1 K-means Clustering

Generally speaking, clustering is a task of assigning different instances to different groups. It can be applied for pattern discovery, data visualization/compression, outlier detection, etc.

A desired assignment typically has two properties
- high intra-cluster similarity, i.e., instances in a group are similar
- low inter-cluster similarity, i.e., instances in different groups are not very similar

Of course, above properties depend on how ‘similar’ is defined.

K-means is a classic clustering algorithm.
It aims to find clusters with high intra-cluster similarity.
It defines dissimilarity between instances as their Euclidean distance.\(^1\)

Specifically, suppose we partition an instance set \(S = \{x\}\) into \(K\) clusters.
Denote cluster \(k\) as \(C_k\) \((k = 1, \ldots, K)\) and denote its sample mean as
\[
\mu_k = \frac{1}{|C_k|} \sum_{x \in C_k} x.\tag{1}
\]

K-means solves the following problem
\[
\min_{C_k, \mu_k} \sum_{k=1}^{K} \sum_{x \in C_k} ||x - \mu_k||^2.\tag{2}
\]

It is easy to see (2) achieves high intra-cluster similarity, in a sense that similar instances are close to the center of their assigned clusters – these centers are sometimes call centroids.

Solving (2) is proven computationally intractable.
In practice, one solves (2) in an alternate manner i.e.
- fix cluster centroids, and update instance assignment
- fix instance assignment, and update cluster centroids

Above steps are repeated until cluster centers no longer change (convergence criterion).
This algorithm is sometimes called Lloyd’s Algorithm. Details are given in Algorithm 1.

K-means is guaranteed to converge. This is supported by two facts:
- fixing cluster centers \(\mu_k\)’s, update (3) does not increase \(L_n = \sum_{k=1}^{K} \sum_{x \in C_k} ||x - \mu_k||^2\)
- fixing cluster assignments, update (4) does not increase \(L_n\)

[Exercise] Verify above facts.

\(^1\)Thus we say \(x\) and \(z\) are similar if \(||x - z||\) is small.
**Algorithm 1** K-means Clustering Algorithm

**INPUT:** an instance set $S$, a desired number of clusters $K$

0: initialize cluster centers $\mu_1, \ldots, \mu_K$ (e.g. randomly)

**while** not converge **do**

1: assign instance $x$ to cluster $C_k$ if

$$k = \arg\min_{j=1,\ldots,K} ||x - \mu_j||.$$  \hspace{1cm} (3)

2: update cluster center $\mu_k$ by

$$\mu_k = \frac{1}{|C_k|} \sum_{x \in C_k} x.$$ \hspace{1cm} (4)

**end while**

However, K-means may converge to local minimum and is sensitive to cluster initialization.

In practice, one can run K-means multiple times and pick up an assignment with the lowest $L_n$.

In K-means, number of clusters $K$ is a hyper-parameter.

[Reading] PRML, Chapter 9.1