2143488 BIG DATA AND ARTIFICIAL INTELLIGENCE

DR. JING TANG

K-NEAREST NEIGHBOR (KNN)

K-NEAREST NEIGHBOR(KNN)

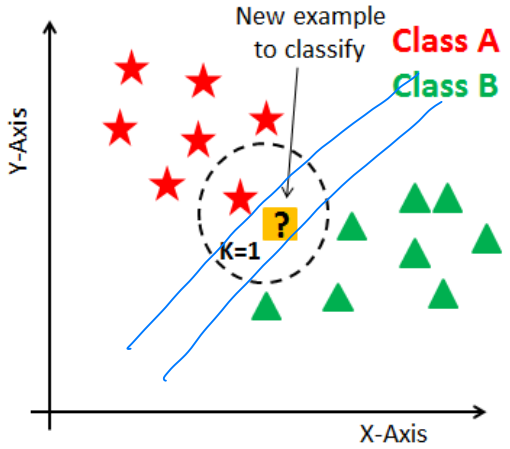
- Very simple, easy to understand, versatile
  - **Non-parametric**: no assumption for underlying data distribution
  - **Lazy**: no need to separate train vs. test dataset
  - Need time and memory to deal with all data
- One of the topmost machine learning **CLASSIFICATION** algorithms
- i.e.:
  - Credit ratings: financial institutes will predict the credit rating of customers
  - Loan disbursement: banking institutes will predict whether the loan is safe or risky
  - Political science: classifying potential voters in two classes will vote or won't vote.

↳ but this one can be used for regression as well
↳ most classification methods can be used for regression as well
  ↳ but not the other way around
↳ For regression problem, try the traditional regression first
  ↳ some classification model could outperform them, like "Gradient boost tree."
  so you can try it afterwards

WHAT IS K?

- In KNN, K is the number of nearest neighbors. The number of neighbors is the core deciding factor.
- K is generally an odd number if the number of classes is 2.
- When $K = 1$, then the algorithm is known as the nearest neighbor algorithm. This is the simplest case. Suppose P1 is the point, for which label needs to predict. First, you find the one closest point to P1 and then the label of the nearest point assigned to P1.

Distance-based
x and y must be numerical values



The k you choose will affect the result
↳ k too small: may be affected by outliers

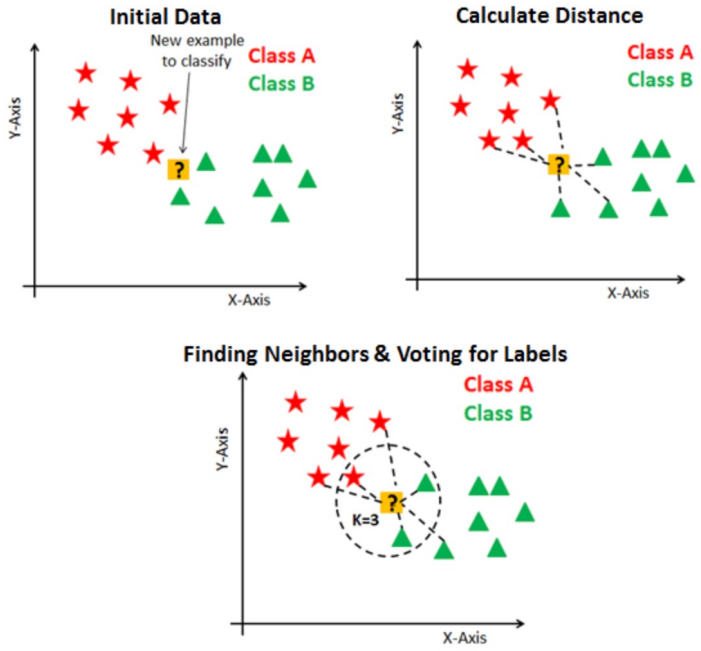
STEPS IN KNN

- Calculate Euclidean distance:

$$D = \sqrt{(X_{11} - X_{21})^2 + \cdots (X_{1n} - X_{2n})^2}$$

- Find closest neighbors
- Vote for labels

Find the k nearest distances

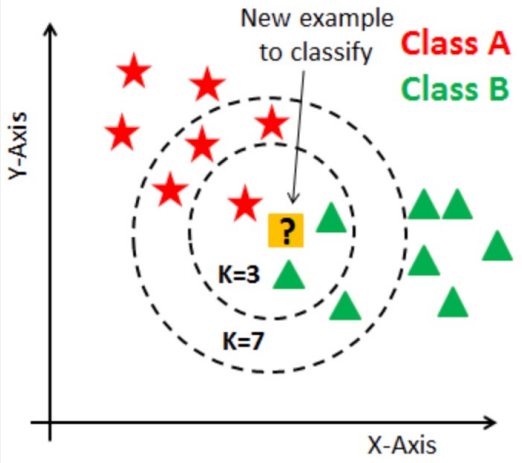


CURSE OF DIMENSIONALITY

- KNN performs better with **a lower number of features** than a large number of features Recommended 1 - 10
- Increase in dimension also leads to the problem of **overfitting**
- To avoid overfitting, the needed data will need to grow exponentially as you increase the number of dimensions
- This problem of higher dimension is known as the **Curse of Dimensionality**
- **Which means you need to select only important features**

HOW TO CHOOSE K

- The number of neighbors(K) in KNN is a **hyperparameter** that you need choose at the time of model building
- NO optimal number of neighbors suits all kind of data sets
  - A small number of neighbors leads to a higher influence of the noise on the result
  - A large number of neighbors make it computationally expensive
- General solutions:
  - Choose as an odd number if the number of classes is even
  - Check by generating the model on different values of k and check their performance. (ELBOW method)



Midterm
↳ ask to design the steps
  ↳ how?
↳ evaluation
  ↳ how to evaluate

PROCEDURE 1

- Load data:
- Explore data:
  - Size (col & row)
  - Quality

PROCEDURE 2

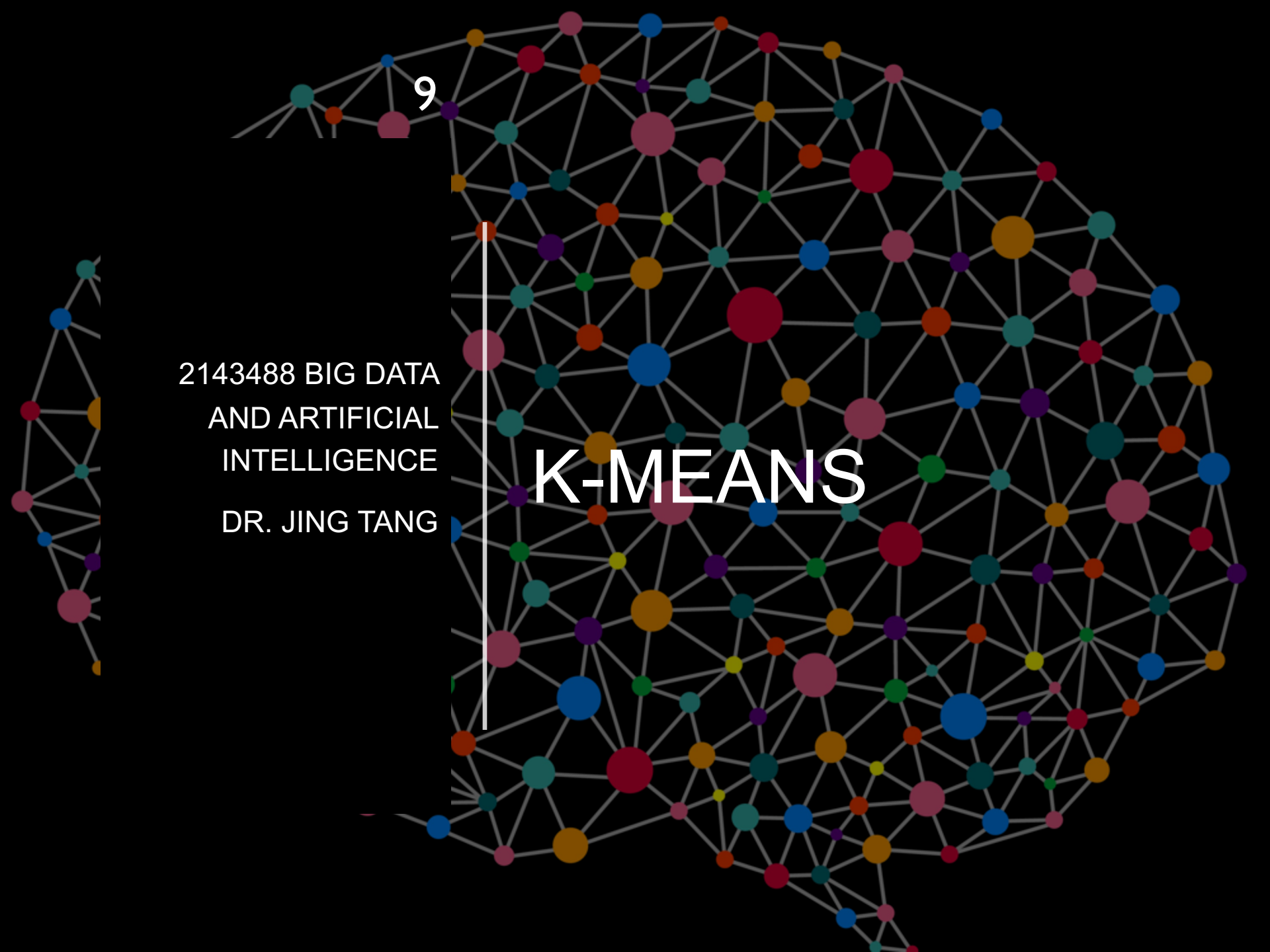
- Preprocess data:
  - Split train-test if you want to check the impact of k
  - Encoding data columns
  - Fill in missing value in train dataset
  - Scale features in train dataset
  - Select features (only numerical) in train dataset
- Model:
  - Debug KNN model to make the model work
  - Preprocess test dataset same as train dataset
  - Optimize k, based on classification report or other metrics

9

2143488 BIG DATA AND ARTIFICIAL INTELLIGENCE

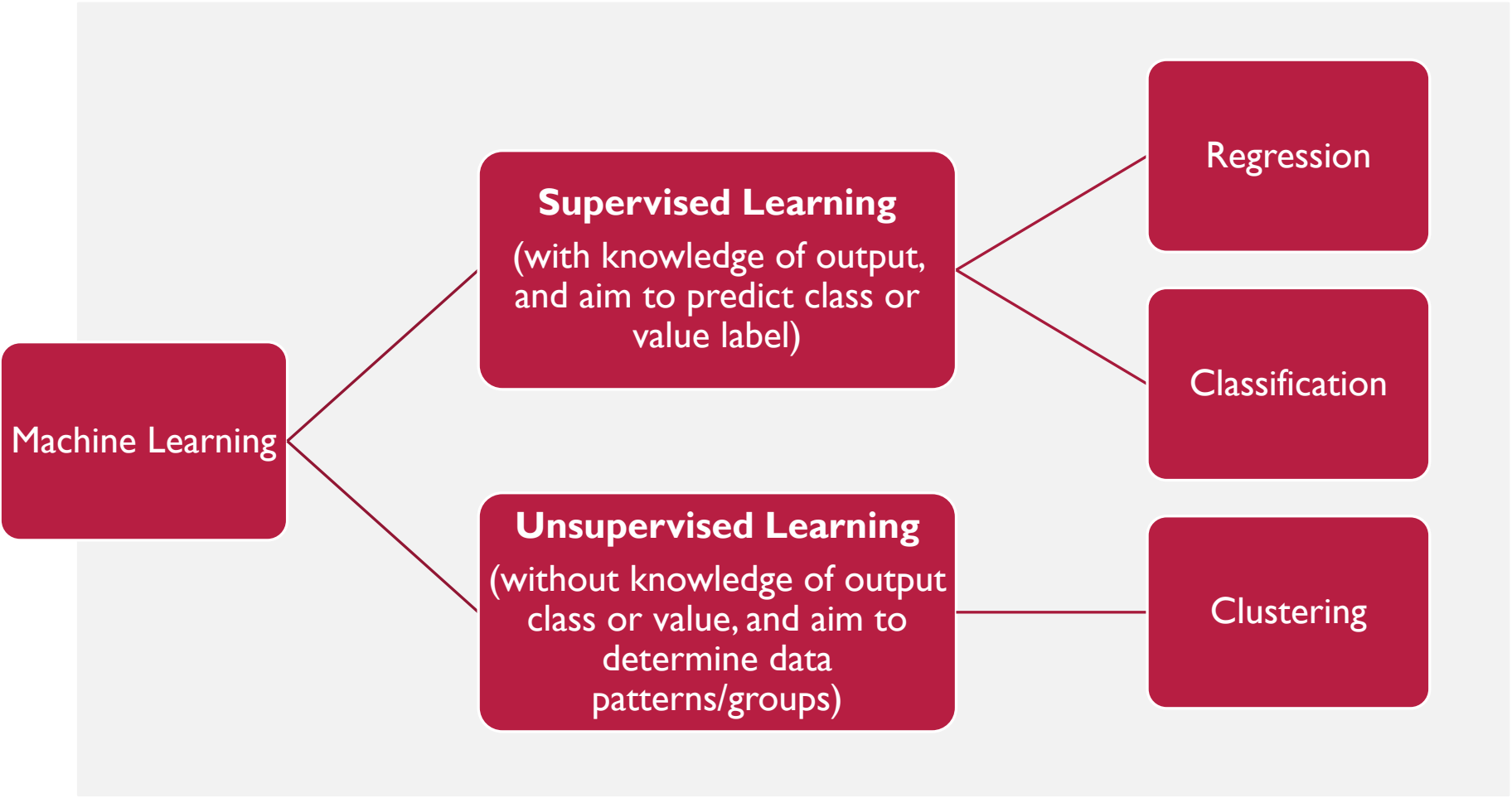
DR. JING TANG

K-MEANS



★ midterm will ask which kind of problem / model

MACHINING LEARNING



K-MEANS CLUSTERING (MACQUEEN 1967)

- K-means clustering is a type of **unsupervised** learning, which is used when you have **unlabeled data** (i.e., data without defined categories or groups).
- The goal of this algorithm is to find **groups** in the data, with the number of groups represented by the variable .
- The algorithm works iteratively to assign each data point to one of K groups based on the features that are provided.
- Data points are clustered based on **feature similarity**.

K-MEANS CLUSTERING

- The results of the *K*-means clustering algorithm are:
  - The **centroids** of the *K* clusters, which can be used to label new data
  - **Labels** for the training data (each data point is assigned to a single cluster)

K-MEANS CLUSTERING

- **Each centroid** of a cluster is a **collection of feature values** which define the resulting groups.
- Examining the **centroid feature weights** can be used to qualitatively interpret what **kind of group** each cluster represents.

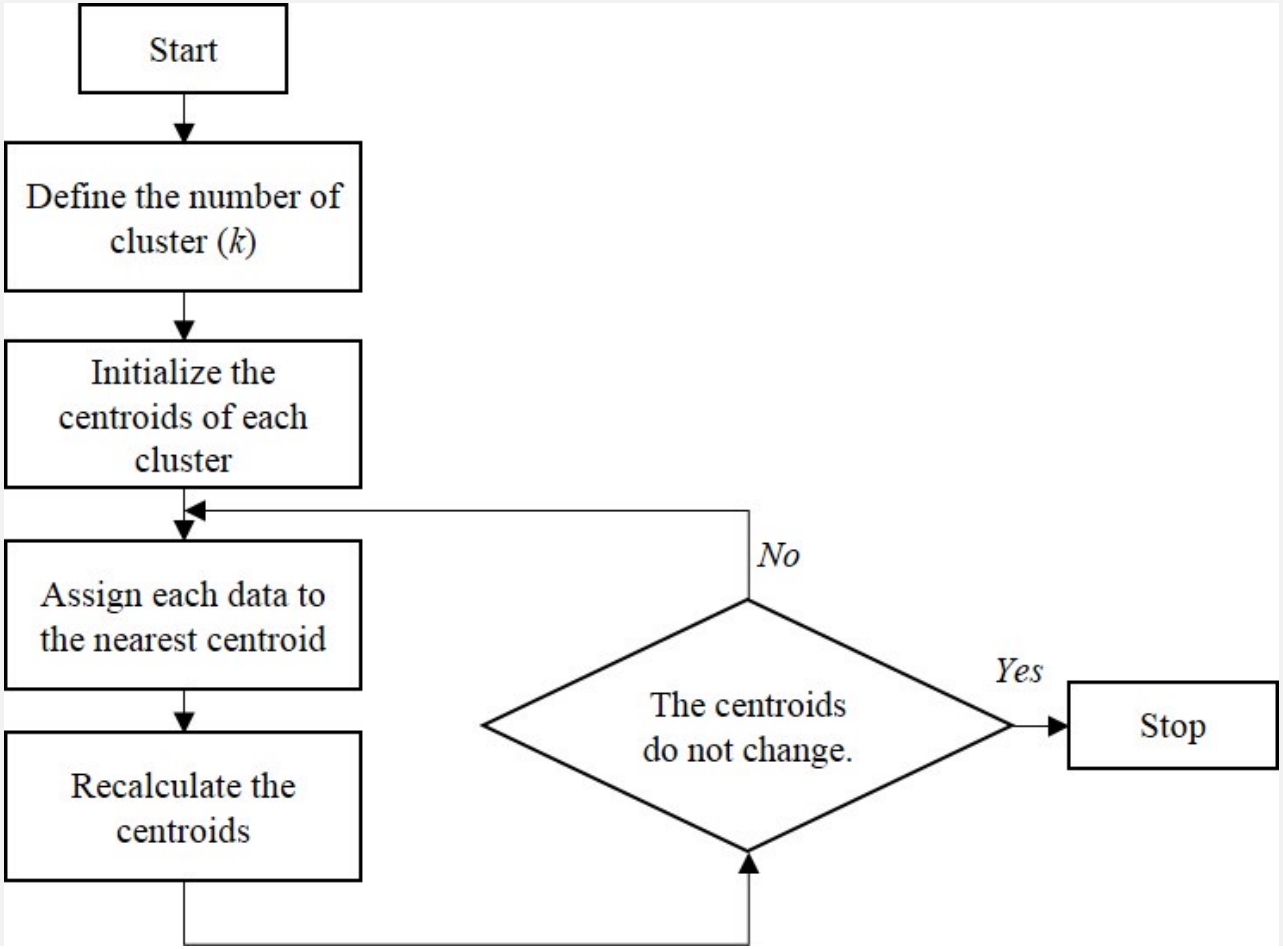
X1 = age

cluster 1 : X1 is large ⇒ adults
cluster 2 : X1 is small ⇒ kids

ALGORITHM

- The *K*-means clustering algorithm uses **iterative refinement** to produce a final result.
- Two inputs: the number of clusters *K* and the data set.
- The data set is a collection of features for each data point.
- The algorithms starts with **initial estimates for the *K* centroids**, which can either be randomly generated or randomly selected from the data set.

ALGORITHM


Start
Define the number of cluster (k)
Initialize the centroids of each cluster
Assign each data to the nearest centroid
Recalculate the centroids
The centroids do not change.
No
Yes
Stop

ALGORITHM

- The algorithm then iterates between two steps:

1. Data assignment step
2. Centroid update step

update until the centroid is unchanged

ALGORITHM
1. DATA ASSIGNMENT STEP:

- Each centroid defines one of the clusters.
- In this step, each data point is assigned to its nearest centroid, based on the Euclidean distance.

EUCLIDEAN DISTANCE

$$d_{ij} = \sqrt{\sum_{v=1}^{V} (x_{iv} - c_{jv})^2}$$

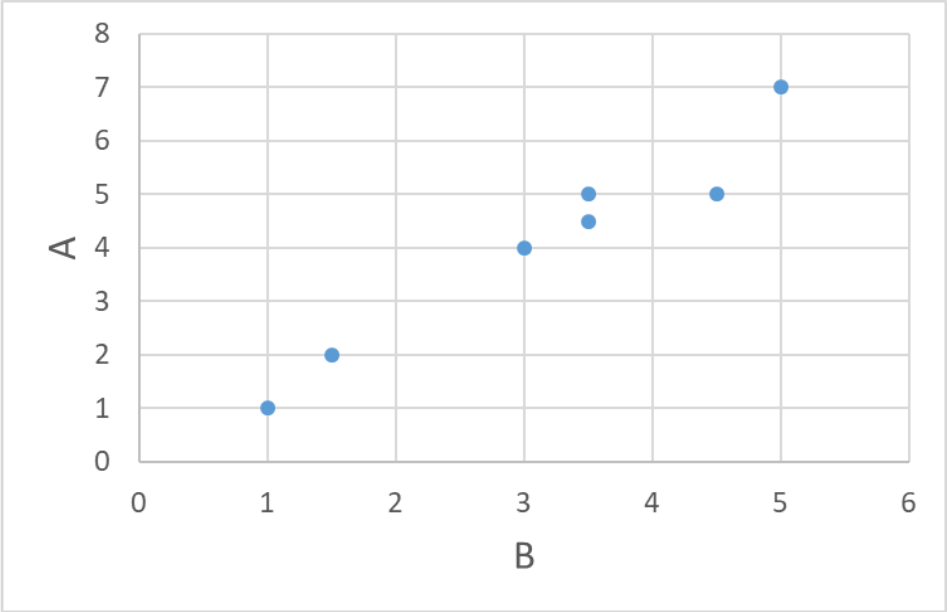
- where x_{iv} is the value of attribute v of the data i, and c_{jv} is the value of the attribute v of the centroid of the cluster j.

ALGORITHM
2. CENTROID UPDATE STEP:

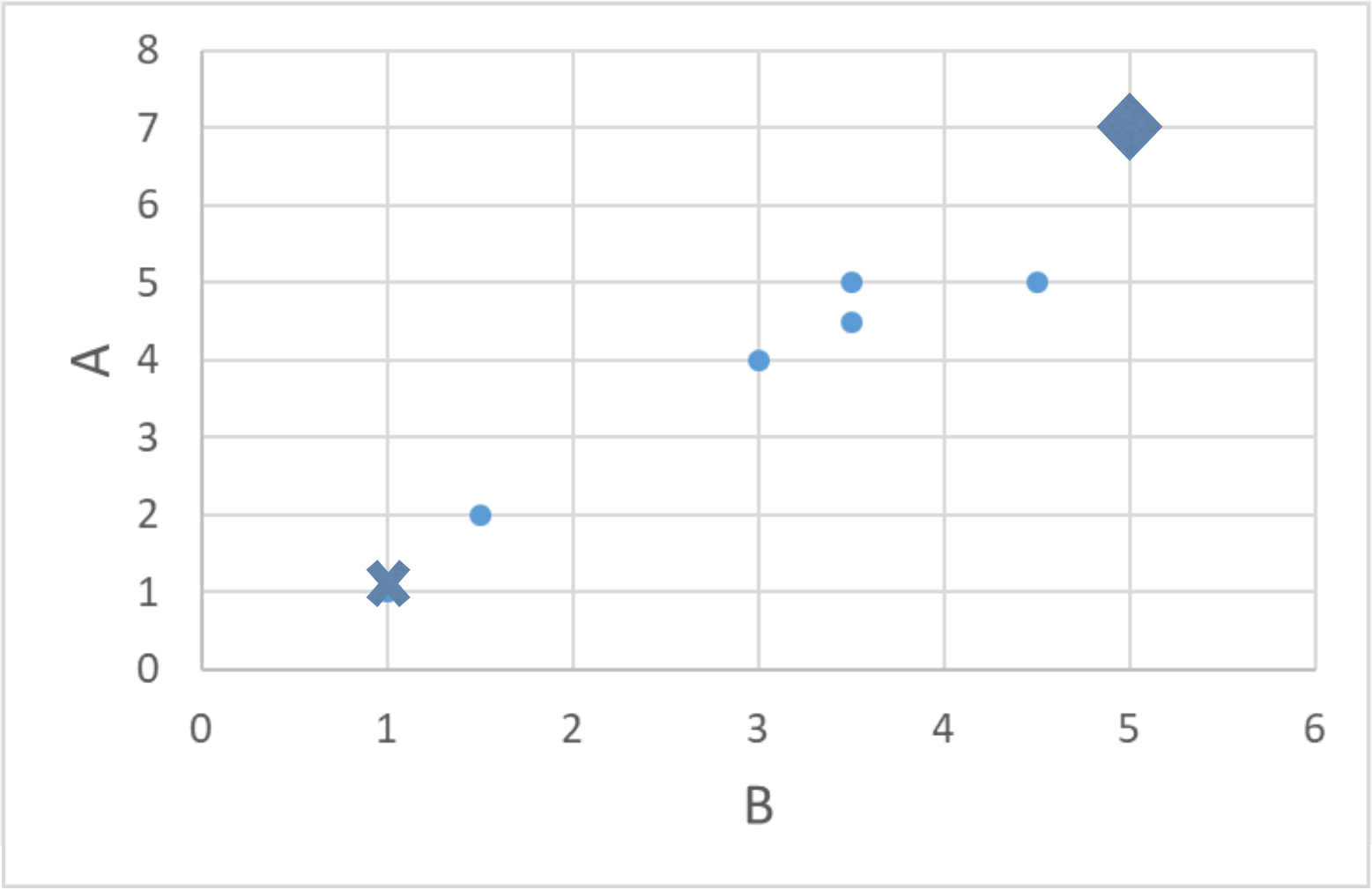
- In this step, the centroids are recomputed. This is done by taking the mean of all data points assigned to that centroid's cluster.

EXAMPLE: DATA

	X_1	X_2
Subject	A	B
1	1	1
2	1.5	2
3	3	4
4	5	7
5	3.5	5
6	4.5	5
7	3.5	4.5



EXAMPLE: ITERATION 1



EXAMPLE: ITERATION 1

Iteration 1

Subject	A	B	Distance C1	Distance C2
1	1	1	0.00	7.21
2	1.5	2	1.12	6.10
3	3	4	3.61	3.61
4	5	7	7.21	0.00
5	3.5	5	4.72	2.50
6	4.5	5	5.32	2.06
7	3.5	4.5	4.30	2.92

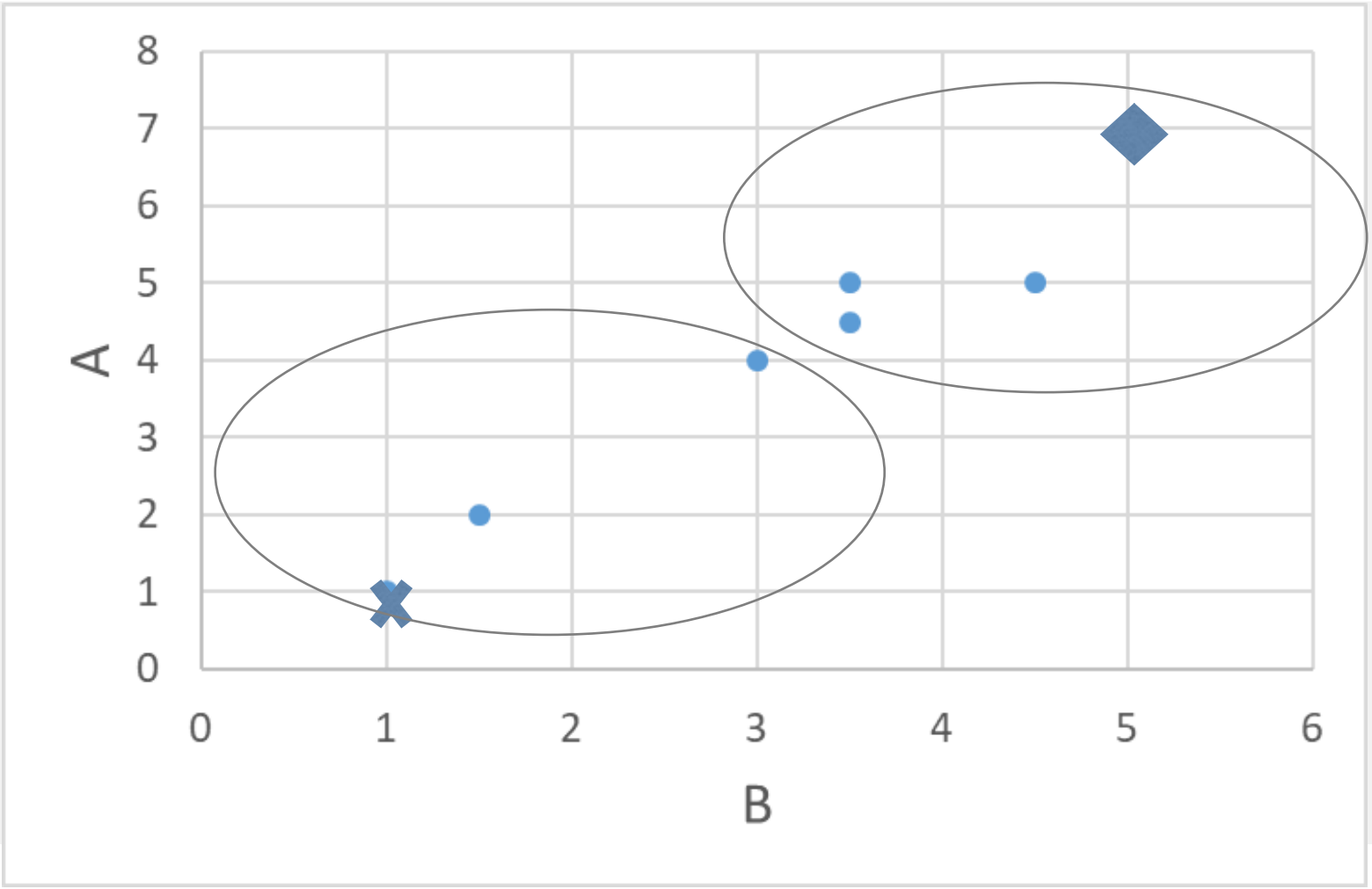
Initial centriod

	A	B
Centroid 1	1.00	1.00
Centroid 2	5.00	7.00

Re-compute centroids

	A	B
Centroid 1	1.8	2.3
Centroid 2	4.1	5.4

EXAMPLE: ITERATION 1



EXAMPLE: ITERATION 2

Iteration 2

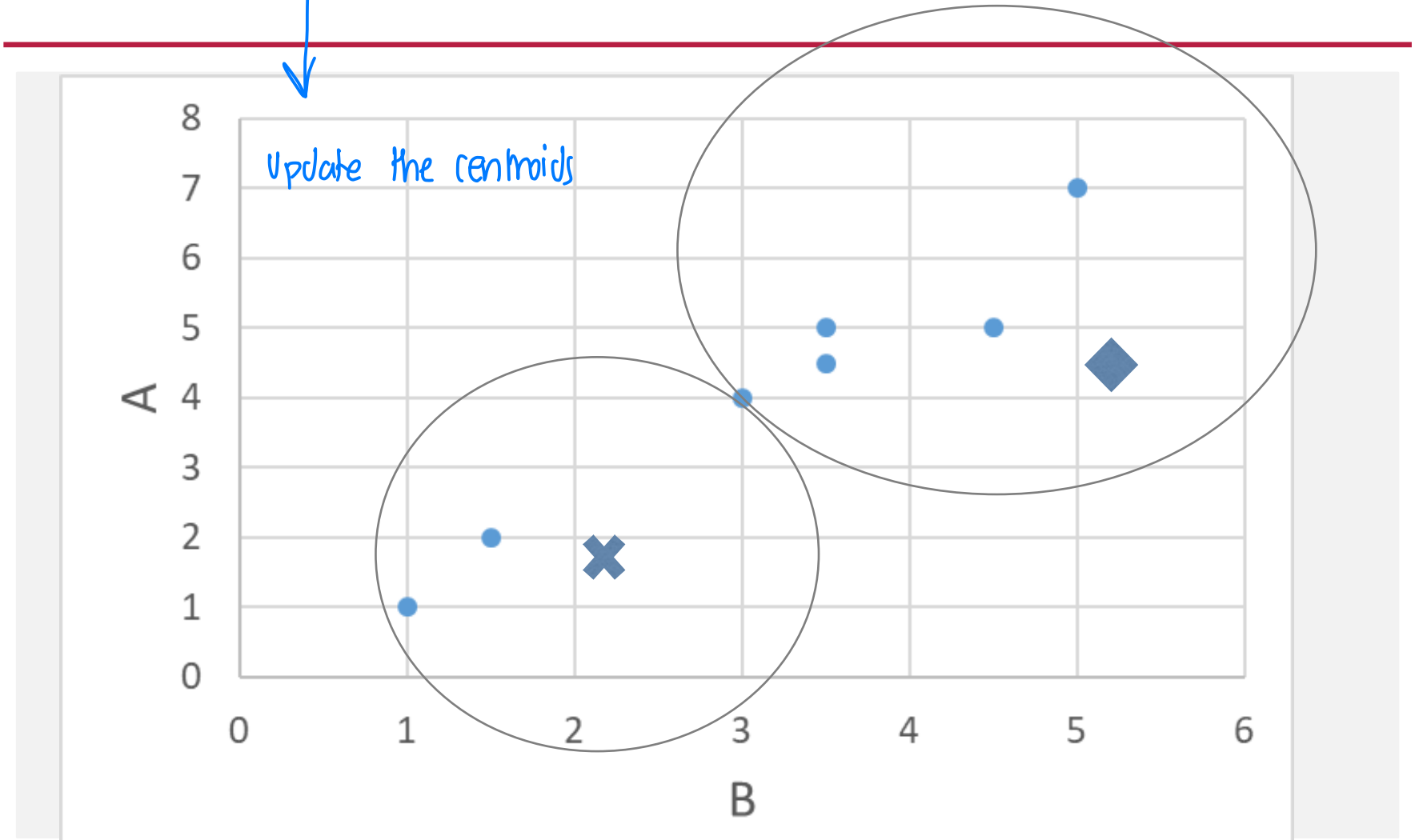
Subject	A	B	Distance C1	Distance C2
1	1	1	1.57	5.38
2	1.5	2	0.47	4.28
3	3	4	2.03	1.78
4	5	7	5.64	1.85
5	3.5	5	3.14	0.73
6	4.5	5	3.77	0.53
7	3.5	4.5	2.73	1.08

	A	B
Centroid 1	1.83	2.33
Centroid 2	4.13	5.38

Re-compute centroids

	A	B
Centroid 1	1.3	1.5
Centroid 2	3.9	5.1

EXAMPLE: ITERATION 2



EXAMPLE: ITERATION 3

Iteration 3

Calculate the distance again

Subject	A	B	Distance C1	Distance C2
1	1	1	0.56	5.02
2	1.5	2	0.56	3.92
3	3	4	3.05	1.42
4	5	7	6.66	2.20
5	3.5	5	4.16	0.41
6	4.5	5	4.78	0.61
7	3.5	4.5	3.75	0.72

	A	B
Centroid 1	1.25	1.50
Centroid 2	3.90	5.10

Re-compute centroids

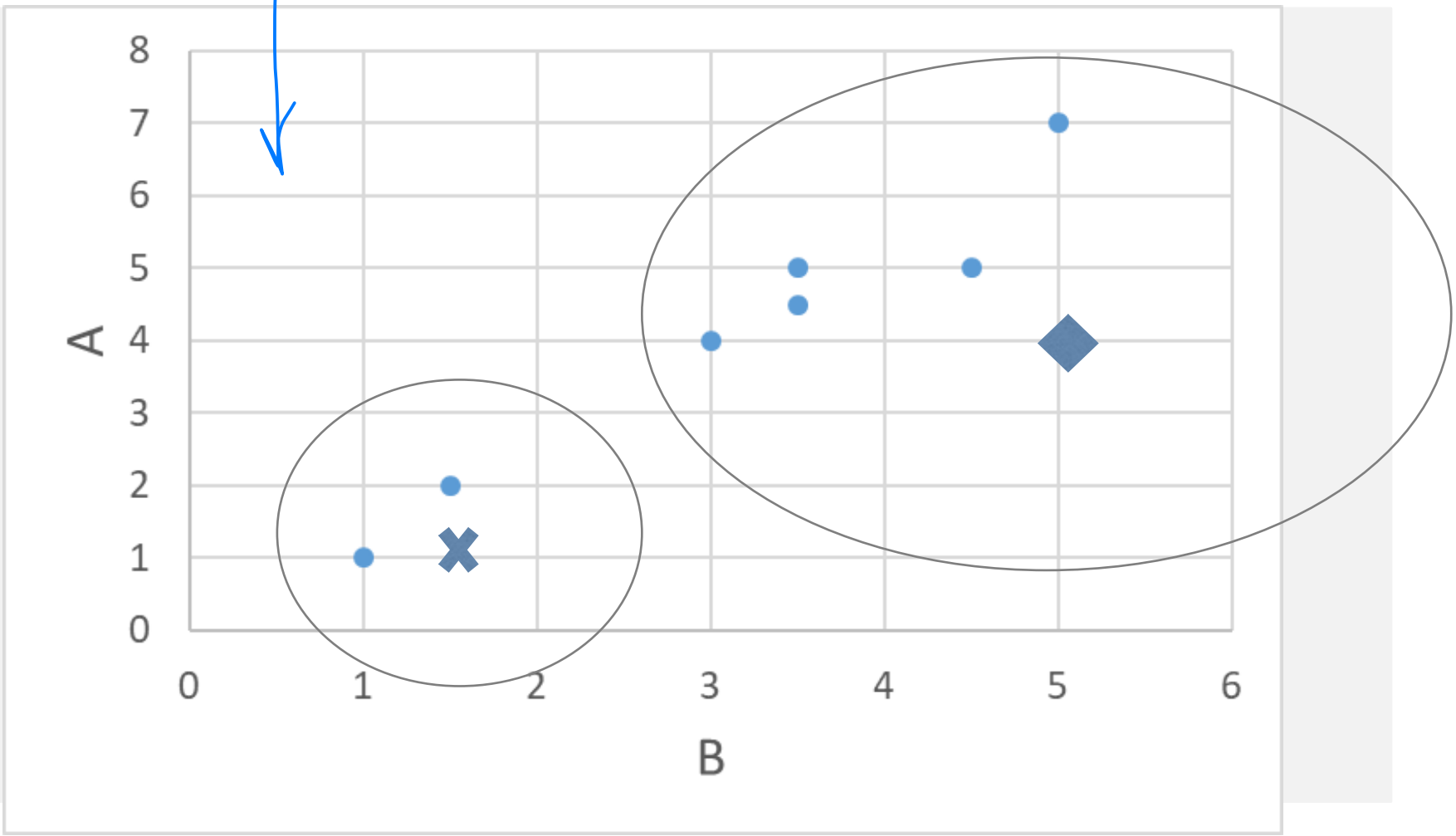
	A	B
Centroid 1	1.25	1.50
Centroid 2	3.90	5.10

Still different?
Continue

Stop

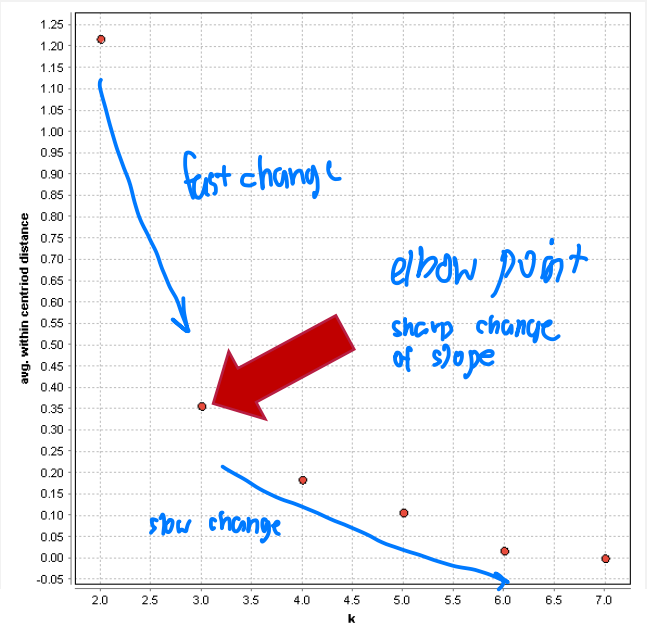
update. stop when centroids remain unchanged.

EXAMPLE: ITERATION 3



SELECT THE BEST K: ELBOW POINT

- Plot graph Within-Cluster-Sum-of Squares (OR **avg. within centroid distance**) vs. **K**
- Select the best K from **elbow point**
  - It demarks **significant drop-in** rate of increase.



SELECT THE BEST K: SILHOUETTE COEFFICIENT

- Silhouette coefficient (Rousseeuw 1987) of observation *i* is calculated as:

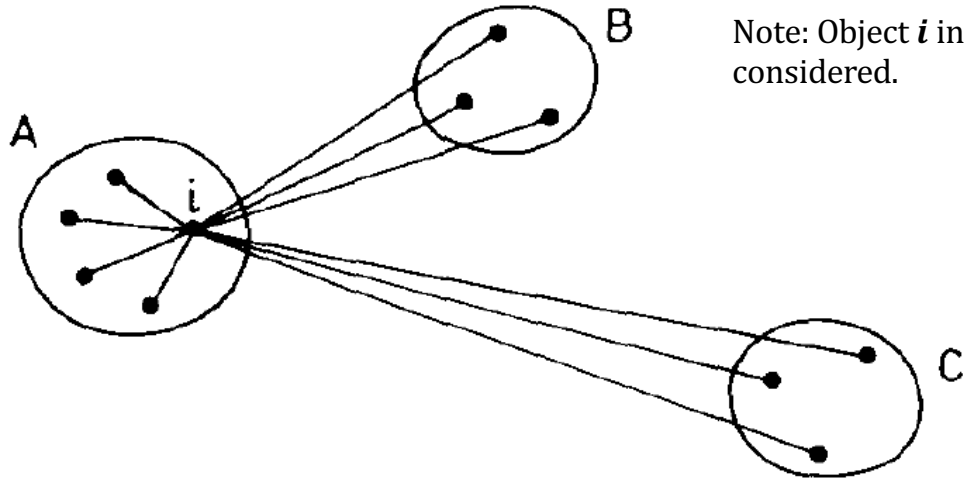
want as big $b_i - a_i$ as possible

$$s_i = \frac{(b_i - a_i)}{\max(a_i, b_i)}, \qquad -1 \leq s_i \leq 1$$

Where:
a_i: **average distance** of observation i to all other observations within same cluster
b_i: **minimum of average distance** of observation i to all other observations from all other clusters.

  - K giving the **highest average** of Silhouette (*S*) is the best K.
- It applies to any cluster, not just k-means.

SELECT THE BEST K: SILHOUETTE



Note: Object ***i*** in cluster ***A*** is being considered.

$a(i)$ = avg. dissimilarity of object ***i*** to all other objects within the **same cluster**

$d(i, O)$ = avg. dissimilarity of object ***i*** to all objects in the **other cluster** O

$$b(i) = \min_{O \neq A} d(i, O)$$

SELECT THE BEST K: SILHOUETTE

- Euclidean distance

A		1	1.5	3	5	3.5	4.5	3.5
	Subject	1	2	3	4	5	6	7
1	1	0.00	0.25	4.00	16.00	6.25	12.25	6.25
1.5	2	0.25	0.00	2.25	12.25	4.00	9.00	4.00
3	3	4.00	2.25	0.00	4.00	0.25	2.25	0.25
5	4	16.00	12.25	4.00	0.00	2.25	0.25	2.25
3.5	5	6.25	4.00	0.25	2.25	0.00	1.00	0.00
4.5	6	12.25	9.00	2.25	0.25	1.00	0.00	1.00
3.5	7	6.25	4.00	0.25	2.25	0.00	1.00	0.00

B		1	2	4	7	5	5	4.5
	Subject	1	2	3	4	5	6	7
1	1	0.00	1.00	9.00	36.00	16.00	16.00	12.25
2	2	1.00	0.00	4.00	25.00	9.00	9.00	6.25
4	3	9.00	4.00	0.00	9.00	1.00	1.00	0.25
7	4	36.00	25.00	9.00	0.00	4.00	4.00	6.25
5	5	16.00	9.00	1.00	4.00	0.00	0.00	0.25
5	6	16.00	9.00	1.00	4.00	0.00	0.00	0.25
4.5	7	12.25	6.25	0.25	6.25	0.25	0.25	0.00

Euclidean								
	Subject	1	2	3	4	5	6	7
	1	0.00	1.12	3.61	7.21	4.72	5.32	4.30
	2	1.12	0.00	2.50	6.10	3.61	4.24	3.20
	3	3.61	2.50	0.00	3.61	1.12	1.80	0.71
	4	7.21	6.10	3.61	0.00	2.50	2.06	2.92
	5	4.72	3.61	1.12	2.50	0.00	1.00	0.50
	6	5.32	4.24	1.80	2.06	1.00	0.00	1.12
	7	4.30	3.20	0.71	2.92	0.50	1.12	0.00

SELECT THE BEST K: SILHOUETTE

- Compute average of Silhouette (Example: K=2)

Subject	A	B	a(i)	d(i,2)	b(i)	s(i)	Distance C1	Distance C2
1	1	1	1.12	5.03	5.03	0.78	0.56	5.02
2	1.5	2	1.12	3.93	3.93	0.72	0.56	3.92
3	3	4	1.81	3.05	3.05	0.41	3.05	1.42
4	5	7	2.77	6.66	6.66	0.58	6.66	2.20
5	3.5	5	1.28	4.16	4.16	0.69	4.16	0.41
6	4.5	5	1.50	4.78	4.78	0.69	4.78	0.61
7	3.5	4.5	1.31	3.75	3.75	0.65	3.75	0.72

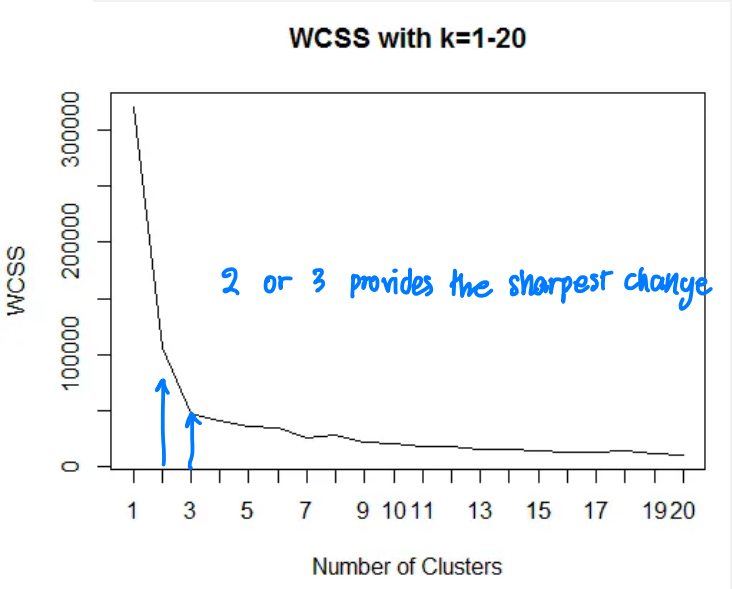
	A	B
Centroid 1	1.25	1.50
Centroid 2	3.90	5.10

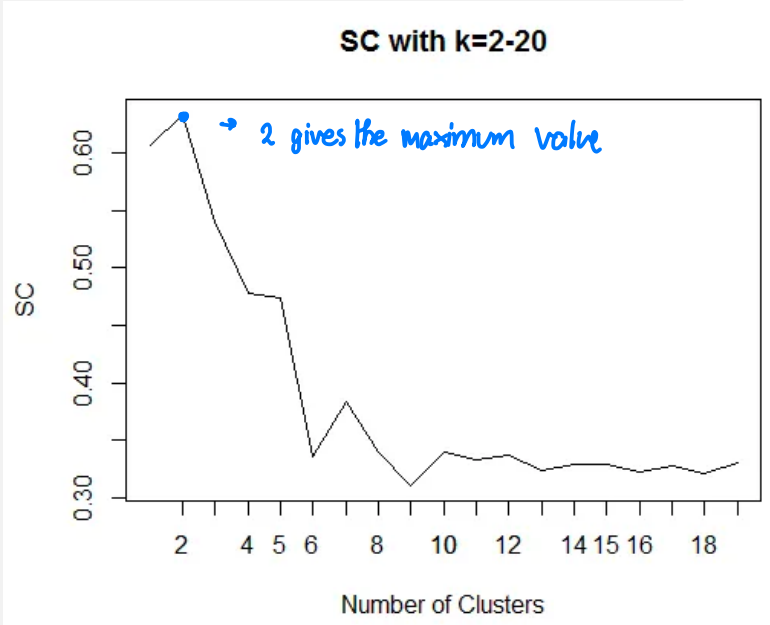
Re-compute centroids

	A	B
Centroid 1	1.25	1.50
Centroid 2	3.90	5.10

AVG of Silhouette 0.65

EXE 1: IDENTIFY K FOR K-MEANS





☆ midterm: given elbow graph, explain the best K
- ↳ balance 2 criteria, instead of only using one score
- ↳ best evaluators
  - ↳ regression : MAPE
    clustering : Accuracy
    k-means : Silhouette score

Conclusion : you should use 2 clusters

PROS AND CONS

- Pros:
  - Easy to implement
  - Low complexity $O(nkt), where\ t = \#\ of\ iterations$
- Con
  - Necessity of specifying k
  - Sensitive to noise and outlier data points
  - Sensitive to initial assignment of centroids
  - No guarantee to find a globally optimal solution

PROCEDURE 1

- Load data:
- Explore data:
  - Size (col & row)
  - Quality

PROCEDURE 2

- Preprocess data:
  - Split train-test if you want to check the impact of k
  - Encoding data columns
  - Fill in missing value in train dataset
  - Scale features in train dataset
  - Select features (only numerical) in train dataset
- Model:
  - Debug KMeans model to make the model work
  - Preprocess test dataset same as train dataset
  - Optimize k, based on elbow methods

K-means: predict without y

HW7: APPLY KNN AND K-MEANS ON IMDB

- movie_title : Title of the Movie
- duration: Duration in minutes
- director_name : Name of the Director of the Movie.
- director_facebook_likes : Number of likes of the Director on his Facebook Page.
- color: Film colorization. 'Black and White' or 'Color'
- genres: Film categorization like 'Animation', 'Comedy', 'Romance', 'Horror', 'Sci-Fi', 'Action', 'Family'
- actor_1_name: Primary actor starring in the movie
- actor_1_facebook_likes : Number of likes of the Actor_1 on his/her Facebook Page.
- actor_2_name: Other actor starring in the movie
- actor_2_facebook_likes : Number of likes of the Actor_2 on his/her Facebook Page.
- actor_3_name: Other actor starring in the movie
- actor_3_facebook_likes : Number of likes of the Actor_3 on his/her Facebook Page.
- num_critic_for_reviews : Number of critical reviews on imdb
- num_voted_users: Number of people who voted for the movie
- cast_total_facebook_likes: Total number of facebook likes of the entire cast of the movie.
- language : English, Arabic, Chinese, French, German, Danish, Italian, Japanese etc
- country: Country where the movie is produced.
- gross: Gross earnings of the movie in Dollars
- budget: Budget of the movie in Dollars
- title_year: The year in which the movie is released (1916:2016)
- imdb_score: IMDB Score of the movie on IMDB
- movie_facebook_likes: Number of Facebook likes in the movie page.

HW7: SOLUTION STEPS 1

- Load data: Merge & Concatenation
- Explore data:
  - Head()/tail()
  - info()→data_type & size
  - describe() & value_counts() plot → distribution
  - isna()→missing value

HW7: SOLUTION STEPS 2

- Preprocess data:
  - Encoding data features:
    - Integer → Float; ~~Categorical → Binary or Label Values~~
  - Fill in missing value in train dataset
  - Scale features (comparable units) in train dataset
  - Select features in train dataset
    - Only numerical features
    - Correlation – use correlation matrix or heat map to ignore x which is highly correlated with another x

HW7: SOLUTION STEPS 3

- Model:
  - KNN for classification
    - Gridsearchcv for best K with train dataset
    - Evaluate classification result of test dataset
  - KMeans for clustering
    - Find the best K based by elbow methods
    - Set k=2, and run KMeans again
    - Use contingency_matrix to compare KMeans result with y

REFERENCES

- MacQueen, J. (1967, June). Some methods for classification and analysis of multivariate observations. In Proceedings of the fifth Berkeley symposium on mathematical statistics and probability (Vol. 1, No. 14, pp. 281-297).
- Rousseeuw, P. J. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. Journal of computational and applied mathematics, 20, 53-65.