Unsupervised Learning

Road map

- Basic concepts
- K-means algorithm
- Representation of clusters
- Distance functions
- Cluster evaluation

- Summary

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.

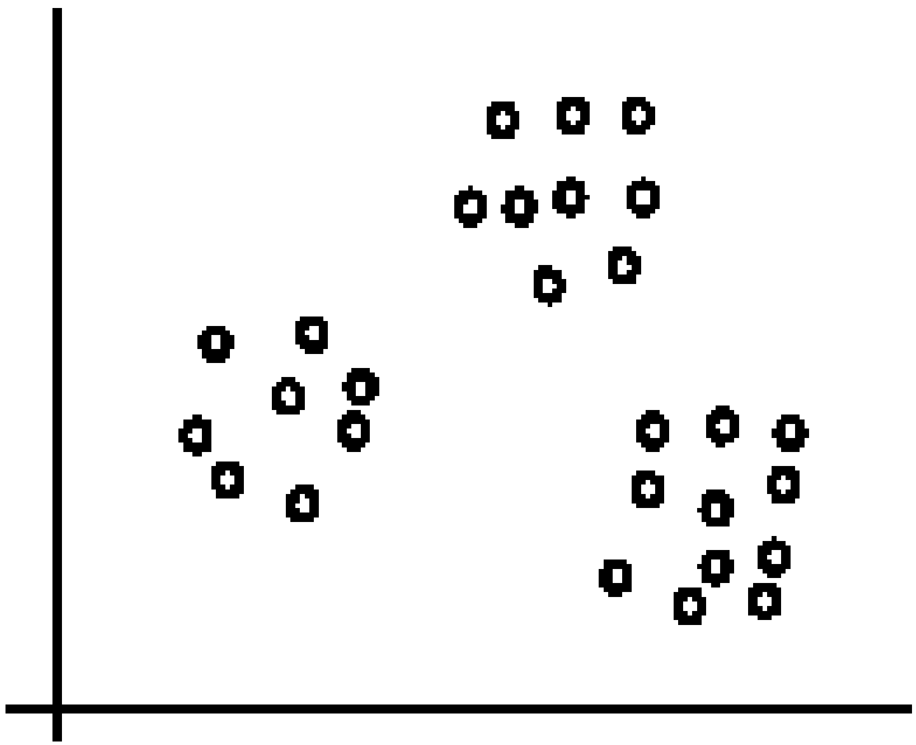
Due to historical reasons, clustering is often considered synonymous with unsupervised learning.

- In fact, association rule mining is also unsupervised

This chapter focuses on clustering.

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 data mining 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.

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.

Road map

- Basic concepts
- K-means algorithm
- Representation of clusters
- Distance functions
- Cluster evaluation
- Summary

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 …)

```
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
```

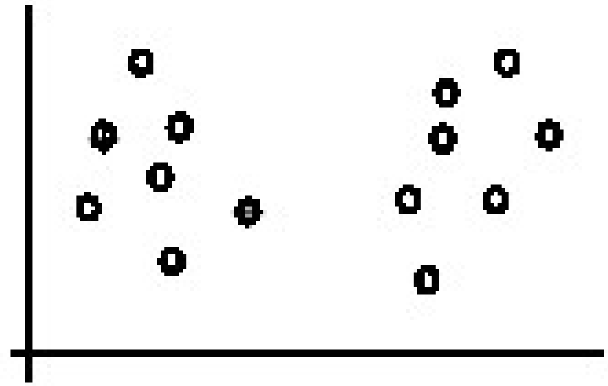
Stopping/convergence criterion

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

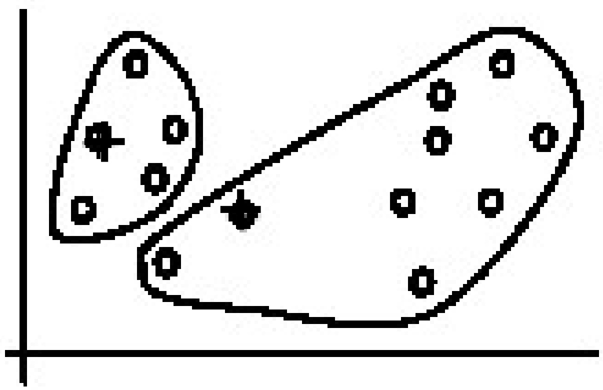
$$SSE = \sum_{j=1}^{k} \sum_{x \in C_j} dist(x, m_j)^2 \qquad (1)$$

- C_i is the jth 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 $\mathbf{m}_j$.

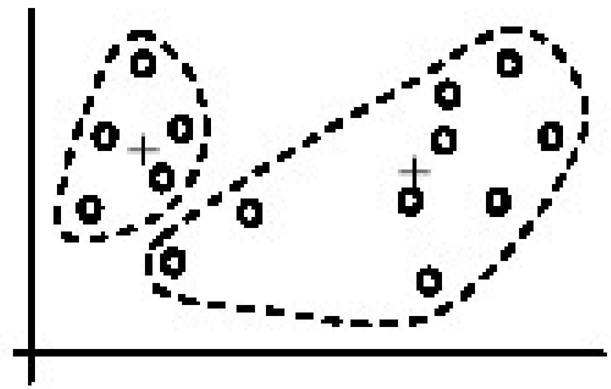
An example



(A). Random selection of *k* centers

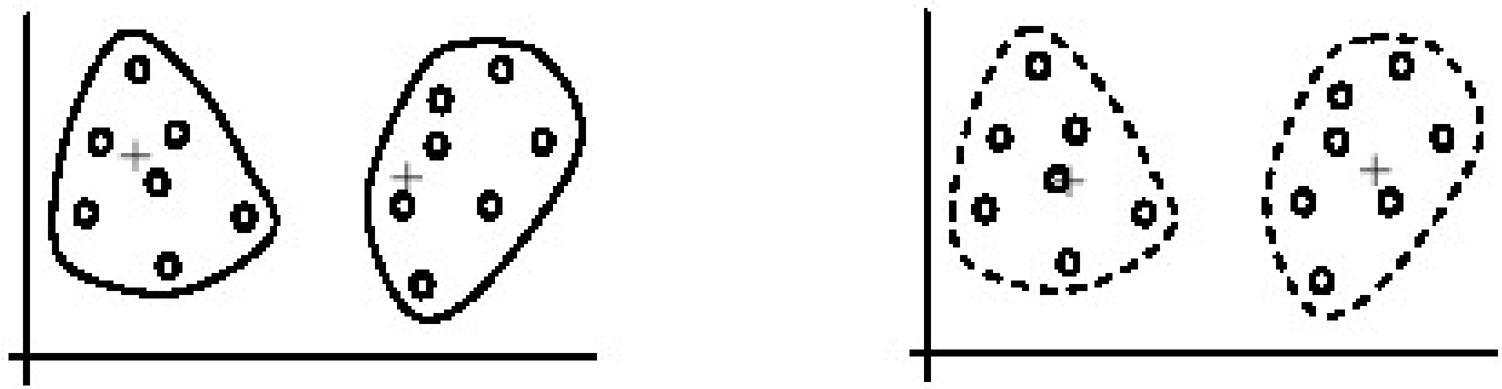


Iteration 1: (B). Cluster assignment

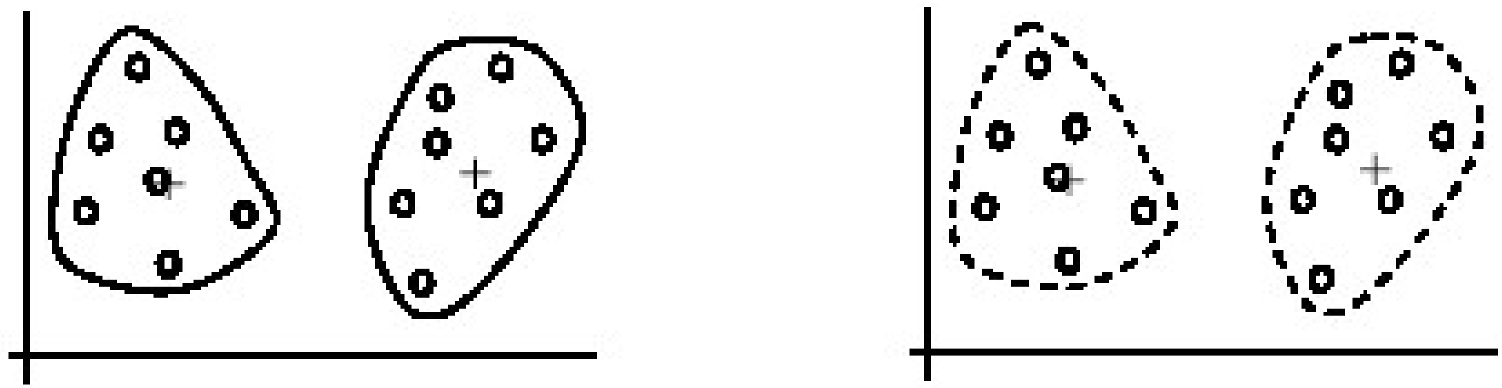


(C). Re-compute centroids

An example (cont …)



Iteration 2: (D). Cluster assignment   (E). Re-compute centroids



Iteration 3: (F). Cluster assignment   (G). Re-compute centroids

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

$$\begin{aligned} dist(\mathbf{x}_i, \mathbf{m}_j) &= \| \mathbf{x}_i - \mathbf{m}_j \| \\ &= \sqrt{(x_{i1} - m_{j1})^2 + (x_{i2} - m_{j2})^2 + \ldots + (x_{ir} - m_{jr})^2} \end{aligned} \qquad (3)$$

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

  $\mathbf{m}_i = \mathbf{s}_j / n_j$, $i = 1, \ldots, k$;

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.

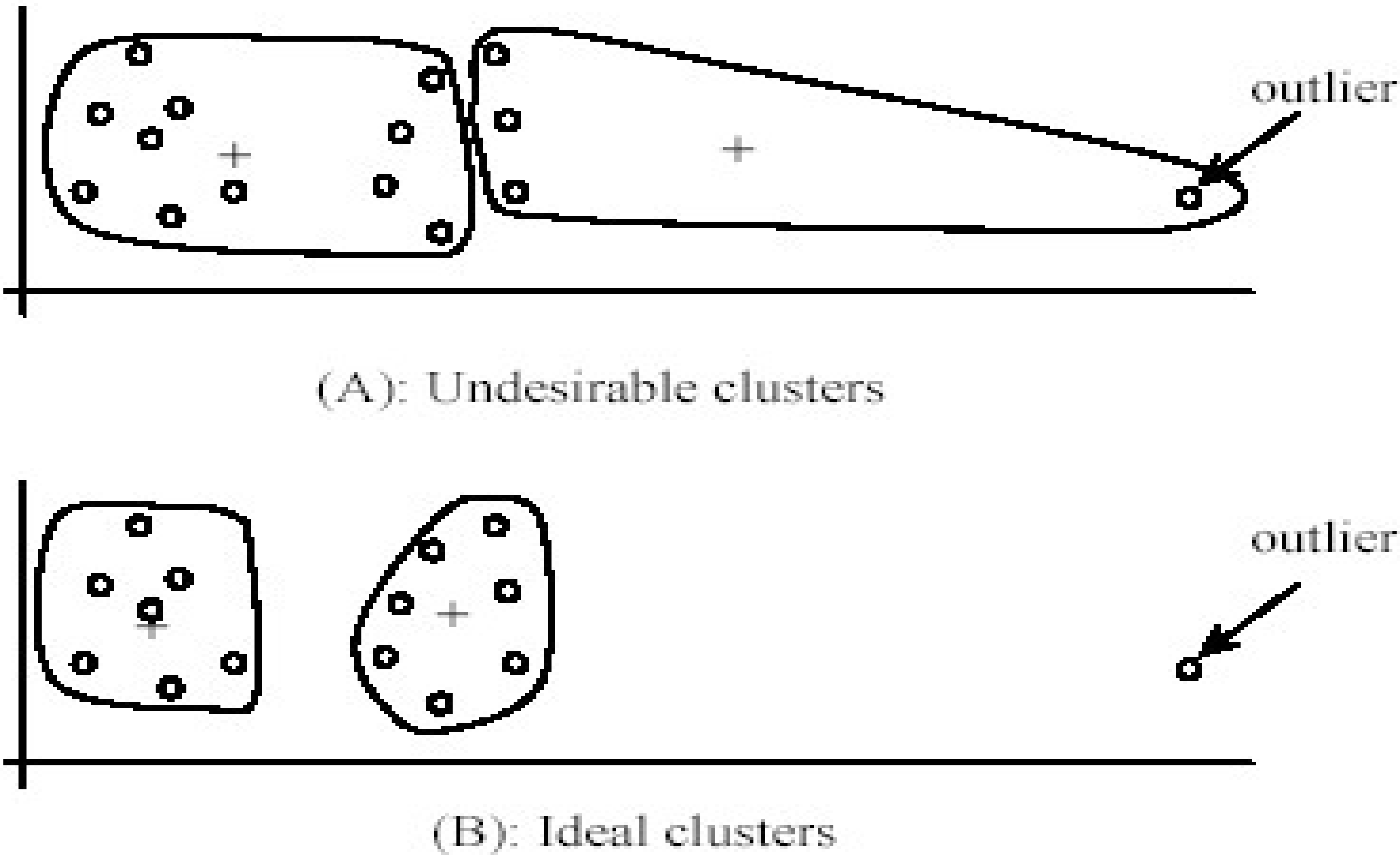
K-means is the most popular clustering algorithm.

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

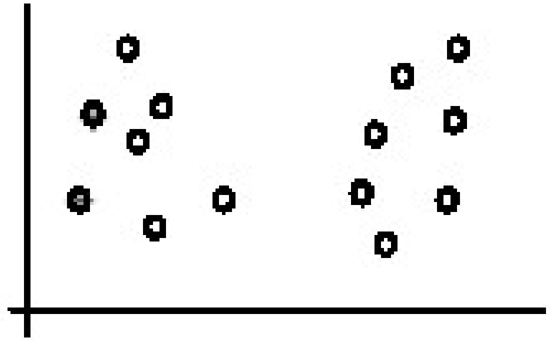


Weaknesses of k-means: To deal with outliers

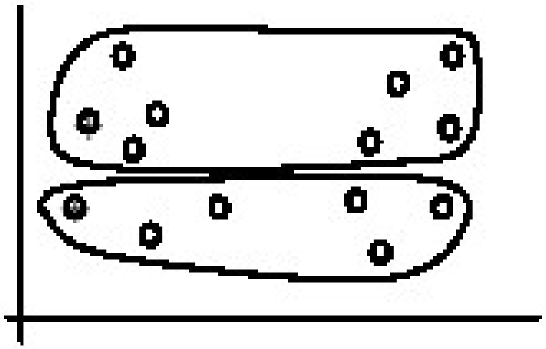
- One method is to remove some data points in the clustering process that are much further away from the centroids than other data points.
  - To be safe, we may want to monitor these possible outliers over a few iterations and then decide to remove them.
- Another method is to perform random sampling. 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 …)

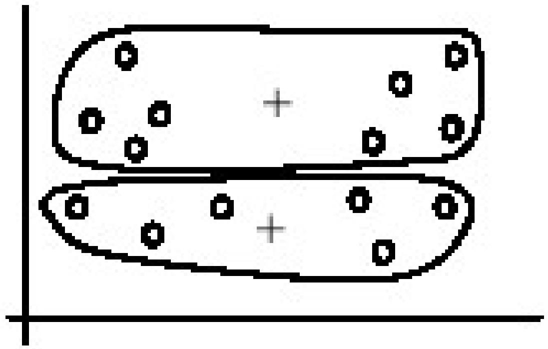
The algorithm is sensitive to initial seeds.



(A). Random selection of seeds (centroids)



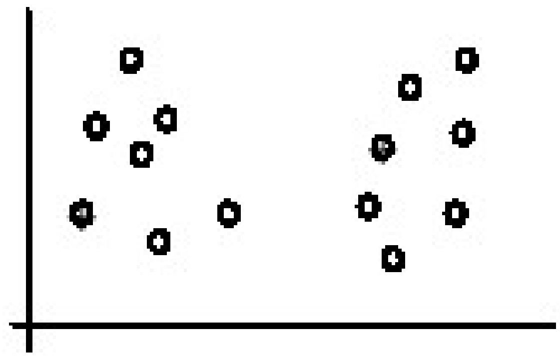
(B). Iteration 1



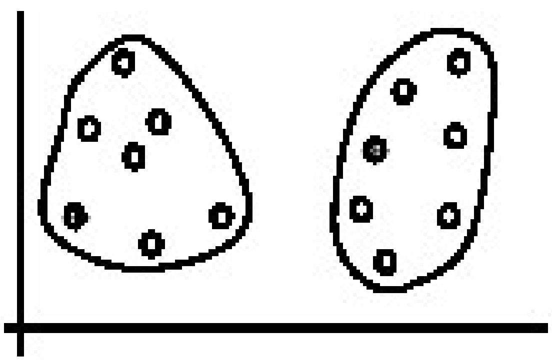
(C). Iteration 2

Weaknesses of k-means (cont …)

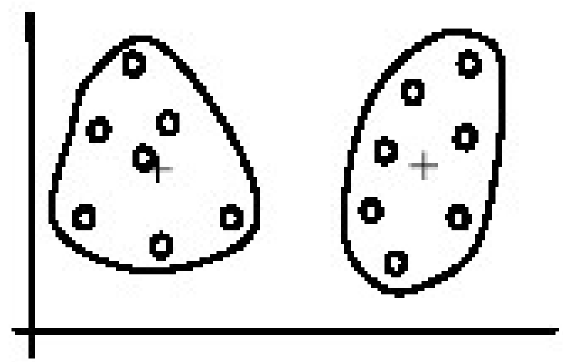
If we use different seeds: good results



(A). Random selection of *k* seeds (centroids)



(B). Iteration 1

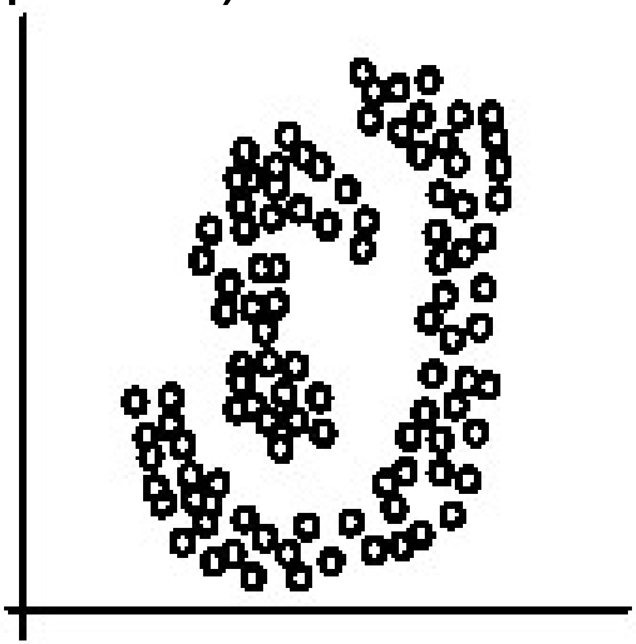


(C). Iteration 2

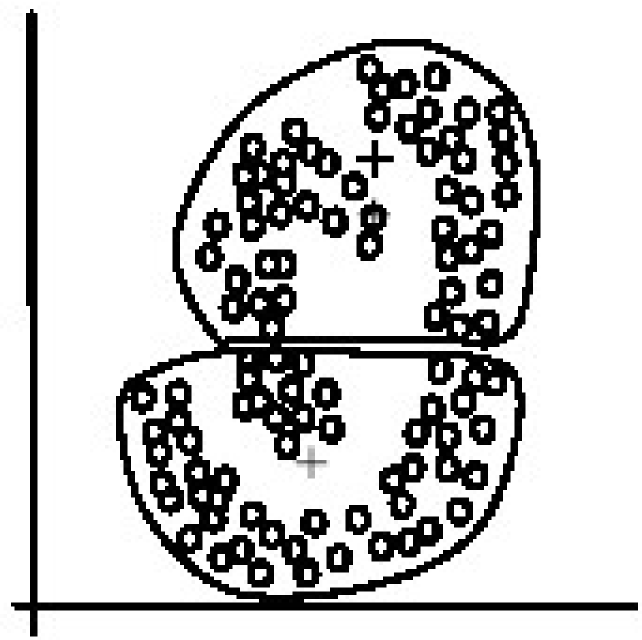
- There are some methods to help choose good seeds

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

K-means summary

Despite weaknesses, *k*-means is still the most popular algorithm due to its simplicity, efficiency and

other clustering algorithms have their own lists of weaknesses.

No clear evidence that any other clustering algorithm performs better in general

although they may be more suitable for some specific types of data or applications.

Comparing different clustering algorithms is a difficult task. No one knows the correct clusters!

Road map

- Basic concepts
- K-means algorithm
- Representation of clusters
- Distance functions
- Cluster evaluation
- Summary

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

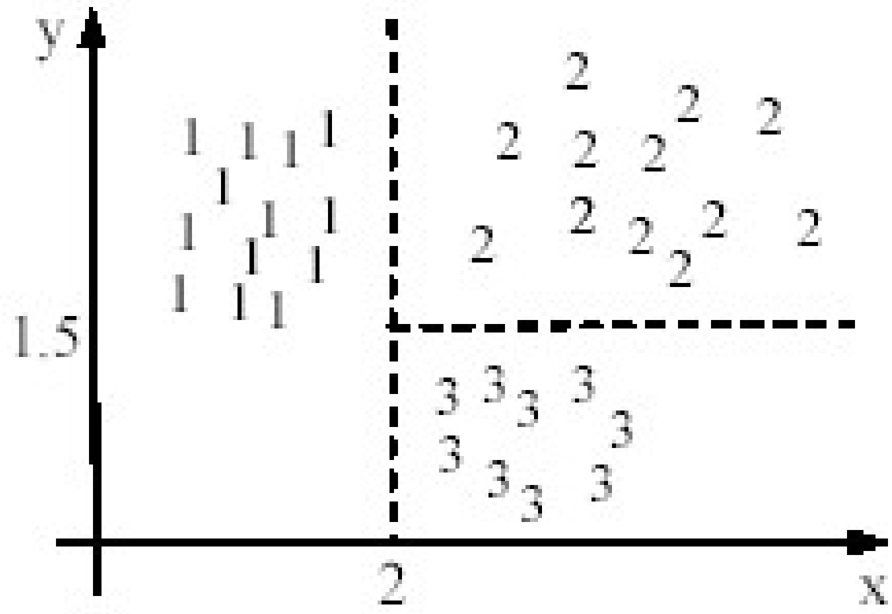
The centroid representation alone works well if the clusters are of the hyper-spherical shape.

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

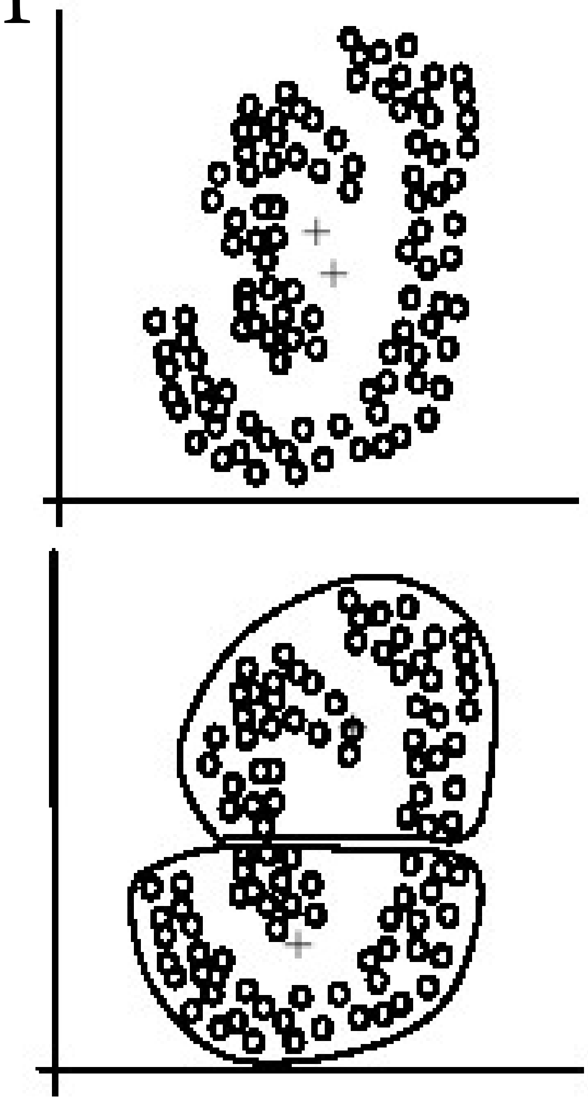
$x > 2,\ y \le 1.5 \rightarrow$ cluster 3

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.



Road map

- Basic concepts
- K-means algorithm
- Representation of clusters
- Distance functions
- Cluster evaluation
- Summary

Distance functions

Key to clustering. “similarity” and “dissimilarity” can also commonly used terms.

There are numerous distance functions for

Different types of data

- Numeric data
- Nominal data

Different specific applications

Distance functions for numeric attributes

Most commonly used functions are

Euclidean distance and

Manhattan (city block) distance

We denote distance with: *dist*($\mathbf{x}_i$, $\mathbf{x}_j$), where $\mathbf{x}_i$ and $\mathbf{x}_j$ are data points (vectors)

They are special cases of Minkowski distance. h is positive integer.

$$dist(x_i, x_j) = ((x_{i1} - x_{j1})^h + (x_{i2} - x_{j2})^h + \ldots + (x_{ir} - x_{jr})^h)^{\frac{1}{h}}$$

Road map

- Basic concepts
- K-means algorithm
- Representation of clusters
- Distance functions
- Cluster evaluation
- Summary

Cluster Evaluation: hard problem

The quality of a clustering is very hard to evaluate because

We do not know the correct clusters

Some methods are used:

User inspection

- Study centroids, and spreads
- Rules from a decision tree.
- For text documents, one can read some documents in clusters.

Cluster evaluation: ground truth

We use some labeled data (for classification)

Assumption: Each class is a cluster.

After clustering, a confusion matrix is constructed. From the matrix, we compute various measurements, entropy, purity, precision, recall and F-score.

Let the classes in the data D be $C = (c_1, c_2, \ldots, c_k)$. The clustering method produces k clusters, which divides D into k disjoint subsets, $D_1, D_2, \ldots, D_k$.

Evaluation measures: Entropy

Entropy: For each cluster, we can measure its entropy as follows:

$$entropy(D_i) = -\sum_{j=1}^{k} \Pr_i(c_j)\log_2 \Pr_i(c_j), \qquad (29)$$

where $\Pr_i(c_j)$ is the proportion of class c_j data points in cluster i or D_i. The total entropy of the whole clustering (which considers all clusters) is

$$entropy_{total}(D) = \sum_{i=1}^{k} \frac{|D_i|}{|D|} \times entropy(D_i) \qquad (30)$$

Evaluation measures: purity

Purity: This again measures the extent that a cluster contains only one class of data. The purity of each cluster is computed with

$$purity(D_i) = \max_j(\Pr_i(c_j)) \quad (31)$$

The total purity of the whole clustering (considering all clusters) is

$$purity_{total}(D) = \sum_{i=1}^{k} \frac{|D_i|}{|D|} \times purity(D_i) \quad (32)$$

An example

Example 14: Assume we have a text collection D of 900 documents from three topics (or three classes), Science, Sports, and Politics. Each class has 300 documents. Each document in D is labeled with one of the topics (classes). We use this collection to perform clustering to find three clusters. Note that class/topic labels are not used in clustering. After clustering, we want to measure the effectiveness of the clustering algorithm.

Cluster	Science	Sports	Politics		Entropy	Purity
1	250	20	10		0.589	0.893
2	20	180	80		1.198	0.643
3	30	100	210		1.257	0.617
Total	300	300	300		1.031	0.711

A remark about ground truth evaluation

- Commonly used to compare different clustering algorithms.
- A real-life data set for clustering has no class labels.
  - Thus although an algorithm may perform very well on some labeled data sets, no guarantee that it will perform well on the actual application data at hand.
- The fact that it performs well on some label data sets does give us some confidence of the quality of the algorithm.
- This evaluation method is said to be based on **external data** or information.

Evaluation based on internal information

Intra-cluster cohesion (compactness):

Cohesion measures how near the data points in a cluster are to the cluster centroid.

Sum of squared error (SSE) is a commonly used measure.

Inter-cluster separation (isolation):

Separation means that different cluster centroids should be far away from one another.

In most applications, expert judgments are still the key.

Indirect evaluation

- In some applications, clustering is not the primary task, but used to help perform another task.
- We can use the performance on the primary task to compare clustering methods.
- For instance, in an application, the primary task is to provide recommendations on book purchasing to online shoppers.
  - If we can cluster books according to their features, we might be able to provide better recommendations.
  - We can evaluate different clustering algorithms based on how well they help with the recommendation task.
  - Here, we assume that the recommendation can be reliably evaluated.

Road map

Basic concepts

K-means algorithm

Representation of clusters

Hierarchical clustering

Distance functions

Data standardization

Handling mixed attributes

Which clustering algorithm to use?

Cluster evaluation

Summary

Summary

Clustering is has 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 a large number of clustering algorithms being devised every year.

Clustering is highly application dependent and to some extent subjective.