Expectation Maximization - Math and Pictures

Johannes Traa

This document covers the basics of the EM algorithm, maximum likelihood (ML) estimation, and maximum a posteriori (MAP) estimation. It also covers the EM derivations for the following mixture models:

- Gaussian Mixture Model (GMM)
- Wrapped Gaussian Mixture Model (WGMM)
- von Mises Mixture Model (vMMM)
- von Mises-Fisher Mixture Model (vMFMM)
- Line Mixture Model (LineMM)
- Laplacian Mixture Model (LapMM)
- Probabilistic Latent Semantic Indexing (PLSI)

The best way to understand this stuff is to code it up. Plot everything.

Excellent references for the EM algorithm and probabilistic methods:

- *Chapter 9: Mixture Models and EM* in "Pattern Recognition and Machine Learning" ('06) (Bishop)
- *Chapter 10: Grouping and Model Fitting* in "Computer Vision: A Modern Approach" ('12) (Forsyth, Ponce)
- "Machine Learning: A Probabilistic Perspective" ('12) (Murphy)

1 Basic idea

Parameter estimation is a general problem that shows up again and again. A typical situation is where we have collected some data and we want to summarize/find structure in that dataset. Take the following simple example. You are trying to model the interaction between good weather and the number of people at the beach (it's a silly example, but just roll with it). If we measure both of these quantities every day of the year and make a scatterplot of our data, it might look the set of points in Figure 1. There's a positive correlation between good weather and people going to the beach and the data is spread around a center point.

Instead of keeping track of the entire dataset, we can represent it with a Gaussian distribution, which looks like a squished and rotated bell-curve in 2 dimensions. A slices through the Gaussian (contours) are shown in Figure 1 as ellipses. This pretty much summarizes all the data with two parameters: a mean vector (2×1) and a covariance matrix (2×2).

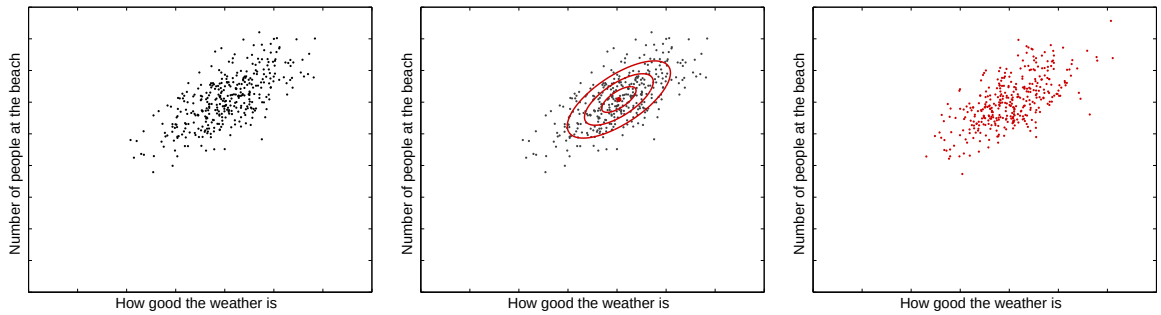

Figure 1: (Left) Dataset representing relationship between the number of people at the beach and the "goodness" of the weather. (Middle) Gaussian distribution fit to the data. (Right) Samples drawn from Gaussian fit.

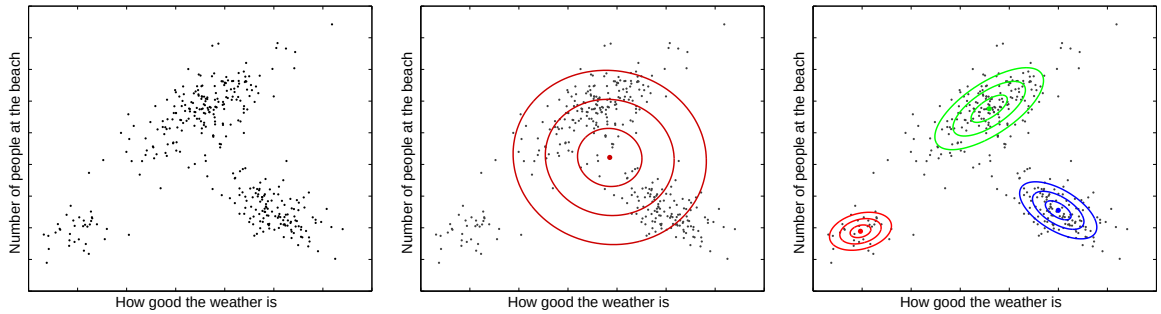

Figure 2: (Left) Multimodal dataset. (Left) Gaussian fit. (Right) Mixture of Gaussians fit.

By fitting a Gaussian, we are implicitly making the assumption that it's reasonable to model the data as having been sampled from a Gaussian. If we generate data from our Gaussian fit, as shown in Figure 1, we can see that the samples are spread in the same way as the actual data. So, in this case, our implicit assumption is reasonable. But what if our data looks like that of Figure 2? The Gaussian assumption doesn't make sense here. At least it looks like we can do better. The underlying distribution appears multimodal: it has *multiple* peaks. So, we can just fit *multiple* Gaussians instead of one.

To fit a single distribution, we typically apply the maximum likelihood (ML) or maximum a-posteriori (MAP) method. The former tries to find the distribution that makes the most sense given the data, while the latter takes into account our belief, independent of the data, of what that distribution should look like. The Expectation-Maximization (EM) algorithm is a straightforward way to fit mixture models that starts with an initial guess and iteratively improves the fit. Each iteration consists of two steps. The first assigns data points to clusters and the second re-estimates the cluster parameters according to the assignments. In EM, we typically associate each data point to each cluster with some probability rather than using binary assignments.

The rest is technical details.

2 MAXIMUM LIKELIHOOD FOR THE GAUSSIAN DISTRIBUTION

The maximum likelihood estimate of the mean of a Gaussian distribution is simple. The multivariate Gaussian distribution has pdf:

$$\mathcal{N}(\mathbf{x};\boldsymbol{\mu},\Sigma) = \frac{1}{|2\pi\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu})} \ . \tag{1}$$

The likelihood function for a dataset drawn i.i.d. from the distribution is:

$$\mathcal{L} = \prod_{i=1}^{N} \frac{1}{|2\pi\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}(\mathbf{x}_i-\boldsymbol{\mu})^T\Sigma^{-1}(\mathbf{x}_i-\boldsymbol{\mu})} \ , \tag{2}$$

so the log likelihood is:

$$\log\mathcal{L} \propto \sum_{i=1}^{N} -\frac{1}{2}\log|\Sigma| - \frac{1}{2}\left(\mathbf{x}_i-\boldsymbol{\mu}\right)^T\Sigma^{-1}\left(\mathbf{x}_i-\boldsymbol{\mu}\right) \ . \tag{3}$$

Since this is a convex function of the parameters, we can differentiate it, set the result equal to zero and solve for the ML parameter estimates:

$$\frac{\partial\log\mathcal{L}}{\partial\mu} = \sum_{i=1}^{N}\Sigma^{-1}\left(\mathbf{x}_i-\boldsymbol{\mu}\right) = \mathbf{0} \tag{4}$$

$$\rightarrow \widehat{\boldsymbol{\mu}}_{\mathrm{ML}} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_i \ , \tag{5}$$

$$\frac{\partial\log\mathcal{L}}{\partial\Sigma} = \sum_{i=1}^{N} -\frac{1}{2}\Sigma^{-1} + \frac{1}{2}\Sigma^{-1}\left(\mathbf{x}_i-\boldsymbol{\mu}\right)\left(\mathbf{x}_i-\boldsymbol{\mu}\right)^T\Sigma^{-1} = \mathbf{0} \tag{6}$$

$$\rightarrow \widehat{\Sigma}_{\mathrm{ML}} = \frac{1}{N}\sum_{i=1}^{N}\left(\mathbf{x}_i-\boldsymbol{\mu}\right)\left(\mathbf{x}_i-\boldsymbol{\mu}\right)^T \ . \tag{7}$$

So the ML estimates are just the sample mean and covariance.

2.1 DATA WEIGHTING

We might also include a weight w_i for each data point to reflect how confident we are that it is reliable. We modify the likelihood as:

$$\widetilde{\mathcal{L}} = \prod_{i=1}^{N}\left[\frac{1}{|2\pi\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}(\mathbf{x}_i-\boldsymbol{\mu})^T\Sigma^{-1}(\mathbf{x}_i-\boldsymbol{\mu})}\right]^{w_i} \ , \tag{8}$$

so the log likelihood is:

$$\log\widetilde{\mathcal{L}} \propto \sum_{i=1}^{N}\left[-\frac{1}{2}\log|\Sigma| - \frac{1}{2}\left(\mathbf{x}_i-\boldsymbol{\mu}\right)^T\Sigma^{-1}\left(\mathbf{x}_i-\boldsymbol{\mu}\right)\right] w_i \ . \tag{9}$$

Maximizing with respect to the parameters gives the weighted ML estimates:

$$\widehat{\boldsymbol{\mu}}_{\text{WML}} = \frac{\sum_{i=1}^{N} \mathbf{x}_i \, w_i}{\sum_{i=1}^{N} w_i} \ , \tag{10}$$

$$\widehat{\Sigma}_{\text{WML}} = \frac{\sum_{i=1}^{N} \left(\mathbf{x}_i - \boldsymbol{\mu}\right)\left(\mathbf{x}_i - \boldsymbol{\mu}\right)^T \, w_i}{\sum_{i=1}^{N} w_i} \ . \tag{11}$$

3 Maximum a Posteriori for the Gaussian Distribution

We can regularize the mean and covariance estimates of a Gaussian distribution by incorporating prior information. This biases the solution towards what we believe it should look like before seeing any data. The conjugate distribution for the mean and covariance are Gaussian and Inverse Wishart distributions, respectively. These conjugate priors ensure that the posterior distribution (likelihood × prior) is of the same form as the prior (this is merely a convenience at this point).

3.1 Prior on the mean

We can regularize the maximum likelihood solution by imposing a Gaussian prior on the mean:

$$P\left(\boldsymbol{\mu}; \boldsymbol{\mu}_s, \Sigma_s\right) = \frac{1}{|2\pi\Sigma_s|^{\frac{1}{2}}} e^{-\frac{1}{2}\left(\boldsymbol{\mu}-\boldsymbol{\mu}_s\right)^T \Sigma_s^{-1} \left(\boldsymbol{\mu}-\boldsymbol{\mu}_s\right)} \ . \tag{12}$$

Thus, we have:

$$\mathscr{P} = \left[\prod_{i=1}^{N} P\left(\mathbf{x}_i; \boldsymbol{\mu}, \Sigma\right)\right] P\left(\boldsymbol{\mu}; \boldsymbol{\mu}_s, \Sigma_s\right) \ , \tag{13}$$

$$\log \mathscr{P} = \left[\sum_{i=1}^{N} -\frac{1}{2}\log|2\pi\Sigma| - \frac{1}{2}\left(\mathbf{x}_i - \boldsymbol{\mu}\right)^T \Sigma^{-1} \left(\mathbf{x}_i - \boldsymbol{\mu}\right)\right] - \frac{1}{2}\log|2\pi\Sigma_s| - \frac{1}{2}\left(\boldsymbol{\mu} - \boldsymbol{\mu}_s\right)^T \Sigma_s^{-1} \left(\boldsymbol{\mu} - \boldsymbol{\mu}_s\right) \tag{14}$$

$$\frac{\partial \log \mathscr{P}}{\partial \boldsymbol{\mu}} = \left[\sum_{i=1}^{N} \Sigma^{-1} \left(\mathbf{x}_i - \boldsymbol{\mu}\right)\right] - \Sigma_s^{-1} \left(\boldsymbol{\mu} - \boldsymbol{\mu}_s\right) = \mathbf{0} \ . \tag{15}$$

Solving for $\boldsymbol{\mu}$, the MAP solution is:

$$\widehat{\boldsymbol{\mu}}_{\text{MAP}} = \left(N\Sigma_s\Sigma^{-1} + \mathbf{I}\right)^{-1} \left(\Sigma_s\Sigma^{-1}\left(\sum_{i=1}^{N} \mathbf{x}_i\right) + \boldsymbol{\mu}_s\right) \tag{16}$$

$$= \left(N\Sigma_s\Sigma^{-1} + \mathbf{I}\right)^{-1} \left(N\Sigma_s\Sigma^{-1}\widehat{\boldsymbol{\mu}}_{\text{ML}} + \boldsymbol{\mu}_s\right) \ . \tag{17}$$

Consider the 1D case for simplicity:

$$\widehat{\mu}_{\text{MAP}} = \frac{\frac{\sigma_s^2}{\sigma^2} N \widehat{\mu}_{\text{ML}} + \mu_s}{\frac{\sigma_s^2}{\sigma^2} N + 1} \ . \tag{18}$$

As $\frac{\sigma_s^2}{\sigma^2} \to \infty$, the prior is uninformative, so $\mu_{\text{MAP}} \to \mu_{\text{ML}}$. And as $\frac{\sigma_s^2}{\sigma^2} \to 0$, the data is uninformative (the prior is strict), so $\mu_{\text{MAP}} \to \mu_s$. When $\frac{\sigma_s^2}{\sigma^2} = 1$, the prior behaves as if one additional measurement at μ_s were present to calculate the ML solution. Also, as $N \to \infty$, the prior becomes redundant, so $\mu_{\text{MAP}} \to \mu_{\text{ML}}$.

3.2 Prior on the covariance

We can also regularize the solution for the covariance matrix using the inverse-Wishart distribution (with the appropriate degrees of freedom):

$$P(\Sigma; \Sigma_0) \propto \frac{|\Sigma_0|^{\frac{n}{2}}}{|\Sigma|^{\frac{n}{2}}} e^{-\frac{1}{2} tr(\Sigma^{-1}\Sigma_0)} \ . \tag{19}$$

Thus,

$$\mathscr{P} = \left[\prod_{i=1}^{N} P\left(\mathbf{x}_i; \boldsymbol{\mu}, \Sigma\right) \right] P(\Sigma; \Sigma_0) \ , \tag{20}$$

$$\log \mathscr{P} \propto \left[\sum_{i=1}^{N} -\frac{1}{2} \log|2\pi\Sigma| - \frac{1}{2} \left(\mathbf{x}_i - \boldsymbol{\mu}\right)^T \Sigma^{-1} \left(\mathbf{x}_i - \boldsymbol{\mu}\right) \right] + \frac{n}{2} \log|\Sigma_0| - \frac{n}{2} \log|\Sigma| - \frac{1}{2} tr\left(\Sigma^{-1}\Sigma_0\right) \ , \tag{21}$$

$$\frac{\partial \log \mathscr{P}}{\partial \Sigma} = \left[\sum_{i=1}^{N} -\frac{1}{2}\Sigma^{-1} + \frac{1}{2}\Sigma^{-1} \left(\mathbf{x}_i - \boldsymbol{\mu}\right)\left(\mathbf{x}_i - \boldsymbol{\mu}\right)^T \Sigma^{-1} \right] - \frac{n}{2}\Sigma^{-1} + \frac{1}{2}\Sigma^{-1}\Sigma_0\Sigma^{-1} = 0 \ . \tag{22}$$

Thus, we have that:

$$\widehat{\Sigma}_{\text{MAP}} = \frac{\Sigma_0 + \sum_{i=1}^{N} \left(\mathbf{x}_i - \boldsymbol{\mu}\right)\left(\mathbf{x}_i - \boldsymbol{\mu}\right)^T}{n + N} = \frac{\Sigma_0 + N\widehat{\Sigma}_{\text{ML}}}{n + N} \ . \tag{23}$$

Thanks to the conjugacy relationship between the likelihood and the prior, the MAP solution is very intuitive. The parameter n in the inverse-Wishart distribution controls how confident we are that Σ_0 is the correct estimate. If we are not very confident, n is set to a small number and a moderate number N of data samples will cause the MAP estimate to ignore the prior.

4 The Math Behind EM

Expectation-Maximization (EM) is a learning algorithm for maximum-likelihood problems with hidden variables. In the case of a mixture model, we have observed data/variables X,

unobserved data/variables Z, and parameters Θ (to be learned). The hidden variables Z indicate how the observed data X are assigned to the mixture components. The complete data likelihood for a mixture model with i.i.d. samples is

$$\mathscr{L} = \prod_{i=1}^{N} P(x_i, z_i; \Theta) \tag{24}$$

$$= \prod_{i=1}^{N} \sum_{k=1}^{K} P(x_i|z_{ik}; \theta_k) P(z_k; \theta_k) \tag{25}$$

$$= \prod_{i=1}^{N} \prod_{k=1}^{K} [P(x_i; \theta_k) P(z_k; \theta_k)]^{z_{ik}} \tag{26}$$

$$= \prod_{i=1}^{N} \prod_{k=1}^{K} [P(x_i; \theta_k) \pi_k]^{z_{ik}} \tag{27}$$

$P(x_i, z_i; \Theta)$ is the complete data likelihood for the i^{th} observation x_i, $P(x_i; \theta_k)$ is the probability model (pdf) of the k^{th} component in the mixture evaluated at x_i, and $\pi_k = P(z_k; \theta_k)$ is the mixing weight of the k^{th} component.

The hidden variables z_{ik} are treated as indicator variables for each i in the above notation. So, for the i^{th} observation x_i, z_{ik} takes the value 1 for a single index k and 0 for all others. This has the effect of selecting one term in the product over k for each i. It's easier to work with the log likelihood, in which case we have

$$\log \mathscr{L} = \sum_{i=1}^{N} \sum_{k=1}^{K} \log [P(x_i; \theta_k) \pi_k] \, z_{ik} \tag{28}$$

This is easy to maximize w.r.t. the parameters θ_k if we know the values of the indicator variables z_{ik}. In that case, we can just estimate the parameters for the k^{th} component using all the data whose indicator is active for that component (i.e. $z_{ik} = 1$). Seeing as we don't know these data associations, we can first lower-bound the log likelihood by taking its expected value w.r.t. the hidden variables (this requires Jensen's inequality). This gives what is known as the "Q function":

$$Q = \mathrm{E}_{z|x,\Theta^{\text{old}}} \left[\log \mathscr{L} \right] \tag{29}$$

$$= \sum_{i=1}^{N} \sum_{k=1}^{K} \log [P(x_i; \theta_k) \pi_k] \, \alpha_{ik} \ , \tag{30}$$

where

$$\alpha_{ik} = \mathrm{E}_{z|x,\Theta^{\text{old}}}[z_{ik}] \tag{31}$$

$$= P(z_{ik}|x_i;\Theta^{\text{old}}) \tag{32}$$

$$= \frac{P(x_i|z_{ik};\theta_k^{\text{old}})P(z_{ik};\theta_k^{\text{old}})}{\sum_{l=1}^{K} P(x_i|z_{il};\theta_l^{\text{old}})P(z_{il};\theta_l^{\text{old}})} \tag{33}$$

$$= \frac{P(x_i|z_{ik};\theta_k^{\text{old}})\pi_k}{\sum_{l=1}^{K} P(x_i|z_{il};\theta_l^{\text{old}})\pi_l} \tag{34}$$

represents our belief that the k^{th} component in the mixture is responsible for generating the i^{th} observation. (32) follows since the expectation of an indicator variable is its probability of being 1.

The Q function is easier to maximize and leads to the EM algorithm. In the E step, we fix the current estimate of the parameters Θ and calculate the posterior probabilities α_{ik}. This captures how much information each data point x_i contributes in estimating the parameters of each component θ_k. Then, in the M step, we use these posteriors as *soft weights* to update the model parameters via maximization of (30). Data points with higher weights for a specific value of k will exert more influence on the update of the k^{th} component's parameters. After the M step, Θ has changed, so the α_{ik} have changed. We can re-estimate α_{ik}, update Θ, and repeat until convergence. This procedure is guaranteed to reach a local maximum of (28).

5 Gaussian Mixture Model (GMM)

The model is a K-component Mixture of Gaussians (MoG). All data is drawn independently from this mixture. The likelihood function is given by

$$\mathscr{L} = \prod_{i=1}^{N} \sum_{k=1}^{K} \pi_k \mathcal{N}\left(x_i\,;\,\mu_k,\Sigma_k\right)\ , \tag{35}$$

$$\log \mathscr{L} = \sum_{i=1}^{N} \log \sum_{k=1}^{K} \pi_k \mathcal{N}\left(x_i\,;\,\mu_k,\Sigma_k\right)\ . \tag{36}$$

The Q function is given by

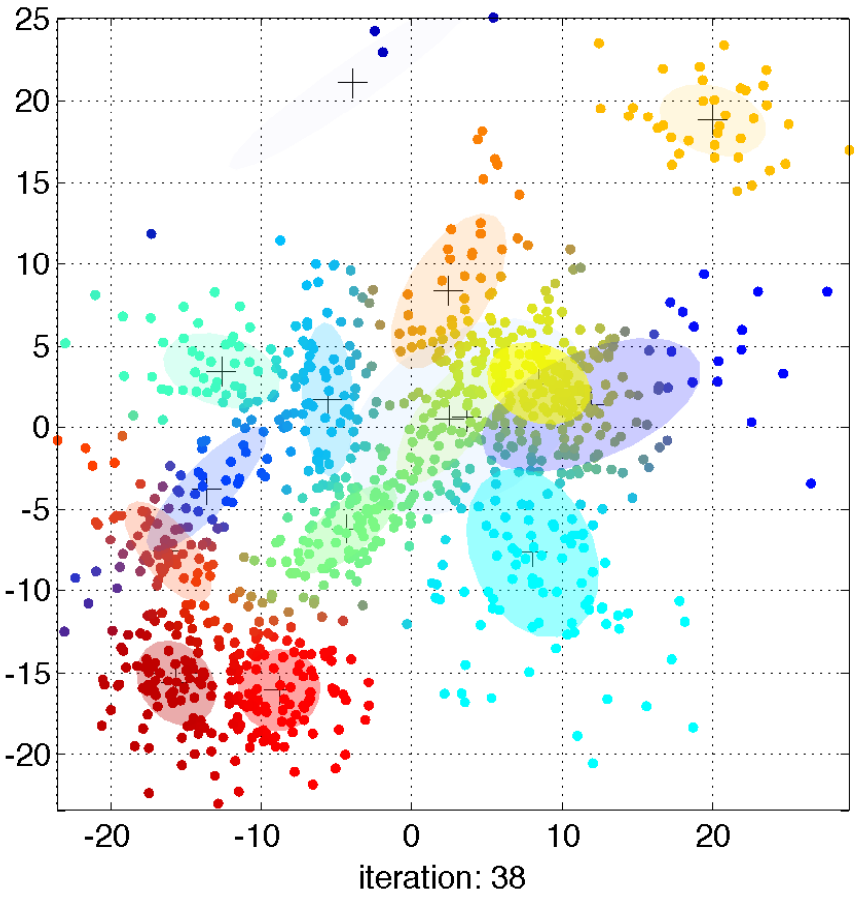

Figure 3: Mixture of Gaussians fit to dataset in 2 dimensions. Each Gaussian is depicted by its mean μ (black '+'), covariance Σ (1-σ ellipse), and mixing weight π (transparency). Data points are colored by their posterior probabilities η_{ik}.

$$Q = \mathrm{E}_{z|x,\Theta^{(t)}}\left[\log P\left(x,z|\Theta\right)\right] \tag{37}$$

$$= \sum_{i=1}^{N}\sum_{k=1}^{K} \log\left[P\left(z_k|\Theta\right)P\left(x_i|z_k;\Theta\right)\right] P\left(z_k|x_i;\Theta^{(t)}\right) \tag{38}$$

$$= \sum_{i=1}^{N}\sum_{k=1}^{K} \log\left[\pi_k \mathcal{N}\left(x_i;\mu_k,\Sigma_k\right)\right]\eta_{ik} \tag{39}$$

$$= \sum_{i=1}^{N}\sum_{k=1}^{K} \log\left[\pi_k \frac{1}{|2\pi\Sigma_k|^{\frac{1}{2}}} e^{-\frac{1}{2}\left(\mu_k - x_i\right)^T \Sigma_k^{-1}\left(\mu_k - x_i\right)}\right]\eta_{ik} \tag{40}$$

$$= \sum_{i=1}^{N}\sum_{k=1}^{K} \left[\log\pi_k - \frac{1}{2}\log|2\pi\Sigma_k| - \frac{1}{2}\left(\mu_k - x_i\right)^T \Sigma_k^{-1}\left(\mu_k - x_i\right)\right]\eta_{ik} \tag{41}$$

where Θ is the parameter set to solve for, $\Theta^{(t)}$ is the previous iteration's parameters, and

$\eta_{ik} = P(z_k|x_i;\Theta^{(t)})$ is the posterior probability of each hidden variable given the parameters from the previous iteration, given by

$$\eta_{ik} = P(z_k|x_i;\Theta^{(t)}) = \frac{P\left(x_i|z_k;\Theta^{(t)}\right)P\left(z_k|\Theta^{(t)}\right)}{P\left(x_i|\Theta^{(t)}\right)} = \frac{P\left(x_i|z_k;\Theta^{(t)}\right)P\left(z_k|\Theta^{(t)}\right)}{\sum_{k=1}^{K} P\left(x_i|z_k;\Theta^{(t)}\right)P\left(z_k|\Theta^{(t)}\right)} \ . \tag{42}$$

The hidden variables indicate what cluster each data point is generated from.

In each iteration, we need to optimize the Q function in each coordinate of the parameter space. To do this, we take derivatives with respect to each of the parameters, set the result to zero, and solve for the locally optimal new values:[1]

$$\frac{\partial Q}{\partial \mu_k} \propto \sum_{i=1}^{N} \Sigma_k^{-1}\left(\mu_k - x_i\right)\eta_{ik} = 0 \tag{43}$$

$$\frac{\partial Q}{\partial \Sigma_k} \propto \sum_{i=1}^{N} \left[-\Sigma_k^{-1} + \Sigma_k^{-1}\left(\mu_k - x_i\right)\left(\mu_k - x_i\right)^T \Sigma_k^{-1}\right]\eta_{ik} = 0 \tag{44}$$

$$\frac{\partial\left(Q + \lambda\left(\sum_{k=1}^{K}\pi_k - 1\right)\right)}{\partial \pi_k} = \sum_{i=1}^{N}\frac{1}{\pi_k}\eta_{ik} + \lambda = 0 \quad , \quad \sum_{k=1}^{K}\pi_k = 1 \tag{45}$$

Re-arranging terms and solving for the new model parameters, we get

$$\widehat{\mu}_k = \frac{\sum_{i=1}^{N}\eta_{ik}x_i}{\sum_{i=1}^{N}\eta_{ik}} \ , \tag{46}$$

$$\widehat{\Sigma}_k = \frac{\sum_{i=1}^{N}\eta_{ik}\left(\mu_k - x_i\right)\left(\mu_k - x_i\right)^T}{\sum_{i=1}^{N}\eta_{ik}} \ , \tag{47}$$

$$\widehat{\pi}_k = \frac{1}{N}\sum_{i=1}^{N}\eta_{ik} \ . \tag{48}$$

These form the M-step update equations for the GMM fitting algorithm. It's interesting to note that the mean and covariance updates are just weighted ML estimates. The posterior probability η_{ik} corresponds to how confident we are that the i^{th} data point was sampled from the j^{th} Gaussian.

5.1 GMM WITH PRIORS AND DATA WEIGHTING

If we place priors on the means and/or covariances, the results are simply posterior-weighted MAP estimators. For example, the MAP update for the means is:

[1] Lagrange multipliers are used to enforce equality constraints.

$$\widehat{\mu}_k = \left[\Omega_k \Sigma_k^{-1} \left(\sum_{i=1}^{N} \eta_{ik}\right) + \mathbf{I}\right]^{-1} \left[\Omega_k \Sigma_k^{-1} \left(\sum_{i=1}^{N} \eta_{ik} x_i\right) + \nu_k\right] \ . \tag{49}$$

We can also include data weighting as in the case of a single Gaussian. The weights just multiply the posteriors:

$$\widetilde{\eta}_{ik} = \eta_{ik}\, w_i \ . \tag{50}$$

6 Wrapped Gaussian Mixture Model (WGMM)

We can also derive a procedure for fitting a GMM on a torus.

6.1 Univariate WGMM

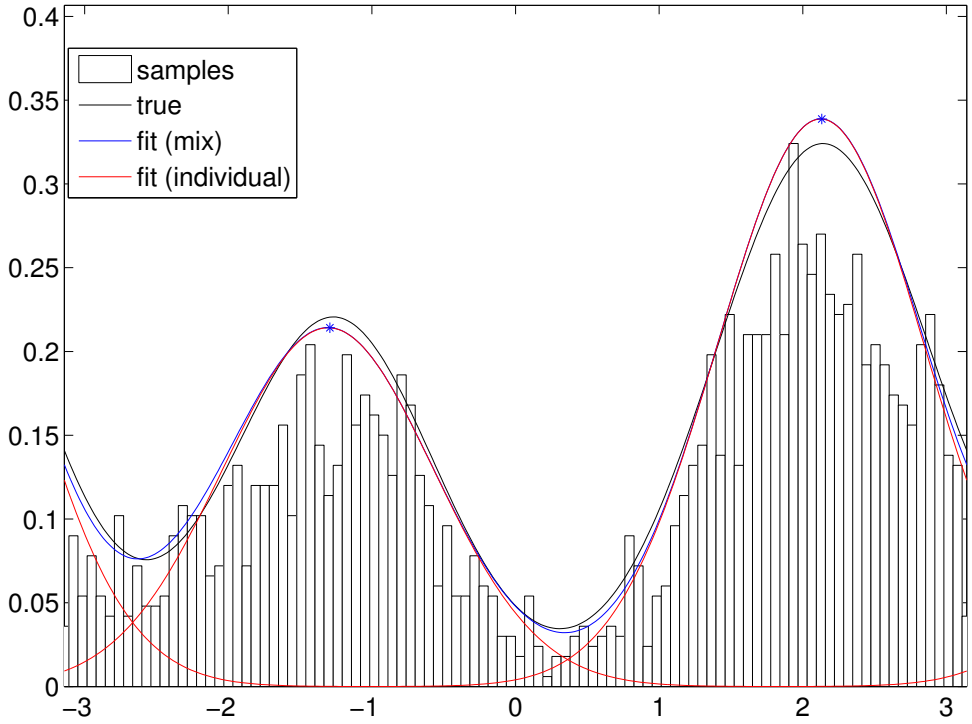

Figure 4: Mixture of univariate wrapped Gaussians (WG) fit to dataset. WG components (red) linearly combine to form a mixture (blue) that describes the distribution of the data (bars).

In the 1D case, this is just a circle. This is useful for when we have data that lies on a circular axis in the range $[-\pi, \pi]$. The EM update equations are derived as in the regular

GMM case.

Likelihood:

$$\mathscr{L} = \prod_{i=1}^{N} \sum_{j=1}^{k} \pi_j w \mathcal{N}(x_i; \mu_j, \sigma_j^2) \tag{51}$$

$$\mathscr{L} = \prod_{i=1}^{N} \sum_{j=1}^{k} \sum_{l=-\infty}^{\infty} \pi_j \mathcal{N}(x_i; \mu_j + 2\pi l, \sigma_j^2) \tag{52}$$

$$\log \mathscr{L} = \sum_{i=1}^{N} \log \sum_{j=1}^{k} \sum_{l=-\infty}^{\infty} \pi_j \mathcal{N}(x_i; \mu_j + 2\pi l, \sigma_j^2) \tag{53}$$

Q function:

$$Q = \sum_{i=1}^{N} \sum_{j=1}^{k} \sum_{l=-\infty}^{\infty} \left(\log \left[\pi_j \mathcal{N}(x_i; \mu_j + 2\pi l, \sigma_j^2) \right] \right) \eta_{ijl} \tag{54}$$

$$= \sum_{i=1}^{N} \sum_{j=1}^{k} \sum_{l=-\infty}^{\infty} \left(\log(\pi_j) - \frac{1}{2}\log(2\pi) - \frac{1}{2}\log(\sigma_j^2) - \frac{(x_i - \mu_j - 2\pi l)^2}{2\sigma_j^2} \right) \eta_{ijl} \ , \tag{55}$$

$$\forall\, i \quad \sum_{j=1}^{k} \sum_{l=-\infty}^{\infty} \eta_{ijl} = 1 \tag{56}$$

Partial derivatives:

$$\frac{\partial Q}{\partial \mu_j} = \sum_{i=1}^{N} \sum_{l=-\infty}^{\infty} \left(\frac{(x_i - \mu_j - 2\pi l)}{\sigma_j^2} \right) \eta_{ijl} = 0 \tag{57}$$

$$\frac{\partial Q}{\partial \sigma_j^2} = \sum_{i=1}^{N} \sum_{l=-\infty}^{\infty} \left(-\frac{1}{2\sigma_j^2} + \frac{(x_i - \mu_j - 2\pi l)}{2(\sigma_j^2)^2} \right) \eta_{ijl} = 0 \tag{58}$$

$$\frac{\partial \left(Q + \lambda \left(\sum_{j=1}^{k} \pi_j - 1 \right) \right)}{\partial \pi_j} = \sum_{i=1}^{N} \sum_{l=-\infty}^{\infty} \frac{1}{\pi_j} \eta_{ijl} + \lambda = 0 \qquad , \qquad \sum_{j=1}^{k} \pi_j = 1 \tag{59}$$

Update rules:

$$\widehat{\mu}_j = \frac{\sum_{i=1}^{N} \sum_{l=-\infty}^{\infty} (x_i - 2\pi l)\, \eta_{ijl}}{\sum_{i=1}^{N} \sum_{l=-\infty}^{\infty} \eta_{ijl}} \tag{60}$$

$$\widehat{\sigma}_j^2 = \frac{\sum_{i=1}^{N} \sum_{l=-\infty}^{\infty} (x_i - \mu_j - 2\pi l)^2\, \eta_{ijl}}{\sum_{i=1}^{N} \sum_{l=-\infty}^{\infty} \eta_{ijl}} \tag{61}$$

$$\widehat{\pi}_j = \frac{1}{N} \sum_{i=1}^{N} \sum_{l=-\infty}^{\infty} \eta_{ijl} \tag{62}$$

In practice, we can't evaluate expressions with an infinite number of terms numerically, so the WG's need to be truncated after a sufficient number of terms. This involves replacing all $\sum_{l=-\infty}^{\infty}(-)$ with $\sum_{l=-L}^{L}(-)$.

6.2 Bivariate WGMM

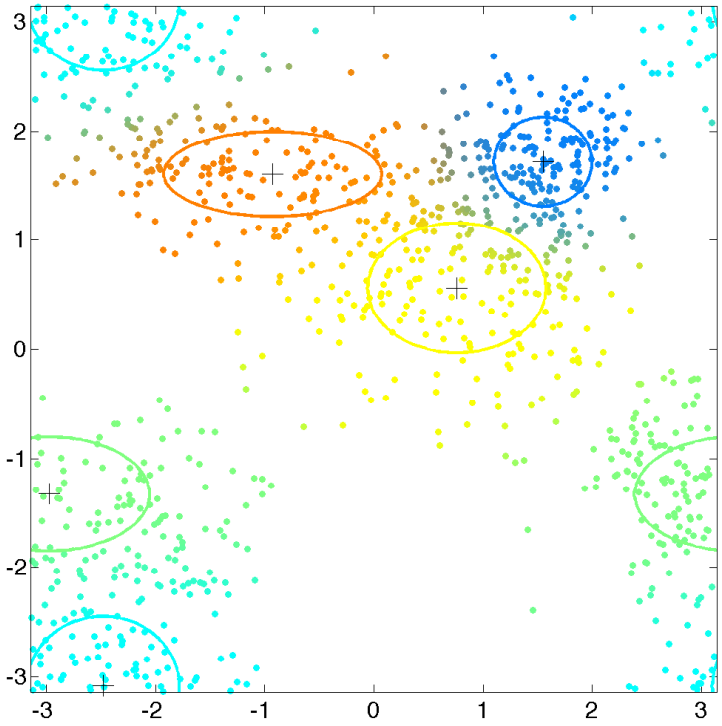

Figure 5: Mixture of bivariate wrapped Gaussians (WG) fit to dataset. Data points are colored by posterior probability.

When there are multiple circular axes to consider, we can make use of the multivariate WG distribution. For the case of two dimensions, we have:

$$P(\mathbf{x};\boldsymbol{\mu},\boldsymbol{\Sigma}) = \sum_{l_1,l_2=-\infty}^{\infty} \mathcal{N}\left(\mathbf{x};\boldsymbol{\mu}+2\pi\begin{bmatrix}l_1\\l_2\end{bmatrix},\boldsymbol{\Sigma}\right) \quad , \quad \mathbf{x}\in\mathbb{S}^2 \ . \tag{63}$$

Likelihood:

$$\mathscr{L} = \prod_{i=1}^{N}\sum_{j=1}^{k}\pi_j \sum_{l_1,l_2=-\infty}^{\infty} \mathcal{N}\left(\mathbf{x}_i;\boldsymbol{\mu}_j+2\pi\begin{bmatrix}l_1\\l_2\end{bmatrix},\boldsymbol{\Sigma}_j\right) \tag{64}$$

$$\log\mathscr{L} = \sum_{i=1}^{N}\log\sum_{j=1}^{k}\pi_j \sum_{l_1,l_2=-\infty}^{\infty} \mathcal{N}\left(\mathbf{x}_i;\boldsymbol{\mu}_j+2\pi\begin{bmatrix}l_1\\l_2\end{bmatrix},\boldsymbol{\Sigma}_j\right) \ . \tag{65}$$

Q function:

$$Q = \sum_{i=1}^{N}\sum_{j=1}^{k}\sum_{l_1,l_2=-\infty}^{\infty} \log\left[\pi_j \mathcal{N}\left(\mathbf{x}_i; \boldsymbol{\mu}_j + 2\pi\begin{bmatrix} l_1 \\ l_2 \end{bmatrix}, \boldsymbol{\Sigma}_j\right)\right] \eta_{ijl_1l_2} \tag{66}$$

$$= \sum_{i=1}^{N}\sum_{j=1}^{k}\sum_{l_1,l_2=-\infty}^{\infty} \left[\log(\pi_j) - \frac{1}{2}\log(|2\pi\boldsymbol{\Sigma}|) - \frac{1}{2}\left(\mathbf{x}_i - \boldsymbol{\mu}_j - 2\pi\begin{bmatrix} l_1 \\ l_2 \end{bmatrix}\right)^T \boldsymbol{\Sigma}_j^{-1}\left(\mathbf{x}_i - \boldsymbol{\mu}_j - 2\pi\begin{bmatrix} l_1 \\ l_2 \end{bmatrix}\right)\right] \eta_{ijl_1l_2} \ . \tag{67}$$

$$\eta_{ijl_1l_2} = \frac{\pi_j \mathcal{N}\left(\mathbf{x}_i; \boldsymbol{\mu}_j + 2\pi\begin{bmatrix} l_1 \\ l_2 \end{bmatrix}, \boldsymbol{\Sigma}_j\right)}{\sum_{j=1}^{k}\sum_{l_1,l_2=-\infty}^{\infty} \pi_j \mathcal{N}\left(\mathbf{x}_i; \boldsymbol{\mu}_j + 2\pi\begin{bmatrix} l_1 \\ l_2 \end{bmatrix}, \boldsymbol{\Sigma}_j\right)} \ . \tag{68}$$

Partial derivatives:

$$\frac{\partial Q}{\partial \boldsymbol{\mu}_j} \propto \sum_{i=1}^{N}\sum_{l_1,l_2=-\infty}^{\infty} \boldsymbol{\Sigma}_j^{-1}\left(\mathbf{x}_i - \boldsymbol{\mu}_j - 2\pi\begin{bmatrix} l_1 \\ l_2 \end{bmatrix}\right) \eta_{ijl_1l_2} = \mathbf{0} \ , \tag{69}$$

$$\frac{\partial Q}{\partial \boldsymbol{\Sigma}_j} \propto \sum_{i=1}^{N}\sum_{l_1,l_2=-\infty}^{\infty} \left[-\boldsymbol{\Sigma}_j^{-1} + \boldsymbol{\Sigma}_j^{-1}\left(\mathbf{x}_i - \boldsymbol{\mu}_j - 2\pi\begin{bmatrix} l_1 \\ l_2 \end{bmatrix}\right)\left(\mathbf{x}_i - \boldsymbol{\mu}_j - 2\pi\begin{bmatrix} l_1 \\ l_2 \end{bmatrix}\right)^T \boldsymbol{\Sigma}_j^{-1}\right] \eta_{ijl_1l_2} = \mathbf{0} \ , \tag{70}$$

$$\frac{\partial\left(Q + \lambda\left(\sum_{j=1}^{k} -1\right)\right)}{\partial \pi_j} = \sum_{i=1}^{N}\sum_{l_1,l_2=-\infty}^{\infty} \frac{1}{\pi_j}\eta_{ijl_1l_2} + \lambda = 0 \ . \tag{71}$$

Update rules:

$$\widehat{\boldsymbol{\mu}}_j = \frac{\sum_{i=1}^{N}\sum_{l_1,l_2=-\infty}^{\infty}\left(\mathbf{x}_i - 2\pi\begin{bmatrix} l_1 \\ l_2 \end{bmatrix}\right)\eta_{ijl_1l_2}}{\sum_{i=1}^{N}\sum_{l_1,l_2=-\infty}^{\infty}\eta_{ijl_1l_2}} \ , \tag{72}$$

$$\widehat{\boldsymbol{\Sigma}}_j = \frac{\sum_{i=1}^{N}\sum_{l_1,l_2=-\infty}^{\infty}\left(\mathbf{x}_i - \widehat{\boldsymbol{\mu}}_j - 2\pi\begin{bmatrix} l_1 \\ l_2 \end{bmatrix}\right)\left(\mathbf{x}_i - \widehat{\boldsymbol{\mu}}_j - 2\pi\begin{bmatrix} l_1 \\ l_2 \end{bmatrix}\right)^T \eta_{ijl_1l_2}}{\sum_{i=1}^{N}\sum_{l_1,l_2=-\infty}^{\infty}\eta_{ijl_1l_2}} \ , \tag{73}$$

$$\widehat{\pi}_j = \frac{1}{N}\sum_{i=1}^{N}\sum_{l_1,l_2=-\infty}^{\infty}\eta_{ijl_1l_2} \ . \tag{74}$$

7 von Mises Mixture Model (vMMM)

We can also cluster on the unit circle with the von Mises distribution, whose pdf is:

$$vM\left(x;\mu_j,\kappa_j\right) = \frac{1}{2\pi I_0(\kappa_j)} e^{\kappa_j \cos(x-\mu_j)} \quad . \tag{75}$$

Because the vM has a $\cos(-)$ term, we will have to numerically update the concentration parameter κ. Otherwise, the derivation is standard.

7.1 Univariate vMMM

Likelihood:

$$\mathscr{L} = \prod_{i=1}^{N} \sum_{j=1}^{k} \pi_j \, vM(x_i;\mu_j,\kappa_j) \tag{76}$$

$$\log \mathscr{L} = \sum_{i=1}^{N} \log \sum_{j=1}^{k} \pi_j \, vM(x_i;\mu_j,\kappa_j) \tag{77}$$

Q function:

$$Q = \sum_{i=1}^{N} \sum_{j=1}^{k} \log\left[\pi_j \, vM(x_i;\mu_j,\kappa_j)\right] \eta_{ij} \tag{78}$$

$$= \sum_{i=1}^{N} \sum_{j=1}^{k} \log\left[\pi_j \frac{1}{2\pi I_0(\kappa_j)} e^{\kappa_j \cos(x_i-\mu_j)}\right] \eta_{ij} \tag{79}$$

$$= \sum_{i=1}^{N} \sum_{j=1}^{k} \left[\log(\pi_j) - \log(2\pi) - \log(I_0(\kappa_j)) + \kappa_j \cos(x_i-\mu_j)\right] \eta_{ij} \tag{80}$$

$$\forall\, i \quad \sum_{j=1}^{k} \eta_{ij} = 1 \tag{81}$$

$I_0(-)$ is the 0^{th}-order modified Bessel function of the first kind.

Partial derivatives:

$$\frac{\partial Q}{\partial \mu_j} = \sum_{i=1}^{N} \left[\kappa_j \sin(x_i-\mu_j)\right] \eta_{ij} \tag{82}$$

$$= \kappa_j \sum_{i=1}^{N} \left[\sin(x_i)\cos(\mu_j) - \cos(x_i)\sin(\mu_j)\right] \eta_{ij} = 0 \tag{83}$$

$$\frac{\partial Q}{\partial \kappa_j} = \sum_{i=1}^{N} \left[-\frac{I_1(\kappa_j)}{I_0(\kappa_j)} + \cos(x_i-\mu_j)\right] \eta_{ij} \tag{84}$$

$$= \sum_{i=1}^{N} \left[-A(\kappa_j) + \cos(x_i-\mu_j)\right] \eta_{ij} = 0 \tag{85}$$

$$\frac{\partial\left(Q + \lambda\left(\sum_{j=1}^{k} \pi_j - 1\right)\right)}{\partial \pi_j} = \sum_{i=1}^{N} \frac{1}{\pi_j} \eta_{ij} + \lambda = 0 \qquad , \qquad \sum_{j=1}^{k} \pi_j = 1 \tag{86}$$

Update rules:

$$\widehat{\mu}_j = \tan^{-1}\left(\frac{\sum_{i=1}^{N}\sin(x_i)\,\eta_{ij}}{\sum_{i=1}^{N}\cos(x_i)\,\eta_{ij}}\right) \tag{87}$$

$$A\left(\widehat{\kappa}_j\right) = \frac{\sum_{i=1}^{N}\cos(x_i-\mu_j)\,\eta_{ij}}{\sum_{i=1}^{N}\eta_{ij}} \quad , \quad A\left(\widehat{\kappa}_j\right) = \frac{I_1(\kappa_j)}{I_0(\kappa_j)} \tag{88}$$

$$\widehat{\pi}_j = \frac{1}{N}\sum_{i=1}^{N}\eta_{ij} \tag{89}$$

We can solve for κ_j with a standard zero-finder (e.g. bisection search). Notice that the vM distribution has wrapping built into its definition, whereas a truncated wG is a good approximation.

8 von Mises-Fisher Mixture Model (vMFMM)

The von Mises-Fisher is a convenient distribution for modeling uncertainty on the unit 2-sphere. The pdf is parameterized by a mean direction $\boldsymbol{\mu}$ and concentration κ.

$$P(\mathbf{x};\boldsymbol{\mu},\kappa) = \frac{\kappa}{2\pi(e^{\kappa}-e^{-\kappa})}e^{\kappa\boldsymbol{\mu}^T\mathbf{x}} \quad , \quad \|\boldsymbol{\mu}\|_2 = 1 \tag{90}$$

Likelihood:

$$\mathscr{L} = \prod_{i=1}^{N}\sum_{j=1}^{k}\pi_j\, vMF(\mathbf{x}_i;\boldsymbol{\mu}_j,\kappa_j) \tag{91}$$

$$\log\mathscr{L} = \sum_{i=1}^{N}\log\sum_{j=1}^{k}\pi_j\, vMF(\mathbf{x}_i;\boldsymbol{\mu}_j,\kappa_j) \tag{92}$$

Q function:

$$Q = \sum_{i=1}^{N}\sum_{j=1}^{k}\log\left[\pi_j\, vMF(\mathbf{x}_i;\boldsymbol{\mu}_j,\kappa_j)\right]\eta_{ij} \tag{93}$$

$$= \sum_{i=1}^{N}\sum_{j=1}^{k}\log\left[\pi_j\frac{\kappa_j}{2\pi(e^{\kappa_j}-e^{-\kappa_j})}e^{\kappa_j\boldsymbol{\mu}_j^T\mathbf{x}_i}\right]\eta_{ij} \tag{94}$$

$$= \sum_{i=1}^{N}\sum_{j=1}^{k}\left[\log(\pi_j)+\log(\kappa_j)-\log(2\pi)-\log(e^{\kappa_j}-e^{-\kappa_j})+\kappa_j\boldsymbol{\mu}_j^T\mathbf{x}_i\right]\eta_{ij} \tag{95}$$

$$\forall\, i \quad \sum_{j=1}^{k}\eta_{ij} = 1 \tag{96}$$

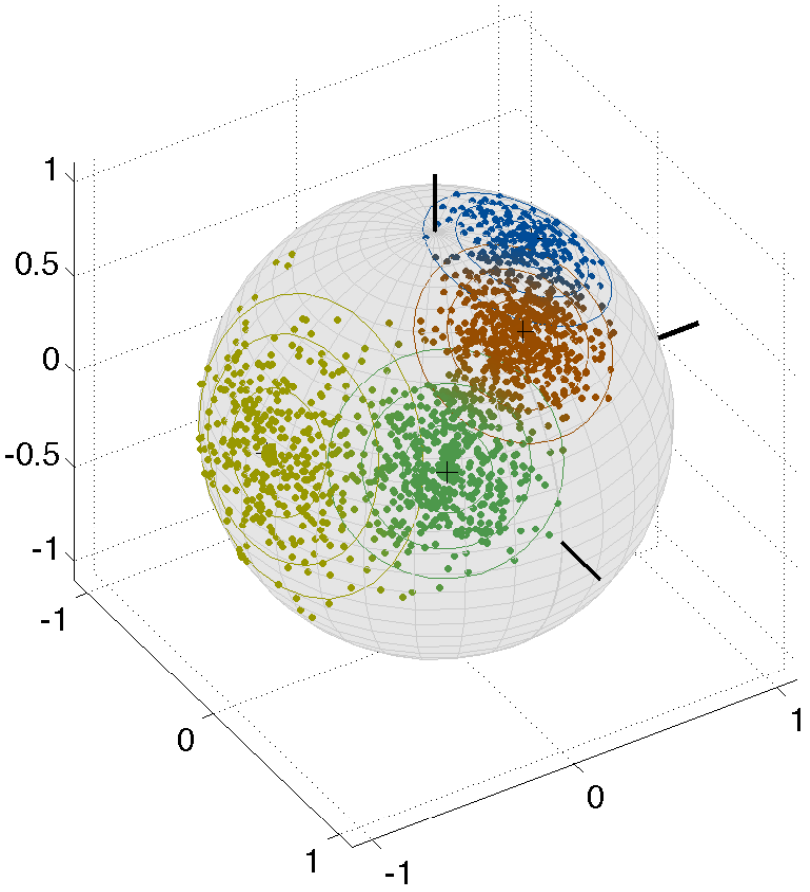

Figure 6: Mixture of von Mises-Fisher distributions on the sphere. von Mises-Fisher distributions are denoted by their mean $\boldsymbol{\mu}$ (black '+') and concentration κ (ellipses). Data points are colored by their posterior probabilities η_{ij}.

Partial derivatives:

$$\frac{\partial\left(Q+\lambda\left(1-\boldsymbol{\mu}_j^T\boldsymbol{\mu}_j\right)\right)}{\partial\boldsymbol{\mu}_j}=\sum_{i=1}^{N}\kappa_j\,\mathbf{x_i}\,\eta_{ij}-\lambda\boldsymbol{\mu}_j=0\quad,\quad\boldsymbol{\mu}_j^T\boldsymbol{\mu}_j=1 \tag{97}$$

$$\frac{\partial Q}{\partial\kappa_j}=\sum_{i=1}^{N}\left[\frac{1}{\kappa_j}-\frac{e^{\kappa_j}+e^{-\kappa_j}}{e^{\kappa_j}-e^{-\kappa_j}}+\boldsymbol{\mu}_j^T\mathbf{x}_i\right]\eta_{ij}=0 \tag{98}$$

$$\frac{\partial\left(Q+\lambda\left(\sum_{j=1}^{k}\pi_j-1\right)\right)}{\partial\pi_j}=\sum_{i=1}^{N}\frac{1}{\pi_j}\eta_{ij}+\lambda=0\qquad,\qquad\sum_{j=1}^{k}\pi_j=1 \tag{99}$$

Update rules:

$$\widehat{\boldsymbol{\mu}}_j = \frac{\sum_{i=1}^{N} \mathbf{x}_i \eta_{ij}}{\left\| \sum_{i=1}^{N} \mathbf{x}_i \eta_{ij} \right\|_2} \tag{100}$$

$$A\left(\widehat{\kappa}_j\right) = \frac{e^{\widehat{\kappa}_j} + e^{-\widehat{\kappa}_j}}{e^{\widehat{\kappa}_j} - e^{-\widehat{\kappa}_j}} - \frac{1}{\widehat{\kappa}_j} = \frac{\widehat{\boldsymbol{\mu}}_j^T \sum_{i=1}^{N} \mathbf{x}_i \eta_{ij}}{\sum_{i=1}^{N} \eta_{ij}} = \frac{\left\| \sum_{i=1}^{N} \mathbf{x}_i \eta_{ij} \right\|_2}{\sum_{i=1}^{N} \eta_{ij}} \tag{101}$$

$$\widehat{\pi}_j = \frac{1}{N} \sum_{i=1}^{N} \eta_{ij} \tag{102}$$

The update of the concentration parameters is a pain, but there are good approximations. For $\kappa \gg 3$ and large $A(\widehat{\kappa}_j)$, the following can be used as an update (approximation 10.3.7 from Mardia and Jupp (2000, pg. 198):

$$\widehat{\kappa}_j \approx \frac{1}{1 - A(\widehat{\kappa}_j)} \ . \tag{103}$$

Even when the conditions of the approximation are not met, the clustering is sufficiently stable and accurate.

9 Line Mixture Model (LineMM)

Here we derive the update equations for fitting a mixture of lines to a 2D dataset. In essence, we place 1D Gaussian distributions on each data point and measure error (negative likelihood) evaluated at the lines in the (vertical) y-coordinate. This is the multi-line extension of linear regression. The mixture model likelihood has the form:

$$\mathscr{L} = \prod_{i=1}^{N} \sum_{k=1}^{K} \pi_k \mathcal{N}\left(y_i | f_k(x_i), \sigma_k^2\right) \ , \tag{104}$$

$$\log \mathscr{L} = \sum_{i=1}^{N} \log \sum_{k=1}^{K} \pi_k \frac{1}{\sqrt{2\pi\sigma_k^2}} e^{-\frac{\left(y_i - f_k(x_i)\right)^2}{2\sigma_k^2}} \ , \tag{105}$$

where

$$f_k(x_i) = a_k x_i + b_k \ . \tag{106}$$

The Q function is:

$$Q \propto \sum_{i=1}^{N} \sum_{k=1}^{K} \left[\log \pi_k - \frac{1}{2} \log\left(\sigma_k^2\right) - \frac{\left(y_i - (a_k x_i + b_k)\right)^2}{2\sigma_k^2} \right] \eta_{ik} \ . \tag{107}$$

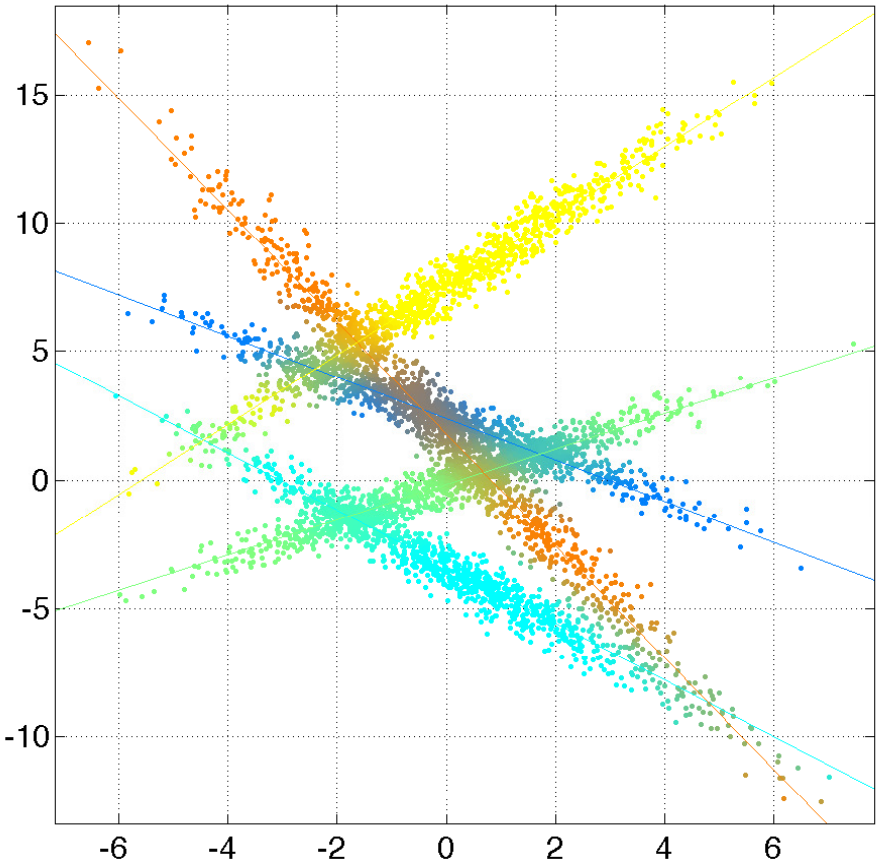

Figure 7: Mixture of lines. Data points are colored by their posterior probabilities η_{ij}.

Taking derivatives,

$$\frac{\partial Q}{\partial a_k} \propto \sum_{i=1}^{N} \left(y_i - a_k x_i - b_k\right) x_i \eta_{ik} = 0 \ , \tag{108}$$

$$\frac{\partial Q}{\partial b_k} \propto \sum_{i=1}^{N} \left(y_i - a_k x_i - b_k\right) \eta_{ik} = 0 \ , \tag{109}$$

$$\frac{\partial Q}{\partial \sigma_k^2} \propto \sum_{i=1}^{N} \left[-\frac{1}{2\sigma_k^2} + \frac{\left(y_i - (a_k x_i + b_k)\right)^2}{\left(\sigma_k^2\right)^2} \right] \eta_{ik} = 0 \tag{110}$$

$$\frac{\partial \left(Q + \lambda \left(\sum_{k=1}^{K} \pi_k - 1 \right) \right)}{\partial \pi_k} \propto \sum_{i=1}^{N} \frac{\eta_{ik}}{\pi_k} + \lambda = 0 \qquad , \qquad \sum_{k=1}^{K} \pi_k = 1 \ . \tag{111}$$

Re-arranging and solving for the parameters gives

$$\widehat{a}_k = \frac{\sum_{i=1}^{N} x_i \left(y_i - b_k\right) \eta_{ik}}{\sum_{i=1}^{N} x_i^2 \, \eta_{ik}} \ , \tag{112}$$

$$\widehat{b}_k = \frac{\sum_{i=1}^{N} \left(y_i - a_k x_i\right) \eta_{ik}}{\sum_{i=1}^{N} \eta_{ik}} \ , \tag{113}$$

$$\widehat{\sigma}_k^2 = \frac{\sum_{i=1}^{N} \left(y_i - (a_k x_i + b_k)\right)^2 \eta_{ik}}{\sum_{i=1}^{N} \eta_{ik}} \ , \tag{114}$$

$$\widehat{\pi}_k = \frac{1}{N} \sum_{i=1}^{N} \eta_{ik} \ . \tag{115}$$

10 Laplacian Mixture Model (LapMM)

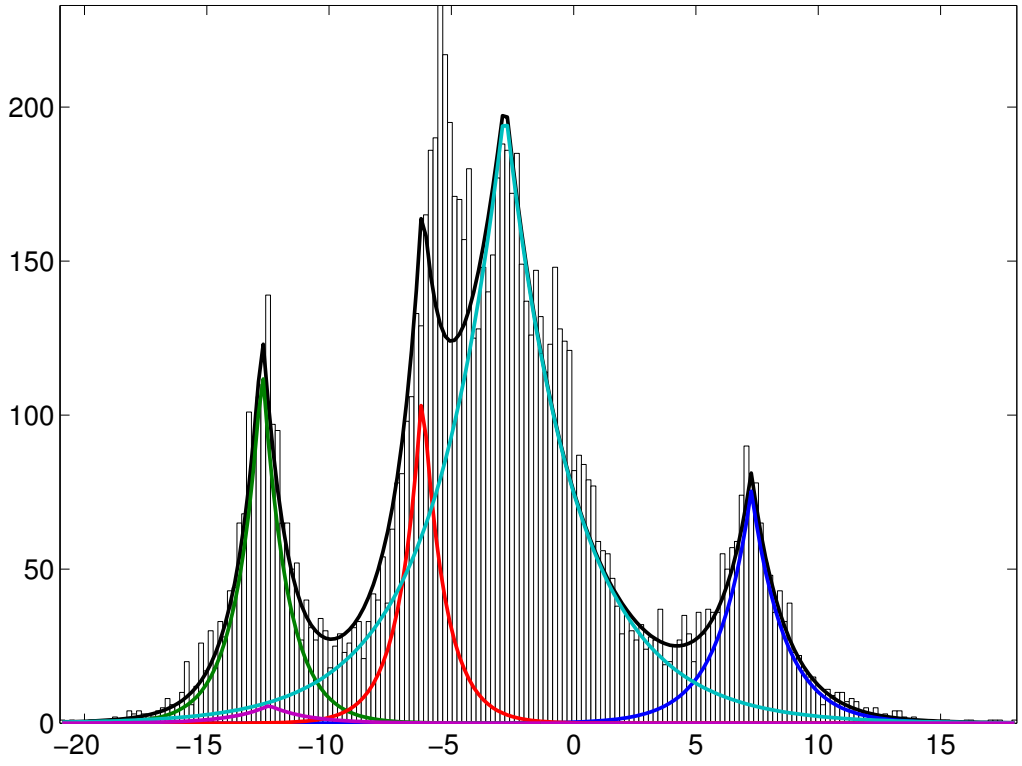

Figure 8: Mixture of Laplacian distributions.

Now we derive the update equations for the M-step to fit a 1D Mixture of Laplacian distributions. This is very much like the derivation for the GMM except that the distribution used is a Laplacian rather than a Gaussian.

$$\mathscr{L} = \prod_{i=1}^{N} \sum_{k=1}^{K} \pi_k L(x_i|\mu_k, b_k) \tag{116}$$

$$\log \mathscr{L} = \sum_{i=1}^{N} \log \sum_{k=1}^{K} \pi_k \frac{1}{2b_k} e^{-\frac{|x_i-\mu_k|}{b_k}} \tag{117}$$

$$Q \propto \sum_{i=1}^{N} \sum_{k=1}^{K} \left[\log \pi_k - \log b_k - \frac{|x_i - \mu_k|}{b_k} \right] \eta_{ik} \ . \tag{118}$$

Taking derivatives,

$$\frac{\partial Q}{\partial \mu_k} \propto \sum_{i=1}^{N} \eta_{ik} \frac{x_i - \mu_k}{|x_i - \mu_k|} = 0 \ , \tag{119}$$

$$\frac{\partial Q}{\partial b_k} \propto \sum_{i=1}^{N} \frac{1}{b_k} \eta_{ik} = 0 \ , \tag{120}$$

$$\frac{\partial \left(Q + \lambda \left(\sum_{k=1}^{K} \pi_k - 1 \right) \right)}{\partial \pi_k} \propto \sum_{i=1}^{N} \frac{1}{\pi_k} + \lambda = 0 \qquad , \qquad \sum_{k=1}^{K} \pi_k = 1 \ . \tag{121}$$

We need to assume that the denominator in equation (119) is a constant for each i and k to continue. This leads to a stable algorithm in practice. Re-arranging and solving for the parameters gives

$$\mu_k = \frac{\sum_{i=1}^{N} \eta_{ik} \frac{x_i}{|x_i - \mu_j|}}{\sum_{i=1}^{N} \eta_{ik} \frac{1}{|x_i - \mu_j|}} \ , \tag{122}$$

$$b_k = \frac{\sum_{i=1}^{N} \eta_{ik} |x_i - \mu_k|}{\sum_{i=1}^{N} \eta_{ik}} \ , \tag{123}$$

$$\pi_k = \frac{1}{N} \sum_{i=1}^{N} \eta_{ik} \ . \tag{124}$$

11 Probabilistic Latent Semantic Indexing (PLSI)

PLSI is a model that represents data in the probability (canonical) simplex as convex combinations of categorical distributions referred to as “basis vectors," “endmembers," and “topics." Here, we look at a derivation of the PLSI update equations used by Hofmann in

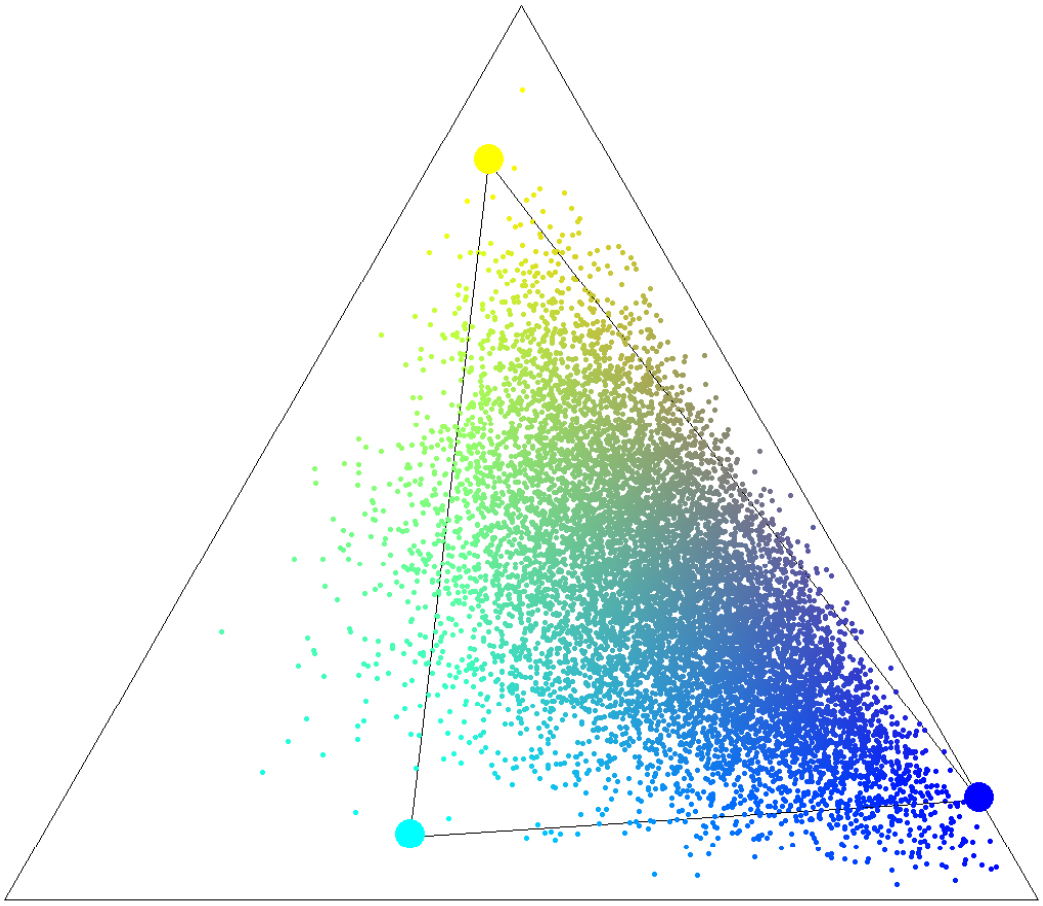

Figure 9: PLSI clustering on a 3-component probability simplex. Data is colored by its activation weight $P(i|k)$.

his paper. This is not a conventional application of the EM algorithm, but the details can be obscured without effecting the derivation.

We consider the following factorization:

$$P(i,j) = \sum_{k=1}^{K} P(i,j|k)P(k) = \sum_{k=1}^{K} P(i|k)P(j|k)P(k) \ , \tag{125}$$

where i is the word (dimension) index and j is the document (data point) index. The word and document variables are assumed to be independent given the latent variable z, which captures which topic generated each data point. The log likelihood is just the negative cross-entropy between an observed data distribution $X(i,j)$ and its reconstruction $P(i,j)$ under the (symmetric) PLSI model:

$$\mathscr{L} = \sum_{i=1}^{N}\sum_{j=1}^{D} X(i,j)\log P(i,j) \tag{126}$$

$$= \sum_{i=1}^{N}\sum_{j=1}^{D} X(i,j)\log \sum_{k=1}^{K} P(i|k)P(j|k)P(k) \ , \tag{127}$$

where we require that

$$\forall k \ \sum_{i=1}^{N} P(i|k) = 1 \ , \tag{128}$$

$$\forall k \ \sum_{j=1}^{D} P(j|k) = 1 \ , \tag{129}$$

$$\sum_{k=1}^{K} P(k) = 1 \ . \tag{130}$$

The Q function is:

$$Q = \sum_{i=1}^{N} \sum_{j=1}^{D} X(i,j) \sum_{k=1}^{K} \log\left[P(i|z)P(j|z)P(k)\right] P(k|i,j) \tag{131}$$

$$= \sum_{i=1}^{N} \sum_{j=1}^{D} X(i,j) \sum_{k=1}^{K} \left[\log P(i|z) + \log P(j|z) + \log P(k)\right] P(k|i,j) \ , \tag{132}$$

and the posterior is given by:

$$P(k|i,j) = \frac{P(i,j,k)}{P(i,j)} = \frac{P(i|k)P(j|k)P(k)}{\sum\limits_{k=1}^{K} P(i|k)P(j|k)P(k)} \ . \tag{133}$$

Taking partial derivatives with the appropriate Lagrange multiplier expressions, we get:

$$\frac{\partial Q + \lambda\left(\sum\limits_{i=1}^{N} P(i|k) - 1\right)}{\partial P(i|k)} = \sum_{j=1}^{D} \frac{X(i,j)}{P(i|k)} P(k|i,j) + \lambda = 0 \ , \tag{134}$$

$$\frac{\partial Q + \lambda\left(\sum\limits_{j=1}^{D} P(j|k) - 1\right)}{\partial P(j|k)} = \sum_{i=1}^{N} \frac{X(i,j)}{P(j|k)} P(k|i,j) + \lambda = 0 \ , \tag{135}$$

$$\frac{\partial Q + \lambda\left(\sum\limits_{k=1}^{K} P(k) - 1\right)}{\partial P(k)} = \sum_{i=1}^{N} \sum_{j=1}^{D} \frac{X(i,j)}{P(k)} P(k|i,j) + \lambda = 0 \ . \tag{136}$$

Solving for the parameters $P(i|k)$, $P(j|k)$, and $P(k)$, and plugging these expressions into the constraint equations (128)-(130), we can solve for the Lagrange multipliers and substitute them into equations (134)-(136). This gives the PLSI updates for the M-step:

$$P(i|k) = \frac{\sum_{j=1}^{D} X(i,j)P(k|i,j)}{\sum_{i=1}^{N}\sum_{j=1}^{D} X(i,j)P(k|i,j)} \ , \tag{137}$$

$$P(j|k) = \frac{\sum_{i=1}^{N} X(i,j)P(k|i,j)}{\sum_{i=1}^{N}\sum_{j=1}^{D} X(i,j)P(k|i,j)} \ , \tag{138}$$

$$P(k) = \sum_{i=1}^{N}\sum_{j=1}^{D} X(i,j)P(k|i,j) \ . \tag{139}$$

11.1 MULTIPLICATIVE UPDATES

We can re-arrange (137)-(139) for efficient computation using matrix algebra. First, we collapse the last two terms into one:

$$P(j,k) = P(j|k)P(k) = \sum_{i=1}^{N} X(i,j)P(k|i,j) \ , \tag{140}$$

and plug the E step (posteriors) into the M step to get:

$$P(i|k) = \frac{\sum_{j=1}^{D} Y(i,j)P(i,j,k)}{P(k)} = \frac{P(i|k)\sum_{j=1}^{D} Y(i,j)P(j,k)}{\sum_{j=1}^{D} P(j,k)} \ , \tag{141}$$

$$P(j,k) = \sum_{i=1}^{N} Y(i,j)P(i,j,k) = P(j,k)\sum_{i=1}^{N} Y(i,j)P(i|k) \ , \tag{142}$$

where

$$Y(i,j) = \frac{X(i,j)}{P(i,j)} \ . \tag{143}$$

Writing this in matrix notation, we get:

$$W = \frac{W \odot \left(YH^T\right)}{JH^T} \ . \quad , \quad J = ones(N,D) \quad , \tag{144}$$

$$H = H \odot \left(W^T Y\right) \ , \tag{145}$$

where $W \in \mathbb{R}^{N\times K}$ and $H \in \mathbb{R}^{K\times D}$ are the probabilities $P(i|k)$ and $P(j,k)$ arranged into matrices such that $P = WH \in \mathbb{R}^{N\times D}$ is equivalent to $P(i,j) = \sum_{k=1}^{K} P(i|k)P(j,k)$. We can recover $P(i|k)$ from W and both $P(j|k)$ and $P(k)$ from H. These updates look suspiciously similar to the NMF updates derived by Lee and Seung for the KL-divergence error criterion. The two algorithms are actually equivalent up to a scaling factor. In practice, the H matrix should be normalized explicitly.