Avisek Gupta (avisek003@gmail.com)

Machine Learning

23 – k-Means variants and Hierarchical Clustering

December 03, 2022

Recap: k-Means Clustering

Data Clustering: Given a data set $X = [x_1, ..., x_n],\ x_i \in \mathbb{R}^d$, identify naturally occuring groups (called *clusters*) in the data so that (i) similar data instances lie in the same cluster (ii) dissimilar data insances lie in different clusters.

An example of a data clustering method: k-Means clustering

k-Means clustering identifies k disjoint clusters $C_1, ..., C_k$, $C_i \cap C_j = \phi\ \forall i, j$, which partitions the data set $\cup_{j=1}^{k} C_j = X$, where each cluster C_j is represented by a single point v_j called the **center** of the cluster.

Recap: k-Means Clustering

The k-Means clustering problem is to estimate the k cluster centers $V = [v_1, ..., v_k]$, so that any data instance x_i lies at minimum distance to the center of its cluster v_j, i.e., if $x_i \in C_j$, then $||x_i - v_j||^2 \leq ||x_i - v_t||^2, \forall t$.

Let $U = [\mu_{ij}]_{n \times k} \in \{0, 1\}$ be the cluster membership of data instance x_i to cluster C_j.

The k-Means clustering objective function is defined as:

$$\min_{V,U} \; J_{KM} = \sum_{j=1}^{k} \sum_{i=1}^{n} \mu_{ij} ||x_i - v_j||^2 \;, \text{ s.t.} \sum_{j=1}^{k} \mu_{ij} = 1$$

Recap: k-Means Clustering

The k-Means clustering problem is to estimate the k cluster centers $V = [v_1, ..., v_k]$, so that any data instance x_i lies at minimum distance to the center of its cluster v_j, i.e., if $x_i \in C_j$, then $||x_i - v_j||^2 \leq ||x_i - v_t||^2, \forall t$.

Let $U = [\mu_{ij}]_{n \times k} \in \{0, 1\}$ be the cluster membership of data instance x_i to cluster C_j.

The k-Means clustering objective function is defined as:

$$\min_{V,U} \; J_{KM} = \sum_{j=1}^{k} \sum_{i=1}^{n} \mu_{ij} ||x_i - v_j||^2 \;, \text{ s.t.} \sum_{j=1}^{k} \mu_{ij} = 1$$

To optimize J_{KM}, update expressions for μ_{ij} and v_j:

$$\mu_{ij} = \begin{cases} 1 & \text{, if } ||x_i - v_j||^2 < ||x_i - v_t||^2, \forall t \neq j \\ 0 & \text{, o/w} \end{cases}$$

$$v_j = \frac{\sum_{i=1}^{n} \mu_{ij} x_i}{\sum_{i=1}^{n} \mu_{ij}} = \frac{\sum_{x_i \in C_j} x_i}{|C_j|}$$

Recap: k-Means Clustering

LLoyd's algorithm for k-Means:

1. Initialize k cluster centers by randomly selecting k unique data instances.
2. Repeat until convergence:
   (i) Calculate the squared Euclidea distances $||x_i - v_j||^2 \; \forall i, j$
   (ii) Calculate the cluster memberships

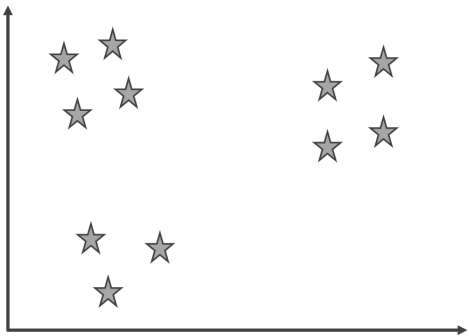
$$\mu_{ij} = \begin{cases} 1 & \text{, if } ||x_i - v_j||^2 < ||x_i - v_t||^2, \forall t \neq j \\ 0 & \text{, o/w} \end{cases}$$

   (ii) Calculate the cluster centers $v_j = \dfrac{\sum\limits_{x_i \in C_j} x_i}{|C_j|}$

Complexity of the k-Means algorithm: $O(kn)$.

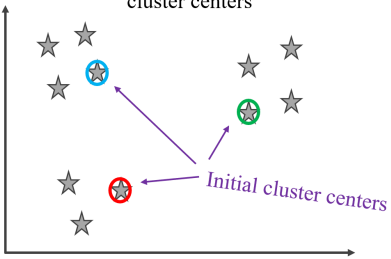
Recap: k-Means Clustering

An unlabeled data set, find k=3 clusters

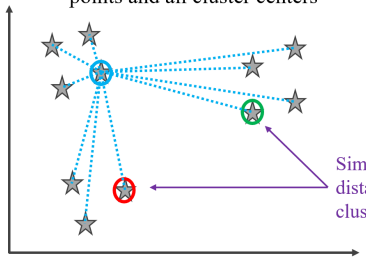


Recap: k-Means Clustering

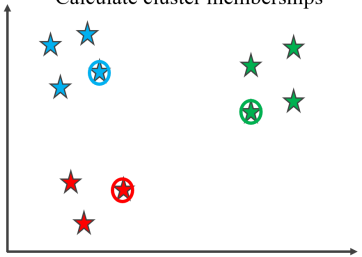
Initialize k=3 data instances as the initial cluster centers



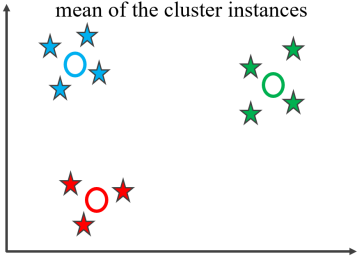
Calculate distances between all data points and all cluster centers



Calculate cluster memberships



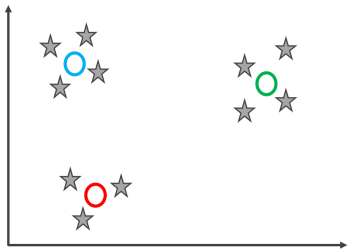
Update the cluster centers to be the mean of the cluster instances



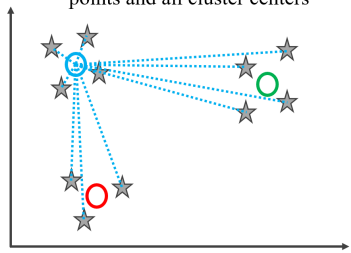
Recap: k-Means Clustering

In the next iteration:

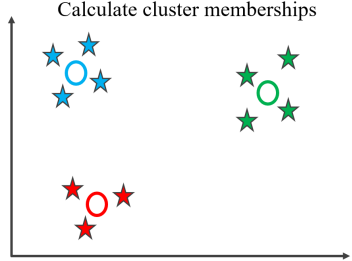
The cluster centers from the previous iteration



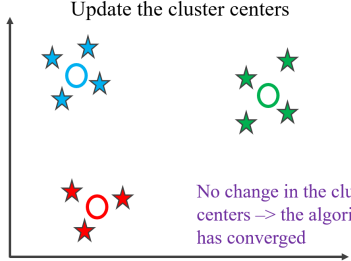
Calculate distances between all data points and all cluster centers



Calculate cluster memberships



Update the cluster centers



No change in the cluster centers –> the algorithm has converged

k-Means Clustering

- the choice of the number of clusters to be identified
- the choice of random initial centers
- the choice of distance metric used
- the choice of data features.

k-Means Clustering

The quality of clusters identified by k-Means depends on:

- **the choice of the number of clusters to be identified**
- the choice of random initial centers
- the choice of distance metric used
- the choice of data features.

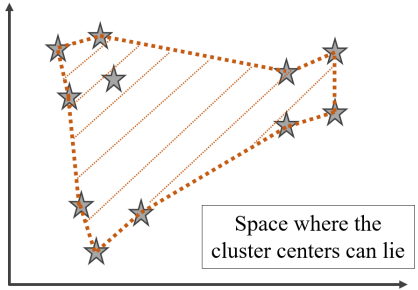
One approach to select the number of clusters:

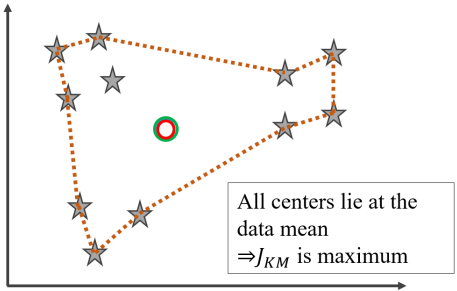
1. Obtain k-Means clusterings for $k = 2, ..., k_{\max}$
1. Calculate the Calinski-Harabasz Index at each $k = 2, ..., k_{\max}$:

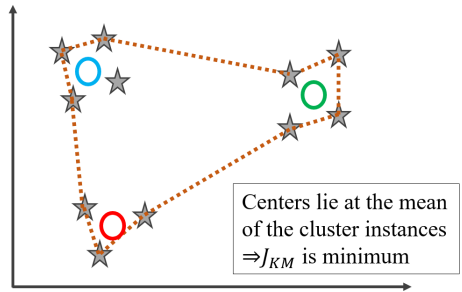
$$CH(k) = \frac{n-k}{k-1} \frac{\sum_{j=1}^{k} |C_j|.||v_j - \bar{x}||^2}{\sum_{j=1}^{k} \sum_{x_i \in C_j} ||x_i - v_j||^2}$$

3. Estimate the number of clusters as $\hat{k} = \arg\max_k CH$

k-Means Clustering





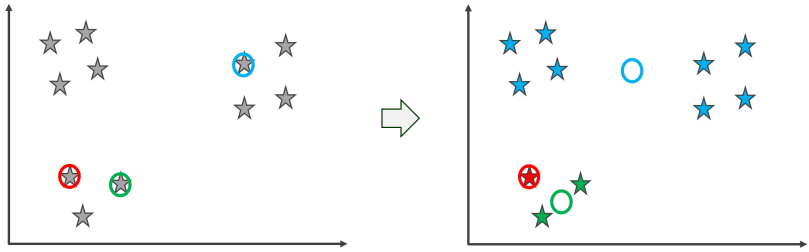


k-Means Clustering

The quality of clusters identified by k-Means depends on:

- the choice of the number of clusters to be identified
- **the choice of random initial centers**
- the choice of distance metric used
- the choice of data features.

Example of one center initialization that leads to inferior quality of clusters:



k-Means++ Center Initialization

Idea: Initial centers should be far apart from each other.

If centers are sequentially chosen, so that the next data instance is selected to be an initial center **with probability proportional to distance** to previous centers, then theoretical bounds can be provided with the optimal solution.

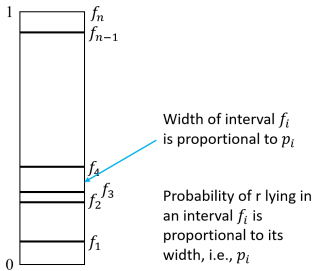
The k-Means++ initialization method:

1. Initialize the first center v_1 randomly from the data instances.
2. Repeat till we have k centers: If we currently have $(j-1)$ centers, then to find the j-th center:
   - 2.1 Calculate the minimum distance $d \in \mathbb{R}^n$, as $d_i = \min_{t=1}^{j-1} ||x_i - v_t||^2$
   - 2.2 Convert the distances to probabilities: $p = \frac{1}{\sum_{i=1}^{n} d_i} d$
   - 2.3 Convert the probabilities to cumulative frequencies: $f_t = \sum_{i=1}^{t} p_i$
   - 2.4 Obtain a random $r \in [0, 1]$, choose $v_j = x_i$ for $\max_i(f_i \leq r)$

k-Means++ Center Initialization

The k-Means++ initialization method:

1. Initialize the first center v_1 randomly from the data instances.
2. Repeat till we have k centers: If we currently have $(j-1)$ centers, then to find the j-th center:
   - 2.1 Calculate the minimum distance $d \in \mathbb{R}^n$, as $d_i = \min_{t=1}^{j-1} ||x_i - v_t||^2$
   - 2.2 Convert the distances to probabilities: $p = \frac{1}{\sum_{i=1}^{n} d_i} d$
   - 2.3 Convert the probabilities to cumulative frequencies: $f_t = \sum_{i=1}^{t} p_i$
   - 2.4 Obtain a random $r \in [0, 1]$, choose $v_j = x_i$ for $\max_i(f_i \leq r)$



k-Means Clustering

The quality of clusters identified by k-Means depends on:

- the choice of the number of clusters to be identified
- the choice of random initial centers
- **the choice of distance metric used**
- the choice of data features.

$$\min_{V,U} \; J_{KM} = \sum_{j=1}^{k} \sum_{x_i \in C_j} d(x_i, v_j)$$

d here can be any distance measure considered to be suitable for a problem.

E.g.: Kernel k-Means: If ϕ is a non-linear map to a higher dimension then, the inner prodict $\phi(a)^T \phi(b)$ can be written in terms of kernel functions $K(a, b)$,

$$\min_{U,V} J_{KKM} = \sum_{j=1}^{k} \sum_{x_i \in C_j} ||\phi(x_i) - v_j||^2$$

Kernel k-Means Clustering

If ϕ is a non-linear map to a higher dimension then, the inner prodict $\phi(a)^T\phi(b)$ can be written in terms of kernel functions $K(a, b)$,

$$\min_{U,V} \sum_{j=1}^{k} \sum_{x_i \in C_j} ||\phi(x_i) - v_j||^2$$

Equating the derivative J_{KKM} w.r.t v_j to zero,

$$v_j = \frac{\sum_{x_i \in C_j} \phi(x_i)}{|C_j|}$$

Substituting this expression of v_j back into the objective,

$$\min_{U} \sum_{j=1}^{k} \sum_{x_i \in C_j} ||\phi(x_i) - \frac{\sum_{x_s \in C_j} \phi(x_s)}{|C_j|}||^2$$

Kernel k-Means Clustering

$$\min_U \sum_{j=1}^{k} \sum_{x_i \in C_j} ||\phi(x_i) - \frac{\sum_{x_s \in C_j} \phi(x_s)}{|C_j|}||^2$$

Expanding the norm,

$$\min_U \sum_{j=1}^{k} \sum_{x_i \in C_j} \{\phi(x_i)^T \phi(x_i) - 2\frac{\sum\limits_{x_s \in C_j} \phi(x_i)^T \phi(x_s)}{|C_j|} + \frac{\sum\limits_{x_s \in C_j} \sum\limits_{x_r \in C_j} \phi(x_s)^T \phi(x_r)}{|C_j|^2}\}$$

To update U:

$$\begin{aligned} \mu_{ij} =& 1, \ \arg\min_j \{K(x_i, x_i) - 2\frac{\sum\limits_{x_s \in C_j} K(x_i, x_s)}{|C_j|} + \frac{\sum\limits_{x_s \in C_j} \sum\limits_{x_r \in C_j} K(x_s, x_r)}{|C_j|^2}\} \\ =& 0, \ o/w \end{aligned}$$

Kernel k-Means Clustering

The Kernel k-Means algorithm:

1. Precompute all $K(x_i, x_s),\ \forall i, s$
2. Randomly initialize U.
3. Repeat till convergence:
   (a) Compute $d_{ij} = \{K(x_i, x_i) - 2\frac{\sum_{x_s \in C_j} K(x_i, x_s)}{|C_j|} + \frac{\sum_{x_s \in C_j} \sum_{x_r \in C_j} K(x_s, x_r)}{|C_j|^2}\}$
   (b) Update U:

$$\mu_{ij} = \arg\min_j d_{ij}$$

Complexity of the Kernel k-Means algorithm: $O(n^2)$.

k-Means Clustering

The quality of clusters identified by k-Means depends on:

- the choice of the number of clusters to be identified
- the choice of random initial centers
- the choice of distance metric used
- **the choice of data features**

Deep learning approaches to data clustering aim to learn a suitable (usually lower dimensional) feature representation, while simultaneously also clustering the feature representations.

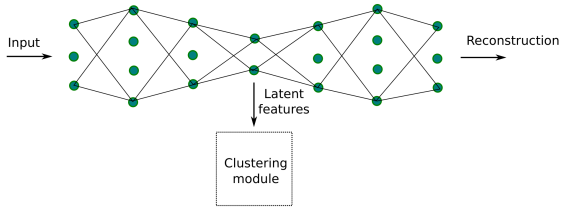


Image Source: Yang B., Fu X., Sidiropoulos N. D., Hong M., **Towards K-means-friendly Spaces: Simultaneous Deep Learning and Clustering**, *ICML* 2017.

Clustering based on pairwise distances

For k-Means clustering, we previously considered minimizing the distances between data instances and cluster centers

$$\min \sum_{j=1}^{k} \sum_{i=1}^{n} \mu_{ij} ||x_i - v_j||^2$$

An equivalent formulation can be considered where the pairwise distances between data instances in a cluster are minimized,

$$\min \sum_{j=1}^{k} \sum_{x_i \in C_j} \sum_{x_t \in C_j} ||x_i - x_t||^2$$

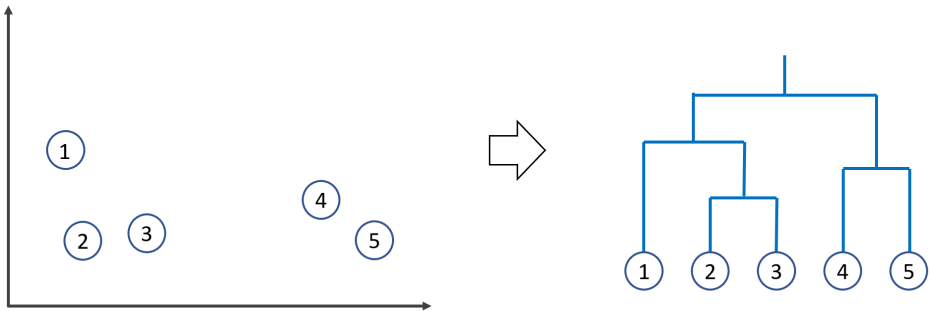
Directly optimizing the pairwise formulation leads to an $O(n^2)$ algorithm, whereas LLoyd's algorithm was $O(kn)$.

These two objectives are equivalent, since one can show that (Prove):

$$\frac{1}{|C_j|} \sum_{x_i, x_t \in C_j} \sum_{p=1}^{d} (x_{ip} - x_{tp})^2 = 2 \sum_{x_i \in C_j} \sum_{p=1}^{d} (x_{ip} - v_{jp})^2$$

Hierarchical Clustering

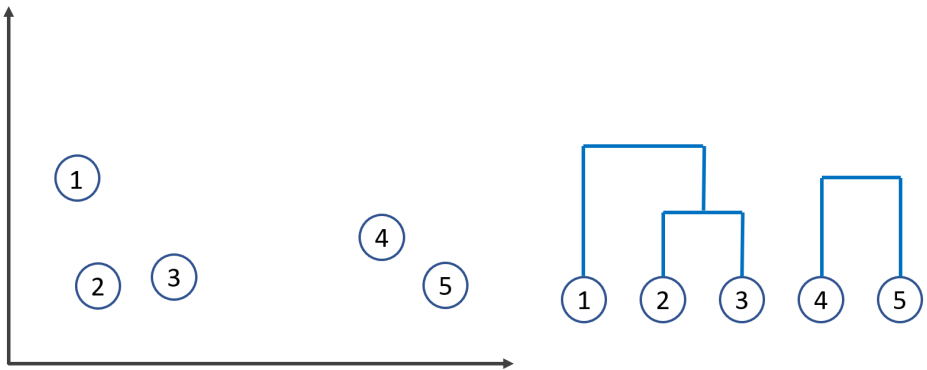
From the data set, we wish to construct a **dendogram** that captures a hierarchical relationship between all data instances, based on the distances between them.



Hierarchical Clustering

Once the dendogram is constructed, a clustering of any cluster number can be retreived from the dendogram.

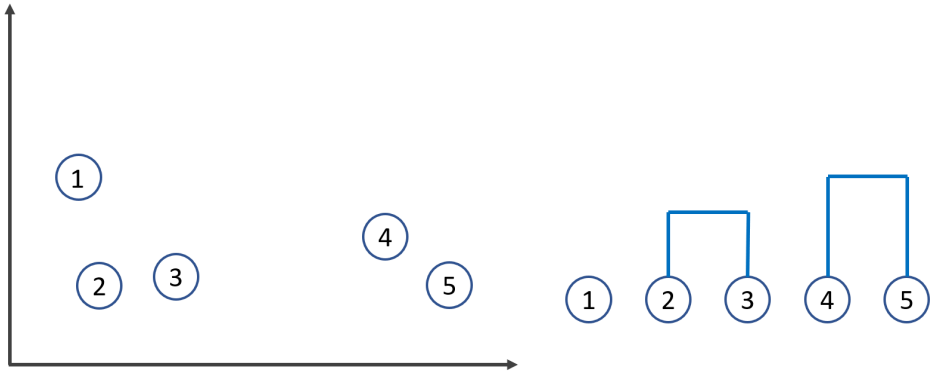
Obtaining $k = 2$:



Hierarchical Clustering

Once the dendogram is constructed, a clustering of any cluster number can be retreived from the dendogram.

Obtaining $k = 3$:



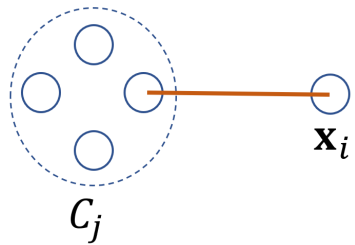
Hierarchical Clustering: linkages

Distance between two data instances: $d_{SE} = ||x_i - x_t||^2$

In order to build a dendogram, we need to define a measure of distance between a data instance x_i and a set of data instances C_j. These distances are called **linkages**. There can be several types of linkages.

1. Single Linkage: Defined as the minimum distance between an instance and an instance in a set.

$$d_s = \min_{x_t \in C_j} ||x_i - x_t||^2$$



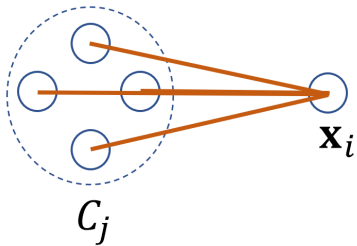
Hierarchical Clustering: linkages

Distance between two data instances: $d_{SE} = ||x_i - x_t||^2$

In order to build a dendogram, we need to define a measure of distance between a data instance x_i and a set of data instances C_j. These distances are called **linkages**. There can be several types of linkages.

2. Average Linkage: Defined as the average distance between an instance and an instance in a set.

$$d_{avg} = \frac{1}{|C_j|} \sum_{x_t \in C_j} ||x_i - x_t||^2$$



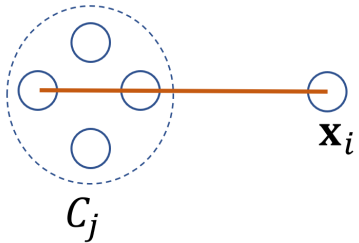
Hierarchical Clustering: linkages

Distance between two data instances: $d_{SE} = ||x_i - x_t||^2$

In order to build a dendogram, we need to define a measure of distance between a data instance x_i and a set of data instances C_j. These distances are called **linkages**. There can be several types of linkages.

3. Complete Linkage: Defined as the maximum distance between an instance and an instance in a set.

$$d_c = \max_{x_t \in C_j} ||x_i - x_t||^2$$



Hierarchical Clustering: linkages

The definitions of linkages are extended to be defined between two sets of data instances C_j and C_l.

1. Single Linkage:

$$d_s = \min_{x_i \in C_j, x_t \in C_l} ||x_i - x_t||^2$$

2. Average Linkage:

$$d_{avg} = \frac{1}{|C_j||C_l|} \sum_{x_i \in C_j, x_t \in C_l} ||x_i - x_t||^2$$

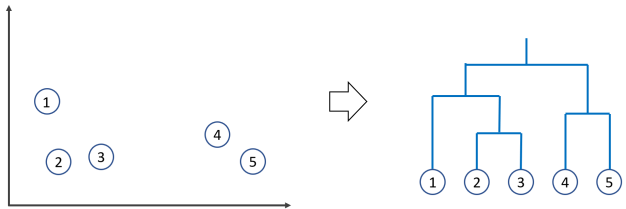
3. Complete Linkage:

$$d_c = \max_{x_i \in C_j, x_t \in C_l} ||x_i - x_t||^2$$

Hierarchical Clustering

Bottom-up: In **Agglomerative** hierarchical clustering, we start from all data instances being in isolated clusters, and progressively join clusters together to finally form a single cluster.

The complexity of agglomerative hierarchical clustering is $O(n^2)$.



Top-down: In **Divisive** hierarchical clustering, all instances are initially considered to be in a single cluster, and progressively one cluster is broken up into two clusters.

However this approach is not practical, since dividing a cluster with n instances requires considering $2^{n-1} - 1$ possible divisions.