K means algorithm

Dror Baron

This supplement provides information about the K means algorithm.

**Objective of algorithm:** The K means algorithm iteratively partitions N training data vectors, $X_1, \ldots, X_N$, into K clusters, where cluster $k$ has a representation level $r_k$ associated with it, $r_k$, and $k \in \{1, \ldots, K\}$. The objective of the K means algorithm is to find the partition into clusters and the representation levels that minimize the aggregate square errors between the N data vectors and their associated representation levels.

**Key idea:** After K means is initialized (a typical approach selects K of the training vectors as initial cluster centers), the algorithm iterates. In each iteration, two steps are performed: (i) each training datum $X_n$ is associated with the nearest representation level and (ii) now that the set of training data associated with cluster $k$ has changed, we recompute the representation level $r_k$.

Overall, the K means algorithm is quite sensitive to its initial conditions. For example, in a research project we conducted several years ago, the data $X$ had some characteristics such that in most cases the K initial cluster centers did not span the $P$-dimensional space well, and K means using a fixed number of iterations did poorly. After we modified the initialization procedure to account for the characterists of our data, the algorithm converged more quickly, yielding clusters that captured the data better than before.

**Mathematical details:** To understand how K means works in greater detail, consider a mapping $C(n)$ that informs us for each $X_n$, $n \in \{1, \ldots, N\}$ the cluster $k = k(n) = C(n)$ that it was mapped to.

As stated previously, our objective is to minimize the aggregate square error. To analyze this aggregate error formally, we first consider the distortion or square error between $X_n$ and the representation level $r_k$ it is mapped to, where $k = C(n)$,

$$d(X_n, r_k) = \sum_{p=1}^{P} (X_{np} - r_{kp})^2 = \|X_n - r_k\|^2.$$  

That is, the distortion $d(X_n, r_k)$ turns out to be the squared Euclidean norm of the difference $X_n - r_k$ between the datum $X_n$ and corresponding representation level $r_k$.

To minimize the aggregate error, we first observe that for each cluster we want $r_k$ to minimize the sum of square errors of training data mapped to that cluster. That is,

$$r_k = \arg \min_{x \in \mathbb{R}^P} \sum_{\{n: C(n) = k\}} \|X_n - r_k\|^2.$$
It can be shown that the optimal $r_k$ that minimizes the aggregate error within the cluster is the *cluster center*,

$$r_k = \frac{1}{|\{n : C(n) = k\}|} \sum_{\{n : C(n) = k\}} X_n,$$

(1)

where we highlight that $r_k \in \mathbb{R}^P$ is obtained by averaging $|\{n : C(n) = k\}|$ training data within cluster $k$ in $P$ dimensions.

Having computed $r_k$ as the cluster center, the overall error can be interpreted as a function of the cluster mapping function $C : \{1, \ldots, N\} \rightarrow \{1, \ldots, K\}$. That is,

$$\text{Error}(C) = \sum_{n=1}^{N} \|X_n - r_{k(n)}\|^2$$

(2)

$$= \sum_{k=1}^{K} \sum_{\{n : C(n) = k\}} \|X_n - r_k\|^2,$$

(3)

where (2) sums over $N$ square error terms, (3) is obtained by partitioning the $N$ previous terms into $K$ sub-summations based on the corresponding clusters, and we highlight that $r_k$ is a function of the set $\{n : C(n) = k\}$. That is, this set is determined by the mapping function $C$, and the optimal $r_k$ is given by the cluster center (1).