CS 340 Lec. 20: Mixture Models and EM Algorithm

AD

March 2011

Limitations of Clustering using K-Means

- No uncertainty about cluster labels $\{z_i\}_{i=1}^{N}$.
- Selection of the cost function optimized quite arbitrary.
- What about if the number of clusters K has to be estimated?

Mixture Models

- We follow a probabilistic approach where the pdf $p(\mathbf{x})$ of individual data $\{\mathbf{x}_i\}_{i=1}^{N}$ is modelled explicitly.
- A mixture model states that the pdf of data $\mathbf{x}_i$ is

$$p(\mathbf{x}_i) = \sum_{k=1}^{K} \pi_k \; p_k(\mathbf{x}_i)$$

  where $K \geq 2$, $0 \leq \pi_k \leq 1$, $\sum_{k=1}^{K} \pi_k = 1$ and $\{p_k(\mathbf{x}_i)\}_{k=1}^{K}$ are pdf.
- You can think of $p_k(\mathbf{x}_i)$ as the pdf of cluster k.

Latent Cluster Labels

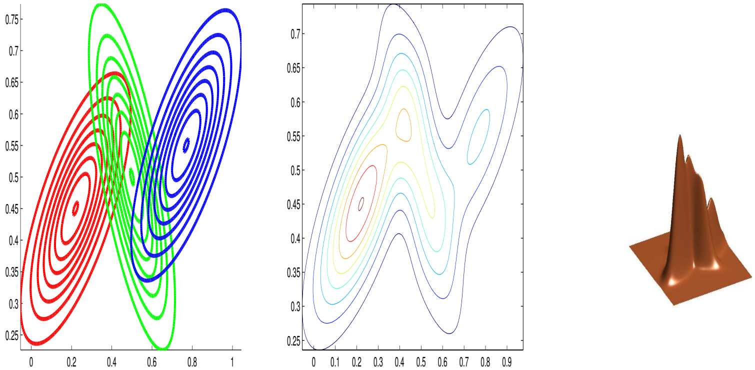
- We associate to each $\mathbf{x}_i$ a cluster label $z_i \in \{1, 2, ..., K\}$ as in K-means.
- If we set $p(z_i = k) = \pi_k$ then we can rewrite

$$p(\mathbf{x}_i) = \sum_{k=1}^{K} p(z_i = k)\, p_k(\mathbf{x}_i)$$

- Alternatively and equivalently, this means that we have now a joint distribution

$$\begin{aligned} p(\mathbf{x}_i, z_i = k) &= p(z_i = k)\, p(\mathbf{x}_i | z_i = k) \\ &= p(z_i = k)\, p_k(\mathbf{x}_i) \end{aligned}$$

Example: Mixture of Three 2D-Gaussians



(left) 3 Gaussians in 2D, we display contours of constant proba for each component (center) contours of constant proba of the mixture density (right) Surface plot of the pdf.

Posterior Distribution of Cluster Labels

- Given $\mathbf{x}_i$, we can determined

$$\begin{aligned} p\left(z_i = k \middle| \mathbf{x}_i\right) &= \frac{p\left(\mathbf{x}_i, z_i = k\right)}{\sum_{l=1}^{K} p\left(\mathbf{x}_i, z_i = l\right)} \\ &= \frac{\pi_k p_k\left(\mathbf{x}_i\right)}{\sum_{l=1}^{K} \pi_l p_l\left(\mathbf{x}_i\right)}, \end{aligned}$$

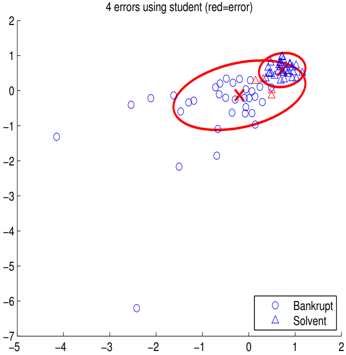
  this is sometimes known as soft clustering.

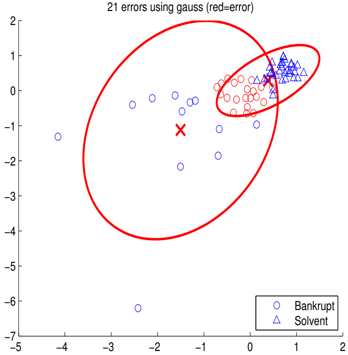
- Assume we can to assign data $\mathbf{x}_i$ to a single cluster, then we could set

$$\widehat{z}_i = \underset{k \in \{1,2,\ldots,K\}}{\arg\max}\; p\left(z_i = k \middle| \mathbf{x}_i\right),$$

  this is known as hard clustering.

Example: Mixture of Two 2D-Gaussians and 2D-Students





Mixture models trained on bankruptcy dataset modelled using a mixture of Gaussians (left) and Students (right). Estimated posterior proba is computed. If correct, blue. If incorrect, red.

Examples of Models

- Mixture of Gaussians

$$p\left(\mathbf{x}_i\right) = \sum_{k=1}^{K} \pi_k \ \mathcal{N}\left(\mathbf{x}_i; \boldsymbol{\mu}_k, \Sigma_k\right)$$

- Mixture of multivariate Bernoullis: $\mathbf{x}_i = \left(x_{i,1}, ..., x_{1,D}\right) \in \{0, 1\}^D$

$$p\left(\mathbf{x}_i\right) = \sum_{k=1}^{K} \pi_k \ p_k\left(\mathbf{x}_i\right)$$

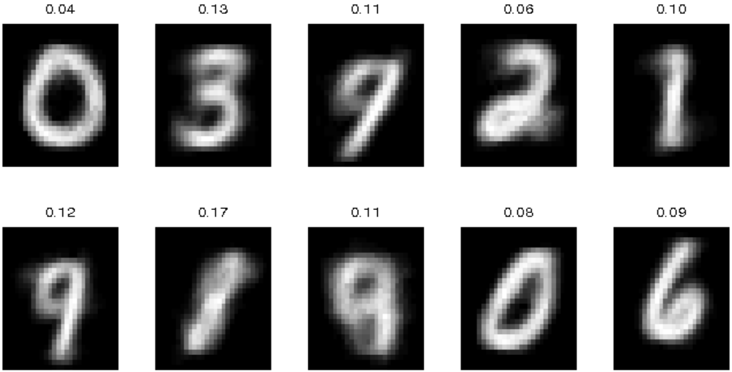
where

$$p_k\left(\mathbf{x}_i\right) = \prod_{j=1}^{D} \left(\mu_{k,j}\right)^{x_{i,j}} \left(1 - \mu_{k,j}\right)^{1-x_{i,j}}$$

Mixture of Bernoullis for MNIST Data

- Binary images of digits; $D = 784$.
- We consider applying a mixture of Bernoullis to unlabeled data.
- We set $K = 10$.
- Parameters are learned using Maximum Likelihood (more later!).

Mixture of Bernoullis for MNIST Data



A mixture of 10 multivariate Bernoulli fitted to binarized MNIST data. We display the MLE of cluster means.

Application of Mixture Models to Machine Learning

- Better models of class conditional distributions for generative classifiers
- Mixture of regressions / Mixture of Experts.
- Applications: astronomy (autoclass), econometrics (mixture of Garch models, SV), genetics, marketing, speech processing.

Maximum Likelihood Parameter Estimation for Mixture Models

- In practice, we typically have

$$p(\mathbf{x}|\,\theta) = \sum_{k=1}^{K} \pi_k \; f(\mathbf{x}; \phi_k)$$

and we need to estimate the parameters $\theta = \{\pi_k, \phi_k\}_{k=1}^{K}$ given ∞.

- The ML parameter estimates is given by

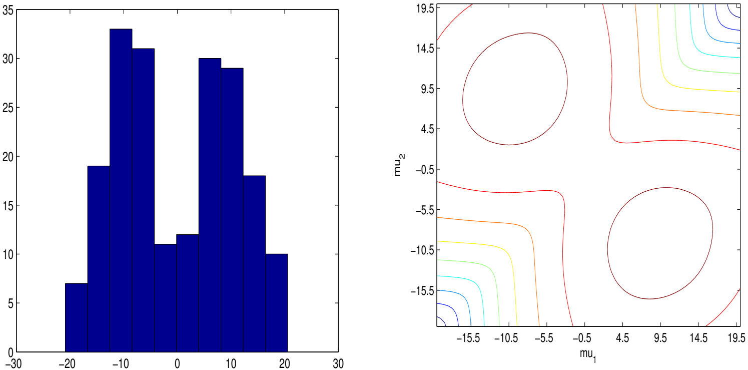
$$\widehat{\theta}_{ML} = \arg\max \; l(\theta)$$

where

$$l(\theta) = \sum_{i=1}^{N} \log \; p(\mathbf{x}_i|\,\theta)$$

- No analytic solution to this problem! Gradient methods could be used but are painful to implement.

Likelihood Surface for a Simple Example



(left) $N = 200$ data points from a mixture of two 2D Gaussians with $\pi_1 = \pi_2 = 0.5$, $\sigma_1 = \sigma_2 = 5$ and $\mu_1 = -\mu_2 = 10$. (right) Log-Likelihood surface $l(\mu_1, \mu_2)$, all the other parameters being assumed known.

Expectation-Maximization

- EM is a very popular approach to maximize $l(\theta)$ in this context.
- The key idea is to introduce explicitly the cluster labels.
- If the cluster labels where known then we would estimate θ by maximizing the so-called complete likelihood

$$
\begin{aligned}
l_c(\theta) &= \sum_{i=1}^{N} \log\; p(\mathbf{x}_i, z_i | \theta) \\
&= \sum_{i=1}^{N} \log\; \pi_{z_i}\; f\left(\mathbf{x}_i; \phi_{z_i}\right)
\end{aligned}
$$

Expectation-Maximization

- We have

$$
\begin{aligned}
l_c(\theta) &= \sum_{k=1}^{K} \left(\sum_{i=1:z_i=k}^{N} \log \ \pi_{z_i} \ f\left(\mathbf{x}_i; \phi_{z_i}\right) \right) \\
&= \sum_{k=1}^{K} \ N_k \log\left(\pi_k\right) + \sum_{i=1:z_i=k}^{N} \log \ f\left(\mathbf{x}_i; \phi_k\right)
\end{aligned}
$$

  where $N_k = \sum_{i=1:z_i=k}^{N} 1$.
- We would obtain the MLE for the complete likelihood

$$
\widehat{\pi}_k = \frac{N_k}{N}, \ \widehat{\phi}_k = \arg\max_{\phi_k} \sum_{i=1:z_i=k}^{N} \log \ f\left(\mathbf{x}_i; \phi_k\right)
$$

- Problem: We don't have access to the cluster labels!

Example: Finite mixture of scalar Gaussians

- In this case, $\phi = \left(\mu, \sigma^2\right)$

$$f\left(x;\phi\right) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{\left(x-\mu\right)^2}{2\sigma^2}\right)$$

and $\theta = \left\{\pi_k, \mu_k, \sigma^2\right\}_{k=1}^{K}$.

- In this case, the MLE estimate of the complete likelihood is

$$\begin{aligned}
\widehat{\pi}_k &= \frac{N_k}{N},\ \widehat{\mu}_k = \frac{1}{N_k} \sum_{i=1:z_i=k}^{N} x_i, \\
\widehat{\sigma}_k^2 &= \frac{1}{N_k} \sum_{i=1:z_i=k}^{N} \left(x_i - \widehat{\mu}_k\right)^2
\end{aligned}$$

Expectation-Maximization

- EM is an iterative algorithm which generates a sequence of estimates $\left\{\theta^{(t)}\right\}$ such that

$$l\left(\theta^{(t)}\right) \geq l\left(\theta^{(t-1)}\right).$$

- At iteration t, we compute

$$\begin{aligned}
&Q\left(\theta, \theta^{(t-1)}\right) = \mathbb{E}\left(l_c(\theta) \middle| \mathbf{x}_{1:N}, \theta^{(t-1)}\right) \\
&= \sum_{z_{1:N} \in \{1,2,\ldots,K\}^N} \left(\sum_{i=1}^N \log\ p\left(\mathbf{x}_i, z_i \middle| \theta\right)\right)\ p\left(z_{1:N} \middle| \mathbf{x}_{1:N}, \theta^{(t-1)}\right) \\
&= \sum_{i=1}^N\ \sum_{k=1}^K \log\ p\left(\mathbf{x}_i, z_i = k \middle| \theta\right)\ p\left(z_i = k \middle| \mathbf{x}_i, \theta^{(t-1)}\right)
\end{aligned}$$

and set

$$\theta^{(t)} = \arg\max_{\theta}\ Q\left(\theta, \theta^{(t-1)}\right)$$

E-step and M-step

- We have

$$
\begin{aligned}
& Q\left(\theta,\theta^{(t-1)}\right)=\textstyle\sum_{i=1}^{N}\ \sum_{k=1}^{K}\log\ p\left(\mathbf{x}_i,z_i=k\middle|\,\theta\right)\ p\left(z_i=k\middle|\,\mathbf{x}_i,\theta^{(t-1)}\right)\\
& =\textstyle\sum_{i=1}^{N}\ \sum_{k=1}^{K}\left\{\log\ \pi_k+\ \log f\left(\mathbf{x}_i;\phi_k\right)\right\}\ p\left(z_i=k\middle|\,\mathbf{x}_i,\theta^{(t-1)}\right)\\
& =\textstyle\sum_{k=1}^{K}\ \left(\sum_{i=1}^{N}p\left(z_i=k\middle|\,\mathbf{x}_i,\theta^{(t-1)}\right)\right)\log\ \pi_k\\
& +\textstyle\sum_{k=1}^{K}\ \left(\sum_{i=1}^{N}p\left(z_i=k\middle|\,\mathbf{x}_i,\theta^{(t-1)}\right)\ \log f\left(\mathbf{x}_i;\phi_k\right)\right)
\end{aligned}
$$

- We obtain

$$
\begin{aligned}
\widehat{\pi}_k^{(t)} &= \frac{\sum_{i=1}^{N}p\left(z_i=k\middle|\,\mathbf{x}_i,\theta^{(t-1)}\right)}{N},\\
\phi_k^{(t)} &= \underset{\phi_k}{\arg\max}\ \sum_{i=1}^{N}p\left(z_i=k\middle|\,\mathbf{x}_i,\theta^{(t-1)}\right)\ \log f\left(\mathbf{x}_i;\phi_k\right)
\end{aligned}
$$

Example: Finite mixture of scalar Gaussians

- In this case, the EM algorithm iterate

$$\widehat{\pi}_k^{(t)} = \frac{\sum_{i=1}^{N} p\left(z_i = k \middle| x_i, \theta^{(t-1)} \right)}{N}$$

and

$$\begin{aligned}
\widehat{\mu}_k^{(t)} &= \frac{\sum_{i=1}^{N} x_i p\left(z_i = k \middle| x_i, \theta^{(t-1)} \right)}{\sum_{i=1}^{N} p\left(z_i = k \middle| x_i, \theta^{(t-1)} \right)}, \\
\widehat{\sigma}_k^{2\ (t)} &= \frac{\sum_{i=1}^{N} p\left(z_i = k \middle| x_i, \theta^{(t-1)} \right) \left(x_i - \widehat{\mu}_k^{(t)} \right)^2}{\sum_{i=1}^{N} p\left(z_i = k \middle| x_i, \theta^{(t-1)} \right)}.
\end{aligned}$$

- We typically iterate the algorithm until $\left\| \theta^{(t)} - \theta^{(t-1)} \right\| < \varepsilon$.

Example: Finite mixture of Bernoulli

- Consider now the case where

$$p_k(\mathbf{x}) = \prod_{j=1}^{D} \left(\mu_{k,j}\right)^{x_j} \left(1 - \mu_{k,j}\right)^{1-x_j}$$

so $\theta = \left\{\pi_k, \mu_{k,1}, \ldots, \mu_{k,D}\right\}_{k=1}^{K}$.

- In this case, the EM algorithm yields

$$\widehat{\pi}_k^{(t)} = \frac{\sum_{i=1}^{N} p\left(z_i = k \middle| x_i, \theta^{(t-1)}\right)}{N}$$

and

$$\widehat{\mu}_{k,j}^{(t)} = \frac{\sum_{i=1}^{N} x_{i,j} p\left(z_i = k \middle| x_i, \theta^{(t-1)}\right)}{\sum_{i=1}^{N} p\left(z_i = k \middle| x_i, \theta^{(t-1)}\right)}.$$

Proof of Convergence for EM Algorithm

- We want to show that $l\left(\theta^{(t+1)}\right) \geq l\left(\theta^{(t)}\right)$ for $\theta^{(t+1)} = \arg\max_{\theta} Q\left(\theta, \theta^{(t)}\right)$.
- *Proof*: We have

$$p\left(z_{1:N} \middle| \theta, \mathbf{x}_{1:N}\right) = \frac{p\left(\mathbf{x}_{1:N}, z_{1:N} \middle| \theta\right)}{p\left(\mathbf{x}_{1:N} \middle| \theta\right)} \Leftrightarrow p\left(\mathbf{x}_{1:N} \middle| \theta\right) = \frac{p\left(\mathbf{x}_{1:N}, z_{1:N} \middle| \theta\right)}{p\left(z_{1:N} \middle| \theta, \mathbf{x}_{1:N}\right)}$$

thus

$$l(\theta) = \log p\left(\mathbf{x}_{1:N} \middle| \theta\right) = \log p\left(\mathbf{x}_{1:N}, z_{1:N} \middle| \theta\right) - \log p\left(z_{1:N} \middle| \theta, \mathbf{x}_{1:N}\right)$$

and for any value $\theta^{(t)}$

$$\begin{aligned} l(\theta) &= \underbrace{\sum_{z_{1:N}} \log p\left(\mathbf{x}_{1:N}, z_{1:N} \middle| \theta\right) . p\left(z_{1:N} \middle| \theta^{(t)}, \mathbf{x}_{1:N}\right)}_{=Q\left(\theta, \theta^{(t)}\right)} \\ &\quad - \sum_{z_{1:N}} \log p\left(z_{1:N} \middle| \theta, \mathbf{x}_{1:N}\right) . p\left(z_{1:N} \middle| \theta^{(t)}, \mathbf{x}_{1:N}\right). \end{aligned}$$

Proof of Convergence for EM Algorithm

- We want to show that $l\left(\theta^{(t+1)}\right) \geq l\left(\theta^{(t)}\right)$ for the EM, so we need to prove that

$$\begin{aligned}
&\sum_{z_{1:N}} \log p\left(z_{1:N} \middle| \theta^{(t+1)}, \mathbf{x}_{1:N}\right) . p\left(z_{1:N} \middle| \theta^{(t)}, \mathbf{x}_{1:N}\right) \\
\leq \quad &\sum_{z_{1:N}} \log p\left(z_{1:N} \middle| \theta^{(t)}, \mathbf{x}_{1:N}\right) . p\left(z_{1:N} \middle| \theta^{(t)}, \mathbf{x}_{1:N}\right)
\end{aligned}$$

- We have

$$\begin{aligned}
&\sum_{z_{1:N}} \log \frac{p\left(z_{1:N} \middle| \theta^{(t+1)}, \mathbf{x}_{1:N}\right)}{p\left(z_{1:N} \middle| \theta^{(t)}, \mathbf{x}_{1:N}\right)} . p\left(z_{1:N} \middle| \theta^{(t)}, \mathbf{x}_{1:N}\right) \\
\leq \quad & \log \sum_{z_{1:N}} \frac{p\left(z_{1:N} \middle| \theta^{(t+1)}, \mathbf{x}_{1:N}\right)}{p\left(z_{1:N} \middle| \theta^{(t)}, \mathbf{x}_{1:N}\right)} p\left(z_{1:N} \middle| \theta^{(t)}, \mathbf{x}_{1:N}\right) \quad \text{(Jensen)} \\
= \quad & \log 1 = 0.
\end{aligned}$$

About the EM Algorithm

- Some good things about EM
  - no learning rate (step-size) parameter
  - automatically enforces parameter constraints
  - very fast for low dimensions
  - each iteration guaranteed to improve likelihood
- Some bad things about EM
  - can get stuck in local minima
  - can be slower than conjugate gradient (especially near convergence)
  - requires expensive inference step
  - is a maximum likelihood/MAP method