UNSUPERVISED LEARNING

CO-4 SESSION-5

AIM

To familiarize students with the concepts of unsupervised machine learning, its difference with supervised machine learning and the use of unsupervised learning, particularly clustering

INSTRUCTIONAL OBJECTIVES

This session is designed to:

1. Introduction to unsupervised learning
2. K-means algorithm
3. Representation of clusters

LEARNING OUTCOMES

At the end of this session, you should be able to:

1. Supervised learning vs. unsupervised learning
2. Clustering algorithm
3. K-means clustering
4. Common ways to represent clusters

Supervised learning vs. unsupervised learning

Supervised learning: discover patterns in the data that relate data attributes with a target (class) attribute.
These patterns are then utilized to predict the values of the target attribute in future data instances.

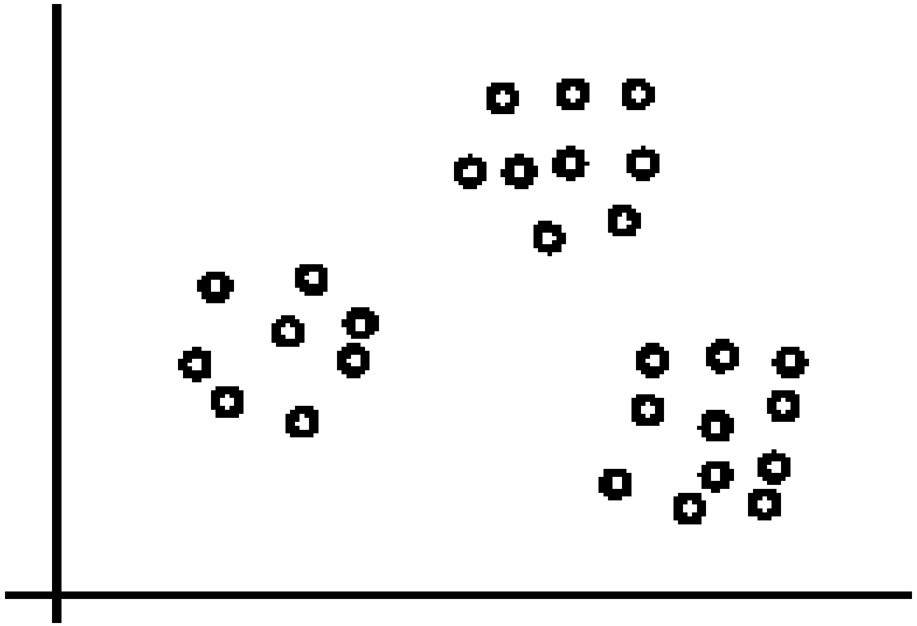
Unsupervised learning: The data have no target attribute.
We want to explore the data to find some intrinsic structures in them.

Clustering

- Clustering is a technique for finding similarity groups in data, called **clusters**. I.e.,
  - it groups data instances that are similar to (near) each other in one cluster and data instances that are very different (far away) from each other into different clusters.
- Clustering is often called an **unsupervised learning** task as no class values denoting an *a priori* grouping of the data instances are given, which is the case in supervised learning.
- Due to historical reasons, clustering is often considered synonymous with unsupervised learning.
  - In fact, association rule mining is also unsupervised

An illustration

- The data set has three natural groups of data points, i.e., 3 natural clusters.



What is clustering for?

- Let us see some real-life examples

- Example 1: groups people of similar sizes together to make “small”, “medium” and “large” T-Shirts.
  - Tailor-made for each person: too expensive
  - One-size-fits-all: does not fit all.

- Example 2: In marketing, segment customers according to their similarities
  - To do targeted marketing.

What is clustering for? (cont.…)

- Example 3: Given a collection of text documents, we want to organize them according to their content similarities,
  - To produce a topic hierarchy
- In fact, clustering is one of the most utilized machine learning techniques.
  - It has a long history, and used in almost every field, e.g., medicine, psychology, botany, sociology, biology, archeology, marketing, insurance, libraries, etc.
  - In recent years, due to the rapid increase of online documents, text clustering becomes important.

KL (DEEMED TO BE UNIVERSITY)

Aspects of clustering

- A clustering algorithm
  - Partitional clustering
  - Hierarchical clustering
  - …
- A distance (similarity, or dissimilarity) function
- Clustering quality
  - Inter-clusters distance $\Rightarrow$ maximized
  - Intra-clusters distance $\Rightarrow$ minimized
- The quality of a clustering result depends on the algorithm, the distance function, and the application.

K-means clustering

- K-means is a partitional clustering algorithm
- Let the set of data points (or instances) D be $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$, where $\mathbf{x}_i = (x_{i1}, x_{i2}, \ldots, x_{ir})$ is a vector in a real-valued space $X \subseteq R^r$, and r is the number of attributes (dimensions) in the data.
- The k-means algorithm partitions the given data into k clusters.
  - Each cluster has a cluster **center**, called **centroid**.
  - k is specified by the user

K-means algorithm

Given *k*, the *k-means* algorithm works as follows:

1) Randomly choose *k* data points (seeds) to be the initial centroids, cluster centers
2) Assign each data point to the closest centroid
3) Re-compute the centroids using the current cluster memberships.
4) If a convergence criterion is not met, go to 2).

K-means algorithm – (cont.…)

Given *k*, the *k-means* algorithm works as follows:

1) Randomly choose *k* data points (seeds) to be the initial centroids, cluster centers
2) Assign each data point to the closest centroid
3) Re-compute the centroids using the current cluster memberships.
4) If a convergence criterion is not met, go to 2).

K-means algorithm – (cont.…)

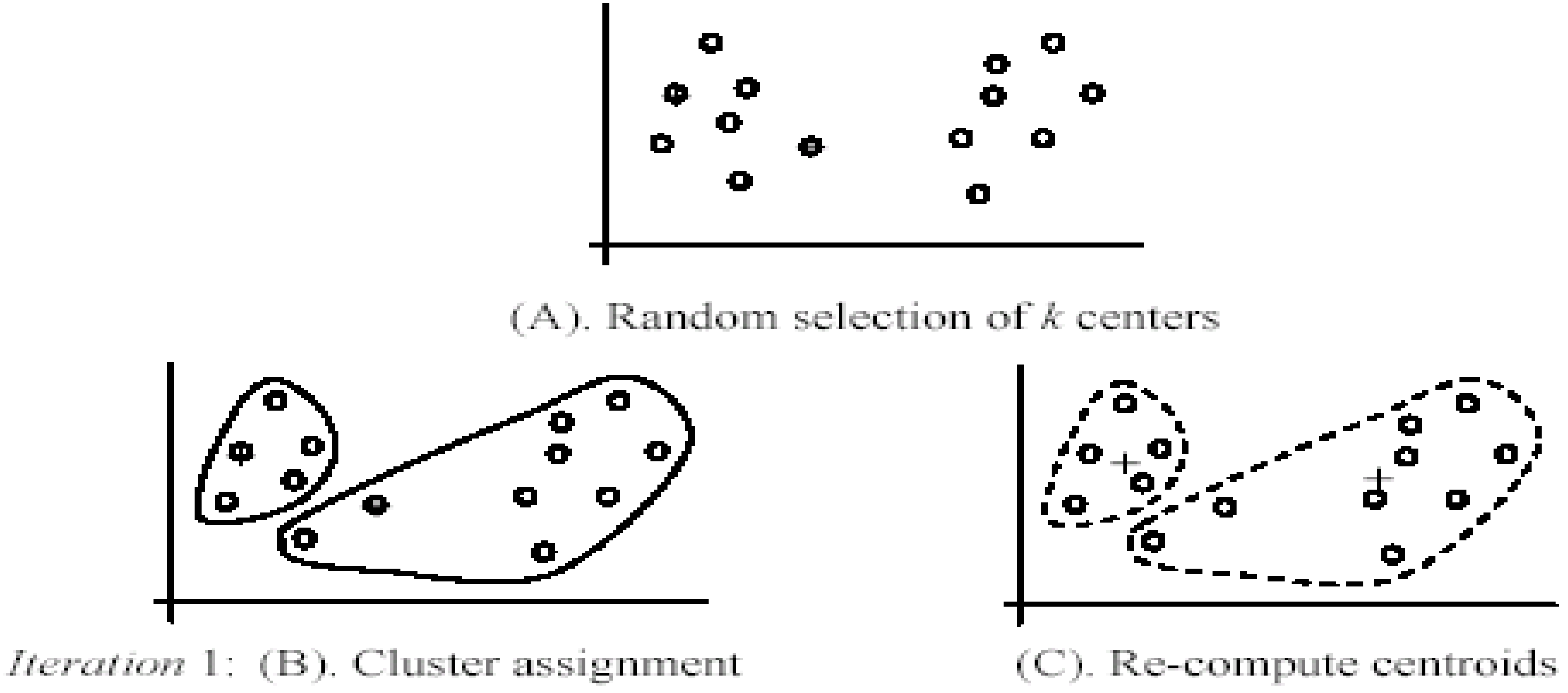
```
Algorithm k-means(k, D)
    Choose k data points as the initial centroids (cluster centers)
    repeat
        for each data point x ∈ D do
            compute the distance from x to each centroid;
            assign x to the closest centroid        // a centroid represents a cluster
        endfor
        re-compute the centroids using the current cluster memberships
    until the stopping criterion is met
```

Stopping/convergence criterion

- no (or minimum) re-assignments of data points to different clusters
- no (or minimum) change of centroids, or
- minimum decrease in the **sum of squared error** (SSE),

  - C_j is the *j*th cluster, $\mathbf{m}_j$ is the centroid of cluster C_j (the mean vector of all the data points in C_j), and *dist*($\mathbf{x}$, $\mathbf{m}_j$) is the distance between data point $\mathbf{x}$ and centroid $\mathbf{m}_j$.

An example

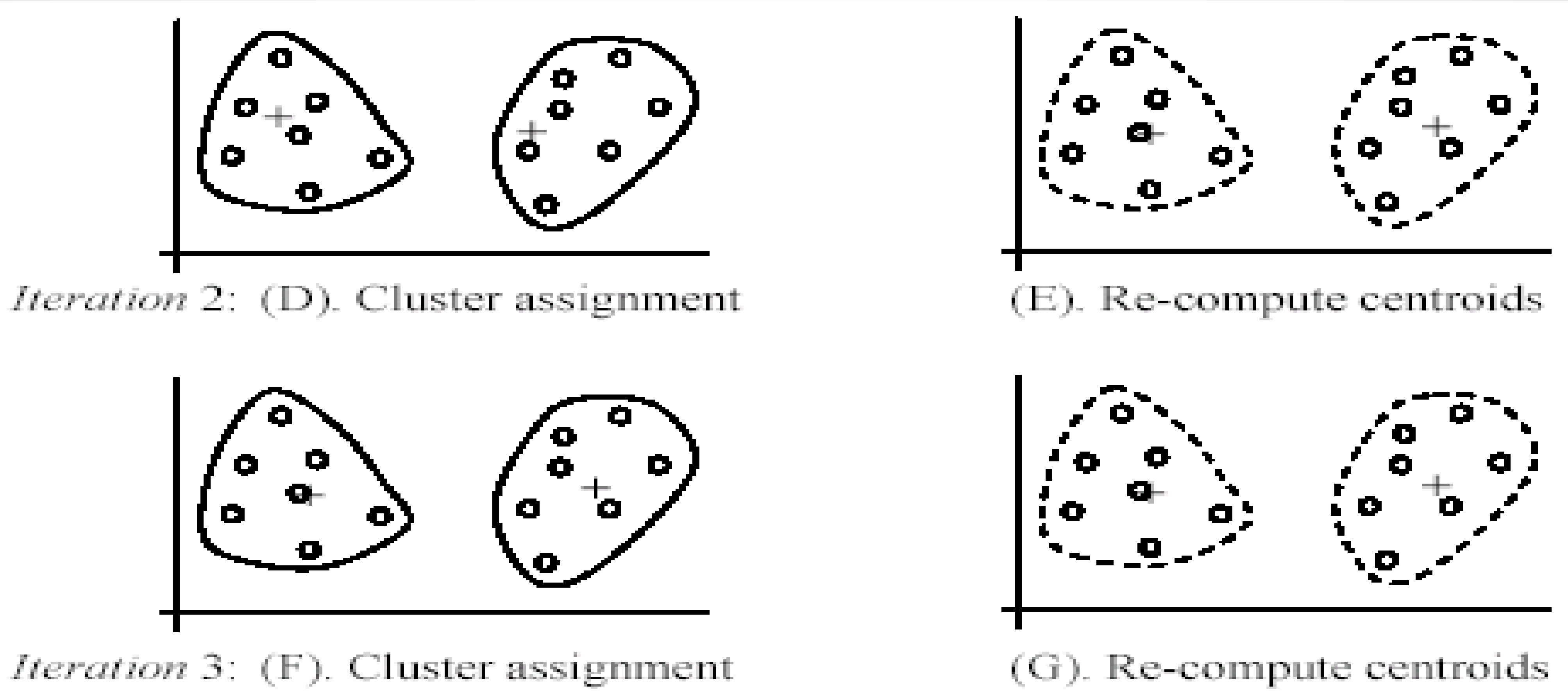


(A). Random selection of k centers

Iteration 1: (B). Cluster assignment

(C). Re-compute centroids

An example (cont….)



An example distance function

The k-means algorithm can be used for any application data set where the **mean** can be defined and computed. In the **Euclidean space**, the mean of a cluster is computed with:

$$\mathbf{m}_j = \frac{1}{|C_j|} \sum_{\mathbf{x}_i \in C_j} \mathbf{x}_i \qquad (2)$$

where $|C_j|$ is the number of data points in cluster C_j. The distance from one data point $\mathbf{x}_i$ to a mean (centroid) $\mathbf{m}_j$ is computed with

$$dist(\mathbf{x}_i, \mathbf{m}_j) = \| \mathbf{x}_i - \mathbf{m}_j \| \qquad (3)$$

$$= \sqrt{(x_{i1} - m_{j1})^2 + (x_{i2} - m_{j2})^2 + \ldots + (x_{ir} - m_{jr})^2}$$

A disk version of *k*-means

- K-means can be implemented with data on disk
  - In each iteration, it scans the data once.
  - as the centroids can be computed incrementally
- It can be used to cluster large datasets that do not fit in main memory
- We need to control the number of iterations
  - In practice, a limited is set (< 50).
- Not the best method. There are other scale-up algorithms, e.g., BIRCH.

A disk version of k-means (cont …)

Algorithm disk-k-means(k, D)

Choose k data points as the initial centriods $\mathbf{m}_j$, $j = 1, \ldots, k$;
repeat
    initialize $\mathbf{s}_j = \mathbf{0}$, $j = 1, \ldots, k$;   // $\mathbf{0}$ is a vector with all 0's
    initialize $n_j = 0$, $j = 1, \ldots, k$;   // n_j is the number points in cluster j
    for each data point $\mathbf{x} \in D$ **do**
        $j = \arg\min_{j} dist(\mathbf{x}, \mathbf{m}_j)$;
        assign $\mathbf{x}$ to the cluster j;
        $\mathbf{s}_j = \mathbf{s}_j + \mathbf{x}$;
        $n_j = n_j + 1$;
    endfor
    $\mathbf{m}_j = \mathbf{s}_j / n_j$, $i = 1, \ldots, k$;
until the stopping criterion is met

Strengths of k-means

- Strengths:
  - Simple: easy to understand and to implement
  - Efficient: Time complexity: $O(tkn)$, where n is the number of data points, k is the number of clusters, and t is the number of iterations.
  - Since both k and t are small. k-means is considered a linear algorithm.

- K-means is the most popular clustering algorithm.

- Note that: it terminates at a local optimum if SSE is used. The global optimum is hard to find due to complexity.

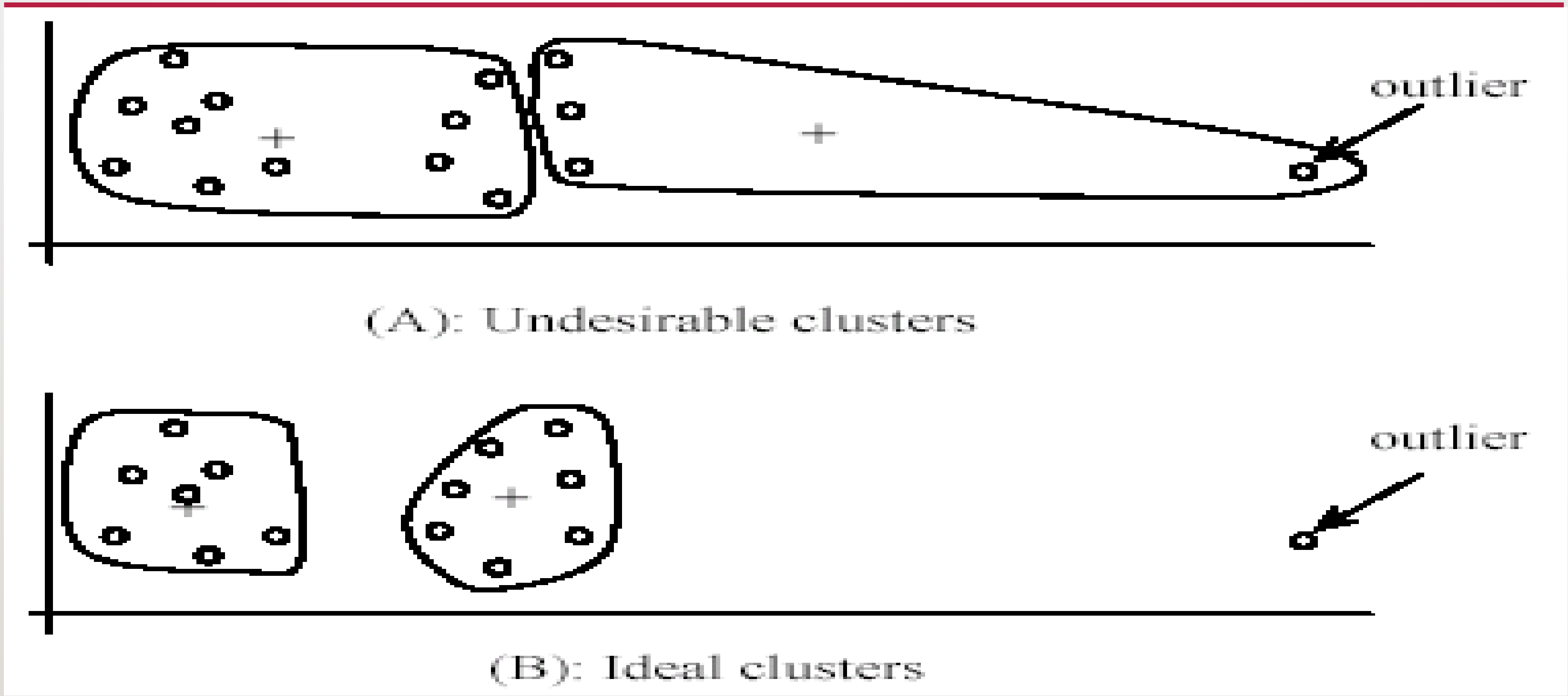
Weaknesses of k-means

- The algorithm is only applicable if the mean is defined.
  - For categorical data, *k*-mode - the centroid is represented by most frequent values.

- The user needs to specify *k*.

- The algorithm is sensitive to **outliers**
  - Outliers are data points that are very far away from other data points.
  - Outliers could be errors in the data recording or some special data points with very different values.

Weaknesses of k-means: Problems with outliers

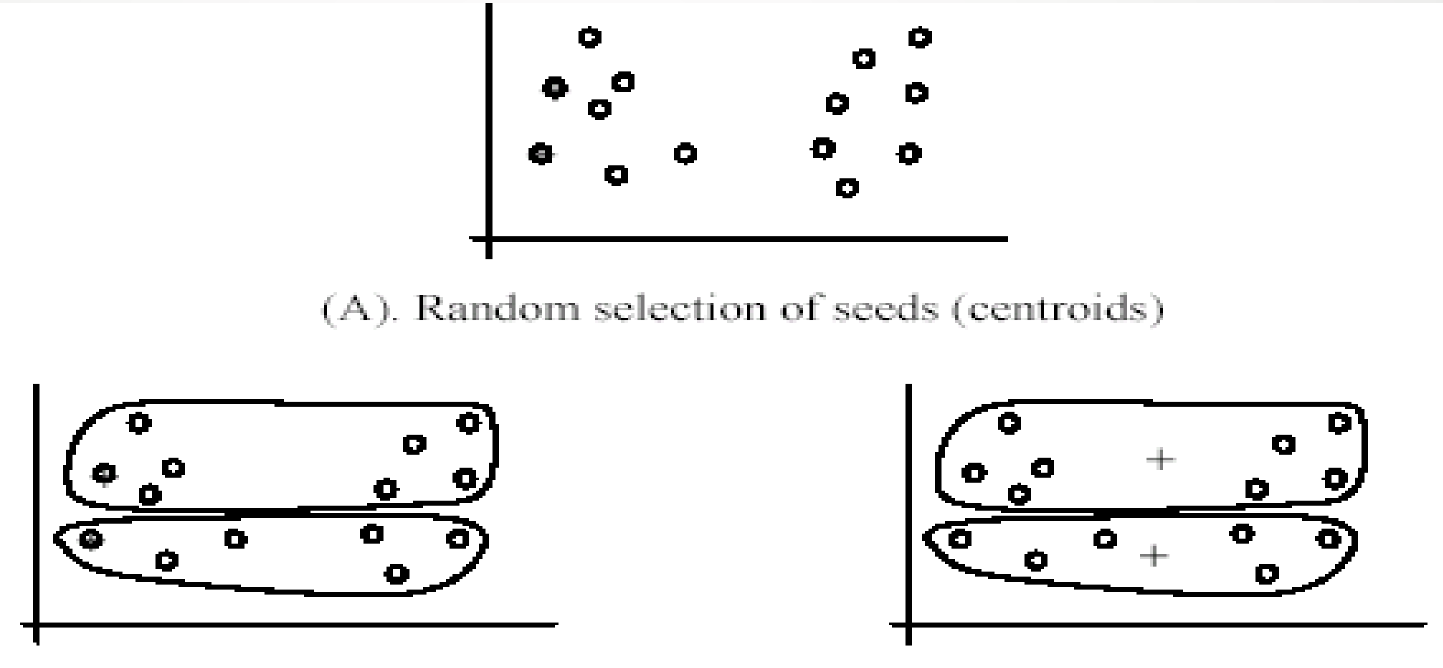


Weaknesses of k-means: To deal with outliers

- One method is to remove some data points in the clustering process that are much further away from the centroids than other data points.
  - To be safe, we may want to monitor these possible outliers over a few iterations and then decide to remove them.
- Another method is to perform random sampling. Since in sampling we only choose a small subset of the data points, the chance of selecting an outlier is very small.
  - Assign the rest of the data points to the clusters by distance or similarity comparison, or classification

Weaknesses of k-means (cont.…)

- The algorithm is sensitive to initial seeds.



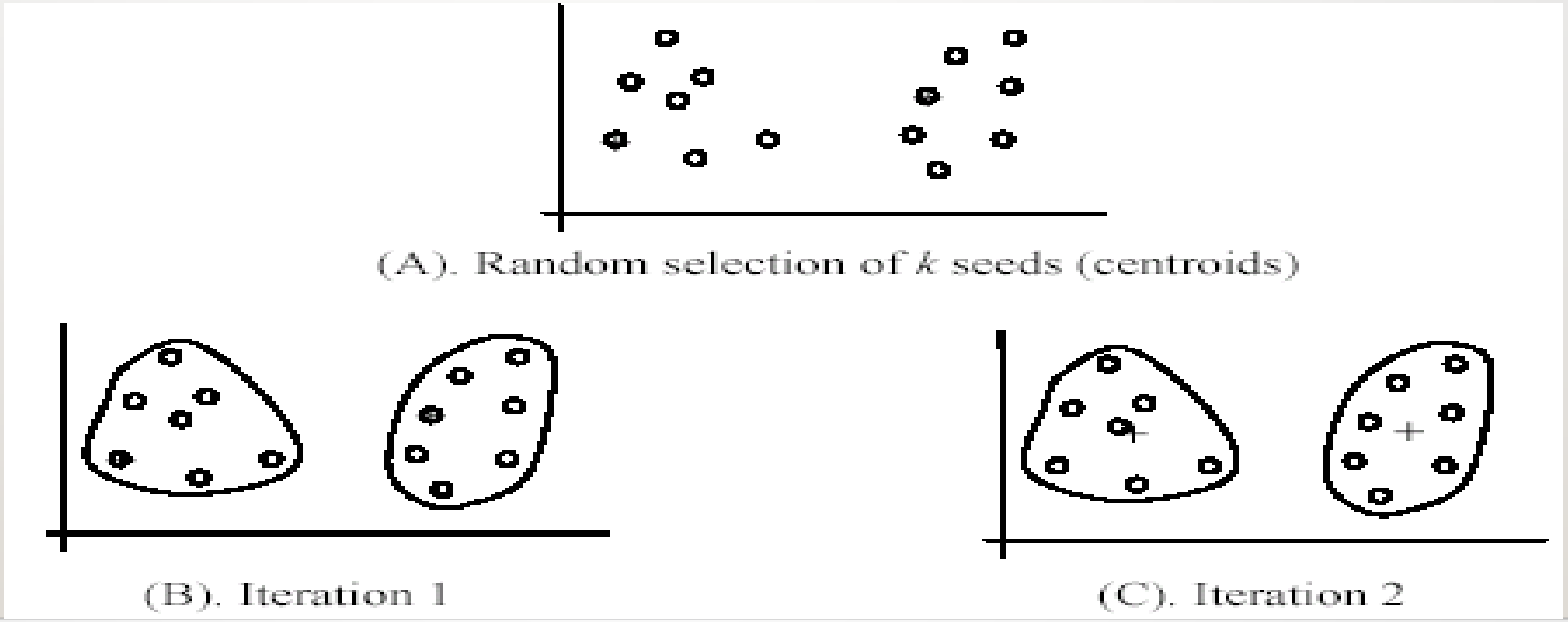
(A). Random selection of seeds (centroids)

(B). Iteration 1

(C). Iteration 2

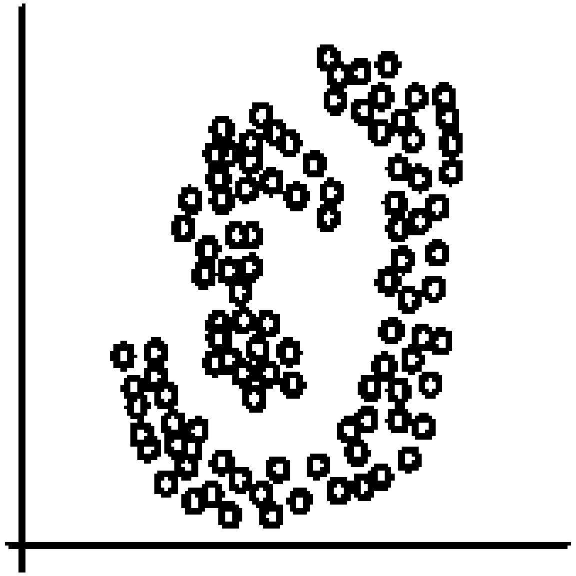
Weaknesses of k-means (cont….)

- If we use different seeds: good results
- There are some methods to help choose good seeds

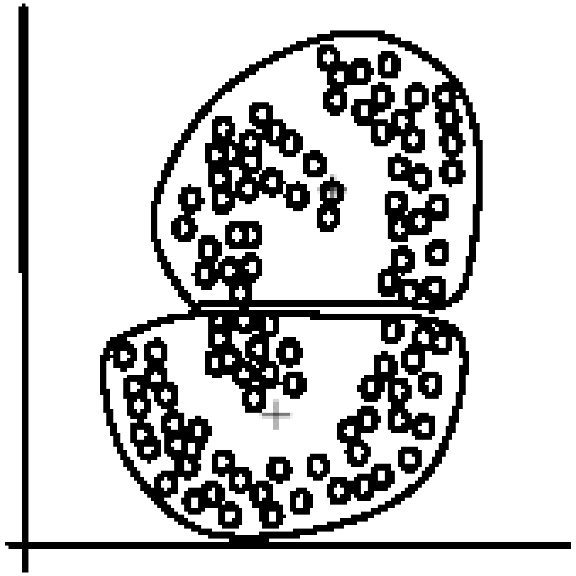


Weaknesses of k-means (cont.…)

- The *k*-means algorithm is not suitable for discovering clusters that are not hyper-ellipsoids (or hyper-spheres).



(A): Two natural clusters



(B): *k*-means clusters

Common ways to represent clusters

- Use the centroid of each cluster to represent the cluster.
  - compute the radius and
  - standard deviation of the cluster to determine its spread in each dimension
  - The centroid representation alone works well if the clusters are of the hyper-spherical shape.
  - If clusters are elongated or are of other shapes, centroids are not sufficient

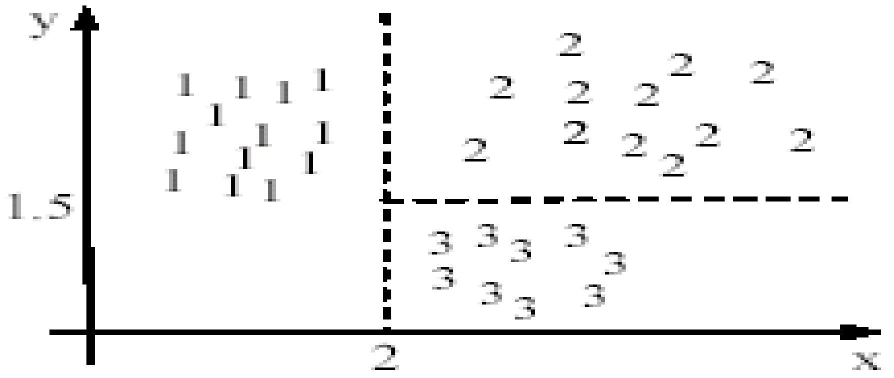
Using classification model

- All the data points in a cluster are regarded to have the same class label, e.g., the cluster ID.
  - run a supervised learning algorithm on the data to find a classification model.

$x \le 2 \rightarrow$ cluster 1

$x > 2,\ y > 1.5 \rightarrow$ cluster 2

$x > 2,\ y \le 1.5 \rightarrow$ cluster 3



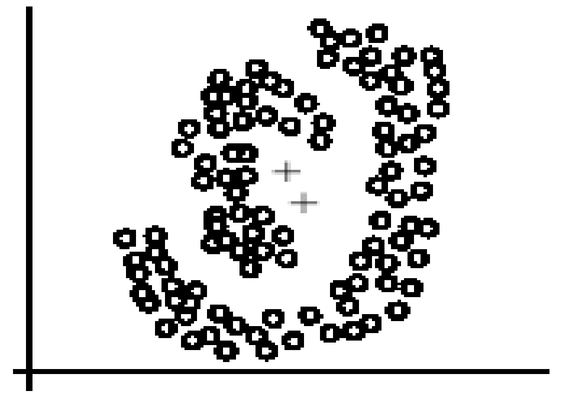
Use frequent values to represent cluster

- This method is mainly for clustering of categorical data (e.g., *k*-modes clustering).

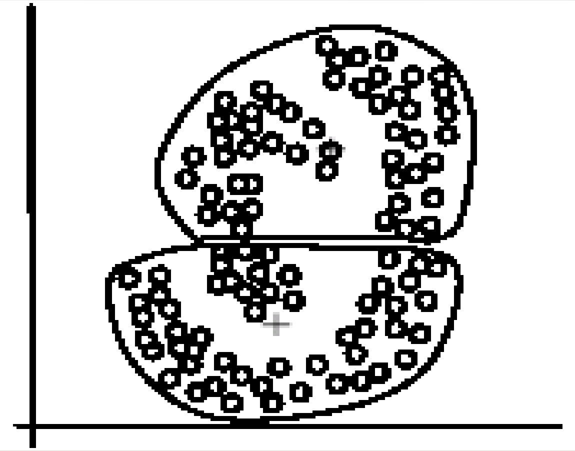
- Main method used in text clustering, where a small set of frequent words in each cluster is selected to represent the cluster.

Clusters of arbitrary shapes

- Hyper-elliptical and hyper-spherical clusters are usually easy to represent, using their centroid together with spreads.



- Irregular shape clusters are hard to represent. They may not be useful in some applications.
  - Using centroids are not suitable (upper figure) in general
  - K-means clusters may be more useful (lower figure), e.g., for making 2 size T-shirts.



Combining individual distances

- This approach computes individual attribute distances and then combine them.

$$dist(\mathbf{x}_i, \mathbf{x}_j) = \frac{\sum_{f=1}^{r} \delta_{ij}^{f} d_{ij}^{f}}{\sum_{f=1}^{r} \delta_{ij}^{f}}$$

This distance value is between 0 and 1. r is the number of attributes in the data set. The indicator δ_{ij}^{f} is 1 when both values x_{if} and x_{jf} for attribute f are non-missing, and it is set to 0 otherwise. It is also set to 0 if attribute f is asymmetric and the match is 0-0. Equation (25) cannot be computed if all δ_{ij}^{f} 's are 0. In such a case, some default value may be used or one of the data points is removed.

d_{ij}^{f} is the distance contributed by attribute f, and it is in the 0-1 range.

Summary

- Clustering is having along history and still active
  - There are a huge number of clustering algorithms
  - More are still coming every year.
  - We only introduced several main algorithms. There are many others, e.g.,
  - density based algorithm, sub-space clustering, scale-up methods, neural networks-based methods, fuzzy clustering, co-clustering, etc.
- Clustering is hard to evaluate, but very useful in practice. This partially explains why there are still many clustering algorithms being devised every year.
- Clustering is highly application dependent and to some extent subjective.

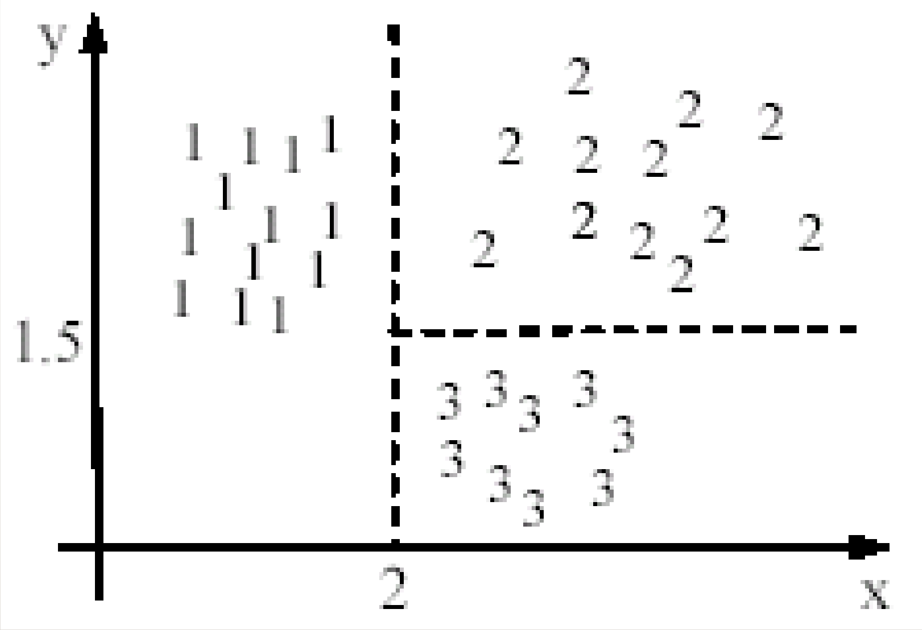
Common ways to represent clusters

- Use the centroid of each cluster to represent the cluster.
  - compute the radius and
  - standard deviation of the cluster to determine its spread in each dimension
  - The centroid representation alone works well if the clusters are of the hyper-spherical shape.
  - If clusters are elongated or are of other shapes, centroids are not sufficient

Using classification model

- All the data points in a cluster are regarded to have the same class label, e.g., the cluster ID.
  - run a supervised learning algorithm on the data to find a classification model.

$x \le 2 \rightarrow$ cluster 1
$x > 2,\ y > 1.5 \rightarrow$ cluster 2
$x > 2,\ y \le 1.5 \rightarrow$ cluster 3



THANK YOU

TEAM ML