The mixture of Gaussians model

Bruno A. Olshausen

November 1, 2010

The mixture of Gaussians model is probably the simplest (interesting) example of a generative model that illustrates the principles of inference and learning. It is particularly well suited to describe data containing clusters. The probability density function over data vectors \( x \in \mathbb{R}^N \) is specified as

\[
p(x) = \sum_{\alpha=1}^{K} p(x|\alpha) P(\alpha) \tag{1}
\]

where \( p(x|\alpha) \) is a Gaussian distribution with mean \( \mu_\alpha \) and variance \( \sigma^2_\alpha \) (in this case, isotropic):

\[
p(x|\alpha) = \frac{1}{(2\pi\sigma^2)^{N/2}} e^{-\frac{|x-\mu_\alpha|^2}{2\sigma^2_\alpha}} \tag{2}
\]

Thus \( p(x) \) is basically a weighted sum of \( K \) different Gaussians, enumerated by \( \alpha \). \( P(\alpha) \) specifies the weighting factors and is a normalized probability distribution parameterized by

\[
P(\alpha) = \frac{e^{\gamma_\alpha}}{\sum_\beta e^{\gamma_\beta}} \tag{3}
\]

To generate data from this model, you first choose one of the Gaussians with probability \( P(\alpha) \), and then you draw a data vector \( x \) from the corresponding Gaussian, \( p(x|\alpha) \).

The problem of inference is to determine \( \alpha \) given a data vector \( x \). Thus, we are essentially asking, “what cluster does this data vector belong to?” The problem of learning is to determine the parameters \( \mu_\alpha, \sigma_\alpha, \) and \( \gamma_\alpha \) that best fit the entire set of data.

**Inference**

In order to determine which cluster a given data vector \( x \) belongs to, we need to compute the posterior distribution:

\[
P(\alpha|x) = \frac{p(x|\alpha) P(\alpha)}{p(x)} \tag{4}
\]

where the three terms on the right side are computed according to equations 1-3 above. One might then choose the cluster \( \alpha \) that maximizes this distribution.
Learning

Learning the parameters is accomplished by doing gradient ascent on the log-likelihood of the model:

$$ L = \langle \log p(x) \rangle $$

where, as before, $\langle \rangle$ means “average over data.” Computing derivatives of $L$ with respect to $\mu_\alpha$, $\sigma^2_\alpha$, and $\gamma_\alpha$ yields the following learning rules:

$$ \Delta \mu_\alpha \propto \frac{\partial L}{\partial \mu_\alpha} = \frac{1}{\sigma^2_\alpha} \langle (x - \mu_\alpha) P(\alpha|x) \rangle $$

$$ \Delta \lambda_\alpha \propto \frac{\partial L}{\partial \lambda_\alpha} = \langle \frac{1}{2} \left[ N/\lambda_\alpha - |x - \mu_\alpha|^2 \right] P(\alpha|x) \rangle $$

$$ \Delta \gamma_\alpha \propto \frac{\partial L}{\partial \gamma_\alpha} = \langle P(\alpha|x) - P(\alpha) \rangle $$

where $\lambda_\alpha = 1/\sigma^2_\alpha$. In this case we can solve for each of the parameters directly by setting the gradients to zero, yielding

$$ \mu_\alpha = \frac{\langle x P(\alpha|x) \rangle}{\langle P(\alpha|x) \rangle} $$

$$ \sigma^2_\alpha = \frac{\langle \frac{1}{N} |x - \mu_\alpha|^2 P(\alpha|x) \rangle}{\langle P(\alpha|x) \rangle} $$

$$ P(\alpha) = \langle P(\alpha|x) \rangle $$

Note that all of the above expressions utilize $P(\alpha|x)$. Thus, learning draws upon inference. That is, in order to update the parameters, you need to infer causes ($\alpha$) given the current model. The amount that each data point influences the parameters of a particular Gaussian depends on the probability that the data point belongs to that Gaussian, given by the posterior $P(\alpha|x)$. Note however that the posterior is determined from the current model, which is initially wrong. Thus, eqs. 9-11 must be iterated until the parameters converge. This iterative procedure is referred to as the “E-M algorithm” because it involves repeated steps of Expectation (averages on the right sides of eqs. 9-11) and Maximization (since the update rules are determined by setting the gradient to zero). Although this process is guaranteed to ascend the log-likelihood, it may still get stuck in a local maximum so it is usually desirable to initialize the parameters in an intelligent manner.