

Mixture Models

Charles Elkan
elkan@cs.ucsd.edu

March 4, 2010

A mixture distribution is a probability density function of the form

$$f(x) = \sum_{k=1}^K \alpha_k f(x; \theta_k).$$

Here, K is the number of components in the mixture model. For each k , $f(x; \theta_k)$ is the pdf of component number k . The scalar α_k is the proportion of component number k . In order for the mixture model to be a proper pdf, it must be the case that $\sum_k \alpha_k = 1$. We normally assume also that $\alpha_k \geq 0$ for all k .

A mixture distribution is a suitable model for data that are divided into natural groups. These groups are usually called clusters, and the learning task of identifying the natural clusters in a training set is called clustering.

For a simple example, suppose we have a dataset where intuitively each point x_i comes from one of two univariate Gaussians. The mixture distribution for this situation is

$$f(x) = \alpha f(x; \mu_1, \sigma_1^2) + (1 - \alpha) f(x; \mu_2, \sigma_2^2).$$

Note that this distribution has five adjustable parameters, namely $\langle \alpha, \mu_1, \sigma_1^2, \mu_2, \sigma_2^2 \rangle$.

The way to formalize the idea that each point comes from one component is with a generative process, i.e. a randomized algorithm for generating data. The process for the example above is ...

The actual training data are just the points x_1 to x_n . Conceptually, for each x_i there is an unknown value $1 \leq z_i \leq K$. The actual data are often called the “incomplete data” or “observed data,” while the set $\{ \langle z_1, x_1 \rangle, \dots, \langle z_n, x_n \rangle \}$ is called the “complete data.” It is important to understand only the observed data are genuine. The principle of maximum likelihood says to choose parameters of

the mixture distribution that maximize the probability of the observed data, not of the notional complete data.

The expectation-maximization (EM) algorithm is the most common method for doing maximum-likelihood estimation of the parameter values of a mixture distribution. EM is an iterative procedure that starts with any initial guesses for the values of the parameters. The algorithm proceeds to repeat two phases that are called the expectation step and the maximization step.

In each expectation step we use compute the “degree of membership” w_{ik} of each data point in each component, using Bayes’ rule. The general formula is

$$w_{ik} = p(k|x_i) = \frac{f(x_i|k)p(k)}{f(x)} = \frac{f(x_i|k)p(k)}{\sum_k f(x_i|k)p(k)}.$$

For the two-Gaussian example the equation is

$$w_{ik} = \frac{\alpha_k f(x_i; \mu_k, \sigma_k^2)}{\alpha_1 f(x_i; \mu_1, \sigma_1^2) + \alpha_2 f(x_i; \mu_2, \sigma_2^2)}.$$

since $\alpha_k = p(k)$ and the parameters of component distribution number k are μ_k and σ_k^2 . In each maximization step, we compute new estimates of the parameters using the degrees of membership computed in the previous E-step. These degrees are the weights for weighted maximum likelihood estimates. For the two-Gaussian example the new estimates are

$$\begin{aligned} \mu_k &= \frac{\sum_i w_{ik} x_i}{\sum_i w_{ik}} \\ \sigma_k^2 &= \frac{\sum_i w_{ik} (x_i - \mu_k)^2}{\sum_i w_{ik}} \\ \alpha_k &= \frac{\sum_i w_{ik}}{n}. \end{aligned}$$

Each equation above is evaluated once for each $1 \leq k \leq K$, where $K = 2$ in this case.

With a new estimate for every parameter, we loop back to the E-step to compute new degrees of membership. We continue looping until changes in the parameter values, and/or changes in the degrees of membership, are tiny.

In general, the likelihood of the observed training set T , called the incomplete likelihood, is

$$L(T; \bar{\alpha}, \bar{\theta}) = \prod_{i=1}^n \sum_{k=1}^K \alpha_k f(x_i; \theta_k).$$

where $\bar{\alpha} = \langle \alpha_1, \alpha_2, \dots, 1 - \sum_{k=1}^K \alpha_k \rangle$ and $\bar{\theta} = \langle \theta_1, \dots, \theta_K \rangle$. The principle of maximum likelihood says that our goal is to maximize this likelihood. The following theorem says that the EM algorithm partially achieves this goal.

Theorem: Each iteration of EM either increases or holds constant the incomplete likelihood.

The theorem implies that EM converges to a local maximum (or to a saddle point) of the observed-data likelihood. However, it does not guarantee convergence to a global maximum of the incomplete likelihood, and in fact no such guarantee exists. Concretely, an EM algorithm typically converges to different local maxima (or saddle points) from different starting values.

There are a variety of heuristic approaches for escaping a local maximum: multiple restarts, clever initialization, and modifications to the EM algorithm itself.

There are also other methods for finding maximum likelihood estimates, such as gradient descent and variations of the Newton method. The EM algorithm is particularly useful when the E-step and/or the M-step is easy, which is true surprisingly often. However, it is ultimately just an optimization method, and completely different methods may be better for many applications.

The intuition behind deterministic annealing is the idea that, if the EM algorithm converges to a solution fast, then it is not doing much search, so the local optimum obtained is close to the initialization, and likely this solution will not be a good local optimum. So, a heuristic goal is to make the EM algorithm converge more slowly. We achieve this by making the assignment of points to clusters softer in each iteration.

In EM for fitting a mixture model, the E-step computes the degree to which each point is assigned to each component. For component k and point x_i , this weight is

$$w_{ik} = p(k|x_i) = \frac{f(x_i|k)p(k)}{f(x)} = \frac{f(x_i|k)p(k)}{\sum_k f(x_i|k)p(k)}.$$

If component k gives x_i much higher probability density $f(x_i|k)$ than any other component does, then the weight for k will be close to 1 and the weight for each other k will be close to zero. We want to bring all these weights closer together, further away from 0 and 1.

Remember what we concluded about naive Bayes: typically the class-conditional probability $p(x|y)$ is too extreme. Suppose it is too extreme because there are ex-

actly two copies of each feature. Then, $\sqrt{p(x|y)}$ would be a correct probability value. This is the t th root of $p(x|y)$ with $t = 2$, but of course there is no reason to think that $t = 2$ specifically is ideal. Heuristically, we can search for a constant t such that the t th root of $p(x|y)$ is empirically a good adjustment. This is the idea of deterministic annealing. In the equation above, we replace $f(x_i|k)$ by its t th root, for some value of t . This adjustment makes the values of $p(k|x_i)$ less extreme, i.e. more soft, for every component k .

Using the setting $t = 1$ corresponds to standard EM, while $t > 1$ gives softer assignments and $t < 1$ gives harder assignments. The limit as t tends to zero is all-or-nothing assignment, as in the k -means algorithm.

Most implementations of the idea of deterministic annealing use a schedule of t values. For example, one can start with $t = 128$, then divide t by 2 until finally $t = 1$. The final parameter values obtained with each value of t are the initial parameter values for the next value of t . For each setting of t , we can run EM to convergence, or just for a fixed number of iterations. However, a full t schedule is often not necessary. Running once to convergence with $t = 10$, then running to convergence again with $t = 1$, is often as successful as using a full schedule, and much faster of course.

How should we initialize EM if we use deterministic annealing? One answer is that because the modified algorithm is effective at searching for a good local optimum, we should initialize it to maximize the amount of search that it does. So, making each initial component a perturbed version of a global model of the data is a good idea.