Mixture models and the EM algorithm

Qingyun Yu, Ke Zhang

School of Mathematics
Sun Yat-sen University

September 23, 2020

Table of Contents

Mixture models

- Suppose $z_i \in \{1, \ldots, K\}$, representing a discrete latent state, $p(z_i) = Multi(\pi)$. For the likelihood, we use $p(\mathbf{x}_i \mid z_i = k) = p_k(\mathbf{x}_i)$, where p_k is the k 'th base distribution for the observations.
- We are mixing together the K base distributions as follows:

  $$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k p_k(\mathbf{x}).$$

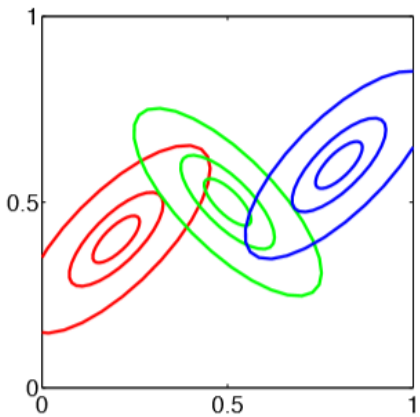
  This is a convex combination of the p_k 's, π_k satisfy $0 \leq \pi_k \leq 1$ and $\sum_{k=1}^{K} \pi_k = 1$.

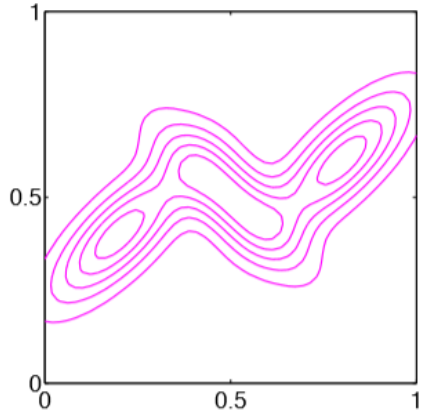
Mixtures of Gaussians(GMM)

- Each base distribution in the mixture is a multivariate Gaussian with mean $\boldsymbol{\mu}_k$ and covariance matrix $\boldsymbol{\Sigma}_k$.

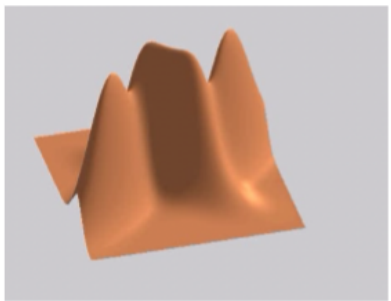
$$p\left(\mathbf{x}_i \mid \boldsymbol{\theta}\right) = \sum_{k=1}^{K} \pi_k \mathcal{N}\left(\mathbf{x}_i \mid \boldsymbol{\mu}_k, \ _k\right).$$

$p(\mathbf{x}|k)$





$p(\mathbf{x})$



$p(\mathbf{x})$

Using mixture models for clustering

- Use them for clustering compute $p(z_i = k \mid \mathbf{x}_i, \boldsymbol{\theta})$, which represents the posterior probability that point i belongs to cluster k .

$$r_{ik} \triangleq p(z_i = k \mid \mathbf{x}_i, \boldsymbol{\theta}) \propto p(z_i = k \mid \boldsymbol{\theta})\, p(\mathbf{x}_i \mid z_i = k, \boldsymbol{\theta}),$$

Assuming $1 - \max_k r_{ik}$ is small, it may be reasonable to compute a hard clustering using the MAP estimate, given by,

$$z_i^* = \arg\max_k r_{ik} = \arg\max_k \log p(\mathbf{x}_i \mid z_i = k, \boldsymbol{\theta}) + \log p(\mathbf{z}_i = k \mid \boldsymbol{\theta}).$$

Note that z_i^* is designed on θ.

Review: LDA

Assume that the conditional probability density functions $p(\vec{x} \mid y = k), k = 1, 2, \ldots, K$ are both normally distributed with mean and covariance parameters $\theta = (\vec{\mu}_k, \Sigma)$, respectively.
The Bayes optimal solution:

$$p(y = c \mid \mathbf{x}, \boldsymbol{\theta}) \propto \pi_c \exp\left[\boldsymbol{\mu}_c^T {}^{-1}\mathbf{x} - \frac{1}{2}\mathbf{x}^T {}^{-1}\mathbf{x} - \frac{1}{2}\boldsymbol{\mu}_c^T {}^{-1}\boldsymbol{\mu}_c\right].$$

To class a new sample, we need to calculate the parameters θ.
If label missing?
Clustering.

An example for clustering

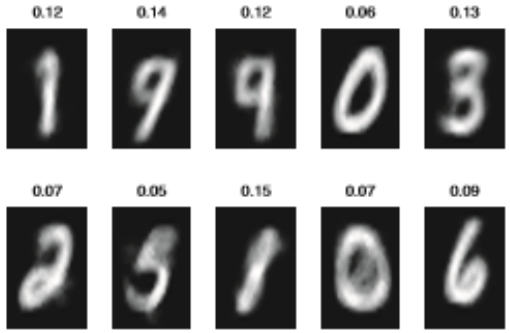


Figure 11.5 We fit a mixture of 10 Bernoullis to the binarized MNIST digit data. We show the MLE for the corresponding cluster means, μ_k. The numbers on top of each image represent the mixing weights $\hat{\pi}_k$. No labels were used when training the model. Figure generated by `mixBerMnistEM`.

suppose our data consist of D -dimensional bit vectors. In this case, an appropriate classconditional density is a product of Bernoullis:

$$p(\mathbf{x}_i \mid z_i = k, \boldsymbol{\theta}) = \prod_{j=1}^{D} \text{Ber}\left(x_{ij} \mid \mu_{jk}\right) = \prod_{j=1}^{D} \mu_{jk}^{x_{ij}} \left(1 - \mu_{jk}\right)^{1-x_{ij}}.$$

where μ_{jk} is the probability that bit j turns on in cluster k.

Note

The latent variables do not have to any meaning, we might simply introduce latent variables in order to make the model more powerful. For example, one can show that the mean and covariance of the mixture distribution are given by

$$\mathbb{E}[\mathbf{x}] = \sum_k \pi_k \boldsymbol{\mu}_k,$$

$$\text{cov}[\mathbf{x}] = \sum_k \pi_k \left[\boldsymbol{\Sigma}_k + \boldsymbol{\mu}_k \boldsymbol{\mu}_k^T\right] - \mathbb{E}[\mathbf{x}]\mathbb{E}[\mathbf{x}]^T,$$

where $\boldsymbol{\Sigma}_k = \text{diag}\left(\mu_{jk}\left(1 - \mu_{jk}\right)\right)$.

Mixtures of experts

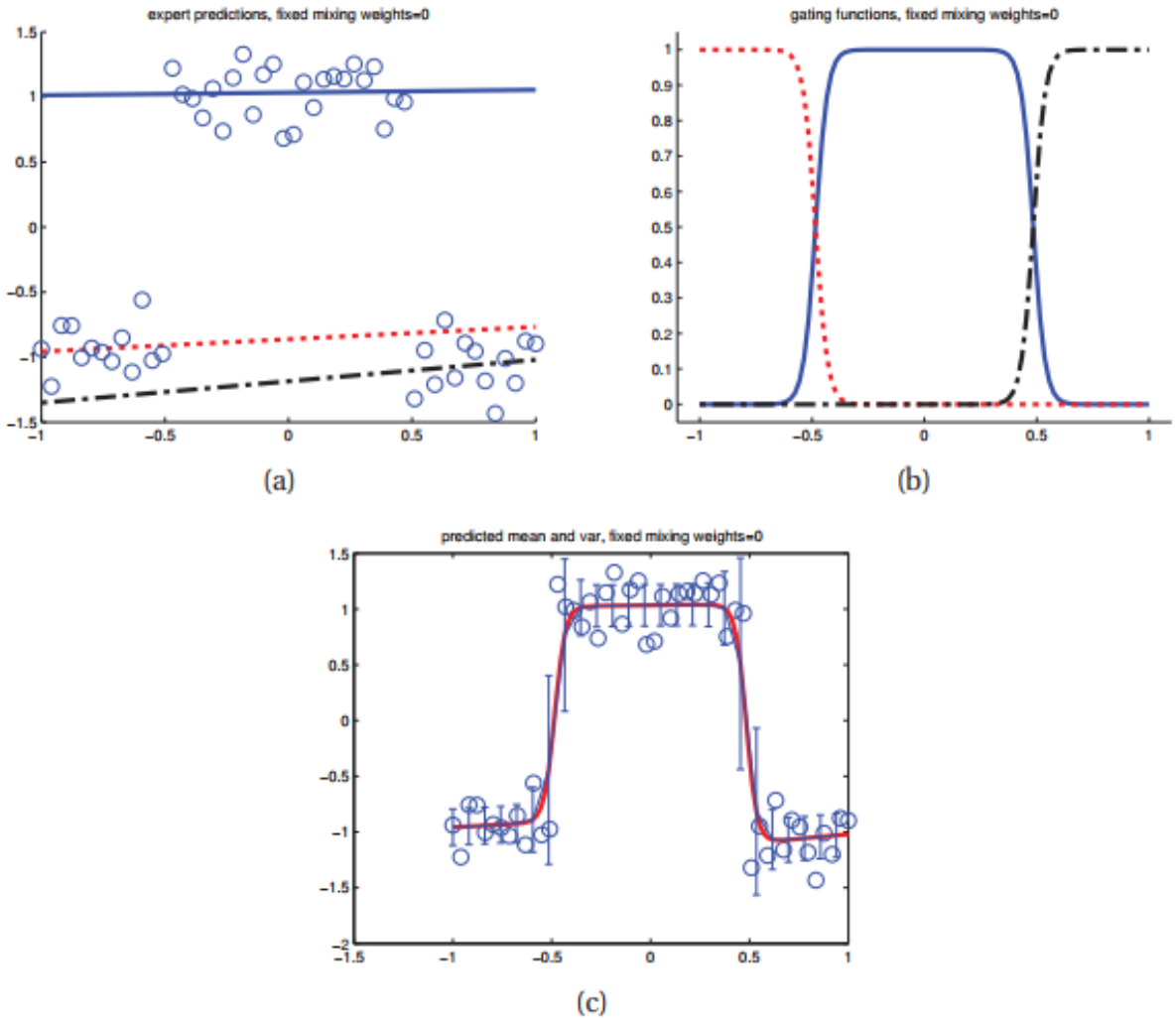


Figure 11.6 (a) Some data fit with three separate regression lines. (b) Gating functions for three different "experts". (c) The conditionally weighted average of the three expert predictions. Figure generated by `mixexpDemo`.

Mixtures of experts

We can model this by allowing the mixing weights and the mixture densities to be input-dependent:

$$p\left(y_i \mid \mathbf{x}_i, z_i = k, \boldsymbol{\theta}\right) = \mathcal{N}\left(y_i \mid \mathbf{w}_k^T \mathbf{x}_i, \sigma_k^2\right),$$
$$p\left(z_i \mid \mathbf{x}_i, \boldsymbol{\theta}\right) = \text{Multi}\left(z_i \mid \mathcal{S}\left(\mathbf{V}^T \mathbf{x}_i\right)\right).$$

The function $p\left(z_i = k \mid \mathbf{x}_i, \boldsymbol{\theta}\right)$ is called a gating function, and decides which expert to use, depending on the input values. The overall prediction of the model, obtained using

$$p\left(y_i \mid \mathbf{x}_i, \boldsymbol{\theta}\right) = \sum_k p\left(z_i = k \mid \mathbf{x}_i, \boldsymbol{\theta}\right) p\left(y_i \mid \mathbf{x}_i, z_i = k, \boldsymbol{\theta}\right).$$

Mixtures of experts

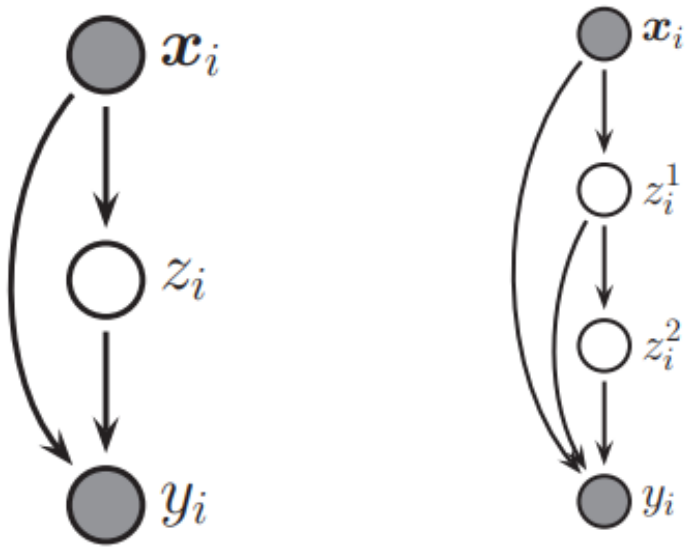


Figure 11.7 (a) A mixture of experts. (b) A hierarchical mixture of experts.

Parameter estimation for mixture models

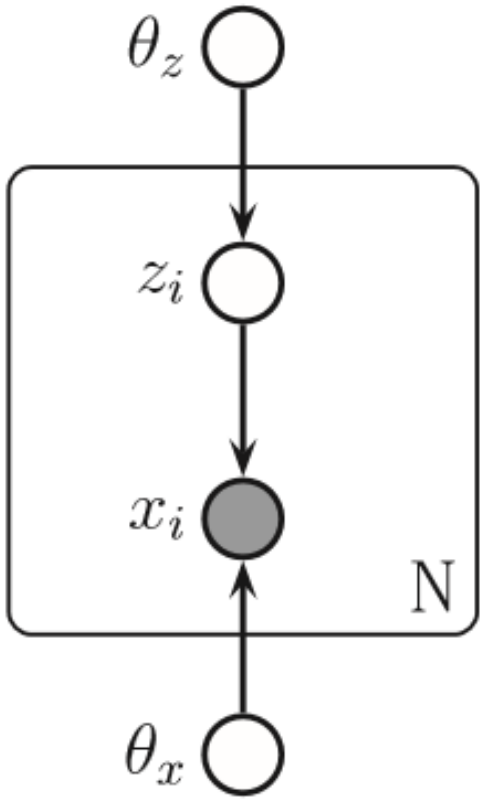
- If the z_i were observed, we see that $\boldsymbol{\theta}_z \perp \boldsymbol{\theta}_x \mid \mathcal{D}$ and hence the posterior will factorize.

$$L(\boldsymbol{\theta} \mid \mathcal{D}) = \prod_{i=1}^{N} p(x_i|\theta_x, z_i).$$

- If the z_i are hidden, the parameters are no longer independent, and the posterior does not factorize, making it much harder to compute.

$$L(\boldsymbol{\theta} \mid \mathcal{D}) = \prod_{i=1}^{N} \{\sum_{k=1}^{K} p(z_i = k|\theta_z) p(x_i|\theta_x, z_i)\}.$$

Parameter estimation for mixture models



(b)

Unidentifiability

- Definition
  Let $\mathcal{P} = \{P_\theta : \theta \in \Theta\}$ be a statistical model. We say that $\mathcal{P}$ is identifiable if the mapping $\theta \mapsto P_\theta$ is one-to-one:

$$P_{\theta_1} = P_{\theta_2} \quad \Rightarrow \quad \theta_1 = \theta_2, \qquad \text{for all } \theta_1, \theta_2 \in \Theta.$$

- Consider a GMM. If the z_i were all observed, we would have a unimodal posterior for the parameters:

$$L(\boldsymbol{\theta} \mid \mathcal{D}) = \prod_{i=1}^{N} p(z_i = k|\theta_z) N(x_i|\mu_k, \boldsymbol{\Sigma}_k).$$

  Consequently we can easily find the globally optimal MLE.
- suppose the z_i's are hidden, for each of the possible ways of "filling in" the z_i's, we get a different unimodal likelihood.(K! possible labelings)

Unidentifiability

Consider a GMM,

$$p(\mathbf{x} \mid \boldsymbol{\theta}) = \pi_1 \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_1, \ _1) + \pi_2 \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_2, \ _2).$$

$$\pi_1 = \pi_2 = 0.5, \mu_1 = -10, \mu_2 = 10, \sigma_1 = \sigma_2 = 5.$$
$$\pi_1 = \pi_2 = 0.5, \mu_1 = 10, \mu_2 = -10, \sigma_1 = \sigma_2 = 5.$$

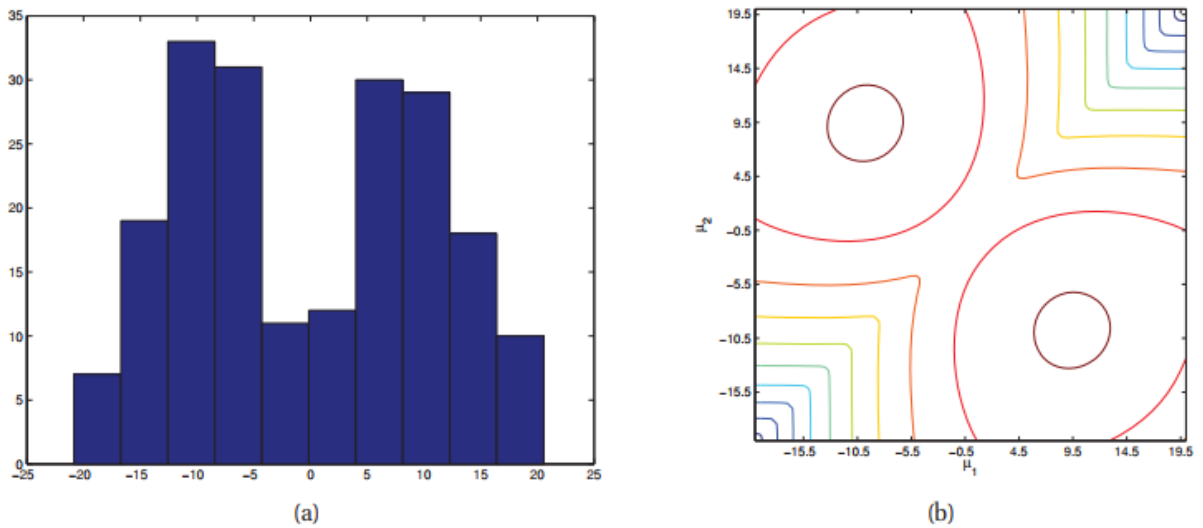


Figure 11.10 Left: $N = 200$ data points sampled from a mixture of 2 Gaussians in 1d, with $\pi_k = 0.5$, $\sigma_k = 5$, $\mu_1 = -10$ and $\mu_2 = 10$. Right: Likelihood surface $p(\mathcal{D}|\mu_1, \mu_2)$, with all other parameters set to their true values. We see the two symmetric modes, reflecting the unidentifiability of the parameters. Figure generated by `mixGaussLikSurfaceDemo`.

Computing a MAP estimate is non-convex

Now suppose the joint probability distribution $p(\mathbf{z}_i, \mathbf{x}_i \mid \boldsymbol{\theta})$ is in the exponential family, which means it can be written as follows:

$$p(\mathbf{x}, \mathbf{z} \mid \boldsymbol{\theta}) = \frac{1}{Z(\boldsymbol{\theta})} \exp\left[\boldsymbol{\theta}^T \boldsymbol{\phi}(\mathbf{x}, \mathbf{z})\right],$$

where $\phi(\mathbf{x}, \mathbf{z})$ are the sufficient statistics, and $Z(\boldsymbol{\theta})$ is the normalization constant. the complete data log likelihood can be written as follows:

$$\ell(\boldsymbol{\theta}|x, z) = \sum_{i=1}^{N} \log p(\mathbf{x}_i, \mathbf{z}_i \mid \boldsymbol{\theta}) = \boldsymbol{\theta}^T \left(\sum_{i=1}^{N} \boldsymbol{\phi}(\mathbf{x}_i, \mathbf{z}_i) \right) - N \log Z(\boldsymbol{\theta}).$$

The first term is clearly linear in $\boldsymbol{\theta}$. One can show that $Z(\boldsymbol{\theta})$ is a convex function, so the overall objective is concave.

Computing a MAP estimate is non-convex

when we assumed z_i is missing. The observed data log likelihood is given by

$$\ell(\boldsymbol{\theta}) = \sum_{i=1}^{N} \log \sum_{\mathbf{z}_i} p(\mathbf{x}_i, \mathbf{z}_i \mid \boldsymbol{\theta}) = \sum_{i=1}^{N} \log \left[\sum_{\mathbf{z}_i} e^{\boldsymbol{\theta}^T \phi(\mathbf{z}_i, \mathbf{x}_i)} \right] - N \log Z(\boldsymbol{\theta}).$$

One can show that the log-sum-exp function is convex , and we know that $Z(\boldsymbol{\theta})$ is convex. However, the difference of two convex functions may be not convex.

The disadvantage of non-convex functions is that it is usually hard to find their global optimum.

Table of Contents

The EM algorithm

One approach is to use a generic gradient-based optimizer to find a local minimum of the negative log likelihood or NLL, given by

$$-\ell(\boldsymbol{\theta} \mid \mathbf{x}), \text{ where } \mathbf{x} \text{ is the observed data.}$$

However, we often have to enforce constraints, such as the fact that covariance matrices must be positive definite(for GMM), mixing weights must sum to one, which can be tricky.
A much simpler algorithm —— Expectation Maximization.

Theoretical basis for EM

Let x be the visible or observed variables, and let z be the hidden or missing variables. Consider an arbitrary distribution $q(\mathbf{z})$ over the hidden variables. The observed data log likelihood can be written as follows:

$$\ell(\boldsymbol{\theta} \mid \mathbf{x}) = \log p(\mathbf{x} \mid \boldsymbol{\theta}) = \log \left[\int p(\mathbf{x}, \mathbf{z} \mid \boldsymbol{\theta}) \, d\mathbf{z} \right]$$
$$= \log \left[\int q(\mathbf{z}) \frac{p(\mathbf{x}, \mathbf{z} \mid \boldsymbol{\theta})}{q(\mathbf{z})} d\mathbf{z} \right]$$

Now log(u) is a concave function, so from Jensen's inequality (Equation 2.113) we have the following lower bound:

$$\ell(\boldsymbol{\theta} \mid \mathbf{x}) = \log \mathbb{E}_z \frac{p(\mathbf{x}, \mathbf{z} \mid \boldsymbol{\theta})}{q(\mathbf{z})} \geq \mathbb{E}_z \log \frac{p(\mathbf{x}, \mathbf{z} \mid \boldsymbol{\theta})}{q(\mathbf{z})}, \textit{where } \mathbf{z} \sim q.$$

The above argument holds for any positive distribution q. Which one should we choose? Intuitively we should pick the q that yields the tightest lower bound.

Theoretical basis for EM

Let $\mathscr{F}(\boldsymbol{\theta}, q) := \mathbb{E}_z \log \frac{p(\mathbf{x},\mathbf{z}|\boldsymbol{\theta})}{q(\mathbf{z})}$, then

$$\begin{aligned}
\mathscr{F}(\boldsymbol{\theta}, q) &= \int q(\mathbf{z}) \log \frac{p(\mathbf{x}, \mathbf{z} \mid \boldsymbol{\theta})}{q(\mathbf{z})} d\mathbf{z} \\
&= \int q(\mathbf{z}) \log \frac{p(\mathbf{z} \mid \mathbf{x}, \boldsymbol{\theta}) p(\mathbf{x} \mid \boldsymbol{\theta})}{q(\mathbf{z})} d\mathbf{z} \\
&= \int q(\mathbf{z}) \log \frac{p(\mathbf{z} \mid \mathbf{x}, \boldsymbol{\theta})}{q(\mathbf{z})} d\mathbf{z} + \int q(\mathbf{z}) \log p(\mathbf{x} \mid \boldsymbol{\theta}) \, d\mathbf{z} \\
&= -\mathbb{KL}(q(\mathbf{z}) \| p(\mathbf{z} \mid \mathbf{x}, \boldsymbol{\theta})) + \log p(\mathbf{x} \mid \boldsymbol{\theta}),
\end{aligned}$$

which means that

$$\ell(\boldsymbol{\theta} \mid \mathbf{x}) = \mathscr{F}(\boldsymbol{\theta}, q) + \mathbb{KL}(q(\mathbf{z}) \| p(\mathbf{z} \mid \mathbf{x}, \boldsymbol{\theta}))$$

Lower bound approximation

Given $\boldsymbol{\theta}^{(k-1)}$, define

$$q^{(k)} = \arg\max \mathscr{F}(q, \boldsymbol{\theta}^{(k-1)}),$$

where $\mathscr{F}(q^{(k)}, \boldsymbol{\theta}^{(k-1)})$ could be viewed as the "optimal" approximation of $\ell(\boldsymbol{\theta}^{(k-1)} \mid \mathbf{x})$

After approximation, we could get $\boldsymbol{\theta}^{(k)} = \arg\max_{\boldsymbol{\theta}} \mathscr{F}\left(\boldsymbol{\theta}, q^{(k)}\right)$.

If the gap between $\mathscr{F}\left(\boldsymbol{\theta}^{(k-1)}, q^{(k)}\right)$ and is $\ell(\boldsymbol{\theta}^{(\boldsymbol{k-1})} \mid \mathbf{x})$ small enough, we could expect that $\ell(\boldsymbol{\theta}^{(k)} \mid \mathbf{x}) \geqslant \ell(\boldsymbol{\theta}^{(k-1)} \mid \mathbf{x})$,since

$$\ell(\boldsymbol{\theta}^{(k)} \mid \mathbf{x}) \geqslant \mathscr{F}\left(\boldsymbol{\theta}^{(k)}, q^{(k)}\right) \geqslant \mathscr{F}\left(\boldsymbol{\theta}^{(k-1)}, q^{(k)}\right) \approx \ell(\boldsymbol{\theta}^{(k-1)} \mid \mathbf{x})$$

Lower bound approximation

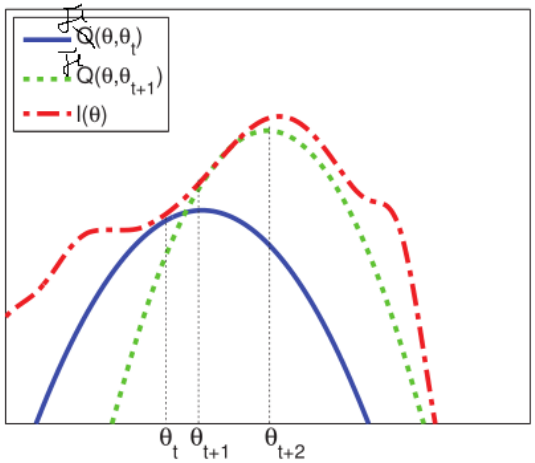


Figure 11.16 Illustration of EM as a bound optimization algorithm. Based on Figure 9.14 of (Bishop 2006a). Figure generated by `emLogLikelihoodMax`.

Lower bound approximation

Now we observed that

$$
\begin{aligned}
q^{(k)} &= \arg\max_{q} \mathscr{F}\left(q, \boldsymbol{\theta}^{(k-1)}\right) \\
&\propto \arg\max_{q} -\mathbb{KL}\left(q(\mathbf{z}) \| p\left(\mathbf{z} \mid \mathbf{x}, \boldsymbol{\theta}^{k-1}\right)\right) \\
&= p\left(\mathbf{z} \mid \mathbf{x}, \boldsymbol{\theta}^{(k-1)}\right),
\end{aligned}
$$

which means that

$$
\begin{aligned}
\boldsymbol{\theta}^{k} &= \arg\max_{\boldsymbol{\theta}} \mathscr{F}\left(q^{(k)}, \boldsymbol{\theta}^{(k-1)}\right) \\
&= \arg\max_{\boldsymbol{\theta}} \mathbb{E}_{z} \log \frac{p(\mathbf{x}, \mathbf{z} \mid \boldsymbol{\theta})}{q^{(k)}(\mathbf{z})} \\
&\propto \arg\max_{\boldsymbol{\theta}} \mathbb{E}_{z} \log p(\mathbf{x}, \mathbf{z} \mid \boldsymbol{\theta}), \ \textit{where}\, \mathbf{z} \sim q\left(\mathbf{z} \mid \mathbf{x}, \boldsymbol{\theta}^{(k-1)}\right) \\
&=: \arg\max_{\boldsymbol{\theta}} Q\left(\boldsymbol{\theta}, \boldsymbol{\theta}^{k-1}\right)
\end{aligned}
$$

This is so called EM

Algorithm:

1.Start with initial guesses for the parameter $\boldsymbol{\theta}^{(0)}$.

2.Expectation Step: at the tth iteration, compute $Q\left(\boldsymbol{\theta}, \boldsymbol{\theta}^{(k-1)}\right)$.

3.Maximization Step: determine the new estimate $\boldsymbol{\theta}^{(k)} = \arg\max_{\boldsymbol{\theta}} Q\left(\boldsymbol{\theta}, \boldsymbol{\theta}^{(k-1)}\right)$.

4.Repeat steps 2 and 3 until covergence.

EM for GMMs

The expected complete data log likelihood is given by

$$
\begin{aligned}
Q\left(\boldsymbol{\theta}, \boldsymbol{\theta}^{(t-1)}\right) &\triangleq \mathbb{E}\left[\sum_i \log p\left(\mathbf{x}_i, z_i \mid \boldsymbol{\theta}\right)\right] \\
&= \sum_i \mathbb{E}\left[\log\left[\prod_{k=1}^{K}\left(\pi_k p\left(\mathbf{x}_i \mid \boldsymbol{\theta}_k\right)\right)^{\mathrm{I}(z_i=k)}\right]\right] \\
&= \sum_i \sum_k \mathbb{E}\left[\mathbb{I}\left(z_i=k\right)\right] \log\left[\pi_k p\left(\mathbf{x}_i \mid \boldsymbol{\theta}_k\right)\right] \\
&= \sum_i \sum_k p\left(z_i=k \mid \mathbf{x}_i, \boldsymbol{\theta}^{(t-1)}\right) \log\left[\pi_k p\left(\mathbf{x}_i \mid \boldsymbol{\theta}_k\right)\right] \\
&= \sum_i \sum_k r_{ik} \log \pi_k + \sum_i \sum_k r_{ik} \log p\left(\mathbf{x}_i \mid \boldsymbol{\theta}_k\right)
\end{aligned}
$$

where $r_{ik} \triangleq p\left(z_i=k \mid \mathbf{x}_i, \boldsymbol{\theta}^{(t-1)}\right)$ is the responsibility that cluster k takes for data point i.

EM for GMMs

- E step: The E step has the following simple form, which is the same for any mixture model:

$$r_{ik} = \frac{\pi_k p\left(\mathbf{x}_i \mid \theta_k^{(t-1)}\right)}{\sum_{k'} \pi_{k'} p\left(\mathbf{x}_i \mid \theta_{k'}^{(t-1)}\right)}.$$

- M step: In the M step, we optimize π and the θ_k of Q function. For π, we obviously have

$$\pi_k = \frac{1}{N} \sum_i r_{ik} = \frac{r_k}{N},$$

where $r_k \triangleq \sum_i r_{ik}$ is the weighted number of points assigned to cluster k.

EM for GMMs

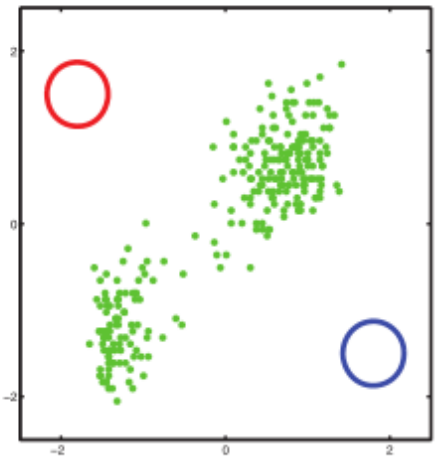
To derive the M step for the $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ terms, we look at the parts of Q that depend on $\boldsymbol{\mu}_k$ and $_k$,

$$\begin{aligned}\ell\left(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right) &= \sum_k \sum_i r_{ik} \log p\left(\mathbf{x}_i \mid \boldsymbol{\theta}_k\right) \\ &= -\frac{1}{2} \sum_i r_{ik} \left[\log \left|\boldsymbol{\Sigma}_k\right| + \left(\mathbf{x}_i - \boldsymbol{\mu}_k\right)^T \boldsymbol{\Sigma}_k^{-1} \left(\mathbf{x}_i - \boldsymbol{\mu}_k\right)\right]\end{aligned}$$

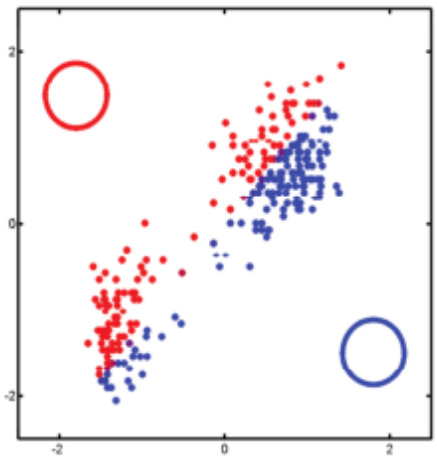
This is just a weighted version of the standard problem of computing the MLEs of an MVN (see Section 4.1.3).the new parameter estimates are given by

$$\begin{aligned}\boldsymbol{\mu}_k &= \frac{\sum_i r_{ik} \mathbf{x}_i}{r_k} \\ \boldsymbol{\Sigma}_k &= \frac{\sum_i r_{ik} (\mathbf{x}_i - \boldsymbol{\mu}_k)(\mathbf{x}_i - \boldsymbol{\mu}_k)^T}{r_k} = \frac{\sum_i r_{ik} \mathbf{x}_i \mathbf{x}_i^T}{r_k} - \boldsymbol{\mu}_k \boldsymbol{\mu}_k^T.\end{aligned}$$

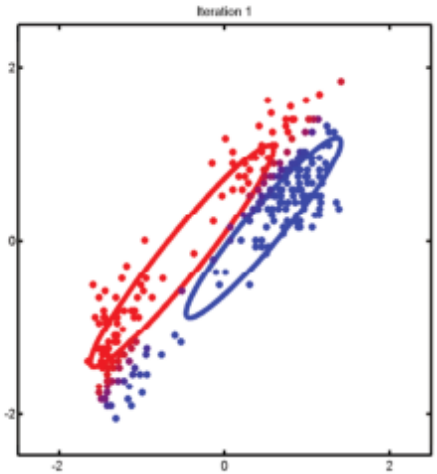
Example:EM for GMMs



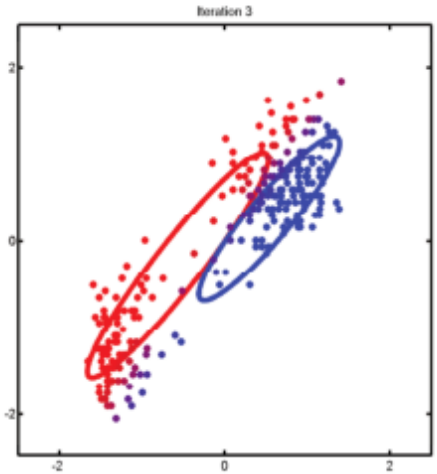
(a)



(b)

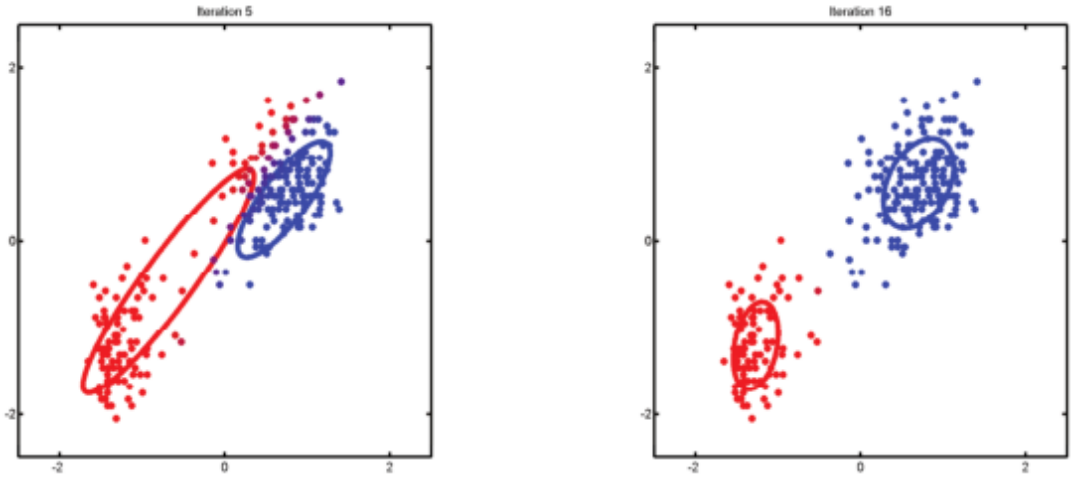


(c)



(d)

Example:EM for GMMs



(e) (f)

Figure 11.11 Illustration of the EM for a GMM applied to the Old Faithful data. (a) Initial (random) values of the parameters. (b) Posterior responsibility of each point computed in the first E step. The degree of redness indicates the degree to which the point belongs to the red cluster, and similarly for blue; this purple points have a roughly uniform posterior over clusters. (c) We show the updated parameters after the first M step. (d) After 3 iterations. (e) After 5 iterations. (f) After 16 iterations. Based on (Bishop 2006a) Figure 9.8. Figure generated by `mixGaussDemoFaithful`.

K-means algorithm

A popular variant of the EM algorithm for GMMs known as the K-means algorithm.
*quad*Consider a GMM in which we assume $\Sigma_k = \sigma^2 \mathbf{I}_D$ and $\pi_k = 1/K$ are fixed, only $\boldsymbol{\mu}_k \in \mathbb{R}^D$, have to be estimated.
During the E step:

$$p(z_i = k \mid \mathbf{x}_i, \boldsymbol{\theta}) \approx \mathbb{I}(k = z_i^*)$$

where $z_i* = \text{argmax}_k \, p(z_i = k \mid \mathbf{x}_i, \boldsymbol{\theta})$.(hard **EM**).
Since we assumed an equal spherical covariance matrix for each cluster,

$$z_i^* = \arg\min_k \|\mathbf{x}_i - \boldsymbol{\mu}_k\|_2^2 .$$

The M step:

$$\mu_k = \frac{1}{N_k} \sum_{i:z_i=k} \mathbf{x}_i.$$

K-means algorithm

Algorithm 11.1: K-means algorithm

initialize $\mathbf{m}_k$;
repeat
  Assign each data point to its closest cluster center: $z_i = \arg\min_k ||\mathbf{x}_i - \boldsymbol{\mu}_k||_2^2$;
  Update each cluster center by computing the mean of all points assigned to it: $\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{i:z_i=k} \mathbf{x}_i$;
until *converged*;

Example:K-means algorithm

1.Starts with guesses for the three cluster centers.
2.For each data point, the closest cluster center (in Euclidean distance) is identified;
3.Each cluster center is replaced by the coordinate-wise average of all data points that are closest to it.
4.Repeat 2 and 3.

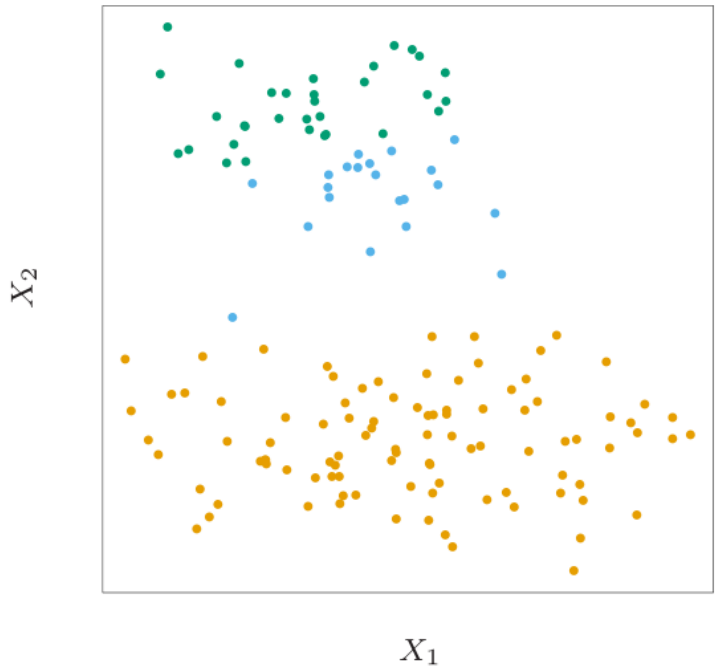


FIGURE 14.4. *Simulated data in the plane, clustered into three classes (represented by orange, blue and green) by the K-means clustering algorithm*

Example:K-means algorithm

Some of the K-means iterations for the simulated data of Figure 14.4

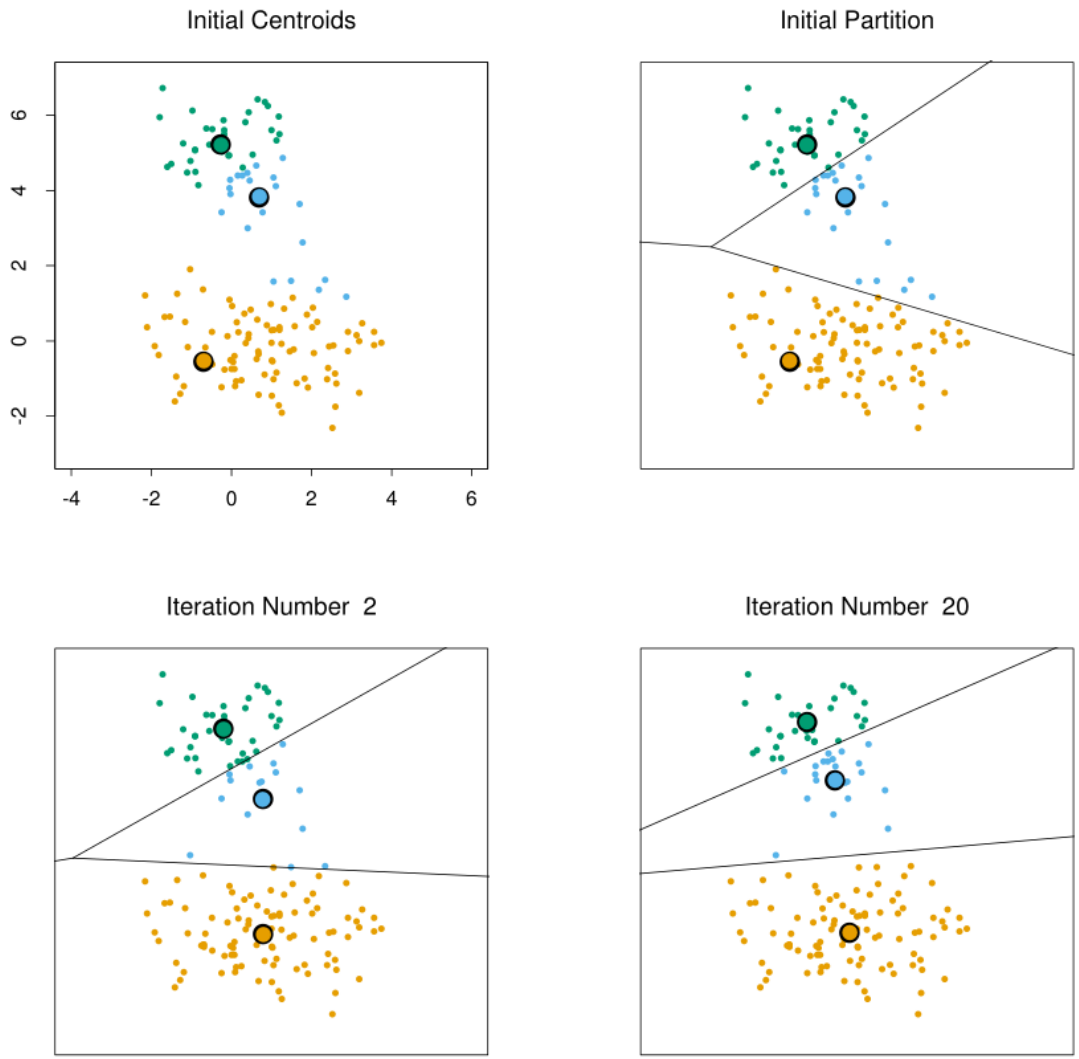


Table of Contents

EM for mixture of experts

We can fit a mixture of experts model using EM in a straightforward manner. The expected complete data log likelihood is given by

$$Q\left(\boldsymbol{\theta}, \boldsymbol{\theta}^{old}\right) = \sum_{i=1}^{N} \sum_{k=1}^{K} r_{ik} \log \left[\pi_{ik} \mathcal{N}\left(y_i \mid \mathbf{w}_k^T \mathbf{x}_i, \sigma_k^2\right)\right]$$

$$\pi_{i,k} \triangleq \mathcal{S}\left(\mathbf{V}^T \mathbf{x}_i\right)_k$$

$$r_{ik} \quad \propto \quad \pi_{ik}^{old} \mathcal{N}\left(y_i \mid \mathbf{x}_i^T \mathbf{w}_k^{old}, \left(\sigma_k^{old}\right)^2\right)$$

So the E step is the same as in a standard mixture model, except we have to replace π_k with $\pi_{i,k}$ when computing r_{ik}.

EM for mixture of experts

In the M step, we need to maximize $Q\left(\boldsymbol{\theta}, \boldsymbol{\theta}^{\text{old}}\right)$ wrt $\mathbf{w}_k, \sigma_k^2$ and $\mathbf{V}$. For the regression parameters for model k, the objective has the form

$$Q\left(\boldsymbol{\theta}_k, \boldsymbol{\theta}^{\text{old}}\right) = \sum_{i=1}^{N} r_{ik} \left\{ -\frac{1}{\sigma_k^2} \left(y_i - \mathbf{w}_k^T \mathbf{x}_i\right)^2 \right\}$$

we can immediately write down the MLE as $\mathbf{w}_k = \left(\mathbf{X}^T \mathbf{R}_k \mathbf{X}\right)^{-1} \mathbf{X}^T \mathbf{R}_k \mathbf{y}$, where $\mathbf{R}_k = \text{diag}\left(r_{:,k}\right)$. The MLE for the variance is given by

$$\sigma_k^2 = \frac{\sum_{i=1}^{N} r_{ik} \left(y_i - \mathbf{w}_k^T \mathbf{x}_i\right)^2}{\sum_{i=1}^{N} r_{ik}}$$

We replace the estimate of the unconditional mixing weights π with the estimate of the gating parameters, $\mathbf{V}$. The objective has the form

$$\ell(\mathbf{V}) = \sum_i \sum_k r_{ik} \log \pi_{i,k}$$

EM for DGMs with hidden variables

For simplicity of presentation, we will assume all CPDs are tabular. Based on Section 10.4.2, let us write each CPT as follows:

$$p\left(x_{it} \mid \mathbf{x}_{i,\mathrm{pa}(t)}, \boldsymbol{\theta}_t\right) = \prod_{c=1}^{K_{\mathrm{pa}(t)}} \prod_{k=1}^{K_t} \theta_{tck}^{\mathbb{I}\left(x_{it}=i, \mathbf{x}_{i,\mathrm{pa}(t)}=c\right)}$$

The log-likelihood of the complete data is given by

$$\log p(\mathcal{D} \mid \boldsymbol{\theta}) = \sum_{t=1}^{V} \sum_{c=1}^{K_{\mathrm{pa}(t)}} \sum_{k=1}^{K_t} N_{tck} \log \theta_{tck}$$

where $N_{tck} = \sum_{i=1}^{N} \mathbb{I}\left(x_{it} = i, \mathbf{x}_{i,\mathrm{pa}(t)} = c\right)$ are the empirical counts. Hence the expected complete data log-likelihood has the form,

$$\mathbb{E}[\log p(\mathcal{D} \mid \boldsymbol{\theta})] = \sum_t \sum_c \sum_k \bar{N}_{tck} \log \theta_{tck}$$

EM for DGMs with hidden variables

where

$$\bar{N}_{tck} = \sum_{i=1}^{N} \mathbb{E}\left[\mathbb{I}\left(x_{it} = i, \mathbf{x}_{i,\mathrm{pa}(t)} = c\right)\right] = \sum_{i} p\left(x_{it} = k, \mathbf{x}_{i,\mathrm{pa}(t)} = c \mid \mathcal{D}_i\right)$$

where $\mathcal{D}_i$ are all the visible variables in case i. The $\bar{N}_{tck}$ are the expected sufficient statistics, and constitute the output of the E step. Given these ESS, the M step has the simple form

$$\hat{\theta}_{tck} = \frac{\bar{N}_{tck}}{\sum_{k'} \bar{N}_{tjk'}}$$

This can be proved by adding Lagrange multipliers (to enforce the constraint $\sum_k \theta_{tjk} = 1$) to the expected complete data log likelihood, and then optimizing each parameter vector $\boldsymbol{\theta}_{tc}$ separately.

Batch EM review

Before explaining online EM, we review batch EM in a more abstract setting. Let $\phi(\mathbf{x}, \mathbf{z})$ be a vector of sufficient statistics for a single data case. Let $\mathbf{s}_i = \sum_{\mathbf{z}} p(\mathbf{z} \mid \mathbf{x}_i, \boldsymbol{\theta}) \boldsymbol{\phi}(\mathbf{x}_i, \mathbf{z})$ be the expected sufficient statistics for case i, and $\boldsymbol{\mu} = \sum_{i=1}^{N} \mathbf{s}_i$ be the sum of the ESS. Given $\boldsymbol{\mu}$, we can derive an ML or MAP estimate of the parameters in the M step; we will denote this operation by $\boldsymbol{\theta}(\boldsymbol{\mu})$.

Algorithm 11.2: Batch EM algorithm

initialize $\boldsymbol{\mu}$;
repeat
  $\boldsymbol{\mu}^{new} = \mathbf{0}$;
  for *each example* $i = 1 : N$ **do**
    $\mathbf{s}_i := \sum_{\mathbf{z}} p(\mathbf{z}|\mathbf{x}_i, \boldsymbol{\theta}(\boldsymbol{\mu}))\boldsymbol{\phi}(\mathbf{x}_i, \mathbf{z})$;
    $\boldsymbol{\mu}^{new} := \boldsymbol{\mu}^{new} + \mathbf{s}_i$; ;
  $\boldsymbol{\mu} := \boldsymbol{\mu}^{new}$;
until *converged*;

Incremental EM

In incremental EM (Neal and Hinton 1998), we keep track of μ as well as the s_i . When we come to a data case, we swap out the old s_i and replace it with the new s_i^{new}.

Algorithm 11.3: Incremental EM algorithm

1 *initialize* $\mathbf{s}_i$ for $i = 1 : N$;
2 $\boldsymbol{\mu} = \sum_i \mathbf{s}_i$;
3 **repeat**
4 **for** *each example* $i = 1 : N$ *in a random order* **do**
5 $\mathbf{s}_i^{new} := \sum_{\mathbf{z}} p(\mathbf{z}|\mathbf{x}_i, \boldsymbol{\theta}(\boldsymbol{\mu}))\boldsymbol{\phi}(\mathbf{x}_i, \mathbf{z})$;
6 $\boldsymbol{\mu} := \boldsymbol{\mu} + \mathbf{s}_i^{new} - \mathbf{s}_i$;
7 $\mathbf{s}_i := \mathbf{s}_i^{new}$;
8 **until** *converged*;

This can be viewed as maximizing the lower bound $Q(\boldsymbol{\theta}, q_1, \ldots, q_N)$ by optimizing q_1, then $\boldsymbol{\theta}$, then q_2, then $\boldsymbol{\theta}$, etc. As such, this method is guaranteed to monotonically converge to a local maximum of the lower bound and to the log likelihood itself.

Stepwise EM

In stepwise EM, whenever we compute a new s_i , we move μ towards it, as shown in Algorithm 7. At iteration k, the stepsize has value η_k , which must satisfy the Robbins-Monro conditions in Equation 8.82. We can get somewhat better behavior by using a minibatch of size m before each update. It is possible to optimize m and k to maximize the training set likelihood, bytrying different values in parallel for an initial trial period; this can significantly speed up the algorithm.

Algorithm 11.4: Stepwise EM algorithm

initialize $\boldsymbol{\mu}$; $k = 0$;
repeat
  for *each example* $i = 1 : N$ *in a random order* **do**
    $\mathbf{s}_i := \sum_{\mathbf{z}} p(\mathbf{z}|\mathbf{x}_i, \boldsymbol{\theta}(\boldsymbol{\mu}))\boldsymbol{\phi}(\mathbf{x}_i, \mathbf{z})$;
    $\boldsymbol{\mu} := (1 - \eta_k)\boldsymbol{\mu} + \eta_k \mathbf{s}_i$;
    $k := k + 1$
until *converged*;

Fitting models with missing data

Suppose we want to fit a joint density model by maximum likelihood, but we have "holes" in our data matrix, due to missing data (usually represented by NaNs). More formally,let $O_{ij} = 1$ if component j of data case i is observed, and let $O_{ij} = 0$ otherwise.Let $\mathbf{X}_v = \{x_{ij} : O_{ij} = 1\}$ be the visible data, and $O_{ij} = 0$ otherwise.Let $\mathbf{X}_h = \{x_{ij} : O_{ij} = 0\}$ be the missing or hidden data.
Our goal is to compute $\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\text{argmax}} p(\mathbf{X}_v \mid \boldsymbol{\theta}, \mathbf{O})$. Under the missing at random assumption (see Section 8.6.2), we have

$$p(\mathbf{X}_v \mid \boldsymbol{\theta}, \mathbf{O}) = \prod_{i=1}^{N} p(\mathbf{x}_{iv} \mid \boldsymbol{\theta})$$

where $\mathbf{x}_{iv}$ is a vector created from row i and the columns indexed by the set $\{j : O_{ij} = 1\}$.

Fitting models with missing data

Hence the log-likelihood has the form

$$\log p(\mathbf{X}_v \mid \boldsymbol{\theta}) = \sum_i \log p(\mathbf{x}_{iv} \mid \boldsymbol{\theta})$$

where

$$p(\mathbf{x}_{iv} \mid \boldsymbol{\theta}) = \sum_{\mathbf{x}_{ih}} p(\mathbf{x}_{iv}, \mathbf{x}_{ih} \mid \boldsymbol{\theta})$$

where x_{iv} is the vector of hidden variables for case i (assumed discrete for notational simplicity). Substituting in, we get

$$\log p(\mathbf{X}_v \mid \boldsymbol{\theta}) = \sum_i \log \left[\sum_{\mathbf{x}_{ih}} p(\mathbf{x}_{iv}, \mathbf{x}_{ih} \mid \boldsymbol{\theta}) \right]$$

Unfortunately, this objective is hard to maximize. since we cannot push the log inside the sum.However, we can use the EM algorithm to compute a local optimum.

EM for the MLE of an MVN with missing data

Suppose we want to fit an MVN by maximum likelihood, but we have missing data. We can useEM to find a local maximum of the objective, as we explain below.
To get the algorithm started, we can compute the MLE based on those rows of the data matrix that are fully observed. If there are no such rows, we can use some ad-hoc imputation procedures, and then compute an initial MLE.

E step

Once we have ^{t1}, we can compute the expected complete data log likelihood at iteration t as follows:

$$
\begin{aligned}
Q\left(\boldsymbol{\theta}, \boldsymbol{\theta}^{t-1}\right) &= \mathbb{E}\left[\sum_{i=1}^{N} \log \mathcal{N}\left(\mathbf{x}_i \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}\right) \mid \mathcal{D}, \boldsymbol{\theta}^{t-1}\right] \\
&= -\frac{N}{2} \log \left|2\pi\boldsymbol{\Sigma}\right| - \frac{1}{2} \sum_{i} \mathbb{E}\left[\left(\mathbf{x}_i - \boldsymbol{\mu}\right)^T \boldsymbol{\Sigma}^{-1} \left(\mathbf{x}_i - \boldsymbol{\mu}\right)\right] \\
&= -\frac{N}{2} \log \left|2\pi\boldsymbol{\Sigma}\right| - \frac{1}{2} \operatorname{tr}\left(\boldsymbol{\Sigma}^{-1} \sum_{i} \mathbb{E}\left[\left(\mathbf{x}_i - \boldsymbol{\mu}\right)\left(\mathbf{x}_i - \boldsymbol{\mu}\right)^T\right]\right). \\
&= -\frac{N}{2} \log \left|\boldsymbol{\Sigma}\right| - \frac{N}{2} \log(2\pi) - \frac{1}{2} \operatorname{tr}\left(\boldsymbol{\Sigma}^{-1} \mathbb{E}[\mathbf{S}(\boldsymbol{\mu})]\right)
\end{aligned}
$$

where

$$
\mathbb{E}[\mathbf{S}(\boldsymbol{\mu})] \triangleq \sum_{i} \left(\mathbb{E}\left[\mathbf{x}_i \mathbf{x}_i^T\right] + \boldsymbol{\mu}\boldsymbol{\mu}^T - 2\boldsymbol{\mu}\mathbb{E}\left[\mathbf{x}_i\right]^T\right)
$$

E step

We see that we need to compute $\sum_i \mathbb{E}[\mathbf{x}_i]$ and $\sum_i \mathbb{E}\left[\mathbf{x}_i \mathbf{x}_i^T\right]$ these are the expected sufficient statistics. Specifically, consider case i, where components v are observed and components h are unobserved. We have

$$\begin{aligned} \mathbf{x}_{ih} \mid \mathbf{x}_{iv}, \boldsymbol{\theta} &\sim \mathcal{N}(\mathbf{m}_i, \mathbf{V}_i) \\ \mathbf{m}_i &\triangleq \boldsymbol{\mu}_h + \boldsymbol{\Sigma}_{hv} \boldsymbol{\Sigma}_{vv}^{-1} (\mathbf{x}_{iv} - \boldsymbol{\mu}_v) \\ \mathbf{V}_i &\triangleq \boldsymbol{\Sigma}_{hh} - \boldsymbol{\Sigma}_{hv} \boldsymbol{\Sigma}_{vv}^{-1} \boldsymbol{\Sigma}_{vh} \end{aligned}$$

Hence the expected sufficient statistics are

$$\mathbb{E}[\mathbf{x}_i] = (\mathbb{E}[\mathbf{x}_{ih}]; \mathbf{x}_{iv}) = (\mathbf{m}_i; \mathbf{x}_{iv})$$

where we have assumed (without loss of generality) that the unobserved variables come before the observed variables in the node ordering.

E step

To compute $\sum_i \mathbb{E}\left[\mathbf{x}_i \mathbf{x}_i^T\right]$, we use the result that $\text{cov}[\mathbf{x}] = \mathbb{E}\left[\mathbf{x}\mathbf{x}^T\right] - \mathbb{E}[\mathbf{x}]\mathbb{E}\left[\mathbf{x}^T\right]$.Hence

$$\mathbb{E}\left[\mathbf{x}_i \mathbf{x}_i^T\right] = \mathbb{E}\left[\begin{pmatrix} \mathbf{x}_{ih} \\ \mathbf{x}_{iv} \end{pmatrix} \begin{pmatrix} \mathbf{x}_{ih}^T & \mathbf{x}_{iv}^T \end{pmatrix}\right] = \begin{pmatrix} \mathbb{E}\left[\mathbf{x}_{ih}\mathbf{x}_{ih}^T\right] & \mathbb{E}\left[\mathbf{x}_{ih}\right]\mathbf{x}_{iv}^T \\ \mathbf{x}_{iv}\mathbb{E}\left[\mathbf{x}_{ih}\right]^T & \mathbf{x}_{iv}\mathbf{x}_{iv}^T \end{pmatrix}$$

$$\mathbb{E}\left[\mathbf{x}_{ih}\mathbf{x}_{ih}^T\right] = \mathbb{E}\left[\mathbf{x}_{ih}\right]\mathbb{E}\left[\mathbf{x}_{ih}\right]^T + \mathbf{V}_i$$

M step

By solving $\nabla Q\left(\boldsymbol{\theta}, \boldsymbol{\theta}^{(t-1)}\right)=\mathbf{0}$, we can show that the M step is equivalent to plugging these ESS into the usual MLE equations to get

$$\boldsymbol{\mu}^{t}=\frac{1}{N} \sum_{i} \mathbb{E}\left[\mathbf{x}_{i}\right]$$

$$\boldsymbol{\Sigma}^{t}=\frac{1}{N} \sum \mathbb{E}\left[\mathbf{x}_{i} \mathbf{x}_{i}^{T}\right]-\boldsymbol{\mu}^{t}\left(\boldsymbol{\mu}^{t}\right)^{T}$$

Thus we see that EM is not equivalent to simply replacing variables by their expectations and applying the standard MLE formula; that would ignore the posterior variance and would result in an incorrect estimate. Instead we must compute the expectation of the sufficient statistics, and plug that into the usual equation for the MLE.

MAP estimation

As usual, the MLE may overfit.

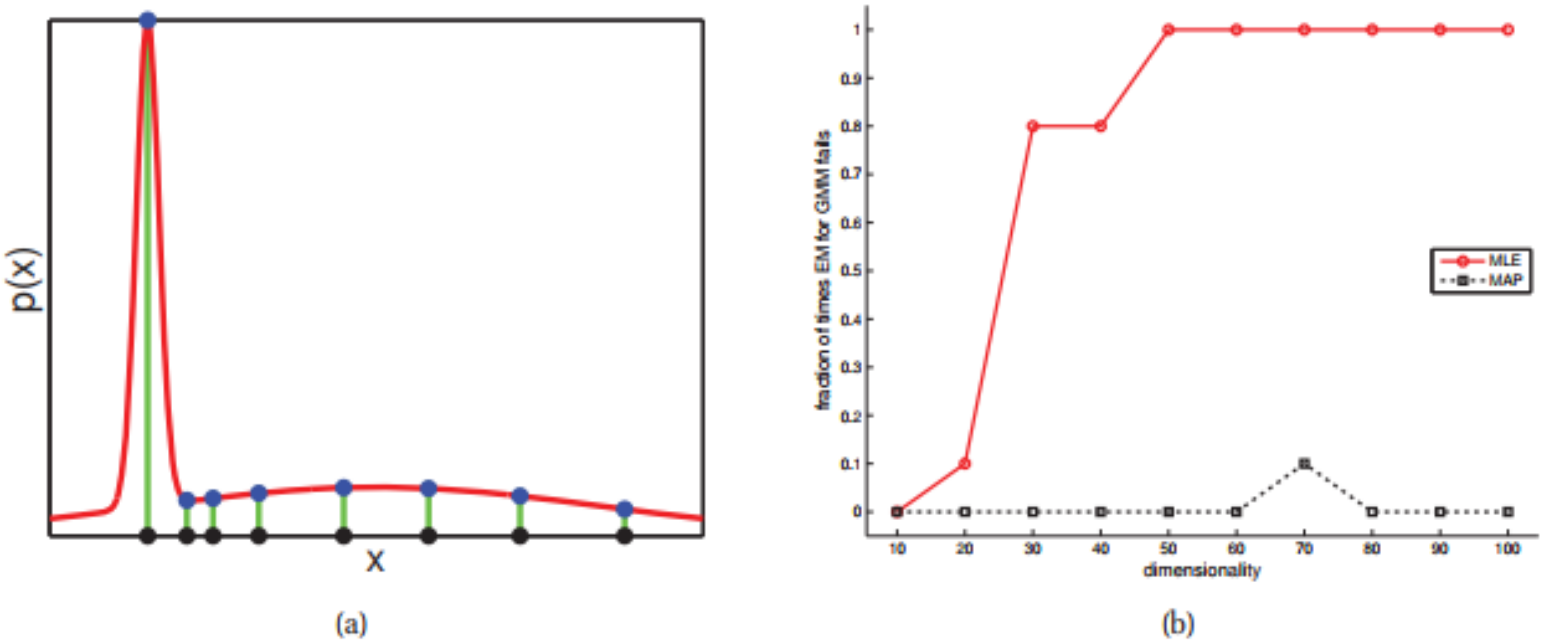


Figure 11.13 (a) Illustration of how singularities can arise in the likelihood function of GMMs. Based on (Bishop 2006a) Figure 9.7. Figure generated by `mixGaussSingularity`. (b) Illustration of the benefit of MAP estimation vs ML estimation when fitting a Gaussian mixture model. We plot the fraction of times (out of 5 random trials) each method encounters numerical problems vs the dimensionality of the problem, for $N = 100$ samples. Solid red (upper curve): MLE. Dotted black (lower curve): MAP. Figure generated by `mixGaussMLvsMAP`.

MAP estimation

An easy solution to this is to perform MAP estimation. The new auxiliary function:

$$Q'\left(\boldsymbol{\theta}, \boldsymbol{\theta}^{\text{old}}\right) = \left[\sum_i \sum_k r_{ik} \log \pi_{ik} + \sum_i \sum_k r_{ik} \log p\left(\mathbf{x}_i \mid \boldsymbol{\theta}_k\right)\right] + \log p(\boldsymbol{\pi}) + \sum_k \log p\left(\boldsymbol{\theta}_k\right)$$

The E step remains unchanged,
it is natural to use a Dirichlet prior, $\pi \sim \mathrm{Dir}(\alpha)$, since this is conjugate to the categorical distribution. The MAP estimate is given by

$$\pi_k = \frac{r_k + \alpha_k - 1}{N + \sum_k \alpha_k - K}.$$

For simplicity, we consider a conjugate prior of the form

$$p\left(\boldsymbol{\mu}_k, \ {}_k\right) = \mathrm{NIW}\left(\boldsymbol{\mu}_k, \ {}_k \mid \mathbf{m}_0, \kappa_0, \nu_0, \mathbf{S}_0\right).$$

MAP estimation

$$\hat{\boldsymbol{\mu}}_k = \frac{r_k \bar{\mathbf{x}}_k + \kappa_0 \mathbf{m}_0}{r_k + \kappa_0}$$
$$\bar{\mathbf{x}}_k \triangleq \frac{\sum_i r_{ik} \mathbf{x}_i}{r_k}$$
$$\hat{}_k = \frac{\mathbf{S}_0 + \mathbf{S}_k + \frac{\kappa_0 r_k}{\kappa_6 + r_k} (\bar{\mathbf{x}}_k - \mathbf{m}_0)(\bar{\mathbf{x}}_k - \mathbf{m}_0)^T}{\nu_0 + r_k + D + 2}$$
$$\mathbf{S}_k \triangleq \sum_i r_{ik} \left(\mathbf{x}_i - \bar{\mathbf{x}}_k\right) \left(\mathbf{x}_i - \bar{\mathbf{x}}_k\right)^T.$$

how to set $\mathbf{S}_0$,

$$\mathbf{S}_0 = \frac{1}{K^{1/D}} \operatorname{diag}\left(s_1^2, \ldots, s_D^2\right)$$

where $s_j = (1/N) \sum_{i=1}^{N} \left(x_{ij} - \bar{x}_j\right)^2$, The parameter ν_0 controls how strongly we believe this prior. The weakest prior we can use, while still being proper, is to set $\nu_0 = D + 2$, so this is a common choice.