Lecture 20 Clustering

The aim of clustering is to group unlabelled data into clusters according to some similarity or distance measure. Informally, a cluster is thought of as a set of points sharing some pattern or structure. Let us consider some common clustering methods.

20.1 K - means

Let us consider an unlabelled date set $\mathcal{D} = \{\vec{x}_n\}_{n=1}^{N}$ where $\vec{x}_n \in \mathbb{R}^p$ with, as usual, p is the number of features. Let us also consider a set of K clusters which have cluster centres (or cluster means) $\{\vec{\mu}_k\}_{k=1}^{K}$ with $\vec{\mu}_k \in \mathbb{R}^p$. These cluster centres can be computed empirically in the cluster procedure. The cluster mean can be thought of as the representative of a given cluster, to which certain data points are assigned. K - means clustering can be formulated as follows: given a fixed integer K, find the cluster means $\{\vec{\mu}\}$ and the data point assignments in order to minimise the following objective function

$$C(\{\vec{x}, \vec{\mu}\}) = \sum_{k=1}^{K} \sum_{n=1}^{N} r_{nk} (\vec{x}_n - \vec{\mu}_k)^2$$

where $r_{nk} \in \{0, 1\}$ is a binary variable called ASSIGNMENT. The assignment r_{nk} is 1 if x_n is assigned to a clusterkand 0 otherwise. Note that $\sum_k r_{nk} = 1 \ \forall n$ and $\sum_n r_{nk} = N_k$

where N_k is the number of points assigned to a cluster k. The minimisation of this objective function can be understoo as trying to find the best cluster means such that the variance in each cluster is minimised. The K-means algorithm alternates two steps.

1 EXPECTATION Given a set of assignments $\{r_{nk}\}$ minimise C with respect to $\vec{\mu}_k$. Taking the derivative and setting it to zero yields the following update rule

$$\vec{\mu}_k = \frac{1}{N_k} \sum_n r_{nk} \vec{x}_n$$

2 MAXIMISATION Given a set of cluster means $\{\vec{\mu}_k\}$ find the assignments $\{r_{nk}\}$ which minimise C. This is achieved by assigning each date point to their nearest cluster mean

$$r_{nk} = \begin{cases} 1 & \text{if } k = \text{argmin}_{k'} (\vec{x}_n - \vec{\mu}_{k'})^2 \\ 0 & \text{otherwise} \end{cases}$$

K means clustering consists in alternating between these two steps until some convergence criterion is met. The algorithm should terminate when the change in the objective function from one iteration to another becomes smaller than a pre-specified threshold. The K-means algorithm always converges to a (local) minimum of C. Because C is generally a non-convex function, one has to run the algorithm with diffrent initial random cluster centre initialisations and post-select the local minimum. The K-means algorithm scales as $O(KN)$. Iterating the algorithm starting from different initial configurations may be inefficient. A way of addressing the problem is to take the initial point with uniform probability and then each of the subsequent points with a probability which is proportional to its square distance w.r.t. the nearest cluster mean

$$p\left(\vec{\mu}_k = \vec{x}_n\right) = \frac{D_{k-1}(\vec{x}_n)}{\sum_{n'=1}^{N} D_{k-1}(\vec{x}_{n'})} \qquad D_k(\vec{x}) = \min_{j=1}^{k-1} \left\| \vec{x} - \vec{\mu}_j \right\|_{\ell_2}^2$$

In this way points that are farther away from the cluster mean have higher probability to be selected. K-means++.

20.2 Hierarchical clustering: agglomerative methods

Agglomerative clustering is a bottom-up approach that starts from small initial clusters which are then progressively merged to form larger clusters. The merging process generates a hierarchy of clusters that can be visualised in the form of a dendogram. Agglomerative methods are usually specified by defining a measure distance between clusters X and Y as $d(X, Y) \in \mathbb{R}$. At each step, the two clusters that are the closest with respect to the distance measure are merged until a single cluster is left. The algorithm can be summarised as follows.

1 Initialise each point to its own cluster

2 Given a set of clusters $X_1, X_2, \dots, X_K$, merge clusters until only one cluster is left $(K = 1)$:

  a) find the closest pair of clusters (X_i, X_j):

$$(i, j) = \arg\min_{(i', j')} d(X_i', X_j')$$

b) Merge the pair. Update $K \leftarrow K - 1$

The most popular distances used in agglomerative methods (often called linkage methods) are as follows

1 single-linkage: the distance between clusters i and j is defined as the minimum distance between two elements of the different clusters

$$d(X_i, X_j) = \min_{\vec{x}_i \in X_i, \vec{x}_j \in X_j} \|\vec{x}_i - \vec{x}_j\|_{\ell_2}$$

2 complete-linkage: the distance between clusters i and j is defined as the maximum distance between two elements of the different clusters

$$d(X_i, X_j) = \max_{\vec{x}_i \in X_i, \vec{x}_j \in X_j} \|\vec{x}_i - \vec{x}_j\|_{\ell_2}$$

3 average-linkage: average distance between points of different clusters

$$d(X_i, X_j) = \frac{1}{|X_i||X_j|} \sum_{\vec{x}_i \in X_i, \vec{x}_j \in X_j} \|\vec{x}_i - \vec{x}_j\|_{\ell_2}$$

4 Ward's linkage: this distance measures is analogous to the K-means method as it seeks to minimise the total inertia. The distance measure is the "error squared" before and after merging, which simplifies to

$$d(X_i, X_j) = \frac{|X_i||X_j|}{|X_i \cup X_j|} (\vec{\mu}_i - \vec{\mu}_j)^2$$

Hierarchical methods do not scale well. They grow as $O(N^2)$, therefore they are not suitable for large date sets. A simple but major speed-up for the method is to initialise the clusters with K-means using a large K (but a small fraction of N) and then proceed with hierarchical clustering.

20.3 Density-based clustering (DBC)

Density clustering makes the intuitive assumption that clusters are defined by regions of space with higher density of date points. Date points that constitute noise or that are outliers are expected to form regions of low density.
The core assumption of DB clustering is that a relative local density estimation of the date is possible. In other words, it is possible to order the date points according to their density. Density estimations are usually accurate for low-dimensional date points. The most widely used DBC algorithm is the DBSCAN algorithm.
Let us consider the usual unlabelled date set

$$X = \{\vec{x}_n\}_{n=1}^{N}$$

Let us define the ε-neighbour~~ood~~ of point $\vec{x}_n$ as follows:

$$N_\varepsilon(\vec{x}_n) = \{\vec{x} \in X \,/\, d(\vec{x}, \vec{x}_n) < \varepsilon\}$$

that is N_ε is the group of date points that are at a distance smaller than ε from a fixed date point $\vec{x}_n$. As before, we assume $d(\cdot,\cdot)$ to be the Euclidean metric, but any metric can be used. $N_\varepsilon(\vec{x}_n)$ can be seen as a crude estimate of the local density; $\vec{x}_n$ is considered a <u>core-point</u> if at least minPts points are in its ε-neighbourhood. minPts is a free parameter of the algorithm that sets the scale of the size of the smallest expected cluster. A point $\vec{x}_i$ is said to be <u>density-reachable</u> if it is in the ε-neighbourhood of a core-point. The algorithm can be formulated as follows

```
→ Until all points in X have been visited, do
    • Pick a point $\vec{x}_i$ that has not been visited
    • Mark $\vec{x}_i$ as a visited point
    • If $\vec{x}_i$ is a core-point
        • find the set $\mathcal{C}$ of all points that are density-
          reachable from $\vec{x}_i$
        • $\mathcal{C}$ now forms a cluster. Mark all points within
          that cluster as being visited.
→ Return the cluster assignments $\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_k$ with
```

k the number of clusters. Points that have not been assigned to a cluster are considered noise or outliers.

Advantages of DBSCAN

<u>1</u> No need to specify the number of clusters beforehand.

<u>2</u> Computational efficiency: it scales as $O(N \log N)$.

<u>Remark</u>

The K-means method requires to specify the number of clusters beforehand. A natural question is therefore how to choose the optimal number of clusters. To this purpose, one can define the SILHOUETTE COEFFICIENTS

$$sc(i) = \frac{b_i - a_i}{\max(a_i, b_i)}$$

where a_i is the average distance between points in the same cluster k and b_i is the average distance between points in a cluster k and the nearest cluster. In other words, a_i is a measure of how compact a cluster is; b_i is a measure of the dispersion of clusters.

One can show that $-1 < sc(i) < +1$. A value of $+1$ denotes a point close to all the other points in the same cluster and far from the points in other clusters ($b_i \gg a_i$) A value of 0 denotes a point close to the boundary of its cluster ($a_i \sim b_i$). A value of -1 denotes a point in the wrong cluster ($b_i \ll a_i$).

We can then define the SILHOUETTE SCORE as the mean $sc(i)$ over the points in a cluster

$$SS = \sum_{i=1}^{C} sc(i)$$

The optimal number of clusters K maximises SS.