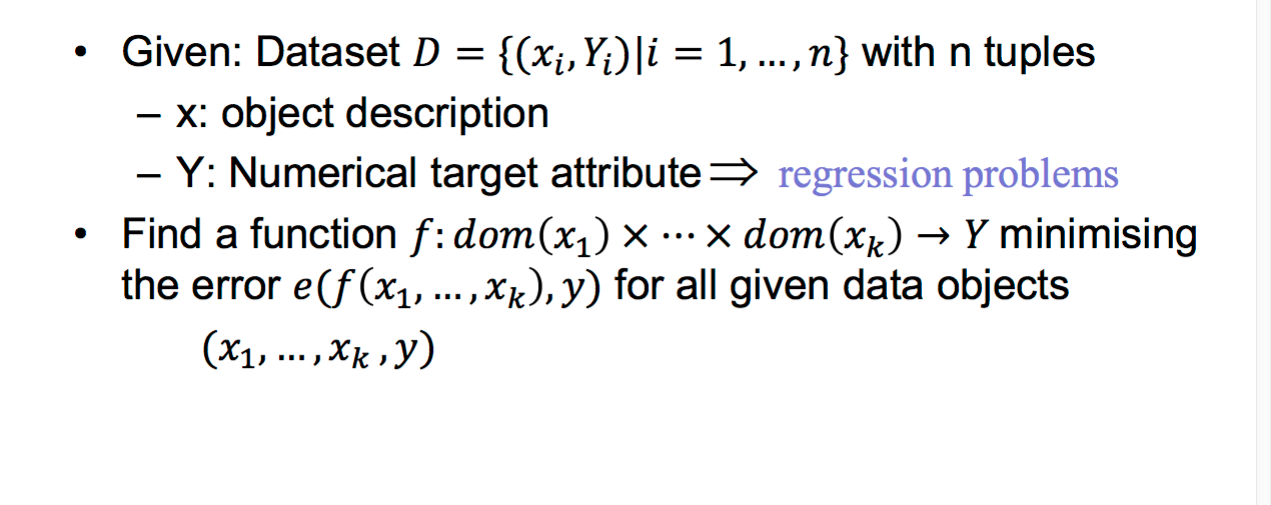
**Data Mining**

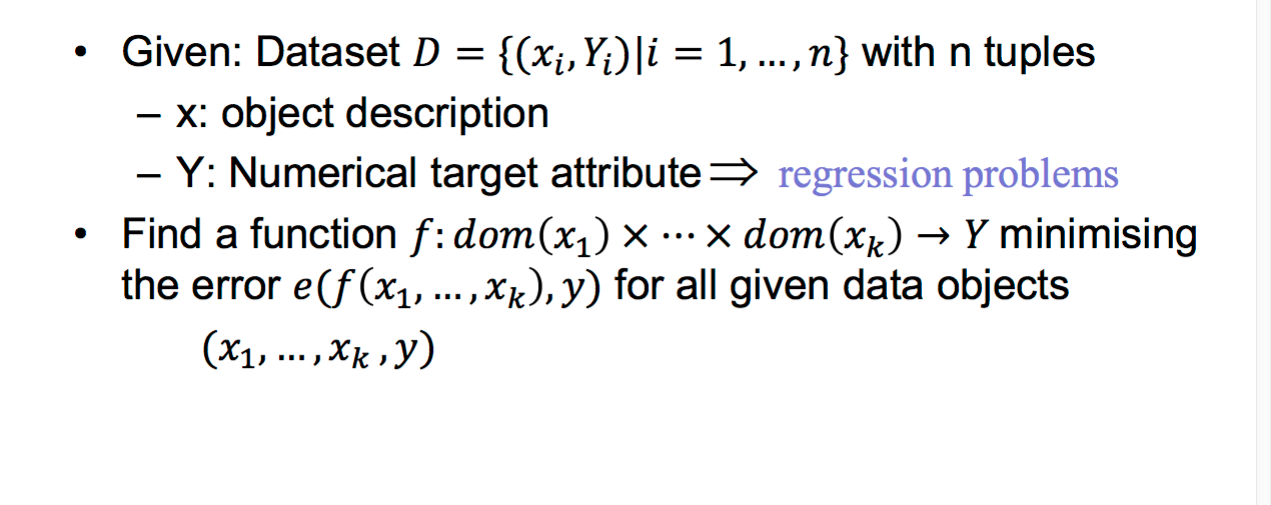
**Clustering**

**From page 10 1-9 revision techniques**

* Classification (predictive)
* Regression (predictive)
* Clustering (descriptive)
* Association rule discovery (descriptive)

**Regression**

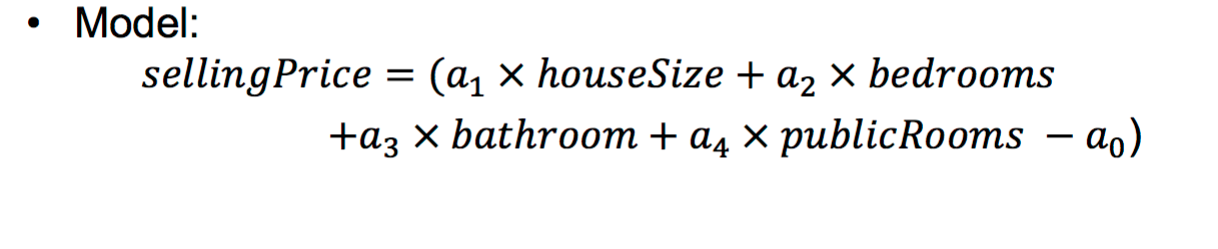




**Linea Regression - Example**

##### Task: Predict housing price

* Attributes
  + houseSize
  + bedrooms
  + bathroom
  + publicRooms
  + sellingPrice



# Association rule mining

* Association rule induction: Originally designed for **market basket analysis.**
* Aims at finding patterns in the shopping behaviour of customers of supermarkets, mail-order companies, on-line shops etc.
* More specifically:

###### Find sets of products that are frequently bought together.

* Example of an association rule:

*If a customer buys bread and wine,*

*then she/he will probably also buy cheese.*

**Association Rules**

##### Assessing the quality of association rules:

* + Support of an item set:
  + Fraction of transactions (shopping baskets/carts) that contain the item set.
  + Support of an association rule X → Y :

Either: Support of X ∪ Y

(more common: rule is correct)

Or: Support of X

(more plausible: rule is applicable)

– Confidence of an association rule X → Y:

Support of X ∪ Y divided by support of X (estimate of P(Y|X)).

## Association Rules

##### Two step implementation of the search for

association rules:

* + Find the frequent item sets (also called large item sets), i.e., the item sets that have at least a user-defined

###### minimum support.

* + Form rules using the frequent item sets found and select those that have at least a user-defined **minimum confidence.**
* Association Rule Induction is a Two Step Process
* Find the frequent item sets (minimum support).
* Form the relevant association rules (minimum confidence).
* Finding the Frequent Item Sets
* Top-down search in the subset lattice / item set tree.
* Apriori: Breadth first search;
* Other algorithms: Eclat, FP-growth, H-Mine, LCM, Mafia, Relim etc.
* Search Tree Pruning:

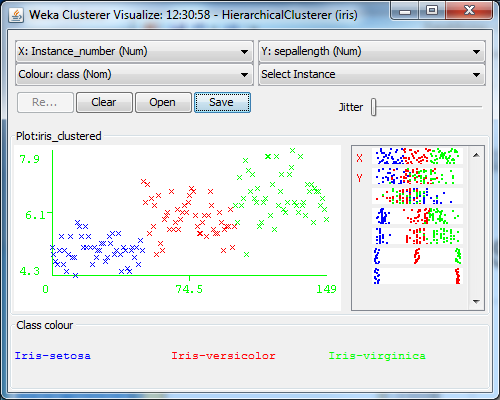
No superset of an infrequent item set can be frequent.

* Generating the Association Rules
* Form all possible association rules from the frequent item sets.
* Filter "interesting" association rules

##### Set a minimum coverage

1. Find all one-attribute associations
2. Find all two-attribute associations
3. Until either reach a specified maximum number of attributes Or can no longer generate associations that have the set minimum coverage
4. Set a minimum accuracy (confidence)
5. Generate rules from each association

Clustering – An example on Iris data



## Finding clusters

#### Types of Clustering Approaches:

##### Linkage Based, e.g. Hierarchical Clustering

* Clustering by Partitioning, e.g. k-Means
* Density Based Clustering, e.g. DBScan

Hierarchical clustering

* Hierarchical clustering builds clusters step by step.
* Usually a bottom up strategy is applied by first considering each data object as a separate cluster and then step by step joining clusters together that are close to each other. This approach is called agglomerative hierarchical clustering.
* In contrast to agglomerative hierarchical clustering, divisive hierarchical clustering starts with the whole data set as a single cluster and then divides clusters step by step into smaller clusters.
* In order to decide which data objects should belong to the same cluster, a (dis-)similarity measure is needed)

#### Hierarchical clustering: Dissimilarity matrix

The dissimilarity matrix [di,j] should at least satisfy the

following conditions:

* di,j ≥ 0, i.e. dissimilarity can’t be negative
* di,j = 0, i.e. each data object is completely similar to itself.
* di,j = dj,i , i.e. data object *i* is (dis)similarity to data object *j* to the same degree as data object *j* is (dis)similarity to data object *i*.

It is often useful if the dissimilarity is a (pseudo-)metric,

satisfying also the following:

* triangle inequality di,k ≤ di,j + dj,k

#### Agglomerative hierarchical clustering: Algorithm

Input: n × n dissimilarity matrix [di,j]

* Start with *n* clusters, each data objects forms a single cluster.
* Reduce the number of clusters by joining those two clusters that are most similar (least dissimilar).
* Repeat step 2 until there is only one cluster left containing all data objects.

Note: Each agglomeration occurs at a greater distance between clusters than the previous agglomeration.

#### Agglomerative hierarchical clustering: Algorithm -- Cont’d

Stop clustering

* + Distance criterion -- when the clusters are too far apart top be merged
  + Number criterion --When there is a sufficiently small number of clusters

#### Measuring dissimilarity between clusters

##### The dissimilarity between two clusters containing only one data object each is simply the dissimilarity of the two data objects specified in the dissimilarity matrix [��,�].

* But how do we compute the dissimilarity between

clusters that contain more than one data object?

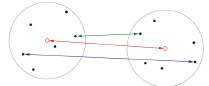
#### Measuring dissimilarity between clusters

* + Centroid - Distance between the centroids (mean value vectors) of the two clusters
  + **Average Linkage** - Average dissimilarity between

all pairs of points of the two clusters.

* + Single Linkage - Dissimilarity between the two most similar data objects of the two clusters.
  + Complete Linkage - Dissimilarity between the two

most dissimilar data objects of the two clusters.



#### Dendrograms

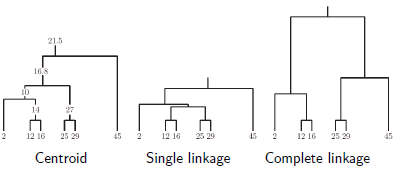
* The cluster merging process arranges the data points in a binary tree.
* Draw the data tuples at the bottom or on the left (equally spaced if they are multi-dimensional).
* Draw a connection between clusters that are merged, with the distance to the data points representing the distance between the clusters.

#### An Example

Clustering of the 1-dimensional data set

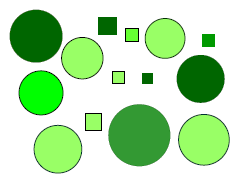
{2, 12, 16, 25, 29, 45}.

All three approaches to measure the distance between clusters lead to different dendrograms

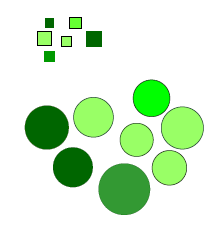


* Simplest Approach:
* Specify a minimum desired distance between clusters.
* Stop merging clusters if the closest two clusters are farther apart than this distance.
* Visual Approach:
* Merge clusters until all data points are combined into one cluster.
* Draw the dendrogram and find a good cut level.
* Advantage: Cut needs not be strictly horizontal.
* More Sophisticated Approaches:
* Analyze the sequence of distances in the merging process.
* Try to find a step in which the distance between the two clusters merged is considerably larger than the distance of the previous step.
* Several heuristic criteria exist for this step selection

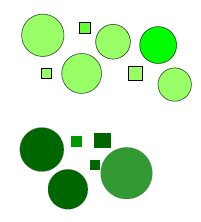
##### How to cluster these objects



**How to cluster these objects**



### How to cluster these objects



**Similarity Measures**

##### Numerical attributes

* + - * Minkowaski
      * Euclidean
      * Cosine

##### Binary Attributes

* + - Nominal Attributes
* Given: dataset of size *n*
* Return: set of typical examples of size

� ≪ �

* Choose a number *k* of clusters to be found (user input).
* Initialize the cluster centres randomly (for instance, by

randomly selecting *k* data points).

* Data point assignment:

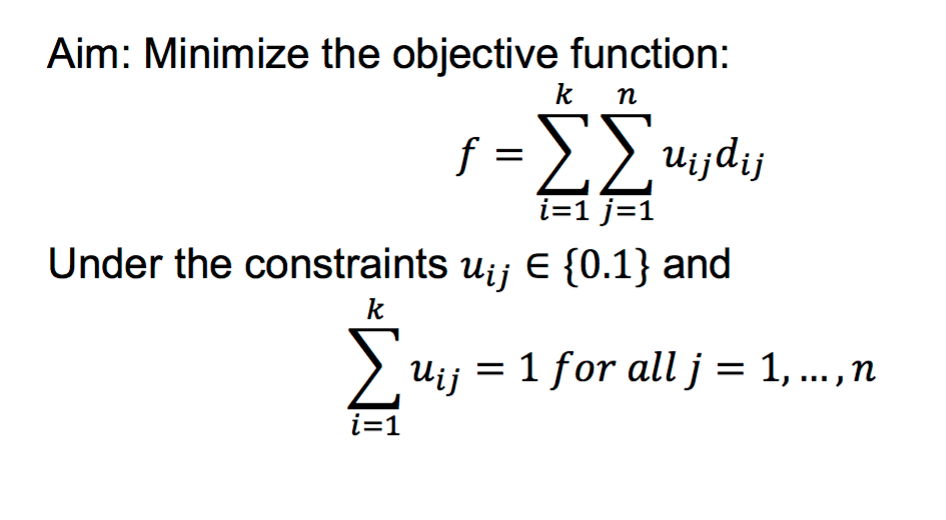
Assign each data point to the cluster centre that is closest to it (i.e. closer than any other cluster centre).

* Cluster centre update:

Compute new cluster centres as the mean vectors of the assigned data points. (Intuitively: centre of gravity if each data point has unit weight.)

* Repeat these two steps (data point assignment and cluster centre update) until the clusters centres do not change anymore.
* It can be shown that this scheme must converge,

i.e., the update of the cluster centres cannot go on forever.



* + Assumption: Data was generated by sampling a set of

normal distributions.

(The probability density is a mixture of normal distributions.)

* + Aim: Find the parameters for the normal distributions and

how much each normal distribution contributes to the data..

* + Assumption: Data were generated by sampling a set of normal distributions.

(The probability density is a mixture of normal distributions.)

* + Aim: Find the parameters for the normal distributions and

how much each normal distribution contributes to the data.

* + Algorithm: EM clustering (expectation maximisation). Alternating scheme in which the parameters of the normal distributions and the likelihoods of the data points to be generated by the corresponding normal distributions are estimated.
  + For numerical data, density-based clustering algorithm often yield the best results.
  + Principle: A connected region with high data density corresponds to one cluster.
  + DBScan is one of the most popular density-based clustering

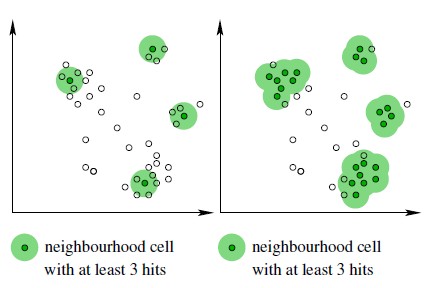
algorithms.

Principle idea of DBScan:

1. Find a data point where the data density is high, i.e. in whose ε-neighbourhood are at least ρ other points. (ε and ρ are parameters of the algorithm to be chosen by the user.)
2. All the points in the ε -neighbourhood are considered to belong to one cluster.
3. Expand this ε -neighbourhood (the cluster) as long as the

high density criterion is satisfied.

1. Remove the cluster (all data points assigned to the cluster) from the data set and continue with 1. as long as data points with a high data density around them can be found..



## References

##### M. Berthold, C. Borgelt, F. Hoppner and F. Klawonn Guide to Intelligent Data Analysis, Spriner-Verlag,

2010

– Chapter 7, Section 7.1, 7.2, 7.3, 7.4

##### I. Witten, E. Frank and M. Hall (2011)

*Data Mining: Practical Machine Learning Tools and Techniques*,

##### Morgan Kaufmann Publishers.

* <http://www.kdnuggets.com/>