DDA3020 Machine Learning: Lecture 15 K-means Clustering

Jicong Fan
School of Data Science, CUHK-SZ

13/11/2025

Outline

1. K-means Clustering
   - Definition
   - Demo and algorithm
   - Optimization perspective
   - Variants of K-means (optional)

2. Performance Evaluation of Clustering

3. References of other clustering algorithms

Definition of K-means Clustering

- K-means clustering is a method of vector quantization, originally from signal processing, that aims to partition n observations/samples into k clusters in which each observation belongs to the cluster with the nearest mean (cluster centers or cluster centroid), serving as a prototype of the cluster.
- K-means clustering minimizes within-cluster variances (squared Euclidean distances).

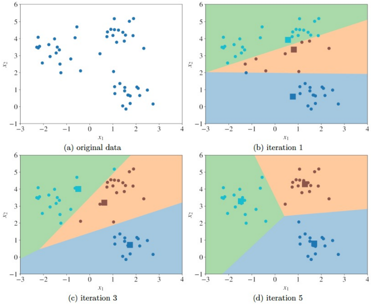
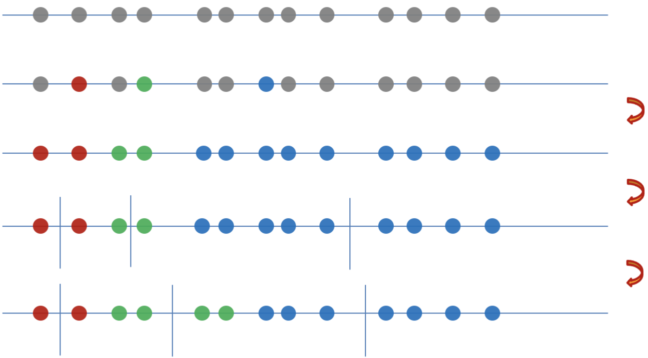


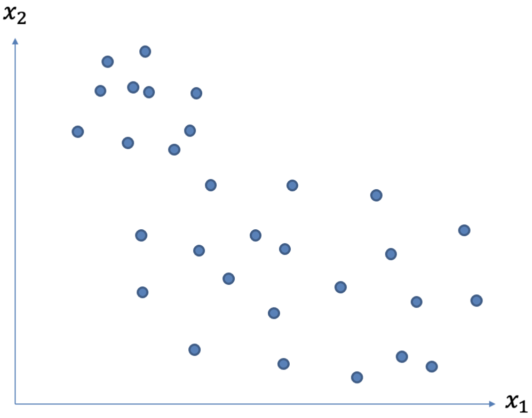
Figure 2: The progress of the k-means algorithm for $k = 3$.

References:
https://en.wikipedia.org/wiki/K-means_clustering
https://en.wikipedia.org/wiki/Vector_quantization

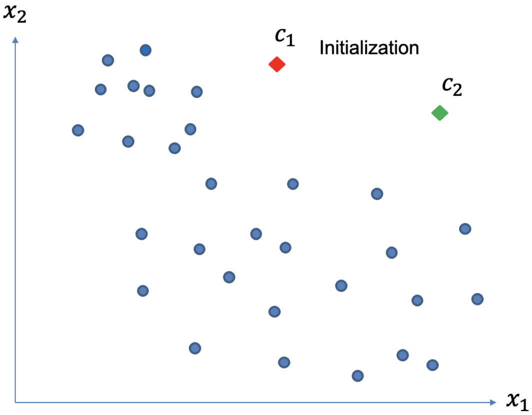
K-means Clustering (1 D)



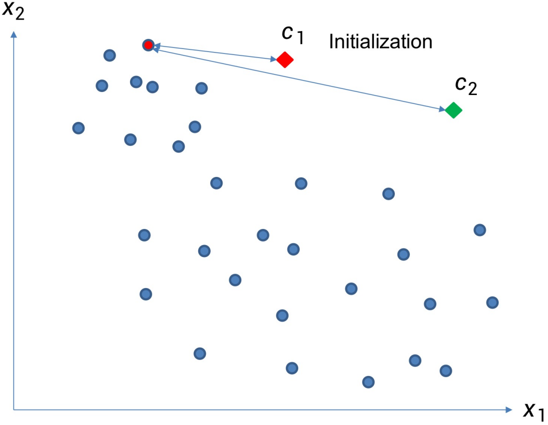
K-means Clustering (2 D)



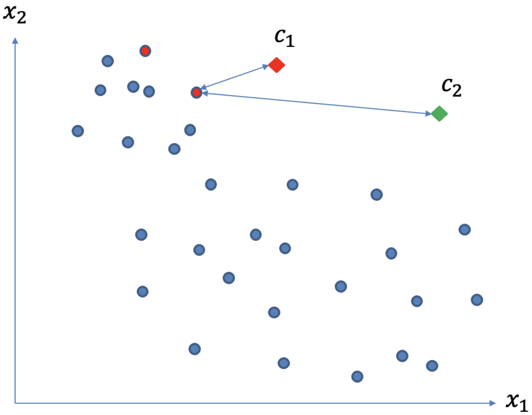
K-means Clustering



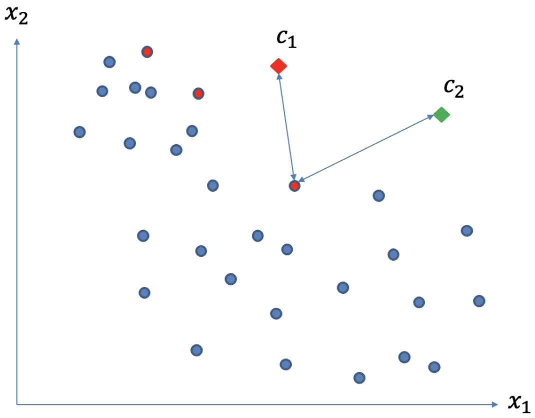
K-means Clustering



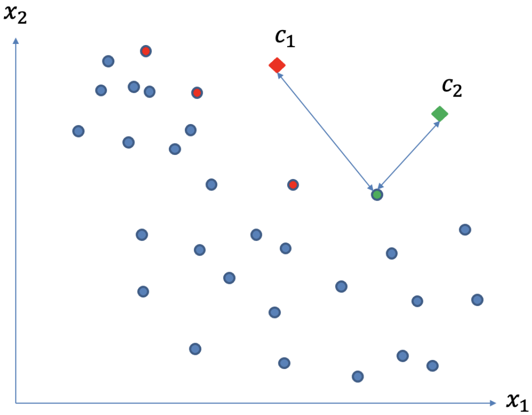
K-means Clustering



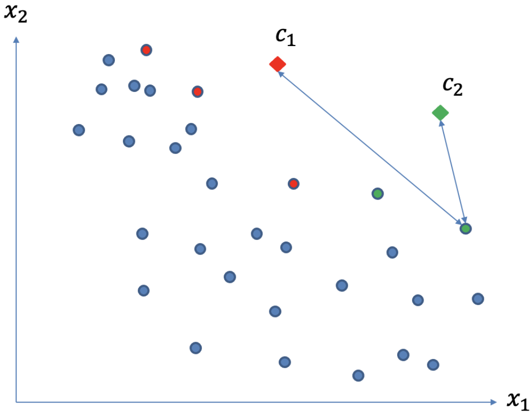
K-means Clustering



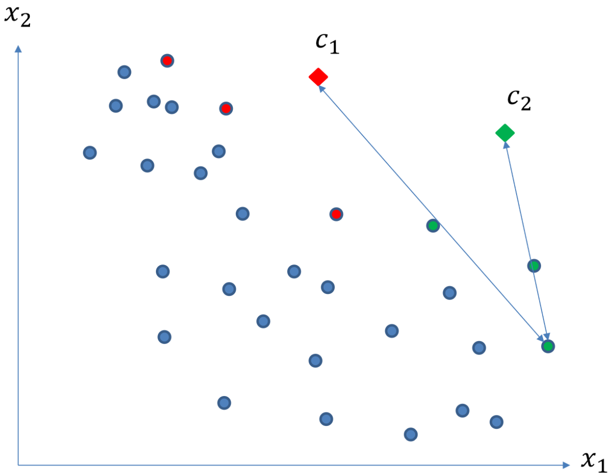
K-means Clustering



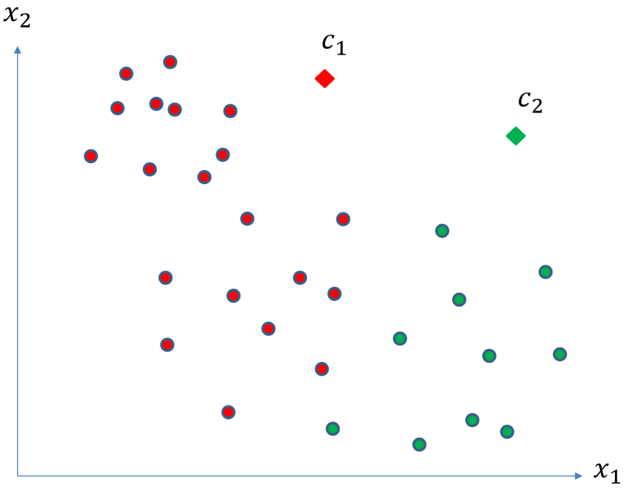
K-means Clustering



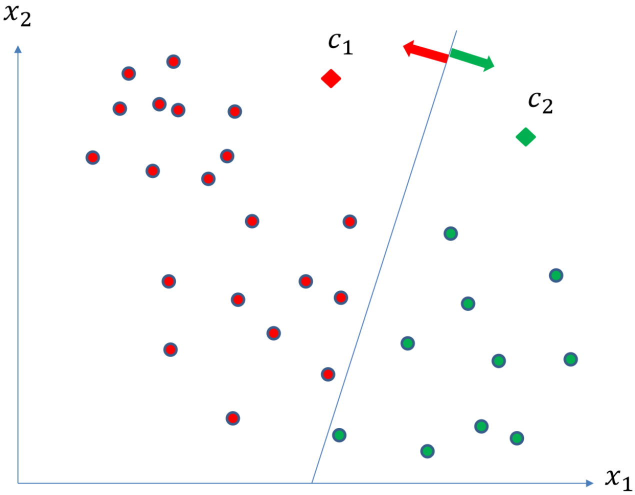
K-means Clustering



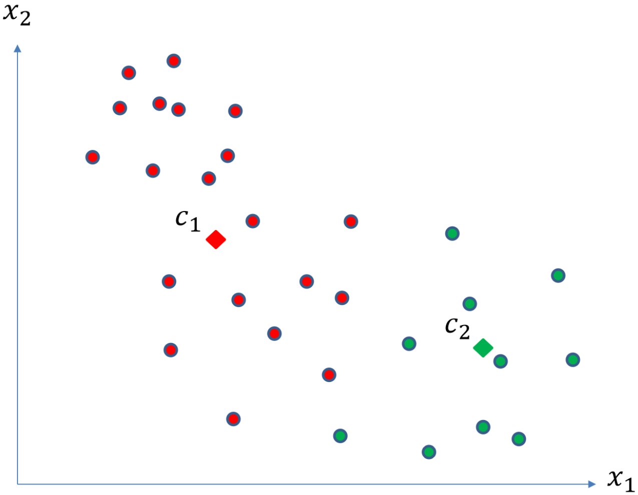
K-means Clustering



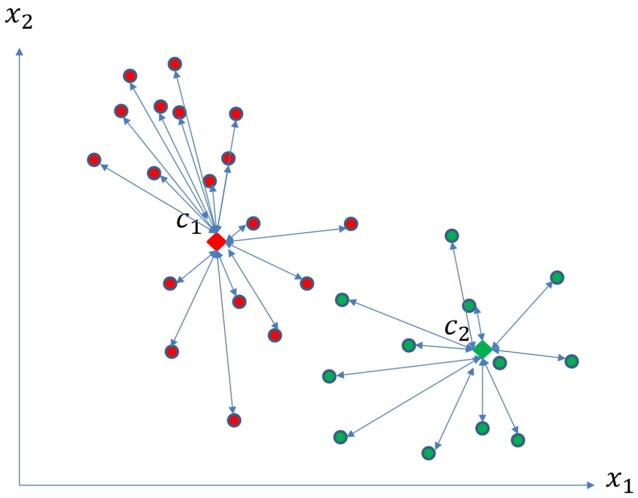
K-means Clustering



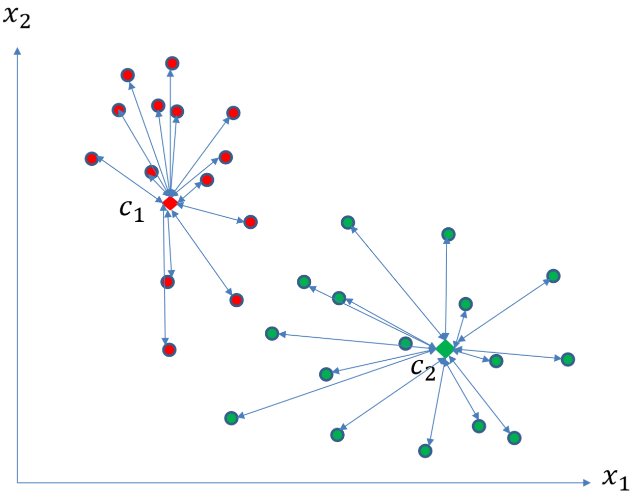
K-means Clustering



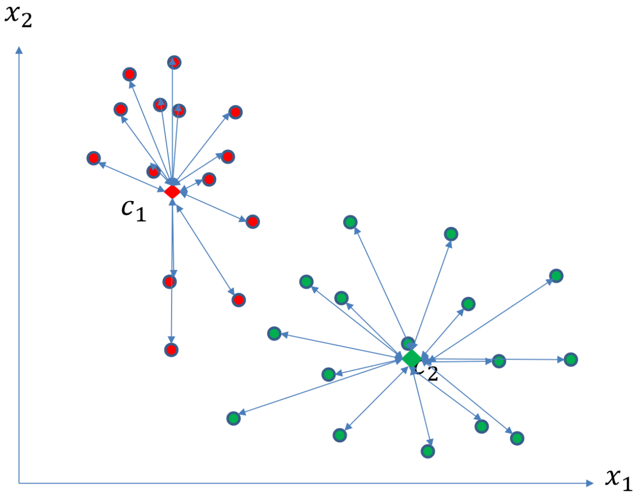
K-means Clustering



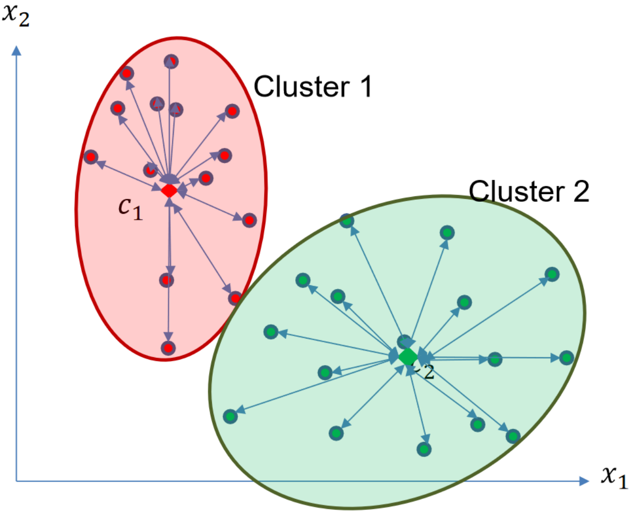
K-means Clustering



K-means Clustering



K-means Clustering



K-means Clustering

Let's see an online demo:
https://user.ceng.metu.edu.tr/~akifakkus/courses/ceng574/k-means/

K-means Clustering

Basic K-means Clustering

1. First, you choose K — the number of clusters. Then you randomly put K feature vectors, called **centroids**, to the feature space.
2. Next, compute the distance from each example $\mathbf{x}$ to each centroid $\mathbf{c}$ using some metric, like the Euclidean distance. Then we assign the closest centroid to each example (like if we labeled each example with a centroid id as the label).
3. For each centroid, we calculate the average feature vector of the examples labeled with it. These average feature vectors become the new locations of the centroids.
4. We recompute the distance from each example to each centroid, modify the assignment and repeat the procedure until the assignments don't change after the centroid locations are recomputed.
5. Finally we conclude the clustering with a list of assignments of centroids IDs to the examples.

Optimization perspective of K-means Clustering

- What is actually being optimized by the basic K-means clustering algorithm?
- Given the data set $\{\mathbf{x}_i\}_{i=1}^n$, K-means aims to find cluster centers $\mathbf{c} = \{\mathbf{c}_j\}_{j=1}^K$ and assignments $\mathbf{r}$, by minimizing the sum of squared distances of data points to their assigned cluster centers. In short, K-means will minimize the within-cluster variance, as follows:

$$\min_{\mathbf{c},\mathbf{r}} J(\mathbf{c},\mathbf{r}) = \min_{\mathbf{c},\mathbf{r}} \sum_{i}^{n} \sum_{k}^{K} r_{ik} \|\mathbf{x}_i - \mathbf{c}_k\|^2,$$

$$\text{Subject to } \ \mathbf{r} \in \{0,1\}^{n \times K}, \ \sum_{k}^{K} r_{ik} = 1,$$

where $r_{ik} = 1$ denotes $\mathbf{x}_i$ is assigned to cluster k.

Optimization perspective of K-means Clustering

$$\min_{\mathbf{c},\mathbf{r}} J(\mathbf{c},\mathbf{r}) = \min_{\mathbf{c},\mathbf{r}} \sum_{i}^{n} \sum_{k}^{K} r_{ik} \|\mathbf{x}_i - \mathbf{c}_k\|^2,$$

$$\text{Subject to } \ \mathbf{r} \in \{0,1\}^{n\times K}, \ \sum_{k}^{K} r_{ik} = 1,$$

The above problem can be solved by coordinate descent algorithm, *i.e.*, update $\mathbf{c}$ and $\mathbf{r}$ alternatively:

- Given the cluster centers $\mathbf{c}$, update the assignments $\mathbf{r}$
- Given the assignments $\mathbf{r}$, update the cluster centers $\mathbf{c}$

Optimization perspective of K-means Clustering

Optimization of K-means clustering:

- Initialization: set K cluster centers $\mathbf{c}$ to random values
- Repeat until convergence (the assignments don't change):
  - Assignment: Given the cluster centers $\mathbf{c}$, update the assignments $\mathbf{r}$ by solving the following sub-problem

$$\min_{\mathbf{r}} \sum_{i}^{n} \sum_{k}^{K} r_{ik} \|\mathbf{x}_i - \mathbf{c}_k\|^2, \quad \text{subject to} \quad \mathbf{r} \in \{0,1\}^{n \times K}, \ \sum_{k}^{K} r_{ik} = 1.$$

    Note that the assignment for each data $\mathbf{x}_i$ can be solved independently. It is easy to know that assigning $\mathbf{x}_i$ to the closest cluster is the optimal solution.
  - Refitting: Given the assignments $\mathbf{r}$, update the cluster centers $\mathbf{c}$:

$$\min_{\mathbf{c}} \sum_{i}^{n} \sum_{k}^{K} r_{ik} \|\mathbf{x}_i - \mathbf{c}_k\|^2.$$

    Note that $\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_K$ can be optimized independently. By setting the derivative *w.r.t.* $\mathbf{c}_k$ as 0, it is easy to obtain the optimal solution:

$$\mathbf{c}_k = \frac{\sum_i^n r_{ik} \mathbf{x}_i}{\sum_i^n r_{ik}}.$$

Optimization perspective of K-means Clustering

Assignment:

- Given the cluster centers $\mathbf{c}$, update the assignments $\mathbf{r}$ by solving the following sub-problem

$$\min_{\mathbf{r}} \sum_{i}^{n} \sum_{k}^{K} r_{ik} \|\mathbf{x}_i - \mathbf{c}_k\|^2, \quad \text{subject to} \quad \mathbf{r} \in \{0,1\}^{n \times K}, \ \sum_{k}^{K} r_{ik} = 1.$$

- Note that the assignment for each data $\mathbf{x}_i$ can be solved independently, *i.e.*,

$$\min_{\mathbf{r}_i} \sum_{k}^{K} r_{ik} \|\mathbf{x}_i - \mathbf{c}_k\|^2, \quad \text{subject to} \quad \mathbf{r}_i \in \{0,1\}^{1 \times K}, \ \sum_{k}^{K} r_{ik} = 1.$$

- It is easy to obtain the solution as follows

$$k^* = \arg \min_{1 \le k \le K} \|\mathbf{x}_i - \mathbf{c}_k\|^2, \text{ and } r_{ik^*} = 1.$$

- Thus, we assign $\mathbf{x}_i$ to the closest cluster, exactly same with the assignment step in basic K-means algorithm.

Optimization perspective of K-means Clustering

Refitting:

- Given the assignments $\mathbf{r}$, update the cluster centers $\mathbf{c}$:

$$\min_{\mathbf{c}} \sum_{i}^{n} \sum_{k}^{K} r_{ik} \|\mathbf{x}_i - \mathbf{c}_k\|^2.$$

- Note that $\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_K$ can be optimized independently, as follows

$$\min_{\mathbf{c}_k} \sum_{i}^{n} r_{ik} \|\mathbf{x}_i - \mathbf{c}_k\|^2.$$

- By setting the derivative *w.r.t.* $\mathbf{c}_k$ as 0, *i.e.*,

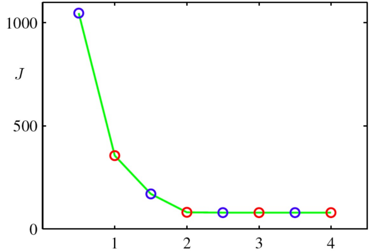
$$\sum_{i}^{n} 2r_{ik} \|\mathbf{x}_i - \mathbf{c}_k\| = 0 \;\Rightarrow\; \mathbf{c}_k = \frac{\sum_i^n r_{ik}\mathbf{x}_i}{\sum_i^n r_{ik}},$$

  where $\sum_i^n r_{ik}$ denotes the number of samples assigned to the kth cluster, and $\sum_i^n r_{ik}\mathbf{x}_i$ is the summation of all samples of the kth cluster.
- Thus, $\mathbf{c}_k$ is the center of the kth cluster, which is exactly same with the step of calculating the cluster center in basic K-means clustering.

Optimization perspective of K-means Clustering

Why does K-means converge?

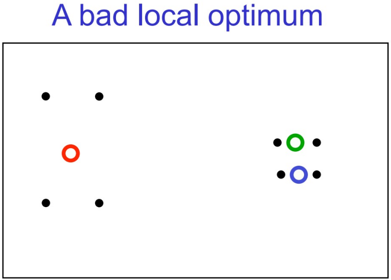
- Convergence guarantee:
  - Whenever an assignment is changed, the sum squared distances J of data points from their assigned cluster centers is reduced.
  - Whenever a cluster center is moved, J is reduced.
- Test for convergence: If the assignments do not change in the assignment step, we have converged (to at least a local minimum).
- Example: As shown below, the objective function of K-means is reduced after each assignment step (blue) and refitting step (red). The algorithm has converged after the third refitting step.



Optimization perspective of K-means Clustering

Local minimum of K-means

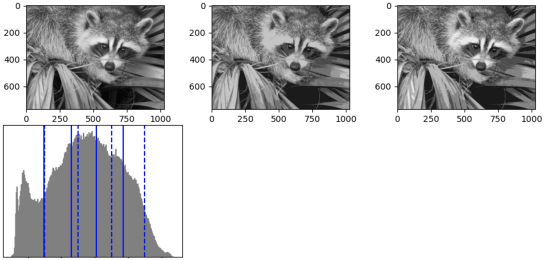
- Since the objective function J is non-convex, the coordinate descent on J is not guaranteed to converge to the global minimum
- There is nothing to prevent k-means from getting stuck at a local minimum, and sometimes it may get stuck at a poor local minimum (shown below)
- What we could do is running K-means with multiple random initializations, and picking the one with the lowest objective value as the final clustering result



Example of K-means Clustering

Example: K-means for vector quantization

- Vector quantization is a classical quantization technique from signal processing
- It works by dividing a large set of points (vectors) into groups having approximately the same number of points closest to them. Each group is represented by its centroid point, as in k-means
- As shown below, vector quantization is used for compressing image



Demo with code:
https://scikit-learn.org/stable/auto_examples/cluster/plot_face_comp
html#sphx-glr-auto-examples-cluster-plot-face-compress-py

Variants of K-means (optional)

- Fuzzy C-means
  - Reference: https://www.sciencedirect.com/science/article/abs/pii/0098300484900207
  - Code: https://pypi.org/project/fuzzy-c-means/
- Constrained K-means
  - Reference: https://web.cse.msu.edu/~cse802/notes/ConstrainedKmeans.pdf
  - Code: https://github.com/Behrouz-Babaki/COP-Kmeans
- Accelerated K-means
  - Reference: https://www.aaai.org/Papers/ICML/2003/ICML03-022.pdf
  - Code: https://github.com/siddheshk/Faster-Kmeans

Performance evaluation of clustering

There are two types of evaluation metrics for clustering:

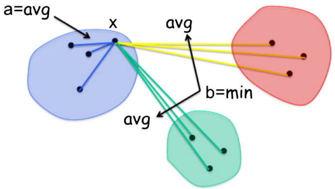
- Internal evaluation metrics: Silhouette coefficient
- External evaluation metrics: These metrics require the knowledge of the ground truth classes while almost never available in practice or require manual assignment by human annotators (as in the supervised learning setting).

Silhouette coefficient

- Given a clustering, we define
  - a: The mean distance between a point and all other points in the **same** cluster.
  - b: The smallest mean distance of a point to all points in any **other** cluster.
- Silhouette coefficient s for a single sample is formulated as:

$$s = \frac{b-a}{\max(a,b)} \;\Rightarrow\; s = \begin{cases} 1-\frac{a}{b} & \text{if } a<b \\ 0 & \text{if } a=b \\ \frac{b}{a}-1 & \text{if } a>b \end{cases}$$

- It is easy to know that $s \in (-1,1)$, and larger s value indicates better clustering performance.
- Silhouette coefficient s for a set of samples is defined as the mean of the Silhouette Coefficient for each sample.



Rand index

- Given a set of n samples $S = \{o_1, o_2, \ldots, o_n\}$, there are two clusterings/partitions of S to compare, including:
  - $X = \{X_1, X_2, \ldots, X_r\}$ with r clusters
  - $Y = \{Y_1, Y_2, \ldots, Y_s\}$ with s clusters
- We can calculate the following values:
  - a: The number of pairs of elements in S that are in the **same** subset in X and in the **same** subset in Y
  - b: The number of pairs of elements in S that are in the **different** subset in X and in the **different** subset in Y
  - c: The number of pairs of elements in S that are in the **same** subset in X and in the **different** subset in Y
  - d: The number of pairs of elements in S that are in the **different** subset in X and in the **same** subset in Y
- The rand index (RI) can be computed as follows:

$$\mathrm{RI} = \frac{a+b}{a+b+c+d} = \frac{a+b}{\frac{n(n-1)}{2}}$$

Note that $\mathrm{RI} \in [0, 1]$, and a higher score corresponds to a higher similarity.

Adjusted rand index

- Given a set of n samples $S = \{o_1, o_2, \ldots, o_n\}$, there are two clusterings/partitions of S to compare, including:
  - $X = \{X_1, X_2, \ldots, X_r\}$ with r clusters
  - $Y = \{Y_1, Y_2, \ldots, Y_s\}$ with s clusters
- The overlap between X and Y can be summarized in a contingency table, of which each entry n_{ij} denotes the number of sample in common between X_i and Y_j, *i.e.*, $n_{ij} = |X_i \cap Y_j|$

$X \diagdown Y$	Y_1	Y_2	$\cdots$	Y_s	sums
X_1	n_{11}	n_{12}	$\cdots$	n_{1s}	a_1
X_2	n_{21}	n_{22}	$\cdots$	n_{2s}	a_2
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
X_r	n_{r1}	n_{r2}	$\cdots$	n_{rs}	a_r
sums	b_1	b_2	$\cdots$	b_s	

Adjusted rand index is formulated as follows:

$$\mathrm{ARI} = \frac{\sum_{ij}\binom{n_{ij}}{2} - \left[\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}\right] / \binom{n}{2}}{\frac{1}{2}\left[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}\right] - \left[\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}\right] / \binom{n}{2}}.$$

Note that ARI could be positive or negative, and a higher score corresponds to a higher similarity.

https://blog.csdn.net/qtlyx/article/details/52678895

Performance evaluation of clustering

More evaluation metrics for clustering, as well as the demos with code, can be found in the following links:

- Wiki: https://en.wikipedia.org/wiki/Cluster_analysis#Internal_evaluation
- Demo with code: https://scikit-learn.org/stable/modules/clustering.html#clustering-evaluation

Other clusterings

Clustering is always an active area in machine learning. Despite of the introduced K-means algorithm, there are lots of other clustering algorithms, such as

- Hierarchical clustering
- Graph based clustering
- Density based clustering
- Probabilistic clustering

Further reading "Survey of Clustering Algorithms":

- https://axon.cs.byu.edu/Dan/678/papers/Cluster/Xu.pdf