Mixture models. Expectation-Maximization. Clustering

Virgil Pavlu November 3, 2008

1 Mixtures

10.2 Mixture Densities and Identifiability

We begin by assuming that we know the complete probability structure for the problem with the sole exception of the values of some parameters. To be more specific, we make the following assumptions:

1. The samples come from a known number c of classes.
2. The prior probabilities $P(\omega_j)$ for each class are known, $j = 1, \ldots, c$.
3. The forms for the class-conditional probability densities $p(\mathbf{x}|\omega_j, \boldsymbol{\theta}_j)$ are known, $j = 1, \ldots, c$.
4. The values for the c parameter vectors $\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_c$ are unknown.
5. The category labels are unknown.

Samples are assumed to be obtained by selecting a state of nature ω_j with probability $P(\omega_j)$ and then selecting an $\mathbf{x}$ according to the probability law $p(\mathbf{x}|\omega_j, \boldsymbol{\theta}_j)$. Thus, the probability density function for the samples is given by

$$p(\mathbf{x}|\boldsymbol{\theta}) = \sum_{j=1}^{c} p(\mathbf{x}|\omega_j, \boldsymbol{\theta}_j) P(\omega_j), \tag{1}$$

where $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_c)$. For obvious reasons, a density function of this form is called a *mixture density*. The conditional densities $p(\mathbf{x}|\omega_j, \boldsymbol{\theta}_j)$ are called the *component densities*, and the prior probabilities $P(\omega_j)$ are called the *mixing parameters*. The mixing parameters can also be included among the unknown parameters, but for the moment we shall assume that only $\boldsymbol{\theta}$ is unknown.

Our basic goal will be to use samples drawn from this mixture density to estimate the unknown parameter vector $\boldsymbol{\theta}$. Once we know $\boldsymbol{\theta}$ we can decompose the mixture into its components and use a Bayesian classifier on the derived densities, if indeed classification is our final goal. Before seeking explicit solutions to this problem, however, let us ask whether or not it is possible in principle to recover $\boldsymbol{\theta}$ from the mixture. Suppose that we had an unlimited number of samples, and that we used one of the nonparametric methods of Chap. ?? to determine the value of $p(\mathbf{x}|\boldsymbol{\theta})$ for every $\mathbf{x}$. If

functions are identifiable, as are most complex or high-dimensional density functions encountered in real-world problems.

Mixtures of discrete distributions are not always so obliging. As a simple example consider the case where x is binary and $P(x|\boldsymbol{\theta})$ is the mixture

$$\begin{aligned} P(x|\boldsymbol{\theta}) &= \frac{1}{2}\theta_1^x(1-\theta_1)^{1-x} + \frac{1}{2}\theta_2^x(1-\theta_2)^{1-x} \\ &= \begin{cases} \frac{1}{2}(\theta_1+\theta_2) & \text{if } x=1 \\ 1-\frac{1}{2}(\theta_1+\theta_2) & \text{if } x=0. \end{cases} \end{aligned}$$

Suppose, for example, that we know for our data that $P(x=1|\boldsymbol{\theta}) = 0.6$, and hence that $P(x=0|\boldsymbol{\theta}) = 0.4$. Then we know the function $P(x|\boldsymbol{\theta})$, but we cannot determine $\boldsymbol{\theta}$, and hence cannot extract the component distributions. The most we can say is that $\theta_1+\theta_2 = 1.2$. Thus, here we have a case in which the mixture distribution is completely unidentifiable, and hence a case for which unsupervised learning is impossible in principle. Related situations may permit us to determine one or *some* parameters, but not all (Problem 3).

This kind of problem commonly occurs with discrete distributions. If there are too many components in the mixture, there may be more unknowns than independent equations, and identifiability can be a serious problem. For the continuous case, the problems are less severe, although certain minor difficulties can arise due to the possibility of special cases. Thus, while it can be shown that mixtures of normal densities are usually identifiable, the parameters in the simple mixture density

$$p(x|\boldsymbol{\theta}) = \frac{P(\omega_1)}{\sqrt{2\pi}}\exp\left[-\frac{1}{2}(x-\theta_1)^2\right] + \frac{P(\omega_2)}{\sqrt{2\pi}}\exp\left[-\frac{1}{2}(x-\theta_2)^2\right] \tag{2}$$

cannot be uniquely identified if $P(\omega_1) = P(\omega_2)$, for then θ_1 and θ_2 can be interchanged without affecting $p(x|\boldsymbol{\theta})$. To avoid such irritations, we shall acknowledge that identifiability can be a problem, but shall henceforth assume that the mixture densities we are working with are identifiable.

---

* Technically speaking, a distribution is not identifiable if we cannot determine the parameters *without bias*. We might guess their correct values, but such a guess would have to be biased in some way.

10.3 Maximum-Likelihood Estimates

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

$$p(\mathbf{x}|\boldsymbol{\theta}) = \sum_{j=1}^{c} p(\mathbf{x}|\omega_j, \boldsymbol{\theta}_j) P(\omega_j), \tag{1}$$

where the full parameter vector $\boldsymbol{\theta}$ is fixed but unknown. The likelihood of the observed samples is, by definition, the joint density

$$p(\mathcal{D}|\boldsymbol{\theta}) \equiv \prod_{k=1}^{n} p(\mathbf{x}_k|\boldsymbol{\theta}). \tag{3}$$

The maximum-likelihood estimate $\hat{\boldsymbol{\theta}}$ is that value of $\boldsymbol{\theta}$ that maximizes $p(\mathcal{D}|\boldsymbol{\theta})$.

If we assume that $p(\mathcal{D}|\boldsymbol{\theta})$ is a differentiable function of $\boldsymbol{\theta}$, then we can derive some interesting necessary conditions for $\hat{\boldsymbol{\theta}}$. Let l be the logarithm of the likelihood, and let $\nabla_{\boldsymbol{\theta}_i} l$ be the gradient of l with respect to $\boldsymbol{\theta}_i$. Then

$$l = \sum_{k=1}^{n} \ln\, p(\mathbf{x}_k|\boldsymbol{\theta}) \tag{4}$$

and

$$\nabla_{\boldsymbol{\theta}_i} l = \sum_{k=1}^{n} \frac{1}{p(\mathbf{x}_k|\boldsymbol{\theta})} \nabla_{\boldsymbol{\theta}_i} \left[\sum_{j=1}^{c} p(\mathbf{x}_k|\omega_j, \boldsymbol{\theta}_j) P(\omega_j) \right]. \tag{5}$$

If we assume that the elements of $\boldsymbol{\theta}_i$ and $\boldsymbol{\theta}_j$ are functionally independent if $i \neq j$, and if we introduce the posterior probability

$$P(\omega_i|\mathbf{x}_k, \boldsymbol{\theta}) = \frac{p(\mathbf{x}_k|\omega_i, \boldsymbol{\theta}_i) P(\omega_i)}{p(\mathbf{x}_k|\boldsymbol{\theta})}, \tag{6}$$

we see that the gradient of the log-likelihood can be written in the interesting form

$$\nabla_{\boldsymbol{\theta}_i} l = \sum_{k=1}^{n} P(\omega_i|\mathbf{x}_k, \boldsymbol{\theta}) \nabla_{\boldsymbol{\theta}_i} \ln\, p(\mathbf{x}_k|\omega_i, \boldsymbol{\theta}_i). \tag{7}$$

Since the gradient must vanish at the value of $\boldsymbol{\theta}_i$ that maximizes l, the maximum-likelihood estimate $\hat{\boldsymbol{\theta}}_i$ must satisfy the conditions

$$\sum_{k=1}^{n} P(\omega_i|\mathbf{x}_k, \hat{\boldsymbol{\theta}}) \nabla_{\boldsymbol{\theta}_i} \ln\, p(\mathbf{x}_k|\omega_i, \hat{\boldsymbol{\theta}}_i) = 0, \quad i = 1, \ldots, c. \tag{8}$$

Among the solutions to these equations for $\hat{\boldsymbol{\theta}}_i$ we may find the maximum-likelihood solution.

It is not hard to generalize these results to include the prior probabilities $P(\omega_i)$ among the unknown quantities. In this case the search for the maximum value of $p(\mathcal{D}|\boldsymbol{\theta})$ extends over $\boldsymbol{\theta}$ and $P(\omega_i)$, subject to the constraints

$$P(\omega_i) \geq 0 \quad i = 1, \ldots, c \tag{9}$$

and

$$\sum_{i=1}^{c} P(\omega_i) = 1. \tag{10}$$

Let $\hat{P}(\omega_i)$ be the maximum-likelihood estimate for $P(\omega_i)$, and let $\hat{\boldsymbol{\theta}}_i$ be the maximum-likelihood estimate for $\boldsymbol{\theta}_i$. It can be shown (Problem **??**) that if the likelihood function is differentiable and if $\hat{P}(\omega_i) \neq 0$ for any i, then $\hat{P}(\omega_i)$ and $\hat{\boldsymbol{\theta}}_i$ must satisfy

$$\hat{P}(\omega_i) = \frac{1}{n} \sum_{k=1}^{n} \hat{P}(\omega_i | \mathbf{x}_k, \hat{\boldsymbol{\theta}}) \tag{11}$$

and

$$\sum_{k=1}^{n} \hat{P}(\omega_i | \mathbf{x}_k, \hat{\boldsymbol{\theta}}) \nabla_{\boldsymbol{\theta}_i} \ln\, p(\mathbf{x}_k | \omega_i, \hat{\boldsymbol{\theta}}_i) = 0, \tag{12}$$

where

$$\hat{P}(\omega_i | \mathbf{x}_k, \hat{\boldsymbol{\theta}}) = \frac{p(\mathbf{x}_k | \omega_i, \hat{\boldsymbol{\theta}}_i) \hat{P}(\omega_i)}{\sum_{j=1}^{c} p(\mathbf{x}_k | \omega_j, \hat{\boldsymbol{\theta}}_j) \hat{P}(\omega_j)}. \tag{13}$$

These equations have the following interpretation. Equation 11 states that the maximum-likelihood estimate of the probability of a category is the average over the entire data set of the estimate derived from each sample — each sample is weighted equally. Equation 13 is ultimately related to Bayes Theorem, but notice that in estimating the probability for class ω_i, the numerator on the right-hand side depends on $\hat{\boldsymbol{\theta}}_i$ and not the full $\hat{\boldsymbol{\theta}}$ directly. While Eq. 12 is a bit subtle, we can understand it clearly in the trivial $n = 1$ case. Since $\hat{P} \neq 0$, this case states merely that the probability density is maximized as a function of $\boldsymbol{\theta}_i$ — surely what is needed for the maximum-likelihood solution.

10.4 Application to Normal Mixtures

It is enlightening to see how these general results apply to the case where the component densities are multivariate normal, $p(\mathbf{x}|\omega_i, \boldsymbol{\theta}_i) \sim N(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$. The following table illustrates a few of the different cases that can arise depending upon which parameters are known ($\times$) and which are unknown (?):

Case	$\boldsymbol{\mu}_i$	$\boldsymbol{\Sigma}_i$	$P(\omega_i)$	c
1	?	$\times$	$\times$	$\times$
2	?	?	?	$\times$
3	?	?	?	?

Case 1 is the simplest, and will be considered in detail because of its pedagogical value. Case 2 is more realistic, though somewhat more involved. Case 3 represents the problem we face on encountering a completely unknown set of data; unfortunately, it cannot be solved by maximum-likelihood methods. We shall postpone discussion of what can be done when the number of classes is unknown until Sect. **??**.

10.4.1 Case 1: Unknown Mean Vectors

If the only unknown quantities are the mean vectors $\boldsymbol{\mu}_i$, then of course $\boldsymbol{\theta}_i$ consists of the components of $\boldsymbol{\mu}_i$. Equation 8 can then be used to obtain necessary conditions on the maximum-likelihood estimate for $\boldsymbol{\mu}_i$. Since the likelihood is

$$\ln p(\mathbf{x}|\omega_i, \boldsymbol{\mu}_i) = -\ln\left[(2\pi)^{d/2}|\boldsymbol{\Sigma}_i|^{1/2}\right] - \frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_i)^t\boldsymbol{\Sigma}_i^{-1}(\mathbf{x}-\boldsymbol{\mu}_i), \tag{14}$$

its derivative is

$$\nabla_{\boldsymbol{\mu}_i} \ln p(\mathbf{x}|\omega_i, \boldsymbol{\mu}_i) = \boldsymbol{\Sigma}_i^{-1}(\mathbf{x}-\boldsymbol{\mu}_i). \tag{15}$$

Thus according to Eq. 8, the maximum-likelihood estimate $\hat{\boldsymbol{\mu}}_i$ must satisfy

$$\sum_{k=1}^{n} P(\omega_i|\mathbf{x}_k, \hat{\boldsymbol{\mu}})\boldsymbol{\Sigma}_i^{-1}(\mathbf{x}_k - \hat{\boldsymbol{\mu}}_i) = 0, \quad \text{where } \hat{\boldsymbol{\mu}} = (\hat{\boldsymbol{\mu}}_1, \ldots, \hat{\boldsymbol{\mu}}_c). \tag{16}$$

After multiplying by $\boldsymbol{\Sigma}_i$ and rearranging terms, we obtain the solution:

$$\hat{\boldsymbol{\mu}}_i = \frac{\sum_{k=1}^{n} P(\omega_i|\mathbf{x}_k, \hat{\boldsymbol{\mu}})\mathbf{x}_k}{\sum_{k=1}^{n} P(\omega_i|\mathbf{x}_k, \hat{\boldsymbol{\mu}})}. \tag{17}$$

This equation is intuitively very satisfying. It shows that the maximum-likelihood estimate for $\boldsymbol{\mu}_i$ is merely a weighted average of the samples; the weight for the kth sample is an estimate of how likely it is that $\mathbf{x}_k$ belongs to the ith class. If $P(\omega_i|\mathbf{x}_k, \hat{\boldsymbol{\mu}})$ happened to be 1.0 for some of the samples and 0.0 for the rest, then $\hat{\boldsymbol{\mu}}_i$ would be the mean of those samples estimated to belong to the ith class. More generally, suppose that $\hat{\boldsymbol{\mu}}_i$ is sufficiently close to the true value of $\boldsymbol{\mu}_i$ that $P(\omega_i|\mathbf{x}_k, \hat{\boldsymbol{\mu}})$ is essentially the true posterior probability for ω_i. If we think of $P(\omega_i|\mathbf{x}_k, \hat{\boldsymbol{\mu}})$ as the fraction of those samples having value $\mathbf{x}_k$ that come from the ith class, then we see that Eq. 17 essentially gives $\hat{\boldsymbol{\mu}}_i$ as the average of the samples coming from the ith class.

Unfortunately, Eq. 17 does not give $\hat{\boldsymbol{\mu}}_i$ explicitly, and if we substitute

$$P(\omega_i|\mathbf{x}_k, \hat{\boldsymbol{\mu}}) = \frac{p(\mathbf{x}_k|\omega_i, \hat{\boldsymbol{\mu}}_i)P(\omega_i)}{\sum_{j=1}^{c} p(\mathbf{x}_k|\omega_j, \hat{\boldsymbol{\mu}}_j)P(\omega_j)}$$

with $p(\mathbf{x}|\omega_i, \hat{\boldsymbol{\mu}}_i) \sim N(\hat{\boldsymbol{\mu}}_i, \boldsymbol{\Sigma}_i)$, we obtain a tangled snarl of coupled simultaneous nonlinear equations. These equations usually do not have a unique solution, and we must test the solutions we get to find the one that actually maximizes the likelihood.

If we have some way of obtaining fairly good initial estimates $\hat{\boldsymbol{\mu}}_i(0)$ for the unknown means, Eq. 17 suggests the following iterative scheme for improving the estimates:

$$\hat{\boldsymbol{\mu}}_i(j+1) = \frac{\sum_{k=1}^{n} P(\omega_i|\mathbf{x}_k, \hat{\boldsymbol{\mu}}(j))\mathbf{x}_k}{\sum_{k=1}^{n} P(\omega_i|\mathbf{x}_k, \hat{\boldsymbol{\mu}}(j))} \tag{18}$$

This is basically a gradient ascent or hill-climbing procedure for maximizing the log-likelihood function. If the overlap between component densities is small, then the

coupling between classes will be small and convergence will be fast. However, when convergence does occur, all that we can be sure of is that the gradient is zero. Like all hill-climbing procedures, this one carries no guarantee of yielding the global maximum (Computer exercise 19). Note too that if the model is mis-specified (for instance we assume the "wrong" number of clusters) then the log-likelihood can actually decrease (Computer exercise 21).

Example 1: Mixtures of two 1D Gaussians

To illustrate the kind of behavior that can occur, consider the simple two-component one-dimensional normal mixture:

$$p(x|\mu_1,\mu_2) = \underbrace{\frac{1}{3\sqrt{2\pi}}\exp\left[-\frac{1}{2}(x-\mu_1)^2\right]}_{\omega_1} + \underbrace{\frac{2}{3\sqrt{2\pi}}\exp\left[-\frac{1}{2}(x-\mu_2)^2\right]}_{\omega_2},$$

where ω_i denotes a Gaussian component. The 25 samples shown in the table were drawn sequentially from this mixture with $\mu_1 = -2$ and $\mu_2 = 2$. Let us use these samples to compute the log-likelihood function

$$l(\mu_1,\mu_2) = \sum_{k=1}^{n} \ln\, p(x_k|\mu_1,\mu_2)$$

for various values of μ_1 and μ_2. The bottom figure shows how l varies with μ_1 and μ_2. The maximum value of l occurs at $\hat{\mu}_1 = -2.130$ and $\hat{\mu}_2 = 1.668$, which is in the rough vicinity of the true values $\mu_1 = -2$ and $\mu_2 = 2$. However, l reaches another peak of comparable height at $\hat{\mu}_1 = 2.085$ and $\hat{\mu}_2 = -1.257$. Roughly speaking, this solution corresponds to interchanging μ_1 and μ_2. Note that had the prior probabilities been equal, interchanging μ_1 and μ_2 would have produced no change in the log-likelihood function. Thus, as we mentioned before, when the mixture density is not identifiable, the maximum-likelihood solution is not unique.

k	x_k	ω_1	ω_2
1	0.608		×
2	-1.590	×	
3	0.235		×
4	3.949		×
5	-2.249	×	
6	2.704		×
7	-2.473	×	
8	0.672		×

k	x_k	ω_1	ω_2
9	0.262		×
10	1.072		×
11	-1.773	×	
12	0.537		×
13	3.240		×
14	2.400		×
15	-2.499	×	
16	2.608		×

k	x_k	ω_1	ω_2
17	-3.458	×	
18	0.257		×
19	2.569		×
20	1.415		×
21	1.410		×
22	-2.653	×	
23	1.396		×
24	3.286		×
25	-0.712	×	

Additional insight into the nature of these multiple solutions can be obtained by examining the resulting estimates for the mixture density. The figure at the top shows the true (source) mixture density and the estimates obtained by using the two maximum-likelihood estimates as if they were the true parameter values. The 25 sample values are shown as a scatter of points along the abscissa — ω_1 points in

10.4.2 Case 2: All Parameters Unknown

If $\boldsymbol{\mu}_i$, $\boldsymbol{\Sigma}_i$, and $P(\omega_i)$ are all unknown, and if no constraints are placed on the covariance matrix, then the maximum-likelihood principle yields useless singular solutions. The reason for this can be appreciated from the following simple example in one dimension. Let $p(x|\mu,\sigma^2)$ be the two-component normal mixture:

$$p(x|\mu,\sigma^2) = \frac{1}{2\sqrt{2\pi}\sigma}\exp\left[-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right] + \frac{1}{2\sqrt{2\pi}}\exp\left[-\frac{1}{2}x^2\right].$$

The likelihood function for n samples drawn from this probability density is merely the product of the n densities $p(x_k|\mu,\sigma^2)$. Suppose that we let $\mu = x_1$, the value of the first sample. In this situation the density is

$$p(x|\mu,\sigma^2) = \frac{1}{2\sqrt{2\pi}\sigma} + \frac{1}{2\sqrt{2\pi}}\exp\left[-\frac{1}{2}x^2\right].$$

Clearly, for the rest of the samples

$$p(x_k|\mu,\sigma^2) \geq \frac{1}{2\sqrt{2\pi}}\exp\left[-\frac{1}{2}x_k^2\right],$$

so that

$$p(x_1,\ldots,x_n|\mu,\sigma^2) \geq \left\{\frac{1}{\sigma} + \exp\left[-\frac{1}{2}x_1^2\right]\right\}\frac{1}{(2\sqrt{2\pi})^n}\exp\left[-\frac{1}{2}\sum_{k=2}^{n}x_k^2\right].$$

Thus, the first term at the right shows that by letting σ approach zero we can make the likelihood arbitrarily large, and the maximum-likelihood solution is singular.

Ordinarily, singular solutions are of no interest, and we are forced to conclude that the maximum-likelihood principle fails for this class of normal mixtures. However, it is an empirical fact that meaningful solutions can still be obtained if we restrict our attention to the largest of the finite local maxima of the likelihood function. Assuming that the likelihood function is well behaved at such maxima, we can use Eqs. 11 – 13 to obtain estimates for $\boldsymbol{\mu}_i$, $\boldsymbol{\Sigma}_i$, and $P(\omega_i)$. When we include the elements of $\boldsymbol{\Sigma}_i$ in the elements of the parameter vector $\boldsymbol{\theta}_i$, we must remember that only half of the off-diagonal elements are independent. In addition, it turns out to be much more convenient to let the independent elements of $\boldsymbol{\Sigma}_i^{-1}$ rather than $\boldsymbol{\Sigma}_i$ be the unknown parameters. With these observations, the actual differentiation of

$$\ln\, p(\mathbf{x}_k|\omega_i,\boldsymbol{\theta}_i) = \ln\frac{|\boldsymbol{\Sigma}_i^{-1}|^{1/2}}{(2\pi)^{d/2}} - \frac{1}{2}(\mathbf{x}_k-\boldsymbol{\mu}_i)^t\boldsymbol{\Sigma}_i^{-1}(\mathbf{x}_k-\boldsymbol{\mu}_i)$$

e

with respect to the elements of $\boldsymbol{\mu}_i$ and $\boldsymbol{\Sigma}_i^{-1}$ is relatively routine. Let $x_p(k)$ be the pth element of $\mathbf{x}_k$, $\mu_p(i)$ be the pth element of $\boldsymbol{\mu}_i$, $\sigma_{pq}(i)$ be the pqth element of $\boldsymbol{\Sigma}_i$, and $\sigma^{pq}(i)$ be the pqth element of $\boldsymbol{\Sigma}_i^{-1}$. Then differentiation gives

$$\nabla_{\boldsymbol{\mu}_i} \ln\, p(\mathbf{x}_k|\omega_i, \boldsymbol{\theta}_i) = \boldsymbol{\Sigma}_i^{-1}(\mathbf{x}_k - \boldsymbol{\mu}_i)$$

and

$$\frac{\partial \ln\, p(\mathbf{x}_k|\omega_i, \boldsymbol{\theta}_i)}{\partial \sigma^{pq}(i)} = \Big(1 - \frac{\delta_{pq}}{2}\Big)\big[\sigma_{pq}(i) - (x_p(k) - \mu_p(i))(x_q(k) - \mu_q(i))\big],$$

where δ_{pq} is the Kronecker delta. We substitute these results in Eq. 12 and perform a small amount of algebraic manipulation (Problem 16) and thereby obtain the following equations for the local-maximum-likelihood estimate $\hat{\boldsymbol{\mu}}_i$, $\hat{\boldsymbol{\Sigma}}_i$, and $\hat{P}(\omega_i)$:

$$\hat{P}(\omega_i) = \frac{1}{n}\sum_{k=1}^{n} \hat{P}(\omega_i|\mathbf{x}_k, \hat{\boldsymbol{\theta}}) \tag{19}$$

$$\hat{\boldsymbol{\mu}}_i = \frac{\sum_{k=1}^{n} \hat{P}(\omega_i|\mathbf{x}_k, \hat{\boldsymbol{\theta}})\mathbf{x}_k}{\sum_{k=1}^{n} \hat{P}(\omega_i|\mathbf{x}_k, \hat{\boldsymbol{\theta}})} \tag{20}$$

$$\hat{\boldsymbol{\Sigma}}_i = \frac{\sum_{k=1}^{n} \hat{P}(\omega_i|\mathbf{x}_k, \hat{\boldsymbol{\theta}})(\mathbf{x}_k - \boldsymbol{\mu}_i)(\mathbf{x}_k - \boldsymbol{\mu}_i)^t}{\sum_{k=1}^{n} \hat{P}(\omega_i|\mathbf{x}_k, \hat{\boldsymbol{\theta}})} \tag{21}$$

where

$$\begin{aligned}\hat{P}(\omega_i|\mathbf{x}_k, \hat{\boldsymbol{\theta}}) &= \frac{p(\mathbf{x}_k|\omega_i, \hat{\boldsymbol{\theta}}_i)\hat{P}(\omega_i)}{\sum_{j=1}^{c} p(\mathbf{x}_k|\omega_j, \hat{\boldsymbol{\theta}}_j)\hat{P}(\omega_j)} \\ &= \frac{|\hat{\boldsymbol{\Sigma}}_i|^{-1/2}\, \exp\big[-\tfrac{1}{2}(\mathbf{x}_k - \hat{\boldsymbol{\mu}}_i)^t\hat{\boldsymbol{\Sigma}}_i^{-1}(\mathbf{x}_k - \hat{\boldsymbol{\mu}}_i)\big]\hat{P}(\omega_i)}{\sum_{j=1}^{c} |\hat{\boldsymbol{\Sigma}}_j|^{-1/2}\, \exp\big[-\tfrac{1}{2}(\mathbf{x}_k - \hat{\boldsymbol{\mu}}_j)^t\hat{\boldsymbol{\Sigma}}_j^{-1}(\mathbf{x}_k - \hat{\boldsymbol{\mu}}_j)\big]\hat{P}(\omega_j)}.\end{aligned} \tag{22}$$

While the notation may make these equations appear to be rather formidable, their interpretation is actually quite simple. In the extreme case where $\hat{P}(\omega_i|\mathbf{x}_k, \hat{\boldsymbol{\theta}})$ is 1.0 when $\mathbf{x}_k$ is from Class ω_i and 0.0 otherwise, $\hat{P}(\omega_i)$ is the fraction of samples from ω_i, $\hat{\boldsymbol{\mu}}_i$ is the mean of those samples, and $\hat{\boldsymbol{\Sigma}}_i$ is the corresponding sample covariance matrix. More generally, $\hat{P}(\omega_i|\mathbf{x}_k, \hat{\boldsymbol{\theta}})$ is between 0.0 and 1.0, and all of the samples play some role in the estimates. However, the estimates are basically still frequency ratios, sample means, and sample covariance matrices.

The problems involved in solving these implicit equations are similar to the problems discussed in Sect. **??**, with the additional complication of having to avoid singular solutions. Of the various techniques that can be used to obtain a solution, the most obvious approach is to use initial estimates to evaluate Eq. 22 for $\hat{P}(\omega_i|\mathbf{x}_k, \hat{\boldsymbol{\theta}})$ and then

e

2 EM

-missing values

- or missing labels

E step: expecteation of the likelihood , involvex marginalization over the missing values

M step : recompute the parameters that maximize the likelihood

where Φ includes the priors $P(\mathcal{G}_i)$ and also the sufficient statistics of the component densities $p(\boldsymbol{x}^t|\mathcal{G}_i)$. Unfortunately, we cannot solve for the parameters analytically and need to resort to iterative optimization.

The *Expectation-Maximization* (EM) algorithm (Dempster, Laird, and Rubin 1977; Redner and Walker 1984) is used in maximum likelihood estimation where the problem involves two sets of random variables of which one, X, is observable and the other, Z, is hidden. The goal of the algorithm is to find the parameter vector Φ that maximizes the likelihood of the observed values of X, $\mathcal{L}(\Phi|\mathcal{X})$. But in cases where this is not feasible, we associate the extra *hidden variables* Z and express the underlying model using both, to maximize the likelihood of the joint distribution of X and Z, the *complete* likelihood $\mathcal{L}_c(\Phi|\mathcal{X},\mathcal{Z})$.

Since the Z values are not observed, we cannot work directly with the complete data likelihood $\mathcal{L}_c$, instead we work with its expectation, $\mathcal{Q}$, given $\mathcal{X}$ and the current parameter values Φ^l, where l indexes iteration. This is the *expectation* (E) step of the algorithm. Then in the *maximization* (M) step, we look for the new parameter values, Φ^{l+1}, that maximize this. Thus

$$\begin{aligned}\text{E-step} &: \mathcal{Q}(\Phi|\Phi^l) = E[\mathcal{L}_c(\Phi|\mathcal{X},\mathcal{Z})|\mathcal{X},\Phi^l]\\ \text{M-step} &: \Phi^{l+1} = \arg\max_{\Phi} \mathcal{Q}(\Phi|\Phi^l)\end{aligned}$$

Dempster, Laird, and Rubin (1977) proved that an increase in $\mathcal{Q}$ implies an increase in the incomplete likelihood

$$\mathcal{L}(\Phi^{l+1}|\mathcal{X}) \geq \mathcal{L}(\Phi^l|\mathcal{X})$$

In the case of mixtures, the hidden variables are the sources of observations, namely, which observation belongs to which component. If these were given, for example, as class labels in a supervised setting, we would know which parameters to adjust to fit that data point. The EM algorithm works as follows: In the E-step we estimate these labels given our current knowledge of components, and in the M-step we update our class knowledge given the labels estimated in the E-step. These two steps are the same as the two steps of k-means; calculation of b_i^t (E-step) and reestimation of $\boldsymbol{m}_i$ (M-step).

We define a vector of *indicator variables* $\boldsymbol{z}^t = \{z_1^t, \ldots, z_k^t\}$ where $z_i^t = 1$ if $\boldsymbol{x}^t$ belongs to cluster $\mathcal{G}_i$, and 0 otherwise. $\boldsymbol{z}$ is a multinomial distribu-

tion from k categories with prior probabilities π_i, shorthand for $P(\mathcal{G}_i)$. Then

$$P(\mathbf{z}^t) = \prod_{i=1}^{k} \pi_i^{z_i^t}$$

The likelihood of an observation $\mathbf{x}^t$ is equal to its probability specified by the component that generated it:

$$p(\mathbf{x}^t|\mathbf{z}^t) = \prod_{i=1}^{k} p_i(\mathbf{x}^t)^{z_i^t}$$

$p_i(\mathbf{x}^t)$ is shorthand for $p(\mathbf{x}^t|\mathcal{G}_i)$. The joint density is

$$p(\mathbf{x}^t, \mathbf{z}^t) = P(\mathbf{z}^t)p(\mathbf{x}^t|\mathbf{z}^t)$$

and the complete data likelihood of the iid sample $\mathcal{X}$ is

$$\begin{aligned}
\mathcal{L}_c(\Phi|\mathcal{X}, \mathcal{Z}) &= \log \prod_t p(\mathbf{x}^t, \mathbf{z}^t|\Phi) \\
&= \sum_t \log p(\mathbf{x}^t, \mathbf{z}^t|\Phi) \\
&= \sum_t \log P(\mathbf{z}^t|\Phi) + \log p(\mathbf{x}^t|\mathbf{z}^t, \Phi) \\
&= \sum_t \sum_i z_i^t [\log \pi_i + \log p_i(\mathbf{x}^t|\Phi)]
\end{aligned}$$

E-step: We define

$$\begin{aligned}
Q(\Phi|\Phi^l) &\equiv E\left[\log P(X, Z)|\mathcal{X}, \Phi^l\right] \\
&= E\left[\mathcal{L}_c(\Phi|\mathcal{X}, \mathcal{Z})|\mathcal{X}, \Phi^l)\right] \\
&= \sum_t \sum_i E[z_i^t|\mathcal{X}, \Phi^l][\log \pi_i + \log p_i(\mathbf{x}^t|\Phi^l)]
\end{aligned}$$

where

$$\begin{aligned}
E[z_i^t|\mathcal{X}, \Phi^l] &= E[z_i^t|\mathbf{x}^t, \Phi^l] \quad \mathbf{x}^t \text{ are iid} \\
&= P(z_i^t = 1|\mathbf{x}^t, \Phi^l) \quad z_i^t \text{ is a 0/1 random variable} \\
&= \frac{p(\mathbf{x}^t|z_i^t = 1, \Phi^l)P(z_i^t = 1|\Phi^l)}{p(\mathbf{x}^t|\Phi^l)} \quad \text{Bayes' rule} \\
&= \frac{p_i(\mathbf{x}^t|\Phi^l)\pi_i}{\sum_j p_j(\mathbf{x}^t|\Phi^l)\pi_j}
\end{aligned}$$

e

$$
\begin{aligned}
&= \frac{p(\mathbf{x}^t|\mathcal{G}_i,\Phi^l)P(\mathcal{G}_i)}{\sum_j p(\mathbf{x}^t|\mathcal{G}_j,\Phi^l)P(\mathcal{G}_j)} \\
&= P(\mathcal{G}_i|\mathbf{x}^t,\Phi^l) \equiv h_i^t
\end{aligned}
$$

We see that the expected value of the hidden variable, $E[z_i^t]$, is the posterior probability that $\mathbf{x}^t$ is generated by component $\mathcal{G}_i$. Because this is a probability, it is between 0 and 1 and is a "soft" label, as opposed to the 0/1 "hard" label of k-means.

M-step: We maximize $\mathcal{Q}$ to get the next set of parameter values Φ^{l+1}:

$$
\Phi^{l+1} = \arg\max_{\Phi} \mathcal{Q}(\Phi|\Phi^l)
$$

which is

$$
\begin{aligned}
\mathcal{Q}(\Phi|\Phi^l) &= \sum_t \sum_i h_i^t [\log \pi_i + \log p_i(\mathbf{x}^t|\Phi^l)] \\
&= \sum_t \sum_i h_i^t \log \pi_i + \sum_t \sum_i h_i^t \log p_i(\mathbf{x}^t|\Phi^l)
\end{aligned}
$$

The second term is independent of π_i and using the constraint that $\sum_i \pi_i = 1$ as the Lagrangian, we solve for

$$
\nabla_{\pi_i} \sum_t \sum_i h_i^t \log \pi_i - \lambda \left(\sum_i \pi_i - 1 \right) = 0
$$

and get

$$
\pi_i = \frac{\sum_t h_i^t}{N}
$$

which is analogous to the calculation of priors in equation 7.2.

Similarly, the first term of equation 7.10 is independent of the components and can be dropped while estimating the parameters of the components. We solve for

$$
\nabla_{\Phi} \sum_t \sum_i h_i^t \log p_i(\mathbf{x}^t|\Phi) = 0
$$

If we assume Gaussian components, $\hat{p}_i(\mathbf{x}^t|\Phi) \sim \mathcal{N}(\mathbf{m}_i, \mathbf{S}_i)$, the M-step is

$$
\begin{aligned}
\mathbf{m}_i^{l+1} &= \frac{\sum_t h_i^t \mathbf{x}^t}{\sum_t h_i^t} \\
\mathbf{S}_i^{l+1} &= \frac{\sum_t h_i^t (\mathbf{x}^t - \mathbf{m}_i^{l+1})(\mathbf{x}^t - \mathbf{m}_i^{l+1})^T}{\sum_t h_i^t}
\end{aligned}
$$

e

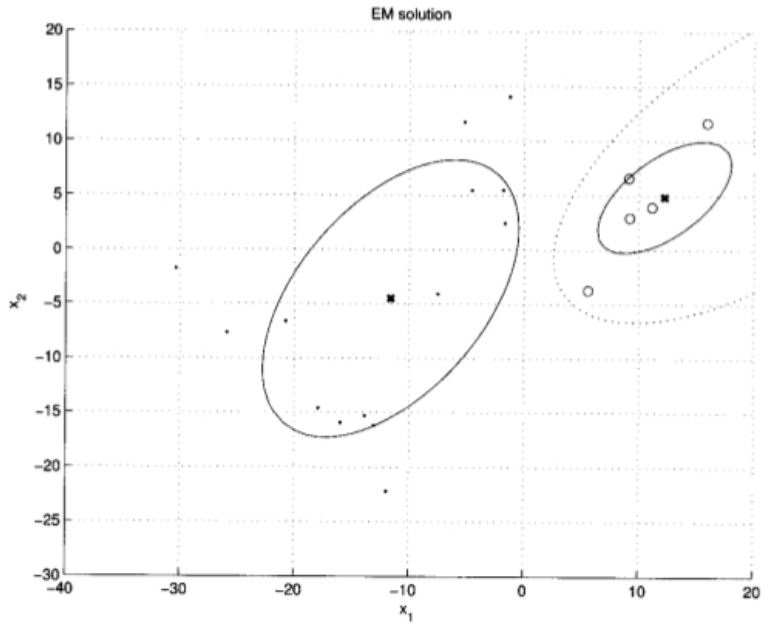


Figure 7.4 Data points and the fitted Gaussians by EM, initialized by one k-means iteration of figure 7.2. Unlike in k-means, as can be seen, EM allows estimating the covariance matrices. The data points labeled by greater h_i, the contours of the estimated Gaussian densities, and the separating curve of $h_i = 0.5$ (dashed line) are shown.

where, for Gaussian components in the E-step, we calculate

$$h_i^t = \frac{\pi_i |\mathbf{S}_i|^{-1/2} \exp[-(1/2)(\mathbf{x}^t - \mathbf{m}_i)^T \mathbf{S}_i^{-1}(\mathbf{x}^t - \mathbf{m}_i)]}{\sum_j \pi_j |\mathbf{S}_j|^{-1/2} \exp[-(1/2)(\mathbf{x}^t - \mathbf{m}_j)^T \mathbf{S}_j^{-1}(\mathbf{x}^t - \mathbf{m}_j)]}$$

Again, the similarity between equations 7.13 and 7.2 is not accidental; the estimated soft labels h_i^t replace the actual (unknown) labels r_i^t.

EM is initalized by k-means. After a few iterations of k-means, we get the estimates for the centers $\mathbf{m}_i$ and using the instances covered by each center, we estimate the $\mathbf{S}_i$ and $\sum_t b_i^t / N$ give us the π_i. We run EM from that point on, as shown in figure 7.4.

Just as in parametric classification (section 5.5), with small samples and large dimensionality we can regularize by making simplifying assumptions. When $\hat{p}_i(\mathbf{x}^t|\Phi) \sim \mathcal{N}(\mathbf{m}_i, \mathbf{S})$, the case of a shared covariance matrix,

e

equation 7.12 reduces to

$$\min_{\boldsymbol{m}_i,\mathbf{S}} \sum_t \sum_i h_i^t (\boldsymbol{x}^t - \boldsymbol{m}_i)^T \mathbf{S}^{-1} (\boldsymbol{x}^t - \boldsymbol{m}_i)$$

When $\hat{p}_i(\boldsymbol{x}^t|\Phi) \sim \mathcal{N}(\boldsymbol{m}_i, s^2\mathbf{I})$, the case of a shared diagonal matrix, we have

$$\min_{\boldsymbol{m}_i,s} \sum_t \sum_i h_i^t \frac{\|\boldsymbol{x}^t - \boldsymbol{m}_i\|^2}{s^2}$$

which is the reconstruction error we defined in k-means clustering (equation 7.3). The difference is that now

$$h_i^t = \frac{\exp\left[-(1/2s^2)\|\boldsymbol{x}^t - \boldsymbol{m}_i\|^2\right]}{\sum_j \exp\left[-(1/2s^2)\|\boldsymbol{x}^t - \boldsymbol{m}_j\|^2\right]}$$

is a probability between 0 and 1. b_i^t of k-means clustering makes a hard 0/1 decision, whereas h_i^t is a *soft label* that assigns the input to a cluster with a certain probability. When h_i^t are used instead of b_i^t, an instance contributes to the update of parameters of all components, to each with a certain probability. This is especially useful if the instance is close to the midpoint between two centers. We thus see that k-means clustering is a special case of EM applied to Gaussian mixtures where inputs are assumed independent with equal and shared variances and where labels are hardened. k-means thus pave the input density with circles, whereas EM in the general case uses ellipses of arbitrary shapes and orientations.

e

3 Clustering. K-means

10.4.3 K-means clustering

Of the various techniques that can be used to simplify the computation and accelerate convergence, we shall briefly consider one elementary, approximate method. From Eq. 22, it is clear that the probability $\hat{P}(\omega_i|\mathbf{x}_k, \hat{\boldsymbol{\theta}})$ is large when the squared Mahalanobis distance $(\mathbf{x}_k - \hat{\boldsymbol{\mu}}_i)^t \hat{\boldsymbol{\Sigma}}_i^{-1}(\mathbf{x}_k - \hat{\boldsymbol{\mu}}_i)$ is small. Suppose that we merely compute the squared Euclidean distance $\|\mathbf{x}_k - \hat{\boldsymbol{\mu}}_i\|^2$, find the mean $\hat{\boldsymbol{\mu}}_m$ nearest to $\mathbf{x}_k$, and approximate $\hat{P}(\omega_i|\mathbf{x}_k, \hat{\boldsymbol{\theta}})$ as

$$\hat{P}(\omega_i|\mathbf{x}_k, \hat{\boldsymbol{\theta}}) \approx \begin{cases} 1 & \text{if } i = m \\ 0 & \text{otherwise.} \end{cases} \tag{23}$$

Then the iterative application of Eq. 20 leads to the following procedure for finding $\hat{\boldsymbol{\mu}}_1, \ldots, \hat{\boldsymbol{\mu}}_c$. (Although the algorithm is historically referred to as k-means clustering, we retain the notation c, our symbol for the number of clusters.)

Algorithm 1 (K-means clustering)

```
1 begin initialize n, c, μ_1, μ_2, ..., μ_c
2         do classify n samples according to nearest μ_i
3             recompute μ_i
4         until no change in μ_i
5   return μ_1, μ_2, ..., μ_c
6 end
```

The computational complexity of the algorithm is $O(ndcT)$ where d the number of features and T the number of iterations (Problem 15). In practice, the number of iterations is generally much less than the number of samples.

This is typical of a class of procedures that are known as *clustering* procedures or algorithms. Later on we shall place it in the class of iterative optimization procedures, since the means tend to move so as to minimize a squared-error criterion function. For the moment we view it merely as an approximate way to obtain maximum-likelihood estimates for the means. The values obtained can be accepted as the answer, or can be used as starting points for the more exact computations.

It is interesting to see how this procedure behaves on the example data we saw in Example 1. Figure 10.1 shows the sequence of values for $\hat{\mu}_1$ and $\hat{\mu}_2$ obtained for several different starting points. Since interchanging $\hat{\mu}_1$ and $\hat{\mu}_2$ merely interchanges the labels assigned to the data, the trajectories are symmetric about the line $\hat{\mu}_1 = \hat{\mu}_2$. The trajectories lead either to the point $\hat{\mu}_1 = -2.176, \hat{\mu}_2 = 1.684$ or to its symmetric

4 Clustering. Graphical models

single link clusters (Connected component) subgraphsuch that each node is connected to at least one other node in the subgraphand the set of nodes is maximal with respect to that property

complete link clusters(Maximal complete subgraph) subgraphsuch that each node is connected to every other node in the subgraph(clique)

average link clusters each cluster member has a greater average similarity to the remaining members of

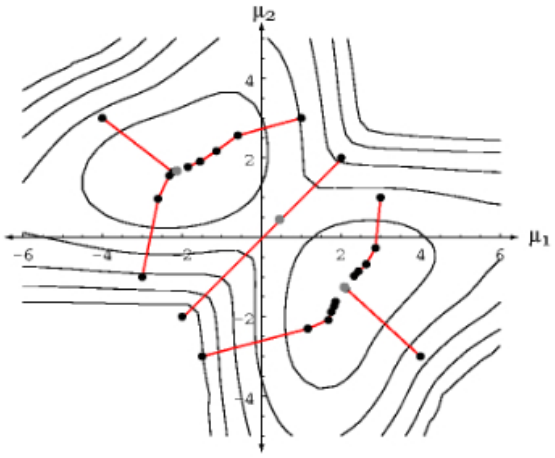


Figure 10.1: The k-means clustering procedure is a form of stochastic hill climbing in the log-likelihood function. The contours represent equal log-likelihood values for the one-dimensional data in Example 1. The dots indicate parameter values after different iterations of the k-means algorithm. Six of the starting points shown lead to local maxima, whereas two (i.e., $\mu_1(0) = \mu_2(0)$) lead to a saddle point near $\boldsymbol{\mu} = \mathbf{0}$.

image. This is close to the solution found by the maximum-likelihood method (viz., $\hat{\mu}_1 = -2.130$ and $\hat{\mu}_2 = 1.688$), and the trajectories show a general resemblance to those shown in Example 1. In general, when the overlap between the component densities is small the maximum-likelihood approach and the k-means procedure can be expected to give similar results.

Figure 10.2 shows a two-dimensional example, with the assumption of $c = 3$ clusters. The three initial cluster centers, chosen randomly from the training points, and their associated Voronoi tesselation, are shown in pink. According to the algorithm, the points in each of the three Voronoi cells are used to calculate new cluster centers (dark pink), and so on. Here, after the third iteration the algorithm has converged (red). Because the k-means algorithm is very simple and works well in practice, it is a staple of clustering methods.

the cluster than it does to all members of any other cluster

- Cluster Representations
  - decide apriori # of clusters or cluster seeds
  - usualy produce partitions
  - usualy run in $O(n)$ or $O(n \log n)$

Graph Theoretic Aproaches - assume graph of objects connected by links w/ similarity greater than some threshold

Single link - if A & B are connected they should be in the same cluster

Complete link - all items in cluster must be connected

Average link - all cluster members must have a greater avg similarity to other cluster members than avg similarity to any other group

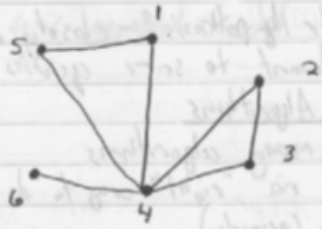
Star Clustering - find greatest # links + cluster them find next greatest # links + cluster

Example

Pick threshold: 0.65

	1	2	3	4	5
1					
2	.6				
3	.6	.8			
4	.9	.7	.7		
5	.9	.6	.6	.9	
6	.5	.5	.5	.9	.5

Threshold Similarity Graph



Clusters:

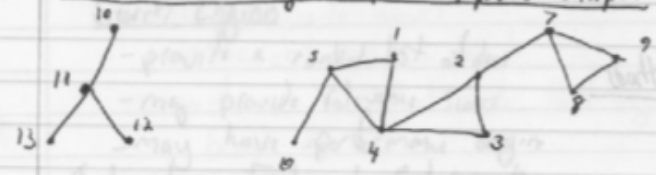
① Single Link: - one cluster (everyone is connected) {1, 2, 3, 4, 5, 6}

② Complete Link - {{1, 4, 5}, {2, 3, 4}, {4, 6}}

③ Star Clustering {1, 2, 3, 4, 5, 6}
star center

Threshold Similarity Graph (superset example)

can choose overlaps or non-overlaps



① Single Link: 2 clusters
{(1, 2, ... 9), (10, 11, 12, 13)}

② Complete Link: {(1,4,5), (2,3,4), (2,8,9), (2,7), (5,6), (10,11), (11,12), (11,13)}
8 clusters

③ Star clusters: {(1, 2, 3, 4, 5), (7, 8, 9, 2), (11, 12, 13, 10), (6, 5)}