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 $\mathbf{x} \in \mathcal{R}^N$ is specified as

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

where $p(\mathbf{x}|\alpha)$ is a Gaussian distribution with mean μ_α and variance σ_α^2 (in this case, isotropic):

$$p(\mathbf{x}|\alpha) = \frac{1}{(2\pi\sigma^2)^{N/2}} e^{-\frac{|\mathbf{x}-\mu_\alpha|^2}{2\,\sigma_\alpha^2}} \tag{2}$$

Thus $p(\mathbf{x})$ is basically a weighted sum of K different Gaussians, enumerated by α. $P(\alpha)$ 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 $\mathbf{x}$ from the corresponding Gaussian, $p(\mathbf{x}|\alpha)$.

The problem of *inference* is to determine α given a data vector $\mathbf{x}$. Thus, we are essentially asking, "what cluster does this data vector belong to?" The problem of *learning* is to determine the parameters μ_α, σ_α, and γ_α that best fit the entire set of data.

Inference

In order to determine which cluster a given data vector $\mathbf{x}$ belongs to, we need to compute the posterior distribution:

$$P(\alpha|\mathbf{x}) = \frac{p(\mathbf{x}|\alpha)\ P(\alpha)}{p(\mathbf{x})} \tag{4}$$

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

Learning

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

$$\mathcal{L} = \langle \log p(\mathbf{x}) \rangle \tag{5}$$

where, as before, $\langle\ \rangle$ means "average over data." Computing derivatives of $\mathcal{L}$ with respect to μ_α, σ_α, and γ_α yields the following learning rules:

$$\begin{aligned}
\Delta\mu_\alpha &\propto \frac{\partial \mathcal{L}}{\partial \mu_\alpha} \\
&= \frac{1}{\sigma_\alpha^2} \langle (\mathbf{x} - \mu_\alpha)\, P(\alpha|\mathbf{x}) \rangle && (6) \\
\Delta\lambda_\alpha &\propto \frac{\partial \mathcal{L}}{\partial \lambda_\alpha} \\
&= \left\langle \frac{1}{2} \left[N/\lambda_\alpha - |\mathbf{x} - \mu_\alpha|^2 \right] P(\alpha|\mathbf{x}) \right\rangle && (7) \\
\Delta\gamma_\alpha &\propto \frac{\partial \mathcal{L}}{\partial \gamma_\alpha} \\
&= \langle P(\alpha|\mathbf{x}) - P(\alpha) \rangle && (8)
\end{aligned}$$

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

$$\begin{aligned}
\mu_\alpha &= \frac{\langle \mathbf{x}\, P(\alpha|\mathbf{x}) \rangle}{\langle P(\alpha|\mathbf{x}) \rangle} && (9) \\
\sigma_\alpha^2 &= \frac{\left\langle \frac{1}{N} |\mathbf{x} - \mu_\alpha|^2\, P(\alpha|\mathbf{x}) \right\rangle}{\langle P(\alpha|\mathbf{x}) \rangle} && (10) \\
P(\alpha) &= \langle P(\alpha|\mathbf{x}) \rangle && (11)
\end{aligned}$$

Note that all of the above expressions utilize $P(\alpha|\mathbf{x})$. Thus, learning draws upon inference. That is, in order to update the parameters, you need to infer causes (α) 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 $P(\alpha|\mathbf{x})$. Note however that the posterior is determined from the current model, which is initially wrong. Thus, eqs. 9-11 must be interated 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.