Lecture 24: Clustering

Artificial Intelligence
CS-GY-6613-I
Julian Togelius
julian.togelius@nyu.edu

Types of learning

- **Supervised learning**
  Learning to predict or classify labels based on labeled input data

- **Unsupervised learning**
  Finding patterns in unlabeled data

- **Reinforcement learning**
  Learning well-performing behavior from state observations and rewards

How Much Information Does the Machine Need to Predict?

Y LeCun

Source: Yann LeCun

- **"Pure" Reinforcement Learning (cherry)**
  - The machine predicts a scalar reward given once in a while.
  - **A few bits for some samples**
- **Supervised Learning (icing)**
  - The machine predicts a category or a few numbers for each input
  - Predicting human-supplied data
  - **10→10,000 bits per sample**
- **Unsupervised/Predictive Learning (cake)**
  - The machine predicts any part of its input for any observed part.
  - Predicts future frames in videos
  - **Millions of bits per sample**

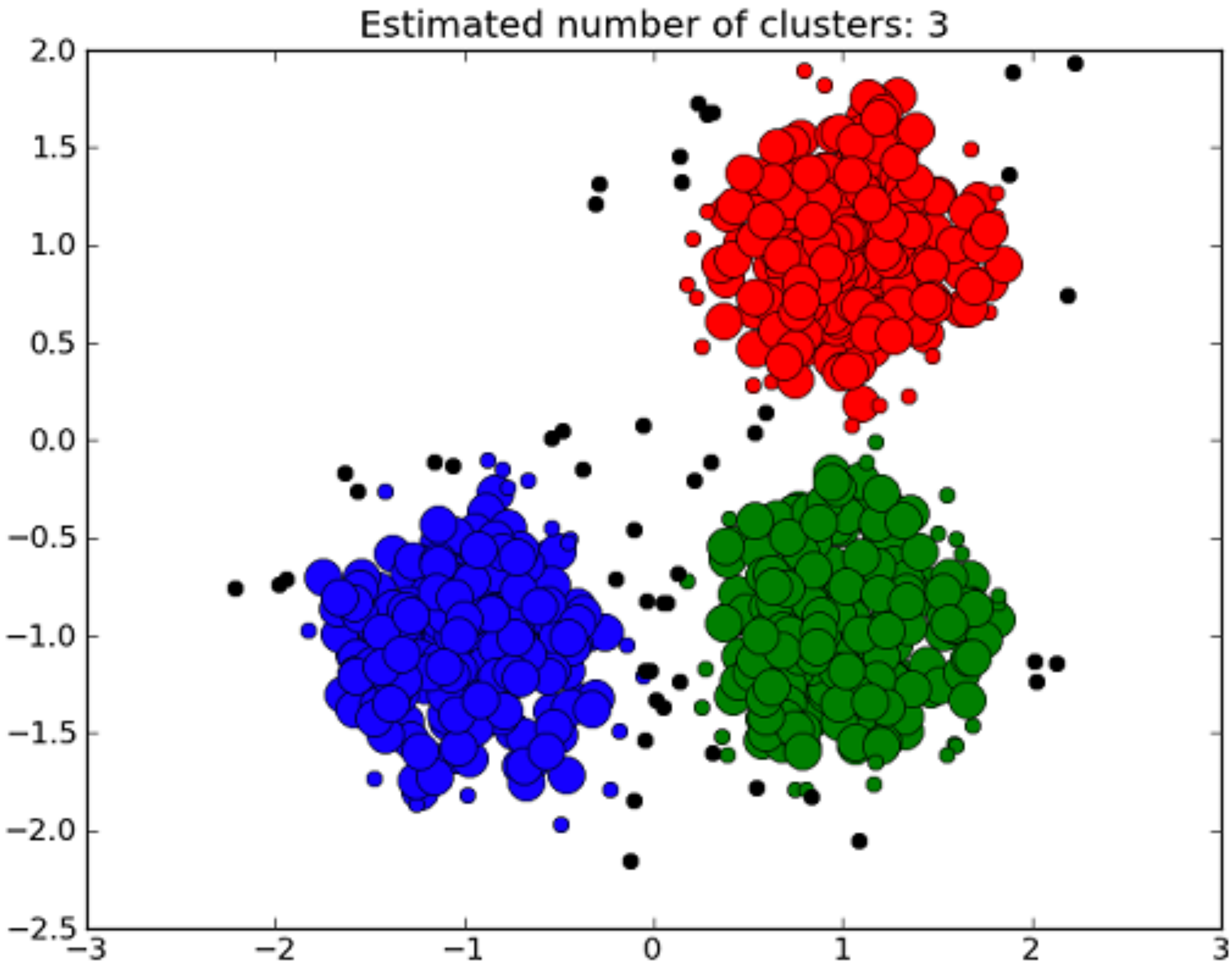
- (Yes, I know, this picture is slightly offensive to RL folks. But I'll make it up)

Unsupervised learning

- Clustering
- Dimensionality reduction
- Data compression
- Generative Adversarial Networks
- Sequence learning (?)

Clustering

- Cluster: a collection of data objects
  - Similar to one another within the same cluster
  - Dissimilar to the objects in other clusters
- Cluster analysis: Finding similarities between data according to the characteristics found in the data and grouping similar data objects into clusters
- Unsupervised learning: no predefined classes

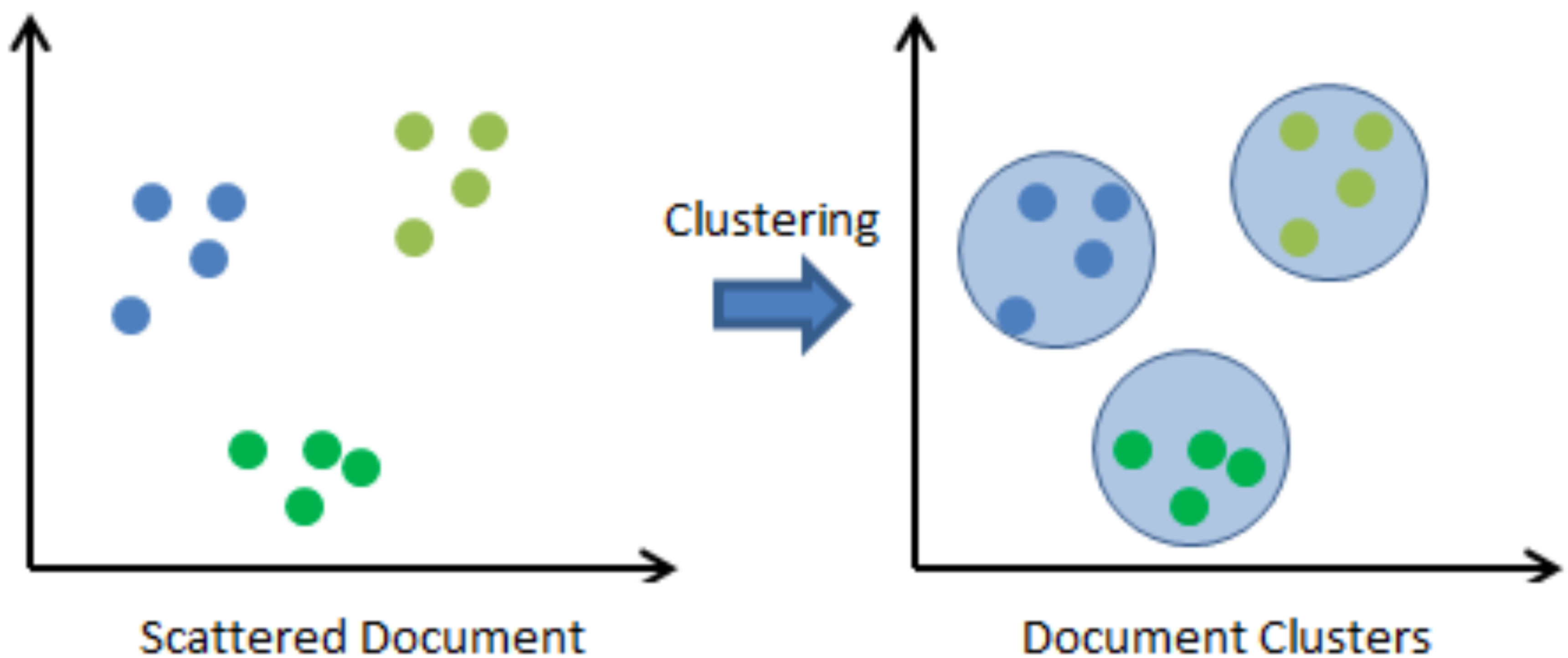


Estimated number of clusters: 3
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2.5
−3
−2
−1
0
1
2
3

Applications

- As a stand-alone tool to get insight into data distribution

- As a preprocessing step for other algorithms



- Marketing: Help marketers discover distinct groups in their customer bases, and then use this knowledge to develop targeted marketing programs

- Land use: Identification of areas of similar land use in an earth observation database

- Insurance: Identifying groups of motor insurance policy holders with a high average claim cost

- City-planning: Identifying groups of houses according to their house type, value, and geographical location

- Earth-quake studies: Observed earth quake epicenters should be clustered along continent faults

- Games: identify player groups / archetypes

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 implementation
- The quality of a clustering method is also measured by its ability to discover some or all of the hidden patterns

Similarity and Dissimilarity Between Objects

- Distances are normally used to measure the similarity or dissimilarity between two data objects
- Some popular ones include: Minkowski distance:

$$d(i,j) = \sqrt[q]{(|x_{i_1} - x_{j_1}|^q + |x_{i_2} - x_{j_2}|^q + \ldots + |x_{i_p} - x_{j_p}|^q)}$$

where i = (x_{i1}, x_{i2}, …, x_{ip}) and j = (x_{j1}, x_{j2}, …, x_{jp}) are two p-dimensional data objects, and q is a positive integer

- If q = 1, d is Manhattan distance

$$d(i,j) = |x_{i_1} - x_{j_1}| + |x_{i_2} - x_{j_2}| + \ldots + |x_{i_p} - x_{j_p}|$$

Similarity and Dissimilarity Between Objects (Cont.)

- If q = 2, d is Euclidean distance:

$$d(i,j) = \sqrt{(|x_{i_1} - x_{j_1}|^2 + |x_{i_2} - x_{j_2}|^2 + ... + |x_{i_p} - x_{j_p}|^2)}$$

- Also, one can use weighted distance, parametric Pearson product moment correlation, or other dissimilarity measures

Some requirements…

- Scalability
- Ability to deal with different types of attributes
- Ability to handle dynamic data
- Discovery of clusters with arbitrary shape
- Able to deal with noise and outliers
- Insensitive to order of input records
- High dimensionality
- Incorporation of user-specified constraints

Clustering approaches

- Partitioning approach: Construct various partitions and then evaluate them by some criterion, e.g., minimizing the sum of square errors
    - Typical methods: k-means, k-medoids, CLARANS
- Hierarchical approach: Create a hierarchical decomposition of the set of data (or objects) using some criterion
    - Typical methods: Diana, Agnes, BIRCH, ROCK, CAMELEON
- Density-based approach: Based on connectivity and density functions
    - Typical methods: DBSCAN, OPTICS, DenClue

- Grid-based approach: based on a multiple-level granularity structure
    - Typical methods: STING, WaveCluster, CLIQUE
- Model-based: A model is hypothesized for each of the clusters and tries to find the best fit of that model to each other
    - Typical methods: EM, SOM, COBWEB
- Frequent pattern-based: Based on the analysis of frequent patterns
    - Typical methods: pCluster
- User-guided or constraint-based: Clustering by considering user-specified or application-specific constraints
    - Typical methods: COD (obstacles), constrained clustering

In this class

- Partitioning approaches
- Hierarchical approaches
- Measuring cluster quality

Partitioning algorithms

- Partitioning method: Construct a partition of a database D of n objects into a set of k clusters, s.t., min sum of squared distance

$$\Sigma_{m=1}^{k}\Sigma_{t_{mi}\in Km}(C_m - t_{mi})^2$$

- Given a k, find a partition of k clusters that optimizes the chosen partitioning criterion

- Which is the simplest possible clustering algorithm?

Partitioning algorithms

- 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

k-means

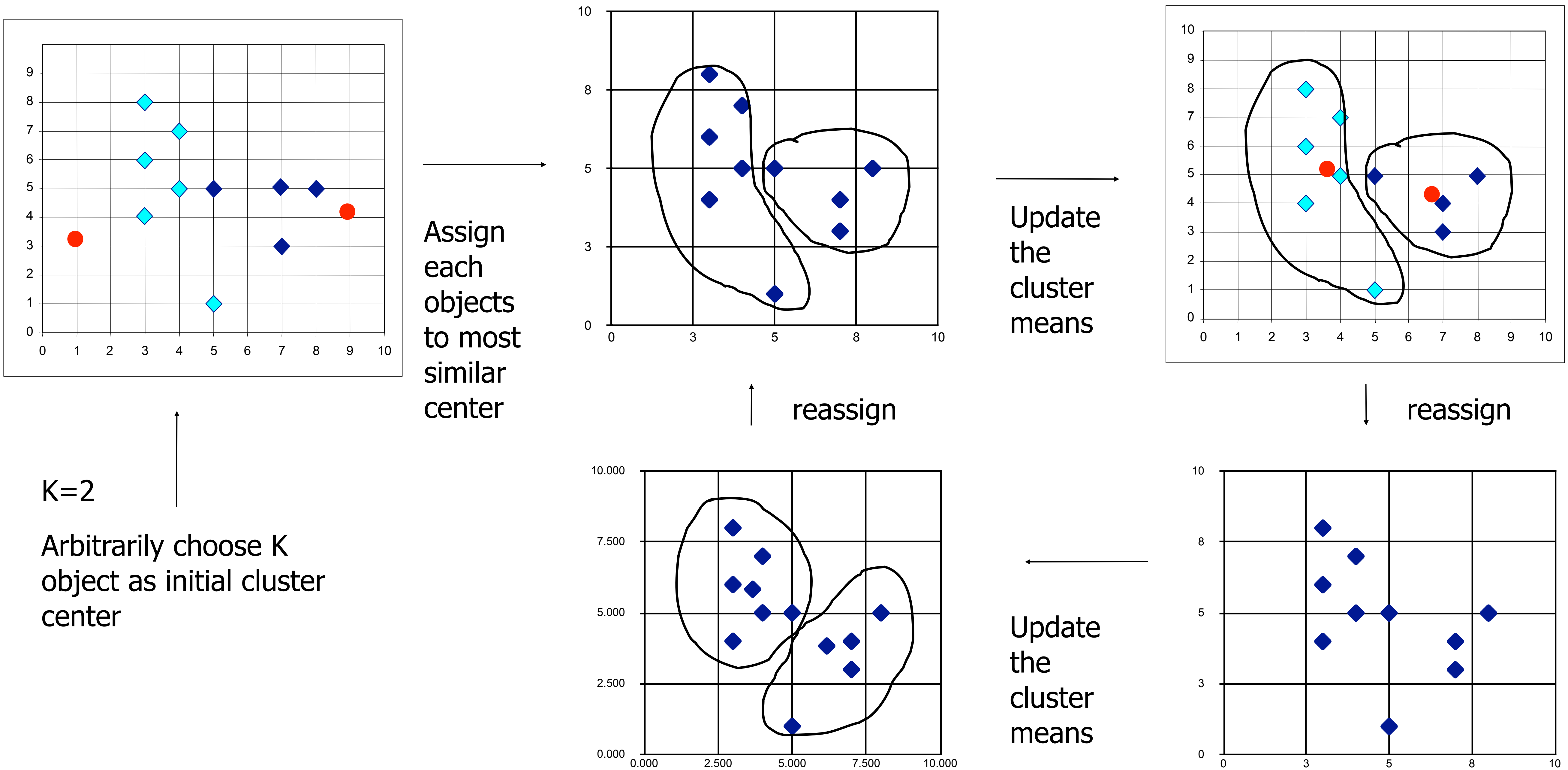
- Given k, the k-means algorithm is implemented in four steps:

    1. Partition objects into k nonempty subsets

    2. Compute seed points as the centroids of the clusters of the current partition (the centroid is the center, i.e., mean point, of the cluster)

    3. Assign each object to the cluster with the nearest seed point

    4. Go back to Step 2, stop when no more new assignment


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

Algorithm: *k*-means. The *k*-means algorithm for partitioning, where each cluster's center is represented by the mean value of the objects in the cluster.

Input:

- *k*: the number of clusters,
- *D*: a data set containing *n* objects.

Output: A set of *k* clusters.

Method:

(1) arbitrarily choose *k* objects from *D* as the initial cluster centers;
(2) **repeat**
(3)   (re)assign each object to the cluster to which the object is the most similar, based on the mean value of the objects in the cluster;
(4)   update the cluster means, that is, calculate the mean value of the objects for each cluster;
(5) **until** no change;

- *Strength*: Relatively efficient: $O(tkn)$, where n is # objects, k is # clusters, and t is # iterations. Normally, $k, t << n$.
- Comparing: PAM: $O(k(n-k)^2)$, CLARA: $O(ks^2 + k(n-k))$
- *Comment*: Often terminates at a local optimum. The global optimum may be found using techniques such as: deterministic annealing and genetic algorithms
- *Weaknesses:*
  - Applicable only when mean is defined, then what about categorical data?
  - Need to specify k, the number of clusters, in advance
  - Unable to handle noisy data and outliers
  - Not suitable to discover clusters with non-convex shapes

Variations

- 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

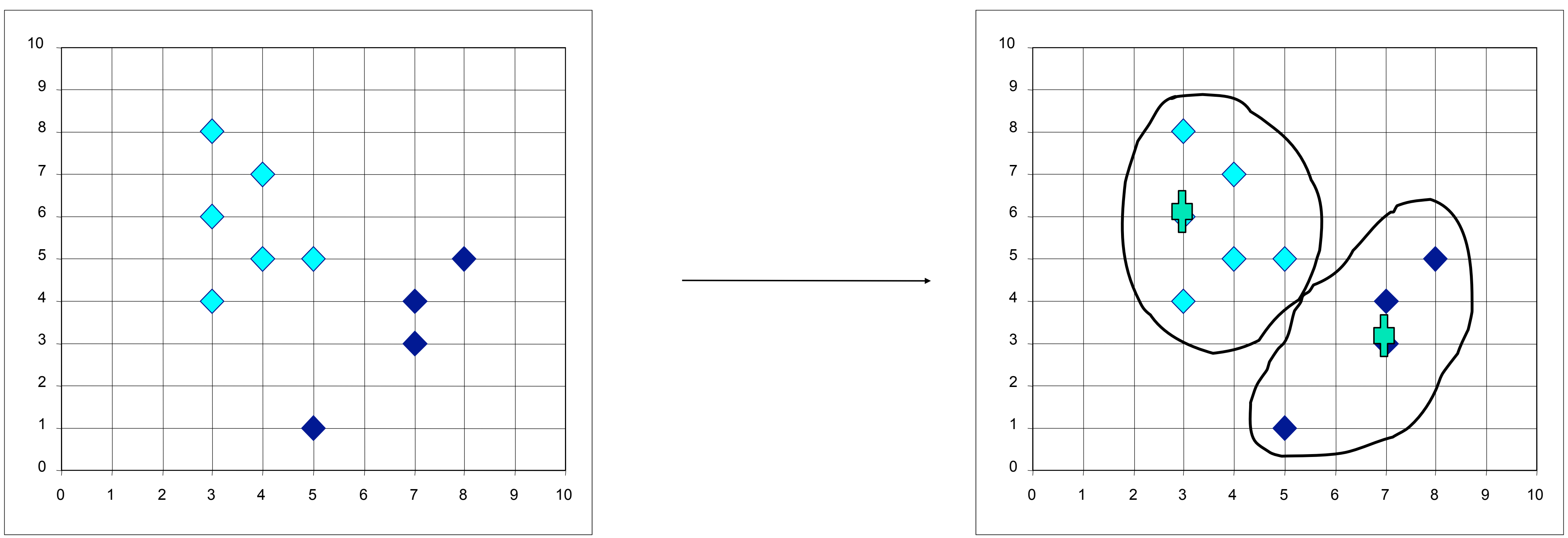
- Handling categorical data: k-modes (Huang'98)
  - Replacing means of clusters with modes
  - Using new dissimilarity measures to deal with categorical objects
  - Using a frequency-based method to update modes of clusters
  - A mixture of categorical and numerical data: k-prototype method

A problem with k-means

- The k-means algorithm is sensitive to outliers !
- Since an object with an extremely large value may substantially distort the distribution of the data.

k-medoids

- Instead of taking the mean value of the object in a cluster as a reference point, medoids can be used, which is the most centrally located object in a cluster.



k-medoids

- 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
- Focusing + spatial data structure (Ester et al., 1995)

- *PAM (Kaufman and Rousseeuw, 1987)*

- Use real object to represent the cluster

    1. Select k representative objects arbitrarily

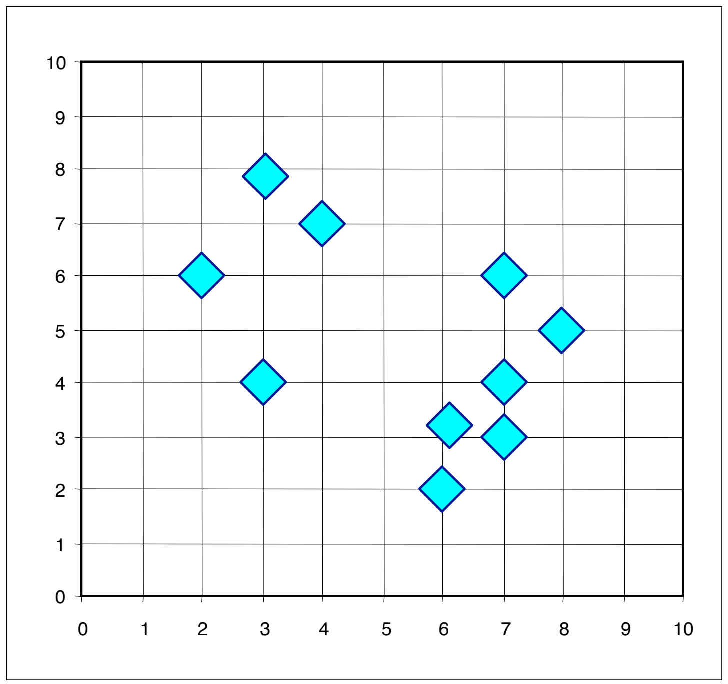
    2. For each pair of non-selected object h and selected object i, calculate the total swapping cost TCih

    3. For each pair of i and h, if TCih < 0, i is replaced by h

    4. Then assign each non-selected object to the most similar representative object

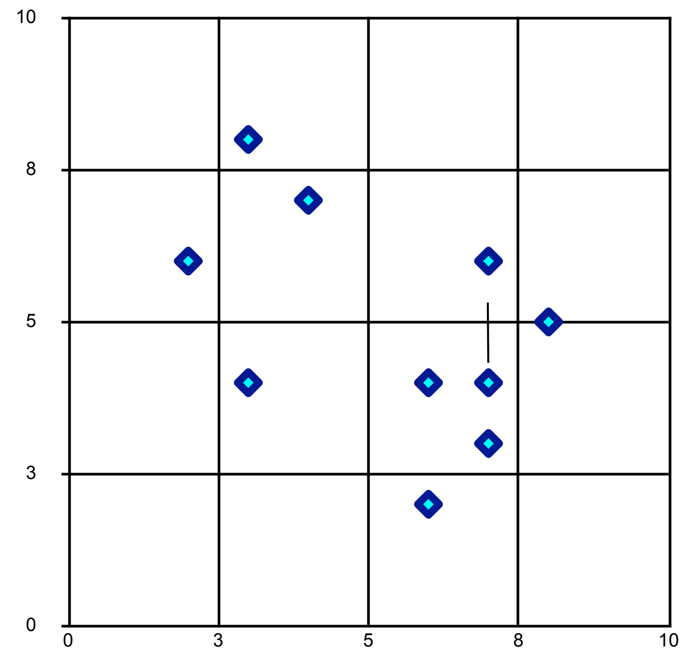
    5. repeat steps 2-4 until there is no change

PAM



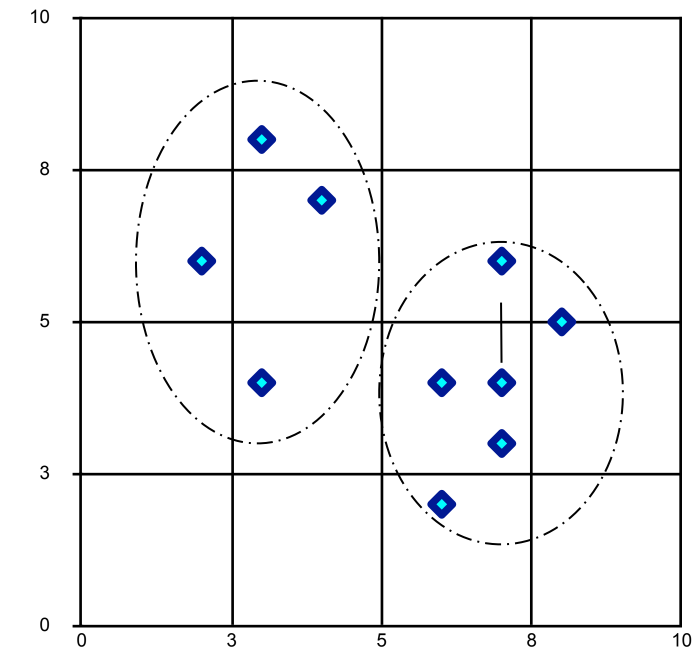
K=2

Arbitrary choose k object as initial medoids

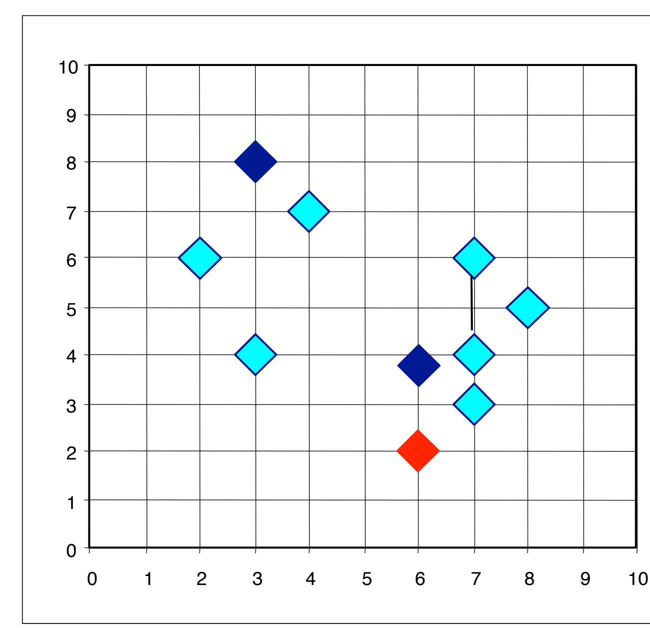


Assign each remaining object to nearest medoids

Total Cost = 20

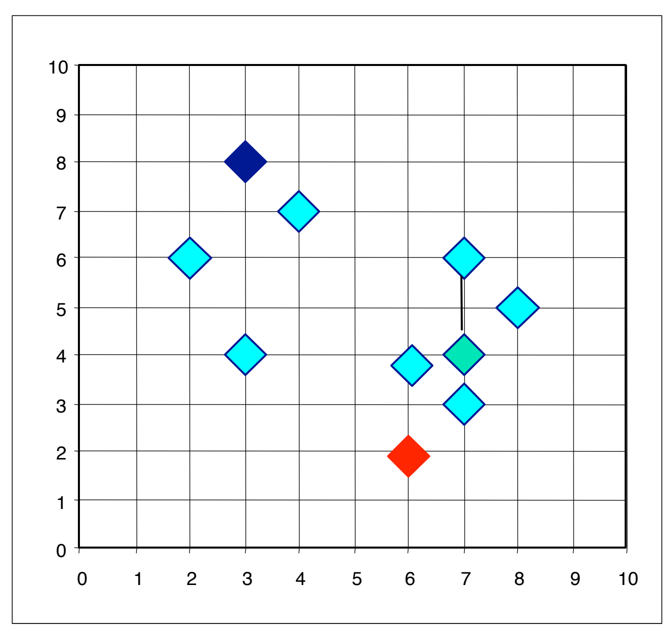


Randomly select a nonmedoid object,O_{ramdom}



Compute total cost of swapping

Total Cost = 26



Swapping O and O_{ramdom}

If quality is improved.

Do loop

Until no change

PAM problem

- Pam is more robust than k-means in the presence of noise and outliers because a medoid is less influenced by outliers or other extreme values than a mean

- Pam works efficiently for small data sets but does not scale well for large data sets.

- $O(k(n-k)^2)$ for each iteration where n is # of data, k is # of clusters