Task 2 Unsupervised Learning: K-Means and Mixture of Gaussians

Garoe Dorta-Perez
CM50246: Machine Learning and AI

December 10, 2014

1 Introduction

Unsupervised learning is a challenging task, given that the only information the clustering has is the data itself. The K-Means technique uses a non probabilistic approach to solve the problem. While the Mixture of Gaussians performs a similar task but using a Bayesian approach.

2 The problem

The main objective is to classify our data in different clusters. We are going to assume that each cluster is linearly separable from the rest.

2.1 K-Means

K-Means algorithm is based on clustering according to a set of prototype vectors. Each prototype represents a cluster and the data it contains.

The input would be the input matrix data X and the number of cluster K. While the output is a matrix with the cluster means $\boldsymbol{\mu}$ and a one dimensional array h with indexes each point to its cluster.

Initially the means of each cluster are assigned randomly, by sampling from a multivariate normal distribution with the data mean and covariance.

The algorithm then does:

1. Assign each point to its nearest cluster.

2. Recalculate the means of the clusters.

This processes in then repeated until no points get reassigned in one iteration. The out of the algorithm varies greatly due to its random initialization, suffering greatly from local minima.

2.2 Mixture of Gaussians

Mixture of Gaussians assumes the data can be modelled using a weighted sum of K normal distributions, as shown in Equation 1. Where for each of the normal distribution, $\boldsymbol{\mu}_k$ is the mean, $\boldsymbol{\Sigma}_k$ is the covariance, λ_k is the weight and $\boldsymbol{\theta} = \{\lambda_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^{K}$

$$\begin{aligned} Pr(\mathbf{x}|\boldsymbol{\theta}) &= \sum_{k=1}^{K} \lambda_k Norm_x \left[\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right], \\ s.t. \qquad & \sum_{k=1}^{K} \lambda_k = 1 \end{aligned} \tag{1}$$

A close form solution is not found if we opt for a maximum likelihood approach to learn the parameters. Introducing a hidden variable .....

The solution is to perform a expectation maximization technique. In order to do that we first need to add a hidden variable $h \in \{1 \cdots K\}$ to our model, as shown in Equation 2.

$$\begin{aligned} Pr(\mathbf{x}|h, \boldsymbol{\theta}) &= Norm_{\mathbf{x}} \left[\boldsymbol{\mu}_h, \boldsymbol{\Sigma}_h\right], \\ Pr(h|\boldsymbol{\theta}) &= Cat_h \left[\boldsymbol{\lambda}\right], \\ Pr(\mathbf{x}|\boldsymbol{\theta}) &= \sum_{k=1}^{K} Pr(\mathbf{x}, h = k, \boldsymbol{\theta}). \end{aligned} \tag{2}$$

This uses a random initialization, then performs an E-step and a M-step. In the first one the posterior probability distribution $Pr(h_i|\mathbf{x}_i)$ of each hidden variable h_i is calculated, using Equation 3;

$$Pr(h_i = k|\mathbf{x}_i, \boldsymbol{\theta}^{[t]}) = \frac{\lambda_k Norm_{\mathbf{x}_i} \left[\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right]}{\sum_{j=1}^{K} \lambda_k Norm_{\mathbf{x}_i} \left[\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j\right]} = r_{ik} \tag{3}$$

In the M-step the λ, $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are updated, as shown in Equation 4.

$$\begin{aligned} \lambda_k^{[t+1]} &= \frac{\sum_{i=1}^{I} r_{ik}}{\sum_{j=1}^{K} \sum_{i=1}^{I} r_{ij}}, \\ \boldsymbol{\mu}_k^{t+1} &= \frac{\sum_{i=1}^{I} r_{ik} \mathbf{x}_i}{\sum_{i=1}^{I} r_{ik}}, \\ \boldsymbol{\Sigma}_{\mathbf{k}}^{\mathbf{t+1}} &= \frac{\sum_{i=1}^{I} r_{ik} (\mathbf{x}_i - \boldsymbol{\mu}_k^{[t+1]})(\mathbf{x}_i - \boldsymbol{\mu}_k^{[t+1]})^T}{\sum_{i=1}^{I} r_{ik}} \end{aligned} \tag{4}$$

A better initialization for the means of the gaussians is to run the k-means first and then use cluster means instead of the previous random values.

3 Results