Modeling Data using Directional Distributions

Inderjit S. Dhillon and Suvrit Sra
Department of Computer Sciences
The University of Texas at Austin
Austin, TX 78712
{suvrit,inderjit}@cs.utexas.edu

25 January, 2003*

Technical Report # TR-03-06

Abstract

Traditionally multi-variate normal distributions have been the staple of data modeling in most domains. For some domains, the model they provide is either inadequate or incorrect because of the disregard for the directional components of the data. We present a generative model for data that is suitable for modeling directional data (as can arise in text and gene expression clustering). We use mixtures of *von Mises-Fisher* distributions to model our data since the von Mises-Fisher distribution is the *natural* distribution for directional data. We derive an Expectation Maximization (EM) algorithm to find the maximum likelihood estimates for the parameters of our mixture model, and provide various experimental results to evaluate the "correctness" of our formulation. In this paper we also provide some of the mathematical background necessary to carry out all the derivations and to gain insight for an implementation.

1 Introduction

Traditional statistical approaches involve multi-variate data drawn from $\mathbb{R}^p$, and little or no significance is attached to the directional nature of the observed data. For many phenomena or processes it makes more sense to consider the directional components of the data involved, rather than just the magnitude alone. For example, modeling wind current directions, modeling geomagnetism, and measurements derived from clocks and compasses all seem to require a directional model [MJ00]. A much wider array of fields and contexts in which directional data arises is enlisted in [MJ00], and the interested reader is urged to atleast gloss over that information.

A fundamental distribution on the circle called the *von Mises* distribution was first introduced by von Mises [vM18]. We address the issue of modeling data using the von Mises-Fisher (vMF) distribution [MJ00], which is a generalization (to higher dimensions) of the von Mises distribution. We concentrate on using the vMF distribution as it is a distribution that arises naturally for directional data—akin to the multivariate Normal distribution ([MJ00, pp. 171-172]). Furthermore, it has been observed that in high dimensional text data, cosine similarity performs much better than a Euclidean distance metric[1] [DFG01]. This observation suggests following a directional model for the text data rather than ascribing significance to a magnitude based (or traditional) model.

*Revised 7th June, 2003.

[1]Empirically cosine similarity has been observed to outperform Euclidean or Mahalanobis type distance measures in information retrieval tasks.

Another application domain for the vMF model is modeling gene micro-array data (gene expression data). Gene expression data has been found to have unique directional characteristics that suggest the use of a directional model for modeling it. Recently Dhillon et. al (see [DMR03]) have found that gene expression data yields interesting diametric clusters. Intuitively these clusters could be thought of as data pointing in opposite directions, hinting at the underlying importance of directional orientation[2].

For text data, one byproduct of using a generative model like a mixture of vMF distributions, is the ability to obtain a soft-clustering of the data. The need for soft-clustering comes to the foreground when the text collections to be clustered can have documents with multiple labels. A more accurate generative model can also serve as an aid for improved classification for text data, especially where more meaningful soft labels are desired[3].

Organization of this report

The remainder of this report is organized as follows. Section 2 presents the multi-variate von Mises-Fisher distribution. Section 3 carries out the maximum likelihood estimation of parameters for data drawn from a single vMF distribution. Section 4 derives and presents the EM algorithm for estimating parameters for data drawn from a mixture of vMFs. In section 5 we show the results of experimentation with simulated mixtures of vMF distributions. Section 6 concludes this report. Some useful mathematical details are furnished by Appendices A and B. Appendix A provides mathematical background that is useful in general for understanding the derivations and Appendix B offers a brief primer on directional distributions.

2 The von Mises-Fisher Distribution

A p-dimensional unit random vector $\mathbf{x}$ ($\|\mathbf{x}\| = 1$) is said to have p-variate *von Mises-Fisher* distribution $M_p(\boldsymbol{\mu}, \kappa)$ if its probability density is:

$$c_p(\kappa) e^{\kappa \boldsymbol{\mu}^T \mathbf{x}}, \quad \mathbf{x} \in S^{p-1}, \tag{2.1}$$

where $\|\boldsymbol{\mu}\| = 1$, $\kappa \geq 0$, S^{p-1} is the p dimensional unit hypersphere (also denoted as S_p in some literature), and $c_p(\kappa)$ the normalizing constant is given by (see B.2)

$$c_p(\kappa) = \frac{\kappa^{p/2-1}}{(2\pi)^{p/2} I_{p/2-1}(\kappa)}. \tag{2.2}$$

For more details the interested reader is urged to look at Appendix B.

2.1 Example vMF distribution

In two dimensions (on the circle S^0), the probability density assumes the form

$$g(\theta; \mu, \kappa) = \frac{1}{2\pi I_0(\kappa)} e^{\kappa \cos(\theta - \mu)}, \quad 0 \leq \theta < 2\pi. \tag{2.3}$$

This is called the *von Mises* distribution. Figure 1 shows a plot of this density with mean at 0 radians and for $\kappa \in \{0, 0.3, 1, 4, 20\}$.

From the figure we can see that as κ increases the density becomes more and more concentrated about the mean direction. Thus κ is called the concentration parameter.

[2]Most clustering algorithms for gene expression data use Pearson correlation, which equals cosine similarity of transformed vectors, and thus our directional model should fit it well.

[3]Though given the nature of high-dimensional sparse data and models based on some member of the exponential family of distributions, the ability to obtain useful soft-label remains difficult without explicitly imposing "softness" constraints.

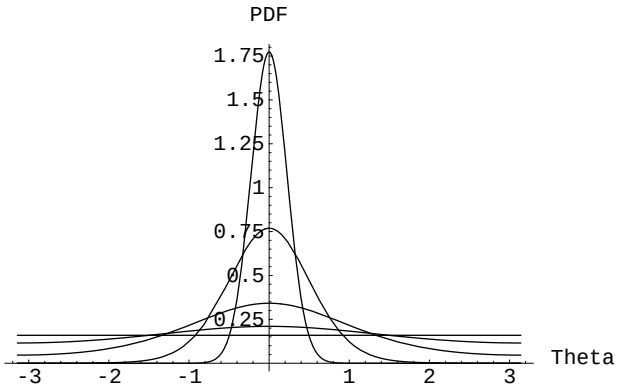

Figure 1: von Mises distribution for various κ ($\kappa = 0, 0.3, 1, 4, 20$).

3 Maximum Likelihood Estimates for von Mises-Fisher Distributions

Let $\mathscr{D} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ be the set of sample unit vectors following $M_p(\boldsymbol{\mu}, \kappa)$. Since each $\mathbf{x}_i \in \mathscr{D}$ is assumed to be independent the likelihood is

$$p(\mathbf{x}_1, \ldots, \mathbf{x}_n | \kappa, \boldsymbol{\mu}) = \prod_{i=1}^{n} c_p(\kappa) e^{\kappa \boldsymbol{\mu}^T \mathbf{x}_i}. \tag{3.1}$$

Thus the log-likelihood is

$$\mathscr{L}(\kappa, \boldsymbol{\mu}) = n \ln c_p(\kappa) + \kappa \boldsymbol{\mu}^T \mathbf{r}, \tag{3.2}$$

where $\mathbf{r} = \sum_i \mathbf{x}_i$ is the resultant vector. Differentiating (3.2) w.r.t $\boldsymbol{\mu}$ subject to the constraint $\boldsymbol{\mu}^T \boldsymbol{\mu} = 1$ we get

$$\kappa \mathbf{r} = 2\lambda \boldsymbol{\mu}, \tag{3.3}$$

where λ is a Lagrange multiplier. Let $\hat{\boldsymbol{\mu}}$ denote the m.l.e. for $\boldsymbol{\mu}$. From (3.3) and the fact that $\hat{\boldsymbol{\mu}}^T \hat{\boldsymbol{\mu}} = 1$ we conclude that $\hat{\boldsymbol{\mu}} = \mathbf{r}/\|\mathbf{r}\|$. Let us write $\|\mathbf{r}\| = n\bar{R}$, where $\bar{R}$ denotes the average resultant length.

Differentiating (3.2) w.r.t κ we obtain

$$\frac{n c_p'(\kappa)}{c_p(\kappa)} + n\bar{R} = 0. \tag{3.4}$$

For brevity, let us write $s = p/2 - 1$. From (2.1),

$$c_p'(\kappa) = \frac{s\kappa^{s-1}}{\alpha I_s(\kappa)} - \frac{\kappa^s I_s'(\kappa)}{\alpha I_s^2(\kappa)}, \tag{3.5}$$

where $\alpha = (2\pi)^{s+1}$ is a constant. We thus simplify $c_p'(\kappa)/c_p(\kappa)$ to be

$$\frac{s}{\kappa} - \frac{I_s'(\kappa)}{I_s(\kappa)}. \tag{3.6}$$

Using the fact that (see for e.g., [AS74])

$$\kappa I_{s+1}(\kappa) = \kappa I_s'(\kappa) - s I_s(\kappa), \tag{3.7}$$

we obtain

$$\frac{-c_p'(\kappa)}{c_p(\kappa)} = \frac{I_{s+1}(\kappa)}{I_s(\kappa)} = \frac{I_{p/2}(\kappa)}{I_{p/2-1}(\kappa)} = A_p(\kappa). \tag{3.8}$$

Using (3.4) and (3.8) we find that the m.l.e. for κ is given by

$$\hat{\kappa} = A_p^{-1}(\bar{R}). \tag{3.9}$$

Since $A_p(\kappa)$ is the ratio of Bessel functions we cannot obtain a closed form functional inverse. Hence to solve for $A_p^{-1}(\bar{R})$ we have to resort to numerical or asymptotic methods.

For large values of κ the following approximation is well known ([AS74], Chapter 9):

$$I_p(\kappa) \approx \frac{1}{\sqrt{2\pi\kappa}} e^{\kappa} \left(1 - \frac{4p^2-1}{8\kappa}\right). \tag{3.10}$$

Using (3.10) we obtain

$$A_p(\kappa) \approx \left(1 - \frac{p^2-1}{8\kappa}\right) \left(1 - \frac{(p-2)^2-1}{8\kappa}\right)^{-1}. \tag{3.11}$$

Now using the fact that κ is large, expanding the second term using the binomial theorem and ignoring terms that have squares or higher powers of κ in the denominator we are left with

$$A_p(\kappa) \approx \left(1 - \frac{p^2-1}{8\kappa}\right) \left(1 + \frac{(p-2)^2-1}{8\kappa}\right). \tag{3.12}$$

On again ignoring terms containing κ^2 in the denominator we finally have

$$A_p(\kappa) \approx 1 - \frac{p-1}{2\kappa}. \tag{3.13}$$

Hence for large κ we obtain

$$\hat{\kappa} = \frac{\frac{1}{2}(p-1)}{1-\bar{R}}. \tag{3.14}$$

We can write $I_p(\kappa)$ as (A.8),

$$I_p(\kappa) = \sum_{k\geq 0} \frac{1}{\Gamma(k+p+1)k!} \left(\frac{\kappa}{2}\right)^{2k+p}. \tag{3.15}$$

For small κ we use only the first two terms of this series, ignoring terms with higher powers of κ to get

$$I_p(\kappa) \approx \frac{\kappa^p}{2^p\, p!} + \frac{\kappa^{2+p}}{2^{p+2}\,(1+p)!}. \tag{3.16}$$

Using (3.16) and on simplifying $A_p(\kappa)$ we obtain

$$A_p(\kappa) \approx \frac{\kappa}{p}, \tag{3.17}$$

so that,

$$\hat{\kappa} = p\bar{R}. \tag{3.18}$$

See [MJ00] for conditions under which the approximations for $\hat{\kappa}$ are valid, at least for $p = 2, 3$.

These approximations for κ do not really take into account the dimensionality of the data and thus for high dimensions (when κ is big by itself but κ/p is not very small or very big) these estimates

fail to yield sufficient accuracy. We have found that the following seems to yield a very reasonable approximation most of the time (Appendix B gives a derivation):

$$\hat{\kappa} = \frac{\bar{R}p - \bar{R}^3}{1 - \bar{R}^2}. \tag{3.19}$$

While implementing the calculation of $A_p(\kappa)$ on a computer it pays to implement it as a continued fraction. To solve for κ we can use the approximation given by (3.19) as a starting point and then do a couple of Newton-Raphson iterations to improve our guess, though most often we do not really need very accurate approximations of κ and (3.19) suffices. Some further details can be found in Appendix B.

4 EM for a vMF mixture

In this section we derive the mixture-density parameter update equations for a mixture of von Mises-Fisher distributions. First we obtain the maximum likelihood estimates (m.l.e.) assuming complete data and then adapt to the incomplete data case viewing the problem in an Expectation Maximization (EM) framework. The Maximum Likelihood Estimates are derived using the method given in §10.3 of [DHS00].

4.1 Maximum Likelihood Estimates

Suppose that we are given a set $\mathscr{D} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ of n unlabeled samples drawn independently from the mixture density:

$$p(\mathbf{x}|\Theta) = \sum_{j=1}^{c} p(\mathbf{x}|\omega_j, \theta_j) P(\omega_j), \tag{4.1}$$

where $\omega_1, \ldots, \omega_c$ are the c classes from which data can come. The full parameter vector Θ is fixed but unknown. Since the $\mathbf{x}_i$ are assumed to be independent the likelihood can be written as

$$p(\mathscr{D}|\Theta) = \prod_{k=1}^{n} p(\mathbf{x}_k|\Theta). \tag{4.2}$$

The M.L.E. $\hat{\Theta}$ is that value of Θ that maximizes $p(\mathscr{D}|\Theta)$. Now let $\mathscr{L}(\mathscr{D}|\Theta)$ be the log-likelihood given by

$$\mathscr{L}(\mathscr{D}|\Theta) = \sum_{k=1}^{n} \ln p(\mathbf{x}_k|\Theta). \tag{4.3}$$

Note that $\Theta = (\theta_1, \ldots, \theta_c)^T$ is the total parameter vector; θ_i is the parameter vector for class i. We can write the gradient of the log-likelihood w.r.t. θ_i as,

$$\nabla_{\theta_i} \mathscr{L}(\Theta) = \sum_{k=1}^{n} \frac{1}{p(\mathbf{x}_k|\Theta)} \nabla_{\theta_i} \left(\sum_{j=1}^{c} p(\mathbf{x}_k|\omega_j, \theta_j) P(\omega_j) \right). \tag{4.4}$$

Now if we assume that θ_i is functionally independent of θ_j then we can simplify the above equations. First let us introduce the posterior probability (using Bayes' rule):

$$P(\omega_i|\mathbf{x}_k, \Theta) = \frac{p(\omega_i, \mathbf{x}_k|\Theta)}{p(\mathbf{x}_k|\Theta)} = \frac{p(\mathbf{x}_k|\omega_i, \Theta) P(\omega_i)}{p(\mathbf{x}_k|\Theta)}. \tag{4.5}$$

Using this posterior probability we see that the m.l.e. $\hat{\theta}_i$ must satisfy:

$$\sum_{k=1}^{n} P(\omega_i|\mathbf{x}_k, \Theta)\nabla_{\theta_i} \ln p(\mathbf{x}_k|\omega_i, \hat{\theta}_i) = 0 \quad i = 1, \ldots, c. \tag{4.6}$$

If the priors are also unknown then maximizing (4.6), subject to the condition

$$\sum_{j=1}^{c} \hat{P}(\omega_j) = 1, \tag{4.7}$$

we obtain the following m.l.e. for them:

$$\hat{P}(\omega_i) = \frac{1}{n}\sum_{k=1}^{n} \hat{P}(\omega_i|\mathbf{x}_k, \hat{\Theta}). \tag{4.8}$$

4.1.1 Maximum Likelihood for vMF mixture

The derivation given in the previous section applies to any probability density. Assuming that each sample $\mathbf{x}_k$ comes from a von Mises-Fisher distribution, i.e.

$$p(\mathbf{x}_k|\omega_i, \theta_i) = c_p(\kappa_i)e^{\kappa_i \boldsymbol{\mu}_i^T \mathbf{x}_k}, \tag{4.9}$$

we can solve the above maximum likelihood equations to obtain values of the parameters $(\kappa_i, \boldsymbol{\mu}_i)$ for $i = 1, \ldots, c$. We maximize the log-likelihood w.r.t. $\boldsymbol{\mu}_i$ subject to the constraint $\boldsymbol{\mu}_i^T \boldsymbol{\mu}_i = 1$ to obtain:

$$\hat{\boldsymbol{\mu}}_i = \frac{\sum_{k=1}^{n} \hat{P}(\omega_i|\mathbf{x}_k, \hat{\Theta})\mathbf{x}_k}{\| \sum_{k=1}^{n} \hat{P}(\omega_i|\mathbf{x}_k, \hat{\Theta})\mathbf{x}_k \|}. \tag{4.10}$$

Writing $-c_p'(\kappa_i)/c_p(\kappa_i) = A_p(\kappa_i)$ as usual, we obtain the following m.l.e. equation for κ_i:

$$A_p(\hat{\kappa}_i) = \frac{\sum_{k=1}^{n} \hat{P}(\omega_i|\mathbf{x}_k, \hat{\Theta})\hat{\boldsymbol{\mu}}_i^T \mathbf{x}_k}{\sum_{k=1}^{n} \hat{P}(\omega_i|\mathbf{x}_k, \hat{\Theta})}. \tag{4.11}$$

Hence we obtain $\hat{\kappa}_i$ by calculating $A_p^{-1}(\cdot)$ for the above argument (see Section 3).

In all these equations the value of the posterior probability is given by:

$$\hat{P}(\omega_i|\mathbf{x}_k, \hat{\Theta}) = \frac{c_p(\kappa_i)e^{\kappa_i \boldsymbol{\mu}_i^T \mathbf{x}_k}\hat{P}(\omega_i)}{\sum_{j=1}^{c} c_p(\kappa_j)e^{\kappa_j \boldsymbol{\mu}_j^T \mathbf{x}_k}\hat{P}(\omega_j)}. \tag{4.12}$$

From these equations it seems that the posterior probability is large when: $c_p(\kappa_i)$ is large and when $\kappa_i \boldsymbol{\mu}_i^T \mathbf{x}_k$ is large. We could thus use these in an explicit objective function while iteratively calculating the m.l.e. for the parameters.

4.2 Parameter estimation using EM

For unlabeled data points the class to which a given data point belongs is not known. In the presence of such incomplete data we have to take resort to an Expectation Maximization scheme for calculating the m.l.e. for parameters. We have the following probabilistic model

$$p(\mathbf{x}|\Theta) = \sum_{j=1}^{c} \alpha_j p(\mathbf{x}|\theta_j), \tag{4.13}$$

where the α_j's are the so called "mixing" parameters (or class priors) and Θ is the parameter vector for the mixture model. The incomplete-data log-likelihood expression for this density from the data $\mathscr{D}$ is given by:

$$\mathscr{L}(\mathscr{D}|\Theta) = \ln \prod_{k=1}^{n} p(\mathbf{x}_k|\Theta). \tag{4.14}$$

Now if we consider $\mathscr{D}$ to be incomplete, but assume the existence of unobserved data items $\mathscr{Y} = \{y_i\}_{i=1}^n$, whose values inform us which component density generated each data item, the problem becomes easier. That is to say that each y_i here corresponds to some ω_j as discussed in the previous section. We let, $y_i = k$ if the i^{th} sample $\mathbf{x}_i$ was generated by the mixture component corresponding to ω_k. Thus we can look at the m.l.e. derivation in the previous section in a manner similar to the one given by [Bil97]. After some tedious algebra we essentially reach the same equations as given in the previous section. The scheme to perform the calculations is an EM algorithm that proceeds by iterative updates to estimate the parameters of the assumed distribution on data.

Algorithm Estimate $\alpha_j, \boldsymbol{\mu}_j, \kappa_j$ for $1 \leq j \leq c$

0: Initialize all $\alpha_j, \boldsymbol{\mu}_j, \kappa_j, P(\omega_j|\mathbf{x}_k, \theta)$
2. **repeat**
3. **for** $k = 1$ **to** N **do**
4. **for** $j = 1$ **to** c **do**
5. $p(\mathbf{x}_k|\omega_j, \Theta) = c_p(\kappa_j) e^{\kappa_j \boldsymbol{\mu}_j^T \mathbf{x}_k}$
$\hat{P}(\omega_j|\mathbf{x}_k, \hat{\Theta}) = \dfrac{p(\mathbf{x}_k|\omega_j, \Theta)\alpha_j}{\sum_{l=1}^{c} p(\mathbf{x}_k|\omega_l, \Theta)\alpha_l}$
6. **end**
7. **end**
8. **for** $j = 1$ **to** c **do**
9. $n_j = \sum_{k=1}^{n} \hat{P}(\omega_j|\mathbf{x}_k, \hat{\Theta})$
$\hat{\alpha}_j = n_j/n$
$\mathbf{r}_j = \sum_{k=1}^{n} \hat{P}(\omega_j|\mathbf{x}_k, \hat{\Theta})\mathbf{x}_k$
$\hat{\boldsymbol{\mu}}_j = \mathbf{r}_j/\|\mathbf{r}_j\|$
$\hat{\kappa}_j = A_p^{-1}(\|\mathbf{r}_j\|/n_j)$
10. **end**
11. **until** *stopping criteria met.*

Figure 2: EM algorithm for a mixture of vMF distributions.

4.3 Implementation Details

The above algorithm was implemented in MATLAB and its source is available upon request. The calculation of $\hat{\kappa}_j$ in step 9 above is implemented using the approximation given by (3.19).

There are various ways in which we could initialize our EM algorithm. An easy and effective method is to initialize the original guesses of the mean directions by using a spherical k-means type algorithm [DM01], and calculate the initial values of the parameters from the clustering obtained.

5 Experiments

This section discusses some of the experiments performed and the results obtained. We tested our algorithm on data sampled from simulated mixtures of vMFs.

5.1 Simulation of vMF mixtures

This information is adapted from Chapter 10 of [MJ00]. For $\kappa > 0$, the associated vMF distribution has a mode at the mean direction $\boldsymbol{\mu}$, whereas when $\kappa = 0$ the distribution is uniform. The larger the value of κ, the greater is the clustering around the mean direction.

Since the vMF density depends on $\mathbf{x}$ only through $\boldsymbol{\mu}^T\mathbf{x}$, this distribution is rotationally symmetric about $\boldsymbol{\mu}$. Further in the tangent normal decomposition:

$$\mathbf{x} = t\boldsymbol{\mu} + (1-t^2)^{1/2}\boldsymbol{\zeta}, \tag{5.1}$$

t is invariant under rotation about $\boldsymbol{\mu}$ while $\boldsymbol{\zeta}$ is equivariant (i.e. any such rotation Q takes $\boldsymbol{\zeta}$ to $Q\boldsymbol{\zeta}$). Thus the conditional distribution $\boldsymbol{\zeta}|t$ is uniform on S^{p-2}. It follows that $\boldsymbol{\zeta}$ and t are independent and $\boldsymbol{\zeta}$ is uniform on S^{p-2}. Further (see [MJ00]), we see that the marginal density of t is:

$$\frac{\left(\frac{\kappa}{2}\right)^{\frac{p}{2}-1}}{\Gamma(\frac{p-1}{2})\Gamma(\frac{1}{2})I_{\frac{p-1}{2}}(\kappa)}e^{\kappa t}(1-t^2)^{\frac{p-3}{2}}, \tag{5.2}$$

on the interval $[-1, 1]$.

---

function mixsamp(n, d, M)
In: n points to sample; d dimensionality, M mixture data structure
Out: M modified mixture, L label of each sampled point.

1. $L \leftarrow$ zeros(n,1);
2. $P \leftarrow$ rand(1,n);
3. $\mathscr{X} \leftarrow$ zeros(n,d);
4. $cp \leftarrow 0$; {Cumulative sum of priors}
5. $cs \leftarrow 0$; {Cumulative sum of number of sampled points}
6. **for** $j \leftarrow 1$ **to** k
   $ns \leftarrow \text{sum}(P \geq cp$ **and** $P < cp + M.P(\omega_j))$;
   $\kappa \leftarrow M.\kappa(j)$;
   $\mathscr{X}(ns+1 : cs+ns, :) \leftarrow \text{vsamp}(M.\boldsymbol{\mu}_j, \kappa, ns)$;
   $L(cs+1 : cs+ns) \leftarrow j$;
   $cp \leftarrow cp + M.P(\omega_j)$;
   $cs \leftarrow cs + ns$;
7. **end**
8. $M.\mathscr{X} \leftarrow \mathscr{X}$

Figure 3: Simulating a mixture of vMFs.

From the facts that $\boldsymbol{\zeta}$ and t are independent and that $\boldsymbol{\zeta}$ is uniformly distributed on S^{p-2} it follows that the simulation of a vMF is easy. If $\boldsymbol{\zeta}$ and t are generated independently from the uniform distribution on S^{p-2} and from (5.2) respectively then

$$\mathbf{x} = t\boldsymbol{\mu} + (1-t^2)^{1/2}\boldsymbol{\zeta},$$

is a pseudo-random unit vector with the $M_p(\boldsymbol{\mu}, \kappa)$ distribution. Further information about this can be found in [Woo94]. We used the MATLAB Statistics Toolbox for aiding our implementation of Wood's algorithm ([Woo94]). Figure 4 gives Wood's algorithm (slight adaptation) that we used to simulate a single vMF distribution. Figure 3 gives the algorithm used to simulate a mixture of vMF distributions with given parameters. The algorithm in Figure 3 makes use of the algorithm in Figure 4.

function vsamp($\boldsymbol{\mu}, \kappa, n$)
{Adapted from [Woo94]}
In: $\boldsymbol{\mu}$ mean vector for vMF, κ parameter for vMF
In: n, number of points to generate
Out: S the Set of n vMF($\boldsymbol{\mu}, \kappa$) samples

1. $d \leftarrow dim(\boldsymbol{\mu})$
2. $t_1 \leftarrow \sqrt{4\kappa^2 + (d-1)^2}$
3. $b \leftarrow (-2\kappa + t_1)/(d-1)$
4. $x_0 \leftarrow (1-b)/(1+b)$
5. $S \leftarrow zeros(n, d)$

6. $m \leftarrow (d-1)/2$
7. $c \leftarrow \kappa x_0 + (d-1)\log(1-x_0^2)$
8. **for** $i \leftarrow 1$ **to** n
   $t \leftarrow -1000$
   $u \leftarrow 1$
   while $(t < \log(u))$
   $z \leftarrow \beta(m, m)$ {$\beta(x, y)$ gives a beta random variable}
   $u \leftarrow rand$ {$rand$ gives a uniformly distributed random number.}
   $w \leftarrow \frac{(1-(1+b)z)}{(1-(1-b)z)}$
   $t \leftarrow \kappa w + (d-1)\log(1-x_0 w)$
   end
   $\mathbf{v} \leftarrow urand(d-1)$ {$urand(p)$ gives a p-dim vector from unif. distr. on sphere.}
   $\mathbf{v} \leftarrow \mathbf{v}/\|\mathbf{v}\|$
   $S(i, 1:d-1) \leftarrow \sqrt{1-w^2}\mathbf{v}^T$
   $S(i, d) = w$
9. **end**

{ We now have n samples from vMF($[0\ 0\ \ldots\ 1]^T, \kappa$) }
10. Perform an orthogonal transformation on each sample in S
    The transformation has to satisfy $Q\boldsymbol{\mu} = [0\ 0\ \ldots\ 1]^T$
11. **return** S.

Figure 4: Algorithm to simulate a vMF

5.2 Experiment 1

In this section we discuss briefly some experiments carried out with the aim of verifying the accuracy of m.l.e. for parameters of a single vMF distribution. A consequence of the experiments is the verification of the vMF simulation algorithm given in Figure 4.

5.2.1 Experiment 1.1

This experiment deals with estimating parametes of a three-dimensional vMF distribution. The results are are summarized in Table1. The true mean and true concentration are denoted by $\boldsymbol{\mu}$

$\boldsymbol{\mu}$	κ	n	$\hat{\boldsymbol{\mu}}^T\boldsymbol{\mu}$	$\hat{\kappa}$
$[.7071\ .7071\ 0]'$	4	100	.9994	4.1568
$[.7071\ .7071\ 0]'$	10	1000	.9998	10.4561
$[.1543\ .6172\ .7715]'$	15	1000	1.000	15.2949

Table 1: MLE for single vMF with $p = 3$

and κ respectively, n denotes the number of samples and $\hat{\boldsymbol{\mu}}, \hat{\kappa}$ denote the estimated parameters. These results clearly indicate that the m.l.e. for κ and $\boldsymbol{\mu}$ are quite accurate, and in the presence of large amounts of sample data m.l.e. approximate the true parameters quite well. Note that the calculations for κ were done using an approximation, but that does not lead to too much inaccuracy.

5.2.2 Experiment 1.2

This experiment is in similar vein to experiment 1.1, except that we tried it for 20-dimensional simulated data. Table 2 summarizes the results. These experiments lend confidence to our belief in

$\boldsymbol{\mu}$	κ	n	$\hat{\boldsymbol{\mu}}^T\boldsymbol{\mu}$	$\hat{\kappa}$
Random vector	10	100	0.9739	10.2989
Random vector	10	1000	0.9983	10.2506

Table 2: MLE for vMF distribution with $p = 20$

both the simulation and the MLE. Next we shall discuss MLE for simulated mixtures of vMFs.

5.3 Experiment 2

We provide a detailed example of clustering for a two component mixture of vMFs on a circle to illustrate the performance of our EM algorithm. The dataset that we considered was a small dataset of 50, two-dimensional points drawn from a mixture of two vMF distributions. The mean direction for each component was set to some random vector and κ was set to 4.

Figure 5(a) shows a plot of the points. From the plot we observe that there are two clusters of points (which is natural because the data was sampled from a mixture of two vMFs). Most points belong to either one component or the other. Some of the points seem to have mixed membership to each component. As we shall soon see, our EM algorithm figures out these points and assigns them fractionally to either component. The components as recovered by EM algorithm are illustrated in Figure 5(b).

From Figure 5(b), we can see that the points that we would have visually assigned to both components, have been given a mixed membership. This assignment seems to concur with our notion of a "correct" assignment. More precisely, in Figure 5(b), a point that has a probability exceeding 0.10, of membership to either component, is called a point with mixed membership. Thus