Data Mining & Machine Learning

Yong Zheng

Illinois Institute of Technology
Chicago, IL, 60616, USA

ILLINOIS TECH
College of Computing

Supervised vs Unsupervised Learning

- **Supervised Learning**
  - Task: Classification & Regression
  - Algorithms: KNN, Naïve Bayes, Tree, SVM, Ensemble
  - Applications: IR, Text Classification
- **Unsupervised Learning**
  - Task: Clustering, Associated Rules, Outlier Detections
  - Algorithms: K-Means, K-Mediods, DBSCAN, Hierarchical clustering, Fuzzy clustering, Association Rules, etc

Unsupervised Learning: Clustering Techniques

Supervised v.s. Unsupervised Learning

- Supervised Learning: infer a (predictive) function from data associated with pre-defined targets/classes/labels
  Example: group objects by predefined labels
  Goal: Learn a model from labelled data (with multiple features) for future predictions
  Outcomes: We know outcomes: the predefined labels
  Evaluation: error/accuracy, and other more metrics
  Data Mining Task: Classification

- Unsupervised Learning: discover or describe underlying structure from unlabelled data
  Example: group objects by multiple features
  Goal: Learn the structure from unlabelled data (with multiple features)
  Outcomes: We do not know the outcomes
  Evaluation: No clear performance or evaluation methods
  Data Mining Task: Clustering

Unsupervised Learning

- Unsupervised Learning: discover or describe underlying structure/correlations from unlabelled data
  Example: group objects without predefined labels
  Goal: Learn the structure from unlabelled data
  Evaluation: No clear performance, but there are some metrics

> **Unsupervised learning** is the machine learning task of inferring a function to describe hidden structure from unlabeled data. Since the examples given to the learner are unlabeled, there is no error or reward signal to evaluate a potential solution. This distinguishes unsupervised learning from supervised learning and reinforcement learning.
>
> From Wikipedia.org

Unsupervised Learning

Approaches related to unsupervised learning

- Clustering
- Association Rule Mining
- Principal Component Analysis
- etc

Unsupervised Learning

How to evaluate unsupervised learning

- Usually, we do not have a metric for evaluations
- But there are two ways
  - We can manually look at the outputs, analyze and interpret it, to see whether there are significant differences and they are useful
  - The outputs of unsupervised learning can be used as inputs to a supervised learning process, to see whether the supervised learning can be improved

Clustering

- Intro: Clustering
- Partitional Clustering
- Density-Based Clustering
- Hierarchical Clustering

Clustering

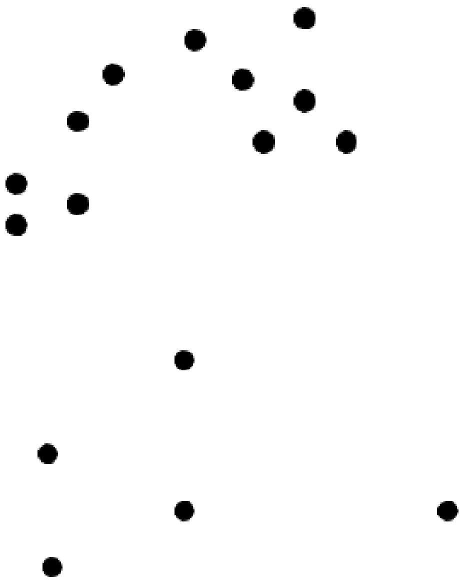
- Intro: Clustering
- Partitional Clustering
- Density-Based Clustering
- Hierarchical Clustering

Clustering Tasks

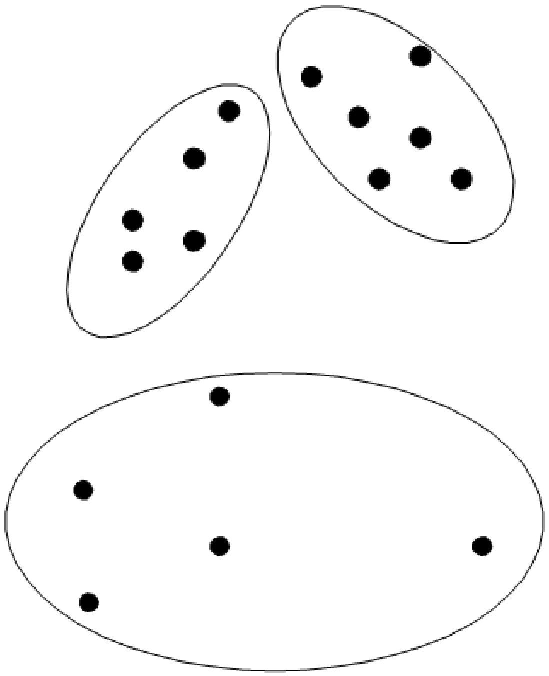
- Partitional Clustering: just group objects to minimize intra-cluster distances and maximize inter-cluster distances
  Example: Document Clustering
- Density-Based Clustering: cluster objects based on the local connectivity and density functions
- Hierarchical Clustering: a clustering process in order to discover the hierarchical structure, like a hierarchical tree
  Example: categories and subcategories; taxonomies

Partitional Clustering

- Partitional Clustering: just group objects to minimize intra-cluster distances and maximize inter-cluster distances



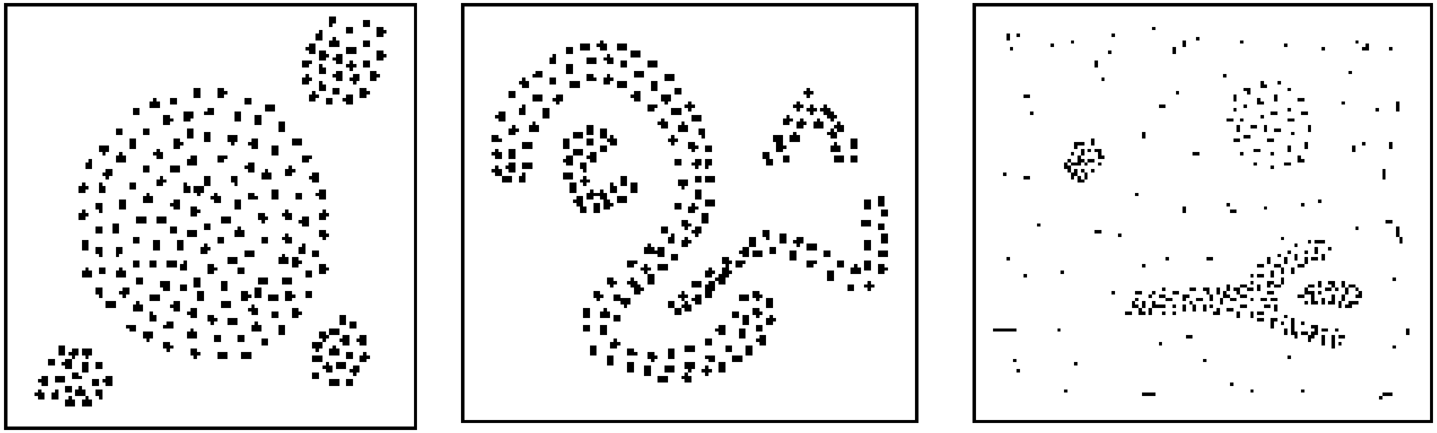
Original Points



A Partitional Clustering

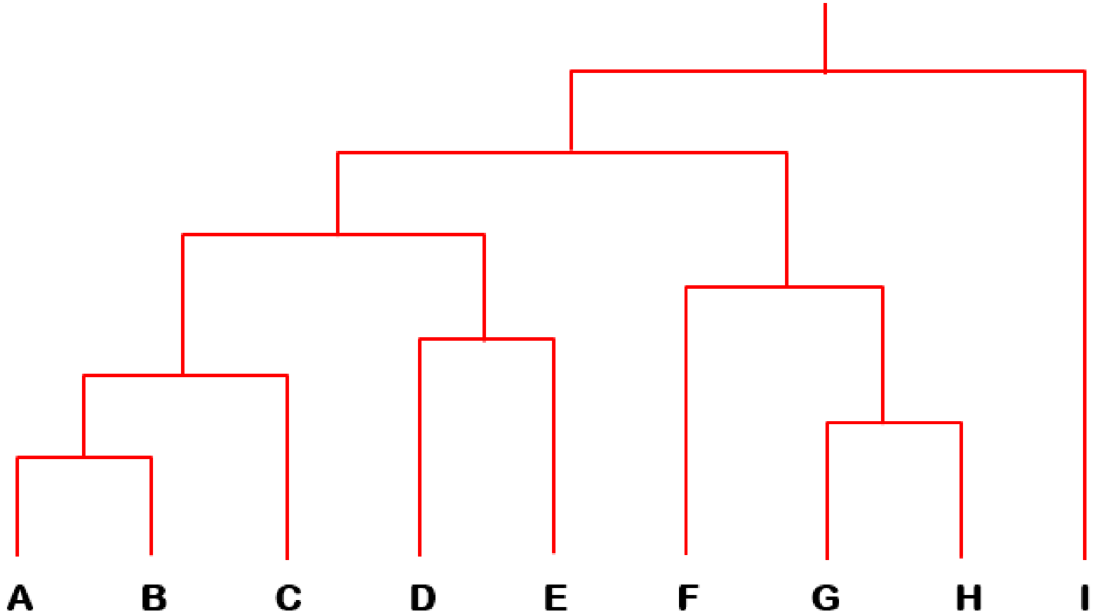
Density-Based Clustering

- Density-Based Clustering: cluster objects based on the local connectivity and density functions. Each cluster has a considerable higher density of points than outside of the cluster



Hierarchical Clustering

- Hierarchical Clustering: a clustering process in order to discover the hierarchical structure, like a hierarchical tree

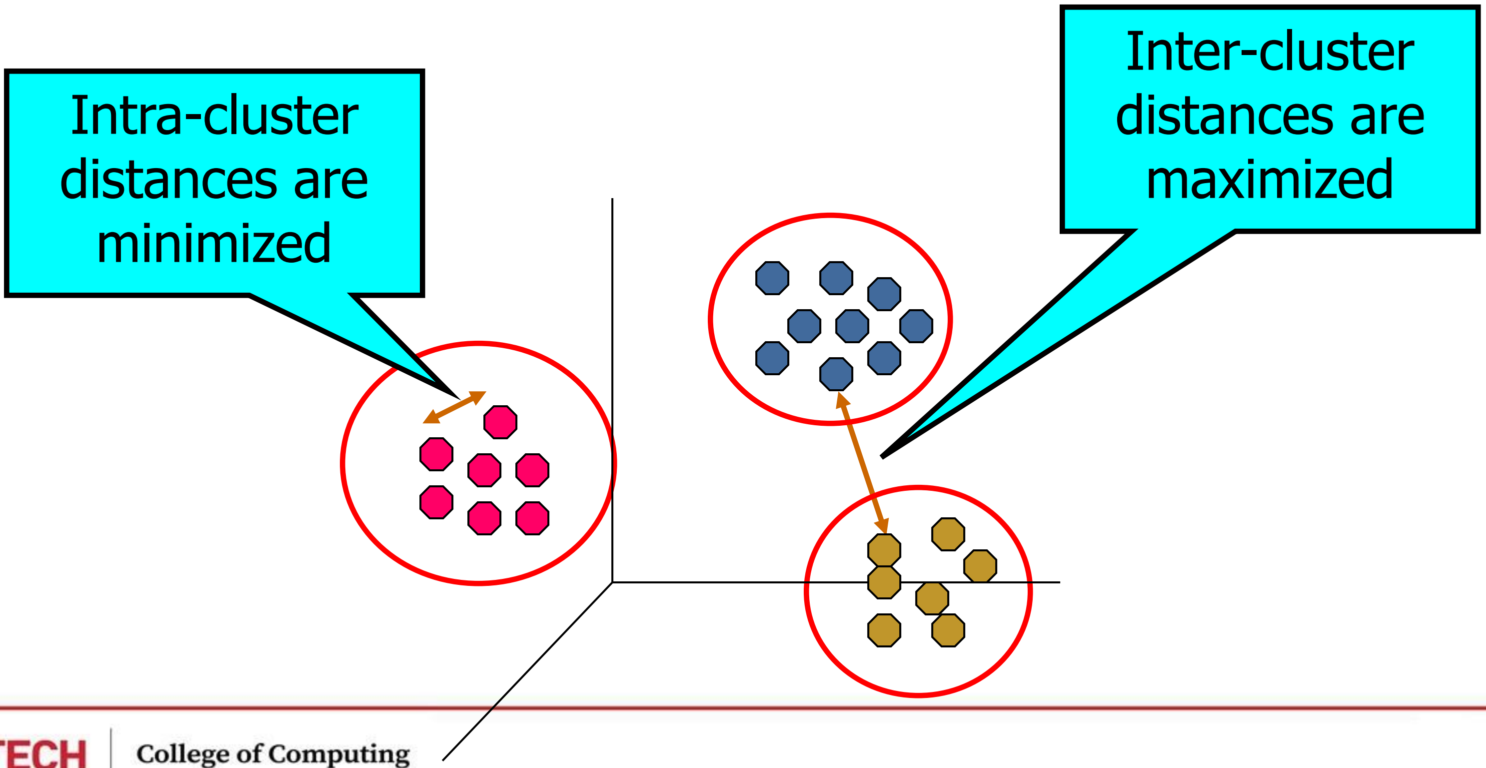


Clustering

- Intro: Clustering
- Partitional Clustering
- Density-Based Clustering
- Hierarchical Clustering

Partitional Clustering

- Partitional Clustering: a unsupervised way to group objects
- Goal: Finding groups of objects in data such that the objects in a group will be similar (or related) to one another and different from (or unrelated to) the objects in other groups

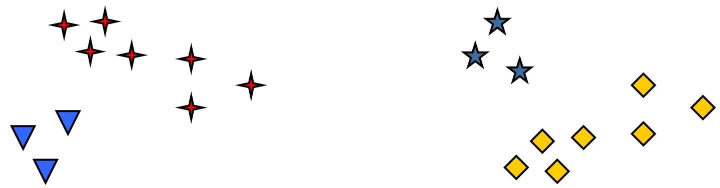


Notion of a Cluster can be Ambiguous

How many clusters?

Six Clusters

Two Clusters

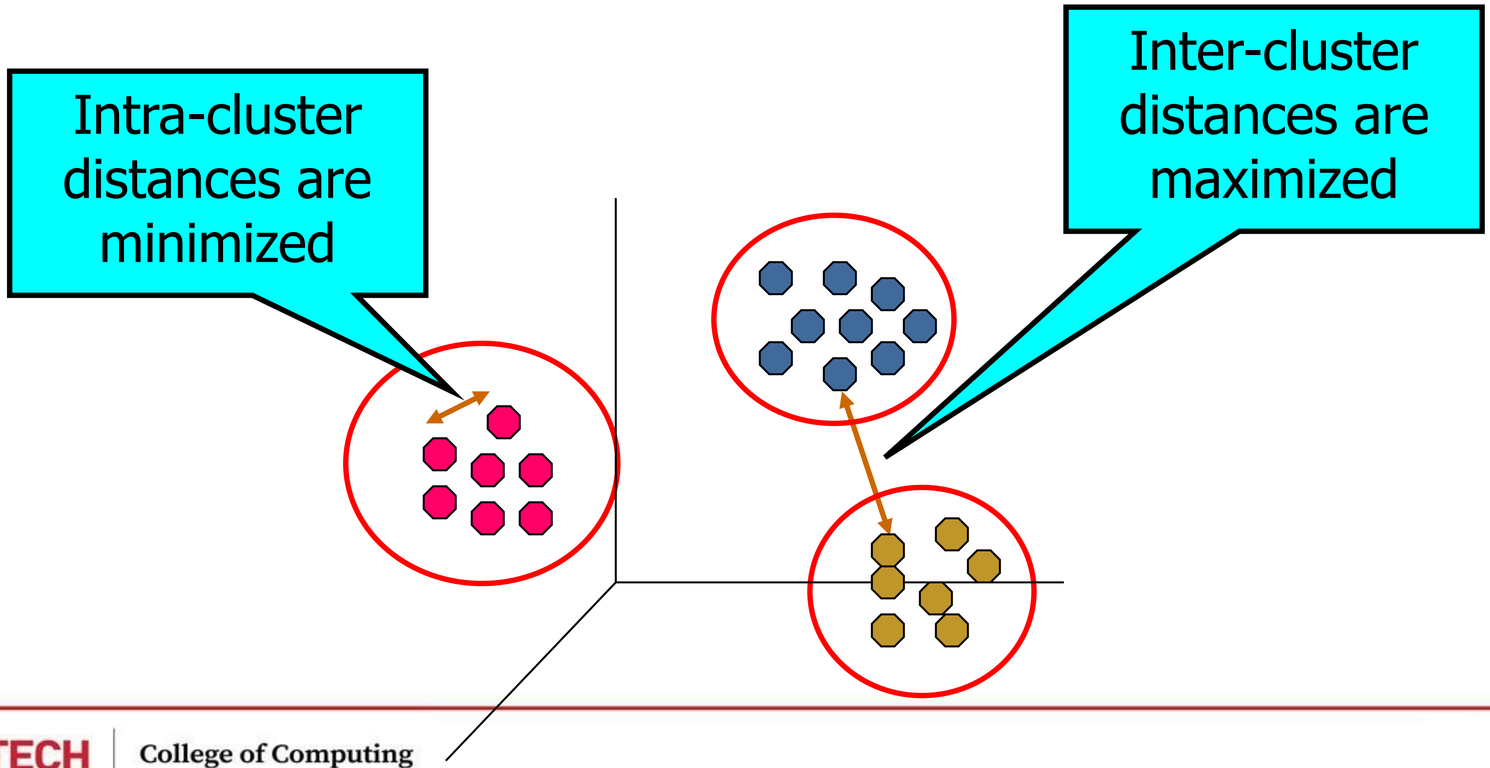


Four Clusters

How many clusters there should be?

Partitional Clustering

- Basic idea
  - Measure similarity or distance between each two objects
  - Group the objects based on these similartiies



Distance or Similarity Measures

- Common Distance Measures:
  - Manhattan distance:

$$X = \langle x_1, x_2, \cdots, x_n \rangle \qquad Y = \langle y_1, y_2, \cdots, y_n \rangle$$

$$dist(X,Y) = |x_1 - y_1| + |x_2 - y_2| + \cdots + |x_n - y_n|$$

  - Euclidean distance:

$$dist(X,Y) = \sqrt{(x_1 - y_1)^2 + \cdots + (x_n - y_n)^2}$$

  - Cosine distance:

$$dist(X,Y) = 1 - sim(X,Y)$$

$$sim(X,Y) = \frac{\sum_i (x_i \times y_i)}{\sqrt{\sum_i x_i^2 \times \sum_i y_i^2}}$$

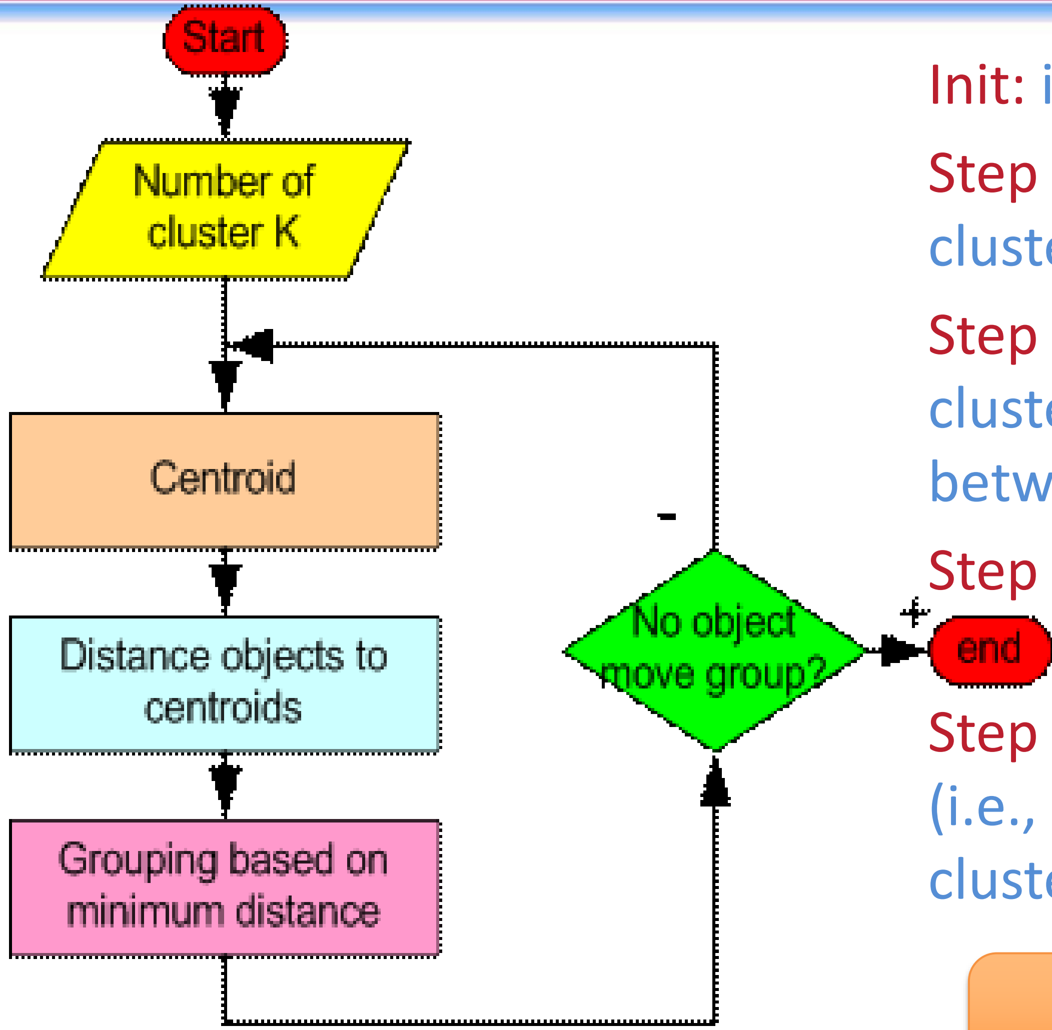
K-Means Clustering Algorithm

- Assume we have many examples/instances, each example can be represented by a vector of features, where the features must be numerical ones, e.g., weight, size, price, profits, etc

- So that, we can use the distance measures to calculate the similarity or the dissimilarity (i.e., distance) between each two examples.

- With such setting, we are able to apply a K-Means clustering algorithms to perform the normal clustering task.

K-Means Clustering Algorithm



Init: initialize K and K clusters

Step 1. Calculate centroids for K clusters

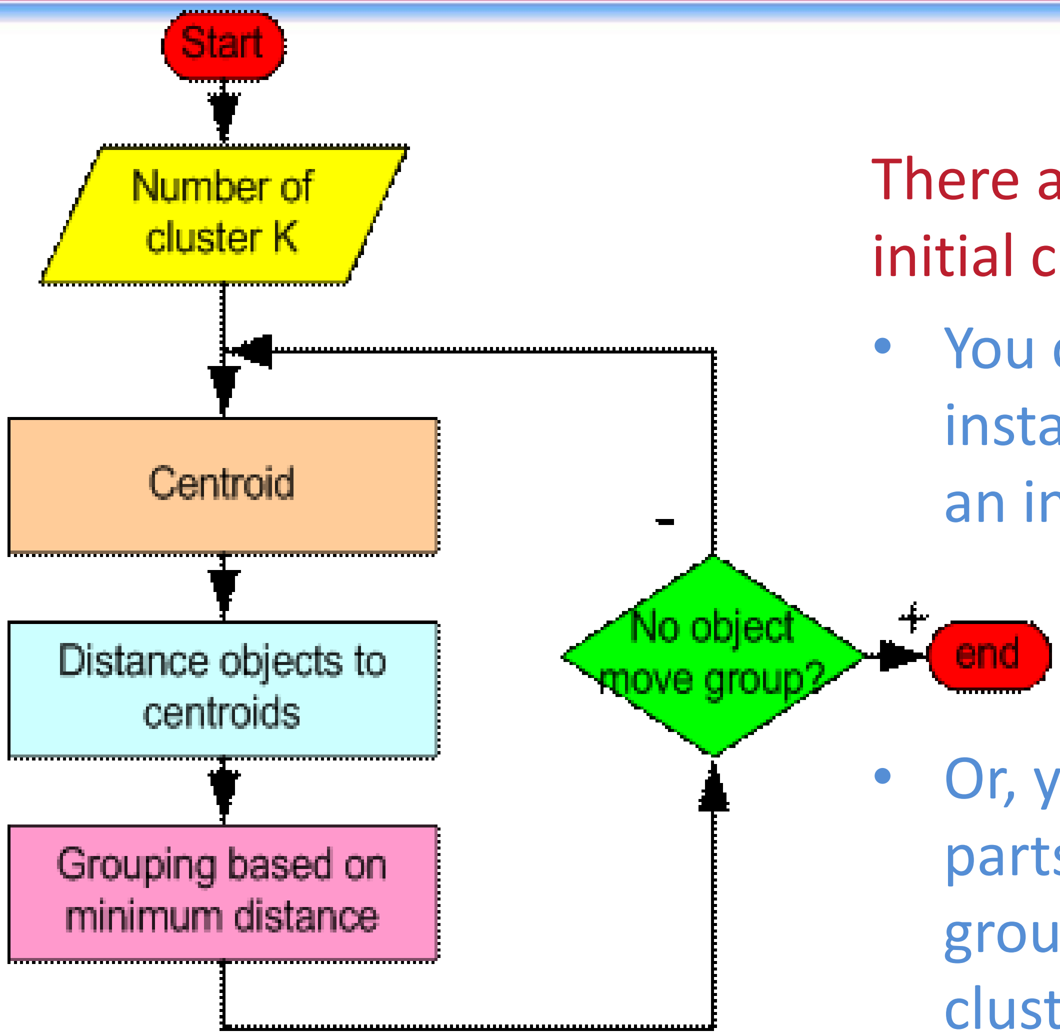
Step 2. Assign data points to each cluster based on the distance between data and centroids

Step 3. get new K clusters, compare them with previous clusters

Step 4. Repeat 1,2,3 until convergence (i.e., no points move between clusters)

Normalize the features!!

K-Means Clustering Algorithm



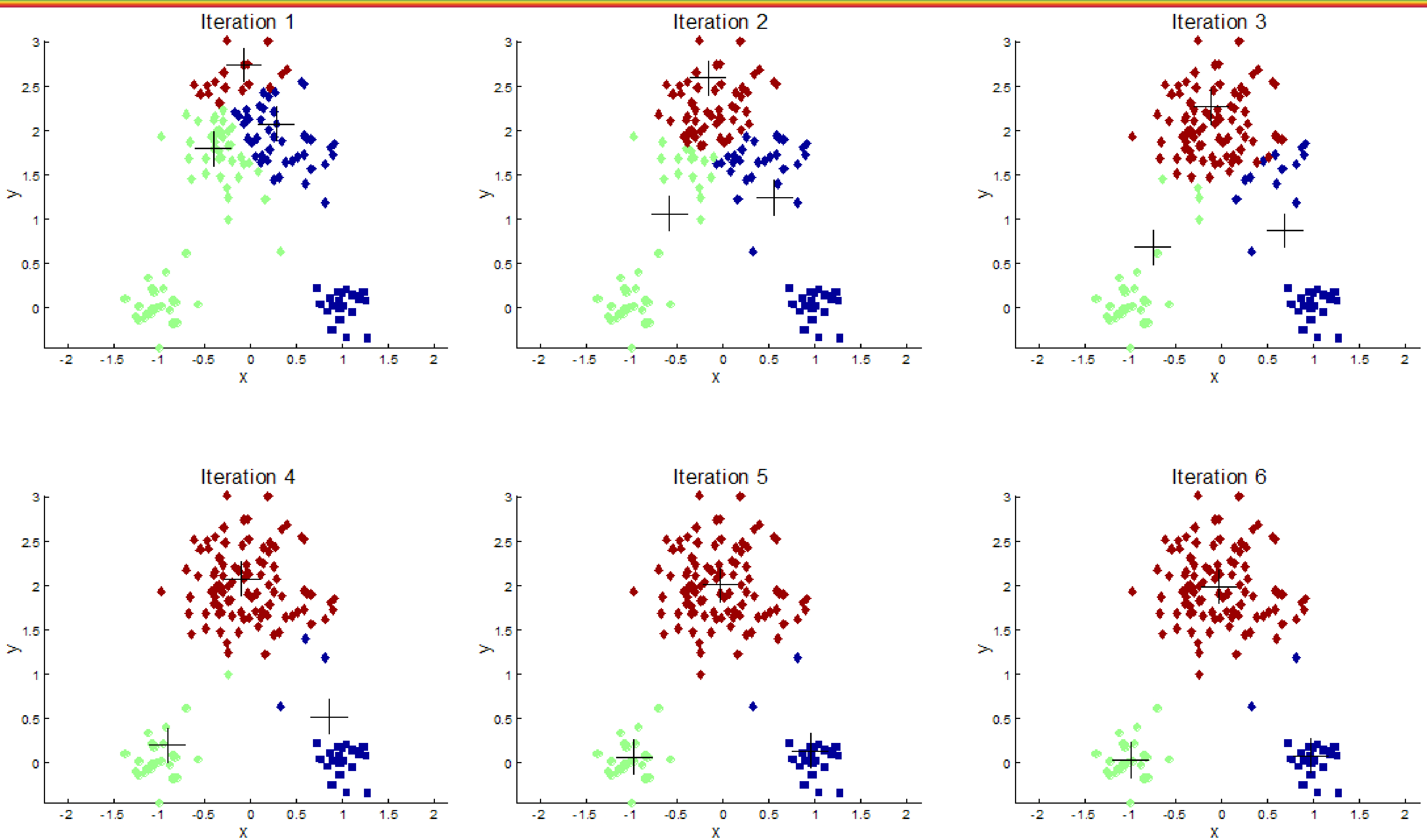
There are multiple ways to define the initial cluster:

- You can randomly choose K instances and each one of them is an individual cluster;
- Or, you can randomly assign all or parts of your instances into K groups. Each group is an individual cluster

K-Means Clustering Algorithm

- Stopping Criterion in Iterative learning
  - We need to stop the learning iterations when it is converged
  - How to determine it is converged?
    - Criterion 1: new clusters = old clusters
      stop learning when no changes on clusters
    - Criterion 2: setup a maximal learning iterations
      stop learning when it got to maximal learning iterations
    - In practice, we usually use 2nd criterion, since clustering may converge after several/unexpected iterations, especially when the data set is large

K-Means Clustering Algorithm



Example: K-Means

Example: Clustering Documents

Initial (arbitrary) assignment:
C1 = {D1,D2},
C2 = {D3,D4},
C3 = {D5,D6}

	T1	T2	T3	T4	T5
D1	0	3	3	0	2
D2	4	1	0	1	2
D3	0	4	0	0	2
D4	0	3	0	3	3
D5	0	1	3	0	1
D6	2	2	0	0	4
D7	1	0	3	2	0
D8	3	1	0	0	2
C1	4/2	4/2	3/2	1/2	4/2
C2	0/2	7/2	0/2	3/2	5/2
C3	2/2	3/2	3/2	0/2	5/2

Cluster Centroids

Example: K-Means

Now compute the similarity (or distance) of each item with each cluster, resulting a cluster-document similarity matrix (here we use dot product as the similarity measure for simplicity).

	D1	D2	D3	D4	D5	D6	D7	D8
C1	29/2	29/2	24/2	27/2	17/2	32/2	15/2	24/2
C2	31/2	20/2	38/2	45/2	12/2	34/2	6/2	17/2
C3	28/2	21/2	22/2	24/2	17/2	30/2	11/2	19/2

For each document, reallocate the document to the cluster to which it has the highest similarity (shown in red in the above table). After the reallocation we have the following new clusters. Note that the previously unassigned D7 and D8 have been assigned, and that D1 and D6 have been reallocated from their original assignment.

C1 = {D2,D7,D8}, C2 = {D1,D3,D4,D6}, C3 = {D5}

This is the end of first iteration (i.e., the first reallocation). Next, we repeat the process for another reallocation…

Example: K-Means

C1 = {D2,D7,D8}, C2 = {D1,D3,D4,D6}, C3 = {D5}

Now compute new cluster centroids using the original document-term matrix

	T1	T2	T3	T4	T5
D1	0	3	3	0	2
D2	4	1	0	1	2
D3	0	4	0	0	2
D4	0	3	0	3	3
D5	0	1	3	0	1
D6	2	2	0	0	4
D7	1	0	3	2	0
D8	3	1	0	0	2
C1	8/3	2/3	3/3	3/3	4/3
C2	2/4	12/4	3/4	3/4	11/4
C3	0/1	1/1	3/1	0/1	1/1

This will lead to a new cluster-doc similarity matrix similar to previous slide. Again, the items are reallocated to clusters with highest similarity.

	D1	D2	D3	D4	D5	D6	D7	D8
C1	7.67	15.01	5.34	9.00	5.00	12.00	7.67	11.34
C2	16.75	11.25	17.50	19.50	8.00	6.68	4.25	10.00
C3	14.00	3.00	6.00	6.00	11.00	9.34	9.00	3.00

New assignment → **C1 = {D2,D6,D8}, C2 = {D1,D3,D4}, C3 = {D5,D7}**

Note: This process is now repeated with new clusters. However, the next iteration in this example Will show no change to the clusters, thus terminating the algorithm.

In-Class Practice

Subject	A	B
1	1.0	1.0
2	1.5	2.0
3	3.0	4.0
4	5.0	7.0
5	3.5	5.0
6	4.5	5.0
7	3.5	4.5

Initialization: K = 2, initial cluster/groups are defined as:

	Individual	Mean Vector (centroid)
Cluster 1	1	(1.0, 1.0)
Cluster 2	4	(5.0, 7.0)

Manhattan distance:

$$\boxed{dist(X,Y) = |x_1 - y_1| + |x_2 - y_2| + \cdots + |x_n - y_n|}$$

K-Means Clustering: Evaluations

- There are no clear evaluations: clustering is good as long as it can serve for your usage or applications
- Most common measure is Sum of Squared Error (SSE)
  - For each point, the error is the distance to the nearest cluster
  - To get SSE, we square these errors and sum them.

$$SSE = \sum_{i=1}^{K} \sum_{x \in C_i} dist^2(m_i, x)$$

It is not a metric to evaluate clustering results

  - x is a data point in cluster C_i and m_i is the representative point for cluster C_i
    - can show that m_i corresponds to the center (mean) of the cluster
  - Drawback: if K is increased, SSE can be decreased
  - It is used to measure how well the clustering process is
    It cannot tell how well the clustering results are

K-Means Clustering: Evaluations

SSE can also be used to find the best K value

Try K = 3, 5, 7, 10, 13, 20, etc…

Observe the K value which can lower SSE

Unsupervised Learning

How to evaluate unsupervised learning

- Usually, we do not have a metric for evaluations
- But there are two ways
  - We can manually look at the outputs, compare centroids, analyze and interpret it, to see whether there are significant differences and they are useful
  - The outputs of unsupervised learning can be used as inputs to a supervised learning process, to see whether the supervised learning can be improved

K-Means Clustering: Evaluations

- How to evaluate the clustering results?
  – Solution 1: compare clusters by using centroid and tell the significant differences among different clusters, to better understand why they were put together

Centroid	Gender	GPA	Study Hours	Course Completed
C1	1	2.5	20	10
C2	0.6	4.0	40	3
C3	0	3.0	25	11

K-Means Clustering: Evaluations

- How to evaluate the clustering results?
  - Solution 2: add the clustering results into a supervised learning process to learn whether they are able to improve supervised learning

Student	Gender	GPA	Study Hours	Course Completed	TA?
S1	1	2.5	20	10	N
S2	0	4.0	40	3	Y
S3	0	3.0	25	11	Y

Student	Gender	GPA	Study Hours	Course Completed	TA?	Cluster
S1	1	2.5	20	10	N	c1
S2	0	4.0	40	3	Y	c2
S3	0	3.0	25	11	Y	c2

K-Means Clustering Algorithm

- **Strength of the *k-means*:**
  - *Relatively efficient*: $O(tkn)$, where n is # of objects, k is # of clusters, and t is # of iterations. Normally, $k, t << n$
  - Often terminates at a *local optimum*
- **Weakness of the *k-means*:**
  - What about categorical data?
  - Performance is sensitive to initializations, e.g., <u>K, initial clusters, and the definition of centriods</u>
  - Need to specify *k,* the *number* of clusters, in advance
  - Unable to handle noisy data and *outliers*
- **Variations of K-Means usually differ in:**
  - Selection of the initial *k* means
  - Dissimilarity calculations
  - Strategies to calculate cluster means

Improve Your Clustering

- Pre-processing
  - Normalize the data
  - Eliminate outliers
- Post-processing
  - Eliminate small clusters that may represent outliers
  - Split 'loose' clusters, i.e., clusters with relatively high SSE
  - Merge clusters that are 'close' and that have relatively low SSE

Variations of K-Means Clustering

- K-Means Clustering: centroid is defined as means
- K-Median Clustering: centroid is defined as medians
- K-Medoids Clustering: medoids as centroid
- X-Means Clustering: figure out a way to find best K
- Fuzzy C-Means Clustering: fuzzy degree as confidence
- Many more…

K-Medoids Clustering

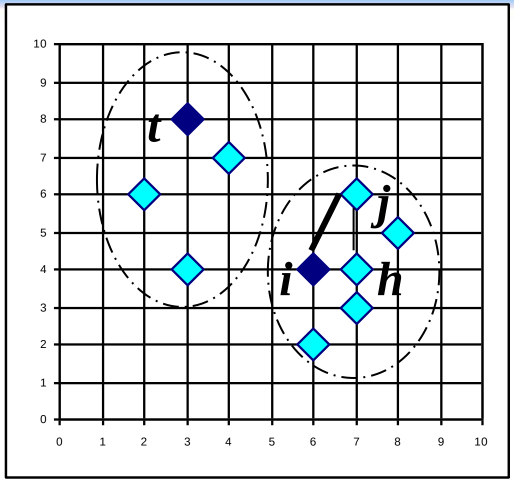
K-Medoids Clustering

- It is built as one of partitonal clustering approaches
- Medoids as centroids
- A medoid is defined as the object of a cluster whose average dissimilarity to all the objects in the cluster is minimal
- In other words, a medoid is the most centrally located points in the cluster

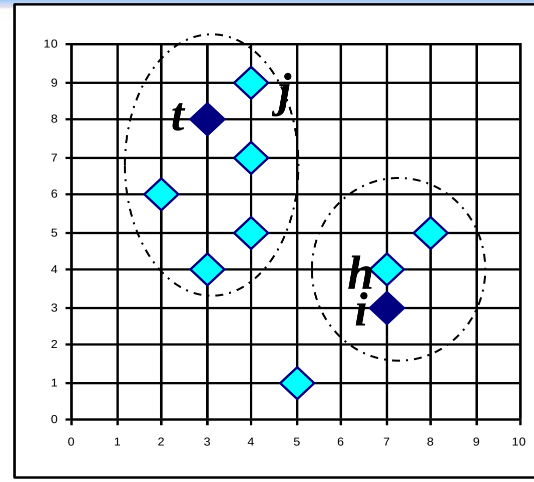
K-Medoids Clustering

- PAM (Kaufman and Rousseeuw, 1987), built in Splus
- Use real object to represent the cluster
  - Select ***k*** representative objects arbitrarily
  - For each pair of non-selected object ***h*** and selected object ***i***, calculate the total swapping cost TC_{ih}
  - For each pair of ***i*** and ***h***,
    - If $TC_{ih} < 0$, ***i*** is replaced by ***h***
    - Then assign each non-selected object to the most similar representative object
  - repeat steps 2-3 until there is no change

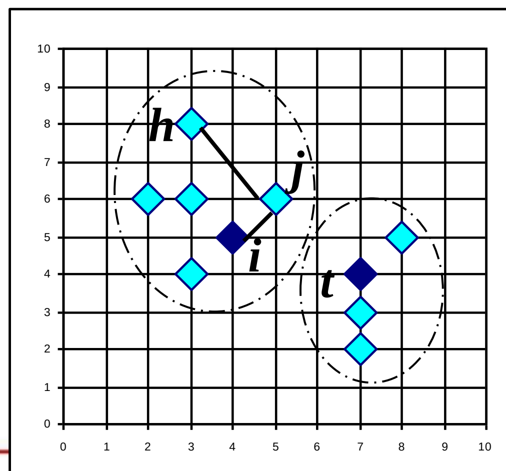
K-Medoids Clustering



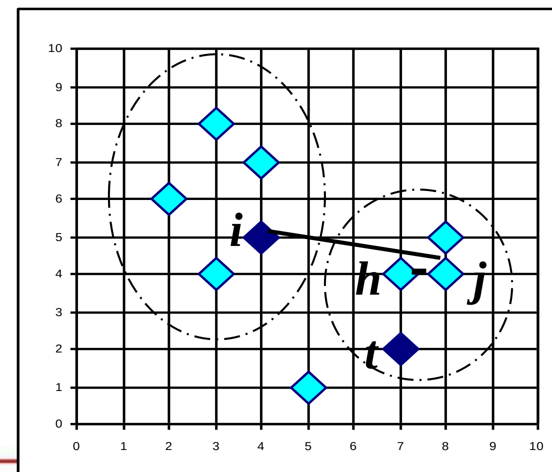
$C_{jih} = d(j, h) - d(j, i)$



$C_{jih} = 0$



$C_{jih} = d(j, t) - d(j, i)$



$C_{jih} = d(j, h) - d(j, t)$

K-Medoids Clustering

K-Medoids Clustering: Pros and Cons

- The centroid is defined as the medoid which is the most centrally located object in one cluster
- To some extent, it helps alleviate the situation of outliers
- But this approach is not scalable – time-consuming for large scale of the data set
- Still sensitive to K, initialization, etc

Next Class

- Intro: Clustering
- Partitional Clustering
- Density-Based Clustering
- Hierarchical Clustering