# Clustering : *k*-means and *k*-medoids

Xian-Hong Wang (王先弘)

Department of Mathematics, National Central University
Jhongli District, Taoyuan City 32001, Taiwan

Revised on November 27, 2021

# Introduction

A common task in data science is to search in a given data set for indications of internal structure that can be further explored, for example, an intrinsic organization of data into clusters.

## Clursting Problem

We start by formulating the basic clustering problem: Given a set of vectors,

$$\mathscr{D} = \{x^{(1)}, x^{(2)}, \ldots, x^{(p)}\}$$

arrange them into $k$ distinct subsets, or clusters.

$$\mathscr{D}_\ell = \{x^{(j)} : j \in \boldsymbol{I}_\ell\}, \ell = 1, \ldots, k$$

where

$$\mathbf{I}_i \cap \boldsymbol{I}_j = \emptyset, \ \forall i \neq j$$

and

$$\bigcup_{\ell=1}^{k} \boldsymbol{I}_\ell = \{1, 2, \ldots, p\}.$$

## The $k$-means algorithm

Assume that the data set consists of real vectors.

$$\mathscr{D} = \{x^{(1)}, x^{(2)}, \ldots, x^{(p)}\}, x^{(j)} \in \mathbb{R}^n$$

that have been partitioned according to some unspecified criterion into $k$ subsets,

$$\mathscr{D}_\ell = \{x^{(j)} : j \in \boldsymbol{I}_\ell\}$$

where $\boldsymbol{I}_\ell$is the index set corresponding to partitioning.
We introduce the notation

$$\boldsymbol{II} = \{\boldsymbol{I}_1, \boldsymbol{I}_2, \ldots, \boldsymbol{I}_k\}$$

and called $\boldsymbol{II}$ a *partitoning* of the data.

## The $k$-means algorithm

For each cluster $\mathscr{D}_\ell$, we define an associated characteristic vector $c^{(\ell)} \in \mathbb{R}^n$ that represents the cluster.
The characteristic vector of the cluster makes it possible to define the *within-cluster tightness* or *within-cluster coherence* of $\mathscr{D}_\ell$ according to the formula

$$q_\ell = q_\ell(c^{(\ell)}) = \sum_{j \in \boldsymbol{I}_\ell} \left\| x^{(j)} - c^{(\ell)} \right\|^2$$

where the distance between the data point and characteristic vector is measured in Euclidean norm.
The quantity of $q_\ell$ is sometimes referred to as the within-cluster sum of squares(WCSS). Initially, a small WCSS should be taken as an indication of a tight cluster for which the characteristic vector is a good representative of each cluster member.

# The $k$-means algorithm

The *overall tightness*, or *overall coherence*, of the clustering $\boldsymbol{II}$ is measured by the quantity

$$Q\big(\boldsymbol{II}, c^{(1)}, c^{(2)}, \dots, c^{(k)}\big) = \sum_{\ell=1}^{k} q_\ell$$

The $k$-means algorithm, also knowns as *Lloyd's* algorithm, is an iterative procedure searching for an optimal clustering $\boldsymbol{II}_{opt}$ and corresponding characteristic vectors $c_{opt}^{(1)}, c_{opt}^{(2)}, \dots, c_{opt}^{(k)}$ such that

$$Q\big(\boldsymbol{II}_{opt}, c_{opt}^{(1)}, c_{opt}^{(2)}, \dots, c_{opt}^{(k)}\big) = \min Q\big(\boldsymbol{II}, c^{(1)}, c^{(2)}, \dots, c^{(k)}\big)$$

where the minimum is taken over all *partitionings* of the data and all characteristic vectors representing the clusters.

# An alternating minimization scheme

The $k$-means algorithm is based on the following alternating optimization scheme:

1. ***Updating step*** **:** Given the current partitioning $\boldsymbol{II}_t$, update the characteristic vectors

$$Q(\boldsymbol{II}_t, c_{t+1}^{(1)}, c_{t+1}^{(2)}, \ldots, c_{t+1}^{(k)}) = \min Q(\boldsymbol{II}, c^{(1)}, c^{(2)}, \ldots, c^{(k)})$$

   where the minimization is over the vectors $c^{(\ell)}$.

2. ***Assignment step*** **:** Given the updated characteristic vectors $c_{t+1}^{(1)}, c_{t+1}^{(2)}, \ldots, c_{t+1}^{(k)}$, update the partitioning,

$$Q(\boldsymbol{II}_{t+1}, c_{t+1}^{(1)}, c_{t+1}^{(2)}, \ldots, c_{t+1}^{(k)}) = \min Q(\boldsymbol{II}_t, c_{t+1}^{(1)}, c_{t+1}^{(2)}, \ldots, c_{t+1}^{(k)}),$$

   minimizing over different partitionings into $k$ clusters.

## An alternating minimization scheme : Updating step

The mapping

$$\mathbb{R}^n \to \mathbb{R},\ c^{(\ell)} \mapsto Q\big(\boldsymbol{II}, c^{(1)}, c^{(2)}, \ldots, c^{(\ell)}, \ldots, c^{(k)}\big),$$

where $c^{(1)}, c^{(2)}, \ldots, c^{(\ell-1)}, c^{(\ell+1)}, \ldots, c^{(k)}$, is differential, so a minimizer must be a critical point for the function.In light of the fact that only $q_{(\ell)}$ depends on $c^{(\ell)}$, the critical points with respect to $c^{(\ell)}$ are found by solving the equation

$$\nabla_{c^{(\ell)}} Q\big(\boldsymbol{II}, c^{(1)}, c^{(2)}, \ldots, c^{(k)}\big) = \nabla_{c^{(\ell)}} q_\ell\big(c^{(\ell)}\big) = 0,$$

which can be written componentwise as

$$\begin{aligned}\frac{\partial}{\partial c_i^{(\ell)}} \sum_{j \in \boldsymbol{I}_\ell} \sum_{q=1}^{n} \big(x_q^{(j)} - c_q^{(\ell)}\big)^2 &= -2 \sum_{j \in \boldsymbol{I}_\ell} \big(x_i^{(j)} - c_i^{(\ell)}\big) \\ &= -2\big(\sum_{j \in \boldsymbol{I}_\ell} x_i^{(j)} - |\mathscr{D}_\ell| c_i^{(\ell)}\big), 1 \le i \le n.\end{aligned}$$

# An alternating minimization scheme : Updating step

where $|\mathscr{D}_\ell|$ denotes the cardinality of the cluster $\mathscr{D}_\ell$, that is, the number of index in the set $\boldsymbol{I}_\ell$.
Therefore, from condition gives

$$c^{(\ell)} = \frac{1}{|\mathscr{D}_\ell|} \sum_{j \in \boldsymbol{I}_\ell} x^{(j)}.$$

since the coherence, which is quadratic function of $c^{(\ell)}$, increases without bounded as $c^{(\ell)}$ goes to infinity, its unique critical point must correspond to the global minimum.

## An alternating minimization scheme : Assignment step

Consider now the minimization with respect to the partitioning, keeping the characteristic vectors $c^{(\ell)}$ fixed.
Given $x^{(j)} \in \mathscr{D}_\ell$, where $\mathscr{D}_\ell$ refers to the current partitioning, if

$$\|x^{(j)} - c^{(m)}\| < \|x^{(j)} - c^{(\ell)}\|, \textit{ for some } m \neq \ell,$$

then reassigning $x^{(j)}$ to $\mathscr{D}_m$ will decrease the overall tightness.

## *k*-means algorithm(Lloyd's algorithm)

```
**Given** the number of clusters $k = 2, 3, \dots,$ and a tolerance $\tau > 0$.
**Initialize:** Assign a partitioning $\boldsymbol{II}_0$ , $\Delta Q = \infty$.
**while** $\Delta Q > \tau$ **do**
    **Updating step:** For each $\boldsymbol{I}_\ell$ in $\boldsymbol{II}_t$ , compute the cluster centroid
        $$c_{t+1}^{(\ell)} = \frac{1}{|\mathscr{D}_\ell|} \sum_{j \in \boldsymbol{I}_\ell} x^{(j)}.$$
    **Assignment step:** For each $x^{(j)}$, find the closest cluster centroid, and assign $x^{(j)}$ to the corresponding cluster.This defines the new partitioning $\boldsymbol{II}_{t+1}$.
    Update
        $$\Delta Q = \left|Q\big(\boldsymbol{II}_t, c_t^{(1)}, \dots, c_t^{(j)}\big) - Q\big(\boldsymbol{II}_{t+1}, c_{t+1}^{(1)}, \dots, c_{t+1}^{(j)},\big)\right|,$$
    and advance the counter $t \to t+1$.
**end while**
```

## An alternating minimization scheme

The assignment step in the $k$-means algorithm can be interpreted in geometric terms as follows. The $k$-means algorithm induces a *Voronoi tessellation* of the data space $\mathbb{R}^n$. Given the cluster centroids $c^{(1)}, c^{(2)}, \ldots, c^{(k)}$, the $\ell$-th Voronoi set $V_\ell$ is defined by

$$\begin{aligned} V_\ell &= \{x \in \mathbb{R}^n \mid \|x - c^{(\ell)}\| < \|x - c^{(j)}\| \ \ \textit{for all } j, \quad j \neq \ell\} \\ &= \bigcap_{j \neq \ell} \{x \in \mathbb{R}^n \mid \|x - c^{(\ell)}\| < \|x - c^{(j)}\|\}. \end{aligned}$$

Hence, the $k$-means algorithm can be seen as a method to implicitly subdivide the data space into $k$ Voronoi sets, and assign each data point to a cluster according to which Voronoi set the point belongs to, without explicitly computing the tessellation.

# Initial partitioning

1. *Random initialization:* Divide the index set $\boldsymbol{I} = \{1, 2, \ldots, p\}$ into $k$ random, nonempty subsets. One possibility is to first randomly permute the elements of $\boldsymbol{I}$ and then divide the permuted index vector into $k$ subvectors of roughly equal size.
2. *Partitioning by hyperplanes:* Subdivide the data space using hyperplanes. The simplest example is for $k = 2$ , when a hyperplane can be used to split the data in two initial clusters.
3. *Initial seed partitioning:* Select $k$ points from the data set, $x^{(j_1)}, x^{(j_2)}, \ldots, x^{(j_k)}$. These *seed vectors* can be chosen either randomly or, if data indicate the presence of clusters, e.g., in a PCA plot, the choice can be one representative for each presumed cluster. Subsequently, partition the data space $\mathbb{R}^n$ into $k$ Voronoi sets $V_j$ defined by seed vectors. The initial clusters are chosen as $\boldsymbol{I}_\ell = \{j \in \boldsymbol{I} \mid x^{(j)} \in V_\ell\}, 1 \le \ell \le k$.

# The $k$-medoids algorithm

Denote the distance $d$ between the $i$th and $j$th data points as

$$d_{ij} = d\left(x^{(i)}, x^{(j)}\right), 1 \le i, j \le p,$$

where $d$ is a metric, that is, a binary from $d : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}_0^+$ such that, for any $x, y, z \in \mathbb{R}^n$,

1. $d(x, x) = 0,$
2. $d(x, y) \ge 0,$
3. $d(x, y) = d(y, x),$
4. $d(x, y) \le d(x, z) + d(z, y).$

Clearly, every norm $\| \cdot \|$ on $\mathbb{R}^n$ defines the metric $d(x, y) = \|x - y\|$.

# The $k$-medoids algorithm

Assume that a partitioning $\boldsymbol{II}$ of the data into $k$ clusters is given, and associated with each $\mathscr{D}_\ell$ a characteristic vector *belonging to the cluster*, that is,

$$c^{(\ell)} \in \mathscr{D}_\ell, 1 \le \ell \le k.$$

Following what we did for $k$-means algorithm, we define the within-cluster tightness $q_\ell\big(c^{(\ell)}\big)$ and the overall tightness $Q\big(\boldsymbol{II}, c^{(1)}, \ldots, c^{(k)}\big)$ according to the formulas

$$Q\big(\boldsymbol{II}, c^{(1)}, \ldots, c^{(k)}\big) = \sum_{\ell=1}^{k} q_\ell\big(c^{(\ell)}\big), \ \ \textit{where}\, q_\ell\big(c^{(\ell)}\big) = \sum_{j \in \boldsymbol{I}_\ell} d\big(x^{(j)}, c^{(\ell)}\big).$$

The setting is very similar to the derivation of the $k$-means algorithm, with the additional requirement that characteristic vectors must belong themselves to the clusters that represent.

# $k$-medoids algorithm(partitioning around medoids)

**1 . Given** the number of clusters $k$ , a metric $d$ , and a tolerance $\tau > 0$.
**2 . Initialize :** Specify the initial set of characteristic vectors, or medoids,

$$\{c_0^{(1)}, c_0^{(2)}, \ldots, c_0^{(k)}\}, \;\; c_0^{(\ell)} \in \mathscr{D}.$$

Set $t = 0$, $\Delta Q = \infty$.
**3 . Iteration : While** $\Delta Q > \tau$ **:**
**Assignment step :** For each $x^{(j)}$, $1 \le j \le p$, find the nearest medoid,

$$d\big(x^{(j)}, c^{(\ell_*)}\big) = \min_{1 \le \ell \le k} d\big(x^{(j)}, c^{(\ell)}\big),$$

and assign $x^{(j)}$ to the cluster $\mathscr{D}_{\ell_*}$.
This process determines a partitioning $\boldsymbol{II}_{t+1}$ into $k$ clusters $\mathscr{D}_1, \ldots, \mathscr{D}_k$.

## $k$-medoids algorithm(partitioning around medoids)

**Updating step :** For each cluster $\mathscr{D}_\ell$, $1 \le \ell \le k$, calculate the within-cluster tightness

$$q_\ell(x^{(j)}) \ \textit{for all } x^{(j)} \in \mathscr{D}_\ell,$$

and select the medoid for which the tightness is smallest,

$$c_{t+1}^{(\ell)} = \arg\min\{q_\ell(x^{(j)}) \mid x^{(j)} \in \mathscr{D}_\ell\}.$$

Calculate the overall tightness,

$$Q_{t+1} = Q\big(\boldsymbol{I\!I}_{t+1}, c_{t+1}^{(1)}, \ldots, c_{t+1}^{(k)}\big).$$

If $t > 1$, update

$$\Delta Q = |Q_t - Q_{t+1}|,$$

and advance the counter by one, $t \to t+1$.

## *k*-means or *k*-medoids ?

While the idea behind *k*-means may be very natural and well in line with a mathematical frame of mind, among the arguments in support of *k*-medoids are the flexibility with respect to how distance between data points is measured and lower sensitivity to presence of cluster outliers that may have a strong adverse effect on the performance of the *k*-means algorithm.

## How to choose $k$ in the algorithm

Consider a data set of $p$ vectors, on which either $k$-means or $k$-medoids clustering is applied with a give $k$, and let the cluster centroids or medoids be $c^{(1)}, c^{(2)}, \dots, c^{(k)}$. The within-cluster mean distance (WCMD) is the average distance of each point from its respective cluster centroid, that is,

$$WCMD_k = \frac{1}{p} \sum_{\ell=1}^{k} \sum_{j \in \boldsymbol{I}_\ell} d\left(x^{(j)}, c^{(\ell)}\right).$$

# References

1. *Daniela Calvetti, Erkki Somersalo*, Clustering: K-means and K-medoids. *Mathematics of Data Science: A Computational Approach to Clustering and Classification*.