Mach Learn (2015) 100:333–378
DOI 10.1007/s10994-015-5493-0

Minimum message length estimation of mixtures of multivariate Gaussian and von Mises-Fisher distributions

Parthan Kasarapu[1] · **Lloyd Allison**[1]

Received: 27 July 2014 / Accepted: 13 March 2015 / Published online: 28 March 2015

Abstract Mixture modelling involves explaining some observed evidence using a combination of probability distributions. The crux of the problem is the inference of an optimal number of mixture components and their corresponding parameters. This paper discusses unsupervised learning of mixture models using the Bayesian Minimum Message Length (MML) criterion. To demonstrate the effectiveness of search and inference of mixture parameters using the proposed approach, we select two key probability distributions, each handling fundamentally different types of data: the multivariate Gaussian distribution to address mixture modelling of data distributed in Euclidean space, and the multivariate von Mises-Fisher (vMF) distribution to address mixture modelling of directional data distributed on a unit hypersphere. The key contributions of this paper, in addition to the general search and inference methodology, include the derivation of MML expressions for encoding the data using multivariate Gaussian and von Mises-Fisher distributions, and the analytical derivation of the MML estimates of the parameters of the two distributions. Our approach is tested on simulated and real world data sets. For instance, we infer vMF mixtures that concisely explain experimentally determined three-dimensional protein conformations, providing an effective *null model* description of protein structures that is central to many inference problems in structural bioinformatics. The experimental results demonstrate that the performance of our proposed search and inference method along with the encoding schemes improve on the state of the art mixture modelling techniques.

Keywords Mixture modelling · Minimum message length · Multivariate Gaussian · von Mises-Fisher · Protein structure

Editors: João Gama, Indrė Žliobaitė, Alípio M. Jorge, and Concha Bielza.

✉ Parthan Kasarapu
parthan.kasarapu@monash.edu

Lloyd Allison
lloyd.allison@monash.edu

[1] Faculty of Information Technology, Monash University, Melbourne, VIC 3800, Australia

1 Introduction

Mixture models are common tools in statistical pattern recognition (McLachlan and Basford 1988). They offer a mathematical basis to explain data in 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 variously correspond to a subtype, kind, species, or subpopulation 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 used in machine learning for tasks such as classification and unsupervised learning (Titterington et al. 1985; McLachlan and Peel 2000).

Formally, mixture modelling involves representing a distribution of data as a weighted sum of individual probability distributions. Specifically, the problem we consider here is to model the observed data using a mixture $\mathscr{M}$ of the form: $\Pr(\mathbf{x}; \mathscr{M}) = \sum_{j=1}^{M} w_j f_j(\mathbf{x}; \Theta_j)$, where $\mathbf{x}$ is a d-dimensional datum, M is the number of mixture components, w_j and $f_j(\mathbf{x}; \Theta_j)$ are the weight and probability density of the jth component respectively; the weights are positive and sum to one. The problem of modelling some observed data using a mixture distribution involves determining the number of components M, and estimating the mixture parameters. Inferring the optimal number of mixture components involves the difficult problem of balancing the trade-off between two conflicting objectives: low *hypothesis complexity* as determined by the number of components *and* their respective parameters, versus good quality of *fit* to the observed data. Generally, a hypothesis with more free parameters can fit observed data better than a hypothesis with fewer parameters. A number of strategies have been used to control this balance (see Sect. 6). These methods provide varied formulations to assess the mixtures and their ability to explain the data. Methods using the minimum message length criterion, a Bayesian method of inductive inference, 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).

Although much of the literature concerns the theory and application of Gaussian mixtures, mixture modelling using other probability distributions has been widely used. Some examples are: Poisson (Wang et al. 1996), exponential (Seidel et al. 2000), Laplace (Jones and McLachlan 1990), t-distribution (Peel and McLachlan 2000), Weibull (Patra and Dey 1999), Kent (Peel et al. 2001), von Mises-Fisher (Banerjee et al. 2005). The use of Gaussian mixtures in several research disciplines has been partly motivated by its computational tractability (McLachlan and Peel 2000). For data where the *direction* of the constituent vectors is important, Gaussian mixtures are inappropriate and distributions such as the von Mises-Fisher may be used (Mardia et al. 1979; Banerjee et al. 2005). In any case, for any kind of component distribution, one needs to estimate the mixture parameters, and provide a sound justification for selecting the appropriate number of components. Software for mixture modelling relies on the following elements: (1) an *estimator* of the parameters of each component of a mixture, (2) an *objective function*, that can be used to score as well as compare two mixtures and decide which is better, and (3) a *search strategy* for the best number of components and their weights.

Traditionally, parameter estimation is done using maximum likelihood (ML) or maximum *a posteriori* probability (MAP) based estimation. In this work, we use the Bayesian minimum message length (MML) principle. Unlike MAP, MML estimators are invariant under non-linear transformations of the data (Oliver and Baxter 1994), and unlike ML, It has been used in the inference of several probability distributions (Wallace 2005). MML-based inference operates by considering the problem as encoding first the parameters and then the data given those parameters. The values that result in the least *overall* message length to explain the whole data are taken as the MML estimates for an inference problem. The MML scheme thus incorporates the cost of stating parameters into model selection. It is self evident that a continuous parameter value can only be stated to some finite precision; the cost of encoding a parameter is determined by its prior and the precision. ML estimation ignores the cost of stating a parameter and MAP based estimation uses the probability *density* of a parameter instead of its probability measure. In contrast, the MML inference process calculates the optimal precision to which parameters should be stated and a corresponding probability value is then computed. This is used to calculate the message length corresponding to those estimates. Thus, models with varying parameters are evaluated based on their resultant total message lengths. We use this characteristic property of MML to evaluate mixtures with different numbers of components.

Although there have been several attempts to address the challenges of mixture modelling, the existing methods have some limitations in their formalism (see Sect. 6). In particular, some methods based on MML are incomplete in their formulation. We aim to rectify these drawbacks by proposing a comprehensive MML formulation and develop a search heuristic that selects the number of mixture components based on the proposed formulation. To demonstrate the effectiveness of the proposed search and parameter estimation, we first consider modelling problems using Gaussian mixtures and include relevant discussion on mixture modelling of directional data using von Mises-Fisher distributions.

The conventional method of estimating the parameters of a mixture, for a *given* number of components, relies on the Expectation–Maximization (EM) algorithm (Dempster et al. 1977). Previous attempts to infer Gaussian mixtures based on the MML framework have been undertaken using simplifying assumptions, such as the covariance matrices being diagonal (Oliver et al. 1996), or coarsely approximating the probabilities of mixture parameters (Roberts et al. 1998; Figueiredo and Jain 2002). Further, the search heuristic adopted in some of these methods is to run the EM several times for different numbers of components, M, and select the M with the best EM outcome (Oliver et al. 1996; Roberts et al. 1998; Biernacki et al. 2000). A search method based on iteratively *deleting* components has been proposed by Figueiredo and Jain (2002). It begins by assuming a very large number of components and selectively eliminates components deemed redundant; there is no provision for recovering from deleting a component in error.

In this work, we propose a search method which selectively *splits, deletes*, or *merges* components depending on improvement to the MML objective function. The operations, combined with 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. Unlike the method of Figueiredo and Jain (2002), our method starts with a one-component mixture and alters the number of components in subsequent iterations. This avoids the overhead of dealing with a large number of components unless required.

The proposed search heuristic can be used with probability distributions for which the MML expressions to calculate message lengths for estimates and for data given those estimates are known. As an instance of this, Sect. 7 discusses mixture modelling of directional data. The statistical properties of directional data have been studied using several types of distributions (Fisher 1953; Watson and Williams 1956; Fisher 1993; Mardia and Jupp 2000), often described on surfaces of compact manifolds, such as the sphere, ellipsoid, torus *etc*. The most fundamental of these is the von Mises-Fisher (vMF) distribution which is analogous to a symmetric multivariate Gaussian distribution, wrapped around a unit hypersphere (Watson and Williams 1956). The estimation of the parameters of the vMF distribution is often done using maximum likelihood. However, the complex nature of the mathematical form presents difficulty in estimating the concentration parameter κ. This has lead to researchers using many different approximations, as discussed in Sect. 2.2. 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. We demonstrate this by the experiments conducted on a range of sample sizes. The problem is particularly evident when the dimensionality of the data is large, also affecting the application in which it is used, such as mixture modelling. We aim to rectify this issue by using MML estimates for κ. Our experiments section demonstrates that the MML estimate of κ provides a more reliable answer and is an improvement on the current state of the art. These MML estimates are subsequently used in mixture modelling of vMF distributions (see Sects. 7 and 10). Previous studies have established the importance of vMF mixture models with demonstrated applications to clustering of protein dihedral angles (Mardia et al. 2007; Dowe et al. 1996a), large-scale text clustering (Banerjee et al. 2003), and gene expression analyses (Banerjee et al. 2005). The merit of using cosine based similarity metrics, which are closely related to the vMF, for clustering high dimensional text data has been investigated in Strehl et al. (2000). For text clustering, there is evidence that vMF mixture models have a superior performance compared to other statistical distributions (Salton and McGill 1986; Salton and Buckley 1988; Zhong and Ghosh 2003; Banerjee et al. 2005).

Contributions: The main contributions of this paper are as follows:

– We derive the analytical estimates of the parameters of a multivariate Gaussian distribution with full covariance matrix, using the MML principle (Wallace and Freeman 1987).
– We derive the expression to infer the concentration parameter κ of a generic d-dimensional vMF distribution using MML-based estimation. We demonstrate, through a series of experiments, that this estimate outperforms the previous ones, therefore making it a reliable candidate to be used in mixture modelling.
– A generalized MML-based search heuristic is proposed to infer the optimal number of mixture components that would best explain the observed data. The search implements a generic approach to mixture modelling and allows, in this instance, the use of d-dimensional Gaussian and vMF distributions under the MML framework. We compare it with the widely cited work of Figueiredo and Jain (2002) and demonstrate its effectiveness.

The rest of the paper is organized as follows: Sect. 2 describes the commonly used estimators of Gaussian and vMF distributions. Section 3 introduces the MML inference framework. Section 4 outlines the derivation of the MML parameter estimates of multivariate Gaussian and vMF distributions. Section 5 describes the formulation of a mixture model using MML and the estimation of the mixture parameters under the framework. Section 6 reviews the existing methods for selecting the mixture components. Section 7 describes our proposed approach to determine the number of mixture components. Section 8 depicts the competitive performance of the proposed MML-based search through experiments conducted with Gaussian mixtures. Section 9 presents the results for MML-based vMF parameter estimation followed by results supporting the application of vMF mixtures to text clustering and protein structural data in Sect. 10. An extended version of the paper is available at http://arxiv.org/abs/1502.07813.

2 Existing methods of estimating the parameters of Gaussian and von Mises-Fisher distributions

2.1 Gaussian parameter estimates

The probability density of a d-variate Gaussian distribution is given by Eq. 1 where $\boldsymbol{\mu}$, $\mathbf{C}$ are the respective mean, covariance matrix of the distribution, and $|\mathbf{C}|$ is the determinant of the covariance matrix. Given data $D = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$, where $\mathbf{x}_i \in \mathbb{R}^d$, the log-likelihood $\mathscr{L}$ is given by Eq. 2. To compute the traditional maximum likelihood estimates, $\mathscr{L}$ needs to be *maximized*. This is achieved by computing the gradient of the log-likelihood function with respect to the parameters and solving the resultant equations.

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

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

The *gradient vector* of $\mathscr{L}$ with respect to $\boldsymbol{\mu}$ and the *gradient matrix* of $\mathscr{L}$ with respect to $\mathbf{C}$ are given below. The maximum likelihood estimates are obtained by solving $\nabla_{\boldsymbol{\mu}}\mathscr{L} = 0$ and $\nabla_{\mathbf{C}}\mathscr{L} = 0$.

$$\nabla_{\boldsymbol{\mu}}\mathscr{L} = \frac{\partial \mathscr{L}}{\partial \boldsymbol{\mu}} = \sum_{i=1}^{N} \mathbf{C}^{-1}(\mathbf{x}_i - \boldsymbol{\mu}) \quad \text{and}$$

$$\nabla_{\mathbf{C}}\mathscr{L} = \frac{\partial \mathscr{L}}{\partial \mathbf{C}} = -\frac{N}{2}\mathbf{C}^{-1} + \frac{1}{2}\sum_{i=1}^{N}\mathbf{C}^{-1}(\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^T\mathbf{C}^{-1} \tag{3}$$

The estimates are given as $\hat{\boldsymbol{\mu}} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_i$ and $\hat{\mathbf{C}}_{\text{ML}} = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{x}_i - \hat{\boldsymbol{\mu}})(\mathbf{x}_i - \hat{\boldsymbol{\mu}})^T$. $\hat{\mathbf{C}}_{\text{ML}}$ is known to be a biased estimate of the covariance matrix (Barton 1961; Basu 1964; Eaton and Morris 1970; White 1982) and issues related with its use in mixture modelling have been documented in Gray (1994) and Lo (2011). An unbiased estimator of $\mathbf{C}$ was proposed by Barton (1961) and is given as $\hat{\mathbf{C}}_{\text{unbiased}} = \frac{1}{N-1}\sum_{i=1}^{N}(\mathbf{x}_i - \hat{\boldsymbol{\mu}})(\mathbf{x}_i - \hat{\boldsymbol{\mu}})^T$. 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.

2.2 Parameter estimates of a von Mises-Fisher distribution

The probability density function of a vMF distribution with parameters $\Theta = (\boldsymbol{\mu}, \kappa) \equiv$ (mean direction, concentration parameter) for a random unit vector $\mathbf{x} \in \mathbb{R}^d$ on a $(d-1)$- dimensional hypersphere $\mathbb{S}^{d-1}$ is given by Eq. 4, where $C_d(\kappa)$ is the normalization constant and I_v is a modified Bessel function of the first kind and order v. Given data $D = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$, such that $\mathbf{x}_i \in \mathbb{S}^{d-1}$, the log-likelihood $\mathscr{L}$ is given by Eq. 5.

$$f(\mathbf{x};\boldsymbol{\mu},\kappa) = C_d(\kappa)e^{\kappa\boldsymbol{\mu}^T\mathbf{x}} \quad \text{such that} \quad C_d(\kappa) = \frac{\kappa^{d/2-1}}{(2\pi)^{d/2}I_{d/2-1}(\kappa)} \tag{4}$$

$$\mathscr{L}(D|\boldsymbol{\mu},\kappa) = N\log C_d(\kappa) + \kappa\boldsymbol{\mu}^T\mathbf{R} \quad \text{where} \quad \mathbf{R} = \sum_{i=1}^{N}\mathbf{x}_i \tag{5}$$

Let R denote the magnitude of the resultant vector $\mathbf{R}$. Let $\hat{\boldsymbol{\mu}}$ and $\hat{\kappa}$ be the maximum likelihood estimators of μ and κ respectively. Under the constraint that $\hat{\boldsymbol{\mu}}$ is a unit vector, these estimates are obtained by maximizing $\mathscr{L}$ as follows:

$$\hat{\boldsymbol{\mu}} = \frac{\mathbf{R}}{R}, \quad \hat{\kappa} = A_d^{-1}(\bar{R}) \quad \text{where} \quad A_d(\kappa) = -\frac{C_d'(\kappa)}{C_d(\kappa)} = \frac{I_{d/2}(\kappa)}{I_{d/2-1}(\kappa)} \tag{6}$$

Solving the non-linear equation: $F(\hat{\kappa}) \equiv A_d(\hat{\kappa}) - \bar{R} = 0$ yields the corresponding maximum likelihood estimate of the concentration parameter κ. As it is difficult to analytically solve Eq. (6), there have been several approaches proposed to approximate $\hat{\kappa}$. Each of them 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). These are briefly summarized below.

Banerjee's approximation: Banerjee et al. (2005) provide an easy to use expression for $\hat{\kappa}$. The formula is very appealing as it eliminates the need to evaluate complex Bessel functions. Banerjee et al. (2005) demonstrated that their approximation κ_B (see Eq. 7) 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 in evaluating the root of Eq. 6.

$$\kappa_B = \frac{\bar{R}(d-\bar{R}^2)}{1-\bar{R}^2} \tag{7}$$

Tanabe's approximation: Tanabe et al. (2007) utilize the properties of Bessel functions to determine the lower and upper bounds for $\hat{\kappa}$. A fixed point iteration function defined by $\phi_{2d}(\kappa) = \bar{R}\kappa A_d(\kappa)^{-1}$ in conjunction with linear interpolation is then used to approximate $\hat{\kappa}$. The approximation κ_T along with the bounds κ_l and κ_u are given below:

$$\kappa_T = \frac{\kappa_l\phi_{2d}(\kappa_u) - \kappa_u\phi_{2d}(\kappa_l)}{(\phi_{2d}(\kappa_u) - \phi_{2d}(\kappa_l)) - (\kappa_u - \kappa_l)} \quad \text{where} \quad \kappa_l = \frac{\bar{R}(d-2)}{1-\bar{R}^2} \le \hat{\kappa} \le \kappa_u = \frac{\bar{R}d}{1-\bar{R}^2}$$

Sra's Truncated Newton approximation: A heuristic approximation proposed by Sra (2012) involves refining the approximation of Banerjee et al. (2005) (Eq. 7). Sra's approximation κ_N is obtained by performing two iterations of Newton's method as follows:

$$\kappa_1 = \kappa_B - \frac{F(\kappa_B)}{F'(\kappa_B)} \quad \text{and} \quad \kappa_N = \kappa_1 - \frac{F(\kappa_1)}{F'(\kappa_1)}$$

$$\text{where} \quad F'(\kappa) = A_d'(\kappa) = 1 - A_d(\kappa)^2 - \frac{(d-1)}{\kappa}A_d(\kappa) \tag{8}$$

Song's Truncated Halley approximation: The approximation due to Song et al. (2012) uses Halley's method which is the second order expansion of Taylor's series of a function $F(\kappa)$. The higher order approximation gives a more accurate estimate as demonstrated by Song et al. (2012). The iterative Halley's method is truncated after two steps of the root finding algorithm (similar to that done by Sra 2012). The following two iterations result in their approximation κ_H:

$$\kappa_1 = \kappa_B - \frac{2F(\kappa_B)F'(\kappa_B)}{2F'(\kappa_B)^2 - F(\kappa_B)F''(\kappa_B)} \quad \text{and}$$
$$\kappa_H = \kappa_1 - \frac{2F(\kappa_1)F'(\kappa_1)}{2F'(\kappa_1)^2 - F(\kappa_1)F''(\kappa_1)}$$
$$\text{where} \quad F''(\kappa) = A_d''(\kappa) = 2A_d(\kappa)^3 + \frac{3(d-1)}{\kappa}A_d(\kappa)^2 + \frac{(d^2-d-2\kappa^2)}{\kappa^2}A_d(\kappa) - \frac{(d-1)}{\kappa} \tag{9}$$

The common theme in all these methods is that they try to approximate the maximum likelihood estimate governed by Eq. (6). It is to be noted that the maximum likelihood estimators of κ have considerable bias (Schou 1978; Best and Fisher 1981; Cordeiro and Vasconcellos 1999). To counter this effect, we explore the MML-based estimation procedure. This Bayesian method of estimation not only results in an unbiased estimate but also provides a framework to choose from several competing models (Wallace and Freeman 1987). Dowe et al. (1996c) have demonstrated the superior performance of the MML estimate for a three-dimensional vMF distribution. We extend their work to derive the MML estimators for a generic d-dimensional vMF distribution and compare its performance with the existing methods.

3 Minimum message length (MML) inference

Model selection using MML: Wallace and Boulton (1968) developed the first practical criterion for model selection based on information theory. As per Bayes's theorem, $\Pr(H\&D) = \Pr(H) \times \Pr(D|H) = \Pr(D) \times \Pr(H|D)$ where D denotes observed data, and H some hypothesis about that data. Further, $\Pr(H\&D)$ is the joint probability, $\Pr(H)$ and $\Pr(D)$ are the prior probabilities of hypothesis H and data D respectively, $\Pr(H|D)$ is the posterior probability, and $\Pr(D|H)$ is the likelihood. As per Shannon (1948), 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, Wallace and Boulton (1968) got the relationship for $I(H\&D) = I(H) + I(D|H) = I(D) + I(H|D)$. As a result, given two competing hypotheses H and H', the difference in message lengths ΔI gives the posterior log-odds ratio between the two.

$$\Delta I = I(H\&D) - I(H'\&D) = I(H|D) - I(H'|D) \text{ bits.} \implies \Pr(H'|D) = 2^{\Delta I}\Pr(H|D) \tag{10}$$

$I(H\&D)$ can be interpreted as the *total* cost to encode a message comprising of the following two parts: (1) the hypothesis H, which takes $I(H)$ bits, and (2) the observed data D using knowledge of H, which takes $I(D|H)$ bits. The framework provides a rigorous means to objectively compare two competing hypotheses. A more complex H may explain D better but takes more bits to be stated itself. 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 $I(H)$ and the goodness of fit $I(D|H)$.

MML based parameter estimation: Wallace and Freeman (1987) introduced a generalized framework to estimate a set of parameters Θ given data D. The method requires a reasonable prior $h(\Theta)$ on the hypothesis and evaluating the *determinant* of the Fisher information matrix $|\mathcal{F}(\Theta)|$ of the *expected* second-order partial derivatives of the negative log-likelihood function, $\mathscr{L}(D|\Theta)$. The parameter vector Θ that minimizes the message length expression (given by Eq. 11) is the MML estimate, where p is the number of free parameters in the model, and q_p is the p-dimensional lattice quantization constant (Conway and Sloane 1984). The total message length $I(\Theta, D)$, therefore, comprises of two parts: (1) the cost of encoding the parameters, $I(\Theta)$, and (2) the cost of encoding the data given the parameters, $I(D|\Theta)$.

$$I(\Theta, D) = \underbrace{\frac{p}{2}\log q_p - \log\left(\frac{h(\Theta)}{\sqrt{|\mathscr{F}(\Theta)|}}\right)}_{\text{I()}} + \underbrace{\mathscr{L}(D|\Theta) + \frac{p}{2}}_{\text{I(D|)}} \tag{11}$$

In ML estimation, the encoding cost of parameters is, in effect, considered constant, and minimizing the message length corresponds to minimizing the negative log-likelihood of the data (the second part). In MAP based estimation, a probability *density* rather than the probability is used. It is self evident that continuous parameter values can only be stated to some finite precision; MML incorporates this in the framework by determining the volume of the uncertainty region in which the parameter is centred as $V = \dfrac{q_p^{-p/2}}{\sqrt{|\mathscr{F}(\Theta)|}}$. This product of V and the density $h(\Theta)$ gives the *probability* of a particular Θ which is used to compute the message length associated with encoding the continuous valued parameters.

4 Derivation of the MML parameter estimates of Gaussian and von Mises-Fisher distributions

Based on the MML inference process discussed in Sect. 3, we now proceed to formulate the message length expressions and derive the parameter estimates of Gaussian and von Mises-Fisher distributions.

4.1 MML-based parameter estimation of a multivariate Gaussian distribution

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 probability 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 $h(\boldsymbol{\mu}, \mathbf{C}) \propto |\mathbf{C}|^{-\frac{d+1}{2}}$.

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

On differentiating the gradient vector in Eq. 3 with respect to $\boldsymbol{\mu}$, we get $\nabla^2_{\boldsymbol{\mu}}\mathscr{L} = -N\,\mathbf{C}^{-1}$. Consequently, $|\mathscr{F}(\boldsymbol{\mu})| = N^d|\mathbf{C}|^{-1}$. To compute $|\mathscr{F}(\mathbf{C})|$, 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 ith row and jth 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 $\boldsymbol{\mu}, \mathbf{C}$, substitute the expressions for $h(\boldsymbol{\mu}, \mathbf{C})$, $|\mathcal{F}(\boldsymbol{\mu}, \mathbf{C})|$, and the negative log-likelihood (Eq. 2) in Eq. 11 with the number of free parameters as $p = d(d+3)/2$. Hence, the message length is $I(\boldsymbol{\mu}, \mathbf{C}, D) = \dfrac{(N-1)}{2} \log |\mathbf{C}| + \dfrac{1}{2} \displaystyle\sum_{i=1}^{N} (\mathbf{x}_i - \boldsymbol{\mu})^T \mathbf{C}^{-1} (\mathbf{x}_i - \boldsymbol{\mu}) + \text{constant}.$
To obtain the MML estimates of $\boldsymbol{\mu}$ and $\mathbf{C}$, $I(\boldsymbol{\mu}, \mathbf{C}, D)$ 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 $I(\boldsymbol{\mu}, \mathbf{C}, D)$ with respect to $\mathbf{C}$ and solving $\nabla_{\mathbf{C}} I = 0$ gives the corresponding MML estimate.

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

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

We observe that the MML estimate $\hat{\mathbf{C}}_{\text{MML}}$ is equivalent to the *unbiased* estimate of the covariance matrix $\mathbf{C}$, thus, lending credibility for its preference over the traditional maximum likelihood estimate.

4.2 MML-based parameter estimation of a von Mises-Fisher distribution

The MML parameter estimates of two dimensional vMF distributions have been derived previously (Wallace and Dowe 1994; Dowe et al. 1996b). The MML estimation of three-dimensional vMF was studied by Dowe et al. (1996c), where they demonstrate that the MML-based inference is more reliable compared to the traditional ML and MAP based estimation methods. We use the Wallace and Freeman (1987) method to formulate the objective function (Eq. 11) and derive the MML parameter estimates corresponding to a generic vMF distribution.
Prior probability of the parameters: Regarding the choice of a reasonable prior for the parameters $\Theta = (\boldsymbol{\mu}, \kappa)$ of a vMF distribution, Wallace and Dowe (1994) and Dowe et al. (1996c) suggest using the following "colourless" prior so that is uniform in direction and normalizable on transforming into Cartesian coordinates in κ: $h(\boldsymbol{\mu}, \kappa) \propto \dfrac{\kappa^{d-1}}{(1+\kappa^2)^{\frac{d+1}{2}}}$.
Fisher information of the parameters: Dowe et al. (1996c) argue that in the general d-dimensional case, the Fisher information $|\mathcal{F}(\boldsymbol{\mu}, \kappa)| = (N\kappa A_d(\kappa))^{d-1} \times N A_d'(\kappa)$, where $A_d(\kappa)$ and $A_d'(\kappa)$ are given by Eqs. 6 and 8 respectively.
Message length formulation: Substituting the expression for $h(\boldsymbol{\mu}, \kappa)$, $|\mathcal{F}(\boldsymbol{\mu}, \kappa)|$, and the negative log-likelihood (Eq. 5) in Eq. 11 with the number of free parameters $p = d$, we get the net message length expression as

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

To obtain the MML estimates of $\boldsymbol{\mu}$ and κ, $I(\boldsymbol{\mu}, \kappa, D)$ needs to be minimized. The estimate for $\boldsymbol{\mu}$ is same as the maximum likelihood estimate (Eq. 6). The resultant equation in κ that needs to be minimized is then given by $I(\kappa)$. To obtain the MML estimate of κ, we need to differentiate Eq. 13 and set it to zero.

$$I(\kappa) = \frac{(d-1)}{2}\log\frac{A_d(\kappa)}{\kappa} + \frac{1}{2}\log A_d'(\kappa) + \frac{(d+1)}{2}\log(1+\kappa^2) - N\log C_d(\kappa) \\ -\kappa R + \text{constant} \tag{12}$$

$$\text{Let} \quad G(\kappa) \equiv \frac{\partial I}{\partial \kappa} = -\frac{(d-1)}{2\kappa} + \frac{(d+1)\kappa}{1+\kappa^2} + \frac{(d-1)}{2}\frac{A_d'(\kappa)}{A_d(\kappa)} + \frac{1}{2}\frac{A_d''(\kappa)}{A_d'(\kappa)} + NA_d(\kappa) - R \tag{13}$$

The non-linear equation: $G(\kappa) = 0$ does not have a closed form solution. We try both the Newton and Halley's method to find an approximate solution. We discuss both variants and comment on the effects of the two approximations in the experimental results. To be fair and consistent with Sra (2012) and Song et al. (2012), we use the initial guess of the root as κ_B (Eq. 7) and iterate twice to obtain the MML estimate.

1. *Approximation using Newton's method:*

$$\kappa_1 = \kappa_B - \frac{G(\kappa_B)}{G'(\kappa_B)} \quad \text{and} \quad \kappa_{\text{MN}} = \kappa_1 - \frac{G(\kappa_1)}{G'(\kappa_1)} \tag{14}$$

2. *Approximation using Halley's method:*

$$\kappa_1 = \kappa_B - \frac{2G(\kappa_B)G'(\kappa_B)}{2G'(\kappa_B)^2 - G(\kappa_B)G''(\kappa_B)} \quad \text{and} \quad \kappa_{\text{MH}} = \kappa_1 - \frac{2G(\kappa_1)G'(\kappa_1)}{2G'(\kappa_1)^2 - G(\kappa_1)G''(\kappa_1)} \tag{15}$$

Equations (14) and (15) give the MML estimates κ_{MN} and κ_{MH} using Newton's and Halley's methods respectively. The details of evaluating $G'(\kappa)$ and $G''(\kappa)$ are discussed in the "Appendix".

5 Minimum message length approach to mixture modelling

Mixture modelling involves representing some observed data as a weighted sum of component density functions. For some observed data $D = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$, the log-likelihood using the mixture distribution is $\mathscr{L}(D|\boldsymbol{\Phi}) = \sum_{i=1}^{N}\log\sum_{j=1}^{M} w_j f_j(\mathbf{x}_i; \Theta_j)$, where $\boldsymbol{\Phi} = \{w_1, \ldots, w_M, \Theta_1, \ldots, \Theta_M\}$, w_j and $f_j(\mathbf{x}; \Theta_j)$ are the weight and probability density of the jth component respectively. For a fixed M, the mixture parameters $\boldsymbol{\Phi}$ are traditionally estimated using a standard *Expectation–Maximization* (EM) algorithm (Dempster et al. 1977; Krishnan and McLachlan 1997). This is briefly discussed below.

5.1 Standard EM algorithm to estimate mixture parameters

The standard EM algorithm is based on maximizing the log-likelihood function of the data, $\mathscr{L}(D|\boldsymbol{\Phi})$. The maximum likelihood estimates are then given as $\boldsymbol{\Phi}_{ML} = \arg\max_{\boldsymbol{\Phi}} \mathscr{L}(D|\boldsymbol{\Phi})$. Because of the absence of a closed form solution for $\boldsymbol{\Phi}_{ML}$, a gradient descent method is employed where the parameter estimates are iteratively updated until convergence to some local optimum 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:

- *Expectation-step*: Each datum $\mathbf{x}_i$ has fractional membership in each of the mixture components. These partial memberships are defined using the *responsibility matrix* (Eq. 16), where r_{ij} denotes the conditional probability of a datum $\mathbf{x}_i$ belonging to the jth component. The effective membership associated with the jth component is then given by n_j.

$$r_{ij} = \frac{w_j f(\mathbf{x}_i; \Theta_j)}{\sum_{k=1}^{M} w_k f(\mathbf{x}_i; \Theta_k)}, \quad \forall\, 1 \leq i \leq N, 1 \leq j \leq M \quad \text{and} \quad n_j = \sum_{i=1}^{N} r_{ij} \tag{16}$$

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

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 and are given below.

- *Gaussian:* The ML updates of the mean and covariance matrix are

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

- *von Mises-Fisher:* The resultant vector sum is updated as $\mathbf{R}_j^{(t+1)} = \sum_{i=1}^{N} r_{ij}^{(t)} \mathbf{x}_i$. If $R_j^{(t+1)}$ represents the magnitude of vector $\mathbf{R}_j^{(t+1)}$, then the updated mean and concentration parameter are

$$\hat{\boldsymbol{\mu}}_j^{(t+1)} = \frac{\mathbf{R}_j^{(t+1)}}{R_j^{(t+1)}}, \quad \bar{R}_j^{(t+1)} = \frac{R_j^{(t+1)}}{n_j^{(t)}}, \quad \hat{\kappa}_j^{(t+1)} = A_d^{-1}\left(\bar{R}_j^{(t+1)}\right)$$

5.2 EM algorithm to estimate mixture parameters using MML

Encoding a mixture model using MML: We refer to the discussion in Wallace (2005) to describe the intuition behind mixture modelling using MML. The framework requires the encoding of (1) the model parameters, and (2) the data using those parameters. The statement costs for encoding the mixture model and the data can be decomposed into:

1. Encoding the *number of components* M: In order to encode the message losslessly, it is required to initially state the number of components. In the absence of background knowledge, one would like to model the prior belief in such a way that the probability decreases for greater number of components. If $h(M) \propto 2^{-M}$, then $I(M) = M \log 2 +$ constant. Alternatively, one could assume a uniform prior over M within some 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* $w_1, \ldots, w_M$ which are treated as parameters of a multinomial distribution with sample size n_j, $\forall 1 \leq j \leq M$. The length of encoding the weights is then given by the expression given by Boulton and Wallace (1969) as $I(\mathbf{w}) = \frac{(M-1)}{2} \log N - \frac{1}{2} \sum_{j=1}^{M} \log w_j - (M-1)!$

3. Encoding each of the *component parameters* Θ_j as given by $I(\Theta_j) = -\log \dfrac{h(\Theta_j)}{\sqrt{|\mathcal{F}(\Theta_j)|}}$ (discussed in Sect. 3).
4. Encoding the *data*: each datum $\mathbf{x}_i$ can be stated to a finite precision ϵ which is dictated by the accuracy of measurement.[1] The *probability* measure of a datum $\mathbf{x}_i \in \mathbb{R}^d$ is then given as $\Pr(\mathbf{x}_i) = \epsilon^d \Pr(\mathbf{x}_i; \mathscr{M})$ where $\Pr(\mathbf{x}_i; \mathscr{M})$ is the *mixture density*. Hence, the *total* length of encoding the entire data $D = \{\mathbf{x}_i\}, 1 \leq i \leq N$ is then given by $I(D|\boldsymbol{\Phi}) = -\sum_{i=1}^{N} \log \Pr(\mathbf{x}_i) = -Nd \log \epsilon - \sum_{i=1}^{N} \log \sum_{j=1}^{M} w_j f_j(\mathbf{x}_i; \Theta_j)$

Thus, the total message length of a M component mixture is given by Eq. 17. 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.

$$I(\boldsymbol{\Phi}, D) = I(M) + I(\mathbf{w}) + \sum_{j=1}^{M} I(\Theta_j) + I(D|\boldsymbol{\Phi}) + \text{constant} \tag{17}$$

Estimating the mixture parameters: The parameters of the mixture model are those that *minimize* Eq. 17. To achieve this, we use the standard EM algorithm (Sect. 5.1), where the parameters are iteratively updated using their respective *MML estimates*. The component weights are obtained by differentiating Eq. 17 with respect to w_j under the constraint $\sum_{j=1}^{M} w_j = 1$ and are computed as:

$$w_j^{(t+1)} = \frac{n_j^{(t)} + \frac{1}{2}}{N + \frac{M}{2}} \tag{18}$$

The parameters of the jth component are updated using $r_{ij}^{(t)}$ and $n_j^{(t)}$ (Eq. 16), the partial memberships assigned to the jth component at some intermediate iteration t and and are given below.

- *Gaussian:* The MML updates of the mean and covariance matrix are

$$\hat{\boldsymbol{\mu}}_j^{(t+1)} = \frac{1}{n_j^{(t)}} \sum_{i=1}^{N} r_{ij}^{(t)} \mathbf{x}_i \quad \text{and} \quad \hat{\mathbf{C}}_j^{(t+1)} = \frac{1}{n_j^{(t)} - 1} \sum_{i=1}^{N} r_{ij}^{(t)} \left(\mathbf{x}_i - \hat{\boldsymbol{\mu}}_j^{(t+1)}\right) \left(\mathbf{x}_i - \hat{\boldsymbol{\mu}}_j^{(t+1)}\right)^T \tag{19}$$

- *von Mises-Fisher:* The resultant vector sum is updated as $\mathbf{R}_j^{(t+1)} = \sum_{i=1}^{N} r_{ij}^{(t)} \mathbf{x}_i$. If $R_j^{(t+1)}$ represents the magnitude of vector $\mathbf{R}_j^{(t+1)}$, then the updated mean is given by Eq. 20. The MML update of the concentration parameter $\hat{\kappa}_j^{(t+1)}$ is obtained by solving $G(\hat{\kappa}_j^{(t+1)}) = 0$ after substituting $N \rightarrow n_j^{(t)}$ and $R \rightarrow R_j^{(t+1)}$ in Eq. 12.

$$\hat{\boldsymbol{\mu}}_j^{(t+1)} = \mathbf{R}_j^{(t+1)} / R_j^{(t+1)} \tag{20}$$

The EM terminates when the change in the total message length (improvement rate) between successive iterations is less than some predefined threshold. The difference between the two

[1] We 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*.

variants of standard EM discussed above is firstly the objective function that is being optimized. In Sect. 5.1, the log-likelihood function is *maximized* which corresponds to $I(D|\boldsymbol{\Phi})$ term in Sect. 5.2. Equation (17) includes additional terms that correspond to the cost associated with stating the mixture parameters. Secondly, in the M-step, in Sect. 5.1, the components are updated using their ML estimates whereas in Sect. 5.2, the components are updated using their MML estimates.

Issues arising from the use of EM: The standard EM algorithms outlined above can be used only when the number of mixture components M is fixed or known *a priori*. Even when the number of components are fixed, EM has potential pitfalls. 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, EM 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.

6 Existing methods of inferring the number of mixture components

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). There are infinitely many mixtures that one can use to model some given data. Any method that aims to selectively determine the optimal number of components should 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). We briefly review some of these methods and then proceed to explain our proposed method.

AIC (Akaike 1974) & *BIC* (Schwarz 1978): AIC in the simplest form adds the *number* of free parameters p to the negative log-likelihood expression. BIC, similar to AIC, adds a constant penalty of $(1/2)\log N$, for each free parameter in the model. Rissanen (1978) formulated minimum description length (MDL) which formally coincides with BIC (Oliver et al. 1996; Figueiredo and Jain 2002). The formulations of AIC and BIC/MDL suggest that the parameter cost associated with adopting a model is dependent only on the number of free parameters and *not* on the parameter values themselves. In other words, the 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 their characterizing means and covariance matrices. This is an oversimplifying assumption that can hinder proper inference.

The criteria can be interpreted under the MML framework wherein 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 defined by Eq. 11 (Figueiredo and Jain 2002). 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).

MML Unsupervised (Oliver et al. 1996): A MML-based scoring function akin to the one shown in Eq. 17 was used to model Gaussian mixtures. However, the authors only consider the specific case of Gaussians with diagonal covariance matrices, and fail to provide a general method dealing with full covariance matrices.
Approximate Bayesian (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 tradeoff between model complexity and the quality of fit. Although the formulation is an improvement over Oliver et al. (1996), 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.
- The computation of the determinant of the Fisher matrix is approximated by computing the Hessain $|H|$. 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. MML formulation requires the use of the expected value.
- Further, 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 eigen values in the Hessian expression. The empirical Fisher computed in this form does not guarantee the characteristic invariance property of the classic MML method (Oliver and Baxter 1994).

Integrated complete likelihood (ICL) (Biernacki et al. 2000): The ICL criterion *maximizes* the *complete log-likelihood* and has a BIC-like formulation. The scoring function penalizes each free parameter by a constant value and does not account for the model parameters.
Search method to determine the optimal number of mixture components: Across the methods that use these criteria (Akaike 1974; Schwarz 1978; Oliver et al. 1996; Roberts et al. 1998; Biernacki et al. 2000), a rigorous treatment on the selection of number of mixture components M is lacking. A rudimentary version used in conjunction with these criteria is to experiment with different values of M and choose the one which results in the optimum value (least AIC/BIC, minimum message length or minimum ICL value). For each M, the standard EM algorithm is initialized a certain number of times and the trial resulting in the best EM outcome is chosen.
Unsupervised learning of finite mixtures (Figueiredo and Jain 2002): The method uses the MML criterion to formulate the scoring function given by Eq. 21, where N_p is the *number* of free parameters per component and w_j is the component weight. The formulation can be interpreted as a two-part message for encoding the model parameters and the observed data. The scoring function is derived from Eq. 17 by assuming the prior density of the component parameters to be a Jeffreys prior. If Θ_j is the vector of parameters describing the jth component, then the prior density $h(\Theta_j) \propto \sqrt{|\mathcal{F}(\Theta_j)|}$ (Jeffreys 1946). Similarly, a prior for weights would result in $h(w_1, \ldots, w_M) \propto (w_1 \ldots w_M)^{-1/2}$.

$$I(D, \boldsymbol{\Phi}) = \underbrace{\frac{N_p}{2} \sum_{j=1}^{M} \log\left(\frac{N w_j}{12}\right) + \frac{M}{2} \log \frac{N}{12} + \frac{M(N_p + 1)}{2}}_{\text{first part}} \underbrace{-\mathcal{L}(D|\boldsymbol{\Phi})}_{\text{second part}} \tag{21}$$

We note that the scoring function is consistent with the MML encoding scheme. However, it can be improved by amending the assumptions as detailed in Sect. 4. Further, the assumptions have the following side effects:

- The value of $-\log \dfrac{h(\Theta_j)}{\sqrt{|\mathscr{F}(\Theta_j)|}}$ gives the cost of encoding the component parameters. By assuming $h(\Theta_j) \propto \sqrt{|\mathscr{F}(\Theta_j)|}$, the message length associated with using any parameters Θ_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 eliminates the need to compute the Fisher. Consequently, the parameters are encoded to a constant precision which is a simplifying assumption. 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." By making this assumption, Figueiredo and Jain (2002) "*sidestep*" the difficulty associated with explicitly computing the Fisher information associated with the component parameters. Hence, for encoding the parameters of the entire mixture, *only* the cost associated with the component weights is considered.
- The code length to state each Θ_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 a MDL formulation because Nw_j gives the expected number of data points generated by the jth component. This is equivalent to the BIC criterion discussed earlier. We note that MDL/BIC are highly simplified versions of MML formulation and therefore, Eq. 21 does not capture the entire essence of complexity and goodness of fit accurately.

Search method of Figueiredo and Jain (2002): The method begins by assuming a large number of components and updates the weights iteratively as given by Eq. 22, where n_j is the effective membership of data points in the jth component. A component is annihilated when its weight becomes zero and consequently the number of mixture components decreases. We note that the search method of Figueiredo and Jain (2002) is an improvement over the methods they compare against. However, we make the following remarks about their method.

$$w_j = \frac{\max\left\{0, n_j - N_p/2\right\}}{\sum_{j=1}^{M} \max\left\{0, n_j - N_p/2\right\}} \tag{22}$$

- The method updates the weights as given by Eq. 22. 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 is ignored in subsequently. This imposes a lower bound on the membership of each component. As an example, for a Gaussian mixture in 10-dimensions, the number of free parameters per component is $N_p = 65$, and hence the lower bound is 33. Hence, in this case, if a component has ∼30 data, 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 originally generated by a mixture with two components and equal mixing proportions. The method would always infer a single component regardless of the separation between the two components. This is clearly a wrong inference! (see Sect. 8.4 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 handling several components.

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 Gaussian and von Mises-Fisher distributions in Sect. 4.

Secondly, in most of these methods, the search for an optimal number of mixture components is achieved by selecting the mixture that results in the best EM outcome out of many trials. 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. We, therefore, compare our method against that of Figueiredo and Jain (2002) and demonstrate its effectiveness. With this background, we formulate an alternate search heuristic to infer an optimal number of mixture components which aims to address the above limitations.

7 Proposed approach to infer an optimal mixture

The space of candidate mixture models to explain any given data is infinitely large. As per the MML criterion, the goal is to search for the mixture that has the least overall message length. If the number of mixture components are fixed, then the EM algorithm in Sect. 5.2 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. Our proposed search heuristic 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.

7.1 The complete algorithm

The pseudocode of our search method is presented in Algorithm 1. The basic idea behind the search strategy is to *perturb* a mixture from its current suboptimal state to obtain a new state (if the perturbed mixture results in a smaller message length). In general, if a (current) mixture has M components, it is perturbed using a series of *Split, Delete*, and *Merge* operations to check for improvement. Each component is split and the new $(M+1)$-component mixture is re-estimated. If there is an improvement (i.e., if there is a decrease in message length with respect to the current mixture), the new $(M+1)$-component mixture is retained. There are M 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 $M+1$ component mixture. The reason for this is, rather than use random initial

Algorithm 1: Achieve an optimal mixture model

```
1  $current$ ← one-component-mixture
2  while $true$ do
3      $components$ ← $current$ mixture components
4      $M$ ← number of$components$
5      for $\alpha \leftarrow 1$ to $M$ do                /* exhaustively split all components */
6          $splits[\alpha] \leftarrow$ SPLIT($current, components[\alpha]$)
7      $BestSplit \leftarrow best(splits)$                       /* remember the best split */
8      if $M > 1$ then
9          for $\alpha \leftarrow 1$ to $M$ do             /* exhaustively delete all components */
10             $deletes[\alpha] \leftarrow$ DELETE($current, components[\alpha]$)
11         $BestDelete \leftarrow best(deletes)$              /* remember the best deletion */
12     for $\alpha \leftarrow 1$ to $M$ do                 /* exhaustively merge all components */
13         $\beta$ ← closest-component($\alpha$)
14         $merges[\alpha] \leftarrow$ MERGE($current, \alpha, \beta$)
15     $BestMerge \leftarrow best(merges)$                       /* remember the best merge */
16     $BestPerturbation \leftarrow best(BestSplit, BestDelete, BestMerge)$    /* select the best
       perturbation */
17     $\Delta I$ ← message_length($BestPerturbation$) − message_length($current$)       /* check for
       improvement */
18     if $\Delta I < 0$ then
19         $current \leftarrow BestPerturbation$
20         $continue$
21     else
22         $break$
23 return $current$
```

values for the EM, it is better if we start from some already optimized state to reach to a better state. Similarly, each of the components is then deleted, one after the other, and the $(M-1)$-component mixture is compared against the current mixture. There are M 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 $(M-1)$-component mixtures are evaluated against the M component mixture. The best among these merged mixtures is then retained (see lines 12–15 in Algorithm 1).

We start by assuming a one component mixture. This component is split into two children that are locally optimized. If the split results in a better model, it is retained. For any M-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 M components. Before the execution of each operation, these memberships need to be adjusted and a EM is subsequently carried out to achieve an optimum with a different number of components. We will now examine each operation in detail and see how the memberships are affected after each operation.

7.2 Strategic operations employed to determine an optimal mixture model

SPLIT OPERATION *(Line 6 in Algorithm 1):* Let $R = [r_{ij}]$ be the $N \times M$ responsibility (membership) matrix and w_j be the weight of jth component in mixture $\mathscr{M}$. As an example, assume a component with index $\alpha \in \{1, M\}$ and weight w_α in the current mixture $\mathscr{M}$ is 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* the component α assuming a *two-component sub-mixture* with the data weighted as per their current memberships $r_{i\alpha}$. The remaining $M - 1$ components are untouched. An EM is carried out to optimize the two-component sub-mixture. The initial state and the subsequent updates in the Maximization-step are described below.

Parameter initialization of the two-component sub-mixture: The goal is to identify two distinct clusters within the component α. For *Gaussian* mixtures, to provide a reasonable starting point, we compute the direction of maximum variance of the parent component and locate two points which 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 available characteristic of data, *i.e.,* the distribution within the component α. For *von Mises-Fisher* mixtures, the maximum variance strategy (as for Gaussian mixtures) cannot be easily adopted, as the data is distributed on the hypersphere. Hence, in this work, we randomly allocate data memberships and compute the components' (initial) parameters.

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 $\Theta^{(k)}_\alpha$ be their respective parameters.

- The effective memberships are updated as $n^{(k)}_\alpha = \sum_{i=1}^N r^c_{ik}$ and $n^{(1)}_\alpha + n^{(2)}_\alpha = N$.
- As the sub-mixture comprises of two child components, substitute $M = 2$ in Eq. 18 to obtain the corresponding weight updates: $w^{(k)}_\alpha = (n^{(k)}_\alpha + \frac{1}{2})/(N+1)$ and $w^{(1)}_\alpha + w^{(2)}_\alpha = 1$.
- The component parameters $\Theta^{(k)}_\alpha = (\hat{\boldsymbol{\mu}}^{(k)}_\alpha, \hat{\mathbf{C}}^{(k)}_\alpha)$ for *Gaussian* mixtures are updated using Eq. 19. For *vMF* mixtures, the component parameters $\Theta^{(k)}_\alpha = (\hat{\boldsymbol{\mu}}^{(k)}_\alpha, \hat{\kappa}^{(k)}_\alpha)$ are updated using Eqs. 20 and 12. However, these updates are using the modified responsibility terms $r^{(k)}_i = r_{i\alpha} r^c_{ik}$. Since we are considering the sub-mixture, 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 untouched $M - 1$ components of $\mathscr{M}$ to result in a $M + 1$ component mixture $\mathscr{M}'$. An EM is finally carried out on the $M + 1$ components to estimate the parameters of $\mathscr{M}'$ and result in an optimized $(M + 1)$-component mixture.

EM initialization for $\mathscr{M}'$: Usually, the EM is started by a random initialization of the members. However, because the two-component sub-mixture is now optimal and the $M - 1$ components in $\mathscr{M}$ are also in an optimal state, we exploit this situation to initialize the EM (for $\mathscr{M}'$) with a reasonable starting point. As mentioned above, the component with index α with component 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 $\mathscr{M}'$. Let $R' = [r'_{ij}] \forall 1 \leq i \leq N, 1 \leq j \leq M+1$ be the responsibility matrix for the new mixture $\mathscr{M}'$ and let w'_j be the component weights in $\mathscr{M}'$.

- *Component weights:* These are initialized as $w'_j = w_j$ if $j < \alpha$ and $w'_j = w_{j-1}$ if $j > \alpha + 1$. Further, $w'_\alpha = w_\alpha w_\alpha^{(1)}$ and $w'_{\alpha+1} = w_\alpha w_\alpha^{(2)}$.
- *Memberships:* The responsibility matrix R' is initialized for $\mathbf{x}_i \, \forall 1 \leq i \leq N$ as $r'_{ij} = r_{ij}$ if $j < \alpha$ and $r'_{ij} = r_{i\,j-1}$ if $j > \alpha + 1$. Further, $r'_{i\alpha} = r_{i\alpha} r^c_{i1}$ and $r'_{i\,\alpha+1} = r_{i\alpha} r^c_{i2}$. The effective memberships in $\mathscr{M}'$ are $n'_j = \sum_{i=1}^N r'_{ij} \, \forall 1 \leq j \leq M+1$.

With these starting points, the parameters of $\mathscr{M}'$ are estimated using the EM algorithm with updates in the Maximization-step given by Eqs. 18, 19, and 20. The EM results in local convergence of the $(M+1)$-component mixture. If the message length of encoding data using $\mathscr{M}'$ is lower than that due to $\mathscr{M}$, that means the perturbation of $\mathscr{M}$ because of splitting component α resulted in $\mathscr{M}'$ that compresses the data better, and hence, is a better mixture model.

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 the corresponding weight w_α is to be deleted from $\mathscr{M}$ to generate a $M - 1$ component mixture $\mathscr{M}'$. 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 $\mathscr{M}'$ as follows.

EM initialization for $\mathscr{M}'$: Let $R' = [r'_{ij}]$ be the $N \times (M-1)$ responsibility matrix for the new mixture $\mathscr{M}'$ and let w'_j be the weight of jth component in $\mathscr{M}'$.

- *Component weights:* The weights are initialized using $w'_j = \dfrac{w_j}{1 - w_\alpha}$ if $j < \alpha$ and $w'_j = \dfrac{w_{j+1}}{1 - w_\alpha}$ if $j \geq \alpha$. 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 Eq. 18).
- *Memberships:* The responsibility matrix R' is initialized for $\mathbf{x}_i \, \forall 1 \leq i \leq N$ as $r'_{ij} = \dfrac{r_{ij}}{1 - r_{i\alpha}}$ if $j < \alpha$ and $r'_{ij} = \dfrac{r_{i\,(j+1)}}{1 - r_{i\alpha}}$ if $j \geq \alpha$. The effective memberships in $\mathscr{M}'$ are $n'_j = \sum_{i=1}^N r'_{ij} \forall 1 \leq j \leq M - 1$. It is possible for a datum $\mathbf{x}_i$ to have complete membership in component α (*i.e.,* $r_{i\alpha} = 1$), in which case, its membership is equally distributed among the other $M-1$ components (i.e., $r'_{ij} = 1/(M-1), \ \forall j \in \{1, M-1\}$).

With these readjusted weights and memberships, and the constituent $M - 1$ components, the traditional EM algorithm is used to estimate the parameters of the new mixture $\mathscr{M}'$. If the resultant message length of encoding data using $\mathscr{M}'$ is lower than that due to $\mathscr{M}$, that means the perturbation of $\mathscr{M}$ because of deleting component α resulted in a new mixture $\mathscr{M}'$ with better explanatory power, which is an improvement over the current mixture.

Merge operation *(Line 14 in Algorithm 1)*: The idea is to join a pair of components in $\mathscr{M}$ and determine whether the resulting $(M-1)$-component mixture $\mathscr{M}'$ is any better than the current mixture $\mathscr{M}$. One strategy to identify an improved mixture 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 $O(M^2)$, which could be significant for large values of M. Another strategy is to consider merging components which are "close" to each other. For a given component, we identify its *closest* component 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 $O(M)$ as computation of KL-divergence is a constant time operation. For every component in $\mathscr{M}$, its closest match is identified and they are merged to obtain a $M-1$ component mixture $\mathscr{M}'$. Merging the pair involves reassigning the component weights and the memberships. An EM algorithm is then employed to optimize $\mathscr{M}'$. Assume components with indices α and β are to be 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 $M-2$ remaining components of $\mathscr{M}$ to produce a $(M-1)$-component mixture $\mathscr{M}'$.

EM initialization for $\mathscr{M}'$: Let $w^{(m)}$ and $r_i^{(m)}$ be the weight and responsibility vector of the merged component m respectively. They are given as: $w^{(m)} = w_\alpha + w_\beta$ and $r_i^{(m)} = r_{i\alpha} + r_{i\beta}, 1 \leq i \leq N$. The parameters of this merged component are estimated using Eq. 19 (for *Gaussian*) and Eqs. 20 and 12 (for *vMF*). These updates are using the modified responsibility terms $r_i^{(m)}$. The merged component m with weight $w^{(m)}$, responsibility vector $r_i^{(m)}$, and parameters $\Theta^{(m)}$ is then integrated with the $M-2$ components. The merged component and its associated memberships along with the $M-2$ other components serve as the starting point for optimizing the new mixture $\mathscr{M}'$. If $\mathscr{M}'$ results in a lower message length compared to $\mathscr{M}$, that means ths perturbation due to merging the pair resulted in an improved mixture.

7.3 Illustrative example of our search procedure

We explain the proposed inference of mixture components through the following example that was also considered by Figueiredo and Jain (2002). The bivariate Gaussian mixture shown in Fig. 1 has three components with equal weights of 1/3 each and means at (−2,0), (0,0), and (2,0). The covariance matrices of the components are the same and are equal to diag{2, 0.2}. We simulate 900 data points from the mixture consistent with the work of Figueiredo and

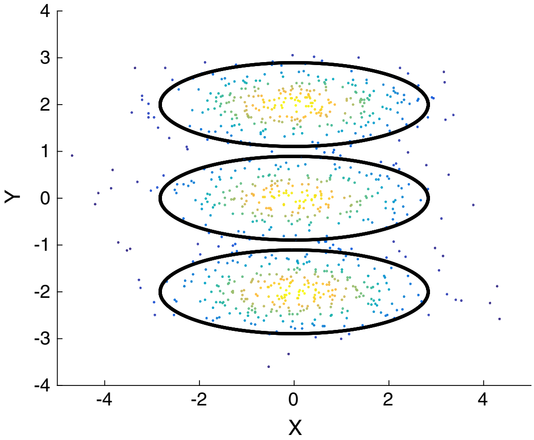
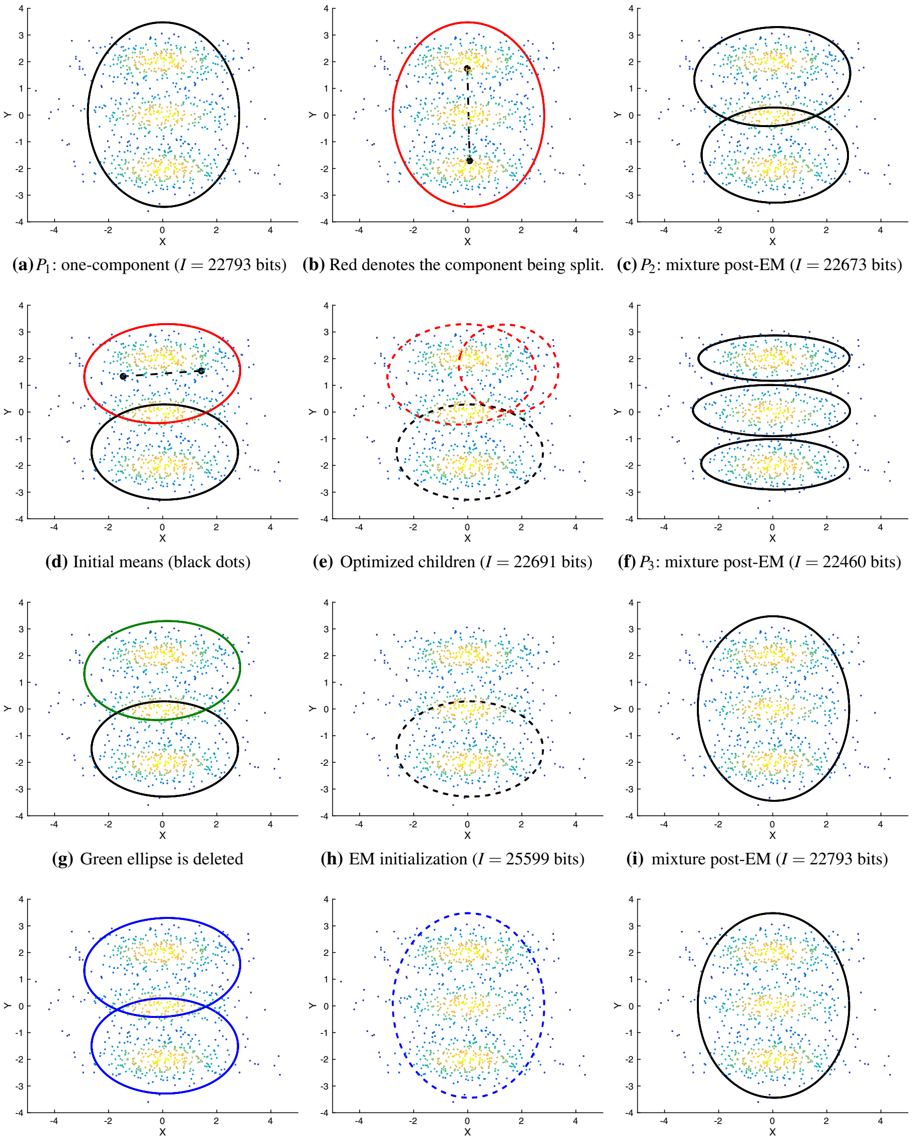


Fig. 1 Original mixture with three components and equal mixing proportions



(a) P_1: one-component ($I = 22793$ bits) **(b)** Red denotes the component being split. **(c)** P_2: mixture post-EM ($I = 22673$ bits)

(d) Initial means (black dots) **(e)** Optimized children ($I = 22691$ bits) **(f)** P_3: mixture post-EM ($I = 22460$ bits)

(g) Green ellipse is deleted **(h)** EM initialization ($I = 25599$ bits) **(i)** mixture post-EM ($I = 22793$ bits)

(j) Blue ellipses are merged **(k)** EM initialization **(l)** mixture post-EM ($I = 22793$ bits)

Fig. 2 **a–c** Splitting of the component in P_1 results in an improved mixture P_2. The *dotted line* is the direction of maximum variance. **d–l** Second iteration (operations on the first component in P_2): **d–f** *Splitting* results in an improved mixture P_3. **g–i** *Deleting*—no improvement. **j–l** *Merging* the two components—no improvement

Jain (2002) and employ the proposed search strategy. The progression of the search method using various operations is detailed here.

Search for the optimal mixture model: The method begins by inferring a one-component mixture P_1 (see Fig. 2a). It then splits this component (as described in *Split* step of Sect. 7.2) and checks whether there is an improvement in explanation. The red ellipse in Fig. 2b 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 and this results in a mixture P_2 shown in Fig. 2c. Since the new mixture has a lower message length, the current is updated as P_2.

In the second iteration, each component in P_2 is iteratively split, deleted, and merged. Fig. 2d–f shows the splitting (red) of the first component. On splitting, the new mixture P_3 results in a lower message length. Deletion of the first component is shown in Fig. 2g–i. Before merging the first component, we identify its closest component (the one with the least KL-divergence) (see Fig. 2j). Deletion and merging operations, in this case, do not result in an improvement. These two operations have different intermediate EM initializations (Fig. 2h,k) 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). P_3 is updated as the new parent and the series of split, delete, and merge operations are carried out on all components in P_3 (not shown here pictorially). However, 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 P_3 is considered as the best.

In different stages of the search method, we have different intermediate mixtures. EM 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.

Variation of the two-part message length: 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 $M = 15$ as an example) and plot their resultant message lengths. For each $M > 3$, the standard EM algorithm (Sect. 5.2) is employed to infer the mixture parameters. Figure 3 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 $M = 3$ components are inferred. Starting from $M = 4$, 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. We further elaborate on the reason for the initial decrease and subsequent increase in the total message length. As per MML evaluation criterion, the message length comprises of two parts – statement cost 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 M. 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 Fig. 3. 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 Fig. 3 (blue curve). There is a sharp fall until $M = 3$; 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 $M = 3$. The optimal balance is achieved when $M = 3$. In summary, the message length at $M = 3$ components was rightly

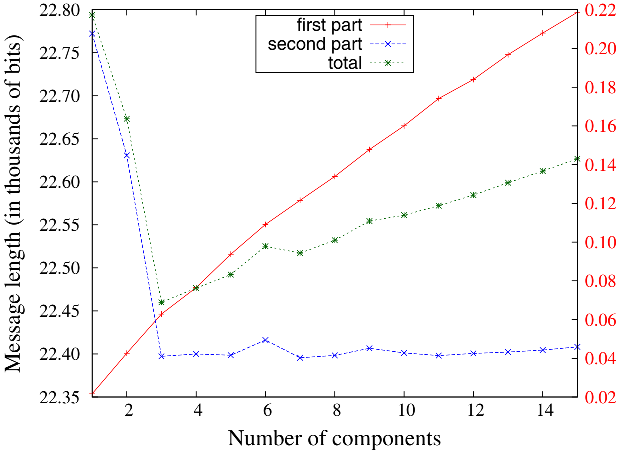


Fig. 3 Variation of the individual parts of the total message length with increasing number of components (note the two Y-axis have different scales—the first part follows the right side Y-axis; the second part and total message lengths follow the left side Y-axis)

observed to be the optimum for this example. We note that for a fixed number of mixture components, the EM algorithm for the MML metric is monotonically decreasing. However, while searching for the number of components, MML continues to decrease until some optimum is found and then steadily increases as illustrated through this example. Another example is discussed in Kasarapu and Allison (2015) (which is an extended version of this paper [2]) where the evolution of the inferred mixture is explored in the case of a mixture with overlapping components.

8 Experiments with Gaussian mixtures

We compare our proposed inference methodology against the widely cited method of Figueiredo and Jain (2002). The authors tested the performance of their method against that of Bayesian Information Criterion (BIC), Integrated Complete Likelihood (ICL), and approximate Bayesian (LEC) methods (discussed in Sect. 6). It was shown that the method of FJ[3] was superior than BIC, ICL and LEC (using Gaussian mixtures). In this section, we demonstrate through a series of experiments that our approach to infer mixtures fares better when compared to that of FJ. The experimental setup is as follows: we use a Gaussian mixture $\mathscr{M}^t$ (true distribution), generate a random sample from it, and infer the mixture using the data. This is repeated 50 times and we compare the performance of our method against that of FJ. As part of our analysis, we compare the number of inferred components as well as the quality of inferred mixtures.

[2] The extended version is available at http://arxiv.org/abs/1502.07813.

[3] From here on, we will use the short form FJ to refer to Figueiredo and Jain (2002).

8.1 Methodologies used to compare the mixtures inferred by our proposed approach and FJ's method

Comparing message lengths: The MML framework allows us to objectively compare competing mixture models by computing the total message length used to encode the data. The difference in message lengths gives the log-odds posterior ratio of any two mixtures (Eq. 10). Given some observed data, and any two mixtures, one can determine which of the two best explains the data. Our search methodology uses the scoring function (I_{MML}) defined in Eq. 17. As elaborated in Sect. 6, FJ use an approximated MML-like scoring function (I_{FJ}) given by Eq. 21.

We employ our search method and FJ's method to infer the mixtures using the same data; let these inferred mixtures be $\mathscr{M}^*$ and $\mathscr{M}^{FJ}$ respectively. We compute two quantities:

$$\Delta I_{MML} = I_{MML}(\mathscr{M}^{FJ}) - I_{MML}(\mathscr{M}^*) \quad \text{and} \quad \Delta I_{FJ} = I_{FJ}(\mathscr{M}^{FJ}) - I_{FJ}(\mathscr{M}^*) \tag{23}$$

We use the two different scoring functions to compute the differences in message lengths of the resulting mixtures $\mathscr{M}^{FJ}$ and $\mathscr{M}^*$. Since the search method used to obtain $\mathscr{M}^*$ optimizes the scoring function I_{MML}, it is expected that $I_{MML}(\mathscr{M}^*) < I_{MML}(\mathscr{M}^{FJ})$ and consequently $\Delta I_{MML} > 0$. This implies that our method is performing better using our defined objective function. However, if $I_{FJ}(\mathscr{M}^*) < I_{FJ}(\mathscr{M}^{FJ})$, this indicates that our inferred mixture $\mathscr{M}^*$ results in a lower value of the scoring function that is defined by FJ. Such an evaluation not only demonstrates the superior performance of our search (leading to $\mathscr{M}^*$) using our defined scoring function but also proves it is better using the scoring function as defined by FJ.

Kullback Leibler (KL) divergence: In addition to using message length based evaluation criterion, we also compare the mixtures using KL-divergence (Kullback and Leibler 1951). The metric gives a measure of the similarity between two distributions (the lower the value, the more similar the distributions). 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 KL-divergence). In experiments relating to mixture simulations, we know the true mixture $\mathscr{M}^t$ from which the data $\{\mathbf{x}_i\}$, $1 \leq i \leq N$ is being sampled. The KL-divergence is given by the following expression:

$$D_{KL}(\mathscr{M}^t \,||\, \mathscr{M}) = E_{\mathscr{M}^t}\left[\log \frac{\Pr(\mathbf{x}, \mathscr{M}^t)}{\Pr(\mathbf{x}, \mathscr{M})}\right] \approx \frac{1}{N}\sum_{i=1}^{N} \log \frac{\Pr(\mathbf{x}_i, \mathscr{M}^t)}{\Pr(\mathbf{x}_i, \mathscr{M})} \tag{24}$$

where $\mathscr{M}$ is an inferred mixture distribution ($\mathscr{M}^*$ or $\mathscr{M}^{FJ}$) whose *closeness* to the true mixture $\mathscr{M}^t$ is to be determined.

8.2 Bivariate 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 $\mathscr{M}^t$ 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. We repeat the same experiment here and compare with the results of FJ. The separation δ between the means is gradually increased 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 increases. For the mixtures inferred using both the approaches, the differences in message lengths ΔI_{MML} and ΔI_{FJ} are close to zero. The KL-divergences for the inferred mixtures

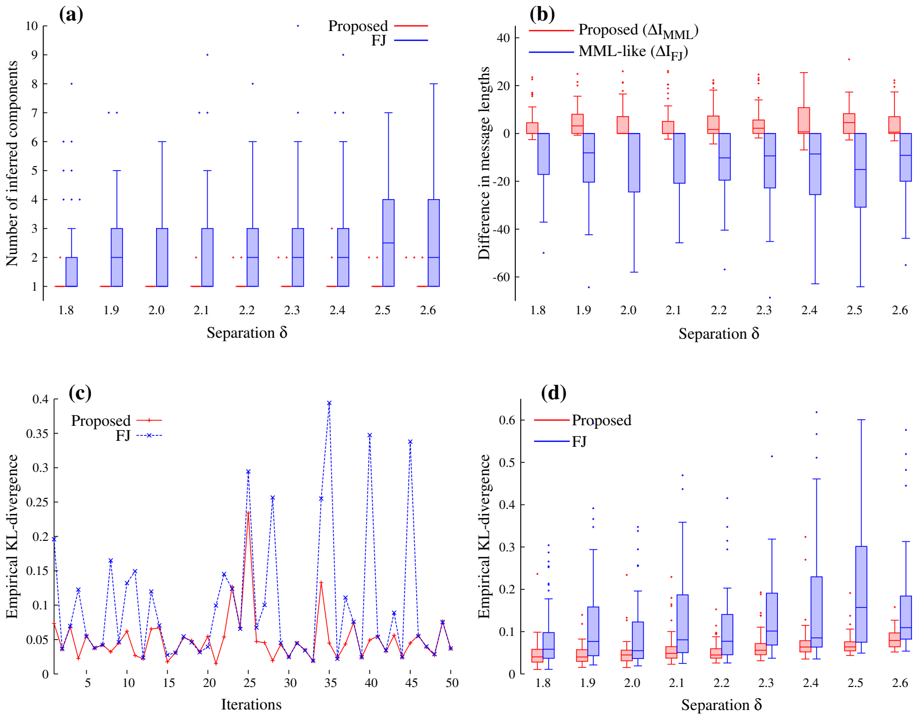


Fig. 4 Bivariate mixture simulation ($N = 100$ and over 50 simulations). **a** *Box-whisker* plot showing the variability in the number of inferred components. **b** Difference in message lengths computed using the two different scoring functions (see Eq. 23). **c** KL-divergence of inferred mixtures when $\delta = 2.0$. **d** KL-divergence for all values of $\delta \in \{1.8, \ldots, 2.6\}$

are also the same. Therefore, for this experimental setup, the performance of both the methods is roughly similar.

As the difference between the two search methods is not apparent from this experiment, we wanted to investigate the behaviour of the methods with smaller samples. We repeated the experiment 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 Fig. 4a). 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 Fig. 4a. 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. Further, we evaluate the correctness of the mixtures inferred by the two search methods by comparisons using the message length formulations and KL-divergence. Figure 4b shows the boxplot of the difference in message lengths of the mixtures $\mathscr{M}^*$ inferred using our proposed search method and the mixtures $\mathscr{M}^{FJ}$ inferred using FJ's method. We observe that $\Delta I_{MML} > 0$ across all values of δ for the 50 simulations. As per Eq. 23, we have $I_{MML}(\mathscr{M}^*) < I_{MML}(\mathscr{M}^{FJ})$. This implies that $\mathscr{M}^*$ has a lower message length compared to $\mathscr{M}^{FJ}$ when evaluated using our scoring function. Similarly, we have $\Delta I_{FJ} < 0$, *i.e.,* $I_{FJ}(\mathscr{M}^*) > I_{FJ}(\mathscr{M}^{FJ})$. This implies that $\mathscr{M}^{FJ}$ has a lower message length compared to $\mathscr{M}^*$ when evaluated using FJ's scoring function. These results are not surprising as $\mathscr{M}^*$ and $\mathscr{M}^{FJ}$ are obtained using the search methods which optimize the respective MML and MML-like scoring functions.

We then analyzed the KL-divergence of $\mathscr{M}^*$ and $\mathscr{M}^{FJ}$ with respect to the true bivariate mixture $\mathscr{M}^t$ over all 50 simulations and across all values of δ. Ideally, the KL-divergence should be close to zero. Figure 4c shows the KL-divergence of the mixtures inferred using the two search methods with respect to $\mathscr{M}^t$ when the separation is $\delta = 2.0$. The proposed search method infers mixtures whose KL-divergence (denoted by red lines) is close to zero, and more importantly less than the KL-divergence of mixtures inferred by FJ's search method of (denoted by blue lines). The same type of behaviour is noticed with other values of δ. Figure 4d compares the KL-divergence for varying values of δ. The median value of the KL-divergence due to the proposed search method is close to zero with not much variation. FJ's search method always result in mixtures whose KL-divergence is higher than that of ours. The results suggest that, in this case, mixtures $\mathscr{M}^{FJ}$ inferred by employing the FJ's search method deviate significantly from the true mixture distribution $\mathscr{M}^t$. This can also be explained by the fact that there is a wide spectrum of the number of inferred components (see Fig. 4a). This suggests that the MML-like scoring function is failing in its objective to control the tradeoff between complexity and quality of fit, and hence, is selecting more complex mixture models than necessary to describe the data.

8.3 Simulation of 10-dimensional mixtures

Along the same lines as the previous setup, FJ conducted another experiment for a 10-variate two-component mixture $\mathscr{M}^t$ 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 5a 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. We observe that our proposed method performs clearly better than that of Figueiredo and Jain (2002) across all values of δ. We also compared the quality of these inferred mixtures by calculating the difference in message lengths using the two scoring functions and the KL-divergence with respect to $\mathscr{M}^t$. For all values of δ, $\Delta I_{MML} > 0$, *i.e.,* our inferred mixtures $\mathscr{M}^*$ have a lower message length compared to $\mathscr{M}^{FJ}$ when evaluated using our scoring function. More interestingly, we also note that $\Delta I_{FJ} > 0$ (see Fig. 5c). This reflects that $\mathscr{M}^*$ have a lower message length compared to $\mathscr{M}^{FJ}$ when evaluated using the scoring function of Figueiredo and Jain (2002). This suggests that their search method results in a sub-optimal mixture $\mathscr{M}^{FJ}$ and fails to infer the better mixture $\mathscr{M}^*$.

In addition to the message lengths, we analyze the mixtures using KL-divergence. Similar to the bivariate example in Fig. 4c, the KL-divergence of our inferred mixtures $\mathscr{M}^*$ is lower than $\mathscr{M}^{FJ}$, the mixtures inferred by Figueiredo and Jain (2002). Figure 5d shows the boxplot of KL-divergence of the inferred mixtures $\mathscr{M}^*$ and $\mathscr{M}^{FJ}$. At higher values of $\delta >= 1.45$, the median value of KL-divergence is close to zero, as the number of correctly inferred components (Fig. 5a) is more than 90 %. However, our method always infers mixtures $\mathscr{M}^*$ with lower KL-divergence compared to $\mathscr{M}^{FJ}$. These experimental results demonstrate the superior performance of our proposed search method. Another experiment was carried out where $\delta = 1.20$ was held constant (i.e., extremely close components), gradually increased the sample size N, and plotted the average number of inferred components by running 50 simulations for each N. Figure 5b shows the results for the average number of inferred components as the amount of data increases. Our search method, on average, infers the true mixture when the sample size is $\sim$1000. However, FJ's search method requires larger amounts

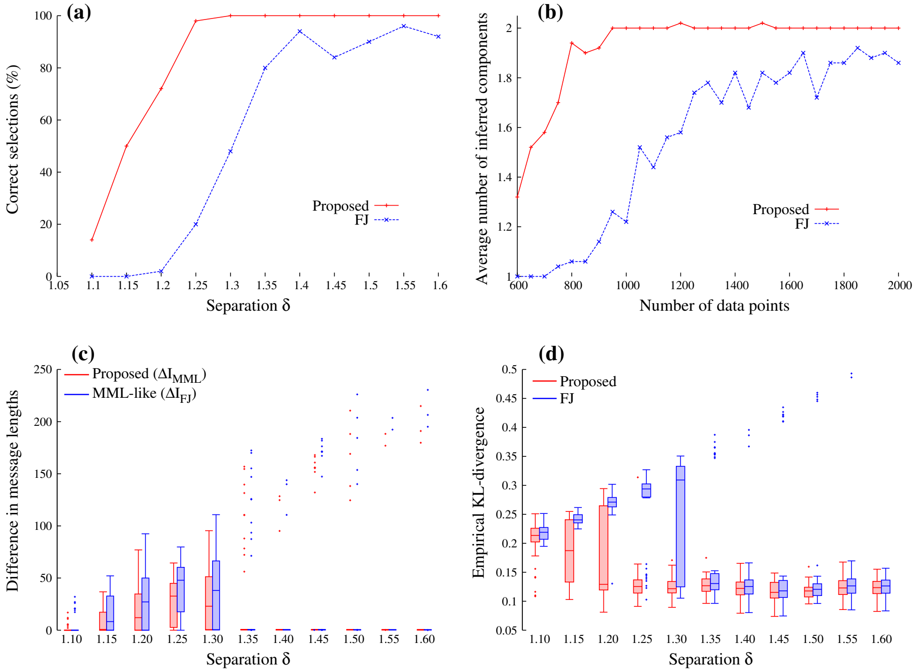


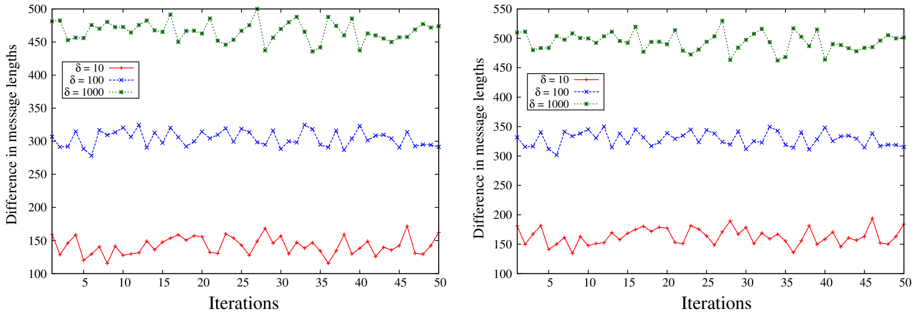
Fig. 5 10-dimensional mixture simulations. **a** Percentage of correct selections with varying δ for a fixed sample size of $N = 800$. **b** Average number of inferred mixture components with different sample sizes and $\delta = 1.20$ between component means. **c** Difference in message lengths of inferred mixtures. **d** *Box-whisker* plot of KL-divergence of inferred mixtures

of data; even with a sample size of 2000, the average number of inferred components is $\sim$1.9. In Fig. 5b, the red curve reaches the true number of 2 and saturates more rapidly than the blue curve.

8.4 The impact of weight updates as formulated by Figueiredo and Jain (2002)

One of the drawbacks associated with FJ's search method is due to the form of the updating expression for the component weights (Eq. 22). As discussed in Sect. 6, a particular instance of wrong 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.

We conducted the following experiment to demonstrate this behaviour: we considered the two-component 10-variate mixture $\mathscr{M}^t$ as before and randomly generate samples of size 50 from the mixture. Since the constituent components of $\mathscr{M}^t$ have equal weights, on average, each component has a membership of 25. We used $\delta = \{10, 100, 1000\}$, so that the two components are well apart from each other. For each δ, we run 50 simulations and analyze the number of inferred components. As expected, FJ's search method always infers a mixture with one component regardless of the separation δ. In contrast, our method always 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 Sect. 8.1.



(a) ΔI_{MML} **(b)** ΔI_{FJ}

Fig. 6 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 (see Eq. 23)

Figure 6a shows the difference in message lengths ΔI_{MML} given in Eq. 23. We observe that $\Delta I_{MML} > 0$ for all δ. This demonstrates that our inferred mixtures $\mathscr{M}^*$ have lower message lengths compared to mixtures $\mathscr{M}^{FJ}$ using our scoring function. The same phenomenon is observed when using FJ's MML-like scoring function. In Fig. 6b, we observe that $\Delta I_{FJ} > 0$, which means our search based mixtures $\mathscr{M}^*$ have lower message lengths compared to mixtures $\mathscr{M}^{FJ}$ when evaluated using their scoring function. This demonstrates that $\mathscr{M}^*$ is a better mixture as compared to $\mathscr{M}^{FJ}$ and their 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 $\mathscr{M}^{FJ}$, the second part of the message (see Eq. 21) which corresponds to the negative log-likelihood term increases because of poorer fit to the data. The two modes in the data 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 poorer fit. This is clearly an incorrect inference. We further strengthen our case by comparing the KL-divergence of the inferred mixtures $\mathscr{M}^*$ and $\mathscr{M}^{FJ}$ with respect to $\mathscr{M}^t$. Figure 7 illustrates the results. As δ increases, the blue coloured plots shift higher. These correspond to mixtures $\mathscr{M}^{FJ}$ inferred by FJ's method. Our search method, however, infers mixtures $\mathscr{M}^*$ which have lower KL-divergence. The figure indicates that the inferred mixtures $\mathscr{M}^*$ are more similar to the true distribution as compared to mixtures $\mathscr{M}^{FJ}$.

8.5 Analysis of the computational cost

At any intermediate stage of the search procedure, a *current* mixture with M components requires M 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 EM, we used a threshold of 10^{-5} which was the same as used by FJ in their experiments. FJ's method also requires to start from an initial large number of components. We used 25 as an initial number 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. We examine with respect to the simulations that were carried out previously. For the bivariate mixture discussed in Sect. 8.2, the number of resulting EM iterations when the sample sizes are $N = 800$ and $N = 100$ are compared in Fig. 8a, b respectively. As per the discussion in Sect. 8.2, at $N = 800$, the average number of components inferred

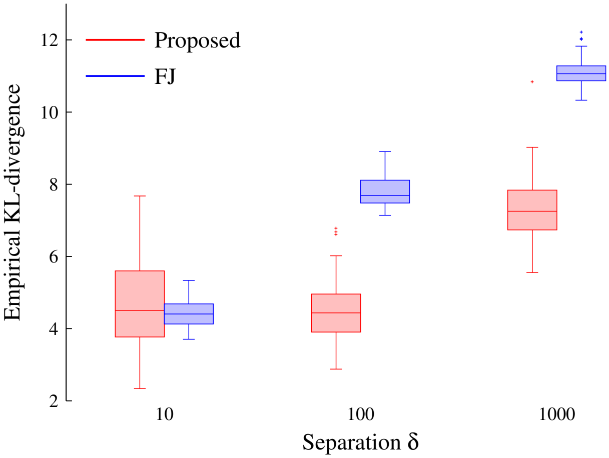
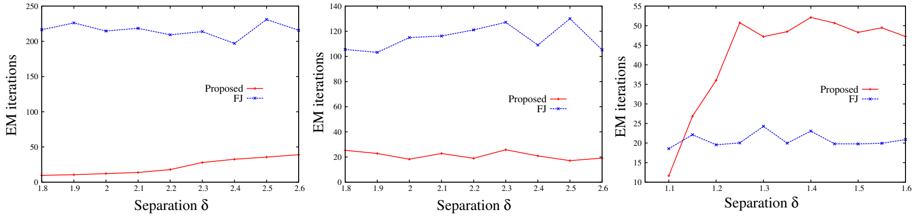


Fig. 7 *Box-whisker plot* of KL-divergence of mixtures inferred by the two search methods. A random sample of size $N = 50$ is generated for each δ and this is repeated 50 times



(a) Bivariate mixture ($N = 800$) **(b)** Bivariate mixture ($N = 100$) **(c)** 10-variate mixture ($N = 800$)

Fig. 8 Number of EM iterations performed during the mixture simulations discussed in Sects. 8.2 and 8.3

by the two methods are about the same. However, the number of EM iterations required by FJ's method is greater than 200 across all values of δ (see Fig. 8a). 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 Fig. 8b). In this case, the proposed method not only infers better mixtures (as discussed in Sect. 8.2) but is also conservative with respect to the computational cost.

For the simulation results corresponding to the 10-variate mixtures (Sect. 8.3), the proposed method requires close to 50 iterations on average, while FJ's method requires about 20 (see Fig. 8c). However, the mixtures inferred by our method fare better when compared to that of FJ (see Fig. 5). Furthermore, for the simulation results (see Sect. 8.4), FJ's method stops after 3 EM iterations. 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 the ability of our search method to perform better than the widely used FJ's method. We compared the resulting mixtures using our 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-divergence, to establish the similarity of our inferred mixtures to the true distributions. The reader is directed to Kasarapu and Allison (2015) for analysis on the *Acidity* (Richardson and Green 1997; McLachlan and Peel 1997) and *Iris* (Anderson 1935; Fisher 1936) datasets.

9 Experiments with von Mises-Fisher distributions

We compare our MML-based parameter inference with the current state of the art vMF estimators (discussed in Sect. 2.2). Tests include the analysis of the MML estimates of the concentration parameter: κ_{MN} is the approximation of MML estimate using Newton's method and κ_{MH} is the approximation using Halley's method (see Eqs. 14 and 15) against the traditionally used approximations. Estimation of the vMF mean direction is the same across all these methods. Estimation of κ, however, differs and hence, the corresponding results are presented. Through these experiments, we demonstrate that the MML estimates perform better than its competitors. These are followed by experiments demonstrating how these estimates aid in the inference of vMF mixtures. These experiments illustrate the application of the proposed search method to infer vMF mixtures using empirical studies and on real world datasets.

MML-based parameter estimation for a vMF distribution: For different values of dimensionality d and concentration parameter κ, data of sample size N are randomly generated from a vMF distribution using the algorithm proposed by Wood (1994). The parameters of a vMF distribution are estimated using the previously mentioned approximations. Let $\hat{\kappa} = \{\kappa_T, \kappa_N, \kappa_H, \kappa_{MN}, \kappa_{MH}\}$ denote the estimate of κ due to the respective methods.

Errors in κ estimation: We first report the errors in κ estimation by calculating the absolute error $|\hat{\kappa} - \kappa|$ and the squared error $(\hat{\kappa} - \kappa)^2$ averaged over 1000 simulations. The relative error $\frac{|\hat{\kappa}-\kappa|}{\kappa}$ can be used to measure the percentage error in κ estimation. The following observations are made based on the results shown in Table 1a.

- For $N = 10, d = 10, \kappa = 10$, the average relative error of $\kappa_T, \kappa_N, \kappa_H$ is ∼25 %; for κ_{MN}, κ_{MH}, it is ∼20 %. When N is increased to 100, the average relative error of κ_T is 5.09 %, κ_N, κ_H is 5.05 %, and κ_{MN}, κ_{MH} 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 inference becomes more accurate. The plots shown in Fig. 9 reflect this behaviour. The mean error at lower values of $N = 5, 10, 20, 30$ is noticeable. However, as N is increased to 1000, there is a drastic drop in the error. We note that this behaviour is consistent across all the different estimation methods.
- For fixed N and d, increasing κ increases the mean absolute error. However, the average relative error decreases. As an example, for $N = 100, d = 100, \kappa = 10$, the average relative error of $\kappa_T, \kappa_N, \kappa_H$ is ∼42 %; for κ_{MN}, κ_{MH}, it is 36.7 and 34 % respectively. When κ is increased to 100, the error rate for $\kappa_T, \kappa_N, \kappa_H$ drops to 2.18 % and for κ_{MN}, κ_{MH}, it drops to 1.68 %. Further increasing κ by an order of magnitude to 1000 results in average relative errors of 1.4 % for $\kappa_T, \kappa_N, \kappa_H$ and 1.1 % for κ_{MN}, κ_{MH}. This indicates that as the data becomes more concentrated, the errors in parameter estimation decrease.
- There is no clear pattern of the variation in error rates that can be observed when d is changed keeping N and κ fixed. However, in any case, MML-based approximations have the least mean absolute and mean squared error.

Table 1 Comparison of κ estimates

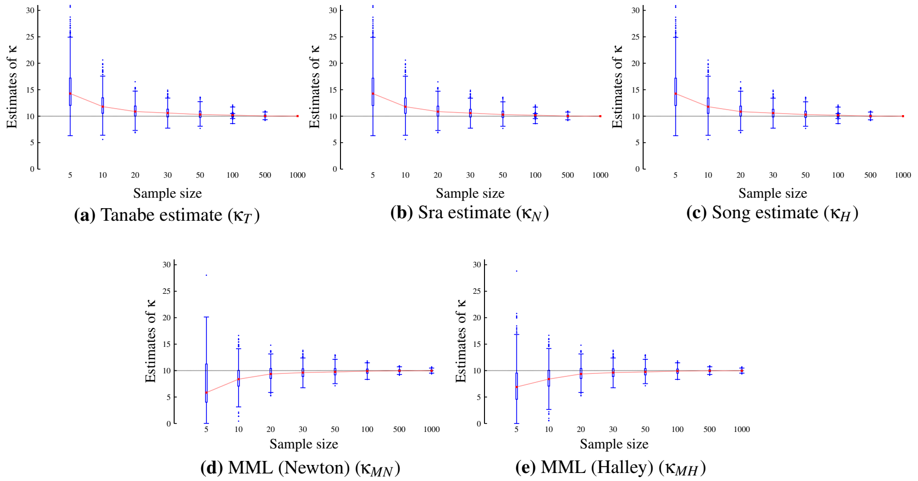
(N, d, κ)	Mean absolute error					Mean squared error				
	Tanabe	Sra	Song	MML		Tanabe	Sra	Song	MML	
	κ_T	κ_N	κ_H	κ_{MN}	κ_{MH}	κ_T	κ_N	κ_H	κ_{MN}	κ_{MH}
(a)										
10,10,10	2.501e+0	2.486e+0	2.486e+0	**2.008e+0**	2.012e+0	1.009e+1	9.984e+0	9.984e+0	**5.811e+0**	5.850e+0
10,10,100	1.879e+1	1.877e+1	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**

Table 1 continued

(N, d, κ)	Average KL-divergence					Average message length				
	Tanabe	Sra	Song	MML		Tanabe	Sra	Song	MML	
	κ_T	κ_N	κ_H	κ_{MN}	κ_{MH}	κ_T	κ_N	κ_H	κ_{MN}	κ_{MH}
(b)										
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**

Bold values indicate the best results with respect to the corresponding competitors

a Errors in κ estimation. The averages are reported over 1000 simulations for each (N, d, κ) triple. *b* Comparison of the κ estimates using KL-divergence and message length formulation (both metrics are measured in bits)



(a) Tanabe estimate (κ_T) **(b)** Sra estimate (κ_N) **(c)** Song estimate (κ_H)

(d) MML (Newton) (κ_{MN}) **(e)** MML (Halley) (κ_{MH})

Fig. 9 *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

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 mean *absolute* error can be used to quantitatively determine the bias due to the estimates. Since the absolute error for the MML estimates is lower compared to the others (see Table 1a), we conclude that the bias due to the MML estimates is lower. Further, the plots in Fig. 9 are also indicative of the deviation between the average κ estimate and the true κ value. Also, the mean *squared* error can be used to determine the combined effect of bias and variance of the estimators. The mean squared error is empirically demonstrated to be the least for the MML estimates.

KL-divergence and message lengths of the estimates: The quality of parameter inference is further determined by computing the KL-divergence and the message lengths associated with the κ estimates. The analytical expression to calculate the KL-divergence of any two vMF distributions is derived in the "Appendix". The KL-divergence 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 Eq. 13. Table 1b lists the average values of both the metrics. The MML estimates of κ result in the least value of KL-divergence across all combinations of N, d, κ. Also, the message lengths associated with the MML based estimates are the least. From Table 1b, we notice that when $N = 10$, κ_{MN} and κ_{MH} clearly have lower message lengths. For $N = 10, d = 10, \kappa = 10$, κ_{MN}, κ_{MH} result in extra compression of $\sim$1.5 bits over $\kappa_T, \kappa_N, \kappa_H$, which makes the MML estimates $2^{1.5}$ times more likely than the others (as per Eq. 10).

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). Here, we examine the behaviour of κ estimates for generic vMF distributions as proposed by Mardia et al. (1984). They derived a likelihood ratio test for the null hypothesis of a vMF distribution (H_0) against the alternative of a Fisher-Bingham distribution (H_a). The asymptotically equivalent Rao's score statistic was used to test the hypothesis.

The score statistic $\mathscr{W}$, in this case, is a function of the concentration parameter. It has an asymptotic $\chi^2(p)$ distribution (with degrees of freedom $p = d(d+1)/2 - 1$) under H_0 as the sample size $N \to \infty$. For $d = \{10, 100, 1000\}$, the critical values at 5% significance level are given in Table 2. If the computed test statistic exceeds the critical value, then the null hypothesis of a vMF distribution is rejected. We conduct a simulation study where we generate random samples of size $N = 1$ million from a vMF distribution with known mean and $\kappa = \{10, 100, 1000\}$. For each inferred estimate $\hat{\kappa}$, we compute the test statistic and compare it with the corresponding critical value. The results are shown in Table 2. For $d = 10$, the approximation κ_T 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 κ_T. However, this is incorrect as the data was generated from a vMF distribution. The p-values due to the estimates $\kappa_N, \kappa_H, \kappa_{MN}, \kappa_{MH}$ 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 of 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 MML estimates: Based on the empirical tests, we have so far seen that MML estimates fare better when compared to the other approximations. We now discuss the behaviour of the MML estimates in the limiting case. For large sample sizes, we plot the errors in κ estimation. Song et al. (2012) demonstrated that their approximation results in the least error in the limiting case. We compute the variation in error when $d = 1000$ and under two scenarios:

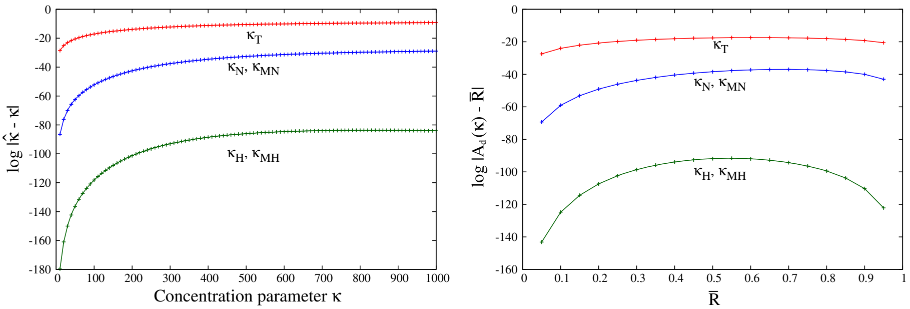
1. *Increasing* κ: Figure 10a illustrates the behaviour of the absolute error with increasing κ. The first observation is that irrespective of the estimation procedure, the error continues to increase with increasing κ values (which corroborates our observations in the empirical tests) and then saturates. According to Song et al. (2012), their estimate κ_H produces the lowest error which we can see in the figure. Further, our MML Newton approximation κ_{MN} actually performs worse than Song's approximation κ_H. However, we note that the errors due to MML Halley's approximation κ_{MH} are identical to those produced by κ_H. This suggests that asymptotically, the approximations achieved by κ_H and κ_{MH} 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 $F(\hat{\kappa}) \equiv A_d(\hat{\kappa}) - \bar{R} = 0$ (Eq. 6). Hence, $\log |A_d(\kappa) - \bar{R}|$ gives a measure of the error corresponding to an estimate of κ. Figure 10b depicts the variation of this error with increasing $\bar{R}$. We observe that κ_H and κ_{MH} produce the least error. We also note that the error produced due to κ_{MN} is greater than that produced by κ_{MH}. However, we highlight the fact that MML-based parameter inference aims to achieve $G(\hat{\kappa}) \equiv 0$ (Eq. 12), 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 κ_{MH} coincides with the ML estimate κ_H. However, κ_H's performance is better compared to the MML Newton's approximation κ_{MN}. The same behaviour is observed for when κ is fixed and the dimensionality is increased. For *enormous* amount of data, the ML approximations converge to the MML ones.

Table 2 Goodness-of-fit tests for the null hypothesis H_0: vMF distribution and the alternate hypothesis H_a: Fisher–Bingham distribution

(d, κ)	Critical	Test statistic					p value of the test				
	value	Tanabe	Sra	Song	MML		Tanabe	Sra	Song	MML	
	$\chi^2(p)$	κ_T	κ_N	κ_H	κ_{MN}	κ_{MH}	κ_T	κ_N	κ_H	κ_{MN}	κ_{MH}
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

Critical values of the test statistic correspond to a significance of 5 %



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

Fig. 10 Errors in κ estimation for $d = 1000$ as the sample size $N \to \infty$

10 Applications of vMF mixtures

10.1 Application to text clustering

We conducted empirical studies where the proposed search method is employed to infer vMF mixtures with varying amounts of data and using true vMF mixtures of varying difficulty levels. For these results, see the extended version (Kasarapu and Allison 2015). In this section, we focus on the applications of vMF mixtures. The use of vMF mixtures in modelling high dimensional text data has been investigated by Banerjee et al. (2005). To compute 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 (which compensates for different document lengths) motivates their modelling using vMF mixture distributions.

Banerjee et al. (2005) used their proposed approximation (see Eq. 7) to estimate the parameters of a mixture with *known* number of components. They did not, however, propose a method to search for the optimal number of mixture components. We not only derived MML estimates which fare better compared to the previous approximations but also apply them to devise a search method to infer the optimal mixtures. Ideally, the search is continued until there is no further improvement in the message length (see Algorithm 1). For practical 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 %. Our proposed method to infer mixtures was employed on the datasets that were used in the analysis by Banerjee et al. (2005). The parameters of the intermediate mixtures are estimated using the MML Halley's estimates (Eq. 15) for the component vMF distributions. Banerjee et al. (2005) use mutual information (MI) to assess the quality of clustering. For given cluster assignments X and the (known) class labels Y, MI is defined as: $E\left[\log\dfrac{\Pr(X,Y)}{\Pr(X)\Pr(Y)}\right]$. Along with the message lengths, we use MI as one other evaluation criterion in our analysis. We also compute the average F-measure when the number of clusters is same as the number of actual classes.

For each of the datasets, in the preprocessing step, we generate feature vectors using the most frequently occuring words and generating a TF-IDF score for each feature (word)

Table 3 Confusion matrix for 16-component assignment (using MML Halley estimate of κ)

	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

based on Okapi BM25 score (Robertson and Zaragoza 2009). 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 greedy search proposed in Sect. 7.

Classic3 dataset:[4] The documents are from three 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: In this example, it is known that there are three distinct categories. However, 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 3. A closer look at the generated assignments illustrate 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. We observe that all but three components are non-overlapping; only M6 has representative documents from all three categories.

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 EM algorithm (Sects. 5.1,5.2) where the components are updated using the respective estimates. The 3-component mixtures inferred using the different estimates perform comparably with each other; there is not much difference in the assignments of data to the individual mixture components. 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 4a). For the 3-component mixture, all the five different estimates result in high F-measure values with Song being the best with an average F-measure of 0.978 and a MI of 0.982. MML (Halley's) estimate are close with an average F-measure of 0.976 and a 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.

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

[4] http://www.dataminingresearch.com/index.php/2010/09/classic3-classic4-datasets/.

Table 4 Clustering performance on the two datasets (a) Classic3 (b) CMU_Newsgroup

Number of clusters	Evaluation metric	Banerjee	Tanabe	Sra	Song	MML (Halley)
(a)						
3	Message length	100,678,069	100,677,085	100,677,087	100,677,080	**100,676,891**
	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	100,458,153	100,452,893	100,439,983	100,444,649	**100,427,178**
	Mutual information	1.029	1.036	0.978	**1.043**	1.040
(b)						
20	Message length	728,666,702	728,545,471	728,585,441	728,536,451	**728,523,254**
	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	728,497,453	728,498,076	728,432,625	728,374,429	**728,273,820**
	Mutual information	1.313	1.229	**1.396**	1.377	1.375

Bold values indicate the best results with respect to the corresponding competitors

and combine them appropriately". However, this is subjective and requires some background information about the likely number of clusters. Our 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:[5] The dataset is a collection of 20 different news categories each containing 1000 documents. Preprocessing of the data 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 4b. 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 MML case is 1.379 which is lower than that of Sra's. However, the total message length is lower for MML mixture compared to that of others.

Optimal number of clusters: The search method when applied to this dataset infers a 21-component mixture. 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 the MML mixture, it is 1.375 (see Table 4b). However, the net message length is the least for the MML mixture.

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 MI evaluation metric for different number of mixture components does not aid in the inference of an optimal mixture model. Our search method balances the tradeoff 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.

A mixture modelling problem of this kind where there is some information available regarding the nature of the data can be studied by altering the proposed search method.

[5] http://archive.ics.uci.edu/ml/datasets/Twenty+Newsgroups.

We provide some alternate strategies where the mixture modelling can be done in a semi-supervised setting.

- 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 that the inferred mixture would have 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. We cannot ignore splits completely because a component after splitting may be merged at a later iteration if there would be an improvement to the message length.

10.2 Mixture modelling of protein coordinate data

The following application concerns the vMF mixture modelling of directional data arising from the orientation of main chain carbon atoms in protein structures. The structures that proteins adopt are largely dictated by the interactions between the constituent atoms. These chemical interactions impose constraints on the orientation of atoms with respect to one another. The directional nature of the protein data and the (almost constant) bond length between the main chain carbon atoms motivate modelling using vMF mixtures. Further, 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 require efficient encoding of protein data (Konagurthu et al. 2012, 2013; Collier et al. 2014). As part of our results, we demonstrate that vMF mixtures offer a better means of encoding and can potentially serve as strong candidate models to be used in such varied tasks.

The dataset considered here is a collection of 8453 non-redundant experimentally determined protein structures from the publicly available ASTRAL SCOP-40 (version 1.75) database (Murzin et al. 1995). For each protein structure, the coordinates of the central carbon, C_α, of successive residues (amino acids) are considered. Protein coordinate data is transformed into directional data and each direction vector is characterized by $(\theta, \phi) =$ (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 $i + 1$, we consider this point in the context of 3 preceding points, forming a 4-mer comprising of the points P_{i-2}, P_{i-1}, P_i, and P_{i+1}. This 4-mer is orthogonally transformed into a canonical orientation in the following steps: (1) translate the 4-mer such that P_i is at the origin, (2) rotate the resultant 4-mer so that P_{i-1} lies on the negative X-axis, and (3) rotate further so that P_{i-2} lies in the XY plane such that the angle between the vector $\mathbf{P_{i-2}} - \mathbf{P_{i-1}}$ and the positive Y-axis is acute. The transformation yields a canonical orientation for P_{i+1} with respect to its previous 3 coordinates. Using the transformed coordinates of P_{i+1}, the direction (θ, ϕ) of P_{i+1} is computed. We repeat this transformation for every successive set of 4-mers in the protein chain, over all possible source structures in our collection. The data collected in this way resulted in a total of $\sim$1.3 million (θ, ϕ) pairs for all the 8453 structures in the database.

Protein data is an example where the number of mixture components are not known a priori. Hence, we use the method outlined in Sect. 7 to infer suitable mixture models. The original dataset comprises of 7 different categories of proteins. The proposed search method using

Table 5 Message lengths (in bits) for the inferred protein mixtures using various methods

Category	Tanabe	Sra	Song	MML (Halley)
β	5,514,800	5,518,679	5,520,073	**5,513,507**
All	27,818,524	27,833,704	27,839,802	**27,803,427**

Bold values indicate the best results with respect to the corresponding competitors
'All' refers to all the 7 protein categories

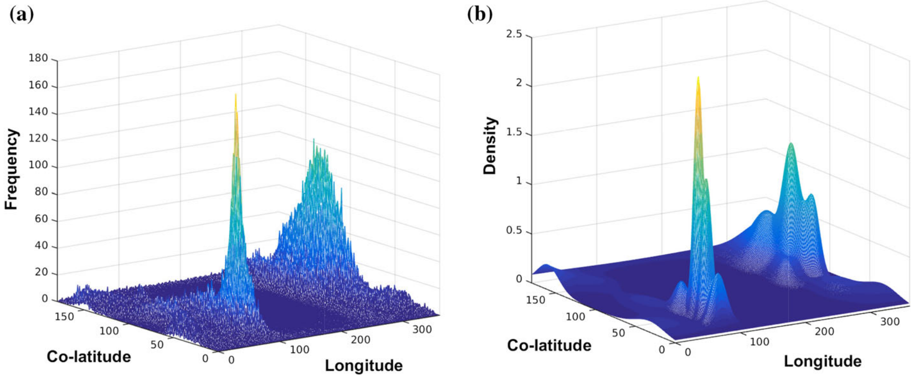


Fig. 11 Distribution of data of β class C_α atoms. **a** Empirical, **b** mixture density corresponding to the 11 inferred vMF components

MML (Halley's) estimates infers a mixture containing 13 vMF components. Further, each protein category can be individually modelled using a mixture. As an example, for the "β class" proteins which contains 1802 protein structures and 251,346 corresponding (θ, ϕ) pairs, our search method terminates after inferring 11 vMF components. We compare the MML-based mixtures with those inferred by the standalone EM algorithm (Sect. 5.2) using other estimates. These values are presented in Table 5. We observe that the mixtures inferred using the MML estimates result in a message length lower than that obtained using the other estimates.

The empirical distribution of directional data i.e., the (θ, ϕ) values corresponding to the C_α coordinates belonging to β class are plotted in Fig. 11a. Figure 11b is a plot illustrating the 11-component vMF mixture density as inferred for this class of proteins. Notice that the two major modes in the figure correspond to commonly observed local secondary structural bias of residues towards, helices and strands of sheet. Also notice the empty region in the middle which corresponds to physically unrealizable directions in the local chain, excluded in the observed samples due to steric hindrance of the atoms in proteins. If we were to model such data using truncated distributions, the regions of zero probability will be modelled using an infinite code length. As an example, at $(\theta, \phi) = (100°, 200°)$, the truncated distribution would have zero probability and consequently an *infinite* code length. However, when the same point is explained using the 11-component mixture, it would have a probability of $\Pr = 3.36 \times 10^{-12}$ and a corresponding code length of $-\log_2 \Pr = 38.11$ bits. For protein data, it is possible to have such (rare exceptional) observations, due to reasons such as experimental error, noise, or the conformation of the protein itself. Hence, although the empirical distribution has distinct modes, it is better off modelled as a vMF mixture distribution, rather than by truncated distributions.

Table 6 Comparison of the uniform and vMF null model encoding schemes

Null model	Total message length (in bits)	Bits per residue
Uniform	36,119,900	27.437
vMF	**32,869,700**	**24.968**

Bold values indicate the best results with respect to the corresponding competitors

Compressibility of protein structures: The explanatory framework of MML allows for testing competing hypotheses. Recently, Konagurthu et al. (2012) developed a null model description of protein coordinate data as part of the statistical inference of protein secondary structure. A null model gives a baseline for transmitting the raw coordinate data. Each C_α atom is described using the distance and orientation with respect to the preceding C_α atoms. Because the distance between successive C_α atoms is highly constrained, compression can only be gained in describing the orientation of a C_α atom with respect to its previous one. Konagurthu et al. (2012) describe their null hypothesis by discretizing the surface of a 3D-sphere into chunks of equal areas (of ϵ^2, where ϵ is the accuracy of measurement of coordinate data). This results in $4\pi r^2/\epsilon^2$ cells distributed uniformly on the surface of the sphere of radius r (the distance between successive C_α coordinates). To encode C_α^{i+1} with respect to C_α^i, the location of C_α^{i+1} on the surface is identified and its cell index is encoded. Using this description, the stated null model results in a message length given by $-\log_2(\epsilon^2/4\pi r^2)$ bits.

The null model of Konagurthu et al. (2012) assumes a uniform distribution of on the surface of the sphere. However, this is a crude assumption and one can leverage the directional properties of protein coordinates to build an efficient null model. To this effect, we explore the use of vMF mixtures as null model descriptors for protein structures. The message length expression to encode the orientation angles using vMF mixtures is then given by Eq. 25, where $\mathbf{x}$ corresponds to a unit vector described by (θ, ϕ) on the surface of the sphere. The uniform and vMF mixtures are two competing null models. These are used to encode the directional data corresponding to the 8453 protein structures in the ASTRAL SCOP-40 database. The results are shown in Table 6. The per residue statistic is calculated by dividing the total message length by the sample size (the number of (θ, ϕ) pairs). This statistic shows that close to 2.5 bits can be saved (on average) if the protein data is encoded using the vMF null model. The vMF null model thus supercedes the naive model of encoding. This can potentially improve the accuracy of statistical inference that is central to the various protein modelling tasks briefly introduced above.

$$\text{vMF Null} = -\log_2\left(\sum_{j=1}^{M} w_j f_j(\mathbf{x};\Theta_j)\right) - 2\log_2\left(\frac{\epsilon}{r}\right) \quad \text{bits.} \tag{25}$$

11 Conclusion

We presented a statistically robust approach for inferring mixtures of (1) multivariate Gaussian and (2) von Mises-Fisher distributions. It is based on the general information-theoretic framework of minimum message length inference. This provides an objective tradeoff between the hypothesis complexity and the quality of fit to the data. We also provide a new search procedure for inferring an optimal mixture model that chooses the number of component distributions and their respective parameters by minimizing the total message length. We demonstrated that our proposed search algorithm performs better by comparing with the popularly used search method of Figueiredo and Jain (2002). We demonstrated the effectiveness of our approach through extensive experimentation and validation of our results. We also applied our method to real-world high dimensional text data and to directional data that arises from protein chain conformations. An extended version of the paper is available at http://arxiv.org/abs/1502.07813.

Acknowledgments The authors would like to thank Arun Konagurthu for discussions pertaining to protein data clustering, and Maria Garcia de la Banda for stimulating discussions and providing interesting insights. The authors would also like to acknowledge Wray Buntine's inputs with regard to text clustering. Parthan Kasarapu thanks NICTA for providing his PhD scholarship. NICTA is funded by the Australian Government through the Department of Communications and the Australian Research Council through the ICT Centre of Excellence Program.

12 Appendix

Supporting derivations required for evaluating κ_{MN} and κ_{MH}: For brevity, we represent $A_d(\kappa)$, $A'_d(\kappa)$, $A''_d(\kappa)$, and $A'''_d(\kappa)$ as A, A', A'', and A''' respectively. Expressions to evaluate A, A', and A'' are given in Eqs. (6), (8), and (9) respectively. We require A''' for its use in the remainder of the derivation and we provide its expression below:

$$\frac{A'''}{A'} = -\frac{2AA''}{A'} - 2A' - \frac{(d-1)}{\kappa}\frac{A''}{A'} - \frac{2(d-1)}{\kappa^3}\frac{A}{A'} + \frac{2(d-1)}{\kappa^2} \tag{26}$$

We now discuss the derivation of $G'(\kappa)$ and $G''(\kappa)$ that are required for computing the MML estimates κ_{MN} and κ_{MH} (Eqs. 14 and 15). On differentiating Eq. 12, we get

$$G'(\kappa) = \frac{(d-1)}{2\kappa^2} + (d+1)\frac{(1-\kappa^2)}{(1+\kappa^2)^2} + \frac{(d-1)}{2}\frac{\partial}{\partial\kappa}\left(\frac{A'}{A}\right) + \frac{1}{2}\frac{\partial}{\partial\kappa}\left(\frac{A''}{A'}\right) + nA'$$

$$\text{and}\quad G''(\kappa) = -\frac{(d-1)}{\kappa^3} + (d+1)\frac{2\kappa(\kappa^2-3)}{(1+\kappa^2)^3} + \frac{(d-1)}{2}\frac{\partial^2}{\partial\kappa^2}\left(\frac{A'}{A}\right) + \frac{1}{2}\frac{\partial^2}{\partial\kappa^2}\left(\frac{A''}{A'}\right) + nA''$$

Using Eqs. 6 and 8, we have

$$\frac{A'}{A} = \frac{1}{A} - A - \frac{(d-1)}{\kappa} \quad \text{and} \quad \frac{\partial}{\partial\kappa}\left(\frac{A'}{A}\right) = -\frac{A'}{A^2} - A' + \frac{(d-1)}{\kappa^2}$$

$$\frac{\partial^2}{\partial\kappa^2}\left(\frac{A'}{A}\right) = 2\frac{(A')^2}{A^3} - \frac{A''}{A^2} - A'' - \frac{2(d-1)}{\kappa^3} \tag{27}$$

Using Eqs. 6, 8, 9, and 26, we have

$$\begin{aligned}
\frac{A''}{A'} &= -2A - \frac{(d-1)}{\kappa} + \frac{(d-1)}{\kappa^2}\frac{A}{A'} \quad \text{and}\\
\frac{\partial}{\partial\kappa}\left(\frac{A''}{A'}\right) &= -2A' + \frac{2(d-1)}{\kappa^2} - \frac{(d-1)}{\kappa^3}\frac{A}{A'}\left(\frac{\kappa A''}{A'}+2\right)\\
\frac{\partial^2}{\partial\kappa^2}\left(\frac{A''}{A'}\right) &= -2A'' - \frac{4(d-1)}{\kappa^3} - (d-1)\frac{\partial}{\partial\kappa}\left(\frac{AA''}{\kappa^2 A'^2}\right)\\
&\quad -2(d-1)\frac{\partial}{\partial\kappa}\left(\frac{A}{\kappa^3 A'}\right)\\
\text{where}\quad \frac{\partial}{\partial\kappa}\left(\frac{AA''}{\kappa^2 A'^2}\right) &= \frac{\kappa AA'A''' + \kappa A'^2A'' - 2\kappa AA''^2 - 2AA'A''}{\kappa^3 A'^3} \quad \text{and}\\
\frac{\partial}{\partial\kappa}\left(\frac{A}{\kappa^3 A'}\right) &= \frac{1}{\kappa^3} - \frac{A}{\kappa^4 A'^2}(\kappa A'' + 3A') \qquad (28)
\end{aligned}$$

Equations 27) and 28 are used to evaluate $G'(\kappa)$ and $G''(\kappa)$ which can then be used to approximate κ_{MN} and κ_{MH}.

Derivation of the Kullback–Leibler (KL) distance between two von Mises-Fisher distributions: The closed form expression to calculate the KL divergence between two vMF distributions is presented below. Let $f(\mathbf{x}) = C_d(\kappa_1)e^{\kappa_1 \boldsymbol{\mu_1}^T\mathbf{x}}$ and $g(\mathbf{x}) = C_d(\kappa_2)e^{\kappa_2 \boldsymbol{\mu_2}^T\mathbf{x}}$ be two von Mises-Fisher distributions with mean directions $\boldsymbol{\mu_1}, \boldsymbol{\mu_2}$ and concentration parameters κ_1, κ_2. The KL distance between any two distributions is given by Eq. 29, where $\mathrm{E}_f[.]$ is the expectation of the quantity [.] using the probability density function f.

$$D_{KL}(f||g) = \int_{\mathbf{x}} f(\mathbf{x}) \log\frac{f(\mathbf{x})}{g(\mathbf{x})} d\mathbf{x} = \mathrm{E}_f\left[\log\frac{f(\mathbf{x})}{g(\mathbf{x})}\right] \qquad (29)$$

For two vMF distributions f and g, we have $\log\frac{f(\mathbf{x})}{g(\mathbf{x})} = \log\frac{C_d(\kappa_1)}{C_d(\kappa_2)} + (\kappa_1\boldsymbol{\mu_1} - \kappa_2\boldsymbol{\mu_2})^T\mathbf{x}$. Using the fact that $\mathrm{E}_f[\mathbf{x}] = A_d(\kappa_1)\boldsymbol{\mu_1}$ (Mardia et al. 1984; Fisher 1993), we have the following expression for KL-divergence:

$$\begin{aligned}
\mathrm{E}_f\left[\log\frac{f(\mathbf{x})}{g(\mathbf{x})}\right] &= \log\frac{C_d(\kappa_1)}{C_d(\kappa_2)} + (\kappa_1\boldsymbol{\mu_1} - \kappa_2\boldsymbol{\mu_2})^T A_d(\kappa_1)\boldsymbol{\mu_1}\\
D_{KL}(f||g) &= \log\frac{C_d(\kappa_1)}{C_d(\kappa_2)} + A_d(\kappa_1)(\kappa_1 - \kappa_2\boldsymbol{\mu_1}^T\boldsymbol{\mu_2}) \qquad (30)
\end{aligned}$$

References

Agusta, Y., & Dowe, D. L. (2003). Unsupervised learning of correlated multivariate Gaussian mixture models using MML. In *AI 2003: advances in artificial intelligence* (pp. 477–489). Berlin: Springer.

Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, *19*(6), 716–723.

Anderson, E. (1935). The Irises of the Gaspé Peninsula. *Bulletin of the American Iris Society*, *59*, 2–5.

Banerjee, A., Dhillon, I., Ghosh, J., & Sra, S. (2003). Generative model-based clustering of directional data. *Proceedings of the 9th international conference on knowledge discovery and data mining* (pp. 19–28). New York: ACM.

Banerjee, A., Dhillon, I. S., Ghosh, J., & Sra, S. (2005). Clustering on the unit hypersphere using von Mises-Fisher distributions. *Journal of Machine Learning Research*, *6*, 1345–1382.

Barton, D. E. (1961). Unbiased estimation of a set of probabilities. *Biometrika*, *48*(1–2), 227–229.

Basu, A. P. (1964). Estimates of reliability for some distributions useful in life testing. *Technometrics*, *6*(2), 215–219.

Best, D., & Fisher, N. (1981). The bias of the maximum likelihood estimators of the von Mises-Fisher concentration parameters. *Communications in Statistics-Simulation and Computation*, *10*(5), 493–502.

Biernacki, C., Celeux, G., & Govaert, G. (2000). Assessing a mixture model for clustering with the integrated completed likelihood. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *22*(7), 719–725.

Bishop, C. M. (2006). *Pattern recognition and machine learning* (Vol. 1). New York: Springer.

Boulton, D., & Wallace, C. (1969). The information content of a multistate distribution. *Journal of Theoretical Biology*, *23*, 269–278.

Bozdogan, H. (1990). 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.

Bozdogan, H. (1993). *Choosing the number of component clusters in the mixture-model using a new informational complexity criterion of the inverse-Fisher information matrix*. Berlin: Springer.

Collier, J. H., Allison, L., Lesk, A. M., de la Banda, M. G., & Konagurthu, A. S. (2014). A new statistical framework to assess structural alignment quality using information compression. *Bioinformatics*, *30*(17), i512–i518.

Conway, J. H., & Sloane, N. J. A. (1984). On the Voronoi regions of certain lattices. *SIAM Journal on Algebraic and Discrete Methods*, *5*, 294–305.

Cordeiro, G. M., & Vasconcellos, K. L. (1999). 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.

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, *39*(1), 1–38.

Dowe, D. L., Allison, L., Dix, T. I., Hunter, L., Wallace, C. S., & Edgoose, T. (1996a). Circular clustering of protein dihedral angles by minimum message length. In *Pacific symposium on biocomputing*, Vol. 96, pp. 242–255.

Dowe, D. L., Oliver, J. J., Baxter, R. A., & Wallace, C. S. (1996b). Bayesian estimation of the von Mises concentration parameter. In *Maximum entropy and Bayesian methods* pp. 51–60. The Netherlands: Springer.

Dowe, D. L., Oliver, J. J., & Wallace, C. S. (1996c). MML estimation of the parameters of the spherical Fisher distribution. In *Algorithmic learning theory* (pp. 213–227). Berlin: Springer.

Dwyer, P. S. (1967). Some applications of matrix derivatives in multivariate analysis. *Journal of the American Statistical Association*, *62*(318), 607–625.

Eaton, M. L., & Morris, C. N. (1970). The application of invariance to unbiased estimation. *The Annals of Mathematical Statistics*, *41*(5), 1708–1716.

Figueiredo, M. A., & Jain, A. K. (2002). Unsupervised learning of finite mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *24*(3), 381–396.

Fisher, N. I. (1993). *Statistical analysis of spherical data*. Cambridge: Cambridge University Press.

Fisher, R. (1953). Dispersion on a sphere. *Proceedings of the Royal Society of London: Series A (Mathematical and Physical Sciences)*, *217*(1130), 295–305.

Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, *7*(2), 179–188.

Gauvain, J., & Lee, C. H. (1994). Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains. *IEEE Transactions on Speech and Audio Processing*, *2*(2), 291–298.

Gray, G. (1994). Bias in misspecified mixtures. *Biometrics*, *50*(2), 457–470.

Jeffreys, H. (1946). 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.

Jones, P., & McLachlan, G. (1990). Laplace-normal mixtures fitted to wind shear data. *Journal of Applied Statistics*, *17*(2), 271–276.

Jorgensen, M. A., & McLachlan, G. J. (2008). Wallace's approach to unsupervised learning: The Snob program. *The Computer Journal*, *51*(5), 571–578.

Kasarapu, P., & Allison, L. (2015). *Minimum message length estimation of mixtures of multivariate gaussian and von Mises-Fisher distributions*. arXiv:1502.07813[cs.LG].

Kent, J. T. (1982). The Fisher–Bingham distribution on the sphere. *Journal of the Royal Statistical Society: Series B (Methodological)*, *44*(1), 71–80.

Konagurthu, A. S., Lesk, A. M., & Allison, L. (2012). Minimum message length inference of secondary structure from protein coordinate data. *Bioinformatics*, *28*(12), i97–i105.

Konagurthu, A. S., Allison, L., Abramson, D., Stuckey, P. J., & Lesk, A. M. (2013). Statistical inference of protein "LEGO bricks". In *2013 IEEE 13th international conference on data mining (ICDM)*, IEEE, (pp 1091–1096).

Krishnan, T., & McLachlan, G. (1997). *The EM algorithm and extensions*. New York: Wiley.

Kullback, S., & Leibler, R. A. (1951). On information and sufficiency. *The Annals of Mathematical Statistics*, *22*(1), 79–86.

Lee, P. (1997). *Bayesian statistics: An introduction*. London: Arnold.

Lo, Y. (2011). Bias from misspecification of the component variances in a normal mixture. *Computational Statistics and Data Anaysis*, *55*(9), 2739–2747.

Magnus, J. R., & Neudecker, H. (1988). *Matrix differential calculus with applications in statistics and econometrics*. New York: Wiley.

Mardia, K., & Jupp, P. (2000). *Directional statistics*. Hoboken, NJ: Wiley.

Mardia, K., Holmes, D., & Kent, J. (1984). A goodness-of-fit test for the von Mises-Fisher distribution. *Journal of the Royal Statistical Society: Series B (Methodological)*, *46*(1), 72–78.

Mardia, K. V., Kent, J. T., & Bibby, J. M. (1979). *Multivariate analysis*. London: Academic Press.

Mardia, K. V., Taylor, C. C., & Subramaniam, G. K. (2007). Protein bioinformatics and mixtures of bivariate von Mises distributions for angular data. *Biometrics*, *63*(2), 505–512.

McLachlan, G., & Peel, D. (1997). Contribution to the discussion of paper by S. Richardson and P.J. Green. *Journal of the Royal Statistical Society B*, *59*, 779–780.

McLachlan, G., & Peel, D. (2000). *Finite mixture models*. New York: Wiley.

McLachlan, G. J., & Basford, K. E. (1988). *Mixture models: Inference and applications to clustering (Statistics: Textbooks and Monographs)*. New York: Dekker.

Murzin, A., Brenner, S., Hubbard, T., Chothia, C., et al. (1995). SCOP: A structural classification of proteins database for the investigation of sequences and structures. *Journal of Molecular Biology*, *247*(4), 536–540.

Oliver, J., & Baxter, R. (1994). *MML and Bayesianism: Similarities and differences*. Dept Comput Sci Monash Univ, Clayton, Victoria, Australia, Tech Rep 206.

Oliver, J. J., Baxter, R. A., & Wallace, C. S. (1996). Unsupervised learning using MML. In *Machine learning: Proceedings of the 13th international conference*, (pp. 364–372).

Patra, K., & Dey, D. K. (1999). A multivariate mixture of Weibull distributions in reliability modeling. *Statistics & Probability letters*, *45*(3), 225–235.

Peel, D., & McLachlan, G. J. (2000). Robust mixture modelling using the t-distribution. *Statistics and Computing*, *10*(4), 339–348.

Peel, D., Whiten, W. J., & McLachlan, G. J. (2001). Fitting mixtures of Kent distributions to aid in joint set identification. *Journal of the American Statistical Association*, *96*(453), 56–63.

Richardson, S., & Green, P. J. (1997). On Bayesian analysis of mixtures with an unknown number of components. *Journal of the Royal Statistical Society: Series B (Methodological)*, *59*(4), 731–792.

Rissanen, J. (1978). Modeling by shortest data description. *Automatica*, *14*(5), 465–471.

Rissanen, J. (1989). *Stochastic complexity in statistical inquiry theory*. River Edge, NJ: World Scientific Publishing Co. Inc.

Roberts, S., Husmeier, D., Rezek, I., & Penny, W. (1998). Bayesian approaches to Gaussian mixture modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *20*(11), 1133–1142.

Robertson, S., & Zaragoza, H. (2009). *The probabilistic relevance framework: BM25 and beyond*. Hanover, MA: Now Publishers Inc.

Salton, G., & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, *24*(5), 513–523.

Salton, G., & McGill, M. J. (1986). *Introduction to modern information retrieval*. New York, NY: McGraw-Hill Inc.

Schou, G. (1978). Estimation of the concentration parameter in von Mises-Fisher distributions. *Biometrika*, *65*(2), 369–377.

Schwarz, G., et al. (1978). Estimating the dimension of a model. *The Annals of Statistics*, *6*(2), 461–464.

Seidel, W., Mosler, K., & Alker, M. (2000). A cautionary note on likelihood ratio tests in mixture models. *Annals of the Institute of Statistical Mathematics*, *52*(3), 481–487.

Shannon, C. E. (1948). A mathematical theory of communication. *The Bell System Technical Journal*, *27*, 379–423.

Song, H., Liu, J., & Wang, G. (2012). High-order parameter approximation for von Mises-Fisher distributions. *Applied Mathematics and Computation*, *218*(24), 11,880–11,890.

Sra, S. (2012). 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.

Strehl, A., Ghosh, J., & Mooney, R. (2000). Impact of similarity measures on web-page clustering. In *Workshop on artificial intelligence for web search (AAAI 2000)*, (pp. 58–64).

Tanabe, A., Fukumizu, K., Oba, S., Takenouchi, T., & Ishii, S. (2007). Parameter estimation for von Mises-Fisher distributions. *Computational Statistics*, *22*(1), 145–157.

Titterington, D. M., Smith, A. F., Makov, U. E., et al. (1985). *Statistical analysis of finite mixture distributions*. New York: Wiley.

Wallace, C. (1986). An improved program for classification. In *Proceedings of the 9th Australian computer science conference*, (pp. 357–366).

Wallace, C., & Dowe, D. (1994). Estimation of the von Mises concentration parameter using minimum message length. In *Proceedings of the 12th Australian statistical society conference*, Monash University, Australia.

Wallace, C. S. (2005). *Statistical and inductive inference using minimum message length. Information Science and Statistics*. Secaucus, NJ: Springer.

Wallace, C. S., & Boulton, D. M. (1968). An information measure for classification. *Computer Journal*, *11*(2), 185–194.

Wallace, C. S., & Dowe, D. L. (1999). Minimum message length and Kolmogorov complexity. *Computer Journal*, *42*, 270–283.

Wallace, C. S., & Freeman, P. R. (1987). Estimation and inference by compact coding. *Journal of the Royal Statistical Society: Series B (Methodological)*, *49*(3), 240–265.

Wang, P., Puterman, M. L., Cockburn, I., & Le, N. (1996). Mixed Poisson regression models with covariate dependent rates. *Biometrics*, *52*(2), 381–400.

Watson, G., & Williams, E. (1956). On the construction of significance tests on the circle and the sphere. *Biometrika*, *43*(3–4), 344–352.

White, H. (1982). Maximum likelihood estimation of misspecified models. *Econometrica*, *50*(1), 1–25.

Wood, A. T. (1994). Simulation of the von Mises Fisher distribution. *Communications in Statistics-Simulation and Computation*, *23*(1), 157–164.

Xu, L., & Jordan, M. I. (1996). On convergence properties of the EM algorithm for Gaussian mixtures. *Neural Computation*, *8*(1), 129–151.

Zhong, S., & Ghosh, J. (2003). A comparative study of generative models for document clustering. In *Proceedings of the workshop on clustering high dimensional data and its applications in SIAM data mining conference*.