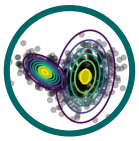
11

Density Estimation with Gaussian Mixture Models



In earlier chapters, we covered already two fundamental problems in machine learning: regression (Chapter 9) and dimensionality reduction (Chapter 10). In this chapter, we will have a look at a third pillar of machine learning: density estimation. On our journey, we introduce important concepts, such as the expectation maximization (EM) algorithm and a latent variable perspective of density estimation with mixture models.

When we apply machine learning to data we often aim to represent data in some way. A straightforward way is to take the data points themselves as the representation of the data; see Figure 11.1 for an example. However, this approach may be unhelpful if the dataset is huge or if we are interested in representing characteristics of the data. In density estimation, we represent the data compactly using a density from a parametric family, e.g., a Gaussian or Beta distribution. For example, we may be looking for the mean and variance of a dataset in order to represent the data compactly using a Gaussian distribution. The mean and variance can be found using tools we discussed in Section 8.3: maximum likelihood or maximum a posteriori estimation. We can then use the mean and variance of this Gaussian to represent the distribution underlying the data, i.e., we think of the dataset to be a typical realization from this distribution if we were to sample from it.

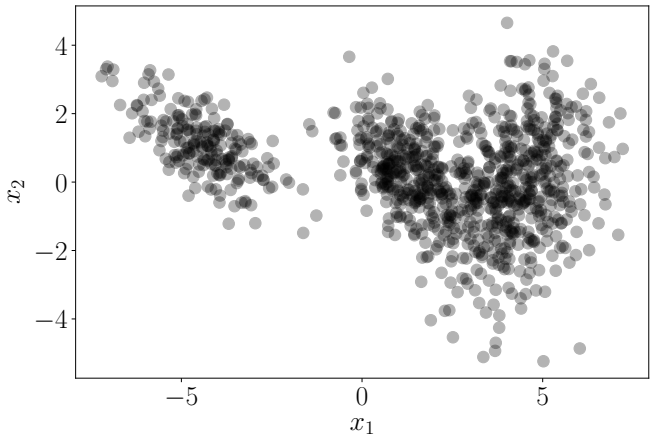


Figure 11.1 Two-dimensional dataset that cannot be meaningfully represented by a Gaussian.

In practice, the Gaussian (or similarly all other distributions we encountered so far) have limited modeling capabilities. For example, a Gaussian approximation of the density that generated the data in Figure 11.1 would be a poor approximation. In the following, we will look at a more expressive family of distributions, which we can use for density estimation: *mixture models*.

mixture model

Mixture models can be used to describe a distribution $p(\boldsymbol{x})$ by a convex combination of K simple (base) distributions

$$p(\boldsymbol{x}) = \sum_{k=1}^{K} \pi_k p_k(\boldsymbol{x}) \tag{11.1}$$

$$0 \leqslant \pi_k \leqslant 1\,, \quad \sum_{k=1}^{K} \pi_k = 1\,, \tag{11.2}$$

where the components p_k are members of a family of basic distributions, e.g., Gaussians, Bernoullis, or Gammas, and the π_k are *mixture weights*. Mixture models are more expressive than the corresponding base distributions because they allow for multimodal data representations, i.e., they can describe datasets with multiple "clusters", such as the example in Figure 11.1.

mixture weight

We will focus on Gaussian mixture models (GMMs), where the basic distributions are Gaussians. For a given dataset, we aim to maximize the likelihood of the model parameters to train the GMM. For this purpose, we will use results from Chapter 5, Chapter 6, and Section 7.2. However, unlike other applications we discussed earlier (linear regression or PCA), we will not find a closed-form maximum likelihood solution. Instead, we will arrive at a set of dependent simultaneous equations, which we can only solve iteratively.

11.1 Gaussian Mixture Model

A *Gaussian mixture model* is a density model where we combine a finite number of K Gaussian distributions $\mathcal{N}\big(\boldsymbol{x} \,|\, \boldsymbol{\mu}_k,\, \boldsymbol{\Sigma}_k\big)$ so that

Gaussian mixture model

$$p(\boldsymbol{x} \,|\, \boldsymbol{\theta}) = \sum_{k=1}^{K} \pi_k \mathcal{N}\big(\boldsymbol{x} \,|\, \boldsymbol{\mu}_k,\, \boldsymbol{\Sigma}_k\big) \tag{11.3}$$

$$0 \leqslant \pi_k \leqslant 1\,, \quad \sum_{k=1}^{K} \pi_k = 1\,, \tag{11.4}$$

where we defined $\boldsymbol{\theta} := \{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \pi_k : \; k = 1, \ldots, K\}$ as the collection of all parameters of the model. This convex combination of Gaussian distribution gives us significantly more flexibility for modeling complex densities than a simple Gaussian distribution (which we recover from (11.3) for $K = 1$). An illustration is given in Figure 11.2, displaying the weighted

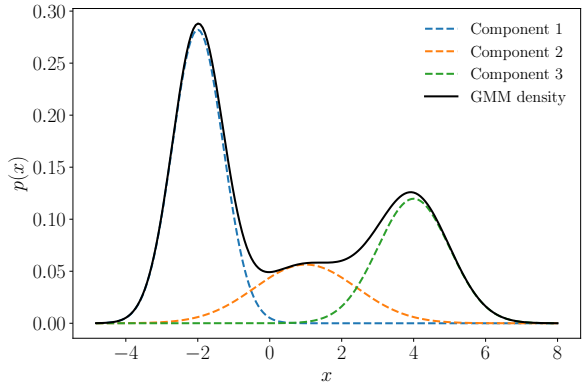


Figure 11.2 Gaussian mixture model. The Gaussian mixture distribution (black) is composed of a convex combination of Gaussian distributions and is more expressive than any individual component. Dashed lines represent the weighted Gaussian components.

components and the mixture density, which is given as

$$p(x \mid \boldsymbol{\theta}) = 0.5\mathcal{N}\left(x \mid -2, \tfrac{1}{2}\right) + 0.2\mathcal{N}\left(x \mid 1, 2\right) + 0.3\mathcal{N}\left(x \mid 4, 1\right). \quad (11.5)$$

11.2 Parameter Learning via Maximum Likelihood

Assume we are given a dataset $\mathcal{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$, where $\boldsymbol{x}_n$, $n = 1, \ldots, N$, are drawn i.i.d. from an unknown distribution $p(\boldsymbol{x})$. Our objective is to find a good approximation/representation of this unknown distribution $p(\boldsymbol{x})$ by means of a GMM with K mixture components. The parameters of the GMM are the K means $\boldsymbol{\mu}_k$, the covariances $\boldsymbol{\Sigma}_k$, and mixture weights π_k. We summarize all these free parameters in $\boldsymbol{\theta} := \{\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k : k = 1, \ldots, K\}$.

Example 11.1 (Initial Setting)

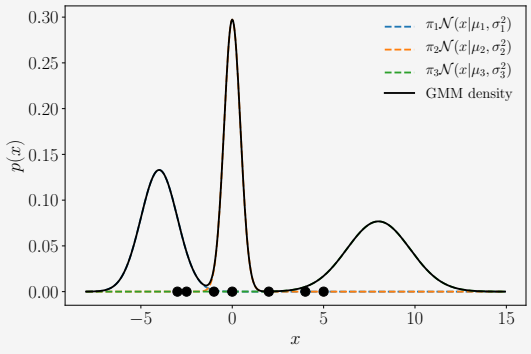


Figure 11.3 Initial setting: GMM (black) with mixture three mixture components (dashed) and seven data points (discs).

Throughout this chapter, we will have a simple running example that helps us illustrate and visualize important concepts.

We consider a one-dimensional dataset $\mathcal{X} = \{-3, -2.5, -1, 0, 2, 4, 5\}$ consisting of seven data points and wish to find a GMM with $K = 3$ components that models the density of the data. We initialize the mixture components as

$$p_1(x) = \mathcal{N}\big(x \,|\, -4,\, 1\big) \tag{11.6}$$

$$p_2(x) = \mathcal{N}\big(x \,|\, 0,\, 0.2\big) \tag{11.7}$$

$$p_3(x) = \mathcal{N}\big(x \,|\, 8,\, 3\big) \tag{11.8}$$

and assign them equal weights $\pi_1 = \pi_2 = \pi_3 = \frac{1}{3}$. The corresponding model (and the data points) are shown in Figure 11.3.

In the following, we detail how to obtain a maximum likelihood estimate $\boldsymbol{\theta}_{\text{ML}}$ of the model parameters $\boldsymbol{\theta}$. We start by writing down the likelihood, i.e., the predictive distribution of the training data given the parameters. We exploit our i.i.d. assumption, which leads to the factorized likelihood

$$p(\mathcal{X} \,|\, \boldsymbol{\theta}) = \prod_{n=1}^{N} p(\boldsymbol{x}_n \,|\, \boldsymbol{\theta}), \quad p(\boldsymbol{x}_n \,|\, \boldsymbol{\theta}) = \sum_{k=1}^{K} \pi_k \mathcal{N}\big(\boldsymbol{x}_n \,|\, \boldsymbol{\mu}_k,\, \boldsymbol{\Sigma}_k\big), \tag{11.9}$$

where every individual likelihood term $p(\boldsymbol{x}_n \,|\, \boldsymbol{\theta})$ is a Gaussian mixture density. Then we obtain the log-likelihood as

$$\log p(\mathcal{X} \,|\, \boldsymbol{\theta}) = \sum_{n=1}^{N} \log p(\boldsymbol{x}_n \,|\, \boldsymbol{\theta}) = \underbrace{\sum_{n=1}^{N} \log \sum_{k=1}^{K} \pi_k \mathcal{N}\big(\boldsymbol{x}_n \,|\, \boldsymbol{\mu}_k,\, \boldsymbol{\Sigma}_k\big)}_{=:\mathcal{L}}. \tag{11.10}$$

We aim to find parameters $\boldsymbol{\theta}^*_{\text{ML}}$ that maximize the log-likelihood $\mathcal{L}$ defined in (11.10). Our "normal" procedure would be to compute the gradient $\mathrm{d}\mathcal{L}/\mathrm{d}\boldsymbol{\theta}$ of the log-likelihood with respect to the model parameters $\boldsymbol{\theta}$, set it to $\mathbf{0}$, and solve for $\boldsymbol{\theta}$. However, unlike our previous examples for maximum likelihood estimation (e.g., when we discussed linear regression in Section 9.2), we cannot obtain a closed-form solution. However, we can exploit an iterative scheme to find good model parameters $\boldsymbol{\theta}_{\text{ML}}$, which will turn out to be the EM algorithm for GMMs. The key idea is to update one model parameter at a time while keeping the others fixed.

Remark. If we were to consider a single Gaussian as the desired density, the sum over k in (11.10) vanishes, and the $\log$ can be applied directly to the Gaussian component, such that we get

$$\log \mathcal{N}\big(\boldsymbol{x} \,|\, \boldsymbol{\mu},\, \boldsymbol{\Sigma}\big) = -\tfrac{D}{2}\log(2\pi) - \tfrac{1}{2}\log\det(\boldsymbol{\Sigma}) - \tfrac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}). \tag{11.11}$$

This simple form allows us to find closed-form maximum likelihood estimates of $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$, as discussed in Chapter 8. In (11.10), we cannot move

the log into the sum over k so that we cannot obtain a simple closed-form maximum likelihood solution. ♢

Any local optimum of a function exhibits the property that its gradient with respect to the parameters must vanish (necessary condition); see Chapter 7. In our case, we obtain the following necessary conditions when we optimize the log-likelihood in (11.10) with respect to the GMM parameters $\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \pi_k$:

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\mu}_k} = \mathbf{0}^\top \iff \sum_{n=1}^{N} \frac{\partial \log p(\boldsymbol{x}_n \mid \boldsymbol{\theta})}{\partial \boldsymbol{\mu}_k} = \mathbf{0}^\top, \tag{11.12}$$

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\Sigma}_k} = \mathbf{0} \iff \sum_{n=1}^{N} \frac{\partial \log p(\boldsymbol{x}_n \mid \boldsymbol{\theta})}{\partial \boldsymbol{\Sigma}_k} = \mathbf{0}, \tag{11.13}$$

$$\frac{\partial \mathcal{L}}{\partial \pi_k} = 0 \iff \sum_{n=1}^{N} \frac{\partial \log p(\boldsymbol{x}_n \mid \boldsymbol{\theta})}{\partial \pi_k} = 0. \tag{11.14}$$

For all three necessary conditions, by applying the chain rule (see Section 5.2.2), we require partial derivatives of the form

$$\frac{\partial \log p(\boldsymbol{x}_n \mid \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \frac{1}{p(\boldsymbol{x}_n \mid \boldsymbol{\theta})} \frac{\partial p(\boldsymbol{x}_n \mid \boldsymbol{\theta})}{\partial \boldsymbol{\theta}}, \tag{11.15}$$

where $\boldsymbol{\theta} = \{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \pi_k, k = 1, \dots, K\}$ are the model parameters and

$$\frac{1}{p(\boldsymbol{x}_n \mid \boldsymbol{\theta})} = \frac{1}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\boldsymbol{x}_n \mid \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}. \tag{11.16}$$

In the following, we will compute the partial derivatives (11.12) through (11.14). But before we do this, we introduce a quantity that will play a central role in the remainder of this chapter: responsibilities.

11.2.1 Responsibilities

We define the quantity

$$r_{nk} := \frac{\pi_k \mathcal{N}(\boldsymbol{x}_n \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\boldsymbol{x}_n \mid \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)} \tag{11.17}$$

responsibility

as the *responsibility* of the kth mixture component for the nth data point. The responsibility r_{nk} of the kth mixture component for data point $\boldsymbol{x}_n$ is proportional to the likelihood

$$p(\boldsymbol{x}_n \mid \pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \pi_k \mathcal{N}(\boldsymbol{x}_n \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \tag{11.18}$$

$\boldsymbol{r}_n$ follows a Boltzmann/Gibbs distribution.

of the mixture component given the data point. Therefore, mixture components have a high responsibility for a data point when the data point could be a plausible sample from that mixture component. Note that $\boldsymbol{r}_n := [r_{n1}, \dots, r_{nK}]^\top \in \mathbb{R}^K$ is a (normalized) probability vector, i.e.,

$\sum_k r_{nk} = 1$ with $r_{nk} \geqslant 0$. This probability vector distributes probability mass among the K mixture components, and we can think of $\boldsymbol{r}_n$ as a "soft assignment" of $\boldsymbol{x}_n$ to the K mixture components. Therefore, the responsibility r_{nk} from (11.17) represents the probability that $\boldsymbol{x}_n$ has been generated by the kth mixture component.

The responsibility r_{nk} is the probability that the kth mixture component generated the nth data point.

Example 11.2 (Responsibilities)
For our example from Figure 11.3, we compute the responsibilities r_{nk}

$$\begin{bmatrix} 1.0 & 0.0 & 0.0 \\ 1.0 & 0.0 & 0.0 \\ 0.057 & 0.943 & 0.0 \\ 0.001 & 0.999 & 0.0 \\ 0.0 & 0.066 & 0.934 \\ 0.0 & 0.0 & 1.0 \\ 0.0 & 0.0 & 1.0 \end{bmatrix} \in \mathbb{R}^{N \times K}. \tag{11.19}$$

Here the nth row tells us the responsibilities of all mixture components for x_n. The sum of all K responsibilities for a data point (sum of every row) is 1. The kth column gives us an overview of the responsibility of the kth mixture component. We can see that the third mixture component (third column) is not responsible for any of the first four data points, but takes much responsibility of the remaining data points. The sum of all entries of a column gives us the values N_k, i.e., the total responsibility of the kth mixture component. In our example, we get $N_1 = 2.058,\ N_2 = 2.008,\ N_3 = 2.934$.

In the following, we determine the updates of the model parameters $\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \pi_k$ for given responsibilities. We will see that the update equations all depend on the responsibilities, which makes a closed-form solution to the maximum likelihood estimation problem impossible. However, for given responsibilities we will be updating one model parameter at a time, while keeping the others fixed. After this, we will recompute the responsibilities. Iterating these two steps will eventually converge to a local optimum and is a specific instantiation of the EM algorithm. We will discuss this in some more detail in Section 11.3.

11.2.2 Updating the Means

Theorem 11.1 (Update of the GMM Means)**.** *The update of the mean parameters $\boldsymbol{\mu}_k$, $k = 1, \ldots, K$, of the GMM is given by*

$$\boldsymbol{\mu}_k^{new} = \frac{\sum_{n=1}^N r_{nk} \boldsymbol{x}_n}{\sum_{n=1}^N r_{nk}}, \tag{11.20}$$

where the responsibilities r_{nk} are defined in (11.17).

Remark. The update of the means $\boldsymbol{\mu}_k$ of the individual mixture components in (11.20) depends on all means, covariance matrices $\boldsymbol{\Sigma}_k$, and mixture weights π_k via r_{nk} given in (11.17). Therefore, we cannot obtain a closed-form solution for all $\boldsymbol{\mu}_k$ at once. ♢

Proof From (11.15), we see that the gradient of the log-likelihood with respect to the mean parameters $\boldsymbol{\mu}_k$, $k = 1, \ldots, K$, requires us to compute the partial derivative

$$\frac{\partial p(\boldsymbol{x}_n \mid \boldsymbol{\theta})}{\partial \boldsymbol{\mu}_k} = \sum_{j=1}^{K} \pi_j \frac{\partial \mathcal{N}(\boldsymbol{x}_n \mid \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}{\partial \boldsymbol{\mu}_k} = \pi_k \frac{\partial \mathcal{N}(\boldsymbol{x}_n \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\partial \boldsymbol{\mu}_k} \tag{11.21a}$$

$$= \pi_k (\boldsymbol{x}_n - \boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1} \mathcal{N}(\boldsymbol{x}_n \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \,, \tag{11.21b}$$

where we exploited that only the kth mixture component depends on $\boldsymbol{\mu}_k$.

We use our result from (11.21b) in (11.15) and put everything together so that the desired partial derivative of $\mathcal{L}$ with respect to $\boldsymbol{\mu}_k$ is given as

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\mu}_k} = \sum_{n=1}^{N} \frac{\partial \log p(\boldsymbol{x}_n \mid \boldsymbol{\theta})}{\partial \boldsymbol{\mu}_k} = \sum_{n=1}^{N} \frac{1}{p(\boldsymbol{x}_n \mid \boldsymbol{\theta})} \frac{\partial p(\boldsymbol{x}_n \mid \boldsymbol{\theta})}{\partial \boldsymbol{\mu}_k} \tag{11.22a}$$

$$= \sum_{n=1}^{N} (\boldsymbol{x}_n - \boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1} \underbrace{\boxed{\frac{\pi_k \mathcal{N}(\boldsymbol{x}_n \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\boldsymbol{x}_n \mid \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}}}_{=r_{nk}} \tag{11.22b}$$

$$= \sum_{n=1}^{N} r_{nk} (\boldsymbol{x}_n - \boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1} \,. \tag{11.22c}$$

Here we used the identity from (11.16) and the result of the partial derivative in (11.21b) to get to (11.22b). The values r_{nk} are the responsibilities we defined in (11.17).

We now solve (11.22c) for $\boldsymbol{\mu}_k^{\text{new}}$ so that $\frac{\partial \mathcal{L}(\boldsymbol{\mu}_k^{\text{new}})}{\partial \boldsymbol{\mu}_k} = \mathbf{0}^\top$ and obtain

$$\sum_{n=1}^{N} r_{nk} \boldsymbol{x}_n = \sum_{n=1}^{N} r_{nk} \boldsymbol{\mu}_k^{\text{new}} \iff \boldsymbol{\mu}_k^{\text{new}} = \frac{\sum_{n=1}^{N} r_{nk} \boldsymbol{x}_n}{\boxed{\sum_{n=1}^{N} r_{nk}}} = \frac{1}{\boxed{N_k}} \sum_{n=1}^{N} r_{nk} \boldsymbol{x}_n \,, \tag{11.23}$$

where we defined

$$N_k := \sum_{n=1}^{N} r_{nk} \tag{11.24}$$

as the total responsibility of the kth mixture component for the entire dataset. This concludes the proof of Theorem 11.1. □

Intuitively, (11.20) can be interpreted as an importance-weighted Monte Carlo estimate of the mean, where the importance weights of data point $\boldsymbol{x}_n$ are the responsibilities r_{nk} of the kth cluster for $\boldsymbol{x}_n$, $k = 1, \ldots, K$.

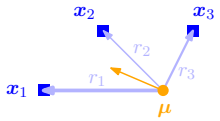
Therefore, the mean $\boldsymbol{\mu}_k$ is pulled toward a data point $\boldsymbol{x}_n$ with strength given by r_{nk}. The means are pulled stronger toward data points for which the corresponding mixture component has a high responsibility, i.e., a high likelihood. Figure 11.4 illustrates this. We can also interpret the mean update in (11.20) as the expected value of all data points under the distribution given by

$$\boldsymbol{r}_k := [r_{1k}, \ldots, r_{Nk}]^\top / N_k \,, \tag{11.25}$$

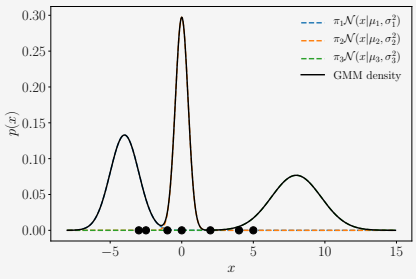
which is a normalized probability vector, i.e.,

$$\boldsymbol{\mu}_k \leftarrow \mathbb{E}_{\boldsymbol{r}_k}[\mathcal{X}] \,. \tag{11.26}$$

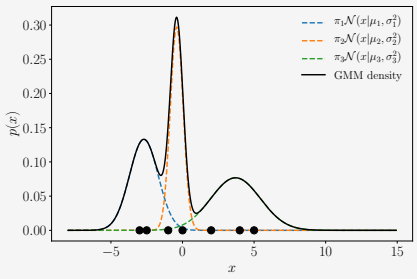
Figure 11.4 Update of the mean parameter of mixture component in a GMM. The mean $\boldsymbol{\mu}$ is being pulled toward individual data points with the weights given by the corresponding responsibilities.



Example 11.3 (Mean Updates)



(a) GMM density and individual components prior to updating the mean values.



(b) GMM density and individual components after updating the mean values.

Figure 11.5 Effect of updating the mean values in a GMM. (a) GMM before updating the mean values; (b) GMM after updating the mean values μ_k while retaining the variances and mixture weights.

In our example from Figure 11.3, the mean values are updated as follows:

$$\mu_1 : -4 \to -2.7 \tag{11.27}$$
$$\mu_2 : 0 \to -0.4 \tag{11.28}$$
$$\mu_3 : 8 \to 3.7 \tag{11.29}$$

Here we see that the means of the first and third mixture component move toward the regime of the data, whereas the mean of the second component does not change so dramatically. Figure 11.5 illustrates this change, where Figure 11.5(a) shows the GMM density prior to updating the means and Figure 11.5(b) shows the GMM density after updating the mean values μ_k.

The update of the mean parameters in (11.20) look fairly straightforward. However, note that the responsibilities r_{nk} are a function of $\pi_j, \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j$ for all $j = 1, \ldots, K$, such that the updates in (11.20) depend on all parameters of the GMM, and a closed-form solution, which we obtained for linear regression in Section 9.2 or PCA in Chapter 10, cannot be obtained.

11.2.3 Updating the Covariances

Theorem 11.2 (Updates of the GMM Covariances)**.** *The update of the covariance parameters $\boldsymbol{\Sigma}_k$, $k = 1, \ldots, K$ of the GMM is given by*

$$\boldsymbol{\Sigma}_k^{new} = \frac{1}{N_k} \sum_{n=1}^{N} r_{nk} (\boldsymbol{x}_n - \boldsymbol{\mu}_k)(\boldsymbol{x}_n - \boldsymbol{\mu}_k)^\top , \tag{11.30}$$

where r_{nk} and N_k are defined in (11.17) *and* (11.24)*, respectively.*

Proof To prove Theorem 11.2, our approach is to compute the partial derivatives of the log-likelihood $\mathcal{L}$ with respect to the covariances $\boldsymbol{\Sigma}_k$, set them to $\mathbf{0}$, and solve for $\boldsymbol{\Sigma}_k$. We start with our general approach

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\Sigma}_k} = \sum_{n=1}^{N} \frac{\partial \log p(\boldsymbol{x}_n \,|\, \boldsymbol{\theta})}{\partial \boldsymbol{\Sigma}_k} = \sum_{n=1}^{N} \frac{1}{p(\boldsymbol{x}_n \,|\, \boldsymbol{\theta})} \frac{\partial p(\boldsymbol{x}_n \,|\, \boldsymbol{\theta})}{\partial \boldsymbol{\Sigma}_k} . \tag{11.31}$$

We already know $1/p(\boldsymbol{x}_n \,|\, \boldsymbol{\theta})$ from (11.16). To obtain the remaining partial derivative $\partial p(\boldsymbol{x}_n \,|\, \boldsymbol{\theta})/\partial \boldsymbol{\Sigma}_k$, we write down the definition of the Gaussian distribution $p(\boldsymbol{x}_n \,|\, \boldsymbol{\theta})$ (see (11.9)) and drop all terms but the kth. We then obtain

$$\frac{\partial p(\boldsymbol{x}_n \,|\, \boldsymbol{\theta})}{\partial \boldsymbol{\Sigma}_k} \tag{11.32a}$$

$$= \frac{\partial}{\partial \boldsymbol{\Sigma}_k} \left(\pi_k (2\pi)^{-\frac{D}{2}} \det(\boldsymbol{\Sigma}_k)^{-\frac{1}{2}} \exp\left(-\tfrac{1}{2} (\boldsymbol{x}_n - \boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1} (\boldsymbol{x}_n - \boldsymbol{\mu}_k) \right) \right) \tag{11.32b}$$

$$= \pi_k (2\pi)^{-\frac{D}{2}} \left[\frac{\partial}{\partial \boldsymbol{\Sigma}_k} \det(\boldsymbol{\Sigma}_k)^{-\frac{1}{2}} \exp\left(-\tfrac{1}{2} (\boldsymbol{x}_n - \boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1} (\boldsymbol{x}_n - \boldsymbol{\mu}_k) \right) \right.$$
$$\left. + \det(\boldsymbol{\Sigma}_k)^{-\frac{1}{2}} \frac{\partial}{\partial \boldsymbol{\Sigma}_k} \exp\left(-\tfrac{1}{2} (\boldsymbol{x}_n - \boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1} (\boldsymbol{x}_n - \boldsymbol{\mu}_k) \right) \right] . \tag{11.32c}$$

We now use the identities

$$\frac{\partial}{\partial \boldsymbol{\Sigma}_k} \det(\boldsymbol{\Sigma}_k)^{-\frac{1}{2}} \overset{(5.101)}{=} -\frac{1}{2} \det(\boldsymbol{\Sigma}_k)^{-\frac{1}{2}} \boldsymbol{\Sigma}_k^{-1} , \tag{11.33}$$

$$\frac{\partial}{\partial \boldsymbol{\Sigma}_k} (\boldsymbol{x}_n - \boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1} (\boldsymbol{x}_n - \boldsymbol{\mu}_k) \overset{(5.103)}{=} -\boldsymbol{\Sigma}_k^{-1} (\boldsymbol{x}_n - \boldsymbol{\mu}_k)(\boldsymbol{x}_n - \boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1} \tag{11.34}$$

and obtain (after some rearranging) the desired partial derivative required in (11.31) as

$$\frac{\partial p(\boldsymbol{x}_n \,|\, \boldsymbol{\theta})}{\partial \boldsymbol{\Sigma}_k} = \pi_k \mathcal{N}\left(\boldsymbol{x}_n \,|\, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right)$$
$$\cdot \left[-\tfrac{1}{2} (\boldsymbol{\Sigma}_k^{-1} - \boldsymbol{\Sigma}_k^{-1} (\boldsymbol{x}_n - \boldsymbol{\mu}_k)(\boldsymbol{x}_n - \boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1}) \right] . \tag{11.35}$$

Putting everything together, the partial derivative of the log-likelihood

with respect to $\boldsymbol{\Sigma}_k$ is given by

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\Sigma}_k} = \sum_{n=1}^{N} \frac{\partial \log p(\boldsymbol{x}_n \mid \boldsymbol{\theta})}{\partial \boldsymbol{\Sigma}_k} = \sum_{n=1}^{N} \frac{1}{p(\boldsymbol{x}_n \mid \boldsymbol{\theta})} \frac{\partial p(\boldsymbol{x}_n \mid \boldsymbol{\theta})}{\partial \boldsymbol{\Sigma}_k} \tag{11.36a}$$

$$= \sum_{n=1}^{N} \underbrace{\frac{\pi_k \mathcal{N}(\boldsymbol{x}_n \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\boldsymbol{x}_n \mid \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}}_{=r_{nk}}$$

$$\cdot \left[-\tfrac{1}{2}(\boldsymbol{\Sigma}_k^{-1} - \boldsymbol{\Sigma}_k^{-1}(\boldsymbol{x}_n - \boldsymbol{\mu}_k)(\boldsymbol{x}_n - \boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1})\right] \tag{11.36b}$$

$$= -\frac{1}{2} \sum_{n=1}^{N} r_{nk} (\boldsymbol{\Sigma}_k^{-1} - \boldsymbol{\Sigma}_k^{-1}(\boldsymbol{x}_n - \boldsymbol{\mu}_k)(\boldsymbol{x}_n - \boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1}) \tag{11.36c}$$

$$= -\frac{1}{2} \boldsymbol{\Sigma}_k^{-1} \underbrace{\sum_{n=1}^{N} r_{nk}}_{=N_k} + \frac{1}{2} \boldsymbol{\Sigma}_k^{-1} \left(\sum_{n=1}^{N} r_{nk} (\boldsymbol{x}_n - \boldsymbol{\mu}_k)(\boldsymbol{x}_n - \boldsymbol{\mu}_k)^\top \right) \boldsymbol{\Sigma}_k^{-1}. \tag{11.36d}$$

We see that the responsibilities r_{nk} also appear in this partial derivative. Setting this partial derivative to $\mathbf{0}$, we obtain the necessary optimality condition

$$N_k \boldsymbol{\Sigma}_k^{-1} = \boldsymbol{\Sigma}_k^{-1} \left(\sum_{n=1}^{N} r_{nk} (\boldsymbol{x}_n - \boldsymbol{\mu}_k)(\boldsymbol{x}_n - \boldsymbol{\mu}_k)^\top \right) \boldsymbol{\Sigma}_k^{-1} \tag{11.37a}$$

$$\iff N_k \boldsymbol{I} = \left(\sum_{n=1}^{N} r_{nk} (\boldsymbol{x}_n - \boldsymbol{\mu}_k)(\boldsymbol{x}_n - \boldsymbol{\mu}_k)^\top \right) \boldsymbol{\Sigma}_k^{-1}. \tag{11.37b}$$

By solving for $\boldsymbol{\Sigma}_k$, we obtain

$$\boldsymbol{\Sigma}_k^{\text{new}} = \frac{1}{N_k} \sum_{n=1}^{N} r_{nk} (\boldsymbol{x}_n - \boldsymbol{\mu}_k)(\boldsymbol{x}_n - \boldsymbol{\mu}_k)^\top, \tag{11.38}$$

where $\boldsymbol{r}_k$ is the probability vector defined in (11.25). This gives us a simple update rule for $\boldsymbol{\Sigma}_k$ for $k = 1, \ldots, K$ and proves Theorem 11.2. □

Similar to the update of $\boldsymbol{\mu}_k$ in (11.20), we can interpret the update of the covariance in (11.30) as an importance-weighted expected value of the square of the centered data $\tilde{\mathcal{X}}_k := \{\boldsymbol{x}_1 - \boldsymbol{\mu}_k, \ldots, \boldsymbol{x}_N - \boldsymbol{\mu}_k\}$.

Example 11.4 (Variance Updates)
In our example from Figure 11.3, the variances are updated as follows:

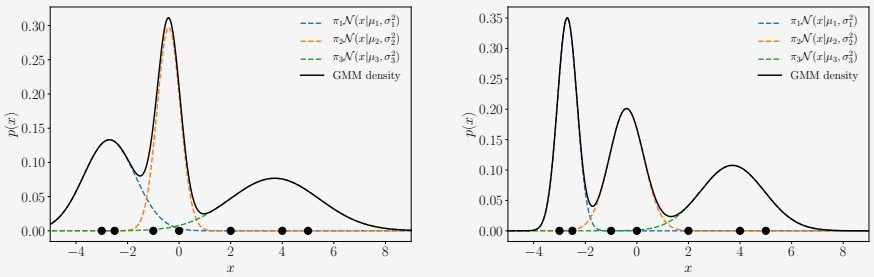
$$\sigma_1^2 : 1 \to 0.14 \tag{11.39}$$

$$\sigma_2^2 : 0.2 \to 0.44 \tag{11.40}$$

$$\sigma_3^2 : 3 \to 1.53 \tag{11.41}$$

Here we see that the variances of the first and third component shrink significantly, whereas the variance of the second component increases slightly.

Figure 11.6 illustrates this setting. Figure 11.6(a) is identical (but zoomed in) to Figure 11.5(b) and shows the GMM density and its individual components prior to updating the variances. Figure 11.6(b) shows the GMM density after updating the variances.



(a) GMM density and individual components prior to updating the variances.

(b) GMM density and individual components after updating the variances.

Figure 11.6 Effect of updating the variances in a GMM. (a) GMM before updating the variances; (b) GMM after updating the variances while retaining the means and mixture weights.

Similar to the update of the mean parameters, we can interpret (11.30) as a Monte Carlo estimate of the weighted covariance of data points $\boldsymbol{x}_n$ associated with the kth mixture component, where the weights are the responsibilities r_{nk}. As with the updates of the mean parameters, this update depends on all $\pi_j, \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j,\ j = 1, \ldots, K$, through the responsibilities r_{nk}, which prohibits a closed-form solution.

11.2.4 Updating the Mixture Weights

Theorem 11.3 (Update of the GMM Mixture Weights)**.** *The mixture weights of the GMM are updated as*

$$\pi_k^{new} = \frac{N_k}{N}, \quad k = 1, \ldots, K, \tag{11.42}$$

where N is the number of data points and N_k is defined in (11.24)*.*

Proof To find the partial derivative of the log-likelihood with respect to the weight parameters π_k, $k = 1, \ldots, K$, we account for the constraint $\sum_k \pi_k = 1$ by using Lagrange multipliers (see Section 7.2). The Lagrangian is

$$\mathfrak{L} = \mathcal{L} + \lambda \left(\sum_{k=1}^{K} \pi_k - 1 \right) \tag{11.43a}$$

$$= \sum_{n=1}^{N} \log \sum_{k=1}^{K} \pi_k \mathcal{N}\big(\boldsymbol{x}_n \,|\, \boldsymbol{\mu}_k,\, \boldsymbol{\Sigma}_k\big) + \lambda \left(\sum_{k=1}^{K} \pi_k - 1 \right) , \tag{11.43b}$$

where $\mathcal{L}$ is the log-likelihood from (11.10) and the second term encodes for the equality constraint that all the mixture weights need to sum up to 1. We obtain the partial derivative with respect to π_k as

$$\frac{\partial \mathfrak{L}}{\partial \pi_k} = \sum_{n=1}^{N} \frac{\mathcal{N}\big(\boldsymbol{x}_n \,|\, \boldsymbol{\mu}_k,\, \boldsymbol{\Sigma}_k\big)}{\sum_{j=1}^{K} \pi_j \mathcal{N}\big(\boldsymbol{x}_n \,|\, \boldsymbol{\mu}_j,\, \boldsymbol{\Sigma}_j\big)} + \lambda \tag{11.44a}$$

$$= \frac{1}{\pi_k} \underbrace{\sum_{n=1}^{N} \frac{\pi_k \mathcal{N}\big(\boldsymbol{x}_n \,|\, \boldsymbol{\mu}_k,\, \boldsymbol{\Sigma}_k\big)}{\sum_{j=1}^{K} \pi_j \mathcal{N}\big(\boldsymbol{x}_n \,|\, \boldsymbol{\mu}_j,\, \boldsymbol{\Sigma}_j\big)}}_{=N_k} + \lambda = \frac{N_k}{\pi_k} + \lambda \,, \tag{11.44b}$$

and the partial derivative with respect to the Lagrange multiplier λ as

$$\frac{\partial \mathfrak{L}}{\partial \lambda} = \sum_{k=1}^{K} \pi_k - 1 \,. \tag{11.45}$$

Setting both partial derivatives to $\mathbf{0}$ (necessary condition for optimum) yields the system of equations

$$\pi_k = -\frac{N_k}{\lambda} \,, \tag{11.46}$$

$$1 = \sum_{k=1}^{K} \pi_k \,. \tag{11.47}$$

Using (11.46) in (11.47) and solving for π_k, we obtain

$$\sum_{k=1}^{K} \pi_k = 1 \iff -\sum_{k=1}^{K} \frac{N_k}{\lambda} = 1 \iff -\frac{N}{\lambda} = 1 \iff \lambda = -N \,. \tag{11.48}$$

This allows us to substitute $-N$ for λ in (11.46) to obtain

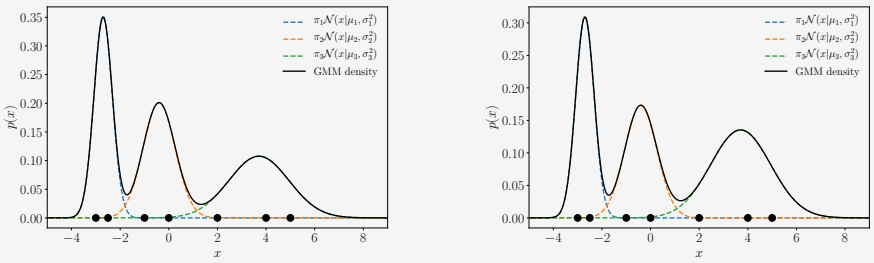
$$\pi_k^{\text{new}} = \frac{N_k}{N} \,, \tag{11.49}$$

which gives us the update for the weight parameters π_k and proves Theorem 11.3. □

We can identify the mixture weight in (11.42) as the ratio of the total responsibility of the kth cluster and the number of data points. Since $N = \sum_k N_k$, the number of data points can also be interpreted as the total responsibility of all mixture components together, such that π_k is the relative importance of the kth mixture component for the dataset.

Remark. Since $N_k = \sum_{i=1}^{N} r_{nk}$, the update equation (11.42) for the mixture weights π_k also depends on all $\pi_j, \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j, j = 1, \ldots, K$ via the responsibilities r_{nk}. ♢

Example 11.5 (Weight Parameter Updates)



(a) GMM density and individual components prior to updating the mixture weights.

(b) GMM density and individual components after updating the mixture weights.

Figure 11.7 Effect of updating the mixture weights in a GMM. (a) GMM before updating the mixture weights; (b) GMM after updating the mixture weights while retaining the means and variances. Note the different scales of the vertical axes.

In our running example from Figure 11.3, the mixture weights are updated as follows:

$$\pi_1 : \frac{1}{3} \to 0.29 \tag{11.50}$$
$$\pi_2 : \frac{1}{3} \to 0.29 \tag{11.51}$$
$$\pi_3 : \frac{1}{3} \to 0.42 \tag{11.52}$$

Here we see that the third component gets more weight/importance, while the other components become slightly less important. Figure 11.7 illustrates the effect of updating the mixture weights. Figure 11.7(a) is identical to Figure 11.6(b) and shows the GMM density and its individual components prior to updating the mixture weights. Figure 11.7(b) shows the GMM density after updating the mixture weights.

Overall, having updated the means, the variances, and the weights once, we obtain the GMM shown in Figure 11.7(b). Compared with the initialization shown in Figure 11.3, we can see that the parameter updates caused the GMM density to shift some of its mass toward the data points.

After updating the means, variances, and weights once, the GMM fit in Figure 11.7(b) is already remarkably better than its initialization from Figure 11.3. This is also evidenced by the log-likelihood values, which increased from -28.3 (initialization) to -14.4 after a full update cycle.

11.3 EM Algorithm

Unfortunately, the updates in (11.20), (11.30), and (11.42) do not constitute a closed-form solution for the updates of the parameters $\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \pi_k$ of the mixture model because the responsibilities r_{nk} depend on those parameters in a complex way. However, the results suggest a simple *iterative scheme* for finding a solution to the parameters estimation problem via maximum likelihood. The expectation maximization algorithm (*EM algo-*

EM algorithm

rithm) was proposed by Dempster et al. (1977) and is a general iterative scheme for learning parameters (maximum likelihood or MAP) in mixture models and, more generally, latent-variable models.

In our example of the Gaussian mixture model, we choose initial values for $\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \pi_k$ and alternate until convergence between

- *E-step:* Evaluate the responsibilities r_{nk} (posterior probability of data point n belonging to mixture component k).
- *M-step:* Use the updated responsibilities to reestimate the parameters $\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \pi_k$.

Every step in the EM algorithm increases the log-likelihood function (Neal and Hinton, 1999). For convergence, we can check the log-likelihood or the parameters directly. A concrete instantiation of the EM algorithm for estimating the parameters of a GMM is as follows:

1. Initialize $\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \pi_k$.
2. *E-step:* Evaluate responsibilities r_{nk} for every data point $\boldsymbol{x}_n$ using current parameters $\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k$:

$$r_{nk} = \frac{\pi_k \mathcal{N}\left(\boldsymbol{x}_n \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right)}{\sum_j \pi_j \mathcal{N}\left(\boldsymbol{x}_n \mid \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j\right)} . \tag{11.53}$$

3. *M-step:* Reestimate parameters $\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k$ using the current responsibilities r_{nk} (from E-step):

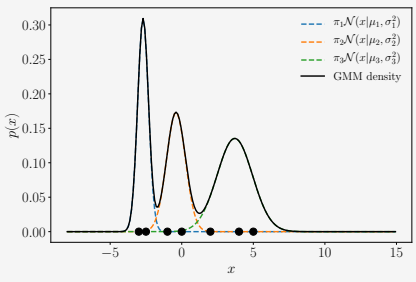
$$\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{n=1}^{N} r_{nk} \boldsymbol{x}_n , \tag{11.54}$$

$$\boldsymbol{\Sigma}_k = \frac{1}{N_k} \sum_{n=1}^{N} r_{nk} (\boldsymbol{x}_n - \boldsymbol{\mu}_k)(\boldsymbol{x}_n - \boldsymbol{\mu}_k)^\top , \tag{11.55}$$

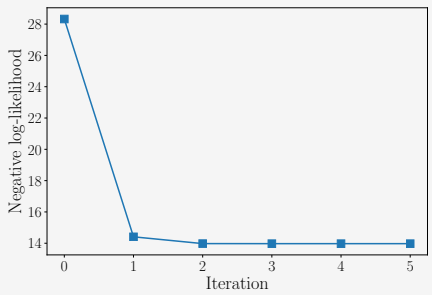
$$\pi_k = \frac{N_k}{N} . \tag{11.56}$$

Having updated the means $\boldsymbol{\mu}_k$ in (11.54), they are subsequently used in (11.55) to update the corresponding covariances.

Example 11.6 (GMM Fit)

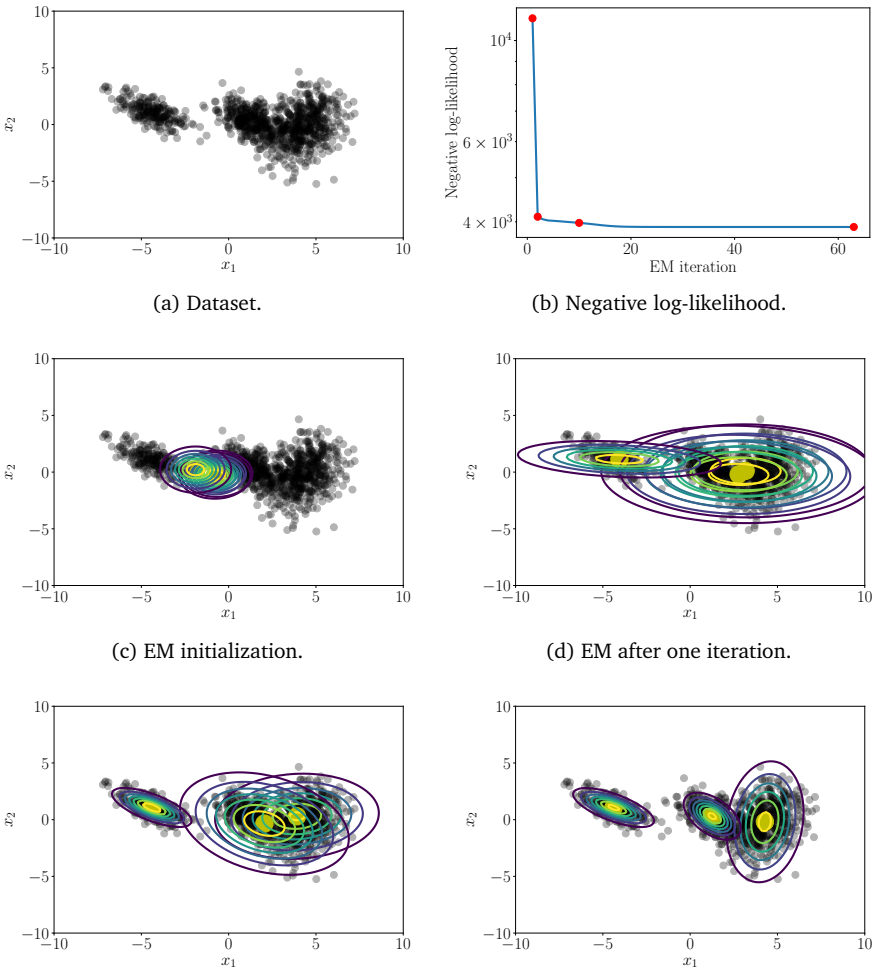


(a) Final GMM fit. After five iterations, the EM algorithm converges and returns this GMM.



(b) Negative log-likelihood as a function of the EM iterations.

Figure 11.8 EM algorithm applied to the GMM from Figure 11.2. (a) Final GMM fit; (b) negative log-likelihood as a function of the EM iteration.



(a) Dataset.

(b) Negative log-likelihood.

(c) EM initialization.

(d) EM after one iteration.

(e) EM after 10 iterations.

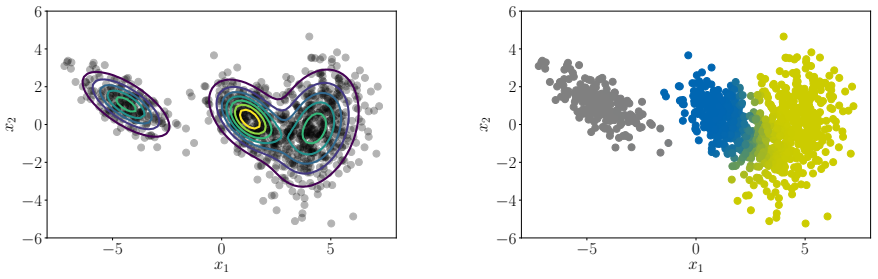
(f) EM after 62 iterations.

Figure 11.9 Illustration of the EM algorithm for fitting a Gaussian mixture model with three components to a two-dimensional dataset. (a) Dataset; (b) negative log-likelihood (lower is better) as a function of the EM iterations. The red dots indicate the iterations for which the mixture components of the corresponding GMM fits are shown in (c) through (f). The yellow discs indicate the means of the Gaussian mixture components. Figure 11.10(a) shows the final GMM fit.

When we run EM on our example from Figure 11.3, we obtain the final result shown in Figure 11.8(a) after five iterations, and Figure 11.8(b) shows how the negative log-likelihood evolves as a function of the EM iterations. The final GMM is given as

$$\begin{aligned} p(x) = &\, 0.29\mathcal{N}\left(x \mid -2.75,\, 0.06\right) + 0.28\mathcal{N}\left(x \mid -0.50,\, 0.25\right) \\ &+ 0.43\mathcal{N}\left(x \mid 3.64,\, 1.63\right) . \end{aligned} \tag{11.57}$$

We applied the EM algorithm to the two-dimensional dataset shown in Figure 11.1 with $K = 3$ mixture components. Figure 11.9 illustrates some steps of the EM algorithm and shows the negative log-likelihood as a function of the EM iteration (Figure 11.9(b)). Figure 11.10(a) shows



(a) GMM fit after 62 iterations.

(b) Dataset colored according to the responsibilities of the mixture components.

Figure 11.10 GMM fit and responsibilities when EM converges. (a) GMM fit when EM converges; (b) each data point is colored according to the responsibilities of the mixture components.

the corresponding final GMM fit. Figure 11.10(b) visualizes the final responsibilities of the mixture components for the data points. The dataset is colored according to the responsibilities of the mixture components when EM converges. While a single mixture component is clearly responsible for the data on the left, the overlap of the two data clusters on the right could have been generated by two mixture components. It becomes clear that there are data points that cannot be uniquely assigned to a single component (either blue or yellow), such that the responsibilities of these two clusters for those points are around 0.5.

11.4 Latent-Variable Perspective

We can look at the GMM from the perspective of a discrete latent-variable model, i.e., where the latent variable $\boldsymbol{z}$ can attain only a finite set of values. This is in contrast to PCA, where the latent variables were continuous-valued numbers in $\mathbb{R}^M$.

The advantages of the probabilistic perspective are that (i) it will justify some ad hoc decisions we made in the previous sections, (ii) it allows for a concrete interpretation of the responsibilities as posterior probabilities, and (iii) the iterative algorithm for updating the model parameters can be derived in a principled manner as the EM algorithm for maximum likelihood parameter estimation in latent-variable models.

11.4.1 Generative Process and Probabilistic Model

To derive the probabilistic model for GMMs, it is useful to think about the generative process, i.e., the process that allows us to generate data, using a probabilistic model.

We assume a mixture model with K components and that a data point $\boldsymbol{x}$ can be generated by exactly one mixture component. We introduce a binary indicator variable $z_k \in \{0, 1\}$ with two states (see Section 6.2) that indicates whether the kth mixture component generated that data point

so that

$$p(\boldsymbol{x} \mid z_k = 1) = \mathcal{N}\big(\boldsymbol{x} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\big)\,. \tag{11.58}$$

We define $\boldsymbol{z} := [z_1, \ldots, z_K]^\top \in \mathbb{R}^K$ as a probability vector consisting of $K-1$ many 0s and exactly one 1. For example, for $K = 3$, a valid $\boldsymbol{z}$ would be $\boldsymbol{z} = [z_1, z_2, z_3]^\top = [0, 1, 0]^\top$, which would select the second mixture component since $z_2 = 1$.

Remark. Sometimes this kind of probability distribution is called "multinoulli", a generalization of the Bernoulli distribution to more than two values (Murphy, 2012). ♢

one-hot encoding
1-of-K representation

The properties of $\boldsymbol{z}$ imply that $\sum_{k=1}^K z_k = 1$. Therefore, $\boldsymbol{z}$ is a *one-hot encoding* (also: 1*-of-*K *representation*).

Thus far, we assumed that the indicator variables z_k are known. However, in practice, this is not the case, and we place a prior distribution

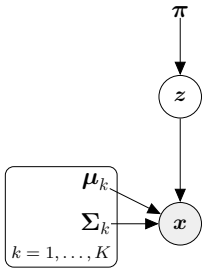
$$p(\boldsymbol{z}) = \boldsymbol{\pi} = [\pi_1, \ldots, \pi_K]^\top\,, \quad \sum_{k=1}^K \pi_k = 1\,, \tag{11.59}$$

on the latent variable $\boldsymbol{z}$. Then the kth entry

$$\pi_k = p(z_k = 1) \tag{11.60}$$

of this probability vector describes the probability that the kth mixture component generated data point $\boldsymbol{x}$.

Figure 11.11 Graphical model for a GMM with a single data point.



Remark (Sampling from a GMM). The construction of this latent-variable model (see the corresponding graphical model in Figure 11.11) lends itself to a very simple sampling procedure (generative process) to generate data:

1. Sample $z^{(i)} \sim p(\boldsymbol{z})$.
2. Sample $\boldsymbol{x}^{(i)} \sim p(\boldsymbol{x} \mid z^{(i)} = 1)$.

In the first step, we select a mixture component i (via the one-hot encoding $\boldsymbol{z}$) at random according to $p(\boldsymbol{z}) = \boldsymbol{\pi}$; in the second step we draw a sample from the corresponding mixture component. When we discard the samples of the latent variable so that we are left with the $\boldsymbol{x}^{(i)}$, we have valid samples from the GMM. This kind of sampling, where samples of random variables depend on samples from the variable's parents in the graphical model, is called *ancestral sampling*. ♢

ancestral sampling

Generally, a probabilistic model is defined by the joint distribution of the data and the latent variables (see Section 8.4). With the prior $p(\boldsymbol{z})$ defined in (11.59) and (11.60) and the conditional $p(\boldsymbol{x} \mid \boldsymbol{z})$ from (11.58), we obtain all K components of this joint distribution via

$$p(\boldsymbol{x}, z_k = 1) = p(\boldsymbol{x} \mid z_k = 1)p(z_k = 1) = \pi_k \mathcal{N}\big(\boldsymbol{x} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\big) \tag{11.61}$$

for $k = 1, \ldots, K$, so that

$$p(\boldsymbol{x}, \boldsymbol{z}) = \begin{bmatrix} p(\boldsymbol{x}, z_1 = 1) \\ \vdots \\ p(\boldsymbol{x}, z_K = 1) \end{bmatrix} = \begin{bmatrix} \pi_1 \mathcal{N}\big(\boldsymbol{x} \,|\, \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1\big) \\ \vdots \\ \pi_K \mathcal{N}\big(\boldsymbol{x} \,|\, \boldsymbol{\mu}_K, \boldsymbol{\Sigma}_K\big) \end{bmatrix}, \tag{11.62}$$

which fully specifies the probabilistic model.

11.4.2 Likelihood

To obtain the likelihood $p(\boldsymbol{x} \,|\, \boldsymbol{\theta})$ in a latent-variable model, we need to marginalize out the latent variables (see Section 8.4.3). In our case, this can be done by summing out all latent variables from the joint $p(\boldsymbol{x}, \boldsymbol{z})$ in (11.62) so that

$$p(\boldsymbol{x} \,|\, \boldsymbol{\theta}) = \sum_{\boldsymbol{z}} p(\boldsymbol{x} \,|\, \boldsymbol{\theta}, \boldsymbol{z}) p(\boldsymbol{z} \,|\, \boldsymbol{\theta}), \quad \boldsymbol{\theta} := \{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \pi_k : \; k = 1, \ldots, K\}. \tag{11.63}$$

We now explicitly condition on the parameters $\boldsymbol{\theta}$ of the probabilistic model, which we previously omitted. In (11.63), we sum over all K possible one-hot encodings of $\boldsymbol{z}$, which is denoted by $\sum_{\boldsymbol{z}}$. Since there is only a single nonzero single entry in each $\boldsymbol{z}$ there are only K possible configurations/ settings of $\boldsymbol{z}$. For example, if $K = 3$, then $\boldsymbol{z}$ can have the configurations

$$\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}. \tag{11.64}$$

Summing over all possible configurations of $\boldsymbol{z}$ in (11.63) is equivalent to looking at the nonzero entry of the $\boldsymbol{z}$-vector and writing

$$p(\boldsymbol{x} \,|\, \boldsymbol{\theta}) = \sum_{\boldsymbol{z}} p(\boldsymbol{x} \,|\, \boldsymbol{\theta}, \boldsymbol{z}) p(\boldsymbol{z} \,|\, \boldsymbol{\theta}) \tag{11.65a}$$

$$= \sum_{k=1}^{K} p(\boldsymbol{x} \,|\, \boldsymbol{\theta}, z_k = 1) p(z_k = 1 \,|\, \boldsymbol{\theta}) \tag{11.65b}$$

so that the desired marginal distribution is given as

$$p(\boldsymbol{x} \,|\, \boldsymbol{\theta}) \overset{(11.65b)}{=} \sum_{k=1}^{K} p(\boldsymbol{x} \,|\, \boldsymbol{\theta}, z_k = 1) p(z_k = 1 | \boldsymbol{\theta}) \tag{11.66a}$$

$$= \sum_{k=1}^{K} \pi_k \mathcal{N}\big(\boldsymbol{x} \,|\, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\big), \tag{11.66b}$$

which we identify as the GMM model from (11.3). Given a dataset $\mathcal{X}$, we immediately obtain the likelihood

$$p(\mathcal{X} \,|\, \boldsymbol{\theta}) = \prod_{n=1}^{N} p(\boldsymbol{x}_n \,|\, \boldsymbol{\theta}) \overset{(11.66b)}{=} \prod_{n=1}^{N} \sum_{k=1}^{K} \pi_k \mathcal{N}\big(\boldsymbol{x}_n \,|\, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\big), \tag{11.67}$$

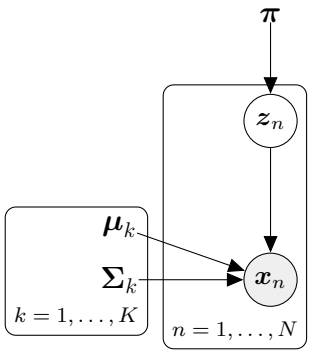


Figure 11.12 Graphical model for a GMM with N data points.

which is exactly the GMM likelihood from (11.9). Therefore, the latent-variable model with latent indicators z_k is an equivalent way of thinking about a Gaussian mixture model.

11.4.3 Posterior Distribution

Let us have a brief look at the posterior distribution on the latent variable $\boldsymbol{z}$. According to Bayes' theorem, the posterior of the kth component having generated data point $\boldsymbol{x}$

$$p(z_k = 1 \mid \boldsymbol{x}) = \frac{p(z_k = 1)p(\boldsymbol{x} \mid z_k = 1)}{p(\boldsymbol{x})}, \tag{11.68}$$

where the marginal $p(\boldsymbol{x})$ is given in (11.66b). This yields the posterior distribution for the kth indicator variable z_k

$$p(z_k = 1 \mid \boldsymbol{x}) = \frac{p(z_k = 1)p(\boldsymbol{x} \mid z_k = 1)}{\sum_{j=1}^{K} p(z_j = 1)p(\boldsymbol{x} \mid z_j = 1)} = \frac{\pi_k \mathcal{N}\big(\boldsymbol{x} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\big)}{\sum_{j=1}^{K} \pi_j \mathcal{N}\big(\boldsymbol{x} \mid \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j\big)}, \tag{11.69}$$

which we identify as the responsibility of the kth mixture component for data point $\boldsymbol{x}$. Note that we omitted the explicit conditioning on the GMM parameters $\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k$ where $k = 1, \ldots, K$.

11.4.4 Extension to a Full Dataset

Thus far, we have only discussed the case where the dataset consists only of a single data point $\boldsymbol{x}$. However, the concepts of the prior and posterior can be directly extended to the case of N data points $\mathcal{X} := \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$.

In the probabilistic interpretation of the GMM, every data point $\boldsymbol{x}_n$ possesses its own latent variable

$$\boldsymbol{z}_n = [z_{n1}, \ldots, z_{nK}]^\top \in \mathbb{R}^K. \tag{11.70}$$

Previously (when we only considered a single data point $\boldsymbol{x}$), we omitted the index n, but now this becomes important.

We share the same prior distribution $\boldsymbol{\pi}$ across all latent variables $\boldsymbol{z}_n$. The corresponding graphical model is shown in Figure 11.12, where we use the plate notation.

The conditional distribution $p(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N \mid \boldsymbol{z}_1, \ldots, \boldsymbol{z}_N)$ factorizes over the data points and is given as

$$p(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N \mid \boldsymbol{z}_1, \ldots, \boldsymbol{z}_N) = \prod_{n=1}^{N} p(\boldsymbol{x}_n \mid \boldsymbol{z}_n)\,. \tag{11.71}$$

To obtain the posterior distribution $p(z_{nk} = 1 \mid \boldsymbol{x}_n)$, we follow the same reasoning as in Section 11.4.3 and apply Bayes' theorem to obtain

$$p(z_{nk} = 1 \mid \boldsymbol{x}_n) = \frac{p(\boldsymbol{x}_n \mid z_{nk} = 1)p(z_{nk} = 1)}{\sum_{j=1}^{K} p(\boldsymbol{x}_n \mid z_{nj} = 1)p(z_{nj} = 1)} \tag{11.72a}$$

$$= \frac{\pi_k \mathcal{N}\big(\boldsymbol{x}_n \mid \boldsymbol{\mu}_k,\, \boldsymbol{\Sigma}_k\big)}{\sum_{j=1}^{K} \pi_j \mathcal{N}\big(\boldsymbol{x}_n \mid \boldsymbol{\mu}_j,\, \boldsymbol{\Sigma}_j\big)} = r_{nk}\,. \tag{11.72b}$$

This means that $p(z_k = 1 \mid \boldsymbol{x}_n)$ is the (posterior) probability that the kth mixture component generated data point $\boldsymbol{x}_n$ and corresponds to the responsibility r_{nk} we introduced in (11.17). Now the responsibilities also have not only an intuitive but also a mathematically justified interpretation as posterior probabilities.

11.4.5 EM Algorithm Revisited

The EM algorithm that we introduced as an iterative scheme for maximum likelihood estimation can be derived in a principled way from the latent-variable perspective. Given a current setting $\boldsymbol{\theta}^{(t)}$ of model parameters, the E-step calculates the expected log-likelihood

$$Q(\boldsymbol{\theta} \mid \boldsymbol{\theta}^{(t)}) = \mathbb{E}_{\boldsymbol{z} \mid \boldsymbol{x}, \boldsymbol{\theta}^{(t)}}[\log p(\boldsymbol{x}, \boldsymbol{z} \mid \boldsymbol{\theta})] \tag{11.73a}$$

$$= \int \log p(\boldsymbol{x}, \boldsymbol{z} \mid \boldsymbol{\theta}) p(\boldsymbol{z} \mid \boldsymbol{x}, \boldsymbol{\theta}^{(t)}) \mathrm{d}\boldsymbol{z}\,, \tag{11.73b}$$

where the expectation of $\log p(\boldsymbol{x}, \boldsymbol{z} \mid \boldsymbol{\theta})$ is taken with respect to the posterior $p(\boldsymbol{z} \mid \boldsymbol{x}, \boldsymbol{\theta}^{(t)})$ of the latent variables. The M-step selects an updated set of model parameters $\boldsymbol{\theta}^{(t+1)}$ by maximizing (11.73b).

Although an EM iteration does increase the log-likelihood, there are no guarantees that EM converges to the maximum likelihood solution. It is possible that the EM algorithm converges to a local maximum of the log-likelihood. Different initializations of the parameters $\boldsymbol{\theta}$ could be used in multiple EM runs to reduce the risk of ending up in a bad local optimum. We do not go into further details here, but refer to the excellent expositions by Rogers and Girolami (2016) and Bishop (2006).

11.5 Further Reading

The GMM can be considered a generative model in the sense that it is straightforward to generate new data using ancestral sampling (Bishop, 2006). For given GMM parameters $\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k$, $k = 1, \ldots, K$, we sample an index k from the probability vector $[\pi_1, \ldots, \pi_K]^\top$ and then sample a data point $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$. If we repeat this N times, we obtain a dataset that has been generated by a GMM. Figure 11.1 was generated using this procedure.

Throughout this chapter, we assumed that the number of components K is known. In practice, this is often not the case. However, we could use nested cross-validation, as discussed in Section 8.6.1, to find good models.

Gaussian mixture models are closely related to the K-means clustering algorithm. K-means also uses the EM algorithm to assign data points to clusters. If we treat the means in the GMM as cluster centers and ignore the covariances (or set them to $\boldsymbol{I}$), we arrive at K-means. As also nicely described by MacKay (2003), K-means makes a "hard" assignment of data points to cluster centers $\boldsymbol{\mu}_k$, whereas a GMM makes a "soft" assignment via the responsibilities.

We only touched upon the latent-variable perspective of GMMs and the EM algorithm. Note that EM can be used for parameter learning in general latent-variable models, e.g., nonlinear state-space models (Ghahramani and Roweis, 1999; Roweis and Ghahramani, 1999) and for reinforcement learning as discussed by Barber (2012). Therefore, the latent-variable perspective of a GMM is useful to derive the corresponding EM algorithm in a principled way (Bishop, 2006; Barber, 2012; Murphy, 2012).

We only discussed maximum likelihood estimation (via the EM algorithm) for finding GMM parameters. The standard criticisms of maximum likelihood also apply here:

- As in linear regression, maximum likelihood can suffer from severe overfitting. In the GMM case, this happens when the mean of a mixture component is identical to a data point and the covariance tends to $\mathbf{0}$. Then, the likelihood approaches infinity. Bishop (2006) and Barber (2012) discuss this issue in detail.
- We only obtain a point estimate of the parameters $\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k$ for $k = 1, \ldots, K$, which does not give any indication of uncertainty in the parameter values. A Bayesian approach would place a prior on the parameters, which can be used to obtain a posterior distribution on the parameters. This posterior allows us to compute the model evidence (marginal likelihood), which can be used for model comparison, which gives us a principled way to determine the number of mixture components. Unfortunately, closed-form inference is not possible in this setting because there is no conjugate prior for this model. However, approximations, such as variational inference, can be used to obtain an approximate posterior (Bishop, 2006).

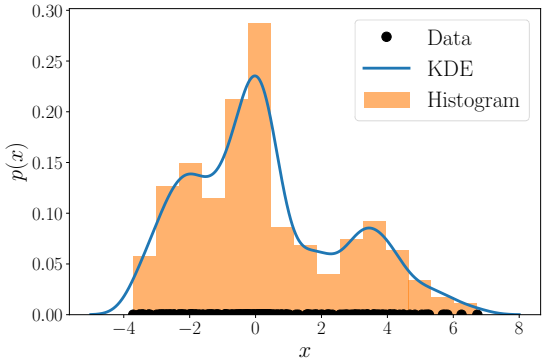


Figure 11.13 Histogram (orange bars) and kernel density estimation (blue line). The kernel density estimator produces a smooth estimate of the underlying density, whereas the histogram is an unsmoothed count measure of how many data points (black) fall into a single bin.

In this chapter, we discussed mixture models for density estimation. There is a plethora of density estimation techniques available. In practice, we often use histograms and kernel density estimation.

histogram

Histograms provide a nonparametric way to represent continuous densities and have been proposed by Pearson (1895). A histogram is constructed by "binning" the data space and count, how many data points fall into each bin. Then a bar is drawn at the center of each bin, and the height of the bar is proportional to the number of data points within that bin. The bin size is a critical hyperparameter, and a bad choice can lead to overfitting and underfitting. Cross-validation, as discussed in Section 8.2.4, can be used to determine a good bin size.

kernel density estimation

Kernel density estimation, independently proposed by Rosenblatt (1956) and Parzen (1962), is a nonparametric way for density estimation. Given N i.i.d. samples, the kernel density estimator represents the underlying distribution as

$$p(\boldsymbol{x}) = \frac{1}{Nh}\sum_{n=1}^{N} k\left(\frac{\boldsymbol{x}-\boldsymbol{x}_n}{h}\right), \tag{11.74}$$

where k is a kernel function, i.e., a nonnegative function that integrates to 1 and $h > 0$ is a smoothing/bandwidth parameter, which plays a similar role as the bin size in histograms. Note that we place a kernel on every single data point $\boldsymbol{x}_n$ in the dataset. Commonly used kernel functions are the uniform distribution and the Gaussian distribution. Kernel density estimates are closely related to histograms, but by choosing a suitable kernel, we can guarantee smoothness of the density estimate. Figure 11.13 illustrates the difference between a histogram and a kernel density estimator (with a Gaussian-shaped kernel) for a given dataset of 250 data points.