EM for Gaussian Mixture Models

Jesse Anderton
Northeastern University

October 17, 2013

1 Problem

- You have n data samples with d real-valued features in each sample
- You want to find clusters of data points, but you also want a probabilistic description of each cluster
  - Maybe you want to assume your data set is representative of future data, and quickly decide which cluster future data points fit in
  - Maybe you want to look at the correlations between your features on a cluster by cluster basis
  - Maybe you want to whiten your data transform your features so they're decorrelated by removing shared information
  - Maybe you want a compact representation of your clusters for a classification or IR task
- One way to set up this task is as a GMM, using EM to find the distributions which define each cluster and the cluster membership of each data point

2 GMMs

- A Gaussian Mixture Model assumes that your data is generated by first choosing a "cluster" and then generating the point from the distribution for that cluster
- Each cluster is a multivariate Gaussian, which is just a Normal with multiple dimensions ("features")

- A univariate Gaussian has density:

$$\phi_1(x|\mu,\sigma) \triangleq \frac{exp\left(-\frac{1}{2}(x-\mu)^2\sigma^{-2}\right)}{\sqrt{2\pi\sigma^2}}$$

- A d-dimensional multivariate Gaussian has density:

$$\phi_d(\vec{x}|\vec{\mu},\Sigma) \triangleq \frac{exp\left(-\frac{1}{2}(\vec{x}-\vec{\mu})^T\Sigma^{-1}(\vec{x}-\vec{\mu})\right)}{\sqrt{(2\pi)^d|\Sigma|}}$$

- A GMM with k components has a weight $w_i > 0$ for each component s.t. $\sum_{i=1}^k w_i = 1$, so its density is:

$$p(\vec{x}|\theta) = \sum_{i=1}^k w_i\phi_d(\vec{x}|\theta)$$

3 EM for GMMs

3.1 Setup

- Given observed data $\mathbf{y} = \vec{y_1}, \ldots, \vec{y_n}$
- Want cluster membership $\mathbf{z} = z_1, \ldots, z_n, \forall i, z_i \in \mathbb{Z}, 1 \leq z_i \leq k$
- Want model parameters $\theta = \{(w_i, \mu_i, \Sigma_i)\}_{i=1}^k$
- Define complete data $\mathbf{x} = (\mathbf{y}, \mathbf{z})$

3.2 EM

1. Initialize: Choose $w_j^{(0)}, \mu_j^{(0)}, \Sigma_j^{(0)}, j = 1, \ldots, k$ and compute likelihood:

$$L^{(0)} = \frac{1}{n}\sum_{i=1}^n \log\left(\sum_{j=1}^k w_j^{(0)}\phi_d\left(\vec{y_i}|\vec{\mu_j}^{(0)}, \Sigma_j^{(0)}\right)\right)$$

2. E-Step:

$$\gamma_{i,j}^{(m)} = \frac{w_j^{(m)}\phi_d\left(\vec{y_i}|\vec{\mu_j}^{(m)}, \Sigma_j^{(m)}\right)}{\sum_{j=1}^k w_j^{(m)}\phi_d\left(\vec{y_i}|\vec{\mu_j}^{(m)}, \Sigma_j^{(m)}\right)} \quad i = 1, \ldots, n; j = 1, \ldots, k \text{ (prob y from j)}$$

$$n_j^{(m)} = \sum_{i=1}^n \gamma_{i,j}^{(m)} \quad j = 1, \ldots, k \text{ (prob for j)}$$

3. M-Step:

$$\begin{aligned}
w_j^{(m+1)} &= \frac{n_j^{(m)}}{n} & j &= 1, \dots, k \\
\vec{\mu_j}^{(m+1)} &= \frac{1}{n_j^{(m)}} \sum_{i=1}^{n} \gamma_{i,j}^{(m)} \vec{y_i} & j &= 1, \dots, k \\
\Sigma_j^{(m+1)} &= \frac{1}{n_j^{(m)}} \sum_{i=1}^{n} \gamma_{i,j}^{(m)} \left(\vec{y_i} - \vec{\mu_j}^{(m+1)} \right) \left(\vec{y_i} - \vec{\mu_j}^{(m+1)} \right)^T & j &= 1, \dots, k
\end{aligned}$$

4. Convergence check:
    - Compute $L^{(m+1)} = \frac{1}{n} \sum_{i=1}^{n} \log \left(\sum_{j=1}^{k} w_j^{(m+1)} \phi_d \left(\vec{y_i} | \vec{\mu_j}^{(m+1)}, \Sigma_j^{(m+1)} \right) \right)$
    - If $|L^{(m+1)} - L^{(m)}| < \delta$, stop

4 Initializing from a clustering

This model is not convex, so EM is going to find some local maximum. One way to find a reasonable maximum is to use many random initializations. Another is to use some other clustering algorithm, such as k-means, and building your initial parameters from there. Using some prior clustering:

$$\gamma_{i,j}^{(-1)} = \begin{cases} 1 & \text{if } x_i \text{ in cluster } j \\ 0 & \text{if not} \end{cases}$$

You can then compute $n_j^{(0)}$ and use the M-step to get $w_j^{(0)}, \vec{\mu_j}^{(0)},$ and $\Sigma_j^{(0)}$.

5 The Singularity Problem

For some initializations, the nearest local maximum has infinite likelihood. This happens because certain initializations prefer components with zero covariance whose mean is on one of the data points. A zero volume one-point cluster with infinite likelihood is a singularity, and isn't usually interesting to us. We want to filter them out. (At least) two ways we can do this: re-initialize, or compute MAP instead of ML using a careful prior.

5.1 Using MAP for EM

Given a parameter prior $p(\theta)$, we wish to compute:

$$\begin{aligned}
\hat{\theta}_{MAP} &= \arg\max_{\theta\in\Theta} \log p(\theta|\mathbf{y}) \\
&= \arg\max_{\theta\in\Theta} (\log p(\mathbf{y}|\theta) p(\theta)) \\
&= \arg\max_{\theta\in\Theta} (L(\theta) p(\theta))
\end{aligned}$$

So the E-step is the same, and the M-step becomes:

$$\theta^{(m+1)} = \arg\max_{\theta\in\Theta} \Big(Q(\theta|\theta^{(m)}) + \log p(\theta) \Big)$$

This version has the same monotonicity guarantees as the ML version.

6 Ways to play with GMM EM

- Using MAP, you can transition gracefully from some prior belief to what the data is telling you as you get more data
- If you choose a careful prior, you can constrain the solutions to favor more useful distributions or certain modeling assumptions, while still moving away from these assumptions if the data just doesn't fit
- Modify the algorithm so all components share a covariance matrix, if you think covariance doesn't change based on cluster. This eliminates singularities, if the number of components is much smaller than n.
- Constrain the covariance matrix so the component contours are spherical (proportional to the identity matrix) or axis-aligned (diagonal). This helps avoid failures of estimation, like singularities.