EECS 553: Machine Learning (ECE) **University of Michigan**

Gaussian Mixture Models

Winter 2023 **Clayton Scott**

1 Clustering with Gaussian Mixture Models

Suppose we wish to cluster the data set in Figure 1. The data cluster naturally into 3 groups. The k-means

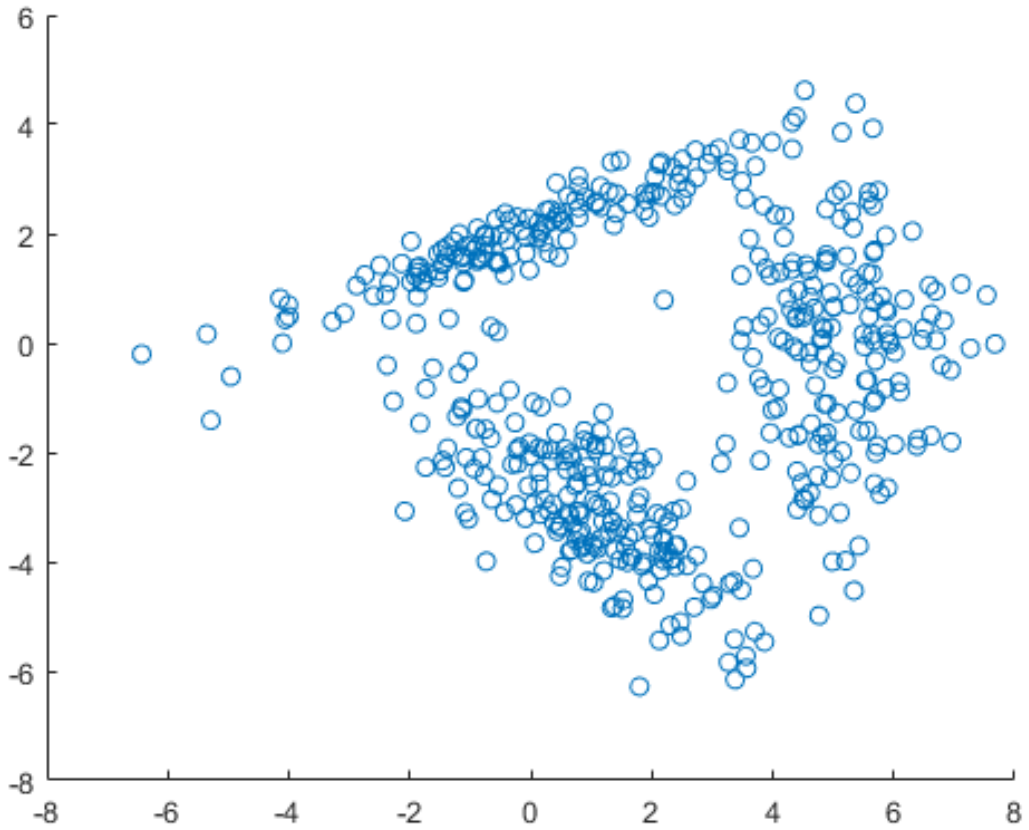

Figure 1: Data that is well-modeled as a Gaussian mixture model.

algorithm would perform poorly, because the clusters are non-spherical. Instead, each cluster is described by a bivariate Gaussian density. In these notes, we will learn about how to perform clustering by modeling the data as a *Gaussian mixture model*, and inferring the parameters of that models by maximum likelihood estimation. This approach may be viewed as a generalization of k-means because the cluster shapes are more general (although still convex), and also because the inferred cluster assignments are *soft*, meaning

each data point can have nonzero association with multiple clusters. Beyond clustering, the method studied here is also a parametric method for density estimation.

In Figure 2, the ellipses represent 90% contours (contours that contain 90% of the probability mass) of components of a Gaussian mixture model (GMM) learned by maximum likelihood estimation. The ellipses are learned by an iterative algorithm for maximum likelihood estimation known as the expectation-maximization algorithm.

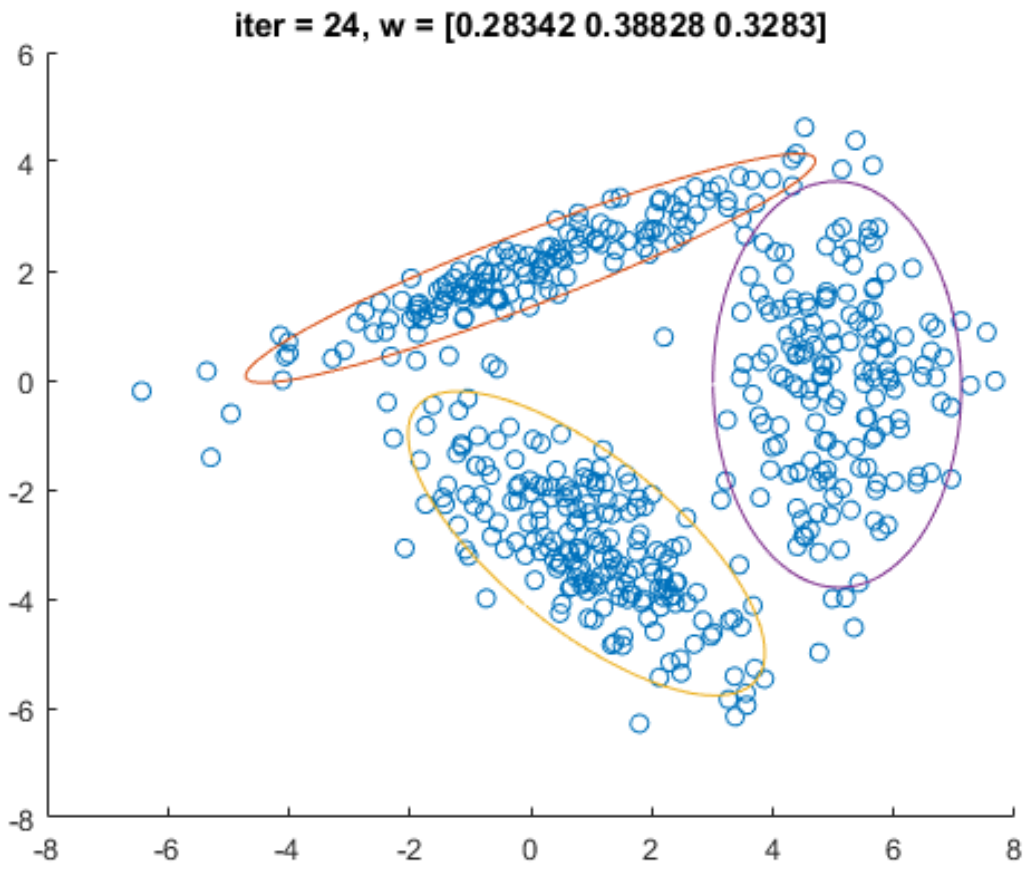

Figure 2: Clusters learned by the EM algorithm.

2 Gaussian Mixture Models

Let $\boldsymbol{\mu} \in \mathbb{R}^d$ and $\boldsymbol{\Sigma} > 0$. Recall the multivariate Gaussian density

$$\phi(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = (2\pi)^{-\frac{d}{2}} |\boldsymbol{\Sigma}|^{-\frac{1}{2}} \exp\{-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{x} - \boldsymbol{\mu})\}.$$

A random variable $\boldsymbol{X}$ follows a *Gaussian mixture model* (GMM) if its probability density function f has the form

$$f(\boldsymbol{x}) = \sum_{k=1}^{K} w_k \phi(\boldsymbol{x}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

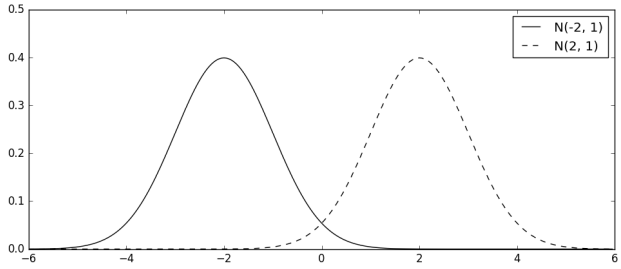

Figure 3: Densities of two different Gaussians RVs.

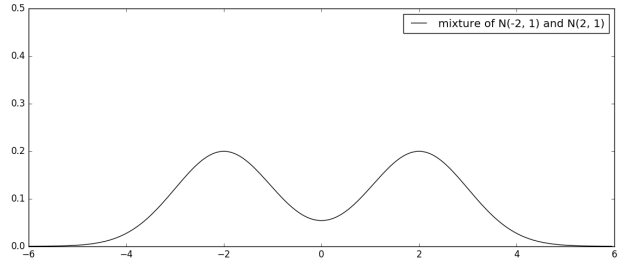

Figure 4: Density of the mixture of the two Gaussians above.

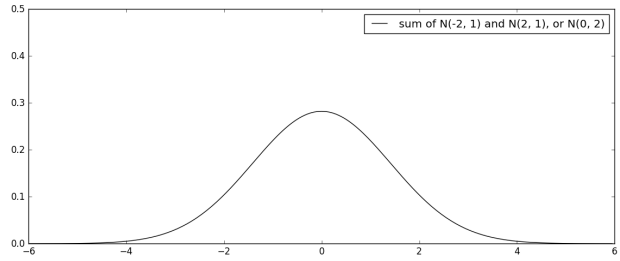

Figure 5: Density of the sum of the two Gaussians above.

where $w_k \geq 0$, $\sum_k w_k = 1$, and for all k, $\boldsymbol{\mu}_k \in \mathbb{R}^d$ and $\boldsymbol{\Sigma}_k \in \mathbb{R}^{d \times d}, \boldsymbol{\Sigma}_k > 0$.

When first learning about GMMs, it is common to confuse a *mixture* of Gaussians with a *sum* of Gaussians. A sum of Gaussian RVs is another Gaussian, and therefore unimodal. On the other hand, a mixture of Gaussians can be multimodel and is therefore non-Gaussian. See Figures 3-5.

A key to understanding GMMs is to know how to simulate a realization from a known GMM. Thus, suppose

$$\boldsymbol{\theta} = (w_1, \ldots, w_K, \boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K, \boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\Sigma}_K)$$

is known. The following two-step procedure generates a realization of the GMM with parameter vector $\boldsymbol{\theta}$.

- First, select $k \in \{1, \ldots, K\}$ at random, according to the pmf $w_1, \ldots, w_K$.
- Then draw a realization of $\boldsymbol{X} \sim \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$

Why does this work? Let $Z \in \{1, \ldots, K\}$ be a discrete RV such that $\Pr\{Z = k\} = w_k$. By definition, the pdf f of a random variable satisfies the property that for every event A, $\Pr(\boldsymbol{X} \in A) = \int_A f(\boldsymbol{x})dx$. By

the law of total expectation

$$\begin{aligned}\Pr(\boldsymbol{X} \in A) &= \sum_{k=1}^{K} \Pr(\boldsymbol{X} \in A | Z = k) \cdot \Pr(Z = k) \\ &= \sum_{k=1}^{K} \left(\int_A \phi\left(\boldsymbol{x}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right) dx \right) w_k \\ &= \int_A \left(\sum_{k=1}^{K} w_k \phi\left(\boldsymbol{x}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right) \right) dx\end{aligned}$$

and therefore the density of $\boldsymbol{X}$ is $\sum_{k=1}^{K} w_k \phi(\boldsymbol{x}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$. In other words, the two step procedure generates a realization of the given GMM.

The random variable Z is an example of a *latent* variable, also referred to as a *hidden* or *state* variable, because it is unobserved. We assume that every realization of a GMM is associated with a hidden state variable.

3 Maximum Likelihood Estimation

To perform clustering, we want to infer the parameters $\boldsymbol{\theta} = (w_1, \ldots, w_k, \boldsymbol{\mu_1}, .., \boldsymbol{\mu_k}, \boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\Sigma}_k)$ from observations $\boldsymbol{x_1}, \ldots, \boldsymbol{x_n} \in \mathbb{R}^d$. To do this we will use maximum likelihood estimation, viewing K as fixed. Recall that when $K = 1$, the MLE has a closed form solution:

$$\begin{aligned}\widehat{\boldsymbol{\mu}}_1 &= \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{x}_i \\ \widehat{\boldsymbol{\Sigma}}_1 &= \frac{1}{n} \sum_{i=1}^{n} (\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}}_1)(\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}}_1)^T.\end{aligned}$$

When $K > 1$, however, there is no closed form solution. Denote $\underline{\boldsymbol{x}} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$ for brevity. The likelihood is

$$\begin{aligned}L(\boldsymbol{\theta}; \underline{\boldsymbol{x}}) &:= \prod_{i=1}^{n} f(\boldsymbol{x}_i; \boldsymbol{\theta}) \\ &= \prod_{i=1}^{n} \left(\sum_k w_k \phi(\boldsymbol{x}_i; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right)\end{aligned}$$

and the log-likelihood is

$$\ell(\boldsymbol{\theta}; \underline{\boldsymbol{x}}) := \sum_{i=1}^{n} \log \left(\sum_{k=1}^{K} w_k \phi(\boldsymbol{x}_i; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right).$$

The value of $\boldsymbol{\theta}$ maximizing this expression does not have a closed-form expression. Because of this, we will pursue an iterative maximization strategy. This strategy hinges critically on the state variables $\underline{\boldsymbol{Z}} = (Z_1, \ldots, Z_n)$ associated with the observations.

Remark 1. As throughout these lecture notes, in the following we use capital letters to refer to random variables, and lowercase letters to refer to realizations of those variables. In particular, $\underline{\boldsymbol{x}}$ are the observed data, $\underline{\boldsymbol{z}}$ are the state variables corresponding to those particular observations, $\underline{\boldsymbol{X}}$ are the training data viewed as random variables, and $\underline{\boldsymbol{Z}}$ are their corresponding random state variables.

A natural idea for an iterative algorithm is an alternating algorithm similar to k-means:

- Given an estimate of $\boldsymbol{\theta}$, update the estimate of $\underline{\boldsymbol{z}}$.
- Given an estimate of $\underline{\boldsymbol{z}}$, update the estimate of $\boldsymbol{\theta}$.

A nice feature of this strategy is that every step can be computed efficiently and in closed-form. The details are left as an exercise.

Instead of this precise strategy, we will employ a slightly different strategy that makes "soft" cluster assignments at each iteration. Soft assignments have several advantages: (1) They allow for overlapping clusters; (2) They make the optimization problem continuous, which means the likelihood can be optimized more gracefully; (3) The algorithm generalizes to a much broader class of missing data estimation problems, as we will see later.

To motivate an algorithm with soft assignments, refer to $(\underline{\boldsymbol{x}}, \underline{\boldsymbol{z}})$ as the *complete data*. Notice that $\Pr\{Z = z; \boldsymbol{\theta}\}$ and $f(\boldsymbol{x}|z; \boldsymbol{\theta})$ determine the joint distribution of $(\boldsymbol{X}, Z)$, and define the *complete data log-likelihood* to be

$$\ell(\boldsymbol{\theta}; \underline{\boldsymbol{x}}, \underline{\boldsymbol{z}}) = \log L(\boldsymbol{\theta}; \underline{\boldsymbol{x}}, \underline{\boldsymbol{z}}) = \log \left(\prod_{i=1}^{n} \Pr\{Z_i = z_i; \boldsymbol{\theta}\} f(\boldsymbol{x}_i|z_i; \boldsymbol{\theta}) \right).$$

This function cannot be computed since $\underline{\boldsymbol{z}}$ is unobserved. However, given a current estimate $\boldsymbol{\theta}^{(j)}$ of $\boldsymbol{\theta}$ in an iterative algorithm, we can compute the expected value of $\ell(\boldsymbol{\theta}; \underline{\boldsymbol{x}}, \underline{\boldsymbol{Z}})$ with respect to $\underline{\boldsymbol{Z}}|\underline{\boldsymbol{x}}$, thereby "averaging out" the randomness due to the unknown variables. It turns out that this produces a useful surrogate for the log-likelihood, because (1) the function is easy to compute; (2) the function can be easily maximized with respect to $\boldsymbol{\theta}$, yielding the next estimate $\boldsymbol{\theta}^{(j+1)}$; (3) the iterative algorithm yields a sequences of estimates $\boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}^{(2)}, \ldots$ that monotonically increases the original (observed data) likelihood $L(\boldsymbol{\theta}; \boldsymbol{x})$.

We now turn to the details.

4 The Expectation-Maximization Algorithm for GMMs

The *expected complete-data log-likelihood* at the j^{th} iteration is denoted

$$Q(\boldsymbol{\theta}; \boldsymbol{\theta}^{(j)}) := \mathbb{E}[\ell(\boldsymbol{\theta}; \underline{\boldsymbol{X}}, \underline{\boldsymbol{Z}})|\underline{\boldsymbol{X}} = \underline{\boldsymbol{x}}; \boldsymbol{\theta}^{(j)}].$$

The expectation is with respect to the randomness of $\underline{\boldsymbol{Z}}$ given $\underline{\boldsymbol{X}} = \underline{\boldsymbol{x}}$.

The basic strategy for estimating $\boldsymbol{\theta}$ is to alternate the following two steps, known as the Expectation or E-step, and the Maximization or M-step.

Initialize $\boldsymbol{\theta}^{(0)}$, $j = 0$
Repeat
E-step: Compute

$$Q(\boldsymbol{\theta}; \boldsymbol{\theta}^{(j)}) = \mathbb{E}[\ell(\boldsymbol{\theta}; \underline{\boldsymbol{X}}, \underline{\boldsymbol{Z}})|\underline{\boldsymbol{X}} = \underline{\boldsymbol{x}}; \boldsymbol{\theta}^{(j)}].$$

M-step: Solve

$$\boldsymbol{\theta}^{(j+1)} = \arg\max_{\boldsymbol{\theta}} \; Q(\boldsymbol{\theta}; \boldsymbol{\theta}^{(j)})$$

$j \longleftarrow j + 1$
Until convergence criterion satisfied

The overall algorithm is called the *expectation-maximization* or EM algorithm. Next, we discuss how to compute the two steps for Gaussian mixture models.

4.1 E-step

Define the indicator variable

$$\Delta_{i,k} = \begin{cases} 1, & \text{if } Z_i = k \\ 0, & \text{if } Z_i \neq k \end{cases}.$$

The complete-data log-likelihood can be expressed

$$\begin{aligned}
\ell(\boldsymbol{\theta}; \underline{\boldsymbol{x}}, \underline{\boldsymbol{z}}) &= \log\left(\prod_{i=1}^{n} \Pr\{Z_i = z_i; \boldsymbol{\theta}\} f(\boldsymbol{x}_i | z_i; \boldsymbol{\theta})\right) \\
&= \log\left(\prod_{i=1}^{n} w_{z_i} \phi(\boldsymbol{x}_i; \boldsymbol{\mu}_{z_i}, \boldsymbol{\Sigma}_{z_i})\right) \\
&= \sum_{i=1}^{n} \log(w_{z_i} \phi(\boldsymbol{x}_i; \boldsymbol{\mu}_{z_i}, \boldsymbol{\Sigma}_{z_i})) \\
&= \sum_{i=1}^{n} \log\left(\sum_{k=1}^{K} \Delta_{i,k} w_k \phi(\boldsymbol{x}_i; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\right) \\
&= \sum_{i=1}^{n} \sum_{k=1}^{K} \Delta_{i,k} [\log w_k + \log \phi(\boldsymbol{x}_i; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)].
\end{aligned}$$

The last two steps hold because only one term in the inner summation is nonzero.

The only part of the last expression that depends on $\underline{\boldsymbol{Z}}$ is $\Delta_{i,k}$, and therefore the E-step amounts to calculating

$$\gamma_{i,k}^{(j)} := \mathbb{E}[\Delta_{i,k} | \underline{\boldsymbol{X}} = \underline{\boldsymbol{x}}; \boldsymbol{\theta}^{(j)}].$$

We can calculate this in terms of the current estimate $\boldsymbol{\theta}^{(j)} = (w_1^{(j)}, \ldots, w_K^{(j)}, \boldsymbol{\mu}_1^{(j)}, \ldots, \boldsymbol{\mu}_K^{(j)}, \boldsymbol{\Sigma}_1^{(j)}, \ldots, \boldsymbol{\Sigma}_K^{(j)})$ by

$$\begin{aligned}
\gamma_{i,k}^{(j)} &= \mathbb{E}[\Delta_{i,k} | \underline{\boldsymbol{X}} = \underline{\boldsymbol{x}}; \boldsymbol{\theta}^{(j)}] \\
&= \Pr(\Delta_{i,k} = 1 | \underline{\boldsymbol{X}} = \underline{\boldsymbol{x}}; \boldsymbol{\theta}^{(j)}) \\
&= \Pr(Z_i = k | \underline{\boldsymbol{X}} = \underline{\boldsymbol{x}}; \boldsymbol{\theta}^{(j)}) \\
&= \frac{\Pr(Z_i = k; \boldsymbol{\theta}^{(j)}) f(\boldsymbol{x}_i | Z_i = k; \boldsymbol{\theta}^{(j)})}{f(\boldsymbol{x}_i; \boldsymbol{\theta}^{(j)})} \\
&= \frac{w_k^{(j)} \phi(\boldsymbol{x}_i; \boldsymbol{\mu}_k^{(j)}, \boldsymbol{\Sigma}_k^{(j)})}{\sum_{\ell=1}^{k} w_\ell^{(j)} \phi(\boldsymbol{x}_i; \boldsymbol{\mu}_\ell^{(j)}, \boldsymbol{\Sigma}_\ell^{(j)})}
\end{aligned}$$

where the third step uses Bayes rule.

Thus, $\gamma_{i,k}^{(j)}$ is the fraction of the overall density value at $\boldsymbol{x}_i$ explained by the k^{th} component at iteration j. It is sometimes called the *responsibility* of cluster k for $\boldsymbol{x}_i$, and is a soft measure of cluster membership.

4.2 M-Step

We need to compute

$$\boldsymbol{\theta}^{(j+1)} = \arg\max_{\boldsymbol{\theta}} \; Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(j)})$$

where

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(j)}) = \sum_{i=1}^{n} \sum_{k=1}^{K} \gamma_{i,k}^{(j)} \left[\log w_k - \frac{d}{2} \log 2\pi - \frac{1}{2} \log |\boldsymbol{\Sigma}_k| - \frac{1}{2} (\boldsymbol{x}_i - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1} (\boldsymbol{x}_i - \boldsymbol{\mu}_k)\right].$$

The solution is

$$\boldsymbol{\mu}_k^{(j+1)} = \frac{\sum_i \gamma_{i,k}^{(j)} \boldsymbol{x}_i}{\sum_i \gamma_{i,k}^{(j)}}$$
$$\boldsymbol{\Sigma}_k^{(j+1)} = \frac{\sum_i \gamma_{i,k}^{(j)} (\boldsymbol{x}_i - \boldsymbol{\mu}_k^{(j+1)})(\boldsymbol{x}_i - \boldsymbol{\mu}_k^{(j+1)})^T}{\sum_i \gamma_{i,k}^{(j)}}$$
$$w_k^{(j+1)} = \frac{1}{n} \sum_{i=1}^{n} \gamma_{i,k}^{(j)}.$$

The formulas have intuitive interpretations. The estimated means and covariances are like ordinary sample means and covariances but weighted by the cluster responsibilities. The k^{th} cluster weight is the fraction of all data explained by k^{th} component.

4.2.1 M-Step: Weights

We need to solve

$$\begin{aligned} \min_{\{w_k\}} \ & \sum_k \gamma_k^{(j)} \log w_k \\ \text{s.t. } & w_k \geq 0 \\ & \sum_k w_k = 1, \end{aligned}$$

where $\gamma_k^{(j)} := \sum_i \gamma_{i,k}^{(n)}$. Instead of solving this problem initially, we will instead solve a relaxed version obtained by dropping the nonnegativity constraints on the weights. We will see that the solution to this relaxed problem has nonnegative weights, and therefore it solves the original problem. Thus consider

$$\begin{aligned} \min_{\{w_k\}} \ & \sum_k \gamma_k^{(j)} \log w_k \\ \text{s.t. } & \sum_k w_k = 1. \end{aligned}$$

The Lagrangian is

$$L(\boldsymbol{w}, \lambda) = \sum_k \gamma_k^{(j)} \log w_k - \lambda \sum_k w_k.$$

Since this problem is convex with an affine constraint, strong duality holds and any solution must satisfy the KKT conditions. Thus, for each k, we must have

$$\frac{\partial L}{\partial w_k} = \frac{\gamma_k^{(j)}}{w_k} - \lambda = 0.$$

Solving for w_k and enforcing the sum to one constraint, we have

$$1 = \sum_k w_k = \frac{1}{\lambda} \sum_k \gamma_k^{(j)} = \frac{n}{\lambda}.$$

Therefore $\lambda = n$ and the stated formula for w_k is obtained.

4.2.2 M-Step: Means and Covariances

The means are obtained by taking the gradient with respect to the means, setting to zero, and solving. The covariances are obtained in a manner analogous to maximum likelihood estimation for the parameters of a multivariate Gaussian random variable. The details are omitted.

4.3 Final Algorithm

To summarize, the EM algorithm for computing the maximum likelihood estimate of a Gaussian mixture model is

Initialize $\boldsymbol{\theta}^{(0)}$, $j = 0$
Repeat
E-step:

$$\gamma_{i,k}^{(j)} = \frac{w_k^{(j)} \phi(\boldsymbol{x}_i; \boldsymbol{\mu}_k^{(j)}, \boldsymbol{\Sigma}_k^{(j)})}{\sum_{\ell=1}^{k} w_\ell^{(j)} \phi(\boldsymbol{x}_i; \boldsymbol{\mu}_\ell^{(j)}, \boldsymbol{\Sigma}_\ell^{(j)})}$$

M-step:

$$\boldsymbol{\mu}_k^{(j+1)} = \frac{\sum_i \gamma_{i,k}^{(j)} \boldsymbol{x}_i}{\sum_i \gamma_{i,k}^{(j)}}$$

$$\boldsymbol{\Sigma}_k^{(j+1)} = \frac{\sum_i \gamma_{i,k}^{(j)} (\boldsymbol{x}_i - \boldsymbol{\mu}_k^{(j+1)})(\boldsymbol{x}_i - \boldsymbol{\mu}_k^{(j+1)})^T}{\sum_i \gamma_{i,k}^{(j)}}$$

$$w_k^{(j+1)} = \frac{1}{n} \sum_{i=1}^{n} \gamma_{i,k}^{(j)}.$$

$j = j + 1$
Until convergence criterion satisfied

4.4 Initialization

Like K-means, the EM algorithm for GMMs is sensitive to initialization. One possible initialization is to set $\boldsymbol{\mu}_k^{(0)}$ to be random, distinct data points, $\boldsymbol{\Sigma}_k^{(0)}$ to be the sample covariance of all data (for all k), and $w_k^{(0)} = \frac{1}{K}$ for all k. As with K-means, it may be beneficial to run the algorithm many times and take the one with largest likelihood. Another idea is to initialize EM by first running the k-means algorithm and basing $\boldsymbol{\theta}^{(0)}$ on that cluster map.

4.5 Termination

The algorithm may be terminated when

$$\ell(\boldsymbol{\theta}^{(j+1)}; \underline{\boldsymbol{x}}) - \ell(\boldsymbol{\theta}^{(j)}; \underline{\boldsymbol{x}}) \leq \epsilon$$

for some small value ϵ.

4.6 Connection to K-means

The K-means algorithm can be viewed as a special case of the EM algorithm. Consider the GMM

$$f(\boldsymbol{x}) = \sum_{i=1}^{n} w_k \phi(\boldsymbol{x}_i; \boldsymbol{\mu}_k, \sigma^2 \boldsymbol{I})$$

where σ^2 is fixed. The EM algorithm is now to iterate

$$\begin{aligned}\gamma_{i,k} &= \frac{w_k \phi(\boldsymbol{x}_i, \boldsymbol{\mu}_k, \sigma^2 \boldsymbol{I})}{\sum_{\ell=1}^K w_\ell \phi(\boldsymbol{x}_i; \boldsymbol{\mu}_\ell, \sigma^2 \boldsymbol{I})} \\ \boldsymbol{\mu}_k &= \frac{\sum \gamma_{i,k} \boldsymbol{x}_i}{\sum \gamma_{i,k}} \\ w_k &= \frac{1}{n} \sum_{i=1}^n \gamma_{i,k}.\end{aligned}$$

As $\sigma^2 \to 0$,

$$\gamma_{i,k} \to \begin{cases} 1 & \text{if } k = \arg\min \|\boldsymbol{x}_i - \boldsymbol{\mu}_\ell\| \\ 0 & \text{otherwise,} \end{cases}$$

and thus, in the limit, the EM algorithm coincides with the K-means algorithm.

Exercises

1. (★★) Determine the solution of the constrained optimization problem

$$\begin{aligned}\min_{w_1,\ldots,w_K} & \sum_{k=1}^K \log w_k \\ \text{s.t. } & w_k \geq 0 \ \forall k, \ \textstyle\sum_k w_k = 1.\end{aligned}$$

   Hint: First solve the problem without the inequality constraints, and use this result to deduce the solution of the problem with the inequality constraints.

2. (★★) Determine closed-form updates for the alternating algorithm described in Section 3.

3. (★★) Verify the formulas in the M-step for $w_k^{(j+1)}$ and $\boldsymbol{\mu}_k^{(j+1)}$.

4. (☆☆☆) Verify the formula in the M-step for $\boldsymbol{\Sigma}_k^{(j+1)}$.

5. (★★) Verify the EM algorithm described in Section 4.6.