Data Mining

Clustering Analysis

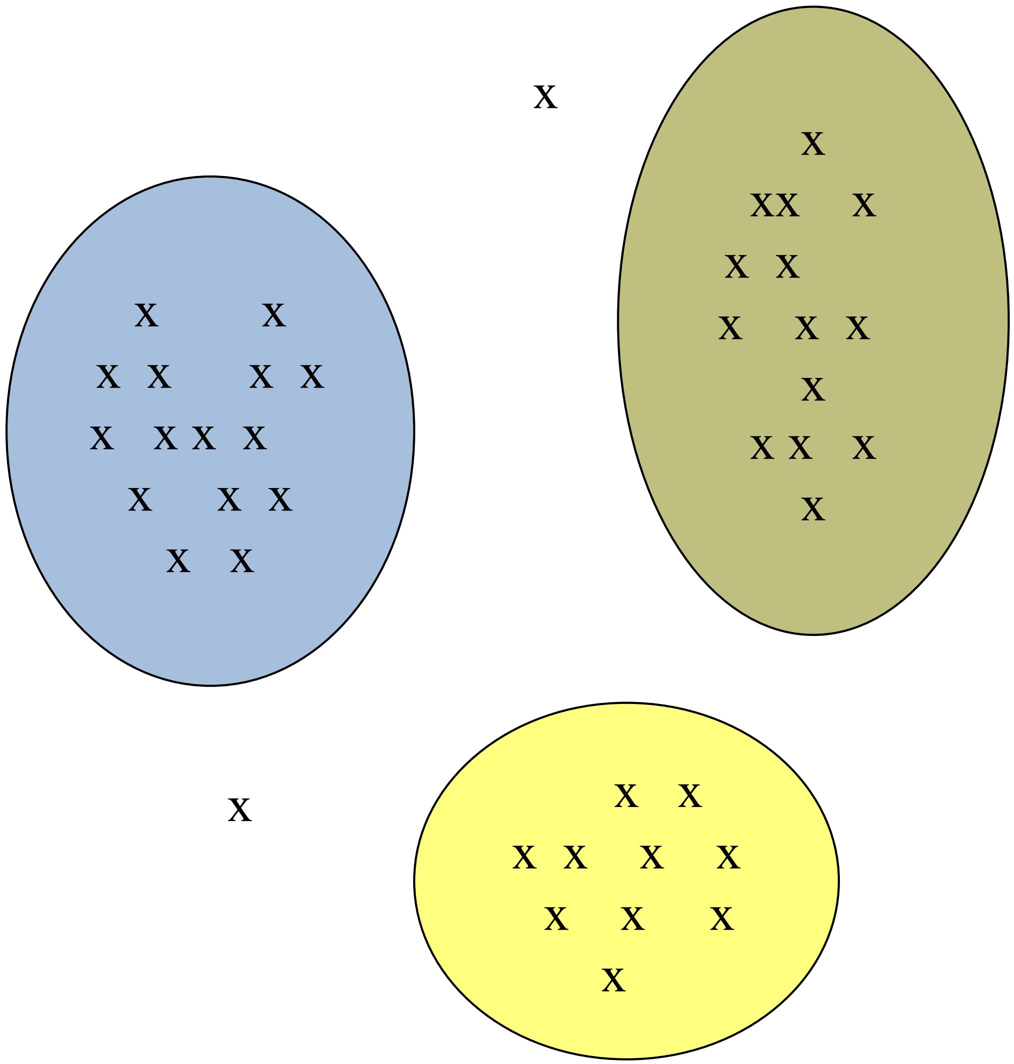
Outline

- What is clustering analysis?
- Types of data in clustering analysis
- A categorization of major clustering methods
  - Partitioning methods
  - Hierarchical methods
  - Model-based clustering methods
- Outlier analysis
- Summary

What Is Clustering Analysis?

- Clustering: a collection of data objects.
  - Similar to one another within the same cluster.
  - Dissimilar to the objects in other clusters.
- Clustering analysis.
  - Grouping a set of data objects into clusters, such that objects within each cluster are similar to each other, objects in different clusters are dissimilar to each other.
- Clustering is unsupervised classification:
  - Objects are not labeled with predefined classes.
  - Different from supervised classification where each training data is labeled with class information

An Example



Problems With Clustering

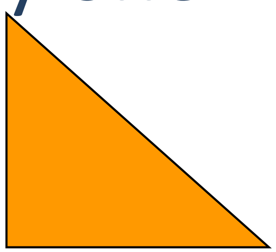
- Clustering in two dimensions looks easy.
- Clustering small amounts of data looks easy.
- And in most cases, looks are *not* deceiving.

The Curse of Dimensionality

- Many applications involve not 2, but 10 or 10,000 dimensions.
- High-dimensional spaces look different: almost all pairs of points are at about the same distance.

Example: Curse of Dimensionality

- Assume random points within a bounding box, e.g., values between 0 and 1 in each dimension.
- In 2 dimensions: a variety of distances between 0 and 1.41.
- In 10,000 dimensions, the difference in any one dimension is distributed as a triangle.



- Actual distance between two random points is the sqrt of the sum of squares of essentially the same set of differences.

General Applications

- Typical applications.
  - As a stand-alone tool to get insight into data distribution.
  - As a preprocessing step for other algorithms.
- (Spatial) data analysis
- Image processing
- Economic science (especially market research)
- WWW
  - Automatic document categorization
  - Web usage mining: cluster web log data to discover groups of similar access patterns
- Business : customer groups
- Biology: animal and plant taxonomy, Categorize genes by functionality

High-Dimension Application: SkyCat

- A catalog of 2 billion "sky objects" represents objects by their radiation in 7 dimensions (frequency bands).
- Problem: cluster into similar objects, e.g., galaxies, nearby stars, quasars, etc.
- Sloan Sky Survey is a newer, better version.

Clustering CD's (Collaborative Filtering)

- Intuitively: music divides into categories, and customers prefer a few categories.
  - But what are categories really?
- Represent a CD by the customers who bought it.
- Similar CD's have similar sets of customers, and vice-versa.

The Space of CD’s

- Think of a space with one dimension for each customer.
  - Values in a dimension may be 0 or 1 only.
- A CD’s point in this space is $(x_1, x_2, \ldots, x_k)$, where $x_i = 1$ *iff* the i^{th} customer bought the CD.
  - Compare with boolean matrix: rows = customers; cols. = CD’s.
- For Amazon, the dimension count is tens of millions.

Clustering Documents

- Represent a document by a vector $(x_1, x_2, \ldots, x_k)$, where $x_i = 1$ *iff* the i^{th} word (in some order) appears in the document.
  - It actually doesn't matter if k is infinite; i.e., we don't limit the set of words.
- Documents with similar sets of words may be about the same topic.

Example: DNA Sequences

- Objects are sequences of {C,A,T,G}.
- Distance between sequences is *edit distance*, the minimum number of inserts and deletes needed to turn one into the other.

What Is Good Clustering?

- A good clustering method will produce high quality clusters with.
  - High intra-class similarity.
  - Low inter-class similarity.
- The quality of a clustering result depends on both the similarity measure used by the method and its clustering approach used.
- The quality of a clustering method is also measured by its ability to discover some or all of the hidden patterns

Requirements of Clustering in Data Mining

- Scalability
- Ability to deal with different types of attributes
- Discovery of clusters with arbitrary shape
- Minimal requirements for domain knowledge to determine input parameters
- Able to deal with noise and outliers
- Insensitive to order of input records
- High dimensionality
- Interpretability and usability

Data Structures

- Data matrix

$$\begin{bmatrix} x_{11} & \cdots & x_{1f} & \cdots & x_{1p} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{i1} & \cdots & x_{if} & \cdots & x_{ip} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{n1} & \cdots & x_{nf} & \cdots & x_{np} \end{bmatrix}$$

- Dissimilarity matrix

$$\begin{bmatrix} 0 & & & & \\ d(2,1) & 0 & & & \\ d(3,1) & d(3,2) & 0 & & \\ : & : & : & & \\ d(n,1) & d(n,2) & \cdots & \cdots & 0 \end{bmatrix}$$

Measure the Quality of Clustering

- Dissimilarity/similarity metric: dissimilarity is expressed in terms of a distance function, which is typically metric: $d(i, j)$.
- The definitions of distance functions are usually very different for interval-scaled, boolean, categorical, ordinal variables, and temporal data.
- Weights should be associated with different variables based on applications and data semantics.
- There is a separate "quality" function that measures the "goodness" of a cluster.

Type of Data in Clustering Analysis

- Interval-scaled variables
- Binary variables
- Nominal, and ordinal variables
- Variables of mixed types
- Text
- Temporal

Major Clustering Approaches

- Partitioning algorithms: Construct various partitions and then evaluate them by some criterion
- Hierarchy algorithms: Create a hierarchical decomposition of the set of data (or objects) using some criterion
- Model-based: A model is hypothesized for each of the clusters and the idea is to find the best fit of that model to each other

Partitioning Algorithms: Basic Concept

- Partitioning method: Construct a partition of a database ***D*** of ***n*** objects into a set of ***k*** clusters
- Given a *k*, find a partition of *k clusters* that optimizes the chosen partitioning criterion
  - Global optimal: exhaustively enumerate all partitions
  - Heuristic methods: *k-means* and *k-medoids* algorithms
  - *k-means* (MacQueen' 67): Each cluster is represented by the center of the cluster
  - *k-medoids* or PAM (Partition around medoids) (Kaufman & Rousseeuw' 87): Each cluster is represented by one of the objects in the cluster

The *K-Means* Clustering Method

- **Objective**: to form a set of clusters that are as compact and separated as possible
- **Distance Measure**: Euclidean distance between data object and cluster center
- **Clustering criterion function**:

  mean squared error (MSE)

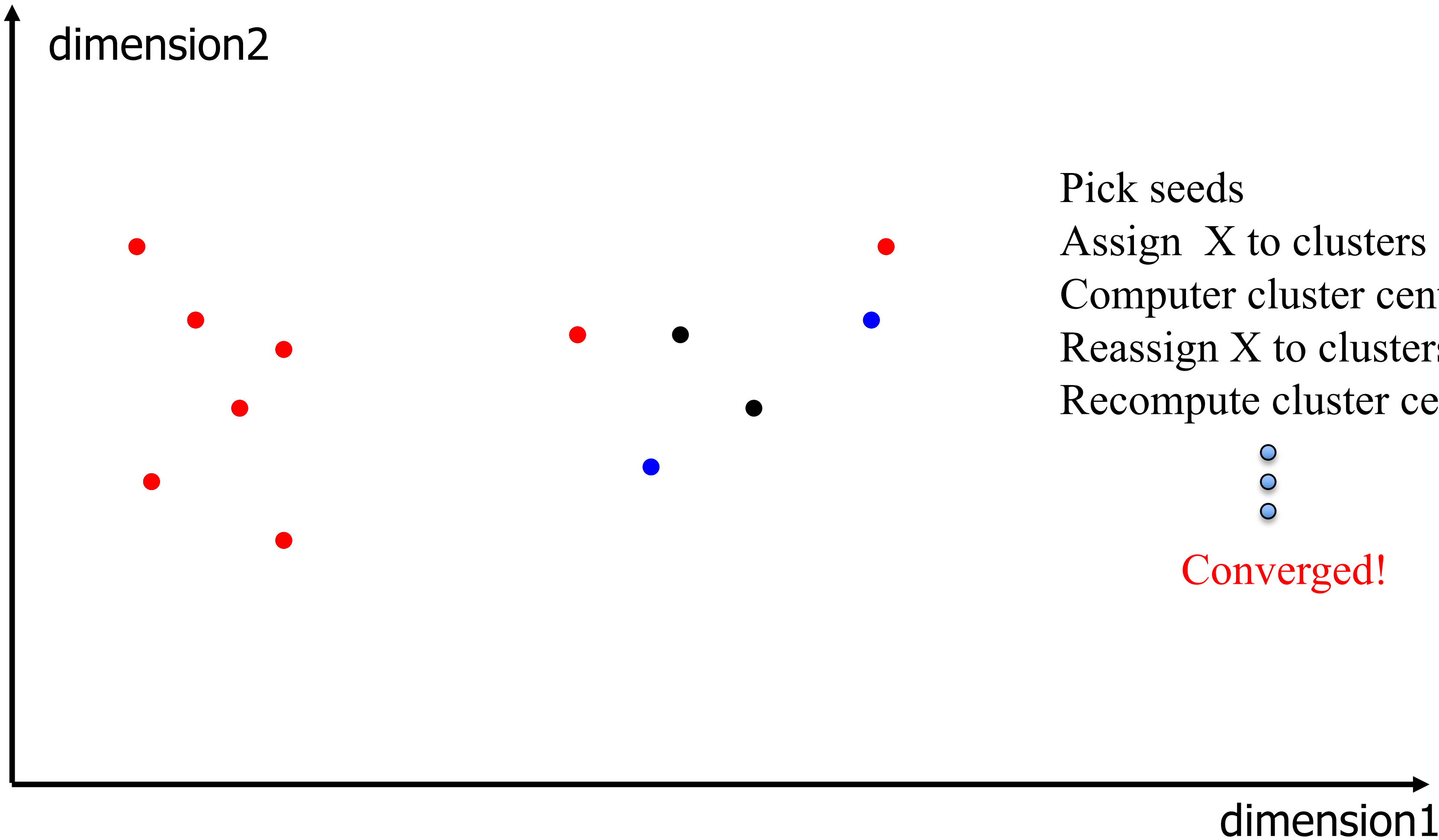
$$MSE = \sum_{i=1}^{k} \sum_{p \subseteq C_i} | x - m_i |^2$$

x: a data object
C_i: cluster i
m_i: center of cluster i
k: number of clusters

The *K-Means* Clustering Method

- **Approach**: Given *k*, the *k-means* algorithm is implemented as the following:
  - arbitrarily choose *K* objects as the initial cluster centers.
  - Repeat:
    - **Compute seed points as the centroids of the clusters of the current partition. The centroid is the center (mean point) of the cluster.**
    - **Assign each object to the cluster with the nearest seed point.**
  - stop when no more new assignment, or when clustering criterion function (mean squared error) converges.

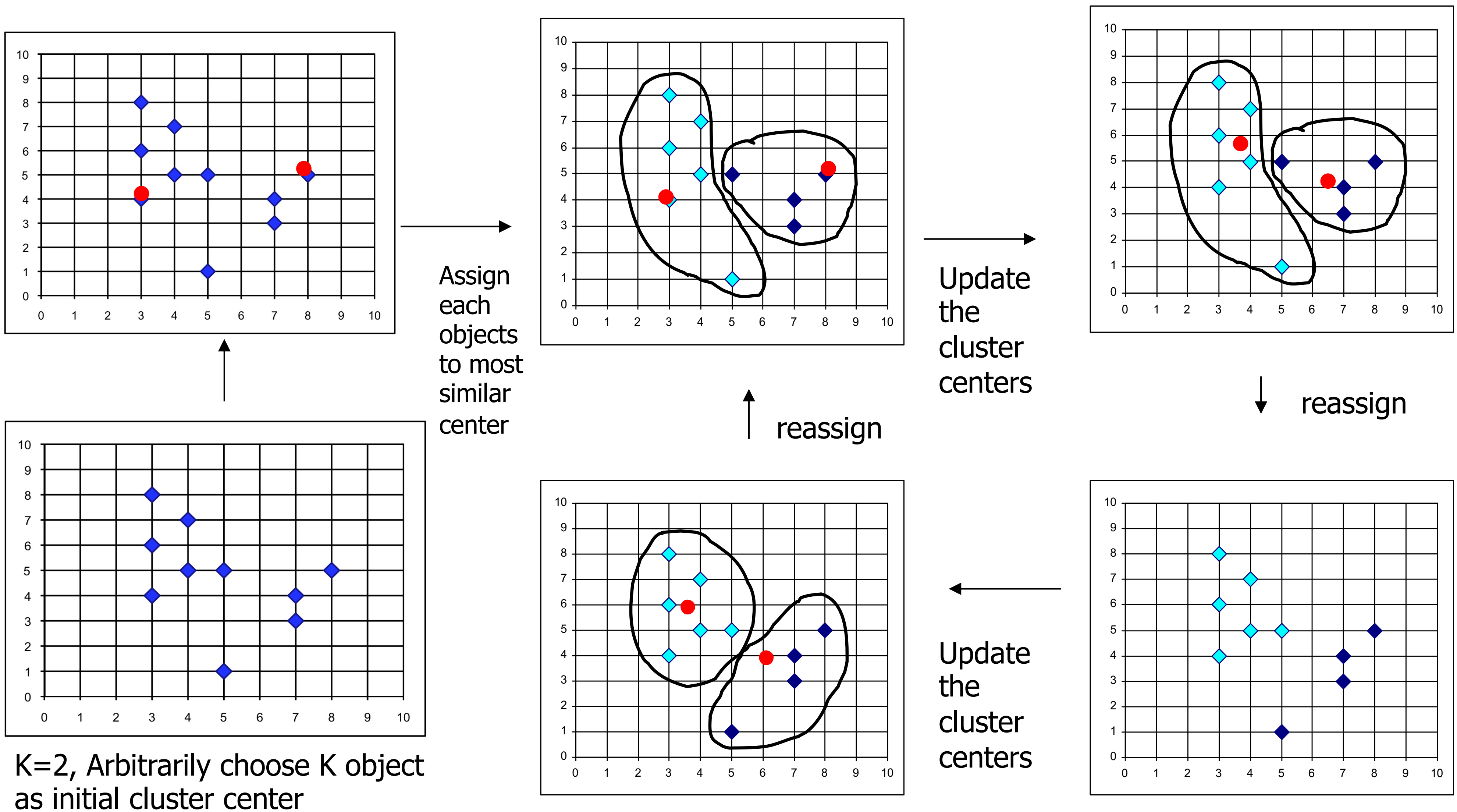
K Means Example (*K*=2)



Pick seeds
Assign X to clusters
Computer cluster centroids
Reassign X to clusters
Recompute cluster centroids

Converged!

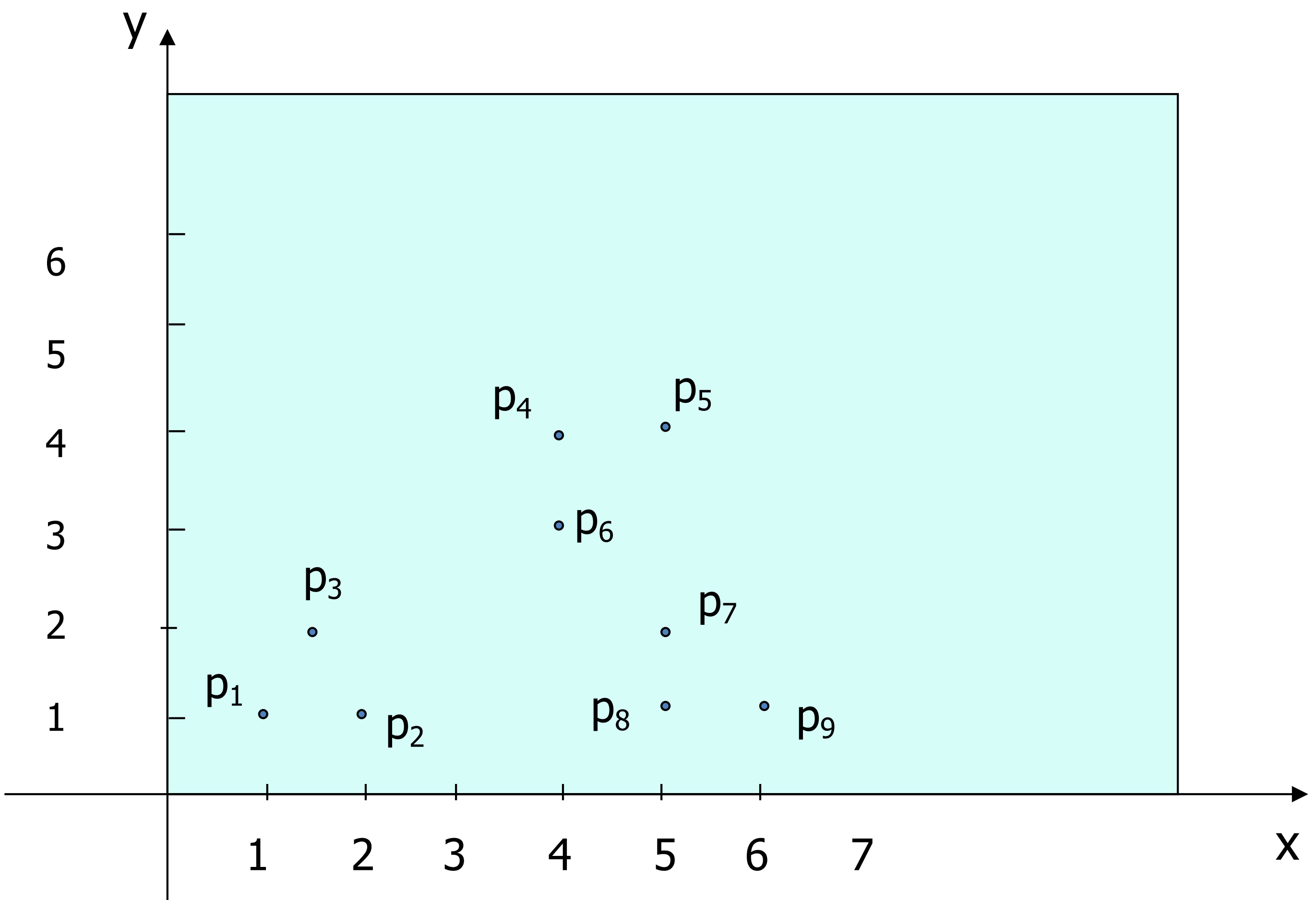
K-Means


Assign each objects to most similar center
Update the cluster centers
reassign
reassign
Update the cluster centers
K=2, Arbitrarily choose K object as initial cluster center

Practice Question

Apply K-means clustering algorithm to partition the following data with 9 data objects:

P1(1, 1)
P2(2,1)
P3(1.5,2)
P4(4, 4)
P5(5,4)
P6(4,3)
P7(5,2)
P8(5,1)
P9(6,1)



Comments on the *K-Means* Method

- <u>Strength</u>
  - *Relatively efficient*: *O*(*tkn*), where *n* is # objects, *k* is # clusters, and *t* is # iterations. Normally, *k*, *t* << *n*.
  - Often terminates at a *local optimum*. The *global optimum* may be found using techniques such as: *deterministic annealing* and *genetic algorithms*
- <u>Weakness</u>
  - Applicable only when *mean* is defined, then what about categorical data?
  - Need to specify *k,* the *number* of clusters, in advance
  - Sensitive to initial seed selection
  - Unable to handle noisy data and *outliers*

Variations of the *K-Means* Method

- A few variants of the *k-means* which differ in
  - Selection of the initial *k* means
  - Dissimilarity calculations
  - Strategies to calculate cluster means
- Handling categorical data: *k-modes* (Huang' 98)
  - Replacing means of clusters with modes
    - Using a frequency-based method to update modes of clusters
  - Using new dissimilarity measures to deal with categorical objects
  - A mixture of categorical and numerical data: *k-prototype* method

The *K-Medoids* Clustering Method

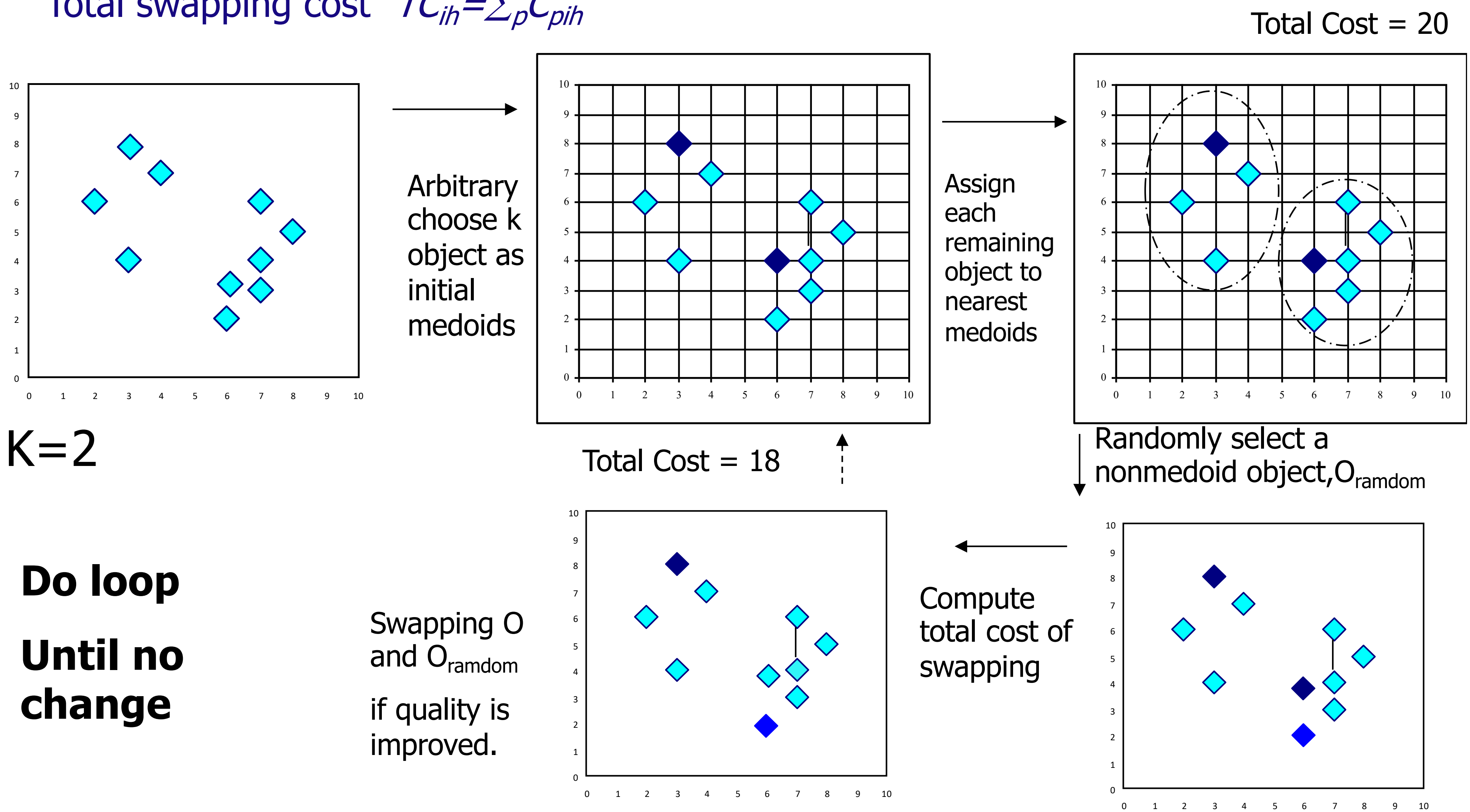
- Find *representative* objects, called medoids, in clusters
- *PAM* (Partitioning Around Medoids, 1987)
  - starts from an initial set of medoids and iteratively replaces one of the medoids by one of the non-medoids if it improves the total distance of the resulting clustering
  - *PAM* works effectively for small data sets, but does not scale well for large data sets
- *CLARA* (Kaufmann & Rousseeuw, 1990)
- *CLARANS* (Ng & Han, 1994): Randomized sampling

K-Medoids

- Arbitrarily choose K objects as the initial medoids;
- Repeat:
  - Assign each remaining object to the cluster with the nearest medoids;
  - Randomly select a nonmedoid object O_{random};
  - Compute the total cost, S, of swapping O_j with O_{random};
  - If $S<0$, then swap O_j with O_{random} to form the new set of k medoids;
- Until no change

k-Medoids

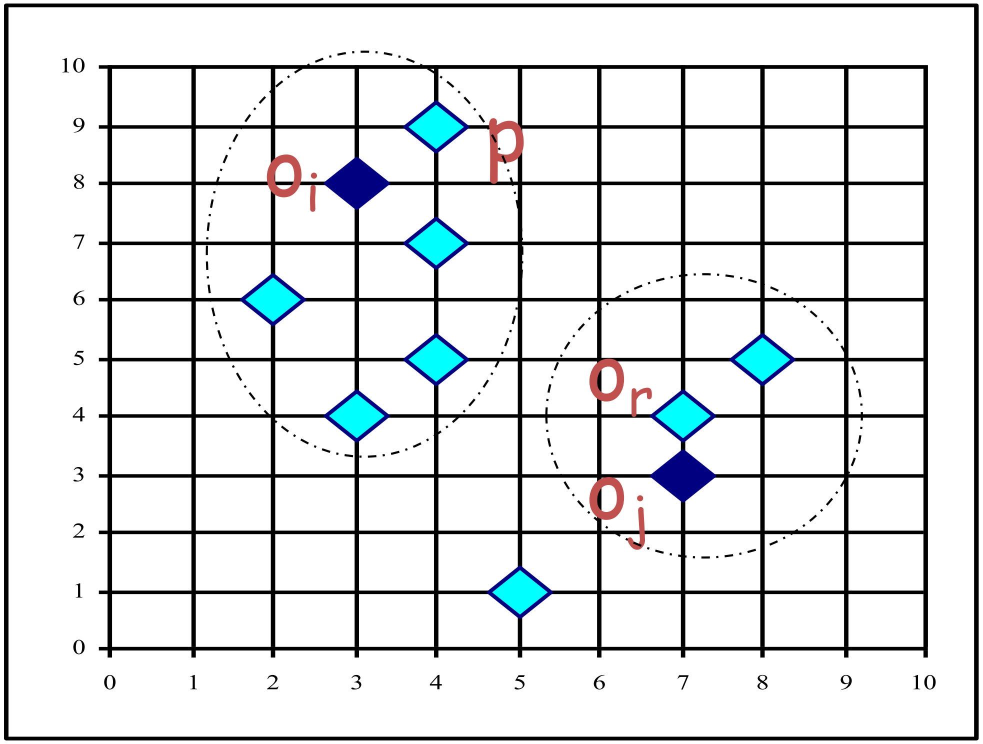
Total swapping cost $TC_{ih}=\sum_p C_{pih}$



Do loop

Until no change

Four Cases – Case A



Replace o_j with o_r

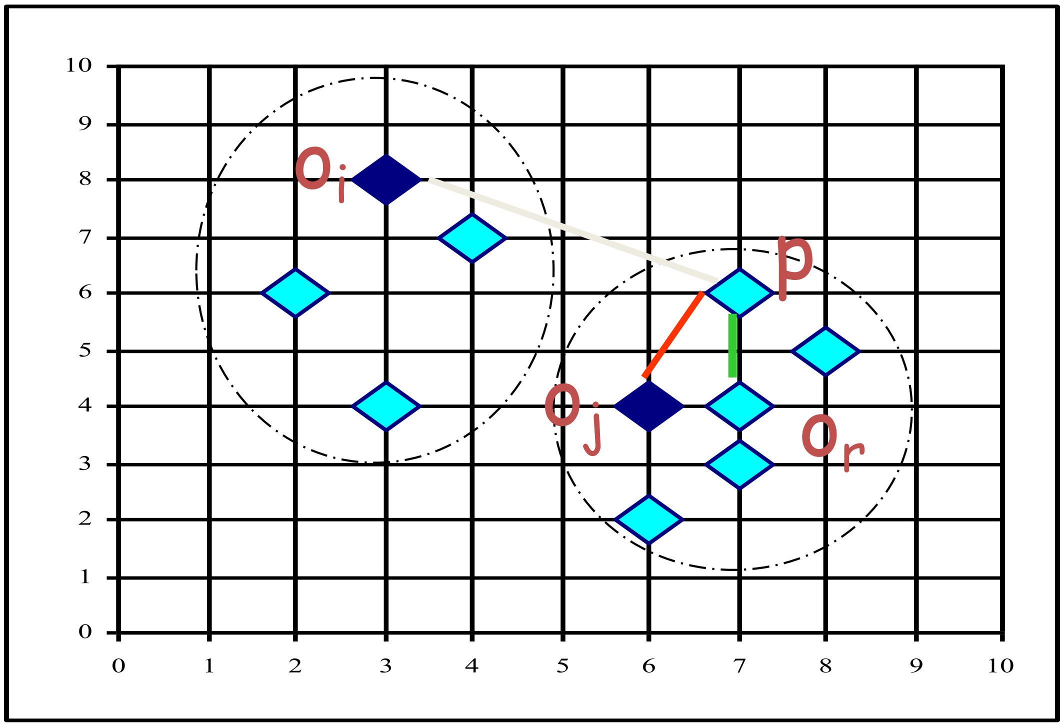
$p \in o_i$; $i \neq j$;

p still closest to o_i

no change

$$C_{p,j,r} = 0$$

Four Cases – Case B



Replace o_j with o_r

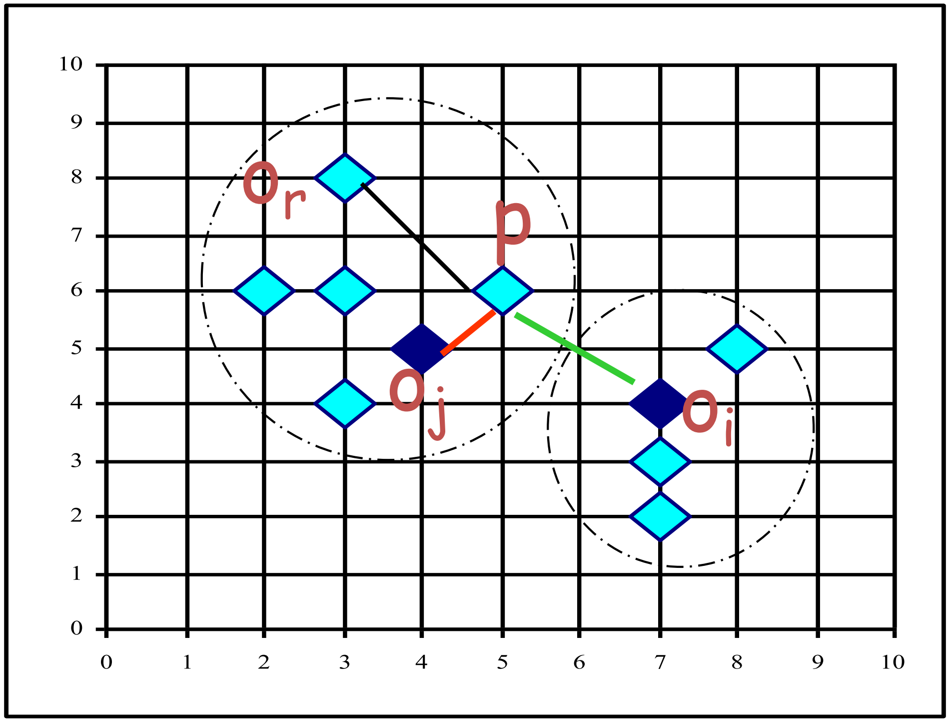
$p \in o_j$

p closest to o_r

Reassign p to O_r

$$C_{p,j,r} = d(p - o_r) - d(p - o_j)$$

Four Cases – Case C



Replace o_j with o_r

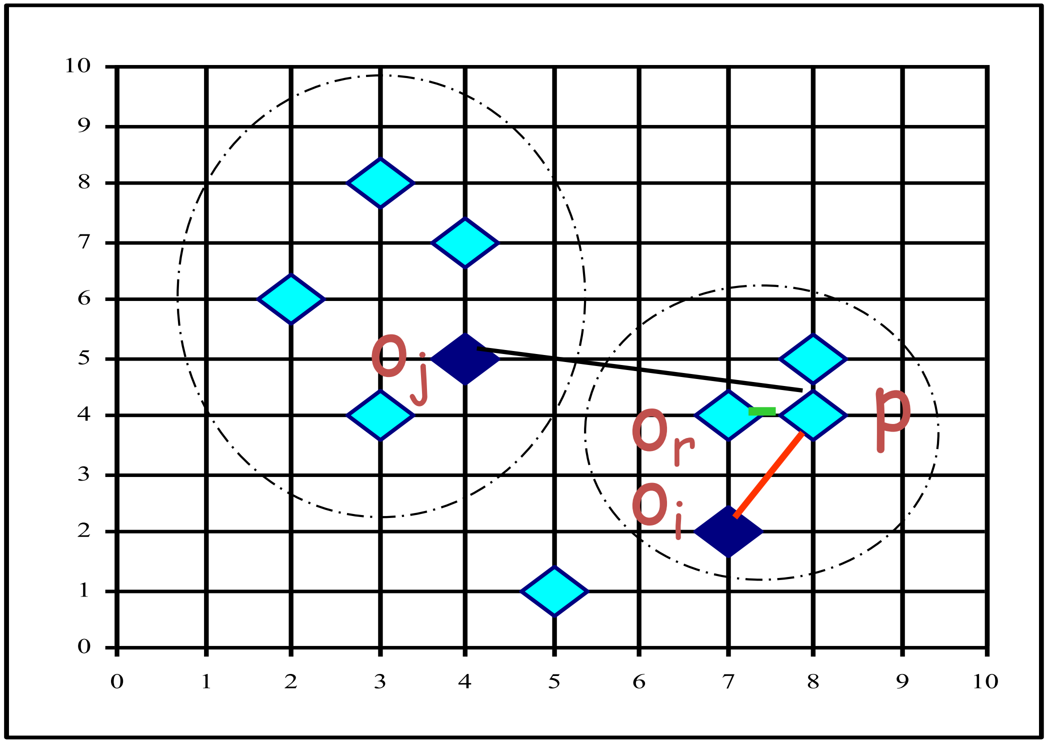
$p \in o_j$

p is now closer to o_i $i \neq j$

Reassign p to O_i

$$C_{p,j,r} = d(p - o_i) - d(p - o_j)$$

Four Cases – Case D



Replace o_j with o_r

$p \in o_i$; $i \neq j$;

p closest to o_r

Reassign p to O_r

$$C_{p,j,r} = d(p - o_r) - d(p - o_i)$$

Practice Question

- Apply PAM on the following data, K=2

	Gender	Age	Time	Fever	Cough
Obj1	F	2	2	Y	N
Obj2	M	2	0.5	N	N
Obj3	F	15	3	Y	Y
Obj4	F	18	0.5	Y	N
Obj5	M	58	4	N	Y
Obj6	F	44	14	N	Y

Assuming O_1 and O_5 are the medoids of the 2 clusters initially, after objects are distributed to the two clusters, and we randomly selected O_2 to replace O_1 as the new medoid, should this replacement be carried out?

Practice Question

Assuming O_1 and O_5 are the medoids of the 2 clusters initially, after objects are distributed to the two clusters, and we randomly selected O_2 to replace O_1 as the new medoid, should this replacement be carried out?

The distance(dissimilarity) between pairwise objects

	O_1	O_2	O_3	O_4	O_5	O_6
O_1	--					
O_2	0.94	--				
O_3	0.36	0.91	--			
O_4	0.19	0.75	0.39	--		
O_5	1.15	1.38	0.99	1.3	--	
O_6	1.38	2.16	1.22	1.5	1.2	--

Practice Question (2)

Assuming the distance(dissimilarity) between pairwise objects is as the following:

	O_1	O_2	O_3	O_4	O_5	O_6
O_1	--					
O_2	0.94	--				
O_3	0.36	0.91	--			
O_4	0.19	0.75	0.39	--		
O_5	1.15	1.38	0.99	1.3	--	
O_6	1.38	2.16	1.22	1.5	1.2	--

Assuming O_1 and O_3 are the medoids of the 2 clusters initially, after objects are distributed to the two clusters, and we randomly selected O_2 to replace O_3 as the new medoid for cluster 2, whats the total cost for this replacement? should this replacement be carried out?

PAM Complexity Analysis

- Total k*(n-k) pairs of (O_i, O_h), k is the number of clusters
- For each pair of (O_i, O_h):
  - compute Tc_{ih} require the examination of (n-k) non-selected objects.
- Total complexity:

  $O(k*(n-k)^2)$

Compare K-means and PAM

- K-means is computationally more efficient
- K-means only handles numeric data
- PAM can handle different types of data
- PAM is better in terms of handling outliers in data

The CLARA algorithm

- Objective: to improve the computational efficiency of PAM, through sampling
- Basic idea:
  - draw a sample (size=40+2k) from the original data set, apply PAM on the sample, and finds the medoids of the sample.
  - Repeat the process a fixed number of times and return the medoids that generate the lowest average dissimilarity from the data objects
- Complexity: $O(k*(40+k)^2 + k*(n-k))$

The CLARA Algorithm

for i=1 to 5, repeat the following steps:

- Draw a sample of 40+2k objects randomly from the entire data set, and call algorithm PAM to find the *k* medoids of the sample
- For each object O_j in the entire data set, determine which of the *k* medoids is the most similar to $O_{j.}$
- Calculate the average dissimilarity of the clustering obtained in the previous step. If this value is < current minimum, set current minimum to this value, and retain the current set of *k* medoids
- Return to step 1 to start the next iteration

CLARANS ("Randomized" CLARA)

- *CLARANS* (A Clustering Algorithm based on Randomized Search)
- CLARANS draws sample of ***neighbors*** dynamically
- The clustering process can be presented as searching a graph where every node is a potential solution, that is, a set of *k* medoids
- If the local optimum is found, *CLARANS* starts with new randomly selected node in search for a new local optimum
- It is more efficient and scalable than both *PAM* and *CLARA*

The CLARANS Algorithm

1. Input *numlocal* and *maxneighbor*
   i=1, mincost=FLT_MAX, bestnode=NULL
2. *current* = an arbitrary *k* modiods
3. j=1
4. Pick random neighbor *S* of *current*, compute the cost difference between *S* and *current*
5. If S has lower cost, set *current* = S, goto 3
   else
      j=j+1;
      if (j <=*maxneighbor*) goto 4
      else
         if (cost(*current*) < *mincost*)
            mincost = cost(*current*)
            bestnode = *current*
6. i= i+1;
7. If (i <= *numlocal*)
      goto step 2
   else
      output *bestnode* and halt

The CLARANS Algorithm

Outer loop:
i → iterate for numlocal times
    find the local maximum point and update "mincost" and "bestnode"
    inner loop:
    j → iterate for maxneighbor times
        finding the best local maximum k medoids