HW3 Intelligent Systems

Unsupervised Machine Learning

Fardin Abbasi 810199456

School of Electrical & Computer Engineering

College of Engineering

University of Tehran

Fall 2023

Questions

Q1: Distance Metrics

A: Choose the best distance metric

Distance Metric	Dataset
Euclidean distance	Astronomical
Cosine similarity	Text documents
Jaccard similarity	Medical experiments
DBSCAN	Housing data

DBSCAN

The pseudocode of DBSCAN algorithm is as followed: [Read More]()

ALGORITHM 1: Pseudocode of Original Sequential DBSCAN Algorithm

```
Input: DB: Database
Input: ε: Radius
Input: minPts: Density threshold
Input: dist: Distance function
Data: label: Point labels, initially undefined
foreach point p in database DB do                          // Iterate over every point
    if label(p) ≠ undefined then continue                  // Skip processed points
    Neighbors N ← RangeQuery(DB, dist, p, ε)               // Find initial neighbors
    if |N| < minPts then                                   // Non-core points are noise
        label(p) ← Noise
        continue
    c ← next cluster label                                 // Start a new cluster
    label(p) ← c
    Seed set S ← N \ {p}                                   // Expand neighborhood
    foreach q in S do
        if label(q) = Noise then label(q) ← c
        if label(q) ≠ undefined then continue
        Neighbors N ← RangeQuery(DB, dist, q, ε)
        label(q) ← c
        if |N| < minPts then continue                      // Core-point check
        S ← S ∪ N
```

Since there are some obstacles between houses that deform their regular shape, it's necessary to use the DBSCAN algorithm, which clusters houses based on their density reachability.

Jaccard similarity

[Read More]

$$J(x, y) = \frac{n(x \cap y)}{n(x \cup y)}$$

Since this metric is designed for categorical features, its recommended for medical experiments dataset which is consist of categorical features.

Euclidean distance

$$d(X, Y) = \sqrt{\sum_{i=1}^{N} (x_i - y_i)^2}$$

Considering, astronomical dataset is represented with its 3D coordination, using Euclidean distance is the best choice.

Cosine similarity

$$cosine\ similarity(A, B) = \frac{A.B}{||A||\ ||B||}$$

If text documents are represented as numerical embedded features, cosine similarity would be the appropriate metric for it.

B: Dissimilarity matrix

#	Categorical feature	Ordinal feature	Numerical feature
1	A	Excellent	45
2	B	Average	22
3	C	Good	64
4	A	Excellent	28

$$\xrightarrow{Ordinal\ Encode}$$

#	Categorical feature	Ordinal feature	Numerical feature
1	A	3	45
2	B	1	22
3	C	2	64
4	A	3	28

For the categorical feature, **Jaccard distance** is used: $d_{i,j} = 1 - \frac{n(x_i \cap x_j)}{n(x_i \cup x_j)}$

The table below is the dissimilarity matrix for the categorical feature:

#	1	2	3	4
1	0	1	1	0
2	1	0	1	1
3	1	1	0	1
4	0	1	1	0

For the ordinal feature **Manhattan distance** is used: $d_{i,j} = |x_i - x_j|$

The table below is the dissimilarity matrix for the categorical feature:

#	1	2	3	4
1	0	2	1	0
2	2	0	1	2
3	1	1	0	1
4	0	2	1	0

For the numerical feature **distance** is defined as: $d(x_i, x_j) = \frac{|x_i - x_j|}{\max x - \min x}$

The table below is the dissimilarity matrix for the numerical feature:

#	1	2	3	4
1	0	0.55	0.45	0.40
2	0.55	0	1	0.14
3	0.45	1	0	0.86
4	0.40	0.14	0.86	0

The final dissimilarity matrix for all features results from averaging the dissimilarity matrix for each feature.

#	1	2	3	4
1	0	1.18	0.81	0.13
2	1.18	0	1	1.04
3	0.81	1	0	0.95
4	0.13	1.04	0.95	0

Q2: Clustering algorithms

A: K-means clustering

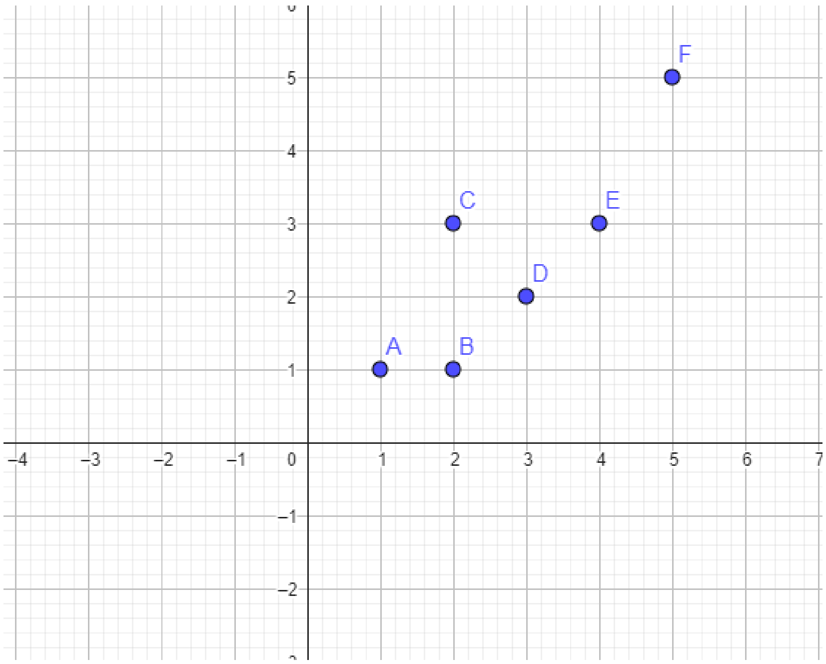
The pseudocode of K-means algorithm is depicted below:

Algorithm 1 k-means clustering

Initialise Cluster Centers
for each iteration l **do**
    Compute r_{nk}:
    for each data point x_n **do**
        Assign each data point to a cluster:
        for each cluster k **do**
            if $\mathrm{k} == \mathrm{argmin}\|\mathrm{x}_n - \mu_k^{l-1}\|$ **then**
                $r_{nk} = 1$
            else
                $r_{nk} = 0$
            end if
        end for
    end for
    for each cluster k **do**
        Update cluster centers as the mean of each cluster:
        $\mu_k^l = \frac{\sum r_{nk} x_n}{\sum r_{nk}}$
    end for
end for

Dataset:

i	x_1	x_2
A	1	1
B	2	1
C	2	3
D	3	2
E	4	3
F	5	5



$c_1 = B$

$c_2 = C$

$$d = \sqrt{\sum_{i=1}^{N} (x_i - y_i)^2}$$

Iter=1:

i/distance	c_1	c_2
A	1	2.23
B	0	1
C	2	0
D	1.414	1.414
E	2.828	2
F	5	3.6

$$c_1 = \frac{A + B + C}{3} = (1.66,1.66)$$

$$c_2 = \frac{D + E + F}{3} = (4,3.33)$$

Iter =2:

i/distance	c_1	c_2
A	0.93	3.8
B	0.74	3.07
C	1.38	2.02
D	1.38	1.66
E	2.7	0.33
F	4.72	1.94

$$c_1 = \frac{A+B+C+D}{4} = (2{,}1.75)$$

$$c_2 = \frac{E+F}{2} = (4.5{,}4)$$

Iter=3:

i/distance	c_1	c_2
A	1.25	4.6
B	0.75	3.9
C	1.25	2.7
D	1.03	2.5
E	2.36	1.11
F	4.42	1.11

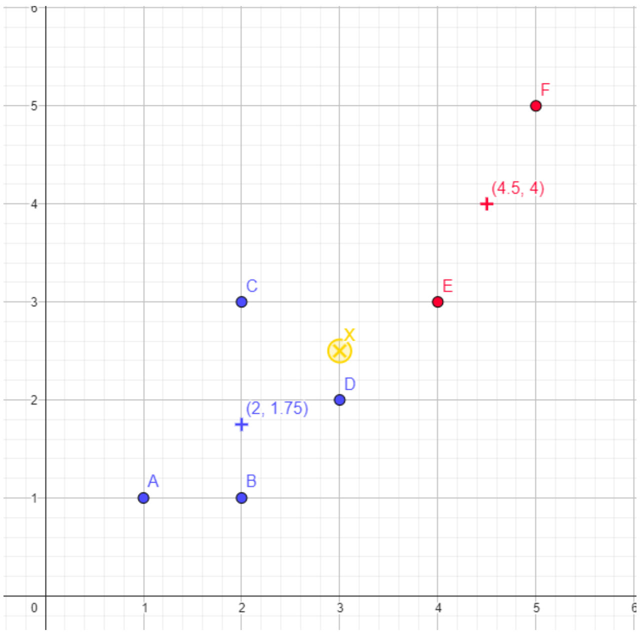
Since the clusters are not changed during last 2 iterations, the algorithm stops with the following centroids:

$$c_1 = \frac{A+B+C+D}{4} = (2{,}1.75)$$

$$c_2 = \frac{E+F}{2} = (4.5{,}4)$$

i/distance	c_1	c_2
$X = (3, 2.5)$	1.25	2.12

X belongs to the first cluster owing to its smaller distance from first cluster’s centroid.



B: Hierarchical clustering

Algorithm AgglomerativeClustering(D, $linkage$)

Input:

D: a distance matrix of size $n \times n$
$linkage(C_1, C_2)$: a distance function between clusters

```
Initialize L with n clusters, each containing a single data point
while |L| > 1 do
    Find pair of clusters (C1, C2) in L with the smallest distance
    Merge C1 and C2 into a new cluster C
    Remove C1 and C2 from L
    for each cluster C' ∈ L do
        d ← linkage(C, C')
        Update the matrix D to set the distance between C and C' to d
    Remove the distances related to C1 and C2 from D
    Add C to L
return the hierarchy of clusters
```

Figure 2 Agglomerative clustering schemes.

Name	Distance update formula FORMULA for $d(I \cup J, K)$	Cluster dissimilarity between clusters A and B
single	$\min(d(I,K), d(J,K))$	$\min\limits_{a \in A, b \in B} d[a,b]$
complete	$\max(d(I,K), d(J,K))$	$\max\limits_{a \in A, b \in B} d[a,b]$
average	$\dfrac{n_I d(I,K) + n_J d(J,K)}{n_I + n_J}$	$\dfrac{1}{\lvert A\rvert \lvert B\rvert} \sum\limits_{a \in A} \sum\limits_{b \in B} d[a,b]$
weighted	$\dfrac{d(I,K) + d(J,K)}{2}$	
Ward	$\sqrt{\dfrac{(n_I + n_K) d(I,K) + (n_J + n_K) d(J,K) - n_K d(I,J)}{n_I + n_J + n_K}}$	$\sqrt{\dfrac{2\lvert A\rvert \lvert B\rvert}{\lvert A\rvert + \lvert B\rvert}} \cdot \lVert \vec{c}_A - \vec{c}_B \rVert_2$
centroid	$\sqrt{\dfrac{n_I d(I,K) + n_J d(J,K)}{n_I + n_J} - \dfrac{n_I n_J d(I,J)}{(n_I + n_J)^2}}$	$\lVert \vec{c}_A - \vec{c}_B \rVert_2$
median	$\sqrt{\dfrac{d(I,K)}{2} + \dfrac{d(J,K)}{2} - \dfrac{d(I,J)}{4}}$	$\lVert \vec{w}_A - \vec{w}_B \rVert_2$

i	x_1	x_2
A	0.45	0.3
B	0.22	0.38
C	0.08	0.41
D	0.26	0.19
E	0.35	0.32

$$d = \sqrt{\sum_{i=1}^{N}(x_i - y_i)^2}$$

Single Linkage:

Dissimilarity matrix	A	B	C	D	E
A	0	0.24	0.38	0.22	0.1
B		0	0.143	0.2	0.143
C			0	0.28	0.28
D				0	0.15
E					0

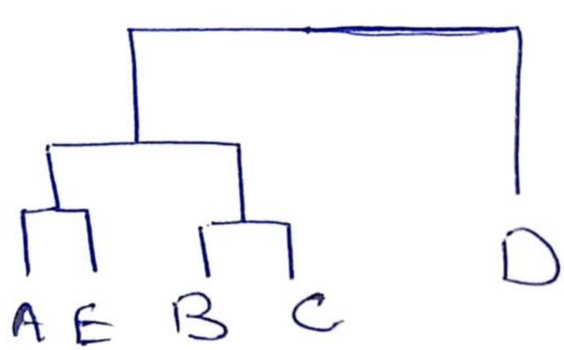
$\min d = d(A, E) \rightarrow A, E \; new \; cluster$

Dissimilarity matrix	B	C	D	A,E
B	0	0.14	0.2	0.14
C		0	0.28	0.28
D			0	0.15
A,E				0

$\min d = d(B, C) \rightarrow B, C \; new \; cluster$

Dissimilarity matrix	D	B,C	A,E
D	0	0.19	0.15
B,C		0	0.14
A,E			0

$\min d = d\big((B, C), (A, E)\big) \rightarrow A, B, C, E \; new \; cluster$



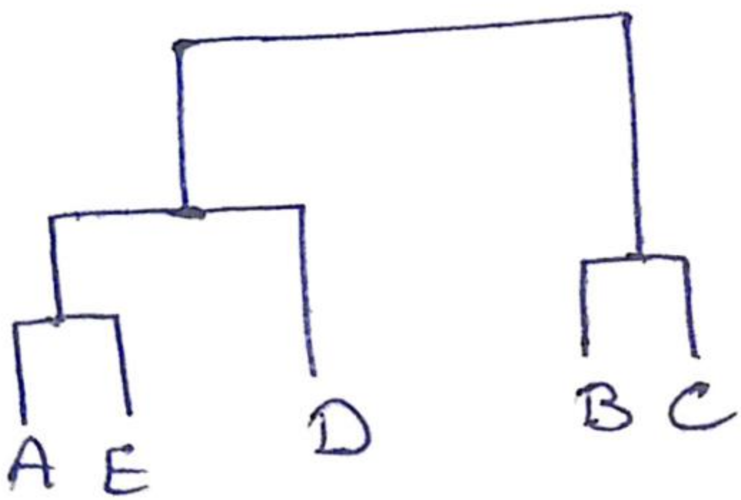
Complete Linkage:

Dissimilarity matrix	B	C	D	A,E
B	0	0.14	0.19	0.24
C		0	0.28	0.38
D			0	0.21
A,E				0

$\min d = d(B, C) \rightarrow B, C\ new\ cluster$

Dissimilarity matrix	D	B,C	A,E
D	0	0.28	0.21
B,C		0	0.38
A,E			0

$\min d = d\big((A, E), D\big) \rightarrow A, D, E\ new\ cluster$



Q3: K-means Vs. K-median

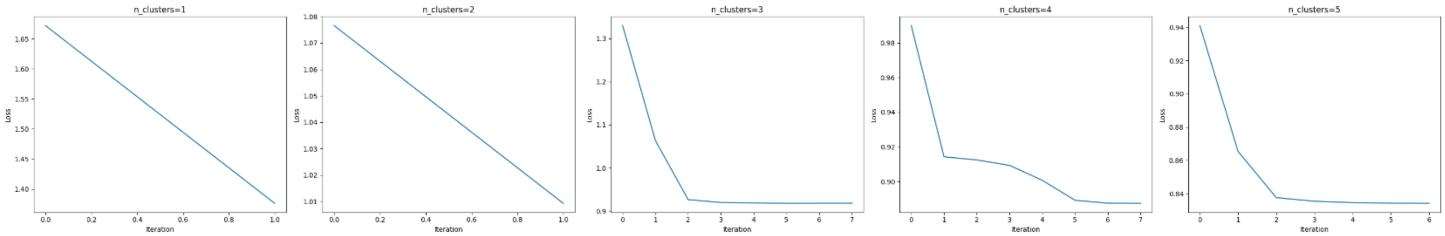
K-means++

Clustering Pseudo Code Using K-means++ and K-means

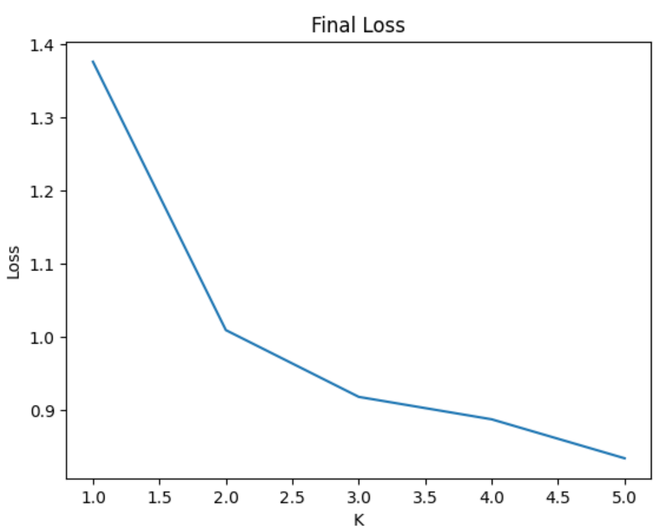
Input: Number of cluster (K), set of Requirements' RPV (Y) = $\{y_1, y_2, y_r\}$
Output: Three clusters of requirements (low, medium, high)
Start
- Choose the first cluster centroid (c_1) uniformly at random from Y
- **Repeat**
  - For each y_r, compute the distance y to the nearest cluster centroid c_i using $E(y_r) = \sqrt{(c_i - y_r)^2}$
  - Choose the next cluster centroid c_i , selecting $c_i = y_r \in$ Y with probability $\frac{E(y_r)^2}{\sum_{y \in Y} E(y)^2}$
- **Until** all the total of cluster centroids have been chosen
- **Do**
  - For each y, compute the distance y_r to each defined cluster centroid c_i using $E_{ir} = \sqrt{(c_i - y_r)^2}$
  - Assign each y to the cluster with the closest centroid c_i.
  - Update each cluster centroids c_i by taking the average of the all assigned y_r in each cluster.
- **While** (no longer changes in the cluster centroids).

End

C:



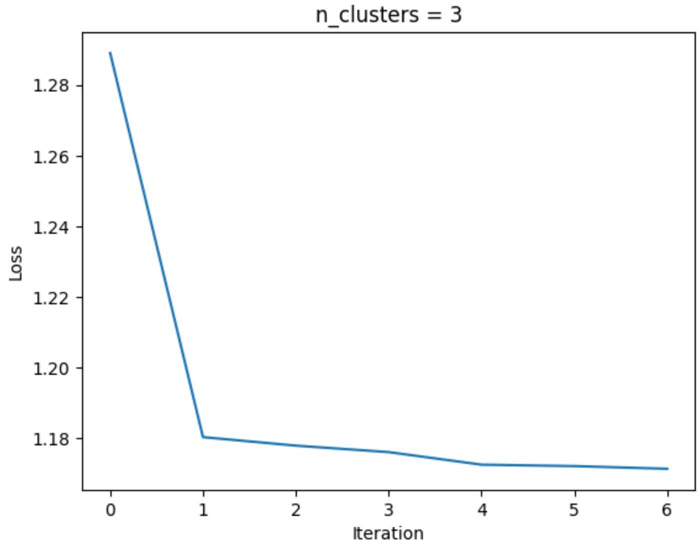
D:



The best number of clusters can be estimated using elbow point which here is 2 or 3 clusters.

K-median++

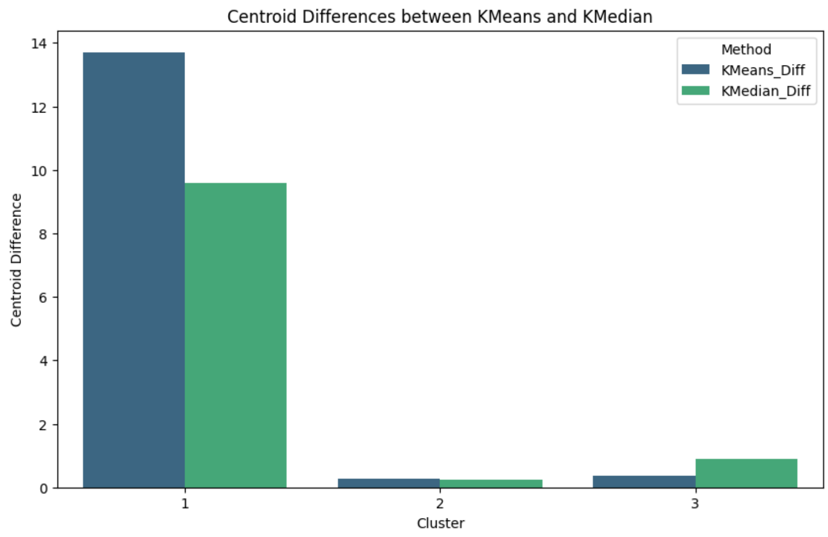
A:



B:

The norm of centroid differences with and without outliers for each cluster is depicted below.

K-median is more robust to outliers rather than k-means since its distance function is Manhattan distance instead of Euclidean distance. K-means is more sensitive to outliers.



[Source code]