CS395T: Large Scale Data-Mining

Lecture 13
EM for Mixture Model density Estimation

Given a set of points $\mathcal{X} = \{\mathbf{x}_i||_{i=1}^n$, the goal of unsupervised learning is to find "structure" in this data. Clusters are one example of a structure. Another is the probability density function $p(\mathbf{x})$. We have already looked at several clustering algorithms. This talk is about density estimation.

1 Density Estimation Problem

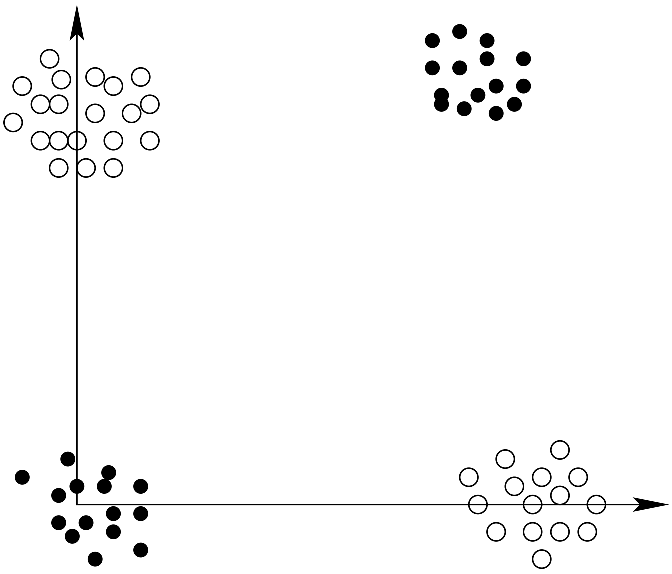
K-means clustering represents a cluster by its mean. But if the data does not contain spherical distributions around the means then K-means fails to capture the real structure in the data. A more general framework for modeling data is to estimate its probability density function (pdf) in terms of a parametric model. The simplest and most commonly used parametric model is a Gaussian or Normal distribution.

$$G(\mathbf{x}|\mu, \Sigma) = \frac{1}{\sqrt{(2\pi)^d ||\Sigma|}} \exp\left(-\frac{1}{2}(\mathbf{x}-\mu)^T \Sigma^{-1}(\mathbf{x}-\mu)\right) \tag{1}$$

where for a d dimensional space, μ is a $d \times 1$ mean vector and Σ is a $d \times d$ covariance matrix that can not only weight different dimensions differently (diagonal elements), but also model correlations between pairs of dimensions (off-diagonal elements). Thus the parameters for a Gaussian are $\{\mu, \Sigma\}$.

2 Motivation for Mixture of Gaussians

Classification algorithms like Maximum likelihood classifiers, feature extraction algorithms like Feature extraction, and visualization tools like CViz make the assumption that each class has a single "mode" i.e. their PDF can be modeled by a single Gaussian. However, this assumption is not correct. A common example is the noisy XOR problem (see Figure 1). ML classifier, Fisher and CViz will fail because the mean of both classes coincide. Also we saw that class "5" had two modes depending on how people draw 5.



Another problem is that we don't normally know in advance how many modes are there in a class. Visualization tools like CViz and any domain knowledge about the problem (e.g. for XOR we know there are two modes per class) can help us pick the number of modes. Asking the question "how many modes in the data?" is same as asking "how many clusters?". Both are still open questions for any dataset.

3 Mixture of Gaussians

Say we are given a set of n points $\mathcal{X}$ and we want to find a mixture of Gaussians representing the PDF of this set of points using K Gaussians. This is a straightforward generalization of finding K clusters. The standard form of a mixture of Gaussians (MOG) is:

$$p(\mathbf{x}) = \sum_{k=1}^{K} P(k)p(\mathbf{x}|k) = \sum_{k=1}^{K} P(k)G(\mathbf{x}|\mu_k, \Sigma_k), \tag{2}$$

where $P(k)$ is the prior (weight) of the k^{th} Gaussian and μ_k, Σ_k are the mean vector and covariance matrix of the k^{th} Gaussian. As opposed to K-means clustering that has only mean as the parameter representing each cluster, MOG has both mean and covariance representing a mode. The problem of MOG is now to find the parameters $\Theta = \{P(k), \mu_k, \Sigma_k\}_{k=1}^{K}$. We investigate the use of Expectation Maximization (EM) algorithm for doing this.

4 The EM algorithm

4.1 Missing Information

The first step in formulating the EM algorithm is to identify what is the "missing information". For example, in clustering if we know which points belong to which clusters, finding their means is easy. Similarly for MOG, if we knew which points belong to which mode, finding parameters Θ would be easy. In particular, let $y_i \in \{1, 2, \ldots, K\}$ denote the mode to which point $\mathbf{x}_i$ belongs. Then the mean μ_k is given by:

$$\mu_k = \frac{\sum_{i=1}^n I(y_i = k)\mathbf{x}_i}{\sum_{i=1}^n I(y_i = k)}, \tag{3}$$

where $I(bool)$ is 1 if the boolean argument *bool* is true and it is 0 otherwise. Similarly, the covariance Σ_k is given by:

$$\Sigma_k = \frac{\sum_{i=1}^n I(y_i = k)(\mathbf{x}_i - \mu_k)(\mathbf{x}_i - \mu_k)^T}{\sum_{i=1}^n I(y_i = k)}, \tag{4}$$

Here T denotes transpose.

Unfortunately y_i are not known. Thus in the EM terminology, the set $\mathcal{Y} = \{y_i\}_{i=1}^n$ is called the "unobserved" data, $\mathcal{X}$ is the "observed" data and the set $\mathcal{Z} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ is called the "complete" dataset. One can think of $\mathcal{Y}$ as representing a *partition* of the data into K modes.

The EM algorithm proceeds very much like the K-means clustering algorithm. Each K-means iteration t has two steps:

1. From the current mean vectors (i.e. parameters $\Theta^{(t)}$), reassign the points to their closest mean vector (i.e. find $\mathcal{Y}^{(t)}$).
2. From the last data partition (i.e. $\mathcal{Y}^{(t)}$), find the new mean vectors (i.e. parameters $\Theta^{(t+1)}$.

EM just extends these two steps in a soft manner for MOG.

4.2 The E-Step

Let $\Theta^{(t)}$ represent the MOG parameters at iteration t. (the parameters are initialized at $t = 0$ by first running K-means and then assigning the mean and covariance of the k^{th} cluster as the parameters of the corresponding Gaussian). For these parameters the **E step** estimates the *expected value of* y_i $P(y_i = k|\mathbf{x}, \Theta^{(t)})$ using the Bayes theorem:

$$P(y_i = k|\mathbf{x}_i, \Theta^{(t)}) = \frac{P(y_i = k|\Theta^{(t)})p(\mathbf{x}_i|y_i = k, \Theta^{(t)})}{p(\mathbf{x}_i|\Theta^{(t)})} \tag{5}$$

The PDF $p(\mathbf{x}_i|y_i = k, \Theta^{(t)}) = G(\mathbf{x}_i, \mu_k, \Sigma_k)$. The prior $P(y_i = k|\Theta^{(t)}) \equiv P(k|\Theta^{(t)})$ is give by the parameters computed in the last iteration as follows:

$$P(k|\Theta^{(t)}) = \frac{1}{n} \sum_{i=1}^{n} P(y_i = k|\mathbf{x}_i, \Theta^{(t-1)}) \tag{6}$$

4.3 Likelihood function

The goal of MOG is to model the data as best as possible. In K-means clustering the notion of "compact" clustering is captured by the objective function: sum of square of distances of data points from their respective means. Similarly, in MOG or density estimation problems we define an objective function that captures the notion of how well the parameters represent the probability density of the data. The most commonly used objective function is the likelihood function:

$$\mathcal{L}(\Theta|\mathcal{X}) = \prod_{i=1}^{n} p(\mathbf{x}_i|\Theta) \tag{7}$$

Since dealing with a product is difficult, the *log likelihood*:

$$\ell(\Theta|\mathcal{X}) = \log \mathcal{L}(\mathcal{X}|\Theta) = \sum_{i=1}^{n} \log p(\mathbf{x}_i|\Theta) \tag{8}$$

Since $\mathcal{X}$ is the incomplete data, $\mathcal{L}(\Theta|\mathcal{X})$ is called the "incomplete log likelihood" measure. Instead of assigning y_i in a hard fashion, as is done in the K-means algorithm, we assign it in a soft fashion. In other words, at any given time, y_i does not take one of the K values. Instead, it takes **all** the K values simultaneously but with different degrees. The degree or the probability $p(y_i = k|\mathbf{x}_i)$ with which it takes the value $y_i = k$ is given by Equation 5. Thus knowing these hidden variables $\mathcal{Y}$, we can rewrite the "complete log likelihood function" $\ell(\Theta|\mathcal{X}, \mathcal{Y})$:

$$\begin{aligned} \ell(\Theta|\mathcal{X}, \mathcal{Y}) &= \textstyle\sum_{i=1}^{n} \sum_{k=1}^{K} P(y_i = k|\mathbf{x}_i, \Theta) \log p(\mathbf{x}_i, y_i = k) \\ &= \textstyle\sum_{i=1}^{n} \sum_{k=1}^{K} P(y_i = k|\mathbf{x}_i, \Theta) \left[\log P(y_i = k|\Theta) + p(\mathbf{x}_i|y_i = k, \Theta)\right] \end{aligned} \tag{9}$$

Adding the constraint that $\sum_{k=1}^{K} P(y_i = k|\mathbf{x}_i, \Theta)$ for all $i = 1 \ldots n$, using a Lagrangian yields the following objective function:

$$\ell'(\Theta|\mathcal{X}, \mathcal{Y}) = \ell(\Theta|\mathcal{X}, \mathcal{Y}) + \sum_{i=1}^{n} \lambda_i \left(\sum_{k=1}^{K} P(y_i = k|\mathbf{x}_i, \Theta) - 1 \right). \tag{10}$$

4.4 The M-Step

The M-step in EM is equivalent to the mean update step in K-means. For a given of partitioning i.e. $\mathcal{Y}^{(t)}$ the set of parameters at the next iteration i.e. $\Theta^{(t)}$ can be computed by optimizing the objective function for the problem. In case of MOG, the objective function is given in Equation 10. Solving for $\frac{\partial \ell'}{\partial P(k)} = 0$ yields:

$$P(k) = \frac{1}{n} \sum_{i=1}^{n} P(y_i|\mathbf{x}_i, \Theta^{(t-1)}) \tag{11}$$

Solving for $\frac{\partial \ell'}{\partial \mu_k} = 0$ yields:

$$\mu_k = \frac{\sum_{i=1}^{n} P(y_i|\mathbf{x}_i, \Theta^{(t-1)})\mathbf{x}_i}{\sum_{i=1}^{n} P(y_i|\mathbf{x}_i, \Theta^{(t-1)})} \tag{12}$$

Solving for $\frac{\partial \ell'}{\partial \Sigma_k} = 0$ yields:

$$\Sigma_k = \frac{\sum_{i=1}^{n} P(y_i|\mathbf{x}_i, \Theta^{(t-1)})(\mathbf{x}_i - \mu_k)(\mathbf{x}_i - \mu_k)^T}{\sum_{i=1}^{n} P(y_i|\mathbf{x}_i, \Theta^{(t-1)})}. \tag{13}$$

5 MOG using EM

The complete algorithm for estimating the MOG using EM can now be summarized as follows:

1. **Input**: $\mathcal{X}$ and K, the number of modes in the data.

2. **Initialize**: Run K-means on $\mathcal{X}$ and compute $\Theta^{(0)}$, the initial mean, covariance and prior of each Gaussian. Set Iteration count $t = 0$.

3. **E-step**: Update the unobserved variables $\mathcal{Y}^{(t)} \leftarrow Estep(\Theta^{(t)})$ i.e. compute $P(y_i = k|\mathbf{x}, \Theta^{(t)})\ \forall\ \ i = 1 \ldots n$ and $\forall\ \ k = 1 \ldots K$ using Equation (5).

4. **M-step**: Update the parameters $\Theta^{(t+1)} \leftarrow Mstep(\mathcal{Y}^{(t)})$ i.e. update the priors, mean, and covariance of each Gaussian using Equations 11, 12, and 13 respectively.

5. If not Convergence then $t \leftarrow t + 1$ and return to step 3.