STAT 561 Lab 4 February 10, 2019

K-means and Gaussian Mixture Model

1 Clustering

Cluster analysis or clustering is the task of grouping a set of objects in such a way that objects in the same group (called a cluster) are more similar (in some sense) to each other than to those in other groups (clusters).

K-means is a *partition or centroid-based* clustering method.

Gaussian Mixture Model is a *probabilistic distribution* model.

2 Gaussian Mixture Model

Mixture models make use of latent variables to model different parameters for different groups (or **clusters**) of data points. A Gaussian Mixture Model (GMM) is a mixture of Gaussians with K components.

The Expectation Maximization Algorithm is a general method for parameter estimation when the model depends on unobserved or latent variables. We use EM to update our estimation for the parameters in the Gaussian Mixture Model.

1. (E-step) Assume we are given N data samples $\{x_i\}_{i=1}^N$ and a current guess of parameters $\theta_0 = (\pi, \mu, \Sigma)$. Write down the formula for $p(z_i = k \mid x_i, \theta_0)$, the posterior probability of assigning data point x_i to cluster component k.

   Let's denote $p(z_i = k \mid x_i, \theta_0)$ as r_{ik}.

2. (M-step) The *log-likelihood* function is given as:

$$l(\theta) = \sum_{i=1}^{N} \sum_{k=1}^{K} r_{ik}(\log \pi_k + \log \mathcal{N}(x_i \mid \mu_k, \Sigma_k))$$

   (a) Write down the optimal choice for mean vectors μ_k for all $k = 1, \ldots, K$:

   $\underset{\mu_k}{\operatorname{argmax}}(l(\theta)) =$

   (b) Write down the optimal choice for covariance matrices Σ_k for all $k = 1, \ldots, K$:

   $\underset{\Sigma_k}{\operatorname{argmax}}(l(\theta)) =$

   (c) Write down the optimal choice for *weight* vectors π_k for all $k = 1, \ldots, K$:

   $\underset{\pi_k}{\operatorname{argmax}}(l(\theta)) =$

3 K-means Clustering

Compared to GMM, K-means is a *hard* clustering method. It has no underlying probability model, each data point is assigned to a specific *centroid*.

Use the K-means algorithm and Euclidean distance to cluster the 8 data points given into K = 3 clusters.

$x^{(1)} = (2,8), x^{(2)} = (2,5), x^{(3)} = (1,2), x^{(4)} = (5,8), x^{(5)} = (7,3), x^{(6)} = (6,4), x^{(7)} = (8,4), x^{(8)} = (4,7)$.

	$x^{(1)}$	$x^{(2)}$	$x^{(3)}$	$x^{(4)}$	$x^{(5)}$	$x^{(6)}$	$x^{(7)}$	$x^{(8)}$
$x^{(1)}$	0	3.0000	6.0828	3.0000	7.0711	5.6569	7.2111	2.2361
$x^{(2)}$	3.0000	0	3.1623	4.2426	5.3852	4.1231	6.0828	2.8284
$x^{(3)}$	6.0828	3.1623	0	7.2111	6.0828	5.3852	7.2801	5.8310
$x^{(4)}$	3.0000	4.2426	7.2111	0	5.3852	4.1231	5.0000	1.4142
$x^{(5)}$	7.0711	5.3852	6.0828	5.3852	0	1.4142	1.4142	5.0000
$x^{(6)}$	5.6569	4.1231	5.3852	4.1231	1.4142	0	2.0000	3.6056
$x^{(7)}$	7.2111	6.0828	7.2801	5.0000	1.4142	2.0000	0	5.0000
$x^{(8)}$	2.2361	2.8284	5.8310	1.4142	5.0000	3.6056	5.0000	0

Table 1: Distance matrix for training data from Table 1.

1. Lets assume that points $x^{(3)}, x^{(4)}$ and $x^{(6)}$ were chosen as the initial centroids. Perform one iteration of the K-means algorithm and report the coordinates of the resulting centroids.

2. Calculate the loss function $J(c^{(1)}, \ldots, c^{(N)}, \mu_1, \ldots, \mu_K = \frac{1}{N}\sum_{i=1}^{N} ||x^{(i)} - \mu_{c^{(i)}}||^2$ before and after the first iteration of K-means using the initialization given in (a).

4 K-means and GMM Example

The k-means clustering model explored in the previous section is simple and relatively easy to understand, but its simplicity leads to practical challenges in its application. In particular, the non-probabilistic nature of k-means and its use of simple distance-from-cluster-center to assign cluster membership leads to poor performance for many real-world situations.

https://jakevdp.github.io/PythonDataScienceHandbook/05.12-gaussian-mixtures.html