. *Computational Statistics*, 22(1):145–157, 2007.

W. R. Taylor. Defining linear segments in protein structure. *Journal of Molecular Biology*, 310(5): 1135–1150, 2001.

W. R. Taylor, J. M. Thornton, and W. G. Turnell. An ellipsoidal approximation of protein shape. *Journal of Molecular Graphics*, 1(2):30–38, 1983.

D. M. Titterington, A. F. M. Smith, and U. E. Makov. *Statistical Analysis of Finite Mixture Distributions*. Wiley, New York, 1985.

N. Ueda, R. Nakano, Z. Ghahramani, and G. E. Hinton. SMEM algorithm for mixture models. *Neural Computation*, 12(9):2109–2128, 2000.

M. Viswanathan and C. S. Wallace. A note on the comparison of polynomial selection methods. In *Proceedings of the Seventh International Workshop on Artificial Intelligence and Statistics*, volume 99, pages 169–177, Fort Lauderdale, FL, USA, 1999. Morgan Kaufmann.

A. Wald. Note on the consistency of the maximum likelihood estimate. *The Annals of Mathematical Statistics*, 20(4):595–601, 1949.

C. S. Wallace. An improved program for classification. In *Proceedings of the Ninth Australian Computer Science Conference*, pages 357–366, 1986.

C. S. Wallace. On the selection of the order of a polynomial model. Technical report, Royal Holloway College, 1997.

C. S. Wallace. *Statistical and Inductive Inference using Minimum Message Length*. Springer-Verlag, Secaucus, NJ, USA, 2005.

C. S. Wallace and D. M. Boulton. An information measure for classification. *The Computer Journal*, 11(2):185–194, 1968.

C. S. Wallace and D. M. Boulton. An invariant Bayes method for point estimation. *Classification Society Bulletin*, 3(3):11–34, 1975.

C. S. Wallace and M. B. Dale. Hierarchical clusters of vegetation types. *Community Ecology*, 6(1): 57–74, 2005.

C. S. Wallace and D. L. Dowe. Intrinsic Classification by MML – the Snob Program. In *Proceedings of the Seventh Australian Joint Conference on Artificial Intelligence*, pages 37–44. World Scientific, 1994a.

C. S. Wallace and D. L. Dowe. Estimation of the von Mises concentration parameter using minimum message length. In *Proceedings of the 12th Australian Statistical Society Conference, Monash University, Australia*, 1994b.

C. S. Wallace and D. L. Dowe. Minimum message length and Kolmogorov complexity. *The Computer Journal*, 42:270–283, 1999.

C. S. Wallace and D. L. Dowe. MML clustering of multi-state, Poisson, von Mises circular and Gaussian distributions. *Statistics Computing*, pages 73–83, 2000.

C. S. Wallace and P. R. Freeman. Estimation and inference by compact coding. *Journal of the Royal Statistical Society: Series B (Methodological)*, 49(3):240–265, 1987.

G. S. Watson and E. J. Williams. On the construction of significance tests on the circle and the sphere. *Biometrika*, 43(3-4):344–352, 1956.

E. W. Weisstein. "Fourier Series." From MathWorld–A Wolfram web resource. `http://mathworld.wolfram.com/FourierSeries.html`. a.

E. W. Weisstein. "Legendre Polynomial." From MathWorld–A Wolfram web resource. `http://mathworld.wolfram.com/LegendrePolynomial.html`. b.

H. White. Maximum likelihood estimation of misspecified models. *Econometrica*, 50(1):1–25, 1982.

S. S. Wilks. The large-sample distribution of the likelihood ratio for testing composite hypotheses. *The Annals of Mathematical Statistics*, 9(1):60–62, 1938.

A. T. A. Wood. Some notes on the Fisher–Bingham family on the sphere. *Communications in Statistics-Theory and Methods*, 17(11):3881–3897, 1988.

A. T. A. Wood. Simulation of the von Mises Fisher distribution. *Communications in Statistics-Simulation and Computation*, 23(1):157–164, 1994.

L. Xu and M. I. Jordan. On convergence properties of the EM algorithm for Gaussian mixtures. *Neural Computation*, 8(1):129–151, 1996.

Y. Zhang and J. Skolnick. TM-align: a protein structure alignment algorithm based on the TM-score. *Nucleic Acids Research*, 33(7):2302–2309, 2005.

D. Ziou and N. Bouguila. Unsupervised learning of a finite Gamma mixture using MML: application to SAR image analysis. In *Proceedings of the 17th International Conference on Pattern Recognition*, volume 2, pages 68–71. IEEE, 2004.

Appendix A

A.1 Supporting derivations required for evaluating the MML estimates, κ_{MN} and κ_{MH}

For brevity, let $A_d(\kappa), A'_d(\kappa), A''_d(\kappa)$, and $A'''_d(\kappa)$ be represented as A, A', A'', and A''' respectively. The expressions to evaluate A, A', and A'' are given by Equations 4.4), 4.8, and 4.10) respectively. We require A''' for its use in the remainder of the derivation. Its expression is provided below

$$\frac{A'''}{A'} = -\frac{2AA''}{A'} - 2A' - \frac{(d-1)}{\kappa}\frac{A''}{A'} - \frac{2(d-1)}{\kappa^3}\frac{A}{A'} + \frac{2(d-1)}{\kappa^2}$$

In the following discussion, the derivations of $G'(\kappa)$ and $G''(\kappa)$ that are required for computing the two versions of the MML estimate of the vMF concentration parameter, κ_{MN} (Equation 4.14) and κ_{MH} (Equation 4.15) are given. On differentiating Equation 4.13, we get

$$G'(\kappa) = \frac{(d-1)}{2\kappa^2} + (d+1)\frac{(1-\kappa^2)}{(1+\kappa^2)^2} + \frac{(d-1)}{2}\frac{\partial}{\partial\kappa}\left(\frac{A'}{A}\right) + \frac{1}{2}\frac{\partial}{\partial\kappa}\left(\frac{A''}{A'}\right) + NA'$$

$$\text{and } G''(\kappa) = -\frac{(d-1)}{\kappa^3} + (d+1)\frac{2\kappa(\kappa^2-3)}{(1+\kappa^2)^3} + \frac{(d-1)}{2}\frac{\partial^2}{\partial\kappa^2}\left(\frac{A'}{A}\right) + \frac{1}{2}\frac{\partial^2}{\partial\kappa^2}\left(\frac{A''}{A'}\right) + NA''$$

Using Equations 4.4 and 4.8, we have

$$\begin{aligned}
\frac{A'}{A} &= \frac{1}{A} - A - \frac{(d-1)}{\kappa} \\
\frac{\partial}{\partial\kappa}\left(\frac{A'}{A}\right) &= -\frac{A'}{A^2} - A' + \frac{(d-1)}{\kappa^2} \\
\frac{\partial^2}{\partial\kappa^2}\left(\frac{A'}{A}\right) &= 2\frac{(A')^2}{A^3} - \frac{A''}{A^2} - A'' - \frac{2(d-1)}{\kappa^3}
\end{aligned} \tag{A.1}$$

Using Equations 4.4, 4.8, 4.10, and the above given expression for A''', we have

$$\begin{aligned}
\frac{A''}{A'} &= -2A - \frac{(d-1)}{\kappa} + \frac{(d-1)}{\kappa^2}\frac{A}{A'} \\
\frac{\partial}{\partial\kappa}\left(\frac{A''}{A'}\right) &= -2A' + \frac{2(d-1)}{\kappa^2} - \frac{(d-1)}{\kappa^3}\frac{A}{A'}\left(\frac{\kappa A''}{A'} + 2\right) \\
\frac{\partial^2}{\partial\kappa^2}\left(\frac{A''}{A'}\right) &= -2A'' - \frac{4(d-1)}{\kappa^3} - (d-1)\frac{\partial}{\partial\kappa}\left(\frac{AA''}{\kappa^2 A'^2}\right) - 2(d-1)\frac{\partial}{\partial\kappa}\left(\frac{A}{\kappa^3 A'}\right)
\end{aligned} \tag{A.2}$$

$$\text{where} \quad \frac{\partial}{\partial\kappa}\left(\frac{AA''}{\kappa^2 A'^2}\right) = \frac{\kappa AA'A''' + \kappa A'^2 A'' - 2\kappa AA''^2 - 2AA'A''}{\kappa^3 A'^3}$$
$$\text{and} \quad \frac{\partial}{\partial\kappa}\left(\frac{A}{\kappa^3 A'}\right) = \frac{1}{\kappa^3} - \frac{A}{\kappa^4 A'^2}(\kappa A'' + 3A')$$

Equations A.1 and A.2 can be used to evaluate $G'(\kappa)$ and $G''(\kappa)$ which can then be used to approximate the MML estimates κ_{MN} and κ_{MH} respectively.

A.2 Derivation of the Kullback-Leibler (KL) distance between two vMF distributions

The closed form expression to calculate the KL distance between two vMF distributions is derived below. Let $f(\mathbf{x}) = C_d(\kappa_1)\exp\{\kappa_1 \boldsymbol{\mu_1}^T\mathbf{x}\}$ and $g(\mathbf{x}) = C_d(\kappa_2)\exp\{\kappa_2\boldsymbol{\mu_2}^T\mathbf{x}\}$ be two vMF distributions with mean directions $\boldsymbol{\mu_1}, \boldsymbol{\mu_2}$ and concentration parameters κ_1, κ_2 respectively. Recall from Section 2.5.2, the KL distance between any two distributions is given by

$$D_{KL}(f||g) = \int_{\mathbf{x}} f(\mathbf{x}) \log \frac{f(\mathbf{x})}{g(\mathbf{x})} d\mathbf{x}$$

where $\mathbb{E}_f[\cdot]$ is the expectation of the quantity $[\cdot]$ using the probability density function f. For a vMF distribution,

$$\log \frac{f(\mathbf{x})}{g(\mathbf{x})} = \log \frac{C_d(\kappa_1)}{C_d(\kappa_2)} + (\kappa_1\boldsymbol{\mu_1} - \kappa_2\boldsymbol{\mu_2})^T\mathbf{x}$$

Using the fact that $\mathbb{E}_f[\mathbf{x}] = A_d(\kappa_1)\boldsymbol{\mu_1}$ (Mardia et al., 1984; Fisher, 1993), we obtain the following expression for the KL distance for vMF distributions as

$$\mathrm{E}_f\left[\log \frac{f(\mathbf{x})}{g(\mathbf{x})}\right] = \log \frac{C_d(\kappa_1)}{C_d(\kappa_2)} + (\kappa_1\boldsymbol{\mu_1} - \kappa_2\boldsymbol{\mu_2})^T A_d(\kappa_1)\boldsymbol{\mu_1}$$
$$D_{KL}(f||g) = \log \frac{C_d(\kappa_1)}{C_d(\kappa_2)} + A_d(\kappa_1)(\kappa_1 - \kappa_2\boldsymbol{\mu_1}^T\boldsymbol{\mu_2}) \tag{A.3}$$

Appendix B

B.1 Prior density governed by the κ prior for the 2D vMF

In the MML estimation of parameters of a vMF distribution on a circle, Wallace and Dowe (1994b) use $\Pr(\kappa) = \frac{\kappa}{(1+\kappa^2)^{3/2}}$. In the following discussion, this prior is considered as this leads to an invertible transformation of all five parameters, as described below, in the context of a FB_5 distribution. As in Section 4.3.3, $\Pr(\psi, \alpha, \eta) = \frac{\sin\alpha}{4\pi^2}$ and $\Pr(\beta|\kappa) = 2/\kappa$. Hence, the joint prior density $\Pr(\mathbf{\Theta})$ is formulated as shown below. Further, using the eccentricity transform described previously, the joint prior density $\Pr(\mathbf{\Theta'})$ in $\mathbf{\Theta'}$ parameterization is given below.

$$\Pr(\mathbf{\Theta}) = \Pr(\psi, \alpha, \eta, \kappa, \beta) = \frac{\sin\alpha}{2\pi^2(1+\kappa^2)^{3/2}} \quad \text{and} \quad \Pr(\mathbf{\Theta'}) = \Pr(\psi, \alpha, \eta, \kappa, e) = \frac{\kappa\sin\alpha}{4\pi^2(1+\kappa^2)^{3/2}}$$

An alternative parameterization of the parameter vector $\mathbf{\Theta}$

In addition to the eccentricity transform, we study another transformation that was proposed by Rosenblatt (1952). We used this transformation in the context of alternative parameterizations of the prior density for BVM Sine distributions (Section 4.4.2). The Rosenblatt (1952) transformation applied on the prior density of the FB_5 parameter vector $\mathbf{\Theta}$ results in the prior transforming to a uniform distribution. Hence, estimation in this transformed parameter space is equivalent to the corresponding maximum likelihood estimation.

For the 5-parameter vector $\mathbf{\Theta} = (\psi, \alpha, \eta, \kappa, \beta)$, the Rosenblatt (1952) transformation to $\mathbf{\Theta''} = (z_1, z_2, z_3, z_4, z_5)$ is given by

$$\begin{aligned}
z_1 &= \Pr(X_1 \leq \psi) = F_1(\psi) \\
z_2 &= \Pr(X_2 \leq \alpha | X_1 = \psi) = F_2(\alpha|\psi) \\
z_3 &= \Pr(X_3 \leq \eta | X_2 = \alpha, X_1 = \psi) = F_3(\eta|\alpha, \psi) \\
z_4 &= \Pr(X_4 \leq \kappa | X_3 = \eta, X_2 = \alpha, X_1 = \psi) = F_4(\kappa|\eta, \alpha, \psi) \\
z_5 &= \Pr(X_5 \leq \beta | X_4 = \kappa, X_3 = \eta, X_2 = \alpha, X_1 = \psi) = F_5(\beta|\kappa, \eta, \alpha, \psi)
\end{aligned}$$

This transformation results in $0 \leq z_i \leq 1, i = 1, \ldots, 5$. As each z_i is uniformly and independently distributed on $[0, 1]$ (Rosenblatt, 1952), the prior density in this transformed parameter space is $\Pr(\mathbf{\Theta''}) = \Pr(z_1, z_2, z_3, z_4, z_5) = 1$.

In order to achieve such a transformation, we need to express z_i in terms of the original parameters. As per the definitions of the prior on $\mathbf{\Theta}$ (Section 4.3.3), the following relationships are derived.

$$\begin{aligned}
z_1 = \psi/\pi &\implies \psi = \pi z_1 \\
z_2 = (1 - \cos\alpha)/2 &\implies \alpha = \arccos(1 - 2z_2) \\
z_3 = \eta/(2\pi) &\implies \eta = 2\pi z_3
\end{aligned} \tag{B.1}$$

Based on the independence assumption in the formulation of priors of angular and scale parameters, $z_4 = F_4(\kappa|\eta,\alpha,\psi) = F_4(\kappa)$. Similarly, $F_5(\beta|\kappa,\eta,\alpha,\psi) = F_5(\beta|\kappa)$. Hence, the invertible transformations corresponding to κ and β are

$$\begin{aligned} z_4 &= \int_0^{\kappa} h(\kappa)d\kappa = \int_0^{\kappa} \frac{\kappa}{(1+\kappa^2)^{3/2}}d\kappa = 1 - \cos(\arctan\kappa) \implies \kappa = \tan(\arccos(1-z_4)) \\ z_5 &= F_5(\beta|\kappa) = 2\beta/\kappa \implies \beta = \kappa\, z_5/2 \end{aligned} \tag{B.2}$$

With the 3D version of vMF κ prior (Section 4.3.3), z_4 evaluates to $\frac{2}{\pi}\left(\arctan\kappa - \frac{\kappa}{1+\kappa^2}\right)$. This version of z_4 is not invertible as it does not allow us to express κ as a closed form expression in z_4. Hence, the Rosenblatt (1952) transformation is discussed only in the context when 2D vMF κ prior is considered, as it is possible to find an inverse transformation.

The example demonstrating the effects of alternative parameterizations

The above discussed prior and its variants are used in the MAP-based parameter estimation of the data from the example discussed in Section 4.3.3. The resulting estimates of ψ, α, and η are

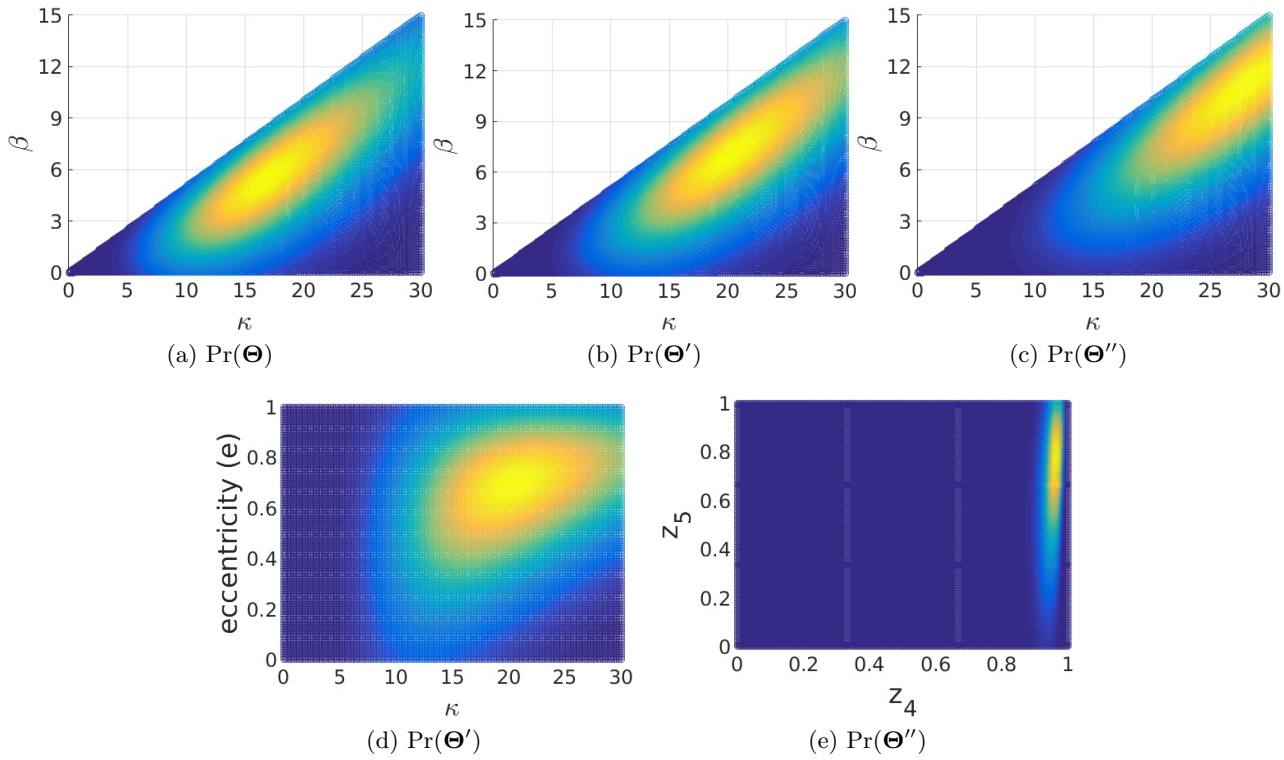
$$\begin{aligned} \Pr(\boldsymbol{\Theta}) &: \widehat{\psi} = 2.070,\ \widehat{\alpha} = 1.493,\ \widehat{\eta} = 1.522 \\ \Pr(\boldsymbol{\Theta}') &: \widehat{\psi} = 2.070,\ \widehat{\alpha} = 1.493,\ \widehat{\eta} = 1.522 \\ \Pr(\boldsymbol{\Theta}'') &: \widehat{z}_1 = 0.659,\ \widehat{z}_2 = 0.461,\ \widehat{z}_3 = 0.242 \end{aligned}$$

As observed, $\widehat{\psi}, \widehat{\alpha}$, and $\widehat{\eta}$ are the same when posteriors corresponding to $\Pr(\boldsymbol{\Theta})$ and $\Pr(\boldsymbol{\Theta}')$ are used. In the case of $\Pr(\boldsymbol{\Theta}'')$, the mapping of $\widehat{z}_1, \widehat{z}_2, \widehat{z}_3$ back to $\widehat{\psi}, \widehat{\alpha}, \widehat{\eta}$ (Equation B.1), results in the same estimates as that of $\Pr(\boldsymbol{\Theta})$ and $\Pr(\boldsymbol{\Theta}')$. Hence, the MAP estimates of ψ, α, η are same across the different versions. The estimates of κ and β are, however, not the same under the various transformations. Their values vary depending on the parameterization and are

$$\begin{aligned} \Pr(\boldsymbol{\Theta}) &: \widehat{\kappa} = 16.975,\ \widehat{\beta} = 5.467 \\ \Pr(\boldsymbol{\Theta}') &: \widehat{\kappa} = 20.547,\ \widehat{e} = 0.701 \implies \widehat{\beta} = \widehat{\kappa}\,\widehat{e}/2 = 7.205 \\ \Pr(\boldsymbol{\Theta}'') &: \widehat{z}_4 = 0.964,\ \widehat{z}_5 = 0.779 \implies \widehat{\kappa} = 28.065,\ \widehat{\beta} = 10.925 \quad \text{(as per Equation B.2)} \end{aligned}$$

The estimated value of κ using $\Pr(\boldsymbol{\Theta})$ is 16.975 whereas it is 20.547 using $\Pr(\boldsymbol{\Theta}')$. The value of $\widehat{e}$ corresponds to a $\widehat{\beta} = 7.205$. Similarly, the value of $\widehat{\kappa}$ and $\widehat{\beta}$ corresponding to $\widehat{z}_4 = 0.964$ and $\widehat{z}_5 =$ 0.0.779 are 28.065 and 10.925 respectively. Clearly, the value of the parameter estimates depend on the parameterization. However, it is required that the estimates obtained in different parameterizations should be the same irrespective of the space in which the parameters are defined. However, through this example, it is observed that for the various parameterizations, the value of MAP estimates differ.

The variation of the posterior density under various transformations of the parameter space are shown in Figure B.1. These are plotted as a function of κ, β (in case of $\Pr(\boldsymbol{\Theta})$), κ, e (in case of $\Pr(\boldsymbol{\Theta}')$), and z_4, z_5 (in case of $\Pr(\boldsymbol{\Theta}'')$), each reflecting the space in which the posterior is defined. It is observed that the modes of the respective posterior distributions occur at different positions and they are not equivalent to each other. The posterior density plots in Figure B.1(b) and (c) correspond to those in Figure B.1(d) and (e) respectively. Ideally, (the modes in) Figure B.1(a)-(c) should be the same. However, as demonstrated, that is not the case. Thus, maximizing the posterior density does not yield consistent estimates as observed through this example.



(a) $\Pr(\boldsymbol{\Theta})$ (b) $\Pr(\boldsymbol{\Theta}')$ (c) $\Pr(\boldsymbol{\Theta}'')$

(d) $\Pr(\boldsymbol{\Theta}')$ (e) $\Pr(\boldsymbol{\Theta}'')$

Figure B.1: Heat maps depicting the modes (MAP estimates) of the posterior density resulting from different parameterizations.

B.2 The partial derivatives of $\boldsymbol{\gamma}_1, \boldsymbol{\gamma}_2, \boldsymbol{\gamma}_3$ with respect to ψ, α, η

The first and second order partial derivatives of the axes are required for the evaluation of the elements of the Fisher information matrix (see Section 4.3.4). The expressions for $\boldsymbol{\gamma}_1, \boldsymbol{\gamma}_2$, and $\boldsymbol{\gamma}_3$ as a function of ψ, α, and η are given by Equation 4.19.

Derivatives of $\boldsymbol{\gamma}_1$

$$\frac{\partial \boldsymbol{\gamma}_1}{\partial \alpha} = \begin{pmatrix} \cos\alpha \\ \sin\alpha\cos\eta \\ \sin\alpha\sin\eta \end{pmatrix}, \quad \frac{\partial \boldsymbol{\gamma}_1}{\partial \eta} = \begin{pmatrix} 0 \\ -\sin\alpha\sin\eta \\ \sin\alpha\cos\eta \end{pmatrix}$$

$$\frac{\partial^2 \boldsymbol{\gamma}_1}{\partial \alpha^2} = -\boldsymbol{\gamma}_1, \quad \frac{\partial^2 \boldsymbol{\gamma}_1}{\partial \eta^2} = \begin{pmatrix} 0 \\ -\sin\alpha\cos\eta \\ -\sin\alpha\sin\eta \end{pmatrix}, \quad \frac{\partial^2 \boldsymbol{\gamma}_1}{\partial \eta \partial \alpha} = \begin{pmatrix} 0 \\ -\cos\alpha\sin\eta \\ \cos\alpha\cos\eta \end{pmatrix}$$

The partial derivatives of $\boldsymbol{\gamma}_1$ involving the parameter ψ are zero vectors.

Derivatives of $\boldsymbol{\gamma}_2$

$$\frac{\partial \boldsymbol{\gamma}_2}{\partial \alpha} = -\cos\psi\,\boldsymbol{\gamma}_1\,, \quad \frac{\partial \boldsymbol{\gamma}_2}{\partial \eta} = \begin{pmatrix} 0 \\ -\cos\psi\cos\alpha\sin\eta - \sin\psi\cos\eta \\ \cos\psi\cos\alpha\cos\eta - \sin\psi\sin\eta \end{pmatrix}, \quad \frac{\partial \boldsymbol{\gamma}_2}{\partial \psi} = \boldsymbol{\gamma}_3$$

$$\frac{\partial^2 \boldsymbol{\gamma}_2}{\partial \alpha^2} = -\cos\psi\frac{\partial \boldsymbol{\gamma}_1}{\partial \alpha}\,, \quad \frac{\partial^2 \boldsymbol{\gamma}_2}{\partial \eta^2} = \begin{pmatrix} 0 \\ -\cos\psi\cos\alpha\cos\eta + \sin\psi\sin\eta \\ -\cos\psi\cos\alpha\sin\eta - \sin\psi\cos\eta \end{pmatrix}, \quad \frac{\partial^2 \boldsymbol{\gamma}_2}{\partial \psi^2} = -\boldsymbol{\gamma}_2$$

$$\frac{\partial^2 \boldsymbol{\gamma}_2}{\partial \eta \partial \alpha} = -\cos\psi\frac{\partial \boldsymbol{\gamma}_1}{\partial \eta}\,, \quad \frac{\partial^2 \boldsymbol{\gamma}_2}{\partial \psi \partial \alpha} = \sin\psi\,\boldsymbol{\gamma}_1\,, \quad \frac{\partial^2 \boldsymbol{\gamma}_2}{\partial \psi \partial \eta} = \frac{\partial \boldsymbol{\gamma}_3}{\partial \eta}$$

Derivatives of $\boldsymbol{\gamma}_3$

$$\frac{\partial \boldsymbol{\gamma}_3}{\partial \alpha} = \sin\psi\,\boldsymbol{\gamma}_1\,, \quad \frac{\partial \boldsymbol{\gamma}_3}{\partial \eta} = \begin{pmatrix} 0 \\ \sin\psi\cos\alpha\sin\eta - \cos\psi\cos\eta \\ -\sin\psi\cos\alpha\cos\eta - \cos\psi\sin\eta \end{pmatrix}, \quad \frac{\partial \boldsymbol{\gamma}_3}{\partial \psi} = -\boldsymbol{\gamma}_2$$

$$\frac{\partial^2 \boldsymbol{\gamma}_3}{\partial \alpha^2} = -\sin\psi\frac{\partial \boldsymbol{\gamma}_1}{\partial \alpha}\,, \quad \frac{\partial^2 \boldsymbol{\gamma}_3}{\partial \eta^2} = \begin{pmatrix} 0 \\ \sin\psi\cos\alpha\cos\eta + \cos\psi\sin\eta \\ \sin\psi\cos\alpha\sin\eta - \cos\psi\cos\eta \end{pmatrix}, \quad \frac{\partial^2 \boldsymbol{\gamma}_3}{\partial \psi^2} = -\boldsymbol{\gamma}_3$$

$$\frac{\partial^2 \boldsymbol{\gamma}_3}{\partial \eta \partial \alpha} = \sin\psi\frac{\partial \boldsymbol{\gamma}_1}{\partial \eta}\,, \quad \frac{\partial^2 \boldsymbol{\gamma}_3}{\partial \psi \partial \alpha} = \cos\psi\,\boldsymbol{\gamma}_1\,, \quad \frac{\partial^2 \boldsymbol{\gamma}_3}{\partial \psi \partial \eta} = -\frac{\partial \boldsymbol{\gamma}_2}{\partial \eta}$$

B.3 Derivation of the KL distance between two FB_5 distributions

The analytical form of the KL distance between two FB_5 distributions is derived below. The KL distance between two probability distributions f_a and f_b is defined by Equation 2.15. Let $f_a(\mathbf{x}) = \mathrm{FB}_5(\kappa_a, \beta_a, \mathbf{Q}_a)$ and $f_b(\mathbf{x}) = \mathrm{FB}_5(\kappa_b, \beta_b, \mathbf{Q}_b)$ be two distributions such that $\mathbf{Q}_a = (\boldsymbol{\gamma}_{a1}, \boldsymbol{\gamma}_{a2}, \boldsymbol{\gamma}_{a3})$ and $\mathbf{Q}_b = (\boldsymbol{\gamma}_{b1}, \boldsymbol{\gamma}_{b2}, \boldsymbol{\gamma}_{b3})$. Let c_a and c_b be their respective normalization constants. Then,

$$\begin{aligned} \mathbb{E}_a\left[\log\frac{f_a(\mathbf{x})}{f_b(\mathbf{x})}\right] &= \log\frac{c_b}{c_a} + (\kappa_a\boldsymbol{\gamma}_{a1}^T - \kappa_b\boldsymbol{\gamma}_{b1}^T)\,\mathbb{E}_a[\mathbf{x}] \\ &\quad + \beta_a\,\boldsymbol{\gamma}_{a2}^T\,\mathbb{E}_a[\mathbf{x}\mathbf{x}^\mathsf{T}]\,\boldsymbol{\gamma}_{a2} - \beta_b\,\boldsymbol{\gamma}_{b2}^T\,\mathbb{E}_a[\mathbf{x}\mathbf{x}^\mathsf{T}]\,\boldsymbol{\gamma}_{b2} \\ &\quad - \beta_a\,\boldsymbol{\gamma}_{a3}^T\,\mathbb{E}_a[\mathbf{x}\mathbf{x}^\mathsf{T}]\,\boldsymbol{\gamma}_{a3} + \beta_b\,\boldsymbol{\gamma}_{b3}^T\,\mathbb{E}_a[\mathbf{x}\mathbf{x}^\mathsf{T}]\,\boldsymbol{\gamma}_{b3} \end{aligned} \tag{B.3}$$

gives the analytical form of the KL distance of two FB_5 distributions. The expressions for $\mathbb{E}_a[\mathbf{x}]$ and $\mathbb{E}_a[\mathbf{x}\mathbf{x}^\mathsf{T}]$ are derived in Equation 4.28.

Appendix C

C.1 Derivation of the KL distance between two BVM Sine distributions

The analytical form of the KL distance between two BVM Sine distributions is derived below. For a datum $\mathbf{x} = (\theta_1, \theta_2)$, where $\theta_1, \theta_2 \in [0, 2\pi)$, let $f_a(\mathbf{x}) = \text{BVM}(\mu_{a1}, \mu_{a2}, \kappa_{a1}, \kappa_{a2}, \lambda_a)$ and $f_b(\mathbf{x}) = \text{BVM}(\mu_{b1}, \mu_{b2}, \kappa_{b1}, \kappa_{b2}, \lambda_b))$ be two BVM Sine distributions whose probability density functions are given by Equation 4.43. Let c_a and c_b be their respective normalization constants, whose expressions are given by Equation 4.44. The computation of the BVM Sine normalization constant is presented in Section 4.4.4.

The KL distance between two probability distributions f_a and f_b is defined by $\mathbb{E}_a\left[\log \frac{f_a(\mathbf{x})}{f_b(\mathbf{x})}\right]$ (Equation 2.15). Using the density function in Equation 4.43, we have

$$\mathbb{E}_a[\log f_a(\mathbf{x})] = -\log c_a + \kappa_{a1}\mathbb{E}_a[\cos(\theta_1 - \mu_{a1})] + \kappa_{a2}\mathbb{E}_a[\cos(\theta_2 - \mu_{a2})] + \lambda_a\mathbb{E}_a[\sin(\theta_1 - \mu_{a1})\sin(\theta_2 - \mu_{a2})]$$

The expressions for the above expectation terms $\mathbb{E}_a[\cos(\theta_1 - \mu_{a1})]$, $\mathbb{E}_a[\cos(\theta_2 - \mu_{a2})]$ and $\mathbb{E}_a[\sin(\theta_1 - \mu_{a1})\sin(\theta_2 - \mu_{a2})]$ can be computed and are given by Equation 4.51. Similarly, the expectation of $\log f_b(\mathbf{x})$ is

$$\mathbb{E}_a[\log f_b(\mathbf{x})] = -\log c_b + \kappa_{b1}\mathbb{E}_a[\cos(\theta_1 - \mu_{b1})] + \kappa_{b2}\mathbb{E}_a[\cos(\theta_2 - \mu_{b2})] + \lambda_b\mathbb{E}_a[\sin(\theta_1 - \mu_{b1})\sin(\theta_2 - \mu_{b2})]$$

In order to compute $\mathbb{E}_a[\cos(\theta_1 - \mu_{b1})]$, we express $\cos(\theta_1 - \mu_{b1})$ as

$$\begin{aligned}\cos(\theta_1 - \mu_{b1}) &= \cos(\theta_1 - \mu_{a1} + \mu_{a1} - \mu_{b1}) \\ &= \cos(\theta_1 - \mu_{a1})\cos(\mu_{a1} - \mu_{b1}) - \sin(\theta_1 - \mu_{a1})\sin(\mu_{a1} - \mu_{b1})\end{aligned}$$

Given that $\mathbb{E}_a[\sin(\theta_1 - \mu_{a1})] = 0$ (Equation 4.51), we have

$$\begin{aligned}\mathbb{E}_a[\cos(\theta_1 - \mu_{b1})] &= \cos(\mu_{a1} - \mu_{b1})\,\mathbb{E}_a[\cos(\theta_1 - \mu_{a1})] \\ \text{Similarly, } \mathbb{E}_a[\cos(\theta_2 - \mu_{b2})] &= \cos(\mu_{a2} - \mu_{b2})\,\mathbb{E}_a[\cos(\theta_2 - \mu_{a2})]\end{aligned}$$

In order to compute $\mathbb{E}_a[\sin(\theta_1 - \mu_{b1})\sin(\theta_2 - \mu_{b2})]$, we express the product of the sine terms as

$$\sin(\theta_1 - \mu_{b1})\sin(\theta_2 - \mu_{b2}) = \sin(\theta_1 - \mu_{a1} + \mu_{a1} - \mu_{b1})\sin(\theta_2 - \mu_{a2} + \mu_{a2} - \mu_{b2})$$

Further, using the property that $\mathbb{E}_a[\cos(\theta_1 - \mu_{a1})\sin(\theta_2 - \mu_{a2})] = \mathbb{E}[\sin(\theta_1 - \mu_{a1})\cos(\theta_2 - \mu_{a2})] = 0$ (Equation 4.52), we have

$$\begin{aligned}\mathbb{E}_a[\sin(\theta_1 - \mu_{b1})\sin(\theta_2 - \mu_{b2})] &= \cos(\mu_{a1} - \mu_{b1})\cos(\mu_{a2} - \mu_{b2})\mathbb{E}_a[\sin(\theta_1 - \mu_{a1})\sin(\theta_2 - \mu_{a2})] \\ &\quad + \sin(\mu_{a1} - \mu_{b1})\sin(\mu_{a2} - \mu_{b2})\mathbb{E}_a[\cos(\theta_1 - \mu_{a1})\cos(\theta_2 - \mu_{a2})]\end{aligned}$$

Then, the KL distance between the two distributions f_a and f_b is derived as

$$\begin{aligned}\mathbb{E}_a\left[\log\frac{f_a(\mathbf{x})}{f_b(\mathbf{x})}\right] &= \log\frac{c_b}{c_a} + \{\kappa_{a1} - \kappa_{b1}\cos(\mu_{a1}-\mu_{b1})\}\,\mathbb{E}_a[\cos(\theta_1-\mu_{a1})] \\ &+ \{\kappa_{a2} - \kappa_{b2}\cos(\mu_{a2}-\mu_{b2})\}\,\mathbb{E}_a[\cos(\theta_2-\mu_{a2})] \\ &+ \{\lambda_a - \lambda_b\cos(\mu_{a1}-\mu_{b1})\cos(\mu_{a2}-\mu_{b2})\}\,\mathbb{E}_a[\sin(\theta_1-\mu_{a1})\sin(\theta_2-\mu_{a2})] \\ &- \lambda_b\sin(\mu_{a1}-\mu_{b1})\sin(\mu_{a2}-\mu_{b2}) \qquad \text{(C.1)}\end{aligned}$$

gives the analytical form of the KL distance of two BVM Sine distributions.

Special case ($\lambda = 0$): The BVM Sine distribution reduces to the product of two individual von Mises circular distributions given by Equation 4.45. To compute the KL distance between two BVM Independent distributions, we can use Equation C.1, with $\lambda = 0$. Alternatively, we may use the KL distance derived in the case of vMF distributions (Equation A.3) and substitute $d = 2$ in that equation to derive a simpler form.

As per Equations 4.1 and 4.4, note that when $d = 2, C_d(\kappa) = \dfrac{1}{2\pi I_0(\kappa)}$, and $A_d(\kappa) = \dfrac{I_1(\kappa)}{I_0(\kappa)}$, where $I_0(\kappa)$ and $I_1(\kappa)$ are the modified Bessel functions. The KL distance between the BVM Independent distributions f_a and f_b is

$$\begin{aligned}\mathbb{E}_a\left[\log\frac{f_a(\mathbf{x})}{f_b(\mathbf{x})}\right] &= \log\frac{I_0(\kappa_{b1})}{I_0(\kappa_{a1})} + \frac{I_1(\kappa_{a1})}{I_0(\kappa_{a1})}\{\kappa_{a1} - \kappa_{b1}\cos(\mu_{a1}-\mu_{b1})\} \\ &+ \log\frac{I_0(\kappa_{b2})}{I_0(\kappa_{a2})} + \frac{I_1(\kappa_{a2})}{I_0(\kappa_{a2})}\{\kappa_{a2} - \kappa_{b2}\cos(\mu_{a2}-\mu_{b2})\} \qquad \text{(C.2)}\end{aligned}$$

Appendix D

D.1 Derivation of the weight estimates in MML mixture modelling

As per Equation 5.5, the total message length expression can be expressed as

$$I(\mathbf{\Phi}, \mathcal{D}) = -\frac{1}{2}\sum_{j=1}^{K} w_j - \sum_{i=1}^{N} \log \sum_{j=1}^{K} w_j f_j(\mathbf{x}_i; \mathbf{\Theta}_j) + \text{terms independent of } w_j$$

To obtain the optimal weights under the constraint $\sum_{j=1}^{K} w_j = 1$, the above equation is optimized using the *Lagrangian* objective function $L(\mathbf{\Phi}, \mathcal{D}, \lambda)$ defined below using the *Lagrangian multiplier* λ.

$$L(\mathbf{\Phi}, \mathcal{D}, \lambda) = I(\mathbf{\Phi}, \mathcal{D}) + \lambda \left(\sum_{j=1}^{K} w_j - 1 \right)$$

For $k \in \{1, K\}$, the equation resulting from computing the partial derivative of L with respect to w_k and equating it to zero gives the optimal weight w_k.

$$\frac{\partial L}{\partial w_k} = 0 \implies \lambda = \frac{1}{2w_k} + \sum_{i=1}^{N} \frac{f_k(\mathbf{x}_i; \mathbf{\Theta}_k)}{\sum_{j=1}^{K} w_j f_j(\mathbf{x}_i; \mathbf{\Theta}_j)} \tag{D.1}$$

We have $$\sum_{i=1}^{N} \frac{f_k(\mathbf{x}_i; \mathbf{\Theta}_k)}{\sum_{j=1}^{K} w_j f_j(\mathbf{x}_i; \mathbf{\Theta}_j)} = \frac{1}{w_k} \sum_{i=1}^{N} \frac{w_k f_k(\mathbf{x}_i; \mathbf{\Theta}_k)}{\sum_{j=1}^{K} w_j f_j(\mathbf{x}_i; \mathbf{\Theta}_j)} = \frac{1}{w_k} \sum_{i=1}^{N} r_{ik} = \frac{n_k}{w_k}$$

where r_{ik} and n_k are the responsibility and effective membership terms given as per Equations 5.3 and 5.4 respectively. Substituting the above value in Equation D.1, we have

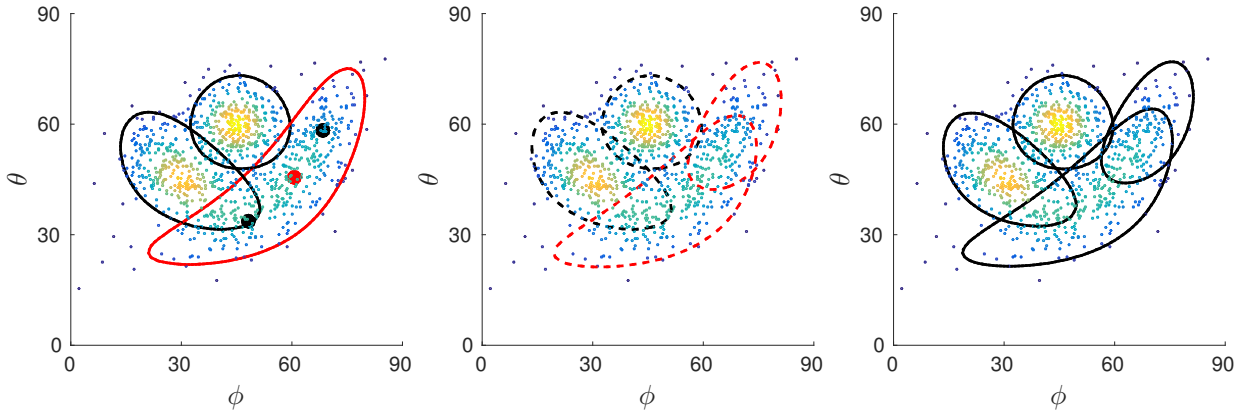
$$\lambda = \frac{1}{2w_k} + \frac{n_k}{w_k} \implies \lambda w_k = n_k + \frac{1}{2} \tag{D.2}$$

There are K equations similar to Equation D.2 for values of $k \in \{1, K\}$. Adding all these equations together, we get

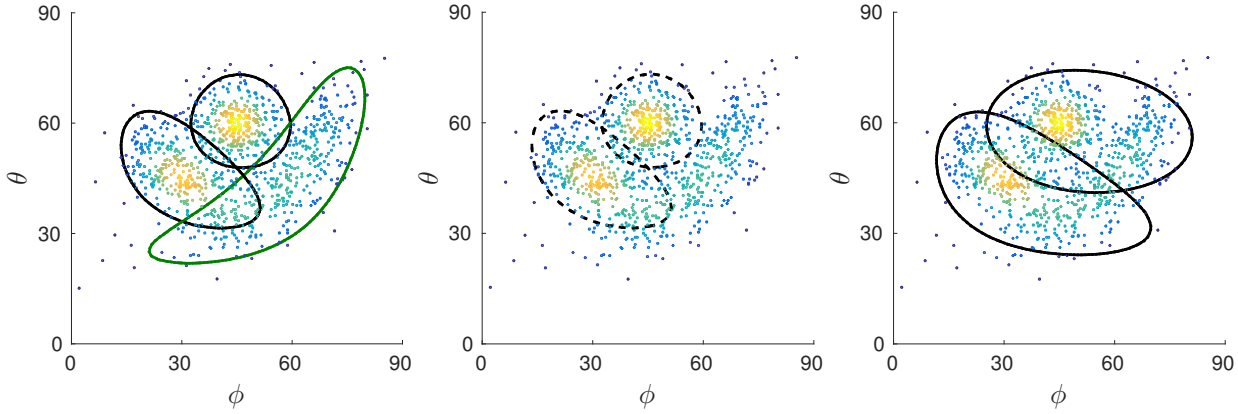
$$\lambda \sum_{j=1}^{K} w_j = \sum_{j=1}^{K} n_j + \frac{K}{2} \implies \lambda = N + \frac{K}{2}$$

Substituting the above value of λ in Equation(D.2), we get $w_k = \dfrac{n_k + \frac{1}{2}}{N + \frac{K}{2}}$.

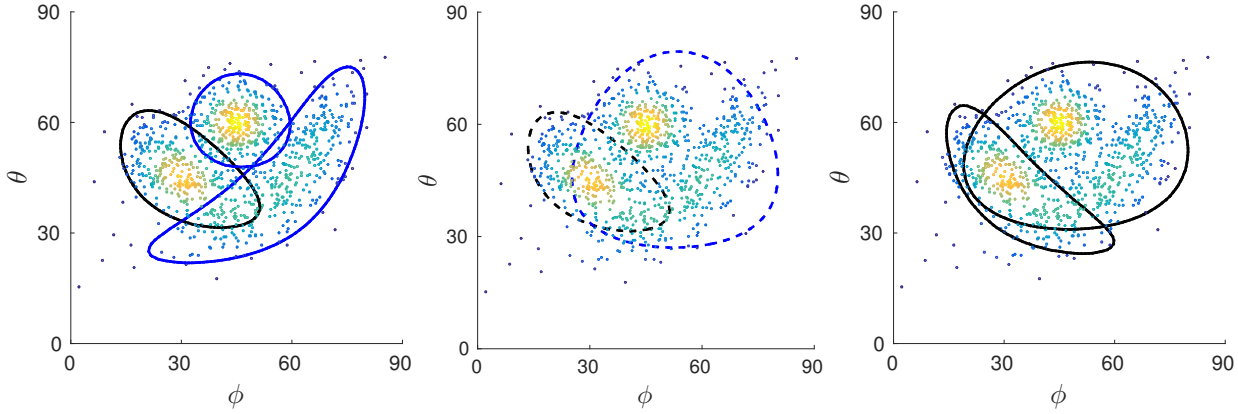
D.2 The perturbations of the FB_5 component: example from Section 6.3.2



(a) Initialize means of children (b) Children ($I = 19186$) (c) $\mathcal{M}_4$: post-EM ($I = 19174$)



(d) (e) Before optimizing ($I = 20741$) (f) post-EM ($I = 19320$)



(g) (h) Before optimizing ($I = 19371$) (i) post-EM ($I = 19312$)

Figure D.1: Iteration 3 - operations on P_1 (a)-(c) splitting, (d)-(f) deletion, (g)-(i) merging

Appendix E

E.1 MML estimates of the parameters of the regression problem

The following derivations are based on the setup described in Section 7.2 and are reproduced from Wallace (2005). The parameters describing the complete regression model (Equation 7.3) are $\mathbf{\Theta} = \{\sigma, w_1, \ldots, w_K\}$. As per the Wallace and Freeman (1987) approximation described in Section 2.4.2, a prior needs to be defined on the parameters, and the expected Fisher information needs to be computed to encode the parameters.

First part message length: encoding the parameters

A locally uninformative prior is assumed for the standard deviation σ of the distribution from which the random values of noise are sampled. If the range of $\log \sigma$ is R, then the prior on σ is given as $h(\sigma) = \dfrac{1}{R\sigma}$. It is further assumed that the weights w_j, $1 \leq j \leq K$ are independently distributed and have a Gaussian prior with zero mean and variance $\dfrac{\sigma^2}{m}$, where m is a constant, that is, $h(w_j) \sim \mathcal{N}\left(0, \dfrac{\sigma^2}{m}\right)$. Hence, the joint prior on $\mathbf{\Theta} = (\sigma, \mathbf{w})$ will be

$$h(\mathbf{\Theta}) = h(\sigma, \mathbf{w}) = \frac{1}{\mathrm{R}\sigma} \prod_{j=1}^{M} h(w_j) \tag{E.1}$$

Computation of the Fisher information $\mathcal{F}(\mathbf{\Theta})$: The computation of the expected Fisher requires the evaluation of the expectation of the second order partial derivatives of the negative log-likelihood expression with respect to the parameters $\mathbf{\Theta}$. Based on Equation 7.3, the likelihood of data $\mathcal{D} = \{(x_i, y_i)\}$, $1 \leq i \leq N$, is expressed as

$$\begin{aligned} f(\mathcal{D}|\mathbf{\Theta}) &= f(\mathcal{D}|\sigma, \mathbf{w}) = f(\mathcal{D}|\sigma) h(\mathbf{w}|\sigma) \\ &= \prod_{i=1}^{N} \frac{\epsilon}{\sqrt{2\pi}\sigma} \exp\left(- \frac{\left(y_i - \sum_{j=1}^{K} w_j \phi_j(x_i) \right)^2}{2\sigma^2} \right) \prod_{j=1}^{K} \frac{\sqrt{m}}{\sqrt{2\pi}\sigma} \exp\left(-\frac{m w_j^2}{2\sigma^2} \right) \end{aligned} \tag{E.2}$$

where ϵ is a constant accuracy of measurement used in MML inference (Section 2.4.1). The negative log-likelihood of the data will then be $\mathcal{L}(\mathcal{D}|\mathbf{\Theta}) = -\log f(\mathcal{D}|\mathbf{\Theta})$ given below

$$\begin{aligned} \mathcal{L}(\mathcal{D}|\mathbf{\Theta}) &= \frac{(N+K)}{2} \log 2\pi - N \log \epsilon - \frac{K}{2} \log m + (N+K) \log \sigma \\ &+ \frac{1}{2\sigma^2} \left[\sum_{i=1}^{N} \left(y_i - \sum_{j=1}^{K} w_j \phi_j(x_i) \right)^2 + m \sum_{j=1}^{K} w_j^2 \right] \end{aligned}$$

For convenience, the above expression for $\mathcal{L}(\mathcal{D}|\boldsymbol{\Theta})$ can be concisely expressed as

$$\mathcal{L}(\mathcal{D}|\boldsymbol{\Theta}) = \frac{(N+K)}{2}\log 2\pi - N\log\epsilon - \frac{K}{2}\log m + (N+K)\log\sigma + \frac{1}{2\sigma^2}\left(||\mathbf{y}-\Phi\mathbf{w}||^2 + m||\mathbf{w}||^2\right) \tag{E.3}$$

$$\text{where, } ||\mathbf{y}-\Phi\mathbf{w}||^2 = (\mathbf{y}-\Phi\mathbf{w})^T(\mathbf{y}-\Phi\mathbf{w}),\ ||\mathbf{w}||^2 = \mathbf{w}^T\mathbf{w}$$

$$\Phi = \begin{bmatrix} \phi_1(x_1) & \phi_1(x_1) & \phi_2(x_1) & \dots & \phi_K(x_1) \\ \phi_1(x_2) & \phi_1(x_2) & \phi_2(x_2) & \dots & \phi_K(x_2) \\ . & . & . & . & . \\ . & . & . & . & . \\ . & . & . & . & . \\ \phi_1(x_N) & \phi_1(x_N) & \phi_2(x_N) & \dots & \phi_K(x_N) \end{bmatrix},\ \mathbf{w} = \begin{bmatrix} w_1 \\ w_2 \\ . \\ . \\ w_K \end{bmatrix}, \text{ and } \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ . \\ . \\ y_N \end{bmatrix}$$

The expectation of the second order partial derivatives of $\mathcal{L}(\mathcal{D}|\boldsymbol{\Theta})$ are computed as

$$\frac{\partial\mathcal{L}}{\partial\sigma} = \frac{(N+K)}{\sigma} - \frac{1}{\sigma^3}\left[(\mathbf{y}-\Phi\mathbf{w})^2 + m\mathbf{w}^2\right]$$

$$\frac{\partial^2\mathcal{L}}{\partial\sigma^2} = -\frac{(N+K)}{\sigma^2} + \frac{3}{\sigma^4}\left[(\mathbf{y}-\Phi\mathbf{w})^2 + m\mathbf{w}^2\right]$$

$$\text{Hence, } \mathrm{E}\left[\frac{\partial^2\mathcal{L}}{\partial\sigma^2}\right] = -\frac{(N+K)}{\sigma^2} + \frac{3}{\sigma^4}\left[N\sigma^2 + K\sigma^2\right] = \frac{2(N+K)}{\sigma^2}$$

$$\frac{\partial\mathcal{L}}{\partial w_j} = \frac{1}{\sigma^2}\sum_{i=1}^{N}\left(y_i - \sum_{j=1}^{M} w_j\phi_j(x_i)\right)(-\phi_j(x_i)) + \frac{m}{\sigma^2}w_j$$

$$\frac{\partial^2\mathcal{L}}{\partial w_j^2} = \frac{1}{\sigma^2}\sum_{i=1}^{N}\phi_j^2(x_i) + \frac{m}{\sigma^2} = \frac{m + \sum_{i=1}^{N}\phi_j^2(x_i)}{\sigma^2}$$

$$\text{Hence, } \mathrm{E}\left[\frac{\partial^2\mathcal{L}}{\partial w_j^2}\right] = \frac{m + \sum_{i=1}^{N}\mathrm{E}\left[\phi_j^2(x_i)\right]}{\sigma^2}$$

$$\frac{\partial^2\mathcal{L}}{\partial w_k\partial w_j} = \frac{1}{\sigma^2}\sum_{i=1}^{N}\phi_k(x_i)\phi_j(x_i)$$

$$\text{Hence, } \mathrm{E}\left[\frac{\partial^2\mathcal{L}}{\partial w_k\partial w_j}\right] = \frac{1}{\sigma^2}\sum_{i=1}^{N}\mathrm{E}\left[\phi_k(x_i)\phi_j(x_i)\right]$$

$$\text{Further, } \frac{\partial^2\mathcal{L}}{\partial\sigma\partial w_j} = \frac{2}{\sigma^3}\sum_{i=1}^{N}\left(y_i - \sum_{j=1}^{K} w_j\phi_j(x_i)\right)\phi_j(x_i) - \frac{2mw_j}{\sigma^3} \implies \mathrm{E}\left[\frac{\partial^2\mathcal{L}}{\partial\sigma\partial w_j}\right] = 0$$

Using the above results, the expected Fisher information would be

$$\mathrm{F}(\sigma,\mathbf{w}) = \begin{pmatrix} \mathrm{E}\left[\frac{\partial^2\mathcal{L}}{\partial\sigma^2}\right] & \mathrm{E}\left[\frac{\partial^2\mathcal{L}}{\partial\sigma\partial w_1}\right] & \mathrm{E}\left[\frac{\partial^2\mathcal{L}}{\partial\sigma\partial w_2}\right] & \dots & \mathrm{E}\left[\frac{\partial^2\mathcal{L}}{\partial\sigma\partial w_K}\right] \\ \mathrm{E}\left[\frac{\partial^2\mathcal{L}}{\partial w_1\partial\sigma}\right] & \mathrm{E}\left[\frac{\partial^2\mathcal{L}}{\partial w_1^2}\right] & \mathrm{E}\left[\frac{\partial^2\mathcal{L}}{\partial w_1\partial w_2}\right] & \dots & \mathrm{E}\left[\frac{\partial^2\mathcal{L}}{\partial w_1\partial w_K}\right] \\ \mathrm{E}\left[\frac{\partial^2\mathcal{L}}{\partial w_2\partial\sigma}\right] & \mathrm{E}\left[\frac{\partial^2\mathcal{L}}{\partial w_2\partial w_1}\right] & \mathrm{E}\left[\frac{\partial^2\mathcal{L}}{\partial w_2^2}\right] & \dots & \mathrm{E}\left[\frac{\partial^2\mathcal{L}}{\partial w_2\partial w_K}\right] \\ . & . & . & . & . \\ . & . & . & . & . \\ \mathrm{E}\left[\frac{\partial^2\mathcal{L}}{\partial w_K\partial\sigma}\right] & \mathrm{E}\left[\frac{\partial^2\mathcal{L}}{\partial w_K\partial w_1}\right] & \mathrm{E}\left[\frac{\partial^2\mathcal{L}}{\partial w_K\partial w_2}\right] & \dots & \mathrm{E}\left[\frac{\partial^2\mathcal{L}}{\partial w_K^2}\right] \end{pmatrix}$$
$$= \frac{2(N+K)}{\sigma^2}\left(\mathrm{E}\left[\frac{\Phi^T\Phi}{\sigma^2} + \frac{m}{\sigma^2}I_K\right]\right)$$

where I_K is the $K \times K$ identity matrix. The determinant of the expected Fisher matrix is

$$|\mathrm{F}(\sigma,\mathbf{w})| = \frac{2(N+K)}{\sigma^2}\frac{|\mathrm{E}\left[\Phi^T\Phi + mI\right]|}{\sigma^{2K}} = \frac{2(N+K)}{\sigma^{2(K+1)}}|Z|$$
$$\text{where} \quad |Z| = \text{determinant}(\mathrm{E}\left[\Phi^T\Phi + mI\right]) \tag{E.4}$$

Using the expressions for the prior density on the parameters (Equation E.1) and the Fisher information as above, the message length corresponding to encoding the parameters (the first part) is formulated using Equation 2.8 and is

$$I(\mathbf{\Theta}) = \frac{p}{2}\log q_p - \log h(\mathbf{\Theta}) + \frac{1}{2}\log|\mathcal{F}(\mathbf{\Theta})|$$
$$= \frac{(K+1)}{2}\log q_p + \log R + \frac{1}{2}\log\left(2(N+K)|Z|\right) - K\log\sigma \tag{E.5}$$

where q_p is the p-dimensional lattice quantization constant with $p = K+1$ (number of free parameters).

Second part message length: encoding the data given the parameters

Using the negative log-likelihood expression (Equation E.3), and the message length expression (Equation 2.8), the second part is formulated as

$$I(\mathcal{D}|\mathbf{\Theta}) = \mathcal{L}(\mathcal{D}|\mathbf{\Theta}) + \frac{p}{2}$$
$$= \frac{(N+K)}{2}\log 2\pi - N\log\epsilon - \frac{K}{2}\log m + (N+K)\log\sigma + \frac{1}{2\sigma^2}\left(||\mathbf{y}-\Phi\mathbf{w}||^2 + m||\mathbf{w}||^2\right) + \frac{(K+1)}{2} \tag{E.6}$$

MML parameter estimates

To determine the MML estimates of the parameters of the regression model, the total message length expression, given by $I(\mathbf{\Theta},\mathcal{D})$, needs to be minimized.

$$I(\mathbf{\Theta},\mathcal{D}) = I(\sigma,\mathbf{w},\mathcal{D}) = N\log\sigma + \frac{1}{2\sigma^2}\left(||\mathbf{y}-\Phi\mathbf{w}||^2 + m||\mathbf{w}||^2\right) + \text{constant}$$

where the constant factor comprises of terms independent of the parameters σ and $\mathbf{w}$. Let the MML estimates of $\mathbf{w}$ and σ be $\hat{\mathbf{w}}$ and $\hat{\sigma}$ respectively. They correspond to the solutions of $\frac{\partial I}{\partial \mathbf{w}} = 0$ and

$\frac{\partial I}{\partial \sigma} = 0$ and are given by

$$\begin{aligned}
\frac{\partial I}{\partial \mathbf{w}} = 0 \Rightarrow \frac{\partial}{\partial \mathbf{w}} \left(||\mathbf{y} - \Phi\mathbf{w}||^2 + m||\mathbf{w}||^2 \right] = 0 \\
\Rightarrow \hat{\mathbf{w}} = \left(\Phi^T \Phi + mI \right)^{-1} \Phi^T \mathbf{y} \\
\frac{\partial I}{\partial \sigma} = 0 \Rightarrow \frac{N}{\hat{\sigma}} - \frac{1}{\hat{\sigma}^3} \left[(\mathbf{y} - \Phi\hat{\mathbf{w}})^2 + m\hat{\mathbf{w}}^2 \right] = 0 \\
\text{Hence,} \quad \hat{\sigma} = \sqrt{\frac{1}{N} \left(||\mathbf{y} - \Phi\hat{\mathbf{w}}||^2 + m||\hat{\mathbf{w}}||^2 \right]}
\end{aligned} \tag{E.7}$$

The MML estimates can be now substituted in the total message length expression $I(\boldsymbol{\Theta}, \mathcal{D})$ to obtain the minimized message length. Note that it is required to evaluate $|Z|$, while computing the Fisher information (Equation E.4). This is required for computing the first part of the message $I(\boldsymbol{\Theta})$.

For orthogonal basis functions, $|Z|$ can be calculated in an explicit form as shown below. If $\phi_j(x), 1 \leq j \leq K$ form an orthonormal basis set, then by definition (Section 7.3),

$$\mathrm{E}\left[\phi_j(x)\phi_k(x)\right] = \begin{cases} N & \text{if } j = k, \\ 0 & \text{if } j \neq k. \end{cases}$$

in which case

$$|Z| = \text{determinant} \left(\mathrm{E}\left[\Phi^T \Phi + mI \right] \right) = (N + m)^K \tag{E.8}$$

Appendix F

F.1 Computation of the second spatial deviation

The point G_m on the Bézier curve (see Figure 8.2) corresponds to the point at which the tangent to the curve is perpendicular to the line joining it and the point F_m. The parametric equation of the Bézier curve is given by

$$\mathbf{p}(t) = \sum_{k=0}^{n} \binom{n}{k} t^k (1-t)^{n-k} \mathbf{p}_k \tag{F.1}$$

Let $G_m \equiv p(t)$ be a point on the curve and $p'(t)$ be the tangent to the curve at point described by the parameter t. Let $F_m \equiv p$ be the point from which the shortest distance is to be measured. The boldface representation ($\mathbf{p}$) indicates that it is a vector comprised of (x, y, z) components. For minimum distance, the dot product of the vector connecting $\mathbf{p}, \mathbf{p}(t)$ and the tangent vector $\mathbf{p'}(t)$ is zero.

$$[\mathbf{p} - \mathbf{p}(t)] \,.\mathbf{p'}(t) = 0 \tag{F.2}$$

In the current work, Bézier curves upto degree 3 are considered, so there are three cases.

1. *Degree 1* corresponds to a straight line between $P_i \equiv p_0$ and $P_j \equiv p_1$. The curve is described using two control points p_0 and p_1. Expanding (F.1),

$$\begin{aligned} \mathbf{p}(t) &= (1-t)\mathbf{p_0} + t\mathbf{p_1} \\ \mathbf{p'}(t) &= -\mathbf{p_0} + \mathbf{p_1} \end{aligned} \tag{F.3}$$

   In this case, the tangent at any point t is a constant for the curve is a straight line. Solving Equations (F.2) and (F.3) gives

$$t = \frac{(\mathbf{p} - \mathbf{p_0}).(\mathbf{p_1} - \mathbf{p_0})}{(\mathbf{p_1} - \mathbf{p_0}).(\mathbf{p_1} - \mathbf{p_0})} \tag{F.4}$$

2. *Degree 2* corresponds to a quadratic curve between $P_i \equiv p_0$ and $P_j \equiv p_2$. The curve is described using three control points – the end points p_0, p_2, and an intermediate control point p_1. Expanding (F.1),

$$\begin{aligned} \mathbf{p}(t) &= (1-t)^2\mathbf{p_0} + 2t(1-t)\mathbf{p_1} + t^2\mathbf{p_2} \\ \mathbf{p'}(t) &= 2(\mathbf{p_0} - 2\mathbf{p_1} + \mathbf{p_2})t + 2(\mathbf{p_1} - \mathbf{p_0}) \end{aligned} \tag{F.5}$$

Solving Equations (F.2) and (F.5) results in a cubic polynomial in t

$$\begin{aligned} At^3 + Bt^2 + Ct + D = 0 \quad &\text{such that} \\ A &= (\mathbf{p_0} - 2\mathbf{p_1} + \mathbf{p_2}).(\mathbf{p_0} - 2\mathbf{p_1} + \mathbf{p_2}) \\ B &= (\mathbf{p_0} - 2\mathbf{p_1} + \mathbf{p_2}).(\mathbf{p_1} - \mathbf{p_0}) \\ C &= (\mathbf{p_0} - 2\mathbf{p_1} + \mathbf{p_2}).(\mathbf{p_0} - \mathbf{p}) + 2(\mathbf{p_1} - \mathbf{p_0}).(\mathbf{p_1} - \mathbf{p_0}) \\ D &= (\mathbf{p_1} - \mathbf{p_0}).(\mathbf{p_0} - \mathbf{p}) \end{aligned}$$

3. *Degree 3* corresponds to a cubic curve between $P_i \equiv p_0$ and $P_j \equiv p_3$. The curve is described using four control points – the end points p_0, p_3, and two intermediate control points p_1 and p_2. Expanding (F.1),

$$\begin{aligned} \mathbf{p}(t) &= (1-t)^3\mathbf{p_0} + 3t(1-t^2)\mathbf{p_1} + 3t^2(1-t)\mathbf{p_2} + t^3\mathbf{p_3} \\ \mathbf{p}'(t) &= 3(-\mathbf{p_0} + 3\mathbf{p_1} - 3\mathbf{p_2} + \mathbf{p_3})t^2 + 6(\mathbf{p_0} - 2\mathbf{p_1} + \mathbf{p_2})t \\ &\quad + 3(\mathbf{p_1} - \mathbf{p_0}) \end{aligned} \tag{F.6}$$

Solving Equations (F.2) and (F.6) results in a quartic in t

$$\begin{aligned} At^5 + Bt^4 + Ct^3 + Dt^2 + Et + F = 0 \quad &\text{such that} \\ A &= (-\mathbf{p_0} + 3\mathbf{p_1} - 3\mathbf{p_2} + \mathbf{p_3}).(-\mathbf{p_0} + 3\mathbf{p_1} - 3\mathbf{p_2} + \mathbf{p_3}) \\ B &= 5(-\mathbf{p_0} + 3\mathbf{p_1} - 3\mathbf{p_2} + \mathbf{p_3}).(\mathbf{p_0} - 2\mathbf{p_1} + \mathbf{p_2}) \\ C &= 6(\mathbf{p_0} - 2\mathbf{p_1} + \mathbf{p_2}).(\mathbf{p_0} - 2\mathbf{p_1} + \mathbf{p_2}) + 4(\mathbf{p_1} - \mathbf{p_0}).(-\mathbf{p_0} + 3\mathbf{p_1} - 3\mathbf{p_2} + \mathbf{p_3}) \\ D &= 9(\mathbf{p_1} - \mathbf{p_0}).(\mathbf{p_0} - 2\mathbf{p_1} + \mathbf{p_2}) + (\mathbf{p_0} - \mathbf{p}).(-\mathbf{p_0} + 3\mathbf{p_1} - 3\mathbf{p_2} + \mathbf{p_3}) \\ E &= 3(\mathbf{p_1} - \mathbf{p_0}).(\mathbf{p_1} - \mathbf{p_0}) + 2(\mathbf{p_0} - \mathbf{p}).(\mathbf{p_0} - 2\mathbf{p_1} + \mathbf{p_2}) \\ F &= (\mathbf{p_1} - \mathbf{p_0}).(\mathbf{p_0} - \mathbf{p}) \end{aligned}$$

The quintic is guaranteed to have one real root. This root is solved for using the Interval Bisection method. A few numerical solvers, namely methods such as Newton-Raphson & Bairstow discussed in Press et al. (2002) have been attempted but none were consistent. The bound on the roots of the polynomial is computed initially, and this interval is used as the starting range for the Bisection method. After the first root is found, the other real roots are computed using numerically stable quartic and cubic solvers. If either the cubic or the quintic had multiple real roots, then the current t^* is chosen to be the one which results in the least $\delta_{m,3}$ value with respect to previous t^*. Once t^* is found, the second deviation would correspond to the distance between $G_m \equiv P(t^*)$ and F_m.

F.2 Case studies of non-linear dissections

F.2.1 Dissection of regions of Clostridium Beijerinckii Flavodoxin: Oxidized

The nature of the dissections generated using the proposed approach is examined using Clostridium Beijerinckii Flavodoxin molecule (Ludwig et al., 1997) with wwPDB code 5NLL. The linear and non-linear representations generated using the approach are further contrasted with each other.

Figure F.1(a) shows the secondary structure assignment of 5NLL. The protein chain is composed of 138 residues. When the standard helices and strands of sheet are replaced by straight lines, we get a representation as depicted in Figure F.1(b). This representation summarized in Table F.1 shows that only 86 ($\sim 62\%$) of the 138 residues are captured using this representation. On the other hand, the representations that result from the proposed approach are able to capture all the protein residues (see Figure F.1c). Here, the resultant *linear* and *non-linear* Bézier abstractions are compared with each other. The linear abstraction produces 13 distinct segments (Table F.2) in contrast to the non-linear

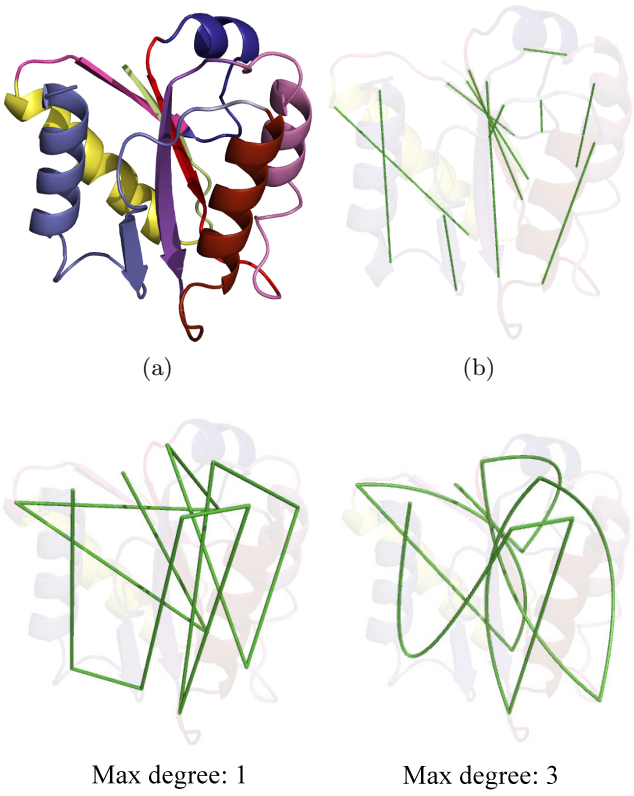


Figure F.1: (a) Structure of Clostridium Beijerinckii Flavodoxin: oxidized (wwPDB code 5NLL) shown in a cartoon representation with standard secondary structures – helices and strands of sheet assigned using DSSP (Kabsch and Sander, 1983). (b) Traditional representation of the folding pattern which replaces each secondary structural element with line segments. (c) The representations produced by our approach by constraining the Bézier curves to a maximum degree of 1 – Linear (left frame), and maximum degree of 3 – Linear through to Cubic (right frame).

Table F.1: The *linear* segmentation of 5NLL obtained by approximating the helices and strands by straight line segments (see Figure F.1b).

Segment index	Residue stretch	Segment index	Residue stretch
1	2 – 6	7	66 – 73
2	11 – 25	8	81 – 88
3	31 – 34	9	94 – 105
4	35 – 37	10	109 – 110
5	40 – 43	11	115 – 118
6	48 – 53	12	122 – 136

Table F.2: The *linear* Bézier segmentation produced for Flavodoxin (5NLL).

Segment index	Residue stretch	Segment index	Residue stretch
1	1 – 8	8	79 – 89
2	8 – 26	9	89 – 107
3	26 – 37	10	107 – 109
4	37 – 46	11	109 – 119
5	46 – 56	12	119 – 122
6	56 – 72	13	122 – 138
7	72 – 79		

Table F.3: The *non-linear* Bézier segmentation produced for Flavodoxin (5NLL).

Segment index	Residue stretch	Segment index	Residue stretch
1	1 – 9	6	58 – 79
2	9 – 27	7	79 – 89
3	27 – 34	8	89 – 107
4	34 – 47	9	107 – 109
5	47 – 58	10	109 – 138

version which generates 10 segments (Table F.3). The notable differences in the segmentations are discussed below.

- Segment 1 (Figure F.3) is abstracted by a quadratic curve while the same segment in Figure F.2 is abstracted by a straight line.

- Segment 3 (Figure F.2) is represented by a straight line that approximates the S-shaped pattern between the residues 26 and 37. However, in Figure F.3, segment 3 is a non-linear curve that approximates the curved region between residues 27 and 34. The curvature between residues 34 and 37 forms the beginning of segment 4, shown in Figure F.3(4).

- The curved section between the residue stretch 46 and 56 is inaccurately represented by a straight line in Figure F.2(5), whereas the same region is represented by a quadratic curve in Figure F.3(5).

- The residue stretch 56 – 79 is represented using two line segments (see segments 6 and 7 in Figure F.2). The same curved region is represented by a quadratic curve in Figure F.3(6).

- The curved region (residues 81 – 88) is represented by a straight line in Figure F.2(8), whereas it is rightly represented by a quadratic curve in Figure F.3(7).

- The region between residues 109 – 138 consists of a helix and a strand of sheet as per the DSSP assigned secondary structure representation. This portion is partitioned by three straight lines in the linear representation shown in Figure F.2(10)-(12), whereas it is abstracted as one non-linear curve in the non-linear Bézier segmentation in Figure F.3(10).

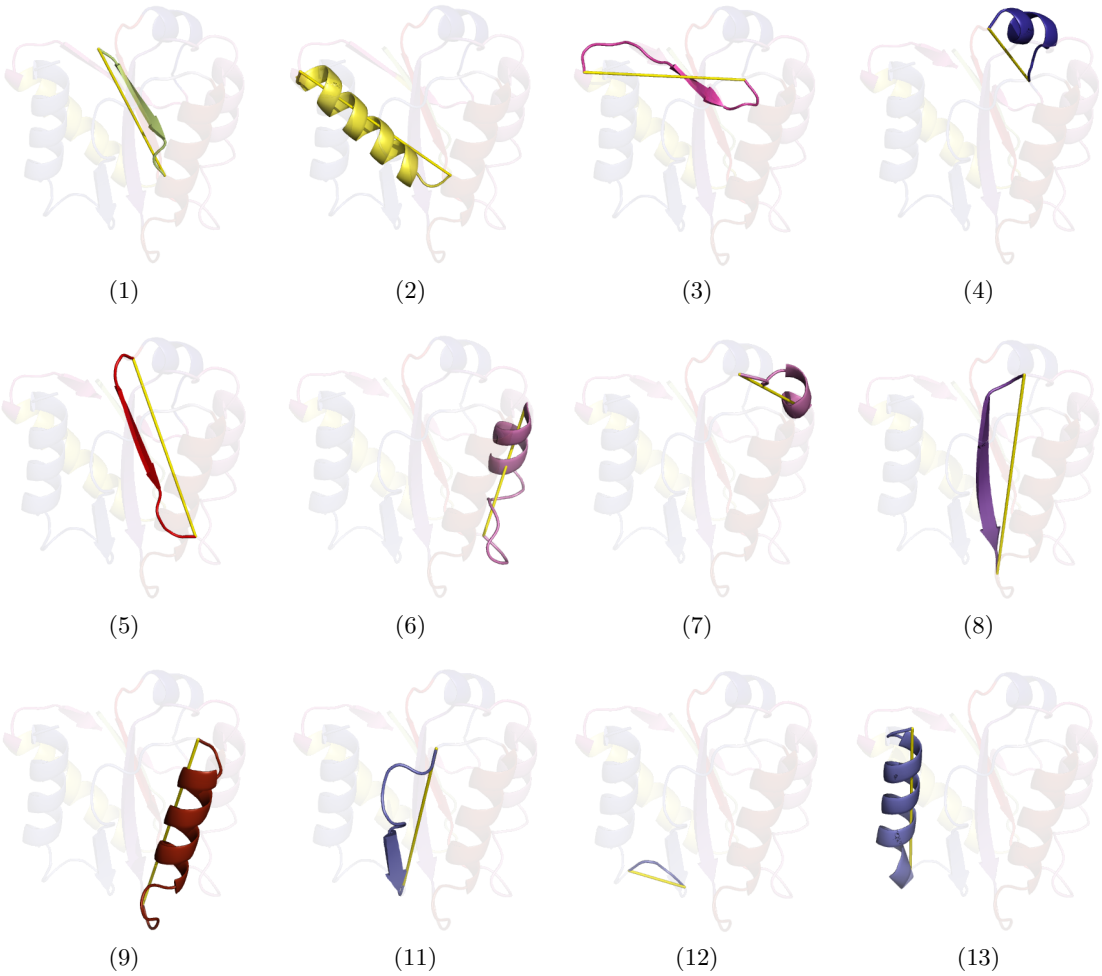


Figure F.2: Linear Bézier abstractions of 5NLL (segment 10 is not shown above as it contains only two residues).

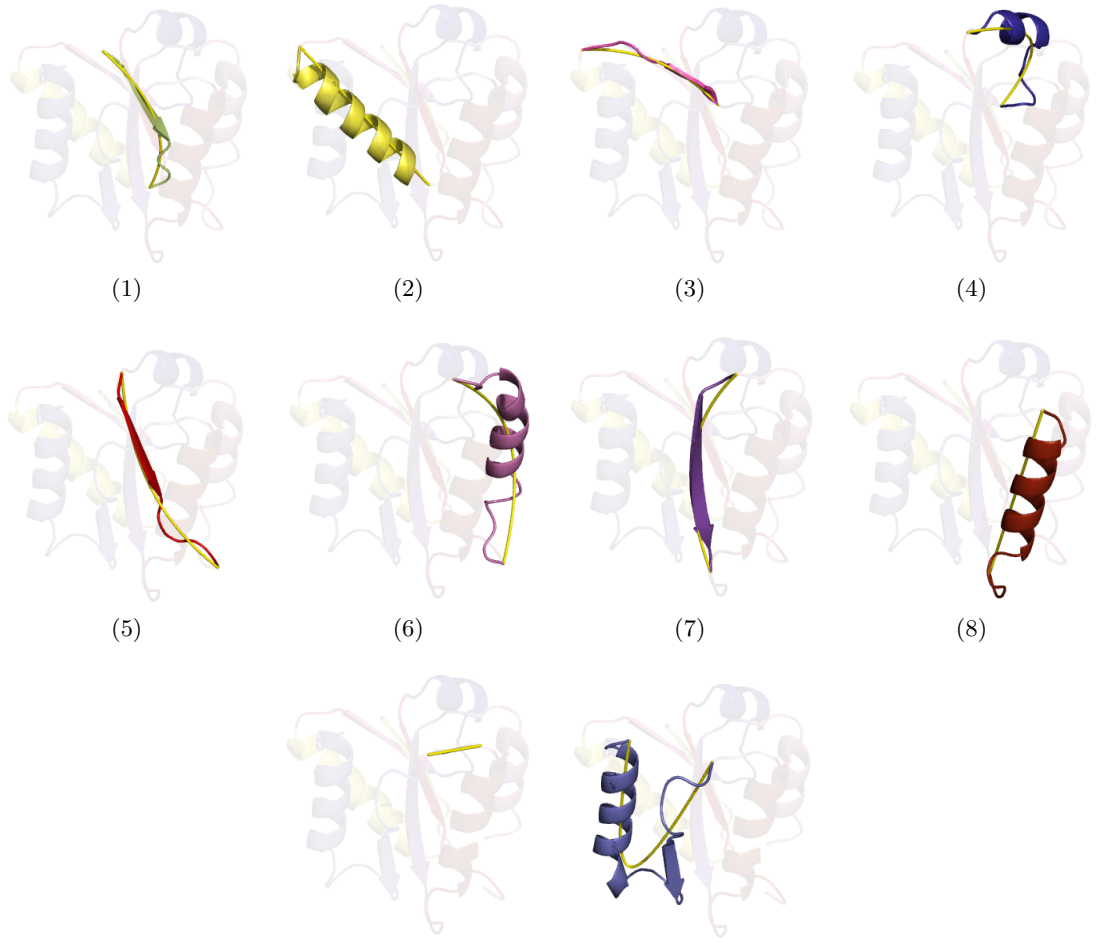
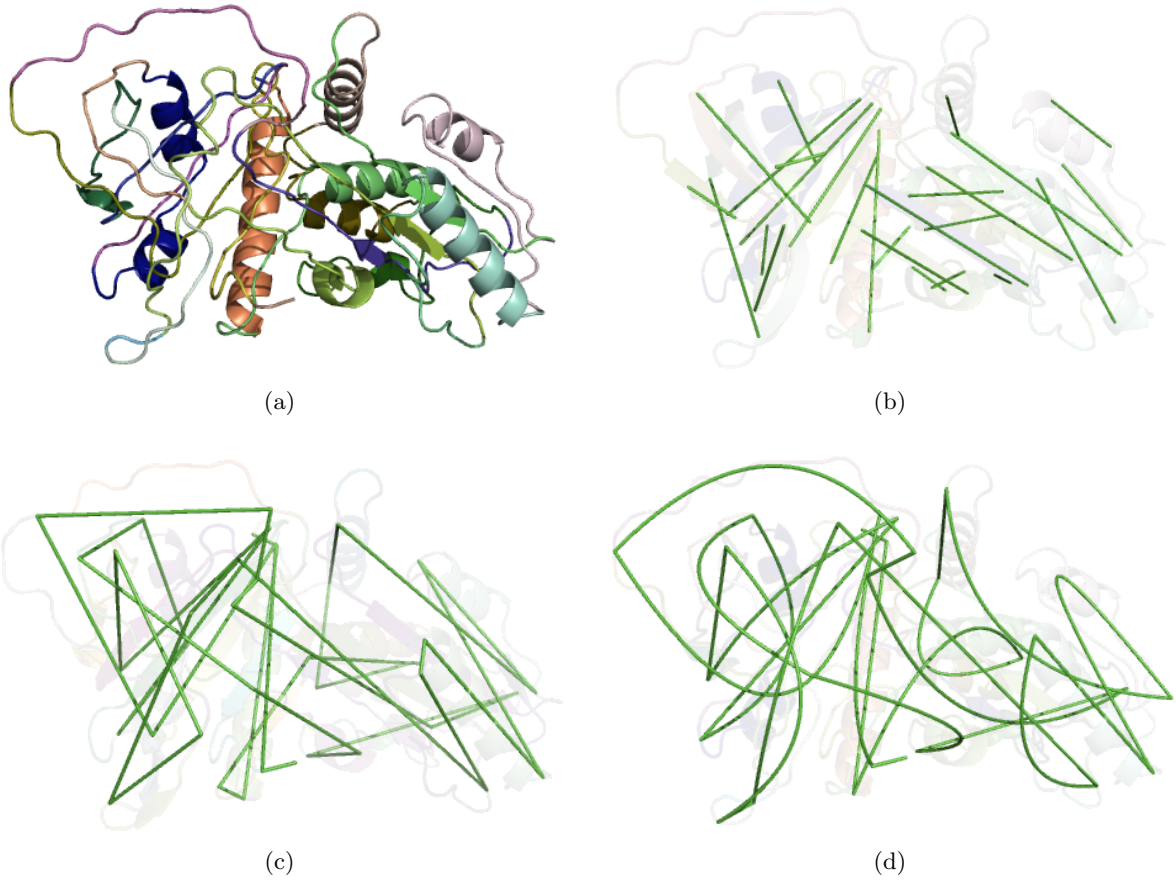


Figure F.3: Non-linear Bézier abstractions of 5NLL. Note how the curved sections of the protein are losslessly represented by the corresponding non-linear curves.



(a) (b)

(c) (d)

Figure F.4: (a) Structure of α1-Antitrypsin (wwPDB code 1QLP) shown in a cartoon representation with standard secondary structures – helices and strands of sheet assigned using DSSP Kabsch and Sander (1983). (b) Traditional representation of the folding pattern which replaces each secondary structural element with line segments. (c) The representations produced by our approach by constraining the Bézier curves to a maximum degree of 1. (d) The representations produced by constraining the Bézier curves to a maximum degree of 3 – Linear through to Cubic

F.2.2 Dissection of regions of α1-Antitrypsin: A canonical template for active Serpins

Another example is presented here to illustrate the nature of the dissections generated using the proposed approach, using (randomly chosen) α1-Antitrypsin molecule (Elliott et al., 2000) with wwPDB code 1QLP. Only the coordinates of protein residues of chain A (in the range 23 – 394) of this molecule are established. As such, the number of residues considered are 372. We have shown the cartoon diagram of the secondary structure assigned using DSSP (see Figure F.4a). The approximation of helices and strands by straight lines results in a representation as shown in Figure F.4(b). Also shown are the linear and non-linear Bézier segmentations generated by out method. (Figures F.4(c) and (d) respectively).

The approximation of secondary structural elements by straight lines (Figure F.4b) represents only 219 of the 372 residues accounting to about 59% of the entire residue range. Such a representation is lossy in the sense that nearly 41% of the protein geometry goes unrepresented. These regions are disordered and hence, the secondary structure assignment methods cannot identify clear structural patterns. The large proportion of undesignated region is a cause for concern because it shows the limitation of representation schemes which rely on secondary structure assignment.

Our Bézier segmentation approach is immune to this problem and completely represent the protein structure. The linear Bézier abstraction results in 39 segments while the non-linear Bézier abstraction results in 26 segments. The resultant non-linear segmentation is listed in Table F.4. Some of the distinctively curved regions of the protein and their corresponding non-linear representations are depicted in Figure F.5.

Table F.4: The *non-linear* Bézier segmentation produced by our method for α1-Antitrypsin (1QLP).

Segment index	Residue stretch	Segment index	Residue stretch
1	23 – 24	14	225 – 236
2	24 – 45	15	236 – 246
3	45 – 69	16	246 – 257
4	69 – 88	17	257 – 279
5	88 – 108	18	279 – 289
6	108 – 123	19	289 – 305
7	123 – 149	20	305 – 324
8	149 – 166	21	324 – 343
9	166 – 179	22	343 – 346
10	179 – 195	23	346 – 359
11	195 – 211	24	359 – 378
12	211 – 213	25	378 – 393
13	213 – 225	26	393 – 394

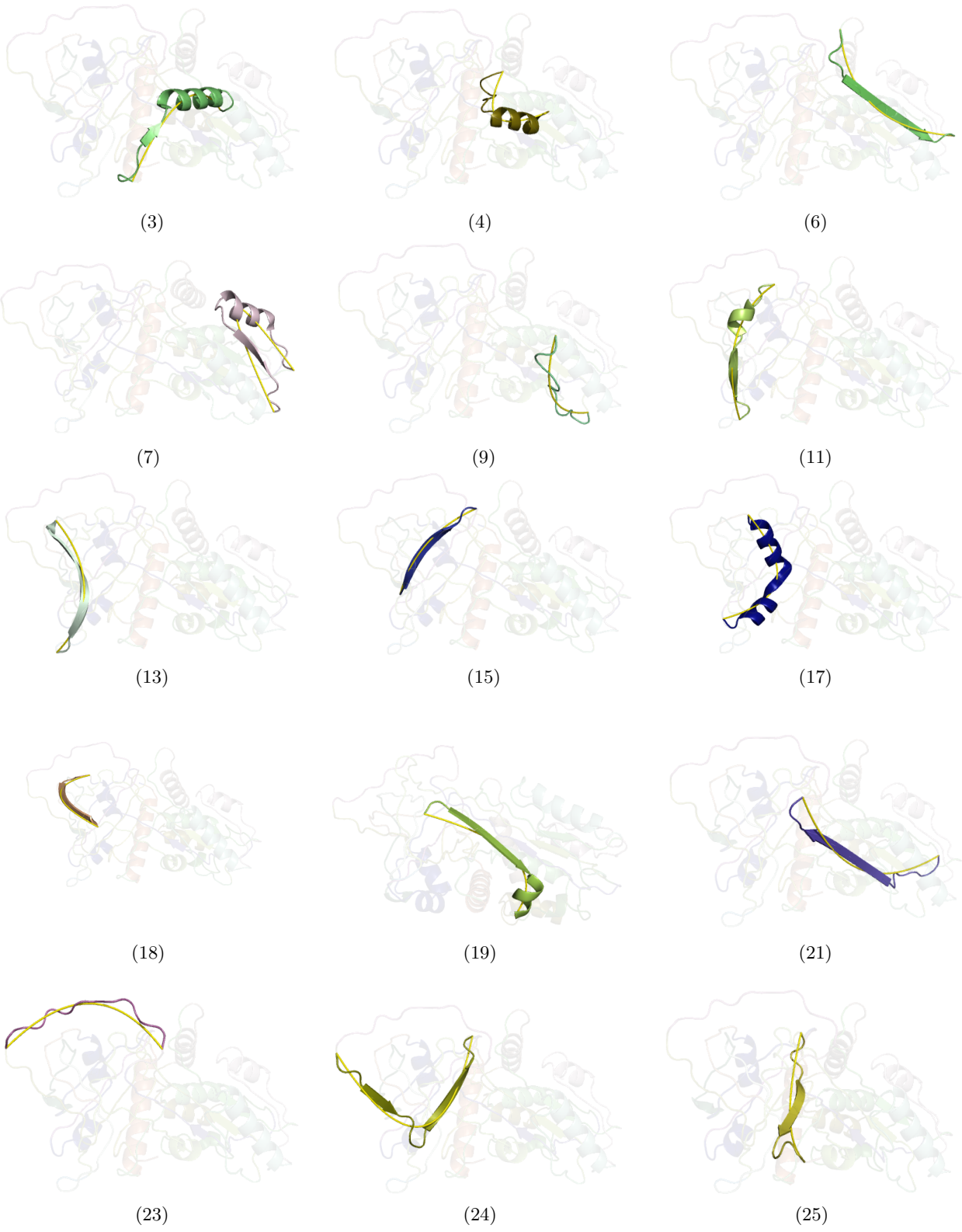


Figure F.5: A selection of some of the non-linear Bézier segments of 1QLP. The segments numbered above correspond to the ones listed in Table F.4.