AIML - CS 337

Lecture 16: Mixture Models

10th October 2022

Lecturer: Abir De *Scribe: Group 33, Group 34*

1 Prologue

Mixture models are useful when we have a large dataset with heterogenous data.
Typical models like regression assume that data is generated by adding noise to a central distribution and is centered around a single mode. However, we can have multimodal spatial characteristics as well, like :

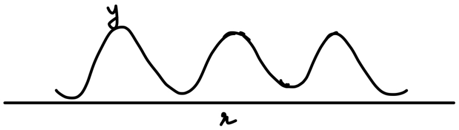

Figure 1: Multi modal spatial characteristics

We will be discussing problems where we are given a dataset and are asked to partition it into a (given) certain number of classes, taking some similarity measure(s) or probabilistic models as a basis. It's the latter part where mixture models come into picture!
We will first investigate the former version a bit and then move on to mixture models.

2 Problem Statement

Given a dataset D, segregate it into k clusters $\{C_1, C_2 \cdots C_k\}$. More formally (and for the sake of introducing notations for the rest of the document)

$$\text{Given a dataset, } D = \{x_1, x_2, \ldots, x_n\} \text{ of } n \text{ points}$$

$$\text{Our task is to find a labelling function } z : \{1, 2, \ldots, n\} \to \{1, 2, \ldots, k\}$$

Here, $z(i) = j$ means that the point x_i belongs to the cluster C_j. For convenience, we will often write z_i to denote $z(i)$.
Now that we have defined our problem,let us approach it in a couple of settings.

3 Objective functions

One recurrent theme in machine learning is to view our model as a function optimizing a reasonable objective function in a reasonable hypothesis class. Throughout this section, our hypothesis class for z will be simply the set of all functions from $\{1, 2, \ldots, n\}$ to $\{1, 2, \ldots, k\}$. As for the objective functions, here are a few candidates proposed in class:

Maximising pair wise distances between points belonging to different clusters

$$\max_{z} \sum_{\substack{i,j \\ z(i) \neq z(j)}} ||x_i - x_j||$$

The intuition behind this optimization problem is to in a sense keep different cluster points as far as possible in the Euclidean space. This function for any z is a bit expensive to compute as We have to iterate over all $\binom{k}{2}$ pairs of clusters and compute sum of distances between all the pair of points in which one point lies in one cluster and the other in other.

Minimizing pair wise distances belonging to same cluster

$$\min_{z} \sum_{\substack{i,j \\ z(i) = z(j)}} ||x_i - x_j||^2$$

The intuition here is to minimize the total sum of pairwise distances of points belonging to the same cluster and thus in a way ensuring that the points in the same cluster are as close to each other as possible. Call this problem P$_0$.

Point-wise distances instead of pair-wise distances

$$\min_{\substack{z \\ \bar{x_1}, \bar{x_2}, \ldots, \bar{x_k}}} \sum_{c=1}^{c=k} \sum_{z_i = c} ||x_i - \bar{x_c}||^2$$

The intuition behind this is to think of clusters as being identified by the points $\bar{x_c}$ and trying to make all the points as close to the representative point of the cluster they belong to.
It actually turns out that $\bar{x_c}$ must be the mean of points lying in cluster C_c since sum of squares of distances from a point to all the points of a given finite set of points is minimized (uniquely) at their mean! Call this optimization problem P$_1$.

Going from hard constraints to soft constraints

$$\min_{\mu_1, \mu_2, \ldots, \mu_n} \sum_{c=1}^{c=k} \sum_{i,j} \mu_{ic}\mu_{jc} ||x_i - x_j||^2$$

Here, μ_i's are probability distributions over the set $\{1, 2, \ldots, k\}$ and μ_{ic} denotes the probability of point i belonging to cluster c.

Note that for this objective function, our focus is to attach probabilities to a point belonging to a cluster instead of deterministically assigning clusters to points.
The intuition behind this objective function is to find those μ_i's for which the expected value of sum of squares of distances between points of same cluster is as small as possible and thus in a way, points of same cluster are $expectedly$ as close to each other as possible.
Call this problem P$_2$.

$$\min_{\substack{\mu_1,\mu_2,\ldots,\mu_n \\ \bar{x_1},\bar{x_2},\ldots,\bar{x_n}}} \sum_{c=1}^{c=k} \sum_{i=1}^{i=n} \mu_{ic}||x_i - \bar{x_c}||^2$$

Here, the meaning of μ_i's the same as in P$_2$. $\bar{x_i}$'s can be viewed as representatives of clusters as in P$_1$. The intuition behind this problem is that we want to make the points as close to their cluster representatives as possible, in expectation.
Call this problem P$_3$.

Note that it is linear in μ_{ic} and hence at optimization condition it will lie at the boundary, i.e. it will either be 0 or 1. We also know that for a fixed i, $\sum_c \mu_{ic} = 1$. Hence, at the optimization condition, for a fixed i, μ_{ic} will be 1 for a particular c_0 and 0 for the rest. This can be interpreted as point i belonging to cluster c_0.

$$\mu_{ic} = \begin{cases} 1, & \text{if } c = \arg\min_{c'} ||x_i - \bar{x_{c'}}||^2 \\ 0, & \text{otherwise} \end{cases}$$

We therefore do the optimization in two steps where we first fix $\bar{x_c}$ making it a linear optimization problem owing to the above condition. Followed by which we optimize for the mean values $\bar{x_c}[\forall c \in 1, 2 \ldots k]$.

Hence we can say that at optimal condition P$_1$ $\equiv$ P$_3$! Note however that we can't say P$_0$ $\equiv$ P$_2$ at optimal since it is a case of quadratic optimization and we can't say μ will hit boundary conditions.

Also, in (5), at optimal, $\bar{x_c} = \sum_{i \in c} x_i/n_c$. This is because the optimal condition for $\min_{x_c} \sum ||x_i - \bar{x_c}||^2$ reduces to x_c being the mean of the points.

Following is the k-means algorithm for optimization of the objective function P$_0$, i.e., without prior probability of distributions of the cluster buckets.

```
Algorithm 1: The k-Means Clustering Algorithm
  input: $\boldsymbol{\chi} \subset \mathbb{R}^n$; Number of clusters $k$
  initialize: Randomly chosen initial centroids $\mu_1, \ldots, \mu_k$
  repeat until convergence
      $\forall i \in [k]$ set $C_i = \{x \in \chi : i = argmin_j ||x - \mu_j||\}$
      (break ties in some arbitrary manner)
      $\forall i \in [k]$ update $\mu_i = \frac{1}{|C_i|} \sum_{x \in C_i} x$
```

Now, we will change our focus to probabilistic models.

4 Probabilistic setting

In this section, we will look at this problem from a probabilistic sampling perspective. We will assume that the data points $x_1, x_2, \ldots, x_n$ are respectively sampled from iid instances $X_1, X_2, \ldots, X_n$ of a random variable X which can be thought of as some sort of a mixture of k random variables $Y_1, Y_2, \ldots, Y_k$ (and hence the term, mixture models) over the sample space from which D is sampled. Sampling from X is done as follows:- first we pick a random number c from a probability distribution π over $\{1, 2, \ldots, k\}$ and then pick a random number x from Y_c and associate x with the cluster c. The Y_i's are generally assumed to be coming from the same class of distribution but with different parameters. Let θ be the set of parameters parametrising Y_i's and π.
Observe that π_c(short for $\pi(c)$) will be the probability of a random point sampled from X belonging to Cluster C_c. Also, let Z_i be the random variable describing the cluster to which x_i belongs to and define π_{ic} as the probability that x_i belongs to cluster c, i.e. $\pi_{ic} = P(Z_i = c|X_i = x_i, \theta)$. Using Baye's rule, we can write π_{ic} as:

$$\begin{aligned}
\pi_{ic} &= P(Z_i = c|X_i = x_i, \theta) \\
&= \frac{P(X_i = x_i|Z_i = c, \theta)P(Z_i = c|\theta)}{P(X_i = x_i)} \\
&= \frac{P(X_i = x_i|Z_i = c, \theta)P(Z_i = c|\theta)}{\sum\limits_{j=1}^{j=k} P(X_i = x_i|Z_i = j, \theta)P(Z_i = j|\theta)} \\
&= \frac{P(Y_c = x_i|\theta)\pi_c}{\sum\limits_{j=1}^{j=k} P(Y_j = x_i|\theta)\pi_j}
\end{aligned} \tag{1}$$

Now that we have our probabilistic model ready, we can can apply a host of probabilistic tools like MLE, Bayesian inference, etc to estimate θ.
The likelihood function will be:

$$\begin{aligned}
\mathcal{L}(\theta|\vec{X} = \vec{x}) &= P(\vec{X} = \vec{x}|\theta) \\
&= \sum_{\vec{z} \in \{1,2,\ldots,k\}^n} P(\vec{X} = \vec{x}|\vec{Z} = \vec{z}, \theta)P(\vec{Z} = \vec{z}|\theta) \\
&= \sum_{\vec{z} \in \{1,2,\ldots,k\}^n} \prod_{i=1}^{i=n} (P(X_i = x_i|Z_i = z_i, \theta)P(Z_i = z_i|\theta)) \\
&= \sum_{\vec{z} \in \{1,2,\ldots,k\}^n} \prod_{i=1}^{i=n} (P(Y_{z_i} = x_i|\theta)\pi_{z_i} \\
&= \mathbb{E}_{\vec{Z}} P(Y_{z_i} = x_i|\theta)
\end{aligned} \tag{2}$$

Note that here $\vec{x} = \{\boldsymbol{x}_1, x_2, \ldots, x_n\}$ and similarly other vector notations are defined. We can maximize the above likelihood function to obtain maximum likelihood estimate for θ and one possible approach to cluster can be to assign that cluster c to a point x_i ,for which π_{ic} is maximum as per the estimated θ.