k-Means and k-Median

Frederik Mallmann-Trenn
6CCS3AIN

k-Means

- Say we have n points $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$.
- We want to partition them into k sets $S_1, S_2, \ldots, S_k$ such that the cost of the partition, $c(S_1, S_2, \ldots, S_k)$, is minimised:

$$c(S_1, S_2, \ldots, S_k) = \sum_{i=1}^{n} \left(\min_{j \in [k]} d(\mathbf{x}_i, \boldsymbol{\mu}_j) \right)^2,$$

where $\boldsymbol{\mu}_i$ is the mid-point of each cluster, i.e.,

$$\boldsymbol{\mu}_i = \frac{1}{|S_i|} \sum_{j \in S_i} \mathbf{x}_j$$

k-Means

- Say we have n points $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n$.
- We want to partition them into k sets $S_1, S_2, \dots, S_k$ such that the cost of the partition, $c(S_1, S_2, \dots, S_k)$, is minimised:

$$c(S_1, S_2, \dots, S_k) = \sum_{i=1}^{n} \left(\min_{j \in [k]} d(\mathbf{x}_i, \boldsymbol{\mu}_j) \right)^2,$$

where $\boldsymbol{\mu}_i$ is the mid-point of each cluster, i.e.,

$$\boldsymbol{\mu}_i = \frac{1}{|S_i|} \sum_{j \in S_i} \mathbf{x}_j$$

- Note that $\mathbf{x}_i$ and $\boldsymbol{\mu}_i$ are vectors for $i \in [n]$.

k-Means

- Say we have n points $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$.
- We want to partition them into k sets $S_1, S_2, \ldots, S_k$ such that the cost of the partition, $c(S_1, S_2, \ldots, S_k)$, is minimised:

$$c(S_1, S_2, \ldots, S_k) = \sum_{i=1}^{n} \left(\min_{j \in [k]} d(\mathbf{x}_i, \boldsymbol{\mu}_j) \right)^2,$$

where $\boldsymbol{\mu}_i$ is the mid-point of each cluster, i.e.,

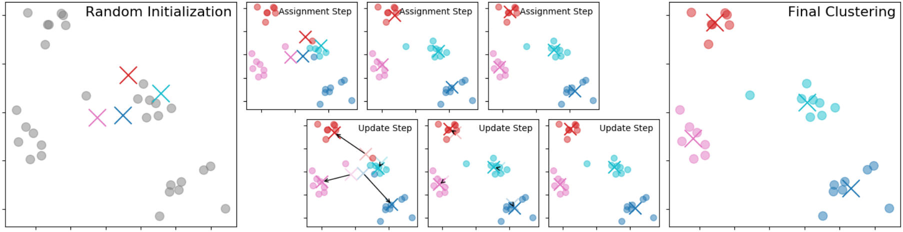
$$\boldsymbol{\mu}_i = \frac{1}{|S_i|} \sum_{j \in S_i} \mathbf{x}_j$$

- Note that $\mathbf{x}_i$ and $\boldsymbol{\mu}_i$ are vectors for $i \in [n]$.
- E.g., if $S_1 = \{\mathbf{x}_1, \mathbf{x}_2\}$ with $\mathbf{x}_1 = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$ and $\mathbf{x}_2 = \begin{pmatrix} 3 \\ 4 \end{pmatrix}$, then $\boldsymbol{\mu}_1 = \begin{pmatrix} 2 \\ 3 \end{pmatrix}$

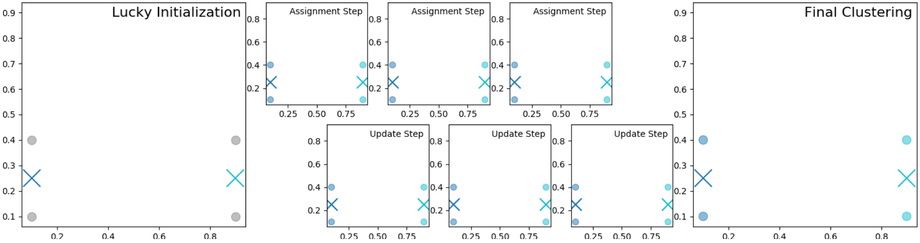
k-Means

1. Select k cluster centres arbitrarily.
2. Repeat until convergence:
   2.1 Assignment Step:
      2.1.1 Assign each point to the cluster with the nearest mean
      2.1.2 $S_i = \{\mathbf{x}_j \mid d(\mathbf{x}_j, \boldsymbol{\mu}_i) \leqslant d(\mathbf{x}_k, \boldsymbol{\mu}_i)$ for all $\ell \in [k]\}$, where each point is assigned to exactly one cluster S_i.
   2.2 Update Step:
      2.2.1 Recalculate the mean point of the cluster
      2.2.2 $\boldsymbol{\mu}_i = \frac{1}{|S_i|} \sum_{j \in S_i} \mathbf{x}_j$

k-Means

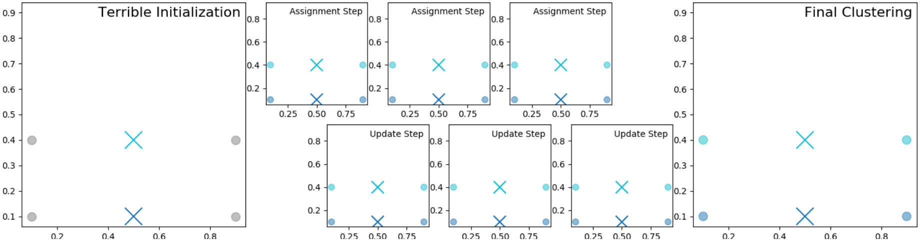

Random Initialization
Assignment Step
Assignment Step
Assignment Step
Update Step
Update Step
Update Step
Final Clustering

k-Means: Example with Optimal Instance



- Consider this example with four points.
- The optimal cluster is shown.

k-Means



- We can see that if we start with sub-optimal clusters, and we never change them!
- This can be made arbitrarily bad (by increasing the width of the rectangle).

k-Means++

- The way this can be solved is by using k-Means++
- It can be shown that the approximation factor is at most $O(\log k)$.

k-Means++

1. Set the first centre to be one of the input points chosen uniformly at random, i.e., $\boldsymbol{\mu}_1 = uniform(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n)$
2. For cluster $i = 2$ to k:
   2.1 For each point $\mathbf{x}_j$ compute the distance to the nearest centre, i.e., calculate $d_j = \min_\ell d(\mathbf{x}_j, \boldsymbol{\mu}_\ell)$
   2.2 Open a new centre at a point using the weighted probability distribution that is proportional to d_j^2. That is,

$$\Pr(\text{new centre in } \mathbf{x}_j) = \frac{d_j^2}{\sum_\ell d_\ell^2}$$

3. Continue with k-Means

k-Median

- Say we have n points $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$.
- We want to partition them into k sets $S_1, S_2, \ldots, S_k$ such that the cost of the partition, $c(S_1, S_2, \ldots, S_k)$, is minimised:

$$c(S_1, S_2, \ldots, S_k) = \sum_{i=1}^{n} \left(\min_{j \in [k]} d(\mathbf{x}_i, \boldsymbol{\mu}_j) \right),$$

where $\boldsymbol{\mu}_i$ is the mid-point of each cluster, i.e.,

$$\boldsymbol{\mu}_i = \frac{1}{|S_i|} \sum_{j \in S_i} \mathbf{x}_j$$

k-Median

- Say we have n points $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n$.
- We want to partition them into k sets $S_1, S_2, \dots, S_k$ such that the cost of the partition, $c(S_1, S_2, \dots, S_k)$, is minimised:

$$c(S_1, S_2, \dots, S_k) = \sum_{i=1}^{n} \left(\min_{j \in [k]} d(\mathbf{x}_i, \boldsymbol{\mu}_j) \right),$$

where $\boldsymbol{\mu}_i$ is the mid-point of each cluster, i.e.,

$$\boldsymbol{\mu}_i = \frac{1}{|S_i|} \sum_{j \in S_i} \mathbf{x}_j$$

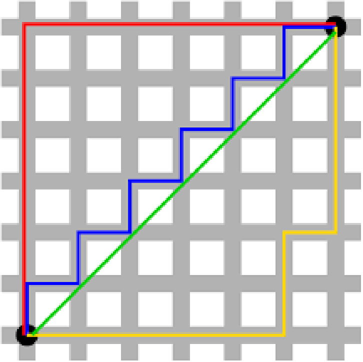
- Recall that k-Means uses

$$\sum_{i=1}^{n} \left(\min_{j \in [k]} d(\mathbf{x}_i, \boldsymbol{\mu}_j) \right)^2$$

-

k-Median

- K-Means minimises the Euclidean/geometric distance
- K-Medians minimises the Manhattan distance



-

- Source: Wikipedia