ELEG 6913: Machine Learning for Big Data

Fall 2016

Lecture 10: Text Clustering

Dr. Xishuang Dong

Outline

- **Text Clustering**
- **Clustering Algorithms**
- **Summary**

(Acknowledgment: some parts of the slides are from Bing Liu, Chengxiang Zhai, and various other sources. The copyright of those parts belongs to their original owners.)

Outline

- **Text Clustering**
- **Clustering Algorithms**
- **Summary**

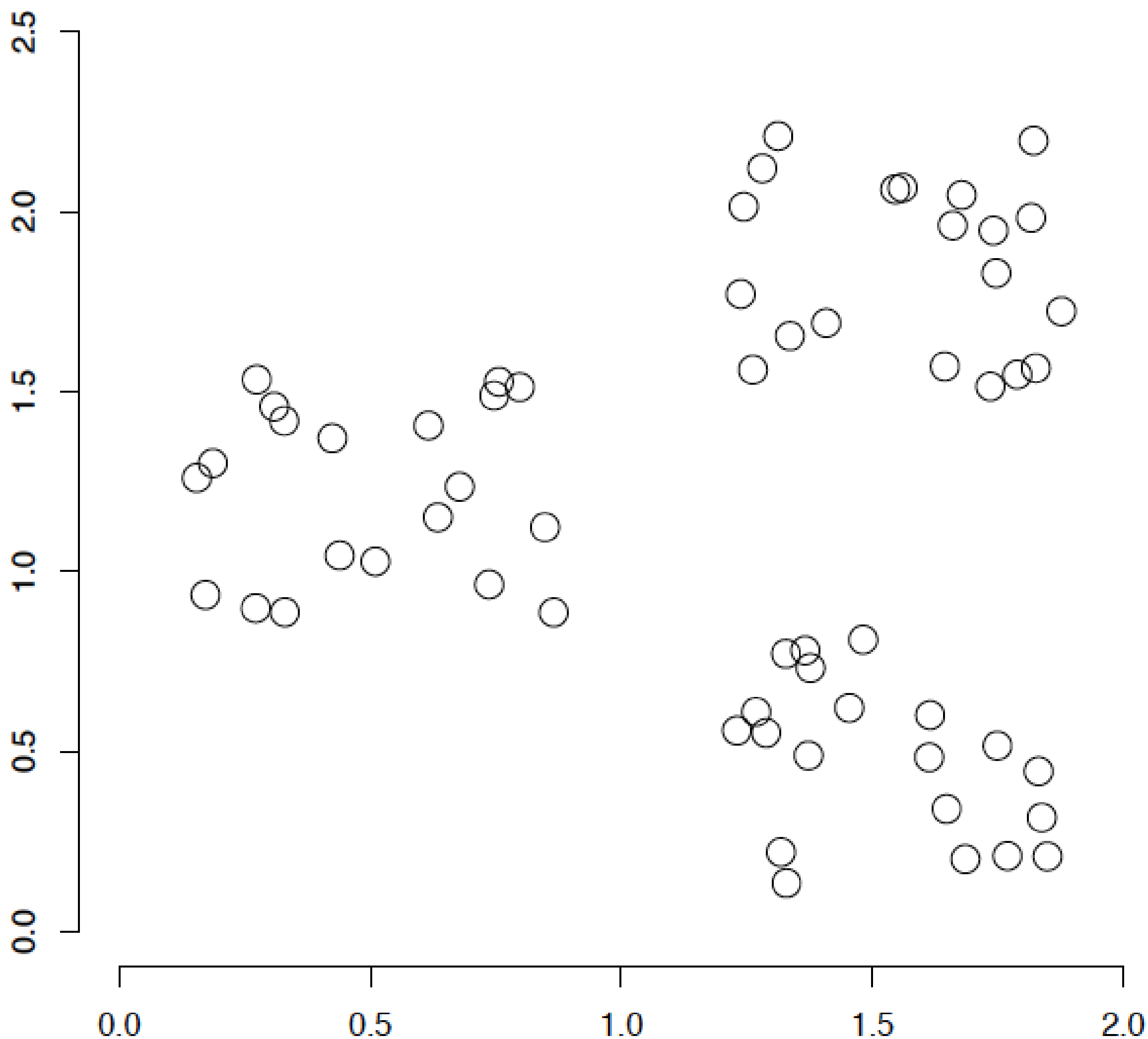
Supervised learning vs. Unsupervised learning

- **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

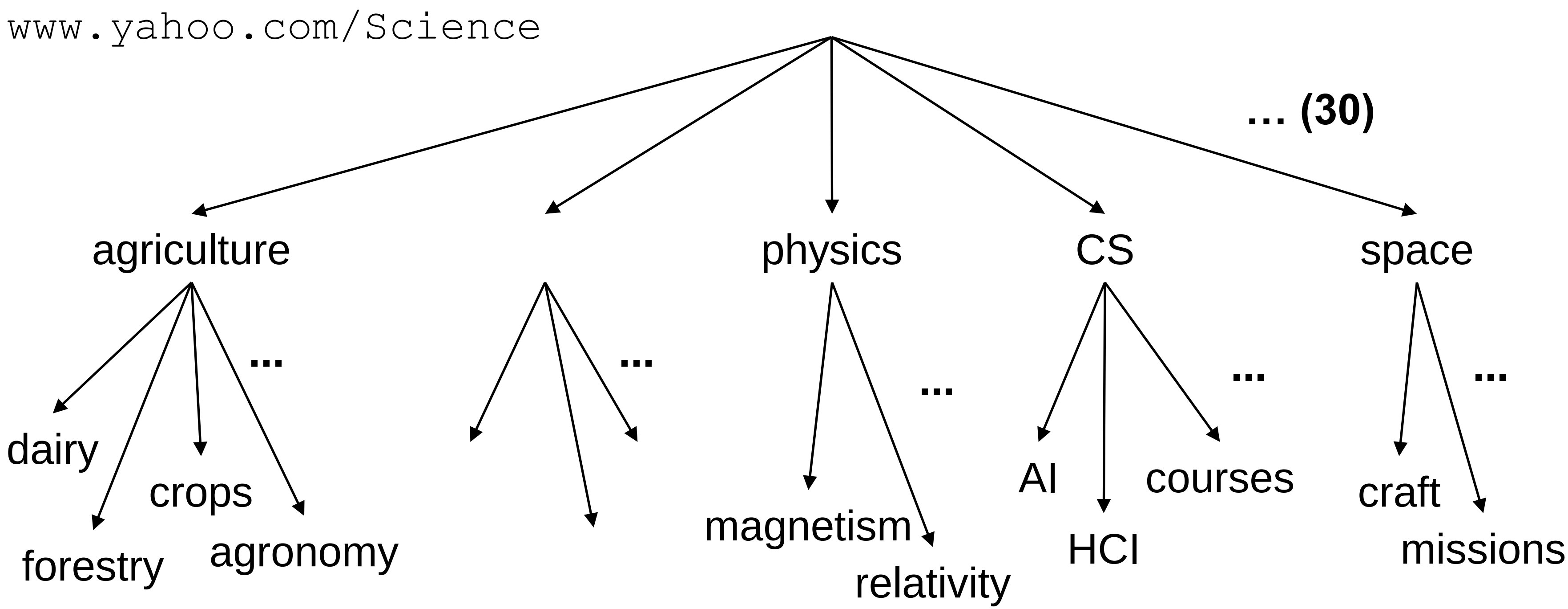
- **Clustering is a technique for finding similarity groups in data, called clusters. I.e.,**
  - **it groups data instances that are similar to (near) each other in one cluster and data instances that are very different (far away) from each other into different clusters.**
- **Clustering is often called 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**

A data set with clear cluster structure

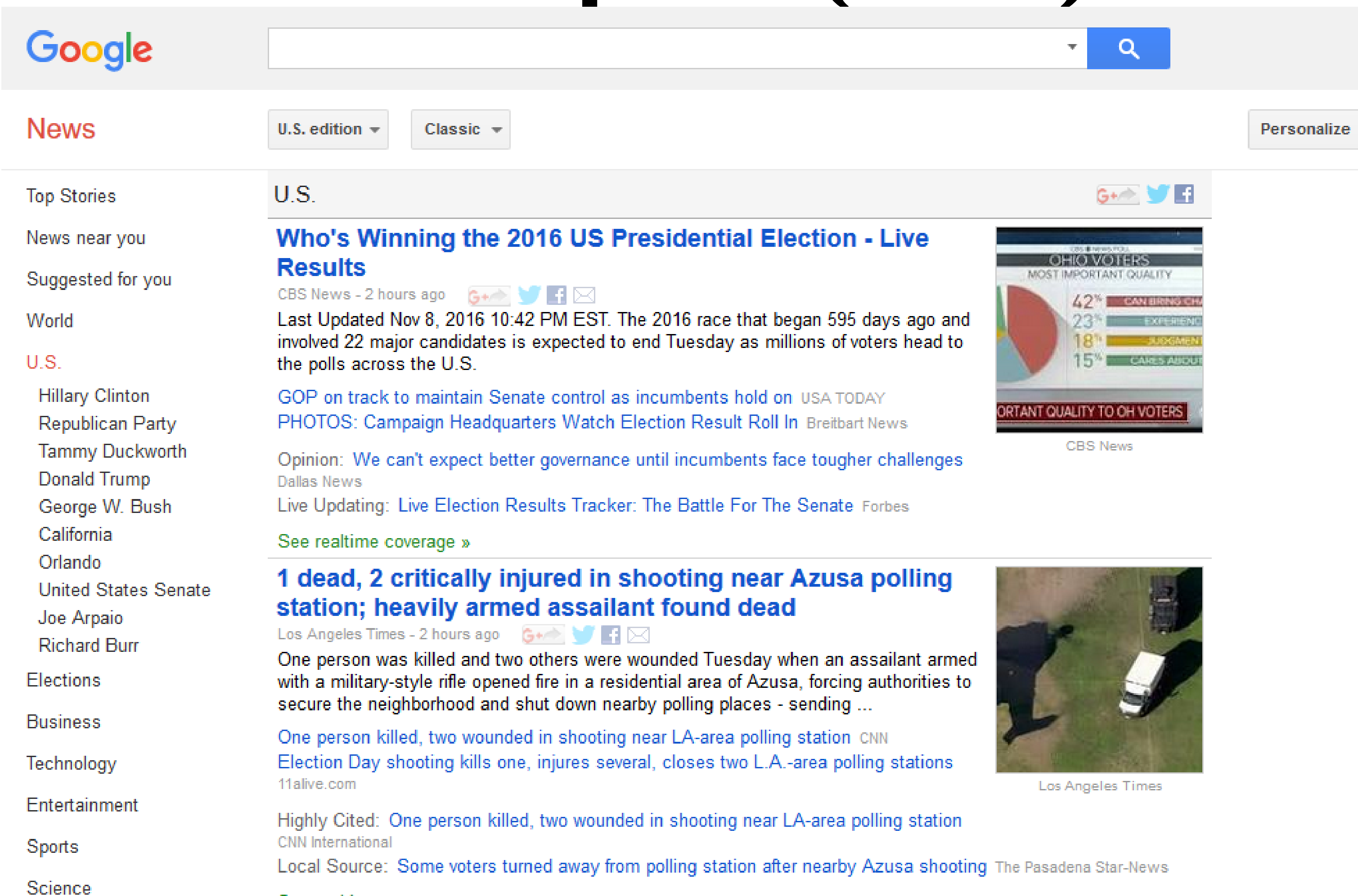


- **How would you design an algorithm for finding the three clusters in this case?**

Yahoo! Hierarchy isn't clustering but is the kind of output you want from clustering

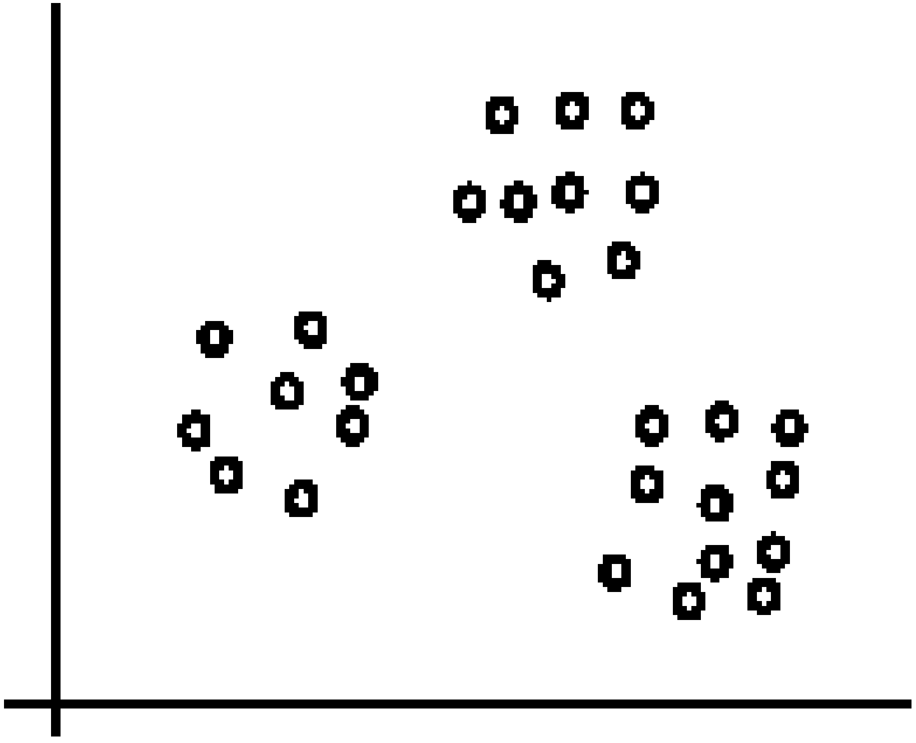


Google News: automatic clustering gives an effective news presentation metaphor (Now)



An illustration

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



What is clustering for?

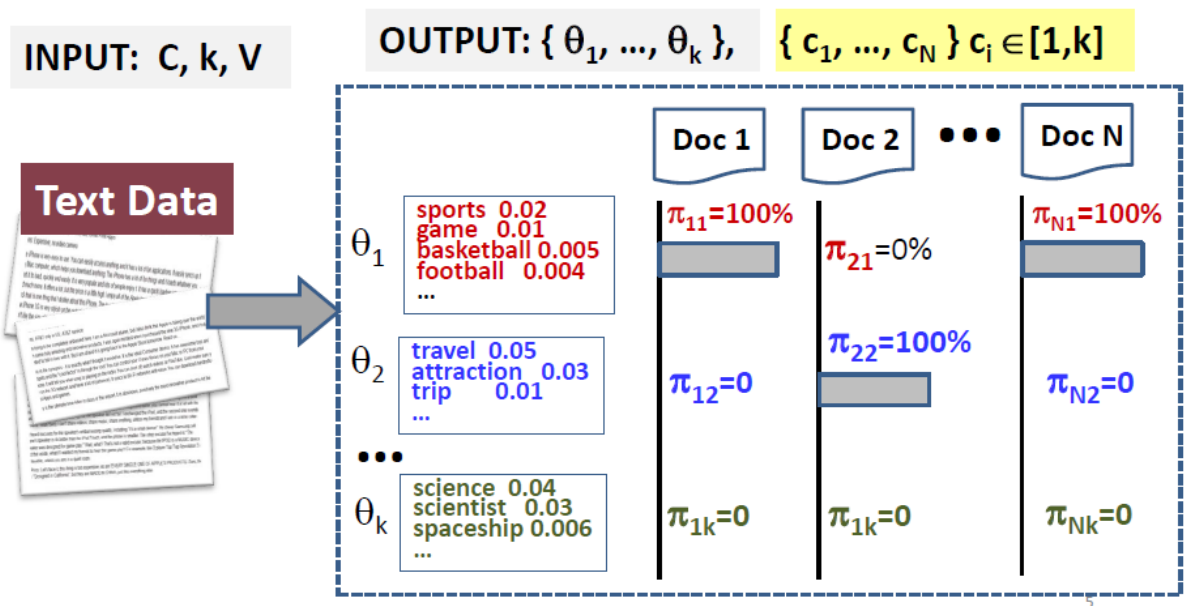
- **Let us see some real-life examples**
- **Example 1: groups people of similar sizes together to make "small", "medium" and "large" T-Shirts.**
- **Example 2: In marketing, segment customers according to their similarities**
  - **To do targeted marketing.**

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.**

Text Clustering

- **Text clustering** most often separates the entire corpus of documents into **mutually exclusive clusters** – each document belongs to one and only one cluster (i.e., *hard clustering*)



whereas **Topic Extraction** assigns a document to multiple topics (i.e., *soft clustering*).

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.**

Outline

- Text Clustering
- **Clustering Algorithms**
- 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
```

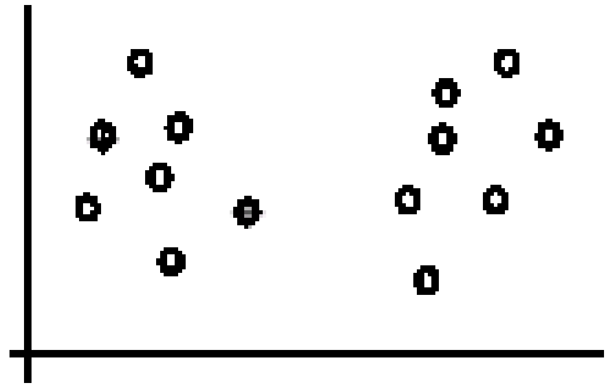
Stopping/convergence criterion

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)**

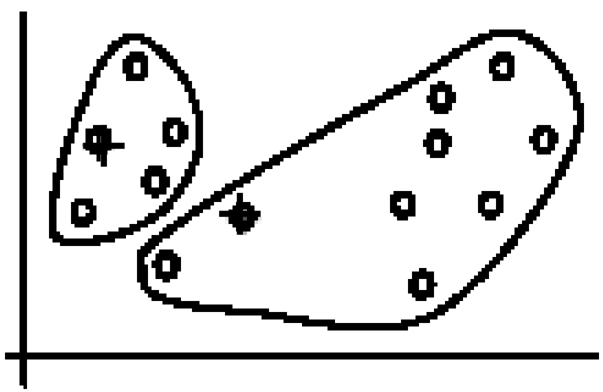
$$SSE = \sum_{j=1}^{k} \sum_{\mathbf{x} \in C_j} dist(\mathbf{x}, \mathbf{m}_j)^2 \qquad \textbf{(1)}$$

C_j 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 $\mathbf{m}_j$.

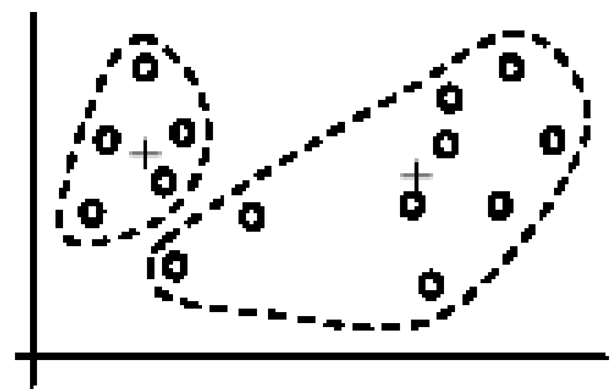
An example



(A). Random selection of *k* centers

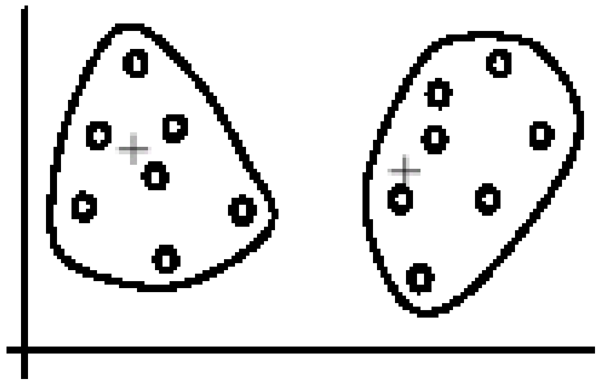


Iteration 1: (B). Cluster assignment

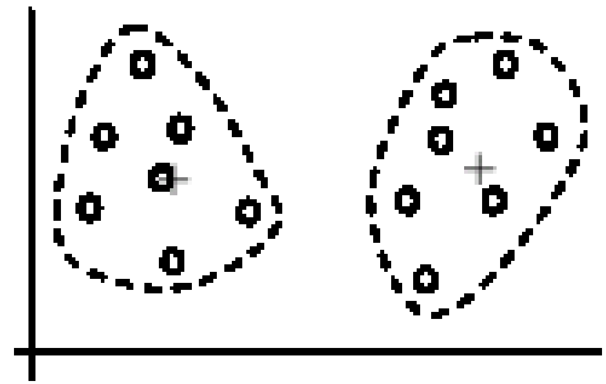


(C). Re-compute centroids

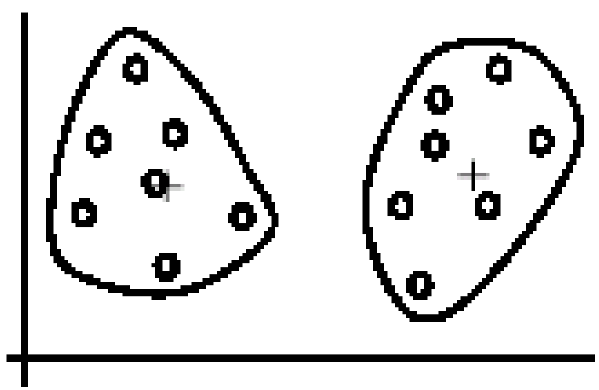
An example (cont …)



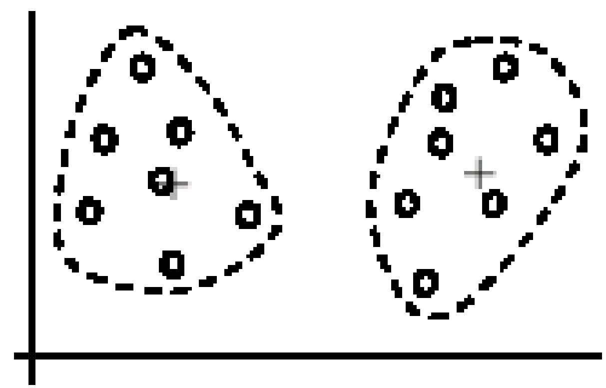
Iteration 2: (D). Cluster assignment



(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

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

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

A disk version of *k*-means

- K-means can be implemented with data on disk
  - In each iteration, it scans the data once.
  - 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,

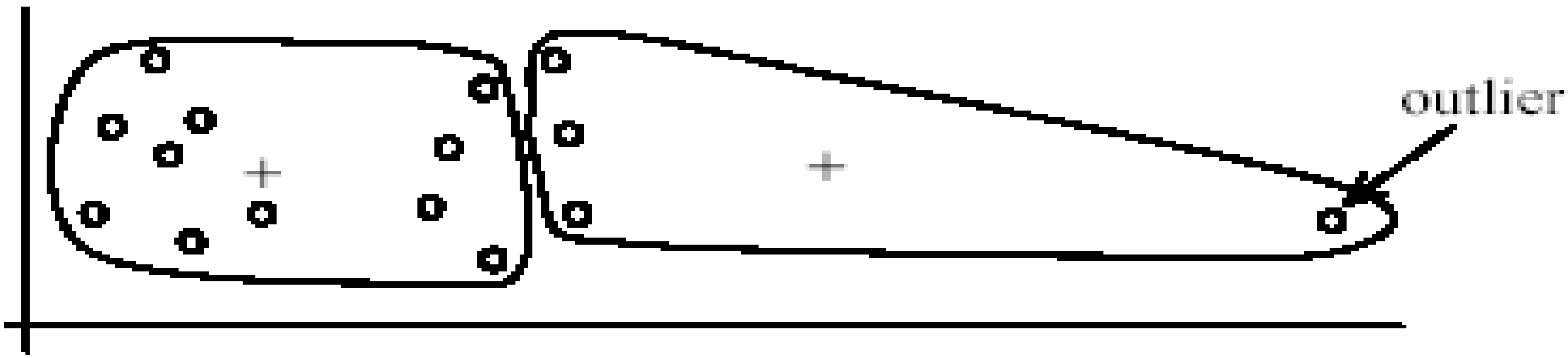
    k is the number of clusters, and

    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.**

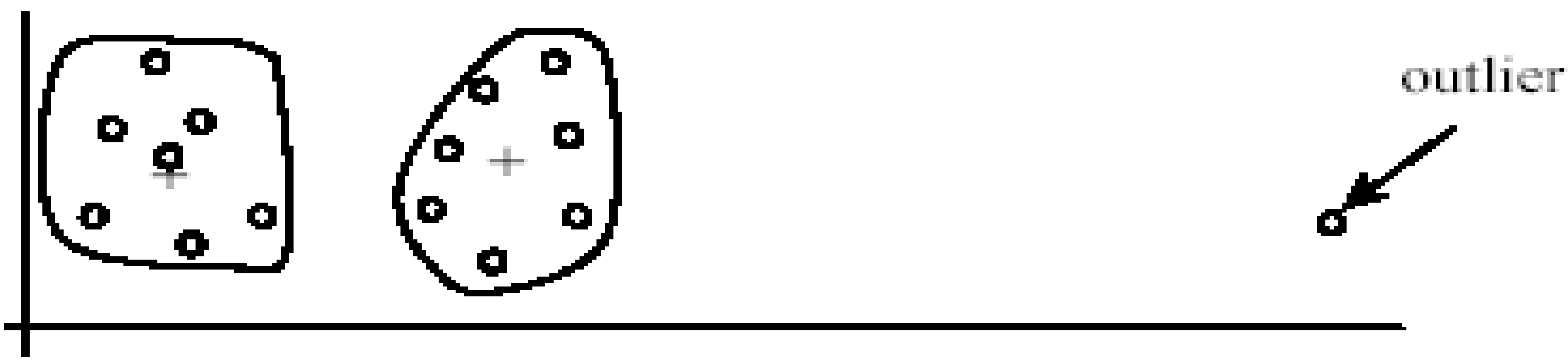
Weaknesses of k-means

- **The algorithm is only applicable if the mean is defined.**
- **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



(A): Undesirable clusters



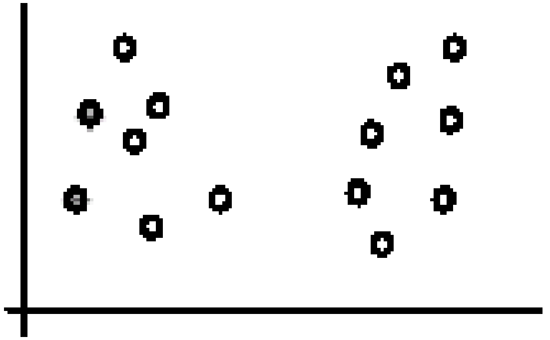
(B): Ideal clusters

Weaknesses of k-means: To deal with outliers

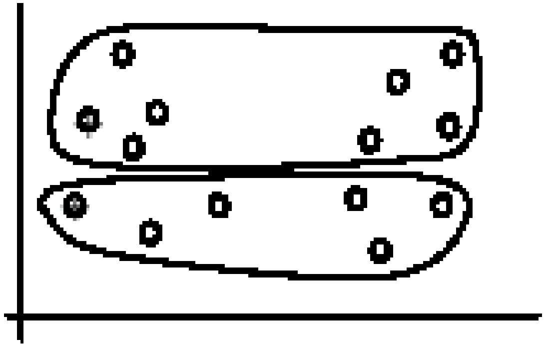
- **One method is to remove some data points in the clustering process that are much further away from the centroids than other data points.**
  - **To be safe, we may want to monitor these possible outliers over a few iterations and then decide to remove them.**

Weaknesses of k-means (cont …)

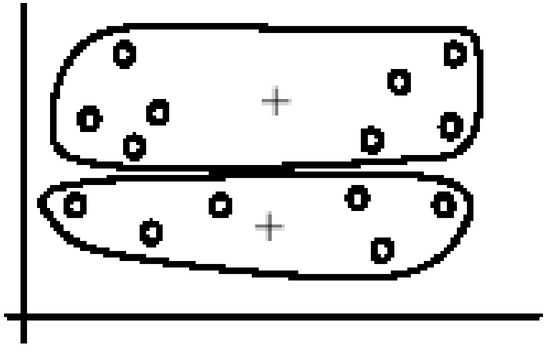
- **The algorithm is sensitive to initial seeds.**



(A). Random selection of seeds (centroids)



(B). Iteration 1

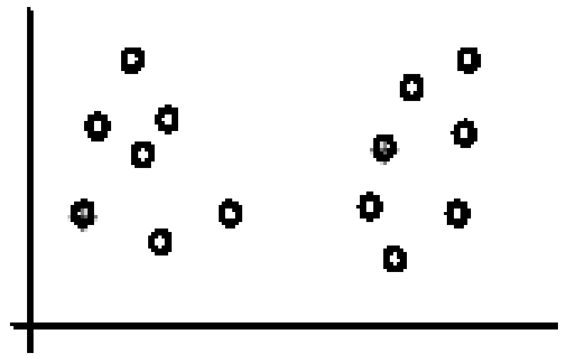


(C). Iteration 2

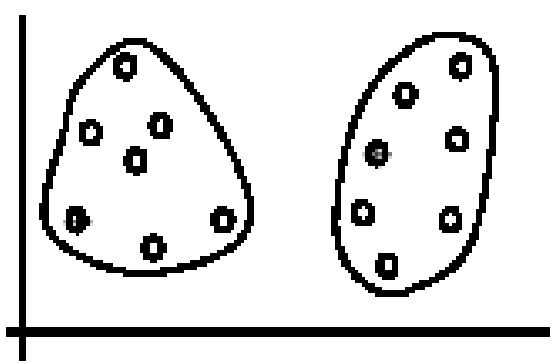
Weaknesses of k-means (cont …)

- **If we use different seeds: good results**

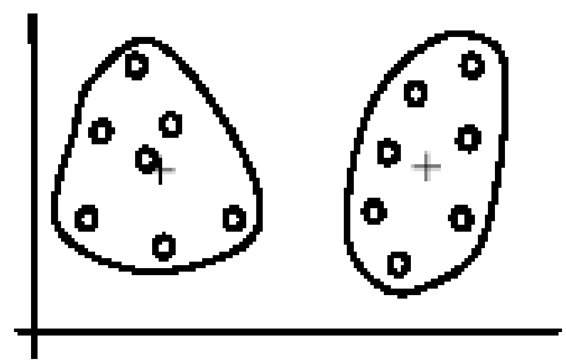
There are some methods to help choose good seeds



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



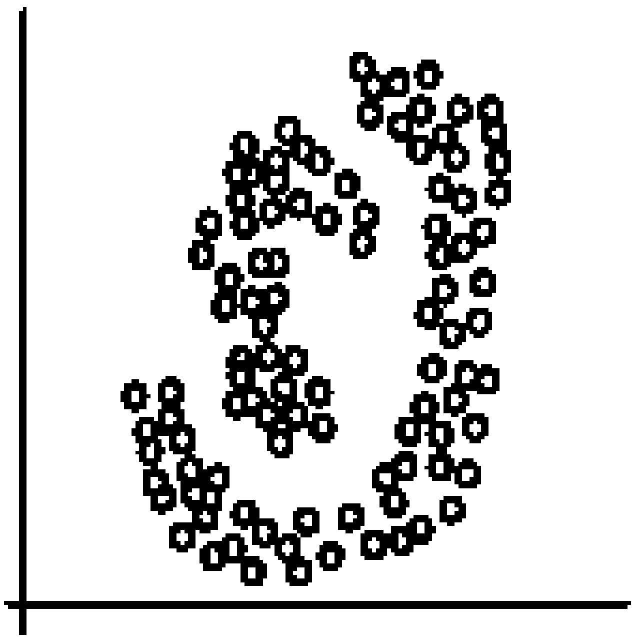
(B). Iteration 1



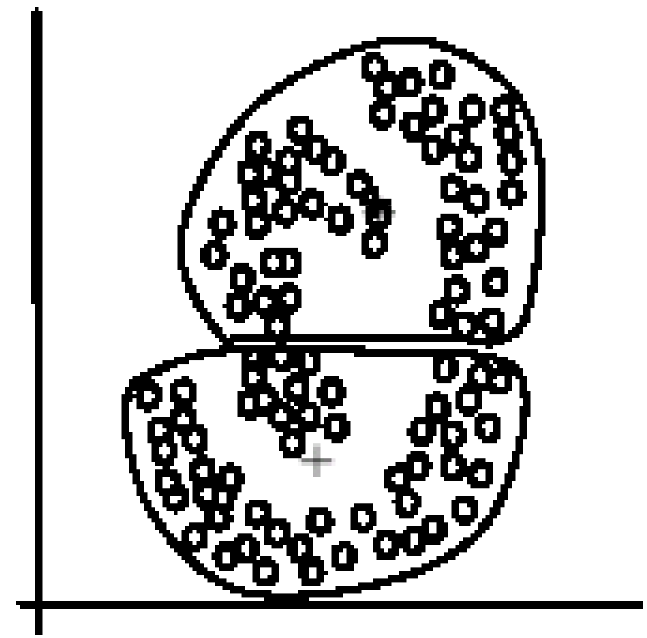
(C). Iteration 2

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!**

Common ways to represent clusters

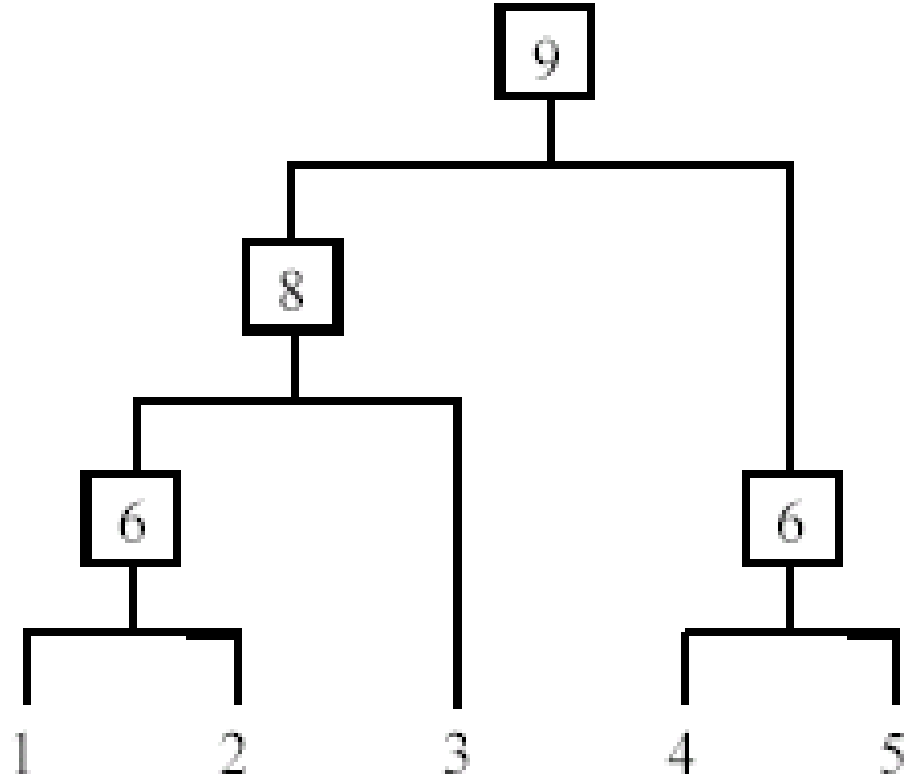
- **Use the centroid of each cluster to represent the cluster.**
  - **compute the radius and**
  - **standard deviation of the cluster to determine its spread in each dimension**
  - **The centroid representation alone works well if the clusters are of the hyper-spherical shape.**
  - **If clusters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

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.**

Hierarchical Clustering

- **Produce a nested sequence of clusters, a tree, also called Dendrogram.**



Types of hierarchical clustering

- **Agglomerative (bottom up) clustering: It builds the dendrogram (tree) from the bottom level, and**
  - **merges the most similar (or nearest) pair of clusters**
  - **stops when all the data points are merged into a single cluster (i.e., the root cluster).**
- **Divisive (top down) clustering: It starts with all data points in one cluster, the root.**
  - **Splits the root into a set of child clusters. Each child cluster is recursively divided further**
  - **Stops when only singleton clusters of individual data points remain, i.e., each cluster with only a single point**

Agglomerative clustering

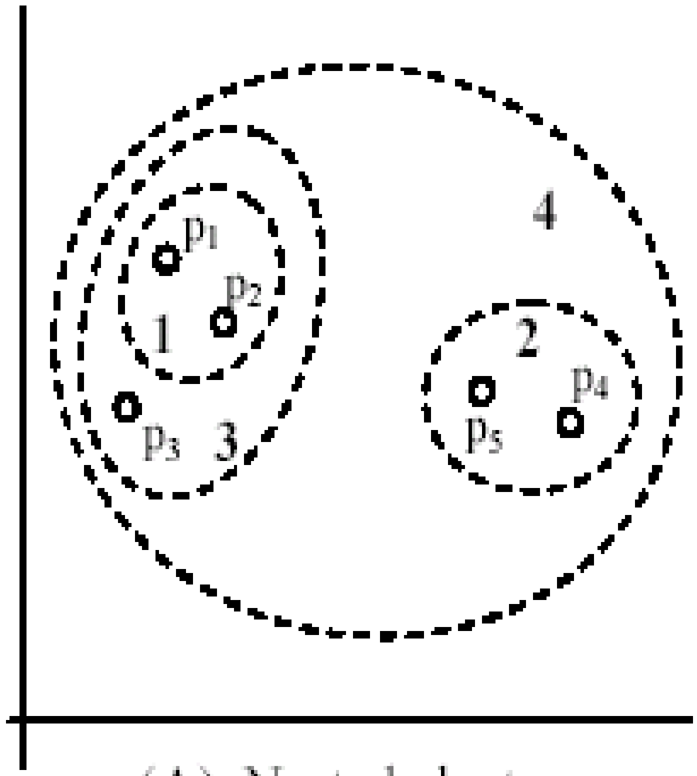
It is more popular then divisive methods.

- **At the beginning, each data point forms a cluster (also called a node).**
- **Merge nodes/clusters that have the least distance.**
- **Go on merging**
- **Eventually all nodes belong to one cluster**

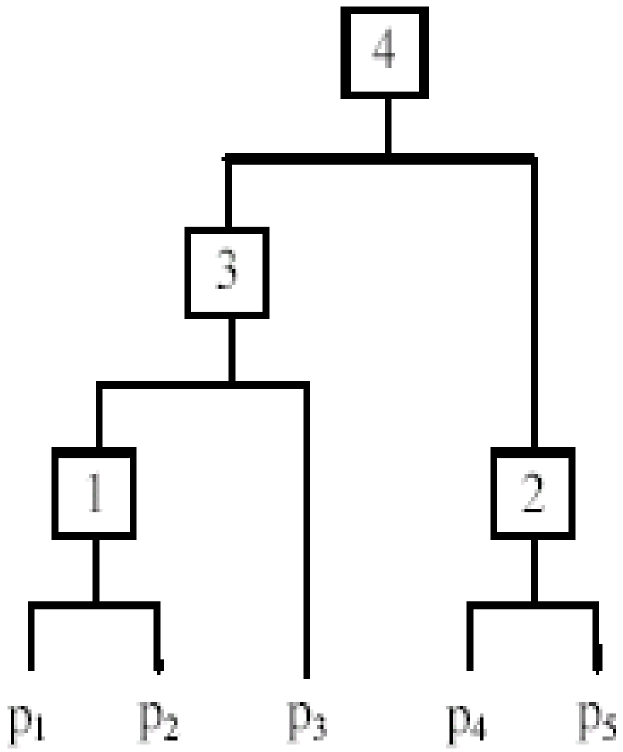
Agglomerative clustering algorithm

```
Algorithm Agglomerative(D)
    Make each data point in the data set D a cluster,
    Compute all pair-wise distances of x_1, x_2, …, x_n ∈ D;
    repeat
        find two clusters that are nearest to each other;
        merge the two clusters form a new cluster c;
        compute the distance from c to all other clusters;
    until there is only one cluster left
```

An example: working of the algorithm



(A). Nested clusters



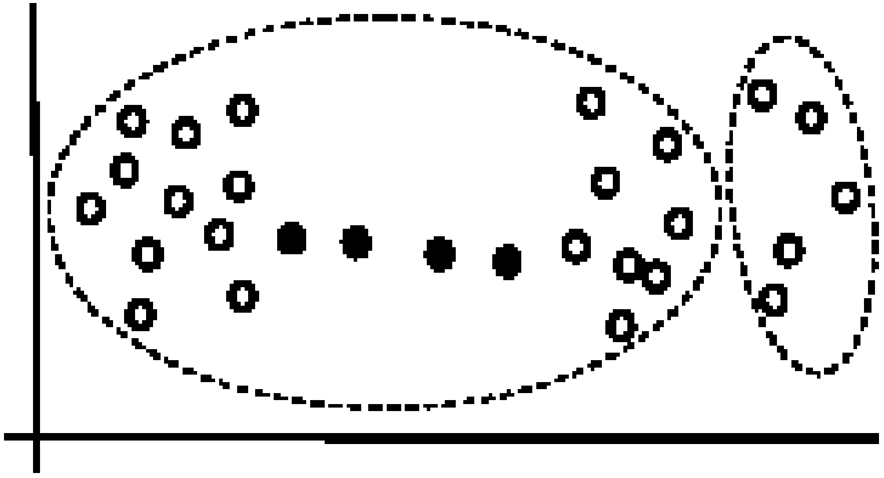
(B) Dendrogram

Measuring the distance of two clusters

- **A few ways to measure distances of two clusters.**
- **Results in different variations of the algorithm.**
  - **Single link**
  - **Complete link**
  - **Average link**
  - **Centroids**
  - **…**

Single link method

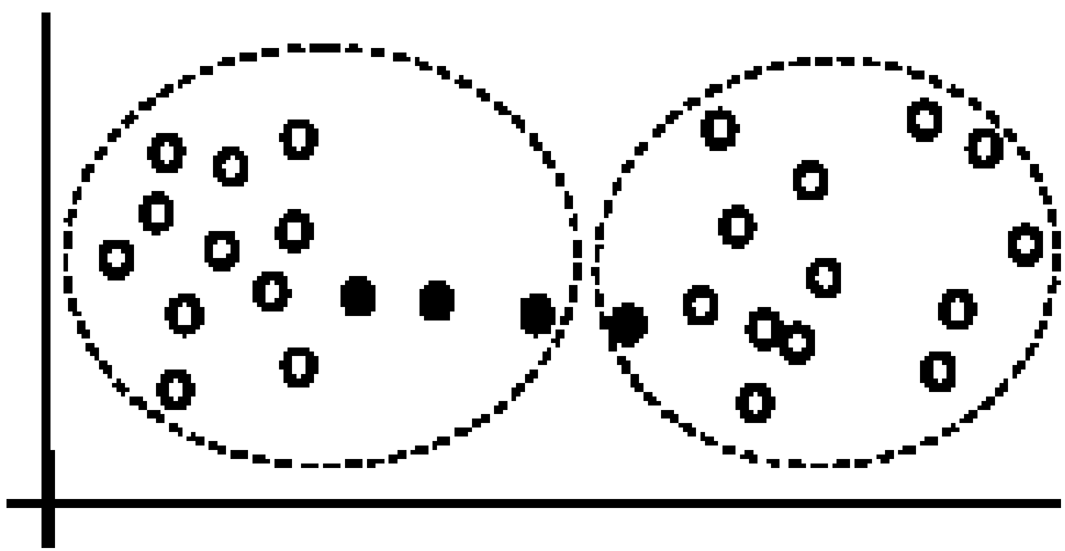
- The distance between two clusters is the distance between two closest data points in the two clusters, one data point from each cluster.
- It can find arbitrarily shaped clusters, but
  - It may cause the undesirable "chain effect" by noisy points



Two natural clusters are split into two

Complete link method

- The distance between two clusters is the distance of two furthest data points in the two clusters.
- It is sensitive to outliers because they are far away



Average link and centroid methods

- **Average link: A compromise between**
  - **the sensitivity of complete-link clustering to outliers and**
  - **the tendency of single-link clustering to form long chains that do not correspond to the intuitive notion of clusters as compact, spherical objects.**
  - **In this method, the distance between two clusters is the average distance of all pair-wise distances between the data points in two clusters.**
- **Centroid method: In this method, the distance between two clusters is the distance between their centroids**

The complexity

- All the algorithms are at least $O(n^2)$. n is the number of data points.
- Due the complexity, hard to use for large data sets.
  - Sampling
  - Scale-up methods (e.g., BIRCH).

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.**

$$dist(\mathbf{x}_i, \mathbf{x}_j) = ((x_{i1} - x_{j1})^h + (x_{i2} - x_{j2})^h + ... + (x_{ir} - x_{jr})^h)^{\frac{1}{h}}$$

Euclidean distance and Manhattan distance

- **If *h* = 2, it is the Euclidean distance**

$$dist(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + ... + (x_{ir} - x_{jr})^2}$$

- **If *h* = 1, it is the Manhattan distance**

$$dist(\mathbf{x}_i, \mathbf{x}_j) = |x_{i1} - x_{j1}| + |x_{i2} - x_{j2}| + ... + |x_{ir} - x_{jr}|$$

- **Weighted Euclidean distance**

$$dist(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{w_1(x_{i1} - x_{j1})^2 + w_2(x_{i2} - x_{j2})^2 + ... + w_r(x_{ir} - x_{jr})^2}$$

Squared distance and Chebychev distance

- **Squared Euclidean distance**

$$dist(\mathbf{x}_i, \mathbf{x}_j) = (x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + \ldots + (x_{ir} - x_{jr})^2$$

- **Chebychev distance: one wants to define two data points as "different" if they are different on any one of the attributes.**

$$dist(\mathbf{x}_i, \mathbf{x}_j) = \max(|x_{i1} - x_{j1}|, |x_{i2} - x_{j2}|, \ldots, |x_{ir} - x_{jr}|)$$

Distance functions for binary and nominal attributes

- **Binary attribute: has two values or states but no ordering relationships, e.g.,**
  - **Gender: male and female.**
- **We use a confusion matrix to introduce the distance functions/measures.**
- **Let the *i*th and *j*th data points be $\mathbf{x}_i$ and $\mathbf{x}_j$ (vectors)**

Confusion matrix

		Data point j		
		1	0	
Data point i	1	a	b	$a+b$
	0	c	d	$c+d$
		$a+c$	$b+d$	$a+b+c+d$

a: the number of attributes with the value of 1 for both data points.

b: the number of attributes for which $x_{if} = 1$ and $x_{jf} = 0$, where x_{if} (x_{jf}) is the value of the fth attribute of the data point $\mathbf{x}_i$ ($\mathbf{x}_j$).

c: the number of attributes for which $x_{if} = 0$ and $x_{jf} = 1$.

d: the number of attributes with the value of 0 for both data points.

Symmetric binary attributes

- **A binary attribute is symmetric if both of its states (0 and 1) have equal importance, and carry the same weights, e.g., male and female of the attribute Gender**
- **Distance function: Simple Matching Coefficient, proportion of mismatches of their values**

$$dist(\mathbf{x}_i, \mathbf{x}_j) = \frac{b+c}{a+b+c+d}$$

Symmetric binary attributes: example

$\mathbf{x}_1$	1	1	1	0	1	0	0
$\mathbf{x}_2$	0	1	1	0	0	1	0

$$dist(\mathbf{x}_i, \mathbf{x}_j) = \frac{2+1}{2+2+1+2} = \frac{3}{7} = 0.429$$

Asymmetric binary attributes

- **Asymmetric: if one of the states is more important or more valuable than the other.**
  - **By convention, state 1 represents the more important state, which is typically the rare or infrequent state.**
  - **Jaccard coefficient is a popular measure**

$$dist(\mathbf{x}_i, \mathbf{x}_j) = \frac{b+c}{a+b+c}$$

  - **We can have some variations, adding weights**

Nominal attributes

- **Nominal attributes: with more than two states or values.**
  - **the commonly used distance measure is also based on the simple matching method.**
  - **Given two data points $\mathbf{x}_i$ and $\mathbf{x}_j$, let the number of attributes be r, and the number of values that match in $\mathbf{x}_i$ and $\mathbf{x}_j$ be q.**

$$dist(\mathbf{x}_i, \mathbf{x}_j) = \frac{r - q}{r}$$

Outline

- Text Clustering
- Clustering Algorithms
- 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.**

Text Clustering

Thank you!

Q&A