Unsupervised Learning
- Part 1 -

Road Map

- ❑Supervised vs. unsupervised learning. Clustering
- ❑Types of clustering
- ❑K-Means
- ❑Distance functions
- ❑Handling different types of attributes
- ❑Summary

Supervised vs. unsupervised

- In the previous chapter (supervised learning), data points (examples) are of two types:
  - Labeled examples (by some experts); these examples are used as training set and sometimes, part of them as validation set.
  - Unlabeled examples; these examples, members of the so-called test set, are new data points and the objective is to label them is the same way the training set examples are labeled.
- Labeled examples are used to build a model or method (called classifier) and this classifier is the ‘machine’ used to label further examples (unlabeled examples from the test set).

Supervised learning

So the starting points of supervised learning are:

1. The set of classes (labels) is known. These classes reflects the inner structure of the data, so this structure is previously known in the case of supervised learning
2. Some labeled examples (at least few for each class) are known. So supervised learning may be characterized also as learning from examples. The classifier is built entirely based on these labeled examples.
3. A classifier is a model or method for expanding the experience kept in the training set to all further new examples.
4. Based on a validation set, the obtained classifier may be evaluated (accuracy, etc).

Unsupervised learning

In unsupervised learning:

- ❑The number of classes (called clusters) is not known. One of the objectives of clustering is also to determine this number.
- ❑The characteristics of each cluster (e.g. its center, number of points in cluster, etc) are not known. All these characteristics will be available only at the end of the process.
- ❑There are no examples or other knowledge related to the inner structure of the data to help in building the clusters

Unsupervised learning

- The objective is not to build a model for further data points but to discover the inner structure of an existing dataset.
- In unsupervised learning there is no target attribute: data points are not labeled at the end of the process but the obtained clusters may be further used as the input of a supervised learning algorithm.

Unsupervised learning

- Because there are no labeled examples, there is no possible evaluation of the result based on previously known information.
- Cluster evaluation is made using computed characteristics of the resulting clusters.

Unsupervised learning is a class of Data mining algorithms including clustering, association rules (already presented), self organizing maps, etc. This chapter focuses on clustering.

Clustering

Any clustering algorithm has the following generic structure:

Input:

1. A set of n objects D = {d_1, d_2, …, d_n} (called usually points). The objects are not labeled and there is no set of class labels defined.

Clustering

Input:

2. A distance function (dissimilarity measure) that can be used to compute the distance between any two points.

- ❑ Low valued distance means ‘near’, high valued distance means ‘far’.
- ❑ **Note:** If a distance function is not available, the distance between any two points in D must be provided as input.

Clustering

Input:

3. For the most part of the algorithms the items are represented by their coordinates in a k dimensional space, called attribute values, as every dimension defines an attribute for the set of points.

❑ In this case the distance function may be Euclidean distance or other attribute based distance.

Clustering

Input:

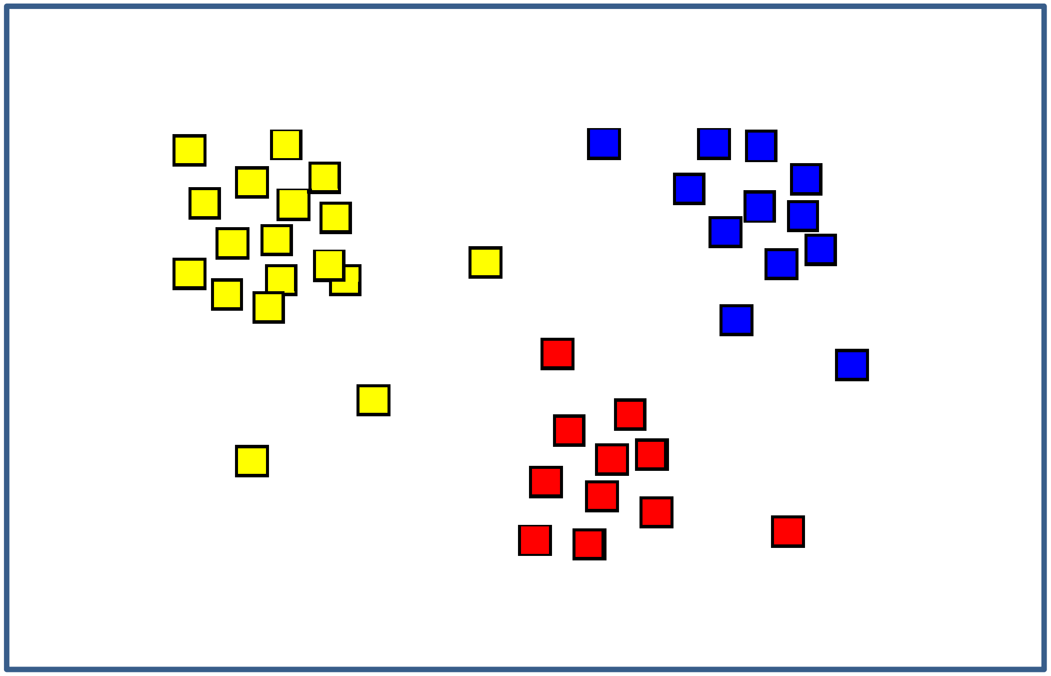
3. Some algorithms also need a predefined value for the number of clusters in the produced result.

Output:

❑ A set of object (point) groups called clusters where points in the same cluster are near one to another and points from different clusters are far one from another, considering the distance function.

Example

- Three clusters in 2D



Features

Each cluster may be characterized by its:

- ***Centroid*** – is the Euclidean center of the cluster, computed as the mass center of the (equally weighted) points in the cluster.
- When the cluster is not in a Euclidean space, the centroid cannot be determined – there are no coordinates. In that case a ***clustroid*** is used as the center of a cluster.
- The clustroid is one of the points of the cluster best approximating its center.

Features

Also each cluster may be characterized by its:

- ❑ ***Radius*** – is the maximum distance from the centroid to the cluster points
- ❑ ***Diameter*** – is the maximum distance between two points within a cluster. Note that the diameter is not twice the radius.

Road Map

❑ Supervised vs. unsupervised learning. Clustering

❑ Types of clustering

❑ K-Means

❑ Distance functions

❑ Handling different types of attributes

❑ Summary

Classification

Based on the method for discovering the clusters, the most important categories are:

❑Centroid based clustering

❑Hierarchical clustering

❑Distribution-based clustering

❑Density-Based clustering

Centroid-based

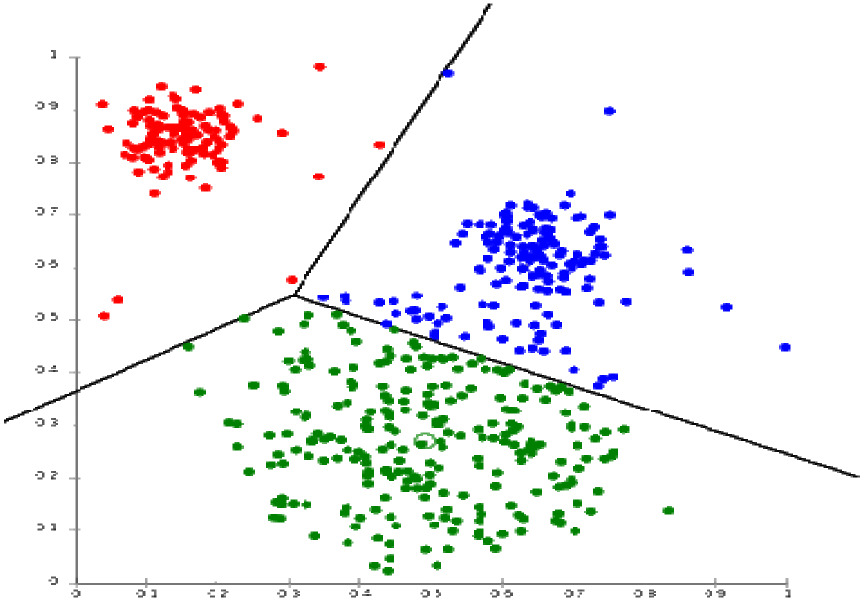
❑In this approach initial centroids are determined in some way and then points are added to the clusters.

❑This method makes directly a partitioning of the dataset.

❑The best known algorithm in this class is k-Means.

Example

❑ K-Means:

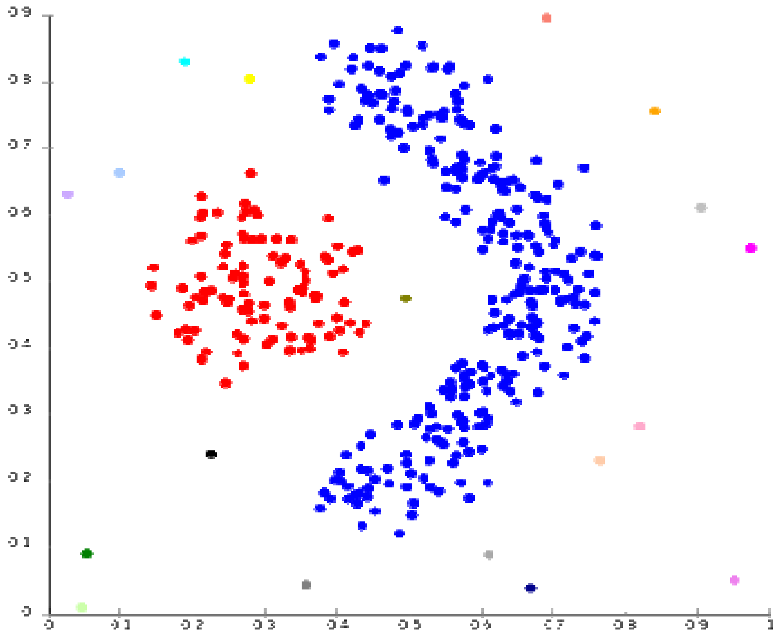


Hierarchical clustering

- The result of a hierarchical clustering algorithm is a dendrogram – a tree having clusters as nodes, leaf nodes containing clusters with a single data point.
- Each node is the reunion (upon merging) of its sons.
- A well known algorithm in this class is BIRCH.

Example

- ❑Results of a hierarchical clustering algorithm:

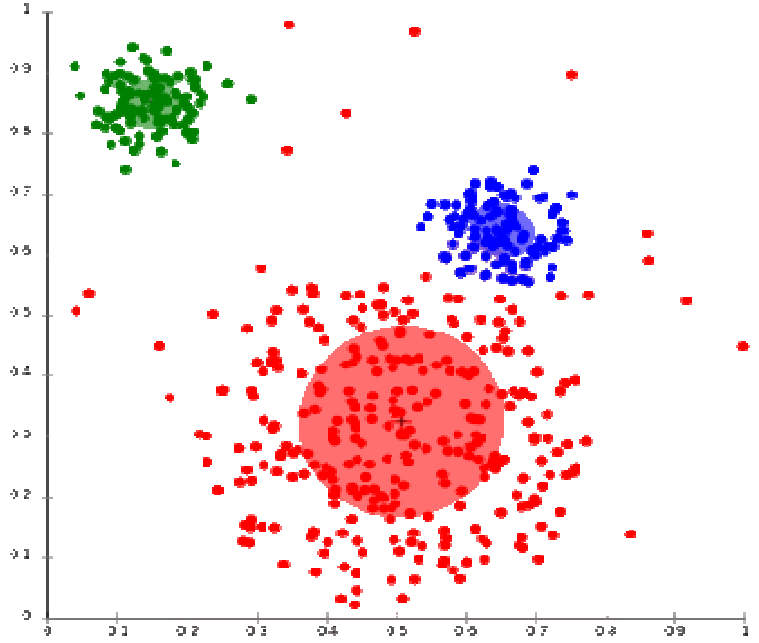


Distribution-based clustering

❑For these algorithms, clusters can be defined as containing objects belonging most likely to the same distribution.

❑Expectation-maximization algorithm is a representative of this class.

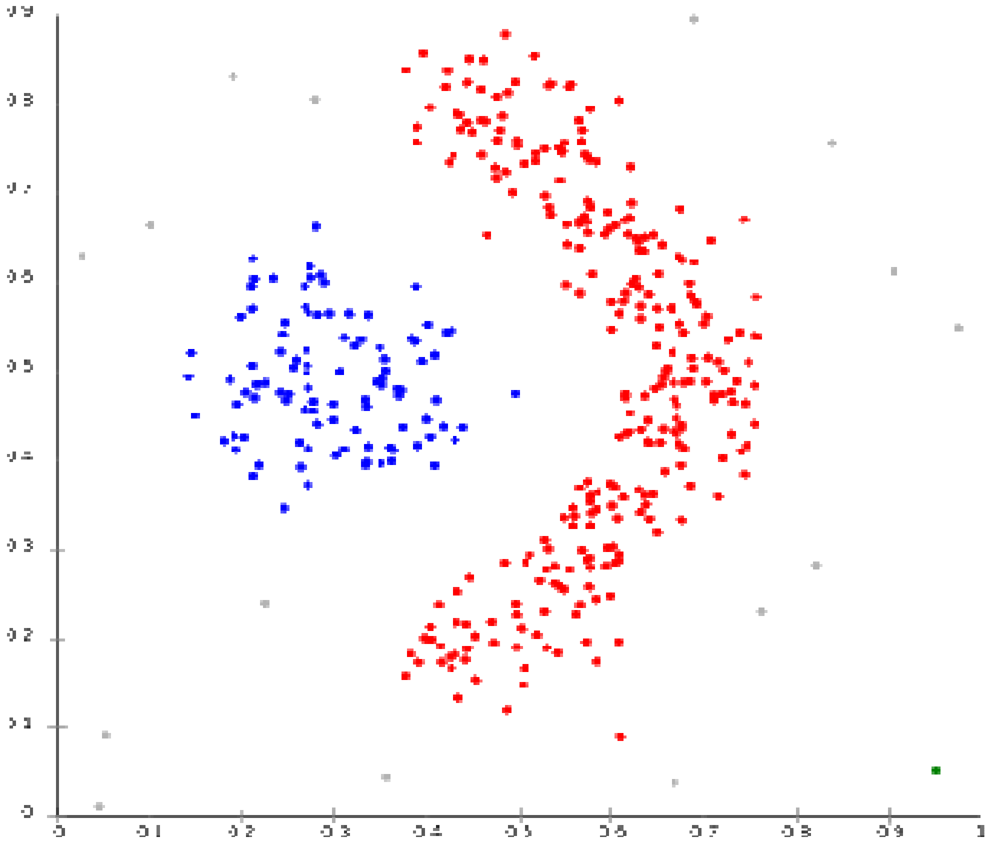
Example



Density-based clustering

- In this case, a cluster is defined as a region with a higher density of points in the data space.
- Examples:
  - DBSCAN
  - OPTIX

Example



Hard vs. Soft clustering

- Based on the number of clusters for each point, clustering techniques may be classified in:

1. Hard clustering. In that case each point belongs to exactly one cluster.
2. Soft clustering. These techniques (called also fuzzy clustering) compute for each data point and each cluster a membership level (the level or degree of membership of that point to that cluster). FLAME algorithm is of this type.

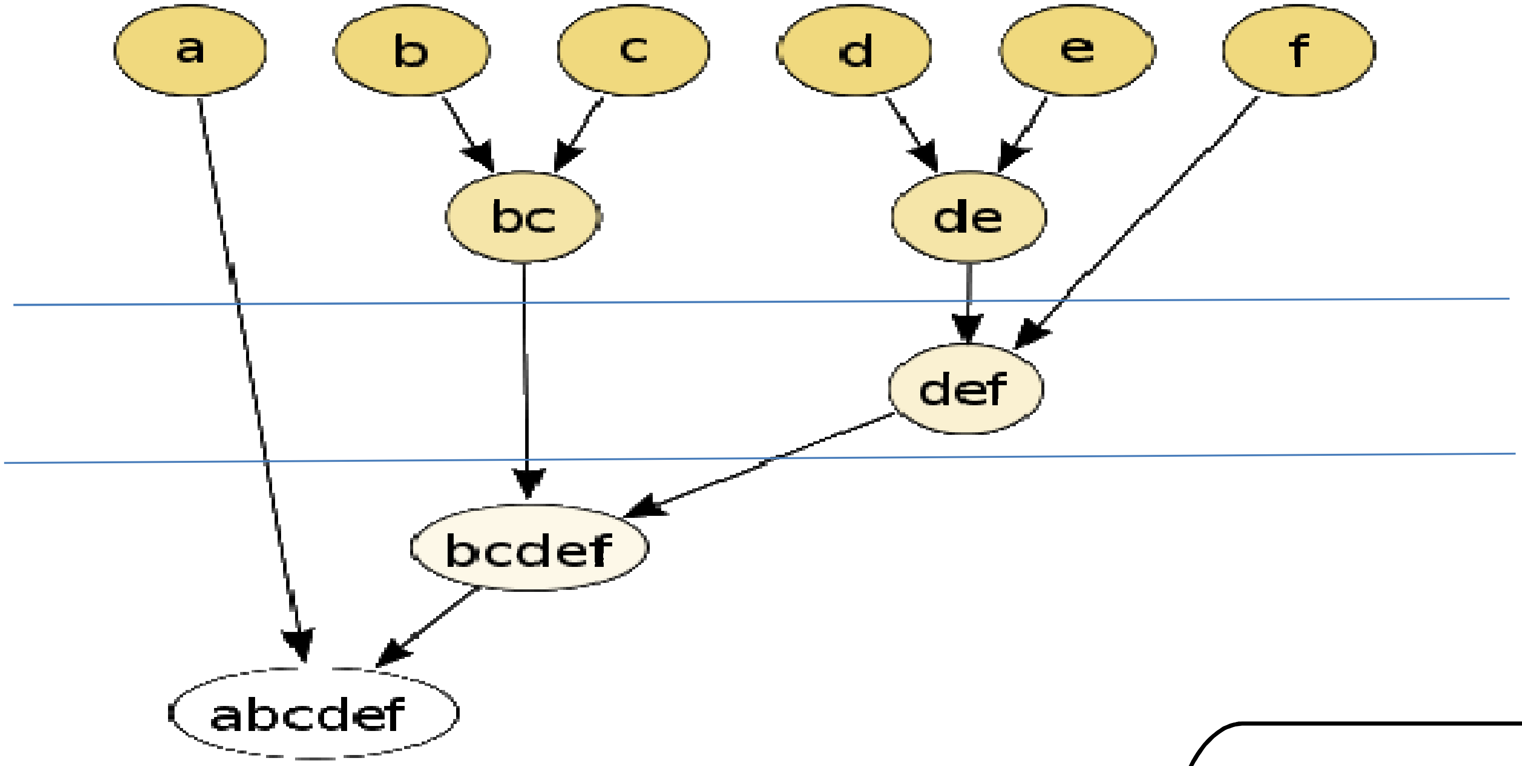
Hierarchical clustering

- ❑Hierarchical clustering algorithms can be further classified in:
  - ❑***Agglomerative*** hierarchical clustering: starts with a cluster for each point and merge the closest clusters until a single cluster is obtained (bottom-up).
  - ❑***Divisive*** hierarchical clustering: starts with a cluster containing all points and split clusters in two, based on density or other measure, until single data point clusters are obtained (top-down).

Dendrogram

- In both cases a dendrogram is obtained.
- The dendrogram is the tree resulting from the merge or split action described above.
- For obtaining some clusters, the dendrogram may be cut at some level.
- For the next example, cutting with the upper horizontal line produces the clusters {(a), (bc), (de), (f)}.
- The second cut produces {(a), (bc), (def)}. Based on clusters' characteristics (see cluster evaluation next week) the best cut may be determined.

Example



Agglomerative hierarchical algorithm

- The agglomerative approach is preferred in hierarchical clustering.
- The sketch of such an algorithm is the following:

Input:

- A set of n points D = {d_1, d_2, …, d_n}, a distance function between them or the distance between any two points.

Output:

- The dendrogram resulting from the clustering process above

Method

```
START with a cluster for each point of D.
COMPUTE the distance between any two clusters
WHILE the number of clusters is greater than 1 DO
        DETERMINE the nearest two clusters
        MERGE clusters in a new cluster c
        COMPUTE the distances from c to the other clusters
ENDWHILE
```

Distance between clusters

❑ For determining the distance between two clusters several methods can be used:

1. ***Single link method***: the distance between two clusters is the minimum distance between a point in the first cluster and a point in the second cluster.
2. ***Complete link method***: the distance between two clusters is the maximum distance between a point in the first cluster and a point in the second cluster.

Distance between clusters

- ❑For determining the distance between two clusters several methods can be used:

3. ***Average link method***: the distance between two clusters is the average distance between a point in the first cluster and a point in the second cluster.
4. ***Centroid method***: the distance between two clusters is the distance between their centroids.

Road Map

- ❑Supervised vs. unsupervised learning. Clustering
- ❑Types of clustering
- ❑K-Means
- ❑Distance functions
- ❑Handling different types of attributes
- ❑Summary

Algorithm description

- ❑ A centroid-based clustering algorithm.
- ❑ The input includes the number of clusters to be obtained (k from the algorithm name).
- ❑ The algorithm structure is:

1. Start choosing k initial cluster centers from the dataset D to be processed.
2. Assign each point in the dataset to the nearest centroid.
3. Re-compute the centroids for each cluster found at step 2.
4. Go to step 2 until some stopping criteria are met.

Conditions

- ❑K-means assumes the existence of a Euclidean space.
- ❑Points have coordinates and re-computation of the centroids is made based on them.
- ❑If the first set of centroids (chosen at step 1) is contained in the dataset D, after the first re-computation the new centroids are not necessarily points in D but some points in the same space.

Conditions

- New centroids are determined as the mass-weight center of the points in each cluster, assuming that each point has the same weight.
- In other words, if each cluster point is represented as a vector, the new centroids are the average values of these vectors.

K-means algorithm

Input:

- A dataset D = {P_1, P_2, …, P_m} containing **m** points in an **n**-dimensional space and a distance function between points in that space.
- **k**: the number of clusters to be obtained

Output:

- The k clusters obtained

Method

```
Choose k points in D as initial centroids
REPEAT
    FOR (i=1; i<=m; i++)
        using the distance function, assign $P_i$ to
            the nearest centroid
    END FOR
    FOR (i=1; i<=k; i++)
        Consider the set of **r** points assigned to centroid i:
                    $\{P_{j1}, …, P_{jr}\}$
        New centroid is $(P_{j1}, …, P_{jr}) / r$
                //(each point is considered a vector)
    END FOR
UNTIL stopping criteria are met
```

Stopping criteria

Stopping criteria may be:

1. Cluster are not changing from an iteration to another.
2. Cluster changes are below a given threshold (for example no more than **p** points are changing the cluster between two successive iterations).
3. Cluster centroids movement is below a given threshold (for example the sum of distances between old and new positions for centroids is no more than **d** between two successive iterations).

Stopping criteria

Stopping criteria may be:

4. The decrease of the SSD (sum of squared distances) described below is under a given threshold.
5. The SSD measures the compactness of the whole set of clusters. It is the sum of the squared distances from each point to its centroid:

$$\mathrm{SSD} = \sum_{i=1}^{k} \sum_{\mathrm{p} \,\in\, \mathrm{Cluster\_i}} \mathrm{Dist}(\mathrm{p}, \mathrm{Centroid}(\mathrm{Cluster\_i})^{2}$$

Weaknesses

1. The algorithm is **sensitive to outliers**.

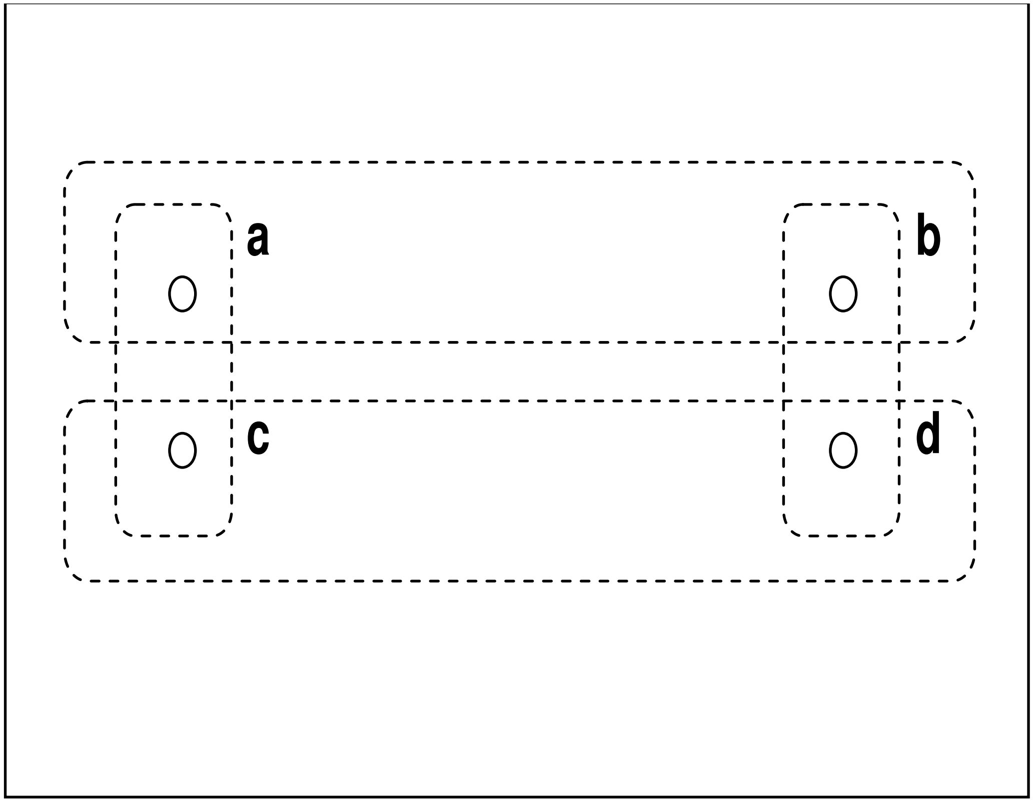
- ❑ These are in many case errors and are points placed far away from any other point.
- ❑ In the presence of outliers the algorithm is trying to include them in some clusters and from that reason the new centroids, computed at each iteration, are far from their natural position (without outliers).

Weaknesses

2. The algorithm is sensitive to the initial position of the centroids.

- ❑ Changing the initial centroids may lead to other resulting clusters, as in the next example.
- ❑ The four points may be grouped in two ‘horizontal’ clusters if the initial centroids are **a** and **c** or in two ‘vertical’ clusters (natural clusters in this case) if initial centroids are **a** and **b**.

Example: initial centroids



Weaknesses

3. From 2 results also that a global optimum solution not guaranteed but a local optimum is obtained.
4. The number of clusters, k, must be provided from outside of the algorithm
5. K-means has good results on clusters with a convex, spherical shape. For non-convex shapes the results are not realistic.

Weaknesses

6. It is not so efficient if data are stored on disks but works well when data may be loaded in the main memory.
7. If the mean of the points may not be computed, the algorithm cannot be used. For categorical data there is a variation of k-means: k-mode.

Strengths

1. It is very simple and easy to implement.

- ❑ There is a great number of packages and individual implementations of k-means.

2. It is an efficient algorithm.

- ❑ Its complexity is linear in number of clusters, number of iterations and number of points.
- ❑ As the first two are small, k-means may be considered a linear algorithm.

Road Map

❑Supervised vs. unsupervised learning. Clustering

❑Types of clustering

❑K-Means

❑Distance functions

❑Handling different types of attributes

❑Summary

A distance function must be:

1. Non-negative: $f(x, y) \geq 0$
2. Identity: $f(x, x) = 0$
3. Symmetry: $f(x, y) = f(y, x)$
4. Triangle inequality:

$$f(x, y) \leq f(x, z) + f(z, y).$$

Distance function

- A distance function is a measure of the dissimilarity between its two arguments.
- The distance between two points is based on the values of the attributes of both arguments.
- If the points are associated with a Euclidean space with k dimensions, then each point has k coordinates and these values may be used for computing the distance.

Euclidean distance

❑ Simple

$$\mathrm{Dist}(x, y) = \sqrt{\sum_{i=1}^{k} (x_i - y_i)^2}$$

❑ Weighted: when some dimensions are more important than others.

$$\mathrm{Dist}(x, y) = \sqrt{\sum_{i=1}^{k} w_i (x_i - y_i)^2}$$

Euclidean distance

❑Squared Euclidean distance

$$Dist(x,y) = \sum_{i=1}^{k} (x_i - y_i)^2$$

❑This squares distance is used when distant points must have more importance

Other distance functions

- ❑ Manhattan distance (city block): the road between the two points may be followed only parallel with the axis

$$\mathrm{Dist}(x,y) = \sum_{i=1}^{k} |x_i - y_i|$$

- ❑Chebychev distance: in the case of hyper dimensionality

$$\mathrm{Dist}(x,y) = \max_i (|x_i - y_i|)$$

Binary attributes

- In some situations all attributes have only two values: 0 or 1 (positive / negative, yes / no, true / false, etc).
- For these cases the distance function may be defined based on the following ***confusion matrix***:
  - a = number of attributes having 1 for x and y
  - b = number of attributes having 1 for x and 0 for y
  - c = number of attributes having 0 for x and 1 for y
  - d = number of attributes having 0 for x and y

Confusion matrix

Data point x	Data point y			
		1	0	
	1	a	b	a+b
	0	c	d	c+d
		a+c	b+d	a+b+c+d

Symmetric binary

❑ When attribute values 0 and 1 have the same weight, the distance can ge computed using the proportion of different values (***Simple Matching Coefficient***):

$$\mathrm{Dist}(x,y) = \frac{b+c}{a+b+c+d}$$

Asymmetric binary

- When 1 is more important than 0, attributes having both a 0 value (their number is d) may be ignored in the distance function:

$$\mathrm{Dist}(x,y) = \frac{b+c}{a+b+c}$$

Nominal attributes

- ❑This is a generalized version of binary attributes above.
- ❑The proportion of dissimilarities may also be used as a distance function.
- ❑Suppose two points x and y having k nominal attribute values each and s the number of attributes where x and y have the same value.

Nominal attributes

- In this case the Simple Matching Coefficient distance is written as:

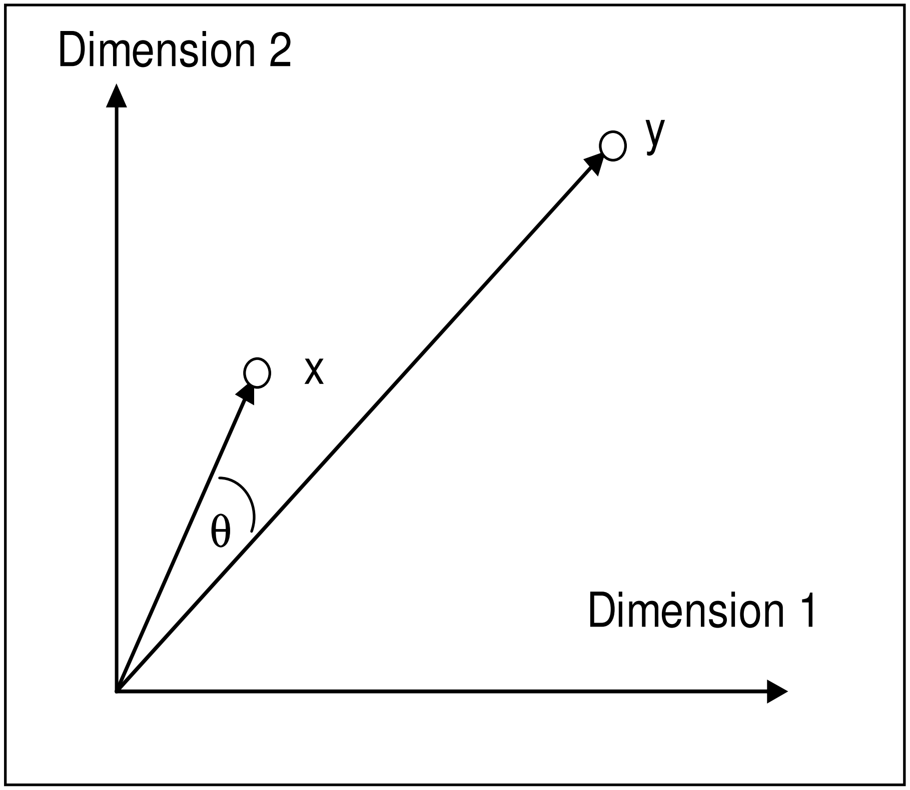
$$\mathrm{Dist}(x,y) = \frac{k-s}{k}$$

Cosine distance

- ❑ Consider two points, $x = (x_1, x_2, \ldots, x_k)$ and $y = (y_1, y_2, \ldots, y_k)$, in a space with k dimensions.
- ❑ In this case each point may be viewed as a vector starting from the origin of axis and pointing to x or y.
- ❑ The angle θ between these two vectors may be used for measuring the similarity: if the angle is 0 or near this value the two points are similar.
- ❑ Because the distance is a measure of the dissimilarity, the cosine of the angle – $\cos(\theta)$ - may be used in the distance function:

$$\mathbf{Dist(x, y) = 1 - \cos(\theta)}$$

Example



Cosine distance

- The value of cos(θ) may be obtained using the dot product of x and y as follows:

$$\cos(\theta) = \frac{\mathbf{x} \cdot \mathbf{y}}{\|\mathbf{x}\| \cdot \|\mathbf{y}\|} = \frac{\sum_{i=1}^{k} x_i y_i}{\sqrt{\sum_{i=1}^{k} x_i^2}\sqrt{\sum_{i=1}^{k} y_i^2}}$$

- The cosine similarity can be used for example in finding the distance between documents

Cosine distance: Example

- If a document is considered a bag of words, each word of the considered vocabulary becomes a dimension. On a dimension, a document has the coordinate:
  - 1 or 0 depending on the presence or absence of the word from the document
  - A natural number, equal with the number of occurrences of the word in the document.
- Considering a document y containing two or more copies of another document x, the angle between x and y is zero so the cosine distance is also equal to 0 (the documents are 100% similar).

No Euclidean space case

- There are cases when the members of the input dataset D have no coordinates.
- In that case the distance function is based on other features.
- An example is cited in [Ullman] and computes the distance between two sequences – for example two character strings or two genes from the DNA.
- The distance function is called Edit Distance

Edit distance

❑ Considering two sequences x and y, the distance between x and y may be defined as the needed number of deletions and insertions of a single sequence element for transforming x in y.

Edit distance example

- Consider strings x and y:

  x = 'Mary had a little lamb'

  y = 'Baby: had a little goat'

- Operations for transforming x in y:
- 2 deletions and 3 insertions to transform 'Mary' in 'Baby:'
- 3 deletions and 3 insertions to transform 'lamb' in 'goat'
- So the distance is 2+3+3+3 = 11.

Edit distance formula

- If we consider LCS(x, y) = the longest common sequence of x and y then the edit distance may be written as follows:

$$Dist(x, y) = \|x\| + \|y\| - 2\|LCS(x, y)\|$$

- For the previous example:
  - ||x|| = 22; ||y|| = 23;
  - LCS(x, y) = 'ay had a little a'; ||LCS(x, y)|| = 17
  - Dist(x, y) = 22 + 23 – 2*17 = 45 – 34 = 11 qed.

Road Map

- ❑Supervised vs. unsupervised learning. Clustering
- ❑Types of clustering
- ❑K-Means
- ❑Distance functions
- ❑Handling different types of attributes
- ❑Summary

Data standardization

- ❑ Sometimes the importance of some attributes is bigger that others only because the range of the values of that attributes is bigger.
- ❑ We can achieve normalization using some transformations
- ❑ Presented also in second chapter

Interval-scaled

- *Min-max normalization:*

$$v_{new} = (v - v_{min}) / (v_{max} - v_{min})$$

- For *positive values* the formula is:

$$v_{new} = v / v_{max}$$

- *z-score* normalization (σ is the standard deviation):

$$v_{new} = (v - v_{mean}) / \sigma$$

Interval-scaled

- *Decimal scaling*:

$$v_{new} = v \; / \; 10^{n}$$

where n is the smallest integer for that all numbers become (as absolute value) less than the range r (for r = 1, all new values of v are smaller or equal to 1).

Ratio-scaled

- *Log transform:*

  - $v_{new} = \log(v)$

- This normalization may be used for ratio scaled attributes with exponential growth.

Nominal, ordinal

❑Nominal attributes:

- ➢Use feature construction tricks presented in the last chapter.
- ➢If a nominal attributes has **n** values it is replaced by **n** new attributes having a 1/0 value (the attribute has/has not that particular value).

❑Ordinal attributes:

- ➢Values of an ordinal attribute are ordered, so it can be treated as a numeric one, assigning some numbers to its values.

Mixed attributes

- In many cases when the attributes of a dataset are not the same type.
- In this case there is no distance function that may be applied to find the distance between points.
- Solutions:
  - Convert to a common type
  - Combine different distances

Convert to a common type

- ❑ If some attribute type is predominant, all other attributes may be converted to that type
  - ➢ Then use a distance function attached to that type.
- ❑ Some conversions make no sense:
  - ➢ Converting a nominal attribute to an interval scaled one is not obvious.
  - ➢ How can we convert values as {sunny, overcast, rain} in numbers?
- ❑ Sometimes we can assign a value (for example the average temperature of a sunny, overcast or rainy day) but this association is not always productive.

Combine different distances

- ❑ A distance for each dimension is computed using an appropriate distance function
- ❑ Then these distances are combined in a single one.
- ❑ If:
  - ➢ d(x, y, i) = the distance between x and y on dimension i
  - ➢ δ(x, y, i) = 0 or 1 depending on the fact that the values of x and y on dimension i are missing (even only one of them) or not.

Combine different distances

- Then:

$$\mathrm{Dist}(x,y) = \frac{\sum_{i=1}^{k} \delta(x,y,i) * d(x,y,i)}{\sum_{i=1}^{k} \delta(x,y,i)}$$

- So δ says if that dimension is considered (value 1) or not (value 0) for the combined distance between x and y.
- The combined distance is the average value of the distances on the considered dimensions.

Summary

This course presented:

- ❑A parallel between supervised vs. unsupervised learning, the definition of clustering and classifications of clustering algorithms
- ❑The description of the k-means algorithm, one of the most popular clustering algorithms
- ❑A discussion about distance functions
- ❑How to handle different types of attributes

❑ Next week: Unsupervised learning – part 2

References

- [Liu 11] Bing Liu, 2011. Web Data Mining, Exploring Hyperlinks, Contents, and Usage Data, Second Edition, Springer, chapter 3.
- [Rajaraman, Ullman 10] Mining of Massive Datasets, Anand Rajaraman, Jeffrey D. Ullman, 2010
- [Ullman] Jeffrey Ullman, Data Mining Lecture Notes, 2003-2009, web page: http://infolab.stanford.edu/~ullman/mining/mining.html