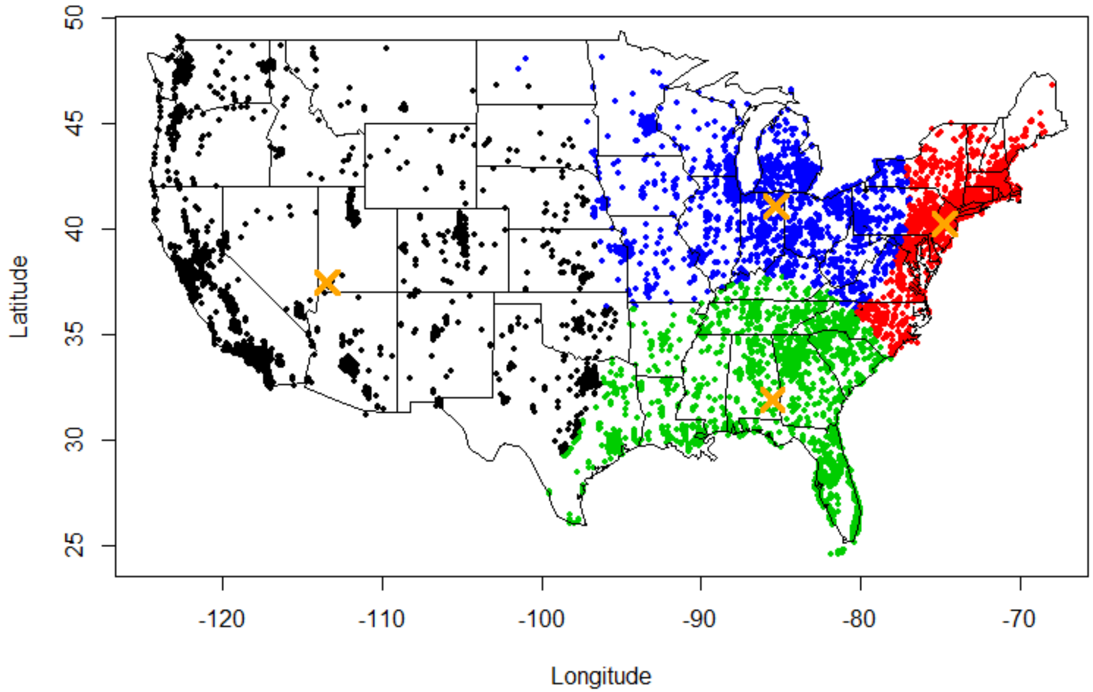


Chapter 10. Cluster Analysis: K-Partitioning

Meng Jiang

Data Science

Outline

- Basic Concepts of K-Partitioning Methods
- The **K-Means** Clustering Method
- Initialization of K-Means Clustering
- The **K-Medoids** Clustering Method
- The **K-Medians** Clustering Method
- The **K-Modes** Clustering Method
- The **Kernel K-Means** Clustering Method

Outline

- Basic Concepts of K-Partitioning Methods
- The **K-Means** Clustering Method
- Initialization of K-Means Clustering
- The **K-Medoids** Clustering Method
- The **K-Medians** Clustering Method
- The **K-Modes** Clustering Method
- The **Kernel K-Means** Clustering Method

Review: Clustering Task

Let *D* denote a dataset containing *N* **data objects**

$$D = \{\mathbf{x}_i \mid i = 1, 2, \ldots, N\}$$

where each $\mathbf{x}_i$ corresponds to the set of **features** of the *i*-th **data object**. **Clustering** is the task of learning a mapping of each **feature** set $\mathbf{x}$ into a previously undefined grouping.

Basic Concepts

- Partitioning method: Discovering the groupings in the data by optimizing a **specific objective function** and **iteratively** improving **the quality of partitions**

Basic Concepts

- Partitioning method: Discovering the groupings in the data by optimizing a **specific objective function** and **iteratively** improving **the quality of partitions**
- *K*-partitioning method: Partitioning a dataset *D* of *n* objects into a set of *K* clusters so that an objective function is optimized (e.g., the *sum of squared distances* is **minimized**, where c_k is the **centroid** or **medoid** of cluster C_k)

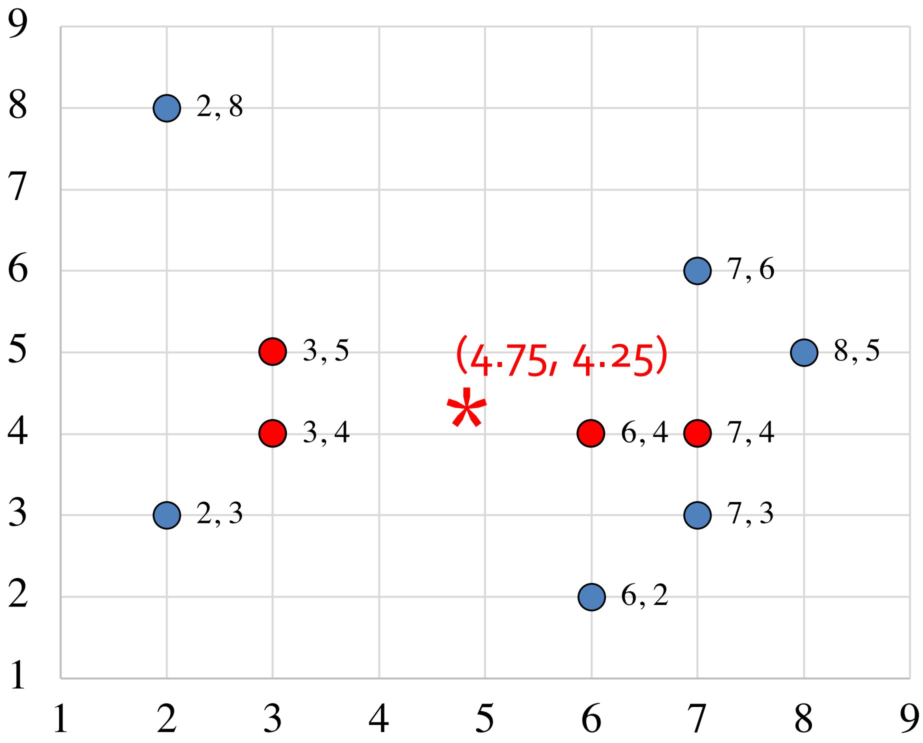
$$SSE(C) = \sum_{k=1}^{K} \sum_{x_{i \in C_k}} \| x_i - c_k \|^2$$

Centroid

Given a cluster of data objects C_k, the centroid c_k is the mean position of all C_k's objects in all of the features.

Suppose the cluster has 4 data objects:
(3, 5), (3, 4)
(6, 4), (7, 4)

So the centroid point is
((3+3+6+7)/4, (5+4+4+4)/4)
= (4.75, 4.25)



Medoid

Given a cluster of data objects C_k, the medoid c_k is the object of C_k whose average distance/dissimilarity in the cluster is minimal.

We use Manhattan distance. Distance matrix:

	(3,5)	(3,4)	(6,4)	(7,4)
(3,5)	0	1	4	5
(3,4)	1	0	3	4
(6,4)	4	3	0	1
(7,4)	5	4	1	0

Average distance:

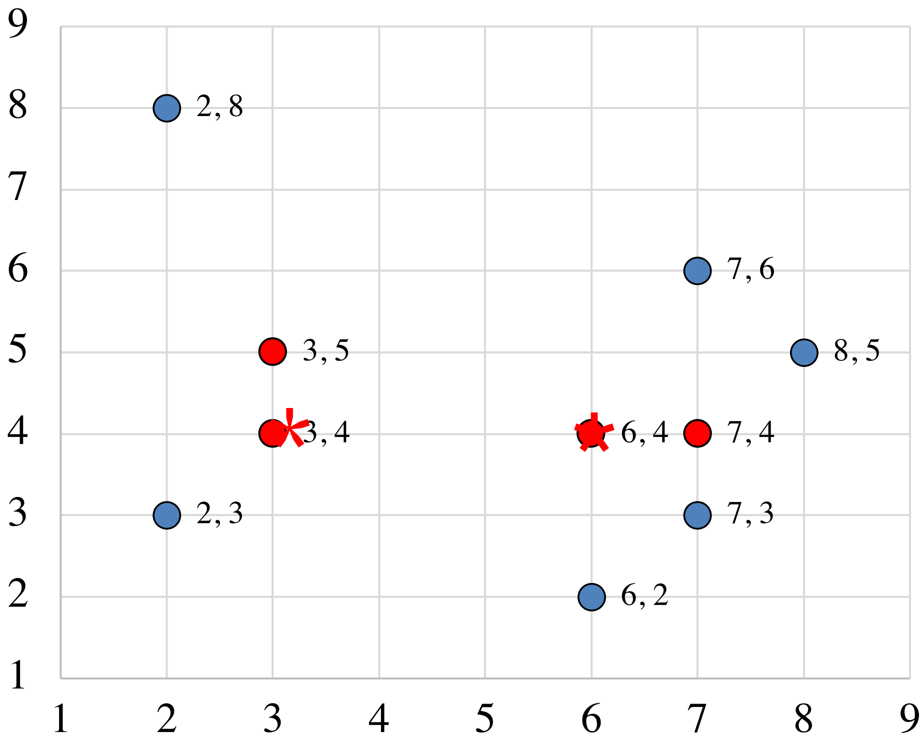
(3,5) : (0+1+4+5)/4 = 2.5

(3,4) : (1+0+3+4)/4 = 2 → minimal

(6,4) : (4+3+0+1)/4 = 2 → minimal

(7,4) : (5+4+1+0)/4 = 2.5

medoid



Median

Given a cluster of data objects C_k, the median point c_k is the median position of all C_k's objects in all of the features.

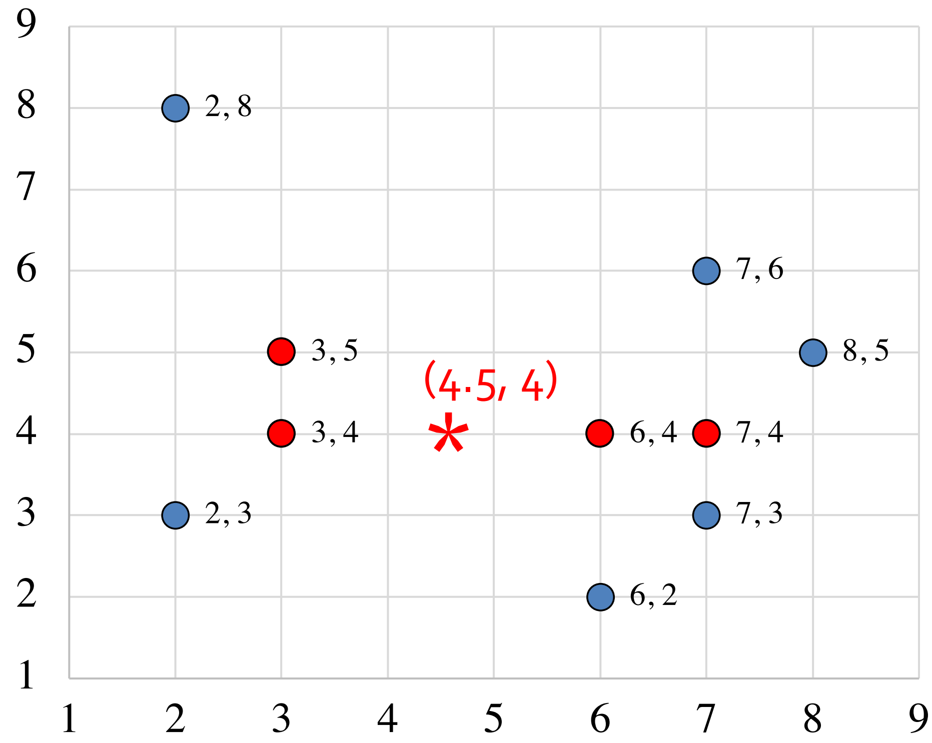
Suppose the cluster has three data objects:
(3, 5), (3, 4)
(6, 4), (7, 4)

Sorted feature values:

3, **3**, **6**, 7

4, **4**, **4**, 5

So the median point is (4.5, 4)

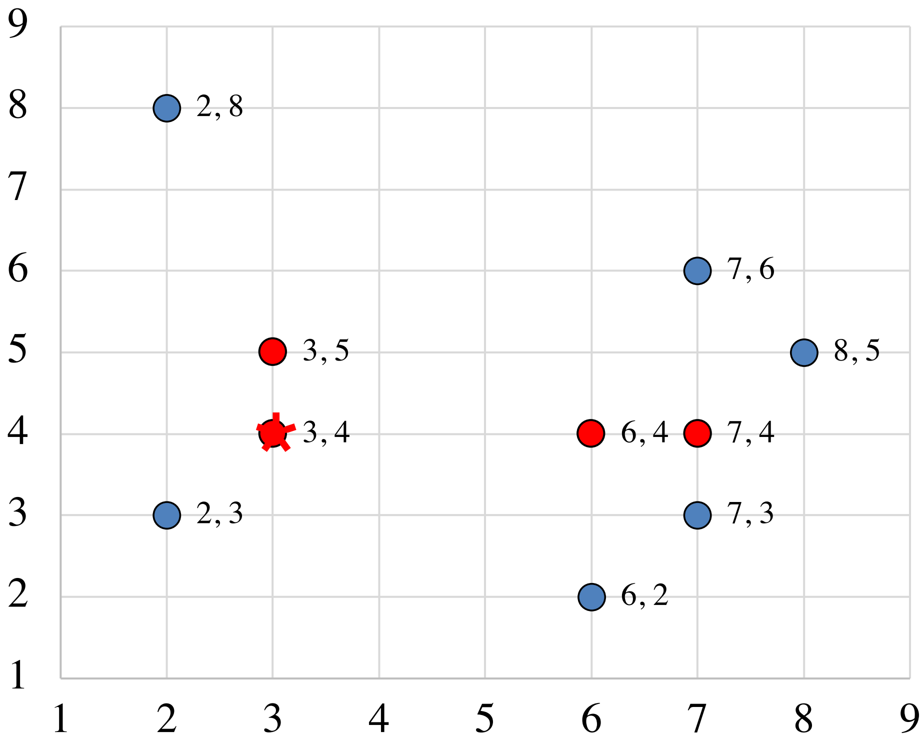


Mode

Given a cluster of data objects C_k, the mode point c_k is the "mode" (most frequent) position of all C_k's objects in all of the features.

Suppose the cluster has three data objects:
(3, 5), (3, 4)
(6, 4), (7, 4)

Sorted feature values:
3, 3, 6, 7
4, 4, 4, 5
So the mode point is
(3, 4)



Problem Definition

- Given *K*, find a partition of *K clusters* that optimizes the chosen partitioning criterion
  - Global optimal: Needs to exhaustively enumerate all partitions

$$SSE(C) = \sum_{k=1}^{K} \sum_{x_{i \in C_k}} \| x_i - c_k \|^2$$

  - Heuristic methods (i.e., greedy algorithms): *K-Means*, *K-Medoids*, *K-Medians*, *K-Modes*, etc.

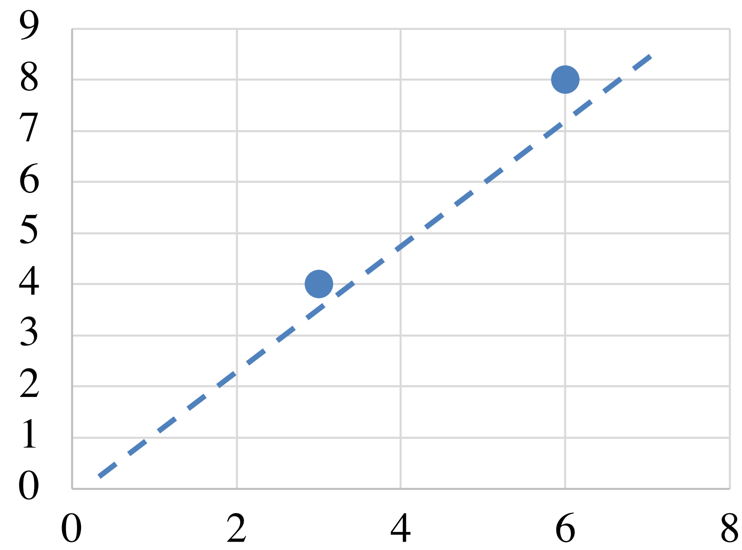
Outline

- Basic Concepts of K-Partitioning Methods
- The **K-Means** Clustering Method
- Initialization of K-Means Clustering
- The **K-Medoids** Clustering Method
- The **K-Medians** Clustering Method
- The **K-Modes** Clustering Method
- The **Kernel K-Means** Clustering Method

K-Means Clustering

- Given *K*, the number of clusters, the *K-Means* clustering algorithm is outlined as follows
  - Select *K* points as initial centroids
  - **Repeat**
    - Form *K* clusters by assigning each data object to its *nearest* centroid using a *distance metric*
    - Move each centroid to the mean of its assigned data objects (i.e., re-compute the centroid of each cluster)
  - **Until** convergence
    - Change in cluster assignment less than a threshold

Distance Metrics



Given two points (3, 4) and (6, 8)

- **Manhattan distance (L_1 norm)**

$|3-6| + |4-8| = 3+4 = 7$

- **Euclidean distance (L_2 norm)**

$((3-6)^2 + (4-8)^2)^{1/2} = 5$

- **Supreme distance or Chebyshev distance (L_∞ norm)**

$\max\{|3-6|, |4-8|\} = 4$

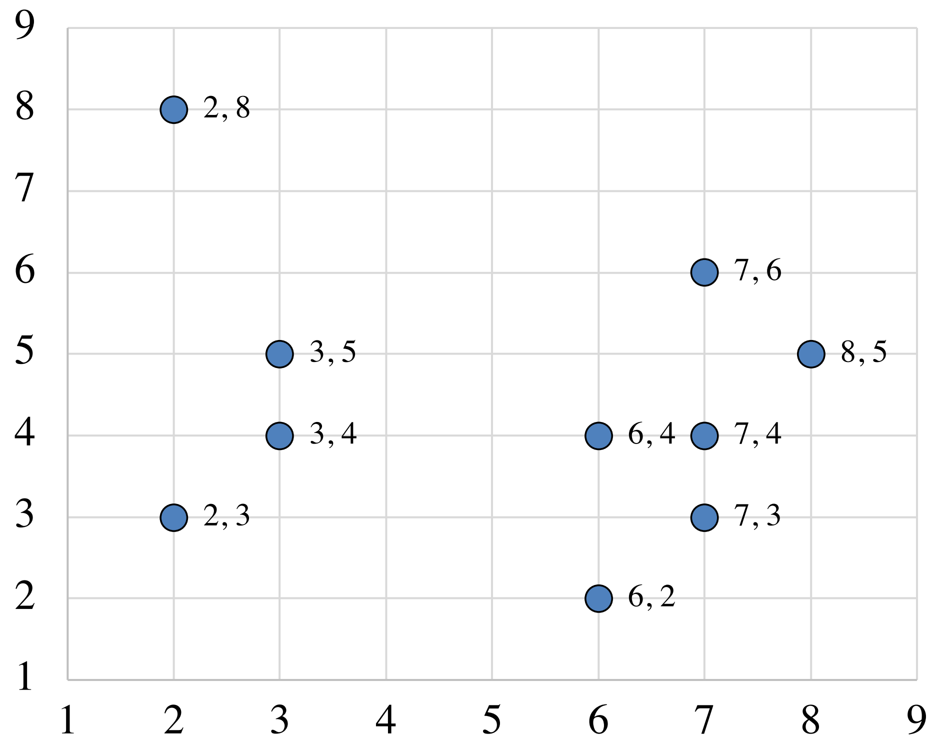
- **1 - Cosine similarity**

normalized: (3/5, 4/5) = (0.6, 0.8), (6/10, 8/10) = (0.6, 0.8)

1 - (0.6*0.6+0.8*0.8) = 0

Data Objects

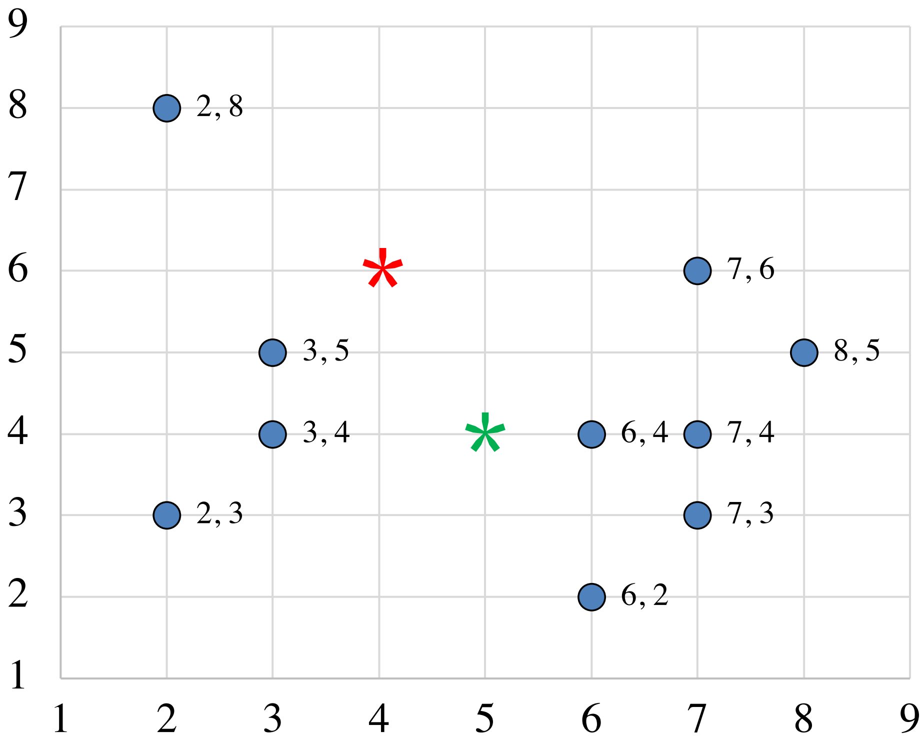
X1	3	5
X2	3	4
X3	2	8
X4	2	3
X5	6	2
X6	6	4
X7	7	3
X8	7	4
X9	8	5
X10	7	6



Q: Suppose we want two clusters… What are they?

Initialize Centroids

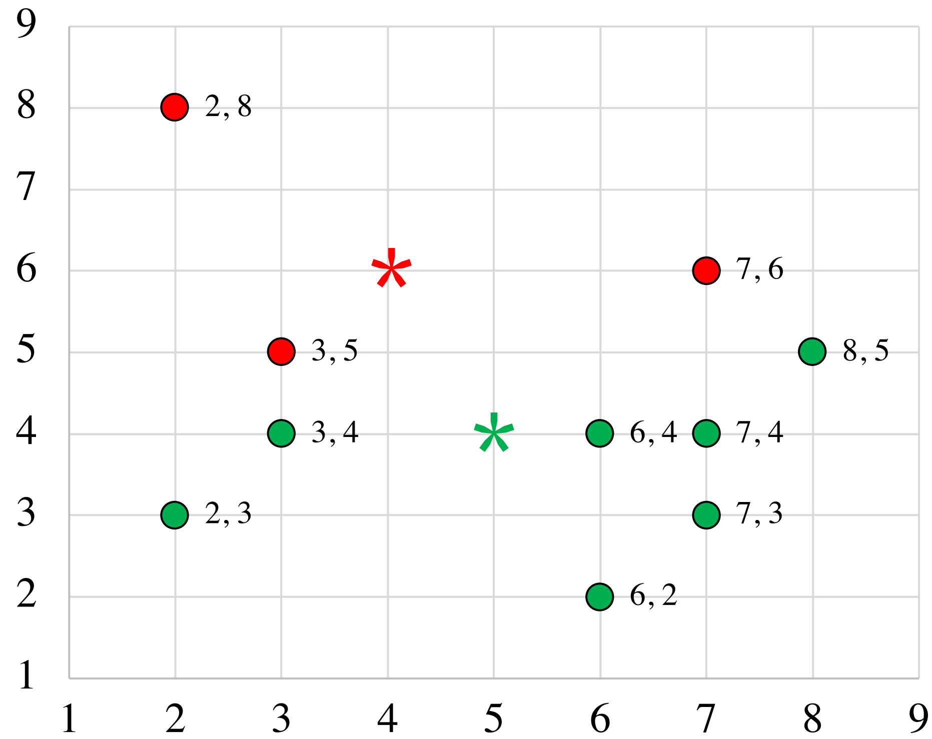
- K = 2
- (4, 6)*
- (5, 4)*



Assign Object to Nearest Centroid

- Manhattan distance

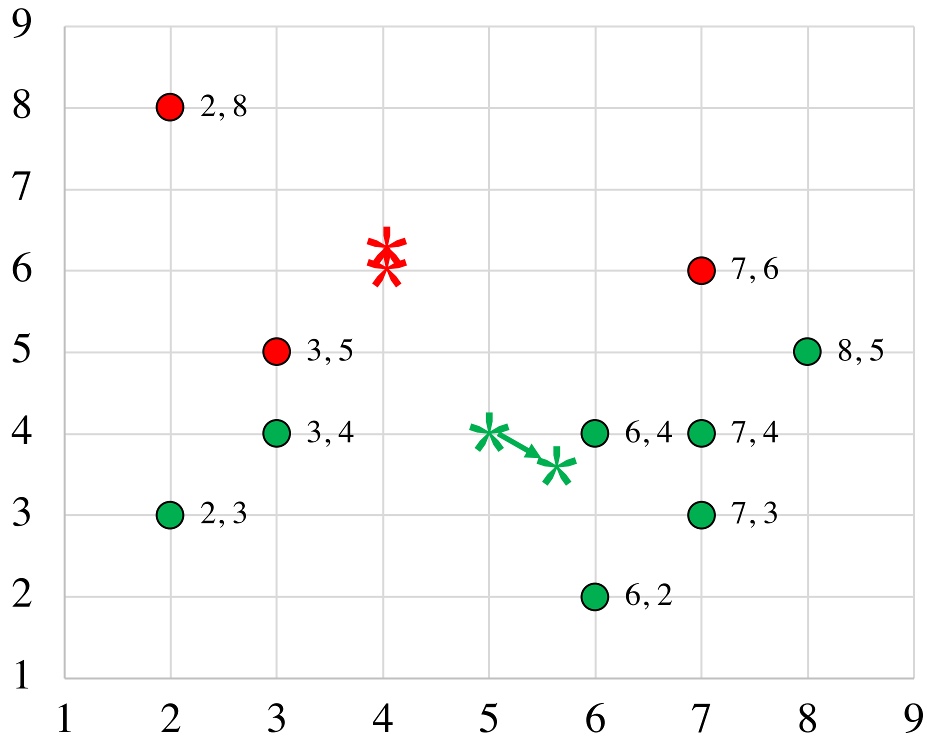
			(4, 6)	(5, 4)
X1	3	5	**2**	3
X2	3	4	3	**2**
X3	2	8	**4**	7
X4	2	3	5	**4**
X5	6	2	6	**3**
X6	6	4	4	**1**
X7	7	3	6	**3**
X8	7	4	5	**2**
X9	8	5	5	**4**
X10	7	6	**3**	4



Move the Centroids

X1	3	5
X3	2	8
X10	7	6
(4, 6)	4	6.33

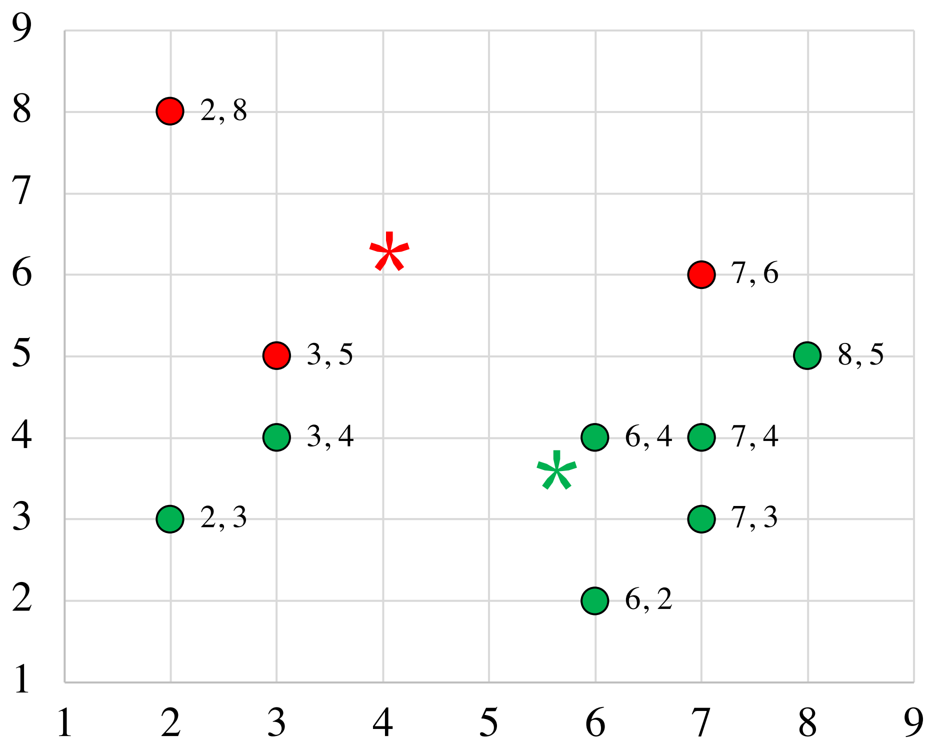
X2	3	4
X4	2	3
X5	6	2
X6	6	4
X7	7	3
X8	7	4
X9	8	5
(5, 4)	5.57	3.57



Assign Object to Nearest Centroid

- Manhattan distance

			(4, 6.33)	(5.57, 4.57)
X1	3	5	**2.33**	4
X2	3	4	3.33	**3**
X3	2	8	**3.67**	8
X4	2	3	5.33	**4.14**
X5	6	2	6.33	**2**
X6	6	4	4.33	**0.86**
X7	7	3	6.33	**2**
X8	7	4	5.33	**1.86**
X9	8	5	5.33	**3.86**
X10	7	6	**3.33**	3.86

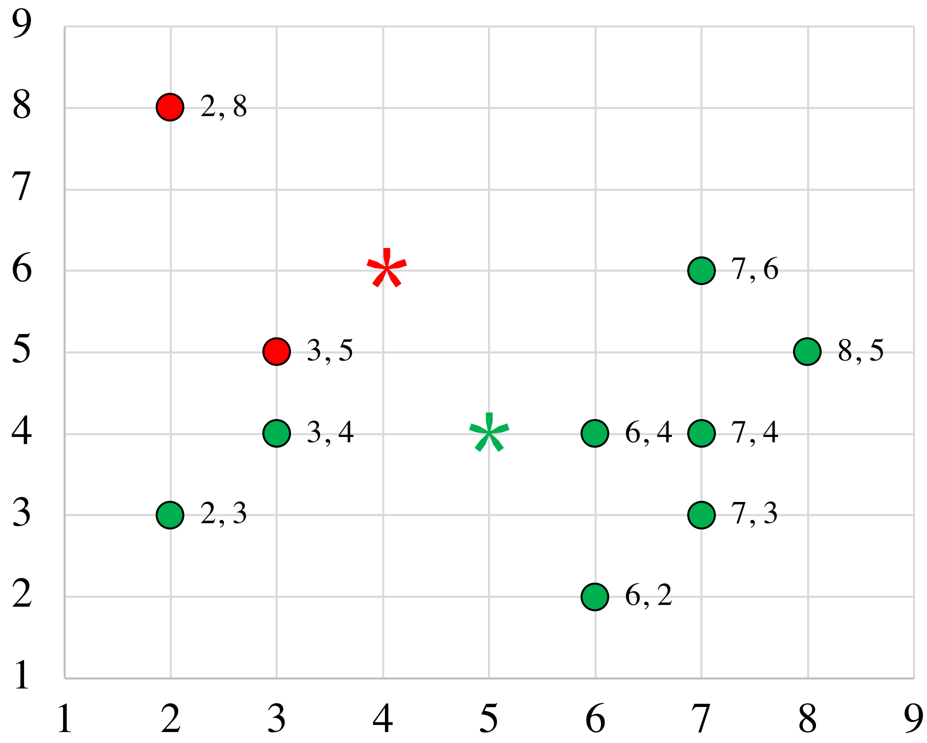


Q: Will the centroids move?

Assign Object to Nearest Centroid

- **Euclidean** distance

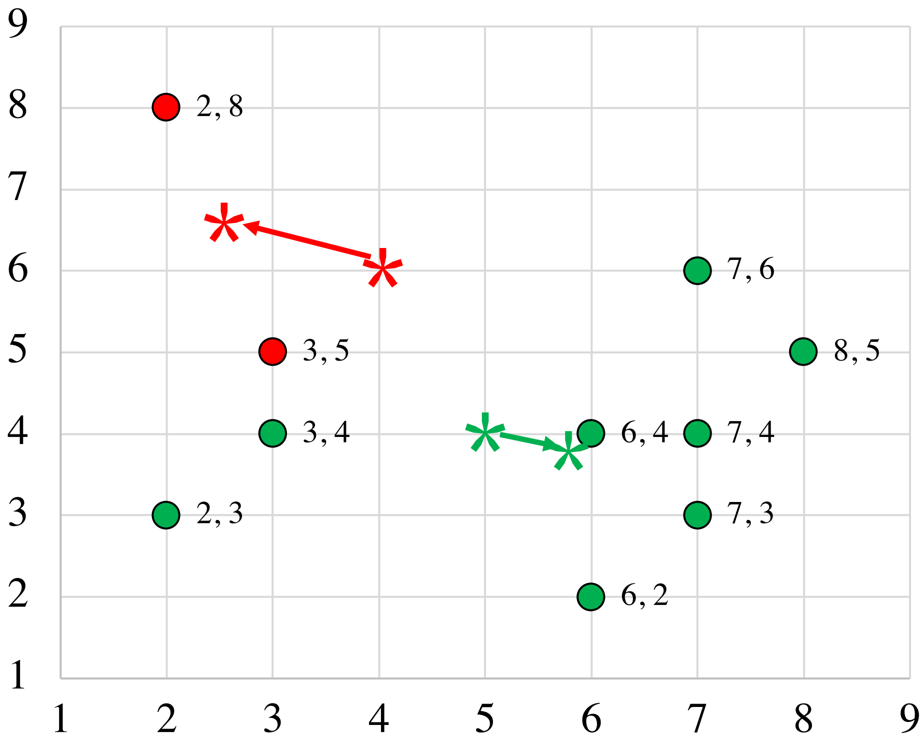
			(4, 6)	(5, 4)
X1	3	5	**1.41**	2.24
X2	3	4	2.24	**2.00**
X3	2	8	**2.83**	5.00
X4	2	3	3.61	**3.16**
X5	6	2	4.47	**2.24**
X6	6	4	2.83	**1.00**
X7	7	3	4.24	**2.24**
X8	7	4	3.61	**2.00**
X9	8	5	4.12	**3.16**
X10	7	6	3.00	**2.83**



Move the Centroids

X1	3	5
X3	2	8
(4, 6)	2.5	6.5

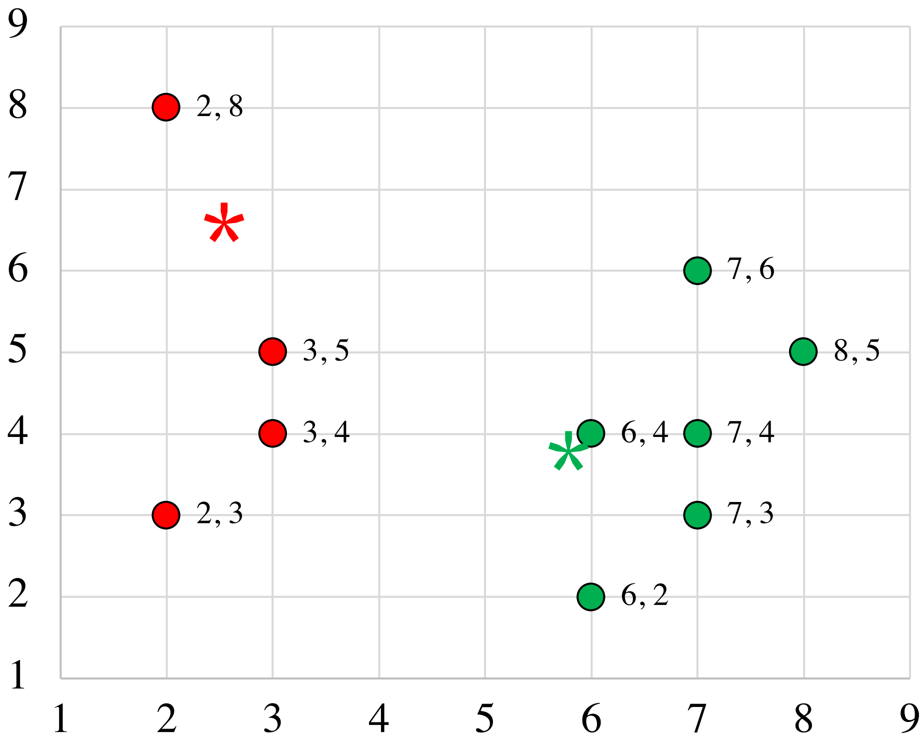
X2	3	4
X4	2	3
X5	6	2
X6	6	4
X7	7	3
X8	7	4
X9	8	5
X10	7	6
(5, 4)	5.75	3.88



Assign Object to Nearest Centroid

- **Euclidean** distance

			(2.5, 6.5)	(5.75, 3.88)
X1	3	5	**1.58**	2.97
X2	3	4	**2.55**	2.75
X3	2	8	**1.58**	5.57
X4	2	3	**3.54**	3.85
X5	6	2	5.70	**1.90**
X6	6	4	4.30	**0.28**
X7	7	3	5.70	**1.53**
X8	7	4	5.15	**1.26**
X9	8	5	5.70	**2.51**
X10	7	6	4.53	**2.46**

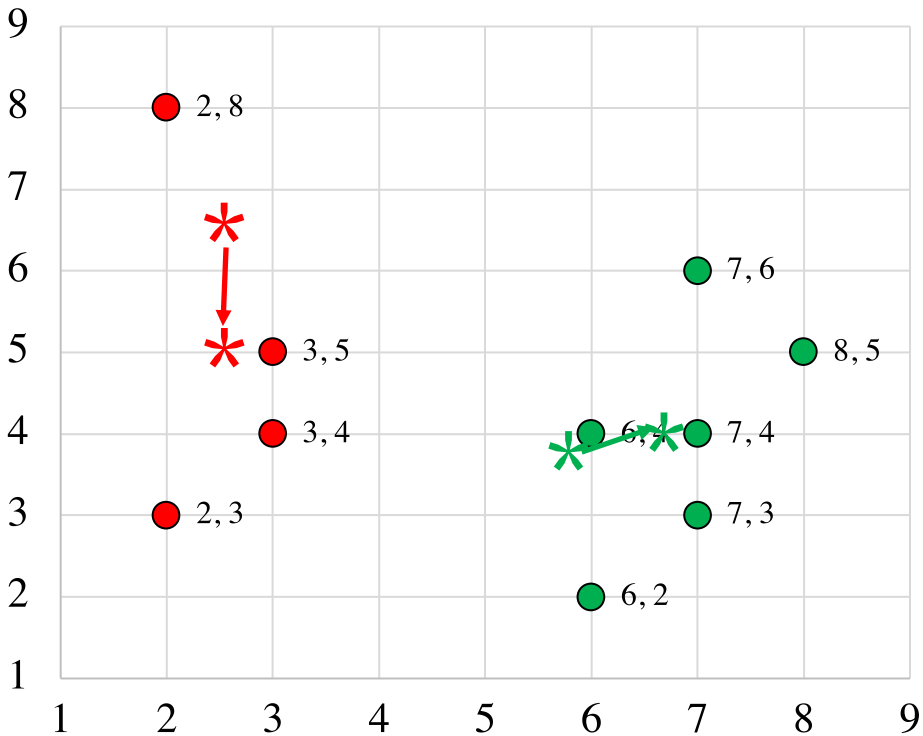


Q: Will the centroids move?

Move the Centroids

X1	3	5
X2	3	4
X3	2	8
X4	2	3
(2.5, 6.5)	2.5	5

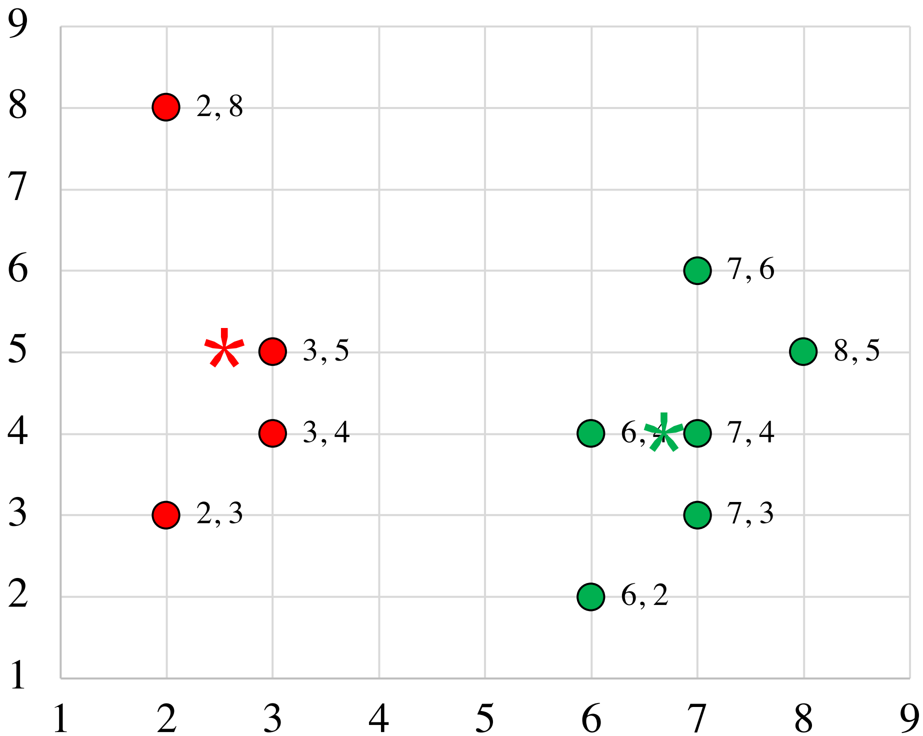
X5	6	2
X6	6	4
X7	7	3
X8	7	4
X9	8	5
X10	7	6
(5.75, 3.88)	6.83	4



Assign Object to Nearest Centroid

- **Euclidean** distance

			(2.5, 5)	(6.83, 4)
X1	3	5	**0.50**	3.96
X2	3	4	**1.12**	3.83
X3	2	8	**3.04**	6.27
X4	2	3	**2.06**	4.93
X5	6	2	4.61	**2.17**
X6	6	4	3.64	**0.83**
X7	7	3	4.92	**1.01**
X8	7	4	4.61	**0.17**
X9	8	5	5.50	**1.54**
X10	7	6	4.61	**2.01**



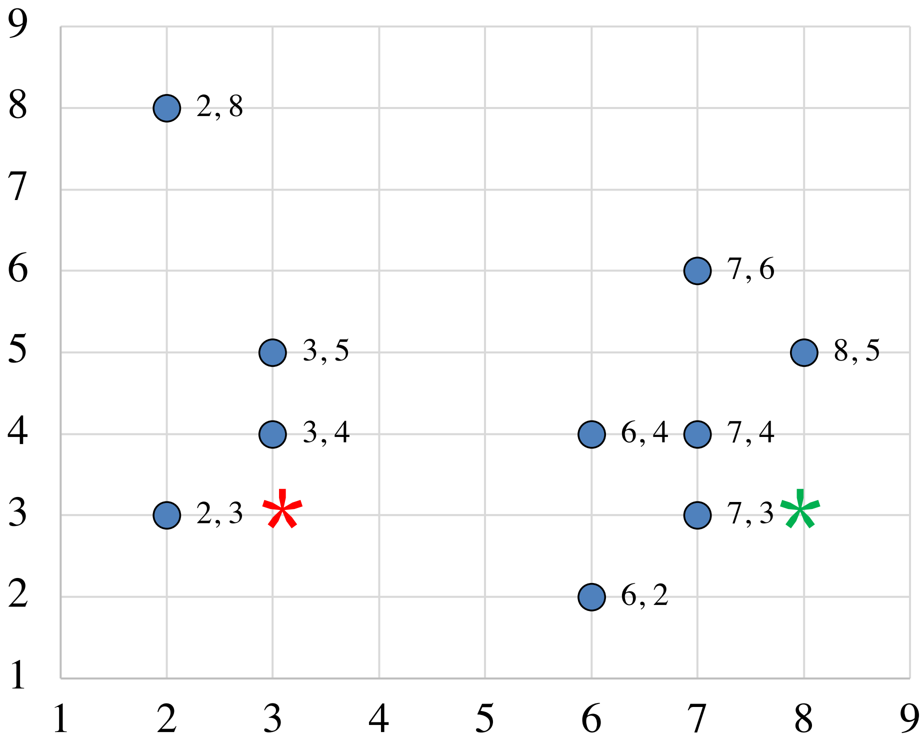
Q: Will the centroids move?

Observations

- Different distance metrics may find different K-means clustering!

Try Another Initialization

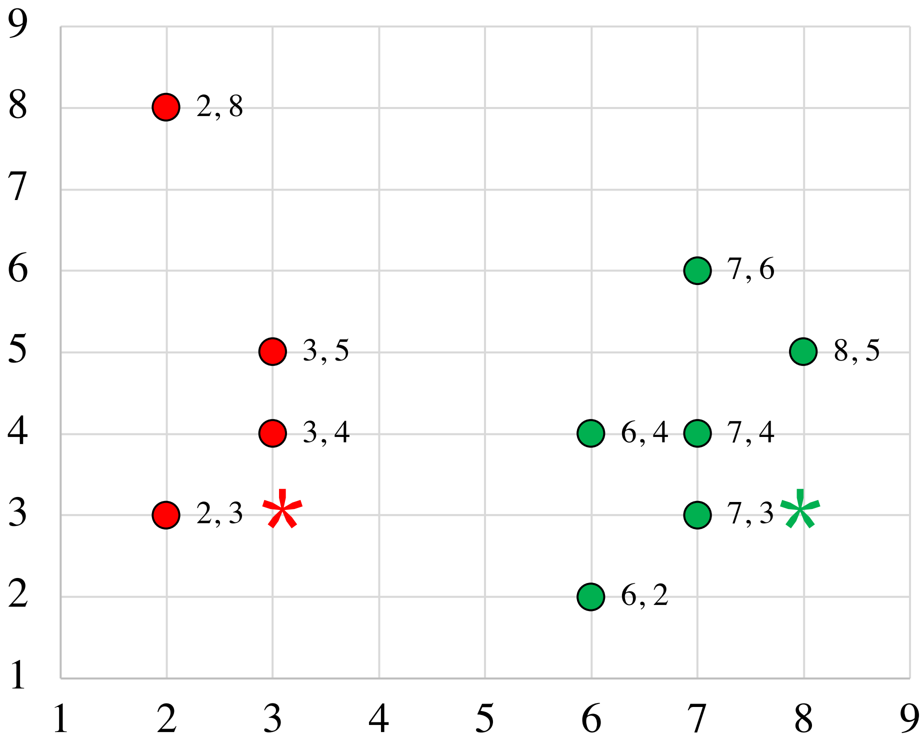
- K = 2
- (3, 3)*
- (8, 3)*



Assign Object to Nearest Centroid

- Manhattan distance

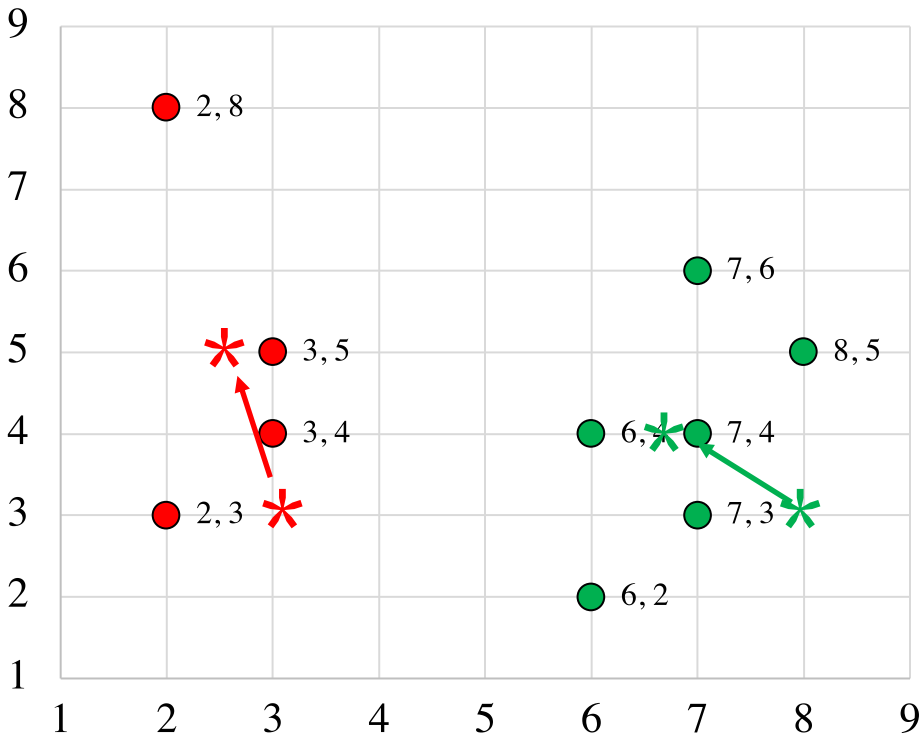
			(3, 3)	(8, 3)
X1	3	5	**2**	7
X2	3	4	**1**	6
X3	2	8	**6**	11
X4	2	3	**1**	6
X5	6	2	4	**3**
X6	6	4	4	**3**
X7	7	3	4	**1**
X8	7	4	5	**2**
X9	8	5	7	**2**
X10	7	6	7	**4**



Move the Centroids

X1	3	5
X2	3	4
X3	2	8
X4	2	3
(3, 3)	2.5	5

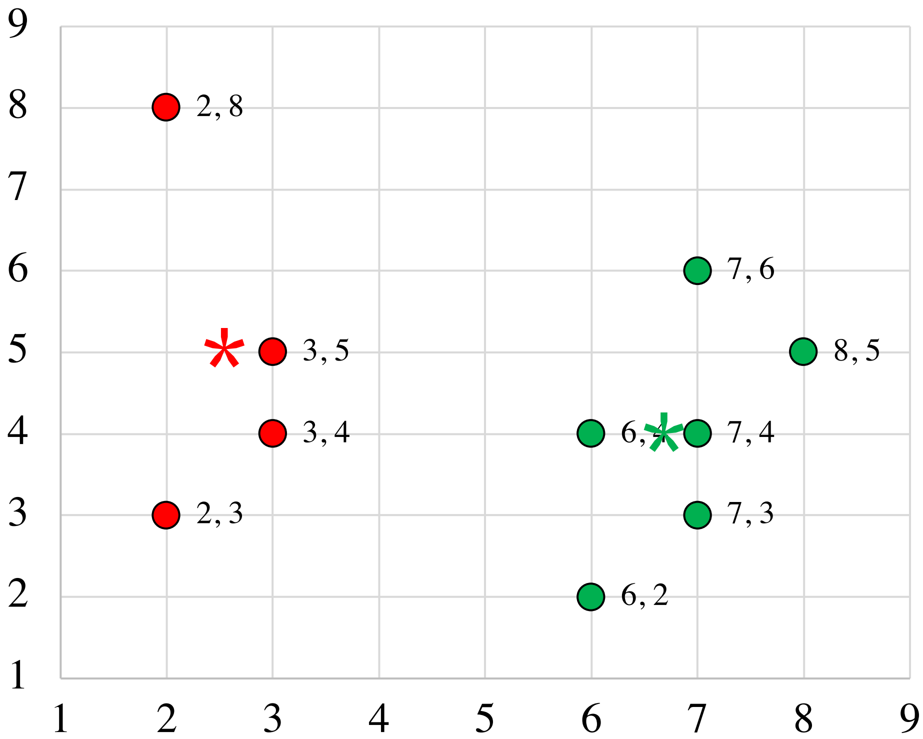
X5	6	2
X6	6	4
X7	7	3
X8	7	4
X9	8	5
X10	7	6
(8, 3)	6.83	4



Assign Object to Nearest Centroid

- Manhattan distance

			(2.5, 5)	(6.83, 4)
X1	3	5	**0.5**	4.83
X2	3	4	**1.5**	3.83
X3	2	8	**3.5**	8.83
X4	2	3	**2.5**	5.83
X5	6	2	6.5	**2.83**
X6	6	4	4.5	**0.83**
X7	7	3	6.5	**1.17**
X8	7	4	5.5	**0.17**
X9	8	5	5.5	**2.17**
X10	7	6	5.5	**2.17**



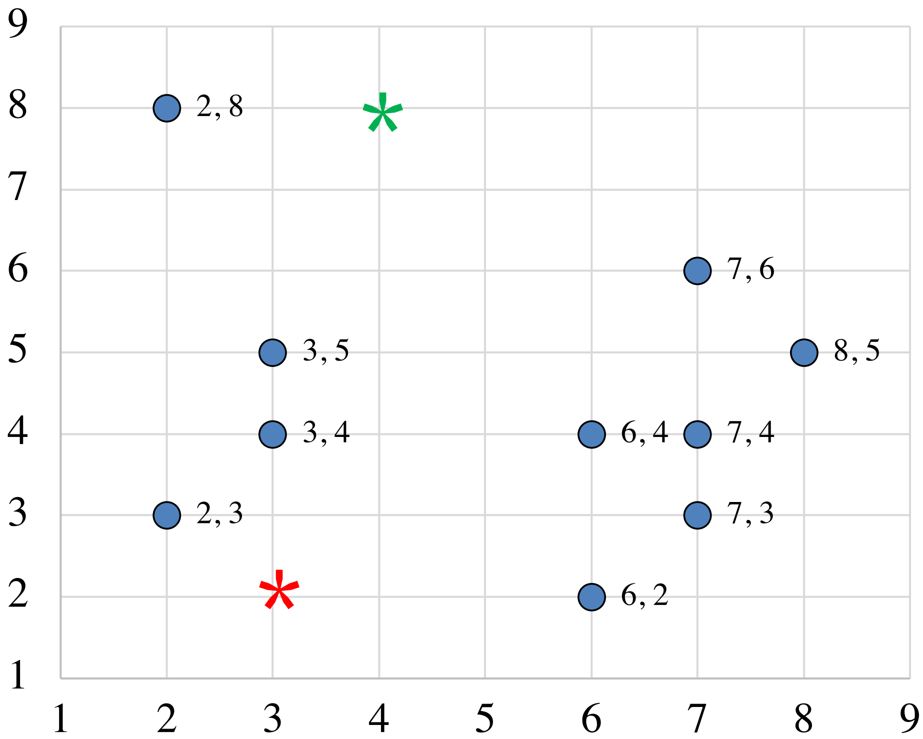
Q: Will the centroids move?

Observations

- Different distance metrics may find different K-means clustering!
- Different initialized centroids may find different clustering and may save your time!

Try One More Initialization

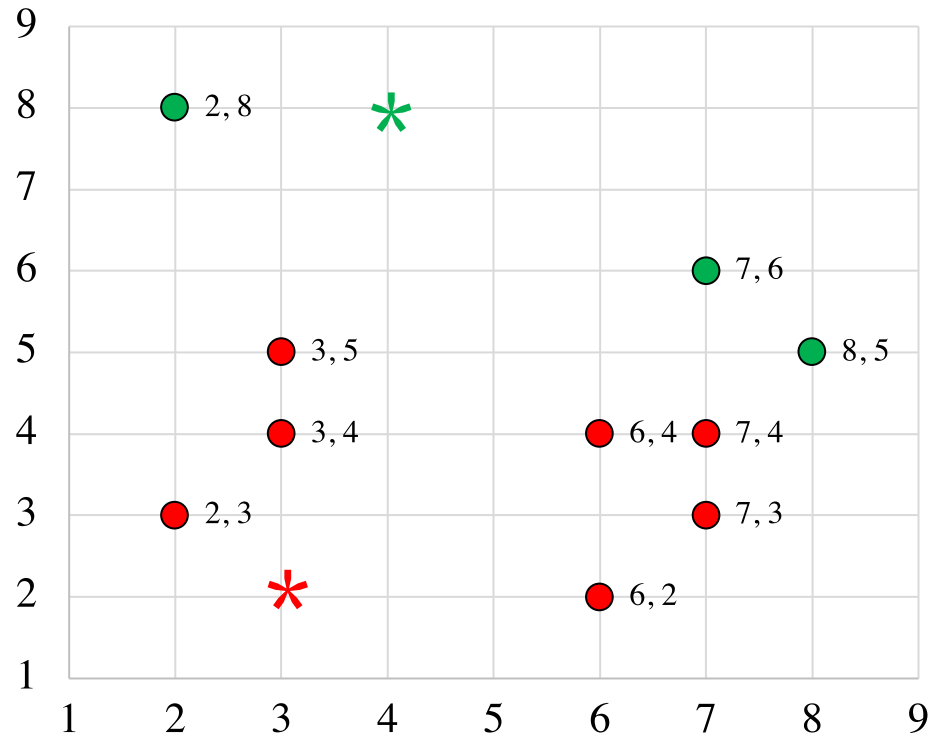
- K = 2
- (3, 2)*
- (4, 8)*



Assign Object to Nearest Centroid

- Manhattan distance

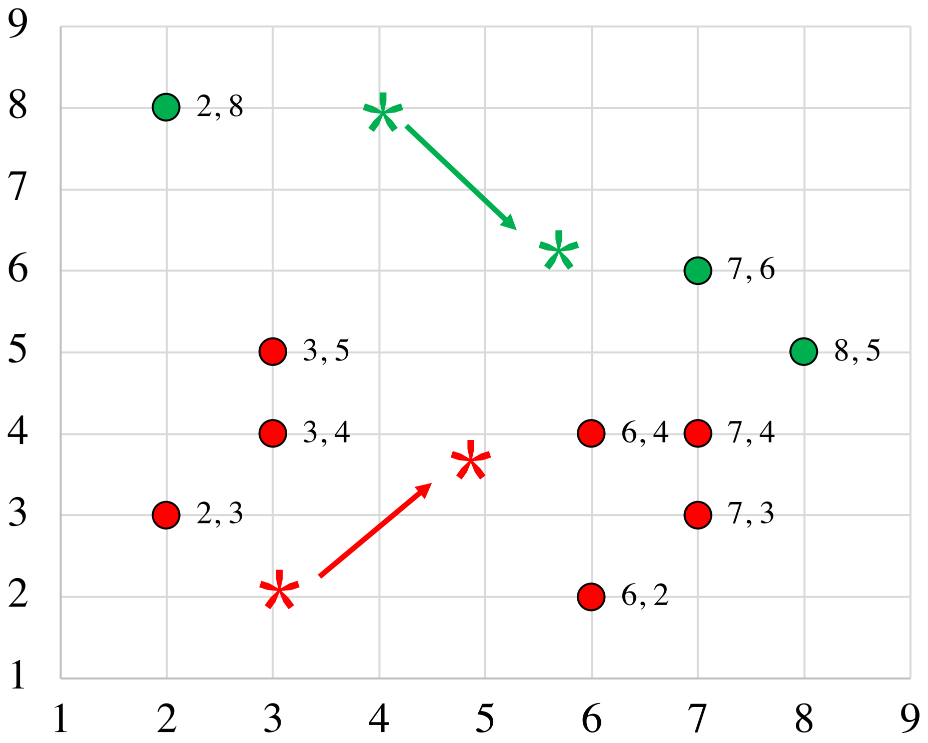
			(3, 2)	(4, 8)
X1	3	5	3	4
X2	3	4	2	5
X3	2	8	7	2
X4	2	3	2	7
X5	6	2	3	8
X6	6	4	5	6
X7	7	3	5	8
X8	7	4	6	7
X9	8	5	8	7
X10	7	6	8	5



Move the Centroids

X1	3	5
X2	3	4
X4	2	3
X5	6	2
X6	6	4
X7	7	3
X8	7	4
(3,2)	4.86	3.57

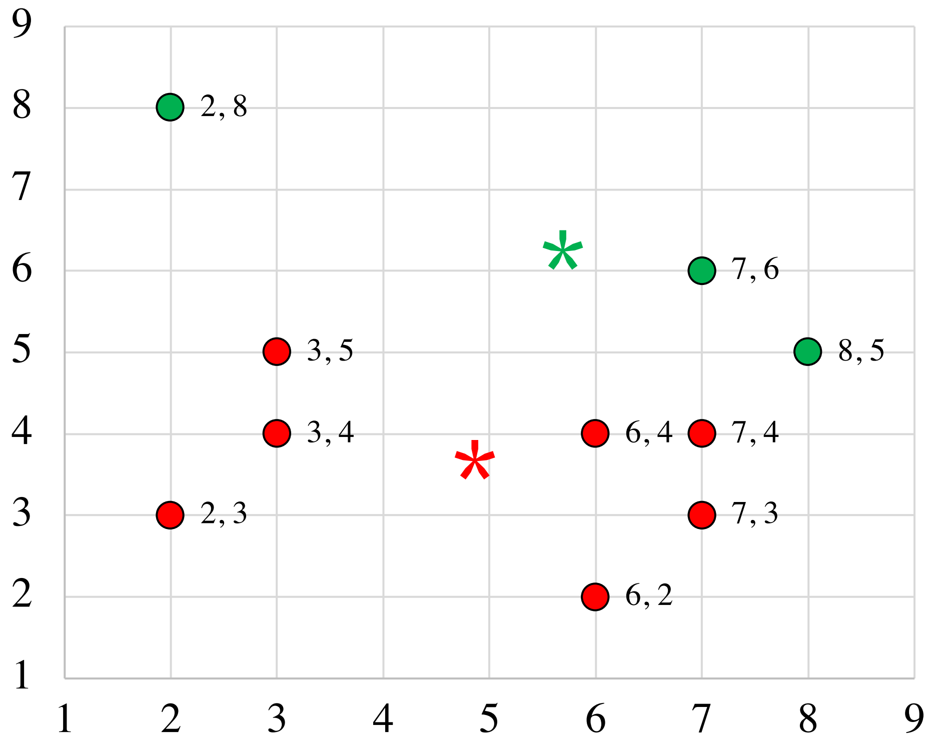
X3	2	8
X9	8	5
X10	7	6
(4,8)	5.67	6.33



Assign Object to Nearest Centroid

- Manhattan distance

			(4.86, 3.57)	(5.67, 6.33)
X1	3	5	**3.29**	4
X2	3	4	**2.29**	5
X3	2	8	7.29	5.34
X4	2	3	**3.43**	7
X5	6	2	**2.71**	4.66
X6	6	4	**1.57**	2.66
X7	7	3	**2.71**	4.66
X8	7	4	**2.57**	3.66
X9	8	5	4.57	**3.66**
X10	7	6	4.57	**1.66**



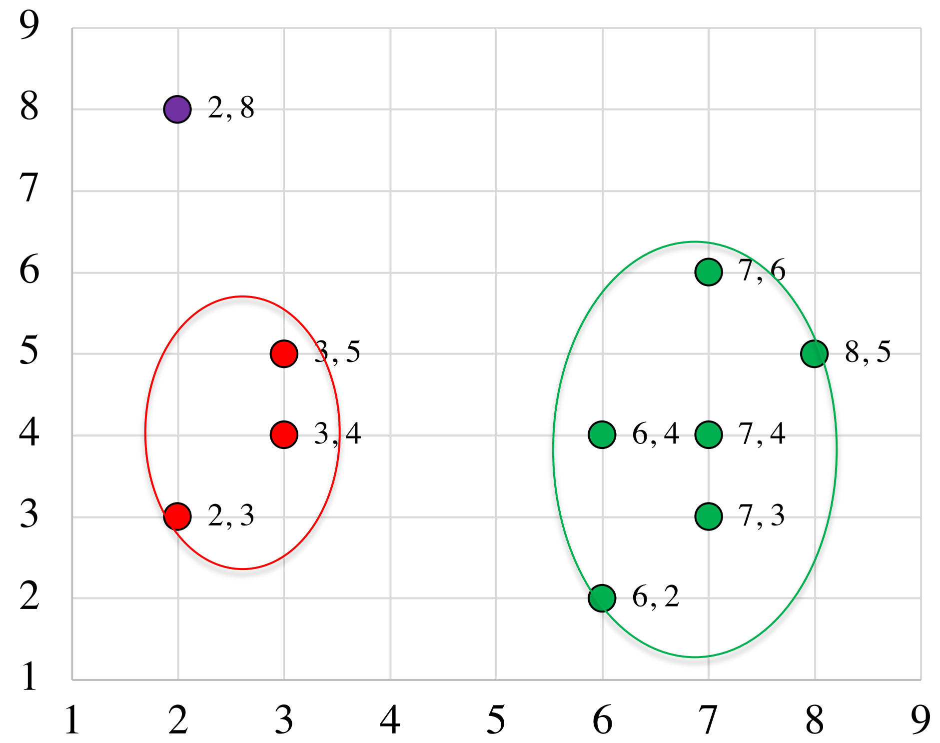
Q: Will the centroids move?

Observations

- Different distance metrics may find different K-means clustering!
- Different initialized centroids may find different clustering and may save your time!
- And maybe the different clustering makes sense!

Recall: Data Objects

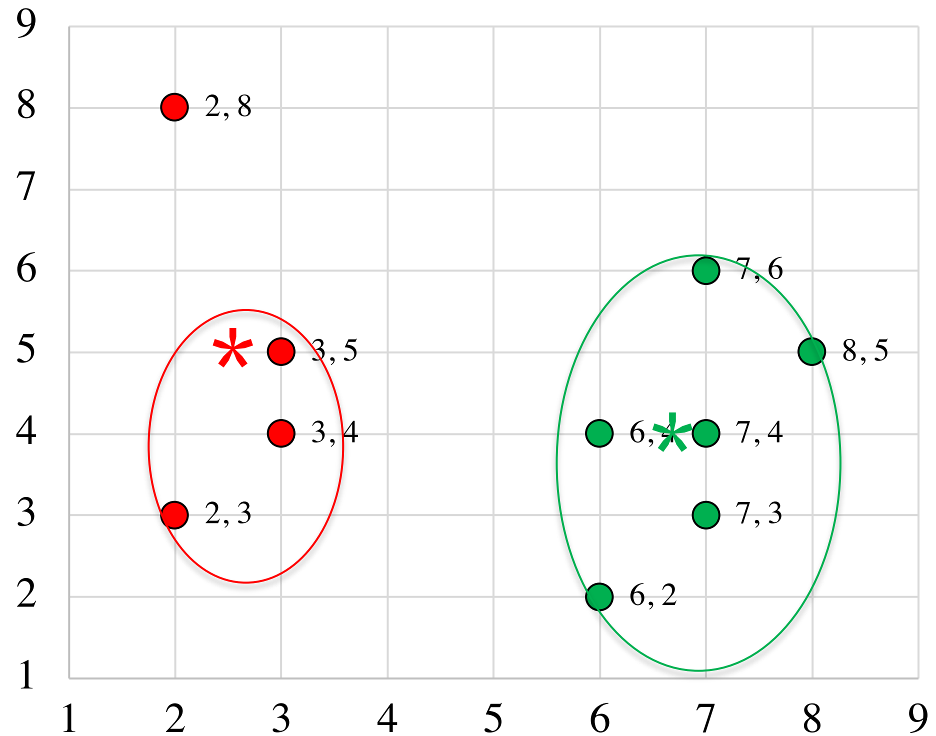
X1	3	5
X2	3	4
X3	2	8
X4	2	3
X5	6	2
X6	6	4
X7	7	3
X8	7	4
X9	8	5
X10	7	6



Ideal clusters + Outlier

Best K-Means Result

- The red centroid seems to be at the boundary, not the center, of the red cluster!



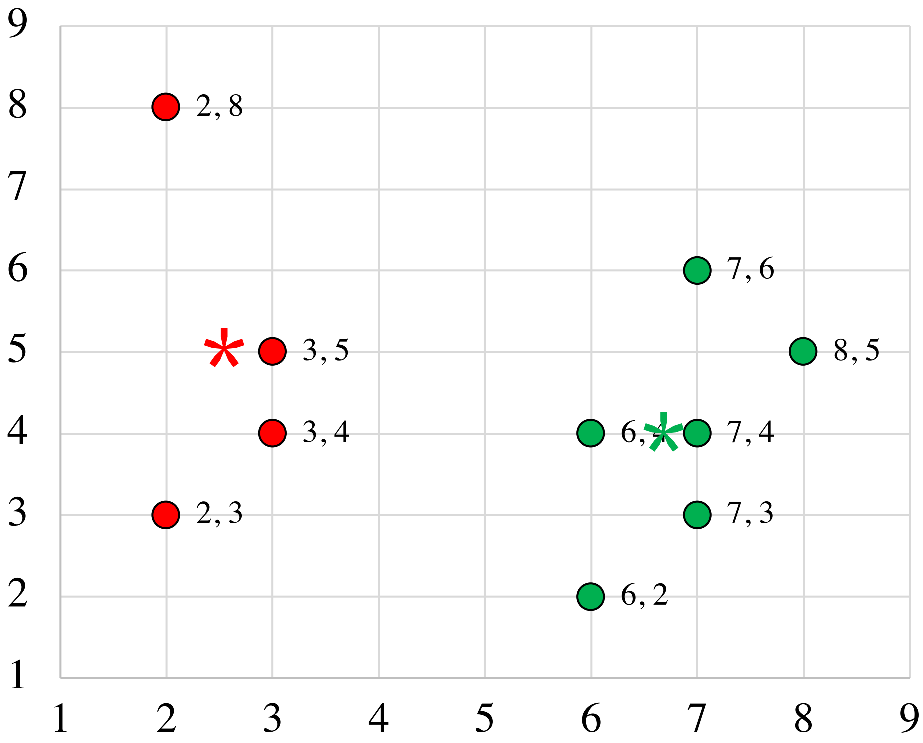
Observations

- Different distance metrics may find different K-means clustering!
- Different initialized centroids may find different clustering and may save your time!
- And maybe the different clustering makes sense!
- K-means clustering is sensitive to outliers!

Kmeans Demo

iPy…

			(2.5, 5)	(6.83, 4)
X1	3	5	**0.5**	4.83
X2	3	4	**1.5**	3.83
X3	2	8	**3.5**	8.83
X4	2	3	**2.5**	5.83
X5	6	2	6.5	**2.83**
X6	6	4	4.5	**0.83**
X7	7	3	6.5	**1.17**
X8	7	4	5.5	**0.17**
X9	8	5	5.5	**2.17**
X10	7	6	5.5	**2.17**



Advantages of K-Means Clustering

- Efficiency: O(*tKn*), where n: # of objects, K: # of clusters, and t: # of iterations
  – Normally, K, t << n; thus, an efficient method!

Disadvantages (from Observations) and Solutions

- O1/D1: Different distance metrics may find different K-means clustering!
  - Just try different metrics. Euclidean distance is consistent to the SSE. Highly recommended.

Disadvantages (from Observations) and Solutions

- O2/O3: Different initialized centroids may find different clustering and may save your time! And maybe the different clustering makes sense!
- D2: K-means clustering terminates at a local optimum
  - Initialization can be important to find high-quality clusters
- D3: Need to specify K, the number of clusters, in advance
  - There are ways to automatically determine the "best" K
  - In practice, one often runs a range of values and selected the "best" K value

Disadvantages (from Observations) and Solutions

- O4: K-means clustering is sensitive to outliers!
  - An object with an extremely large value may substantially distort the distribution of the data
- D4: Sensitive to noisy data and outliers
  - Variations: Using K-medians, K-medoids, etc.

Disadvantages and Solutions

- D5: K-means is applicable only to objects in a continuous n-dimensional space
  - Using the K-modes for categorical data
- D6: Not suitable to discover clusters with non-convex shapes
  - Using density-based clustering, kernel K-means, etc.

Summarize the Disadvantages

- Need to specify K, the number of clusters, in advance
  - There are ways to automatically determine the “best” K
  - In practice, one often runs a range of values and selected the “best” K value
- K-means clustering often terminates at a local optimum
  - Initialization can be important to find high-quality clusters
- Sensitive to noisy data and outliers
  - Variations: Using K-medoids, K-medians, etc.
- K-means is applicable only to objects in a continuous n-dimensional space
  - Using the K-modes for categorical data
- Not suitable to discover clusters with non-convex shapes
  - Using density-based clustering, kernel K-means, etc.

Outline

- Basic Concepts of K-Partitioning Methods
- The **K-Means** Clustering Method
- Initialization of K-Means Clustering
- The **K-Medoids** Clustering Method
- The **K-Medians** Clustering Method
- The **K-Modes** Clustering Method
- The **Kernel K-Means** Clustering Method

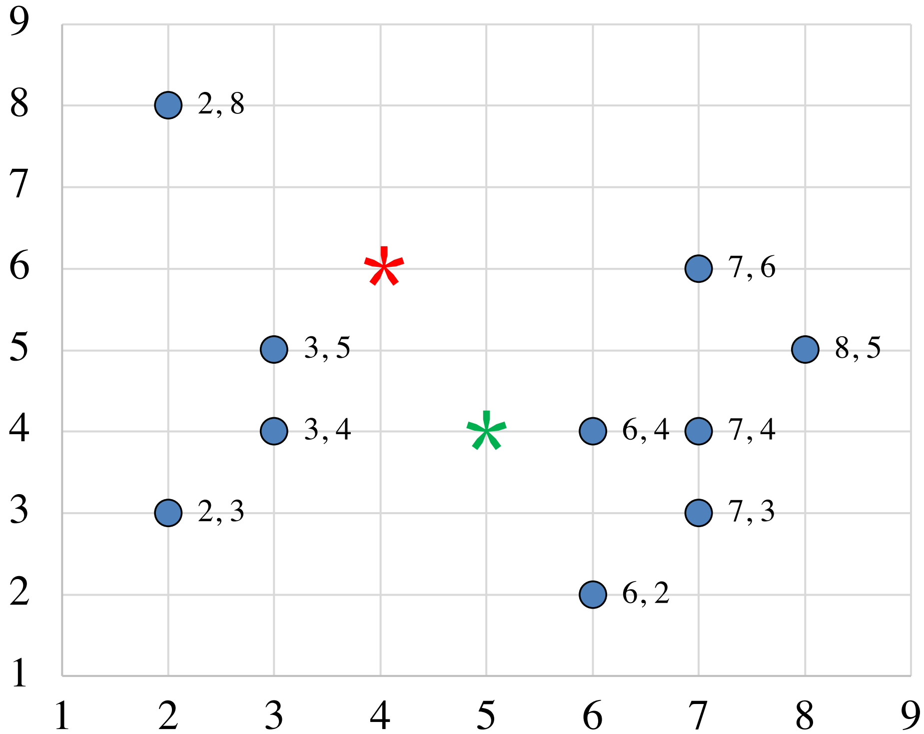
Choosing K in K-Means

- How to determine number of clusters in data?
  - Choice of K is often ambiguous!
  - Depends on scale and distribution of data
- Rule of thumb
  - K ≈ sqrt(n/2), where n is number of data objects
    - Average cluster size: sqrt(2n)
    - If n = 8, K = 2, size = 4. If K = 18, n = 3, size = 6.
  - Good starting point, but not very reliable.

Initialization

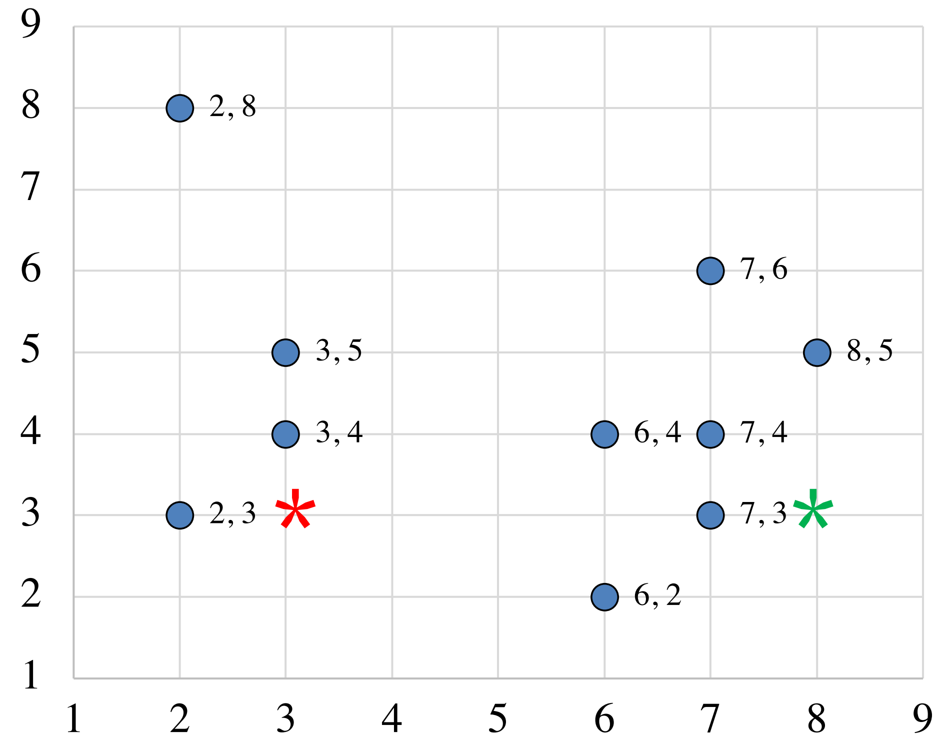
- There are many methods proposed for better initialization of k seeds
  - K-Means++ (Arthur & Vassilvitskii'07):
    - The first centroid is selected at random
    - The next centroid selected is the one that is **farthest** from the currently selected (selection is based on a weighted probability score)
    - The selection continues until K centroids are obtained

Initialization (cont.)



Manhattan ×
Euclidean √

Manhattan √
Euclidean √



Outline

- Basic Concepts of K-Partitioning Methods
- The **K-Means** Clustering Method
- Initialization of K-Means Clustering
- The **K-Medoids** Clustering Method
- The **K-Medians** Clustering Method
- The **K-Modes** Clustering Method
- The **Kernel K-Means** Clustering Method

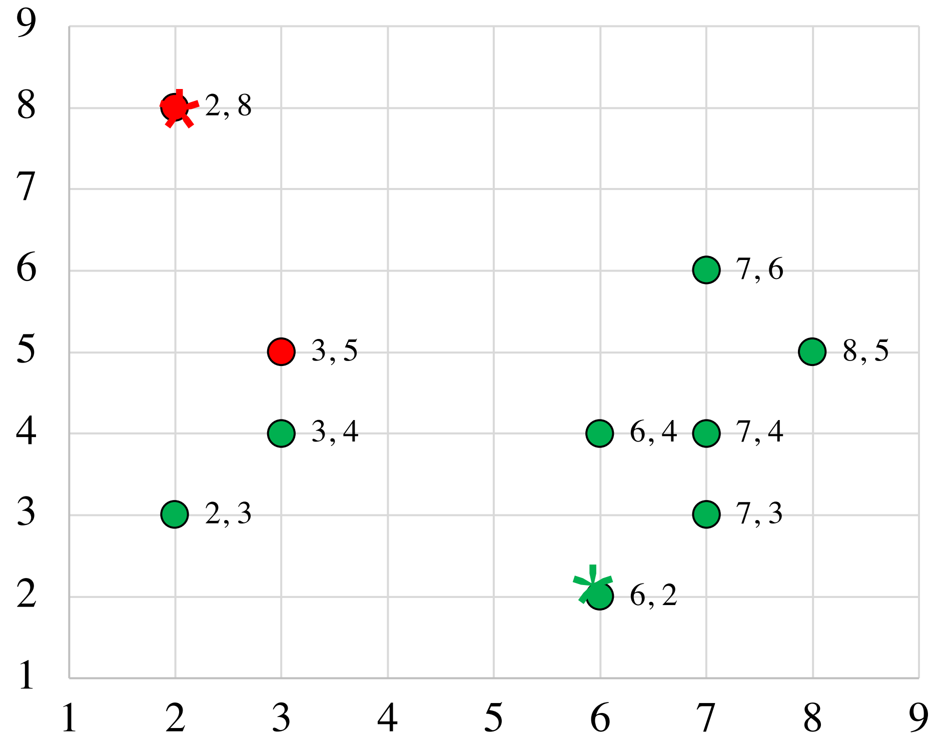
K-Medoids Clustering

- Instead of taking the ***mean*** value of the objects in a cluster as a reference point, ***medoids*** can be used, which is the most centrally located ***object*** in a cluster
- The K-Medoids clustering algorithm:
  - Select K initial representative **objects** (i.e., as initial K **medoids**)
  - Repeat
    - Assigning each **object** to the cluster with the **nearest medoid**
    - Randomly select a **non-medoid o_i**
      - » Either go through i = 1…K (recommended; why?) or randomly select an i
    - Compute the total cost ***S*** of **swapping the medoid m_i with o_i**
    - If $S < 0$, then swap m_i with o_i to form the new medoid
  - Until convergence

K-Medoids: Example

- Euclidean distance

			(2, 8)	(6, 2)
X1	3	5	**3.16**	4.24
X2	3	4	4.12	**3.61**
X3	2	8	**0.00**	7.21
X4	2	3	5.00	**4.12**
X5	6	2	7.21	**0.00**
X6	6	4	5.66	**2.00**
X7	7	3	7.07	**1.41**
X8	7	4	6.40	**2.24**
X9	8	5	6.71	**3.61**
X10	7	6	5.39	**4.12**

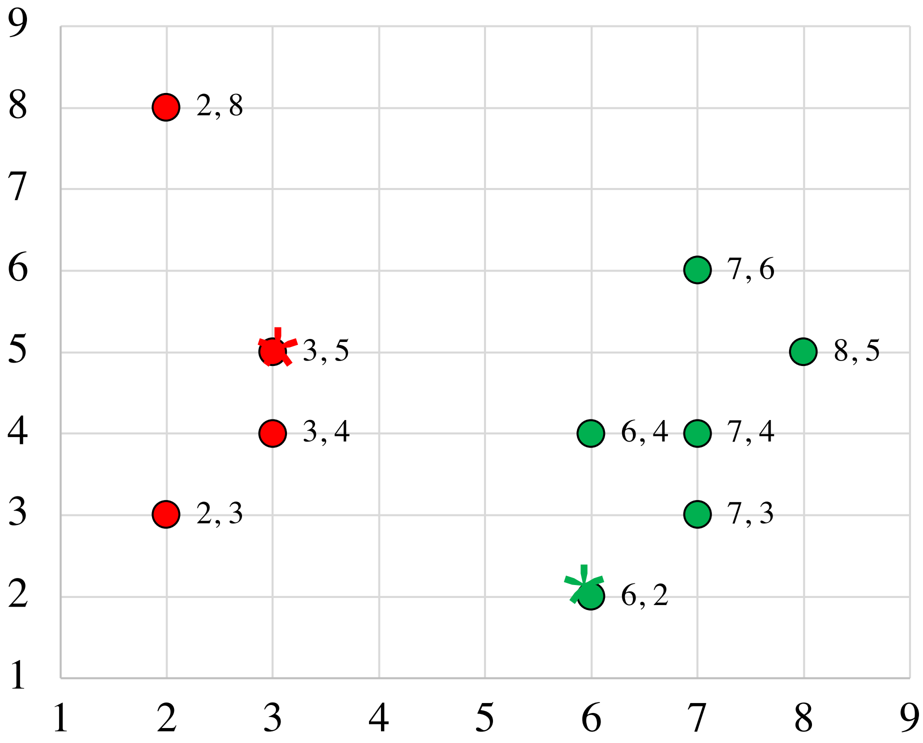


$$SSE = 3.16^2+3.61^2+4.12^2+2^2+1.41^2+2.24^2+3.61^2+4.12^2 = 81.0$$

K-Medoids: Example

- Swap the red medoid (2,8) with (3, 5)?

			(3, 5)	(6, 2)
X1	3	5	**0.00**	4.24
X2	3	4	**1.00**	3.61
X3	2	8	**3.16**	7.21
X4	2	3	**2.24**	4.12
X5	6	2	4.24	**0.00**
X6	6	4	3.16	**2.00**
X7	7	3	4.47	**1.41**
X8	7	4	4.12	**2.24**
X9	8	5	5.00	**3.61**
X10	7	6	**4.12**	**4.12**



$SSE = 1^2+3.16^2+2.24^2+2^2+1.41^2+2.24^2+3.61^2+4.12^2 = 57.0$

$S = 57.0-81.0 = -24 < 0$, so we swap them!

K-Medoids: Complexity

- PAM (Partitioning Around Medoids: Kaufmann & Rousseeuw 1987)
  - Starts from an initial set of medoids, and
  - Iteratively replaces one of the medoids by one of the non-medoids if it improves the total sum of the squared errors (SSE) of the resulting clustering
  - PAM works effectively for small data sets but does not scale well for large data sets (due to the computational complexity)
  - **Computational complexity: PAM: $O(K(n - K)^2)$ (quite expensive!)**

K-Medoids: Complexity

- Efficiency improvements on PAM
  - CLARA (Kaufmann & Rousseeuw, 1990):
    - PAM on samples; $O(Ks^2 + K(n - K))$, s is the sample size
  - CLARANS (Ng & Han, 1994): Randomized re-sampling, ensuring efficiency + quality

Outline

- Basic Concepts of K-Partitioning Methods
- The **K-Means** Clustering Method
- Initialization of K-Means Clustering
- The **K-Medoids** Clustering Method
- The **K-Medians** Clustering Method
- The **K-Modes** Clustering Method
- The **Kernel K-Means** Clustering Method

K-Medians: Handling Outliers by Computing Medians

- Medians are less sensitive to outliers than means
  - Think of the median salary vs. mean salary of a large firm when adding a few top executives!
- K-Medians: Instead of taking the mean value of the objects in a cluster as the center point, medians are used (L1-norm as the distance measure)
- The criterion function for the K-Medians algorithm:

$$S = \sum_{k=1}^{K} \sum_{x_{i \in C_k}} | x_{ij} - med_{kj} |$$

K-Medians

- The *K-Medians* clustering algorithm:
  - Select *K* **points** as initial *K medians*
  - **Repeat**
    - Assign every point to its nearest median
    - Re-compute the median using the median of each individual feature
  - **Until** convergence

Outline

- Basic Concepts of K-Partitioning Methods
- The **K-Means** Clustering Method
- Initialization of K-Means Clustering
- The **K-Medoids** Clustering Method
- The **K-Medians** Clustering Method
- The **K-Modes** Clustering Method
- The **Kernel K-Means** Clustering Method

K-Modes: Clustering Categorical Data

- *K-Means* cannot handle non-numerical (categorical) data
  - Mapping categorical value to 1/0 cannot generate quality clusters for high-dimensional data
- ***K-Modes*: An extension to *K-Means* by replacing means of clusters with ***modes***
- Dissimilarity measure between object X and the center of a cluster Z
  - $\Phi(x_j, z_j) = 1 - n_j^r/n_l$ when $x_j = z_j$; 1 when $x_j \neq z_j$
    - where z_j is the categorical value of attribute j in Z_l, n_l is the number of objects in cluster l, and n_j^r is the number of objects whose attribute value is r
- This dissimilarity measure (distance function) is **frequency-based**

K-Modes

- Algorithm is still based on iterative *object cluster assignment* and *centroid update*
- A ***fuzzy K-Modes*** method is proposed to calculate a ***fuzzy cluster membership value*** for each object to each cluster

Summary

- Basic Concepts of K-Partitioning Methods
- The **K-Means** Clustering Method
  – What are the disadvantages and solutions?
- Initialization of K-Means Clustering
- The **K-Medoids** Clustering Method
- The **K-Medians** Clustering Method
- The **K-Modes** Clustering Method
- The **Kernel K-Means** Clustering Method

References: Partitioning Methods

- J. MacQueen. Some Methods for Classification and Analysis of Multivariate Observations. In Proc. of the 5th Berkeley Symp. on Mathematical Statistics and Probability, 1967
- S. Lloyd. Least Squares Quantization in PCM. IEEE Trans. on Information Theory, 28(2), 1982
- A. K. Jain and R. C. Dubes. Algorithms for Clustering Data. Prentice Hall, 1988
- R. Ng and J. Han. Efficient and Effective Clustering Method for Spatial Data Mining. VLDB'94
- B. Schölkopf, A. Smola, and K. R. Müller. Nonlinear Component Analysis as a Kernel Eigenvalue Problem. Neural computation, 10(5):1299–1319, 1998
- I. S. Dhillon, Y. Guan, and B. Kulis. Kernel K-Means: Spectral Clustering and Normalized Cuts. KDD'04
- D. Arthur and S. Vassilvitskii. K-means++: The Advantages of Careful Seeding. SODA'07
- C. K. Reddy and B. Vinzamuri. A Survey of Partitional and Hierarchical Clustering Algorithms, in (Chap. 4) Aggarwal and Reddy (eds.), Data Clustering: Algorithms and Applications. CRC Press, 2014