Chapter 2 Clustering and Mixture Models

2.1 Clustering of Data

We have laid out our overall goal as follows: given a group of observations $\mathcal{X} = \{x_i \mid i = 1 \ldots N\}$, x_i not necessarily univariate or independent, discover the structure of the stochastic process from which the data arose. In this chapter we will investigate one particular form of structure: we will examine ways to discover if the data fall naturally into distinct clusters of points.

Clustering has a long history of essentially *ad hoc* techniques (Duda and Hart 1973; Jain and Dubes 1988). In recent years, however, considerable progress has been made with various statistically well-founded techniques. In our treatment of the problem we will pass very quickly to one particular statistical model, the **mixture**, which will be seen to be a particularly simple example of a latent variable model.

In general, the clustering problem assumes that the observations are independent and identically distributed (iid), and further that some measure of dissimilarity between observations is available. This measure may be quite general; there is no need for it be symmetric, to obey the triangle inequality, or even to be always nonnegative. Many of the techniques which work with these weak assumptions are fundamentally **agglomerative**, that is they form the data into progressively larger clusters by merging together smaller groups that display significant similarity. We shall not discuss such algorithms; many examples are reviewed by Jain and Dubes (1988).

Probabilistic models require well-defined measures in the space of observations, which in turn require a defined metric. Thus, we will examine clustering problems where the similarity measure obeys all the requirements of a metric. Indeed, we will go further and assume that each of our observations defines a point in $\mathbb{R}^D$, and that the similarity measure is simply the Euclidean distance between the points. In particular, this assumption allows us to speak of distances to points that were not observed, and thus to speak quantitatively of the *process* that generated the data, something not always possible in the extremely general spaces.

In this early treatment we shall also assume that the number of clusters, M, is known. Once we have achieved a properly probabilistic framework, the problem of determining the number of clusters will be reduced to that of model selection and so the techniques of the previous chapter will become applicable.

A particularly straightforward criterion for the assignment of D-dimensional observations $\{x_i\}$ to M clusters is as follows. We associate with each cluster a central point $\mu_m \in \mathbb{R}^D, m = 1 \ldots M$, and then require that the sum of the squared distances from each point to the center of its assigned cluster be minimal. For this to be the case, it is clear that μ_m must be the mean of the observations assigned to the mth cluster, hence this approach is often referred to as the **k-means** clustering criterion (McQueen 1967). (The 'k' in k-means refers to the number of clusters, a quantity for which we have chosen the symbol M.)

The clustering is fully specified by the location of the μ_m, since the assignments of the x_i are then determined by which mean is closest. How are we to find the optimal locations of the μ_m? Iterative algorithms to do this have been known since the 60's. The basic approach was provided by Forgy (1965) (this approach is also known, in the related vector quantization literature, as the Lloyd–Max algorithm). We begin with an initial, random partitioning of the data into M sets. The μ_m are placed at the means of these data sets. We then iterate the following two steps

1. Re-assign all data points to the closest μ_m.
2. Move each μ_m is the mean of its assigned data.

This basic iteration (which, as we will see, is quite reminiscient of the EM algorithm) is what we shall call the k-means algorithm.

A number of variants of this basic approach have been suggested. For completeness, we mention them here; no details will be provided and we will not encounter them again in this dissertation, preferring instead the probabilistic approach described below. A more complete review is available in Duda and Hart (1973), Jain and Dubes (1988) or Ripley (1996).

The ISODATA algorithm (Hall and Ball 1965; see also Duda and Hart 1973) introduces an additional step to the iteration above, in which the number of clusters may be adjusted. Hartigan and Wong (1979) re-assign only one data point at a time, updating the means each time a point changes hands. McQueen (1967) gives an incremental algorithm, in which data are considered one-by-one in a single pass and the corresponding cluster mean updated after each assignment. Adaptive resonance theory (ART) (Carpenter and Grossberg 1987a, 1987b, 1990) provides a similar scheme within a "neural" framework; rather than choosing the closest mean, the data point is compared to each in a set order and the assignment is made to the first cluster for which the data point falls within a distance threshold. In addition, the distortion measures involved in ART are not exactly the squared-distance measures of the other techniques.

2.2 A Statistical Interpretation

As presented, the k-means and related algorithms appear *ad hoc*, but in fact they can be given a statistical interpretation (Scott and Symons 1971). We note that the sum of squared distance from μ_m is (up to a normalization constant) the negative log-likelihood of the model that the data are generated by an isotropic (that is, identity covariance matrix) multivariate Gaussian distribution with mean μ_m. Thus, we can introduce the following likelihood function

$$\mathcal{L}_{\mathcal{X}}\left(\{\mu_m\}, \mathcal{Y}\right) = \prod_i G(x_i - \mu_{y_i}) \tag{2.1}$$

where $\mathcal{Y} = \{y_i\}$ is a set of assignment variables taking values between 1 and M, which tell us in which cluster the each observation falls, while $G(\cdot)$ denotes a standard multivariate Gaussian density with mean 0 and covariance I. The values of $\{\mu_m\}$ and $\mathcal{Y}$ which maximize this likelihood are precisely the solutions to the k-means sum-of-squares criterion. We have therefore converted our clustering problem into maximum-likelihood estimation.

This viewpoint also allows us to easily generalize the sum-of-squares criterion. In place of the isotropic Gaussian, we might choose Gaussians with arbitrary covariance matrices, so that each cluster is ellipsoidal but can have a different size and orientation. Indeed, we can in general choose any parameterized family of densities, and require that each cluster be represented by one of them (Scott and Symons 1971; Banfield and Raftery 1993). The likelihood is then

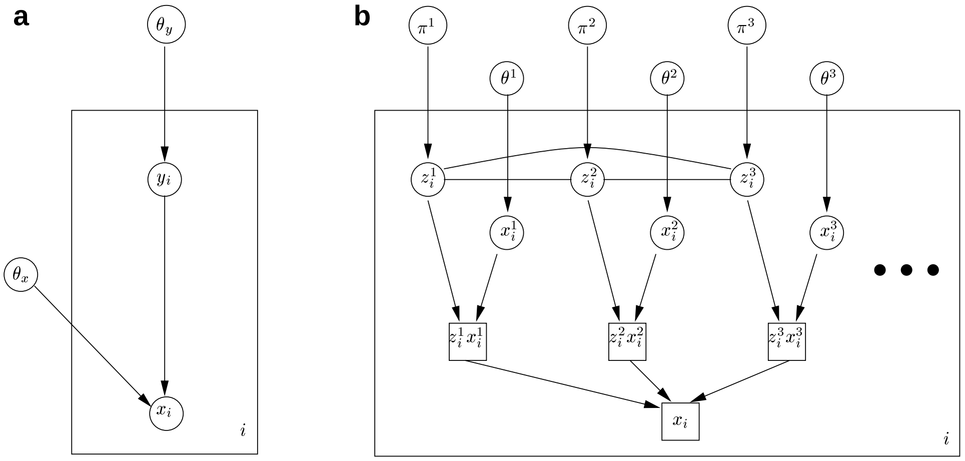
$$\mathcal{L}_{\mathcal{X}}\left(\theta, \mathcal{Y}\right) = \prod_i \mathsf{P}_{\theta_{y_i}}\left(x_i\right) \tag{2.2}$$

where the $\theta_m, m = 1 \ldots M$ parameterize the densities. If we are to retain the intuitive notion of a cluster being spatially compact we would expect the densities to all be well localized. Algorithms to maximize these likelihoods are exactly analogous to the procedures we discussed above in what we now see was the isotropic Gaussian case.

In this framework we maximize the likelihood with respect to both the density parameters and the assignment variables simultaneously. This is appropriate if our goal is to group the data at hand, as is often the case. However, the project we laid out was to discover the nature of the process that generated the data. The process is characterized only by the density parameters, along with the probability distribution of the y_i. The particular choices of the y_i are not important, and indeed we wish to maximize not the likelihood (2.2), but its marginal taken over all the possible assignments $\mathcal{Y}$. This leads to the mixture model.

2.3 Mixture Models

The **mixture model** is perhaps the simplest example of a latent variable statistical model. It consists of a single observed vector variable and one discrete scalar latent variable. Both observations and latent variables are iid. This model is represented by the graph in figure 2.1a, using all the



$$\mathsf{P}_{\theta_x}\left(x_i \mid y_i\right)\mathsf{P}_{\theta_y}\left(y_i\right)$$

$$\prod_m \left(\mathsf{P}_{\theta_m}\left(x_i\right)\mathsf{P}_{\pi_m}\left(z_{m,i}\right)\right)^{z_{m,i}}$$

Figure 2.1: A mixture model.

conventions introduced in section 1.4. The marginal density of the ith observation x_i is

$$\mathsf{P}_\theta\left(x_i\right) = \sum_{y_i} \mathsf{P}_\theta\left(y_i\right)\mathsf{P}_\theta\left(x_i \mid y_i\right) \tag{2.3}$$

where the sum is taken over all the possible values the latent variable might assume. The choice of discrete values available to y_i is arbitrary, although the number of such values is not. We will write M for the number of distinct values the latent variable can take, and will assume that these values lie in the range $1 \ldots M$. The distribution function of the y_i is unconstrained, and so is parameterized by the probabilities associated with each value (strictly, by the probabilities of the first $M-1$ values). We will write π_m for $\mathsf{P}_\theta\left(y_i = m\right)$ and $\mathsf{P}_{\theta_m}\left(x_i\right)$ or even just $\mathsf{P}_m\left(x_i\right)$ for $\mathsf{P}_\theta\left(x_i \mid y_i = m\right)$. We can then rewrite the marginal density thus,

$$\mathsf{P}_\theta\left(x_i\right) = \sum_{m=1}^{M} \pi_m \mathsf{P}_{\theta_m}\left(x_i\right) \tag{2.4}$$

where the parameter set $\theta = \{\pi_1 \ldots \pi_M, \theta_1 \ldots \theta_M\}$.

Why the name "mixture model"? The latent variable can be viewed as a gate that, for each observation, selects one of the densities $\mathsf{P}_m\left(\cdot\right)$, from which the x_i is then drawn. Thus, the resultant set of observations is formed by mixing together sets of data drawn from each of the **component** densities $\mathsf{P}_m\left(\cdot\right)$. The relative sizes of these sets are defined by the **mixing parameters** π_m.

2.4 EM for Mixtures

The EM algorithm for mixture distributions has a particularly appealing form. The log-likelihood function for the parameters is

$$\ell_{\mathcal{X}}(\theta) = \sum_i \log \sum_{m=1}^{M} \pi_m \mathsf{P}_{\theta_m}(x_i) \tag{2.5}$$

which has the log-of-sum structure common to latent variable models. The joint data log likelihood is

$$\ell_{\mathcal{X},\mathcal{Y}}(\theta) = \sum_i \log \pi_{y_i} \mathsf{P}_{\theta_{y_i}}(x_i) \tag{2.6}$$

Written in this way, it is hard to manipulate. For this reason we will first re-express the mixture density in a way more conducive to application of EM.

In place of the single M-valued latent variable y_i we introduce a set of M binary-valued indicator latent variables $z_{m,i}$. For any observation, the one of these corresponding to the value of y_i takes the value 1, while the others are all 0. This version of the model is drawn in figure 2.1b. The $z_{m,i}$ are all dependent on each other. A random variable $x_{m,i}$ is drawn from the mth component distribution and multiplied by the value of $z_{m,i}$. All of these products are summed to produce the final observation. The square nodes in the graph represent deterministic combinations of random variables.

Armed with the variables $z_{m,i}$ we can rewrite the joint data log-likelihood

$$\ell_{\mathcal{X},\mathcal{Z}}(\theta) = \sum_i \sum_m z_{m,i} \log \pi_m \mathsf{P}_{\theta_m}(x_i) \tag{2.7}$$

with only one term in the inner sum being non-zero. The fact that this expression is linear in the $z_{m,i}$ makes the E-step of the EM algorithm quite straightforward.

$$\begin{aligned}
Q^n(\theta) &= \mathcal{E}_{\mathcal{Z}|\mathcal{X},\theta^{n-1}}\left[\ell_{\mathcal{X},\mathcal{Y}}(\theta)\right] \\
&= \mathcal{E}_{\mathcal{Z}|\mathcal{X},\theta^{n-1}}\left[\sum_i \sum_m z_{m,i} \log \pi_m \mathsf{P}_{\theta_m}(x_i)\right] \\
&= \sum_i \sum_m \mathcal{E}_{z_{m,i}|x_i,\theta^{n-1}}\left[z_{m,i}\right] \log \pi_m \mathsf{P}_{\theta_m}(x_i) \\
&= \sum_i \sum_m r^n_{m,i} \log \pi_m \mathsf{P}_{\theta_m}(x_i) \qquad (2.8)
\end{aligned}$$

where we have written $r^n_{m,i}$ for $\mathcal{E}_{z_{m,i}|x_i,\theta^{n-1}}[z_{m,i}]$. The variable $z_{m,i}$ is binary, and so its expected value is just the probability that it assumes the value 1, which it does when the gating variable y_i

is equal to m. Thus,

$$\begin{aligned} r^n_{m,i} = \mathcal{E}_{z_{m,i}|x_i,\theta^{n-1}}\left[z^i_m\right] &= \mathsf{P}_{\theta^{n-1}}\left(y_i = m \mid x_i\right) \\ &= \frac{\mathsf{P}_{\theta^{n-1}}\left(x_i \mid y_i = m\right)\mathsf{P}_{\theta^{n-1}}\left(y_i = m\right)}{\mathsf{P}_{\theta^{n-1}}\left(x_i\right)} \\ &= \frac{\pi^{n-1}_m \mathsf{P}_{\theta^{n-1}_m}\left(x_i\right)}{\sum_l \pi^{n-1}_l \mathsf{P}_{\theta^{n-1}_l}\left(x_i\right)} \end{aligned} \tag{2.9}$$

In other words, the number $r^n_{m,i}$ is the posterior probability that the ith observation was generated from mth component, under the $(n-1)$th iteration of the parameters. It is called the **responsibility** of the mth component for the ith observation. In clustering terms it can be thought of as the degree to which observation x_i is associated with cluster m.

We can also say some general things about the M-step without knowing the form of the component densities. Rewriting (2.8), we have

$$Q^n(\theta) = \sum_m \log \pi_m \sum_i r^n_{m,i} + \sum_m \sum_i r^n_{m,i} \log \mathsf{P}_{\theta_m}\left(x_i\right) \tag{2.10}$$

and so the maximization with respect to π_m and θ_m can proceed separately. We can find the new values of the π_m directly. We impose the constraint $\sum \pi_m = 1$ using a Lagrange multiplier λ and differentiate to obtain

$$\left.\frac{\partial}{\partial \pi_m}\right|_{\pi^n_m} \left(\sum_m \log \pi_m \sum_i r^n_{m,i} - \lambda \sum \pi_m\right) = \sum_i \frac{r^n_{m,i}}{\pi^n_m} - \lambda = 0 \tag{2.11}$$

and so π^n_m is proportional to $\sum_i r^n_{m,i}$. The normalization constraint then gives us

$$\pi^n_m = \frac{\sum_i r^n_{m,i}}{|\mathcal{X}|} \tag{2.12}$$

where the denominator is the number of observations and we have used the fact that $\sum_m r^n_{m,i} = 1$.

We cannot, of course, solve for the θ^n_m without knowing the forms of the component densities, but even here we can make a little headway. First, note that the θ_m (unlike the π_m) are independent of each other, and so we can maximize with respect to each component separately. Furthermore, the only term in (2.10) that depends on θ_m is $\sum_i r^n_{m,i} \log \mathsf{P}_{\theta_m}\left(x_i\right)$. Now, if we were to fit the mth component density alone to all of the observations, we would find the parameters by maximizing the log-likelihood $\sum_i \log \mathsf{P}_{\theta_m}\left(x_i\right)$. Thus, we can interpret the M-step as fitting each of the component distributions to all of the observations, weighting the contribution of the ith datum to the log-likelihood by the responsibility $r^n_{m,i}$.

Here, then, is the EM algorithm for mixture distributions:

E-step: Calculate the responsibilities at the nth iteration

$$r_{m,i}^{n} = \frac{\pi_m^{n-1} \mathsf{P}_{\theta_m^{n-1}}(x_i)}{\sum_l \pi_l^{n-1} \mathsf{P}_{\theta_l^{n-1}}(x_i)} \tag{2.13}$$

M-step: Estimate the new mixing parameters

$$\pi_m^n = \frac{\sum_i r_{m,i}^n}{|\mathcal{X}|} \tag{2.14}$$

and the new component distribution parameters

$$\theta_m^n = \underset{\theta_m}{\operatorname{argmax}} \sum_i r_{m,i}^n \log \mathsf{P}_{\theta_m}(x_i) \tag{2.15}$$

2.5 Applications of Mixture Models

We have introduced the mixture model from the point of view of clustering. The component densities are thus taken to represent different physical processes, the observed data being a mixture of points generated by these processes. The mixture-model likelihood and the EM algorithm used to optimize it, differ in focus from the clustering likelihood of (2.2) and the k-means algorithms: the mixture parameter estimates describe the generating process, while the sum-of-squares and related methods find the best grouping of the observed data. In general, if we consider many sets of data that generated by mixing the outputs of the same group of processes, we expect the mixture parameter estimates to exhibit much tighter variance than their clustering analogues. In situations where we expect to classify new data, or to make predictions, it is clear that the former approach is to be preferred.

The difference may also be viewed in another way. The likelihood of (2.2) dictates a "hard" clustering scheme — the solution involves an explicit assignment of observations into clusters. In contrast, fitting the mixture model describes a "soft" or "fuzzy" clustering scheme where observations are not, in fact, classified, but are partially associated with clusters through the responsibilities. We might intuitively expect these techniques to yield different answers. Fuzzy clustering schemes have been proposed, without the probabilistic interpretation, within the theory of fuzzy sets (Backer 1978; Bezdek 1981).

The clustering view of mixture modeling is only really meaningful in situations where the component densities are reasonably well separated. In such cases the likelihood landscape generally exhibits sharp maxima to which EM converges quickly.

Mixture models can also be employed in situations where the component densities overlap for the purposes of density estimation. The mixture density (2.4) can be quite complex, even when

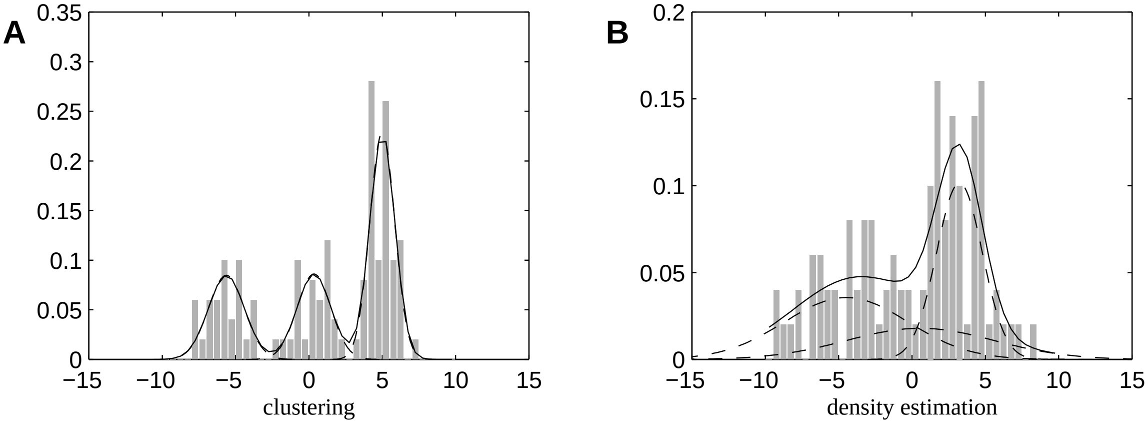


Figure 2.2: Two views of a mixture model.

the component distributions are relatively simple. As a result, complicated densities can be "non-parametrically" fit, with mixtures of Gaussians for instance, by the EM algorithm. From this viewpoint, there is no significance to the gating variable or to the component distributions – there is only one process with a complicated density and the mixture is just a convenient and flexible representation of the unknown density function. Indeed, one could view the familiar kernel-estimation technique as a particular case of a mixture model used in this way. The two views of the mixture model are illustrated in figure 2.2 where mixture models (the scaled components are shown by the dashed lines, the resulting mixture density by the solid lines) are fit to different types of one-dimensional data (histogrammed and shown by the grey bars).

We should make a short observation on our choice of the EM algorithm for learning the mixture model. If the component distributions overlap considerably it has been argued (Redner and Walker 1984) that the convergence of the EM algorithm to the optimal parameters of the mixture is slow (first order) and that superlinear methods should be preferred. However Redner and Walker (1984) themselves point out, and Xu and Jordan (1996) later elaborate, that the convergence of the *likelihood* of the mixture model is rapid, and that the mixture density approximates the true density quite quickly under EM. Thus, when the mixture model is used for clustering and thus the estimates of parameters are of importance, the components are likely to be reasonably well separated and therefore EM will converge well; while in the density estimation case, the criterion of importance is the convergence of the density estimate, and again this is rapid under EM.

2.6 Mixtures of Gaussians

A particularly fruitful mixture model, both in the context of clustering and of density estimation, arises when the components are (possibly multivariate) Gaussian densities. The parameters θ_m are then a mean vector μ_m and a covariance matrix Σ_m. The log-likelihood of the model is

$$\ell_{\mathcal{X}}(\theta) = \sum_i \log \sum_{m=1}^{M} \pi_m \left|2\pi\Sigma_m\right|^{-1/2} e^{-\frac{1}{2}(x-\mu_m)^T\Sigma_m^{-1}(x-\mu_m)} \tag{2.16}$$

The joint data log-likelihood with the indicator latent variables (2.7) is then

$$\ell_{\mathcal{X},\mathcal{Z}}(\theta) = \sum_i \sum_m z_{m,i} \left(\log \pi_m - \frac{1}{2} \log \left|2\pi\Sigma_m\right| - \frac{1}{2}(x_i - \mu_m)^T \Sigma_m^{-1} (x_i - \mu_m) \right) \tag{2.17}$$

where the exchange of the logarithm and the sum has eliminated the exponentials. The E-step is as for a generic mixture distribution (2.13), in this case given by

$$r^n_{m,i} \propto \pi^{n-1}_m \left|2\pi\Sigma^{n-1}_m\right|^{-1/2} e^{-\frac{1}{2}(x-\mu^{n-1}_m)^T(\Sigma^{n-1}_m)^{-1}(x-\mu^{n-1}_m)} \tag{2.18}$$

with the responsibilities normalized so as to sum to 1. In the M-step, the estimation of the mixing parameters is as for the generic mixture (2.14). The estimation of the mth component parameters is achieved by maximizing

$$Q^n_m(\theta) = -\sum_i r^n_{m,i} \left(\frac{1}{2} \log \left|2\pi\Sigma_m\right| + \frac{1}{2}(x_i - \mu_m)^T \Sigma_m^{-1} (x_i - \mu_m) \right) \tag{2.19}$$

Differentiating and equating to 0 we obtain

$$\begin{aligned} \left.\frac{\partial Q^n_m}{\partial \mu_m}\right|_{\mu^n_m} &= -\sum_i r^n_{m,i} (\Sigma^n_m)^{-1} (x_i - \mu^n_m) = 0 \\ \mu^n_m &= \frac{\sum_i r^n_{m,i} x_i}{\sum_i r^n_{m,i}} \end{aligned} \tag{2.20}$$

and (differentiating with respect to $R_m = \Sigma_m^{-1}$)

$$\begin{aligned} \left.\frac{\partial Q^n_m}{\partial R_m}\right|_{R^n_m} &= \sum_i r^n_{m,i} \left(\frac{1}{2}(R^n_m)^{-1} - \frac{1}{2}(x_i - \mu^n_m)(x_i - \mu^n_m)^T \right) = 0 \\ \Sigma^n_m &= \frac{\sum_i r^n_{m,i} (x_i - \mu^n_m)(x_i - \mu^n_m)^T}{\sum_i r^n_{m,i}} \end{aligned} \tag{2.21}$$

Thus the mean is updated to the responsibility-weighted mean of the observations, and the covariance to their responsibility-weighted covariance. This is a particularly elegant and fast update.

2.7 Practical Issues

We have argued that in situations where predictive power is desired, or where the parameters of the generating model are to be estimated as accurately as possible, the mixture model approach to clustering is to be preferred. Can we then blindly fit (with the EM algorithm) a basic mixture model to solve all clustering problems that confront us? Unfortunately, we will find that a number of practical issues need to be examined quite closely before we can achieve robust and repeatable parameter estimates.

We shall raise the issues one by one, discussing briefly some of the possible solutions to them as we proceed. The order is arbitrary, and some of the more basic and serious points are not discussed until last. In chapter 3 we will discuss in depth an elaboration of the EM algorithm which provides a new way to address a number of these issues.

2.7.1 Outliers

It is often the case that some of the data under consideration do not fall into any of the data clusters. These **outliers** may be caused by measurement errors, such as sensor artifacts or data mis-entry, or may be due to an additional data generating process which is diffuse and for which no model is available. The outliers may have a considerable effect on the estimates of the cluster parameters. For example, in a mixture of Gaussians clustering algorithm, the estimate of the mean for each Gaussian component is disproportionally sensitive to data from the tails of the distribution. The outliers fall far from all of the Gaussian clusters but nevertheless must be assigned to one or the other of them. As such, they will perturb the estimates of the means.

We can resolve this problem by introducing an additional generative component in the mixture which can take responsibility for the outliers[1]. This component density must be far more diffuse that the cluster densities, and must perturb the component density estimates as little as possible.

The most suitable choice for the outlier component probability is found in the uniform density. More precisely,

$$\mathsf{P}_O\left(x_i\right) = \begin{cases} \frac{1}{\|A\|} & \text{if} \quad x_i \in A \\ 0 & \text{if} \quad x_i \notin A \end{cases} \tag{2.22}$$

for some region A. This choice correctly embodies (in the Bayesian sense) our utter lack of knowledge of the distribution from which the outliers are drawn. Furthermore, it tends to minimize the pertubation in the cluster parameter estimates. We will make this assertion more precise in the particular case of Gaussian clusters.

Without loss of generality, we consider data drawn from a single Gaussian cluster, with mean μ and covariance Σ, corrupted by the addition of some outliers. We fit a model that has two components: one Gaussian and the other uniform. For simplicity in this analysis, assume that any outliers fall far from the center of the cluster and, as a result, have negligible responsibility assigned to the Gaussian. Under this assumption, the outliers themselves do not disturb the estimates of

[1] Banfield and Raftery (1993) take a similar approach in the context of hard clustering, introducing a Poisson distribution for outlier generation

the Gaussian parameters. However, the density of the uniform component within the region of the cluster is not negligible, and so responsibility for points that were, in fact, generated from the Gaussian is shared between the Gaussian and the uniform component. How will this sharing affect the estimates of the parameters of the Gaussian?

Consider the transform $\Sigma^{-1/2}$ applied to the data space. Both the Gaussian and the Uniform densities enjoy the property of mapping to another member of their respective families under a linear transformation, so that the nature of the mixture is unchanged. In this space, the data that belong to the cluster will be distributed according to a unit Gaussian (one with a covariance matrix equal to the identity). Without loss of generality, take the mean to be 0. We write $\tilde{\mu}$ and $\tilde{\Sigma}$ for the estimated mean and covariance, respectively, of the Gaussian component. Let the value of the uniform density in this space be $\tilde{u}$. The mixing probabilities are π_g and π_u for the Gaussian and uniform components respectively.

The following system of equations must hold at the maximum likelihood parameter values,

$$\begin{aligned} r_{g,i} &= 1 - \frac{\pi_u \tilde{u}}{\left(\pi_u \tilde{u} + \pi_g \left|2\pi\tilde{\Sigma}\right|^{-1} \exp\left(-\frac{1}{2}(\tilde{x}_i - \tilde{\mu})^T \tilde{\Sigma}^{-1}(\tilde{x}_i - \tilde{\mu})\right)\right)} \\ \tilde{\mu} &= \frac{\sum_i r_{g,i} x_i}{\sum_i r_{g,i}} \\ \tilde{\Sigma} &= \frac{\sum_i r_{g,i}(x_i - \tilde{\mu})(x_i - \tilde{\mu})^T}{\sum_i r_{g,i}} \end{aligned} \tag{2.23}$$

It is difficult to derive expressions for the estimates $\tilde{\mu}$ and $\tilde{\Sigma}$ directly, however we can make some arguments based on the symmetry of the situation. The data within the cluster are generated from a spherically symmetric distribution. Neglecting edge effects, the uniform density is also completely symmetric. Thus, on the average, there cannot be any directional bias to the estimates. This means that the expected value of $\tilde{\mu}$ must be 0, since any other value would break symmetry. Similarly, the expected value of $\tilde{\Sigma}$ must be isotropic, and will generally be slightly smaller than the true covariance in the transformed space I. These comments are about the *expected* values of the estimates, particular values of the estimates will be different based on the particular data instances being fit.

What do these results tell us about the estimated Gaussian in the original space? The linear transform $\Sigma^{1/2}$ maps from the whitened space to the original one. Since expectations are linear functions, the expected values of the parameter estimates are simply the transforms of the corresponding values in the whitened space. The estimated mean is thus distributed around the true value of the mean. The expected value of the covariance estimate is slightly smaller than the true covariance, but has the same shape in the sense of the same eigenvectors, and eigenvalue ratios.

It is important to note that this invariance came as a result of the uniform density being sub-

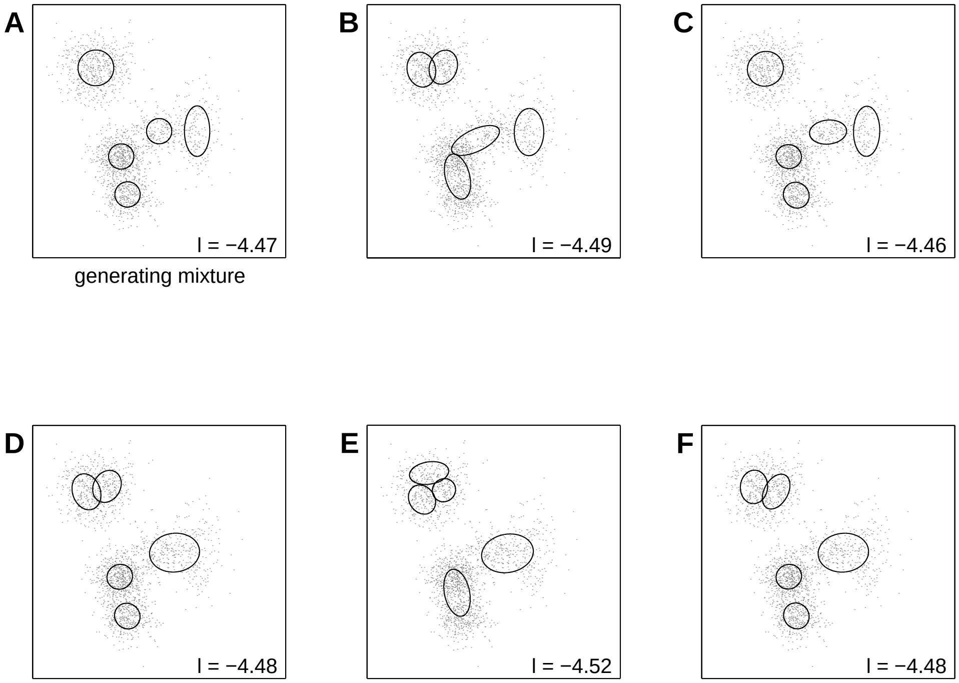


Figure 2.3: Multiple maxima in the mixture likelihood

stantially symmetric under any linear transform. Any other distribution would have had to have been carefully crafted to be symmetric. Furthermore, we would have to know a good deal about the cluster distribution to do so. With many, differently shaped, clusters only the uniform density will suffice.

2.7.2 Multiple maxima

The likelihood surface associated with a typical mixture model tends to exhibit multiple maxima. Trivially, given locally optimal parameters $\{\pi_m, \theta_m\}$, another maximum can be identified by retaining the same numerical values but permuting the component indices. In this case, the different maxima are equivalent in all practical senses and any one of them provides an equally good fit. Unfortunately, the system also exhibits non-trivial multiplicity.

Figure 2.3 illustrates the problem. Two-dimensional data are generated from the Gaussian mixture shown in A (each Gaussian in the mixture is represented by its 1-*sigma* contour). Panels B–F show the results of 5 separate fits to these data. The average log likelihood per point for each model (including the generating model) is recorded in the bottom right corner. Each model is the result of an EM optimization, and each optimization has converged. The difference between the results lies in the initial values of the parameters which are used to seed the EM process. (As an aside note that the best optimum (C) has a larger log-likelihood than the generating model — the

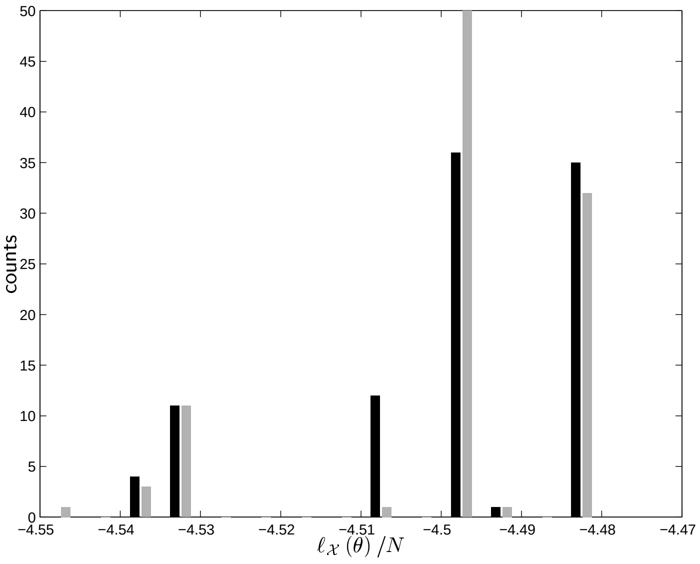


Figure 2.4: Likelihoods obtained from random restarts

data have permitted a small degree of over-fitting).

How are these initial values chosen? One generic approach, that does not depend on the type of component densities, is to randomly assign responsibilities for each data point and then derive the initial parameters using the M-step update rules. In large data sets, this approach tends to make the initial parameter values for each component virtually identical. This initial condition is similar to that of the REM algorithm to be discussed in chapter 3, however applying it in the standard EM context does not seem to be efficient. Convergence from such an initial point tends to be slow, and is no more reliable at finding a good maximum than the other techniques mentioned below.

An alternative approach, particularly useful in the case of mixtures of Gaussians (or the similar, well-localized, densities that are commonly used for clustering), is to pick a single covariance matrix (scale parameter) and initialize the means (location parameters) to randomly chosen data points. This is the method that was used to generate the fits in figure 2.3. We can refine the technique slightly by using these initial locations as the seed for a k-means clustering algorithm, and then using the output of that algorithm to provide the initial values of location parameters of the mixture model. K-means algorithms are also sensitive to the seed parameter values, but often less so than the full mixture, and so this initial stage tends to stabilize the estimates slightly. Nevertheless, experiments (an example appears in figure 2.4, to be described more completely below) suggest that in many situations the improvement is only very slight.

In general, optimization problems of this sort are known to be NP-hard, and so no entirely reliable, efficient solution can be found. Various approximate approaches are well-known in the optimization literature, and most may be adapted to the present problem. We will not dicuss most of these here, instead referring the reader to the books by Hertz *et al.* (1991), for general techniques and McLachlan and Krishnan (1996) for EM specific approaches. One general method, simulated annealing (Kirkpatrick *et al.* 1983), will be described briefly in chapter 3, although we will not elaborate on the application of this approach to mixture models. However, the principal subject of chapter 3, relaxation EM, is extremely pertinant to this issue and application to mixture models will be discussed in some detail.

For the moment, we note one quite straightforward approach, which is often remarkably effective. This is simply to choose a number of random starting conditions by one of the means described above, maximize the mixture likelihood starting from each of these initial values, and then choose the result that provides the largest likelihood. Figure 2.4 shows a histogram of the different values of the log-likelihood per point obtain by running 100 optimizations on the data of figure 2.3. The dark bars show the results when the EM algorithm started directly from randomly chosen parameter values; the lighter bars show the results obtained when a simple k-means algorithm was run first. On the basis of this experiment, we conclude that approximately one-third of the random selected conditions yield the best maximum (given either initialization). Thus, in only 10 restarts of the algorithm, the probability of finding the best optimum is 0.985. Of course, this probability will be dependent on the problem being examined: an appropriate number of restarts will need to be determined through simulation for each new type of problem.

2.7.3 The number of clusters

In general, when presented with a clustering problem we have no *a priori* information about how many different clusters we will encounter. This number, along with the optimal parameters to describe each cluster, must be estimated from the available data. This is a classic example of the general problem of model selection, which was addressed at some length in section 1.3. All of the analysis of that section applies to the present problem, and the methods described there are frequently employed.

In this section we will add another result to the battery of approximations to the marginal likelihood. This new approximation, introduced by (Cheeseman and Stutz 1996), is peculiar to mixture models and related latent variable models. In the following chapter, we shall introduce a novel framework, cascading model selection, for the efficient application of these various techniques.

The Cheeseman-Stutz criterion

The marginal likelihood for a mixture model with M components is given by

$$\mathsf{P}_M\left(\mathcal{X}\right) = \int d\theta \; \mathsf{P}_M\left(\theta\right) \prod_{i=1}^{N} \left(\sum_{m=1}^{M} \pi_m \mathsf{P}_{\theta_m}\left(x_i\right) \right) \tag{2.24}$$

Even if the individual cluster likelihood $\mathsf{P}_{\theta_m}(x_i)$ can be integrated with respect to θ_m, the overall integral proves to be intractable due to the M^N terms that appear once the product is distributed over the sum.

On the other, hand, if the latent variable values (expressed as the indicators $z_{m,i}$) were known, the marginal likelihood in this case could be written in a simpler form (compare the joint log-likelihood (2.7))

$$\begin{aligned}
\mathsf{P}_M(\mathcal{X}) &= \int d\theta\ \mathsf{P}_M(\theta) \prod_{i=1}^{N} \prod_{m=1}^{M} \left(\pi_m \mathsf{P}_{\theta_m}(x_i)\right)^{z_{i,m}} && (2.25)\\
&= \int d\theta\ \mathsf{P}_M(\theta) \prod_{m=1}^{M} \pi_m^{(\sum_i z_{i,m})} \prod_{i=1}^{N} \left(\mathsf{P}_{\theta_m}(x_i)\right)^{z_{i,m}} && (2.26)
\end{aligned}$$

This integral is more likely to be tractable. If the prior factors over the different cluster parameters θ_m the expression above reduces to the product of the marginal likelihoods of each cluster, given only the data assigned to that cluster.

Cheeseman and Stutz (1996) propose that we use this form, with the indicator values $z_{m,i}$ replaced by their expected values at the optimum, $r^*_{m,i}$, as the basis for an approximation of the true integral. In fact, direct substitution of the responsibilities into (2.26) will under-estimate the correct integral; however, the size of the error can be estimated from the mismatch between the value of the approximate integrand and the true likelihood at the estimated parameter values, θ^*. The complete approximation is

$$\mathsf{P}_M(\mathcal{X}) \approx \frac{\prod_{i=1}^{N} \left(\sum_{m=1}^{M} \pi^*_m \mathsf{P}_{\theta^*_m}(x_i)\right)}{\prod_{m=1}^{M} \pi_m^{R^*_m} \prod_{i=1}^{N} \left(\mathsf{P}_{\theta_m}(x_i)\right)^{r^*_{i,m}}} \int d\theta\ \mathsf{P}_M(\theta) \prod_{m=1}^{M} \pi_m^{R^*_m} \prod_{i=1}^{N} \left(\mathsf{P}_{\theta_m}(x_i)\right)^{r^*_{i,m}} \qquad (2.27)$$

where we have written $R^*_m = \sum_i r^*_{m,i}$.