**Data Mining**

**Lab 5:**
**Association rules**
**Regression models**

---

## 1. Association rules

**Example (market basket problem).** Let us consider a set of transactions (T1,T2,…,Tn), each one containing a set of items. For instance:

T1: {bread, milk, water}
T2: {bread, meat, water}
T3: {bread, butter, meat, water}
T4: {fruits, water}

We are looking for items which are frequently purchased together and for IF… THEN rules expressing associations between items (e.g. IF bread AND water THEN meat).

In an association rule IF A THEN B (also denoted as A->B) the left hand side term (A) is an antecedent, and the right hand side term (B) is the consequent.

From a given set of transactions one can extract many rules – it is necessary to evaluate their relevance in order to provide ranked list of rules (with the most relevant rules in top).

To evaluate the relevance of a rule we can use at least two measures:

- Support: supp(A->B) = the number of transactions which contain both A and B divided by the total number of transactions
- Confidence: conf(A->B) = the number of transactions which contain both A and B divided by the number of transactions which contain A

**Example:** IF bread AND water THEN meat

A={bread, water}, B={meat}
Supp(A->B)=2/4=0.5
Conf(A->B)=2/3=0.6

Remark: besides these measures there are other indicators which quantify the degree of novelty (or interestingness) of the rule. Such an indicator is the lift, computed as in the following equation:

Lift(A->B)=prob(A,B)/(prob(A)prob(B))

The probability involved in the computation can be estimated as the relative frequency. The rule is interesting if lift value is large. If the lift value is close to 1 this suggests that A and B are not correlated thus one cannot extract useful association rules of type A->B

**Example:** R=IF bread AND meat THEN water
Conf(R)=2/2=1
Lift(R)=0.5/(0.5*1)=1

**APRIORI algorithm**

Input data: set of transaction (each transaction contains a list of items)
Control parameters:

- Minimum support threshold (e.g.: 0.2)
- Minimum confidence threshold (e.g.: 0.9)

The general structure of the algorithm:

**Step 1:** identify the frequent itemsets (itemsets with a support higher than the threshold):

- Identify the frequent 1-itemsets (sets containing only one frequent item) - list $L_1$
- FOR k=2,K DO construct the list $L_k$ containing frequent k-itemsets by joining elements from $L_{k-1}$ (two elements from $L_{k-1}$ having k-2 common elements are joined)

**Step 2:** construct rules by partitioning the itemsets identified at Step 1 in two parts (one part for the antecedent and the other part for the consequent of the rule); only the rules with a confidence level higher than the threshold are kept.

**Exercise 1/ Rattle+R.** Find all association rules with support larger than 0.2 and confidence larger than 0.7 from the set of transactions used in the lecture (file datasets/transactions.csv).
Remark: untick Partition (there is no need to split the dataset into training/validation/testing as rules mining is an unsupervised task).

Rattle: open the file, mark all attributes as input, use Associate (set the values for Support and Confidence) and then Show Rules
R: read the file with transactions (using read.transactions), extract the frequent itemset (using eclat) and the association rules (using apriori). See AssociationRules_ExampleLecture9.r

**Exercise 2/R.** Load the Groceries dataset from the arules package and

a) Identify the top 10 most frequent items (hint: use itemFrequencyPlot)
b) Find the frequent itemsets with a support of at least 0.1 (hint: use eclat)
c) Find the association rules with a support of at least 0.005 and a confidence of 0.7 (hint: use apriori)

**Exercise 3.** Find association rules with a support of at least 0.2 and a confidence of at least 0.7 based on the list of transactions from supermarket.arff

**R:** Hint: the dataset read from the arff file should be converted in a list of transactions (e.g. TransactionData <- as(ListData, “transactions”)
**Weka:**

a) Open in Weka the file supermarket.arff
b) Find association rules using Associate->Apriori (with the default values of the parameters)
c) Apply the same algorithm for other values of the thresholds for the support (lowerBoundMinSupport=0.2) and for the confidence (minMetric=0.75).

## 2. Regression models

### 2.1. Linear regression

In the linear models, the dependence between the predicted variables and the predictors is described by a linear function Y=WX. Depending on the number of components of X and Y there are several types of regression:

- Simple regression: one predictor and one target (e.g. $y=w_1*x+w_0$, where x and y are scalar values)
- Multiple regression: many predictors and one target (e.g. $y=w_k x_k+\ldots+w_1 x_1+w_0$)
- Multivariate multiple regression: many predictors, many targets (e.g. Y=WX); in most cases multivariate multiple regression can be reduced to several multiple regression subproblems

The parameters of the model (elements of matrix W) are estimated based on the data by using a least squares minimization procedure. The typical R function is lm.

### 2.2 Nonlinear regression

If the output values do not depend linearly on the input values a nonlinear model should be estimated. Nonlinear models can be obtained by using nonlinear fitting methods (e.g. nls function in R), regression trees (e.g. rt function in R), RBF networks (e.g. rbf function in R).

**Exercise 4/R.** Estimate the nonlinear dependence between the enzymatic reaction rate and the enzyme concentration (so called Michaelis-Menten equation) by using the nls function. Starting point: SimpleNonlinearRegression_Bioinfo.r

**Exercise 5.** Car price estimation based on various characteristics (dataset autoPrice).

**Rattle:** Open the file autoPrice.arff in Rattle then ignore the first two attributes (symbolling and normalized-losses) and set class as target (it corresponds to the price of the car but it is called class according to Weka rules). Compare the performance of the following regression models:

- Linear regression (Model -> Linear)
- Regression trees (Model -> Tree)
- Neural networks (Model -> Neural Net)

Which car characteristics have a significant impact on the price?

**Weka:**

a) Open in Weka the file autoPrice.arff
b) Use Classify->Functions->SimpleLinearRegression to find a linear relationship between the output attribute (price) and the most relevant input attribute. Analyze the values corresponding to the Correlation Coeficient and Mean Absolute Error.
c) Use Classify->Functions->LinearRegression to do the same thing

d) Use Classify->Functions->MultilayerPerceptron (with the default values of the parameters). Analyze the values corresponding to Correlation Coeficient and Mean Absolute Error.
e) Identify in the category Classify->Trees the variant which allows the construction of a regression tree

Remark. Versions of Weka less than 3.8 contained also a simple RBF Network implementation

**Case study:** prediction of frequency of various types of Algae in rivers based on the river characteristics and the presence of some chemical substances.

Starting point: CaseStudy_Algae.r

**Lab/Home work.**

1. Identify an appropriate regression model to estimate “miles per gallon” depending on various characteristics of the dataset (autoMPG.arff) and analyze the differences (particularly with respect to the regression tress).
2. Implement an RBF network for the estimation of “miles per gallon” starting from the example in exRBF.r (the implementation should be changed in order to be adapted for a multiple nonlinear regression problem).