*Big Data Analytics*

# Exercise 2

This assignment will be discussed in the next exercise lecture on 19.11.2014.

## 1 Generic Data Analysis Tool

The goal of this exercise is to write a small generic data analysis tool in R. In general, this is a free assignment, giving you some leeway for the implementation. However, try to implement the following features:

- Implement your data analysis tool as a function, which takes a data frame as parameter. This parameter should be the only mandatory parameter, i.e., if your tool requires more parameters, give them a default value to make them optional.
- Generate histogram plots for all data attributes. Optionally: Save the plots to files with meaningful filenames.
- Find attribute pairs that either show a very strong or a very weak correlation. To this end, rank all attribute pairs according to their absolute value of the correlation coefficient, and print out the top/bottom elements of the ranking.
- Generate scatter plots for the attribute pairs with very strong/weak correlation.
- Provide support for a label column, by adding an optional parameter (i.e., default value `NULL`) that allows a user to define a certain column as a class label (a reasonable assumption is that only factor columns can be class labels). Modify your code to create histograms and scatter plots to support the visualization of the class labels. For histograms this means that the histograms should be drawn per-class (either absolutely or stacked). In a scatter plot, use different colors or symbols for the different classes.

To test your analysis tool, we will use a data set from the Data Mining Cup 2010 (cf. "dmc2010.zip" in the lecture's download area). Load the file "dmc2010_train.txt" in R as a data frame (hint: `read.csv`) and use your data analysis tool to get a first impression of the data set. Additionally, use the `target90` column as a class label for the data visualization. A quick summery of the DMC 2010 task: Each entry in the data set corresponds to an order in an online shop (the DMC assignment was to predict the column `target90`, i.e., whether or not a customer will re-order within the next 90 days).

## 2 Entropy-based Discretization

Our next goal is to apply the entropy-based discretization approach (lecture: Chapter 3, slide 47) to the features of the DMC 2010 data set.

a) Analyze the entropy of all possible cut-points for every feature of the data set. To this end, generate a plot that shows the entropy in dependence of the cut point for all features.

b) Now determine the best cut-point for every feature of the data set. For this exercise we limit the approach to the case of finding a single cut, i.e., we do not split the intervals recursively. Which feature produces the overall lowest entropy? What can you conclude from this result?

## 3 Principal Component Analysis

Generate a data set by sampling 1000 elements from a two-dimensional Gaussian distribution (hint: check library "MASS") with

$$\mu = \begin{pmatrix} 1 \\ 2 \end{pmatrix} \quad \Sigma = \begin{pmatrix} 2.0 & 1.3 \\ 1.3 & 1.0 \end{pmatrix}$$

a) Inspect the data set by creating a scatter plot.

b) Now compute a PCA (either on paper or in R) by diagonalizing the covariance matrix of the data set.

c) Create a new (two-dimensional) data set that is obtained by projecting the original data onto the principal components. Compare the scatter plot of this new data set to your original data.

d) We now perform a dimensionality reduction by simply removing one of the two dimensions. Apply this dimensionality reduction to both the original and the transformed data. Compute the MSE (mean squared error) resulting from the dimensionality reduction in both cases and compare the result.

## 4 Probability Densities of a Data Generator

The goal of this assignment is to determine the probability densities associated with a data generator. We will write four different data generators, which generate two-dimensional data sets (features $x$ and $y$) of a size $N$. Furthermore, the generated entries of the data sets will have a class label $c$ which is either $A$ or $B$. The rules for the data generators are:

**Generator 1:** Generate the $(x, y)$ data points uniformly on $[-1, +1] \times [-1, +1]$. Assign class label $A$ if a points fulfills $x < 0.2y$, otherwise assign class label $B$.

**Generator 2:** Again, generate $N$ data points uniformly on $[-1, +1] \times [-1, +1]$. The labels are assigned according to the euclidean distance $r$ from the origin $(0, 0)$: Assign class label $A$ if a point has a distance $r \leq 0.5$, otherwise it is of class $B$.

**Generator 3:** Again, generate $N$ data points uniformly on $[-1, +1] \times [-1, +1]$. This time the labels are assigned probabilistically depending on $r$: Assign class label $A$ with a probability of $P = 1 - r$, i.e., at the origin the probability to assign label $A$ is 100%. At a distance $r \geq 1$, the probability is zero, so here we always assign label $B$.

**Generator 4:** For class label $A$, generate $N/2$ data points according to the two-dimensional Gaussian distribution that we have used in task 2. For class label $B$, generate the other $N/2$ data points by a two-dimensional Gaussian with the same $\mu$ as before, but multiple the elements of $\Sigma$ by 10.

a) Implement each data generator as a data structure (hint: use a "list") which offers the functions:

- `generate(N)`, which generates a sample of size $N$ (returned as a data frame).
- `density(x, y)`, which returns $f_{XY}(x, y)$, i.e., the value of the joint probability density evaluated at point $(x, y)$.
- `densityA(x, y)`, which returns $f_{XY}(x, y \mid A)$, i.e., the value of the conditional probability density function of class $A$ evaluated at point $(x, y)$.
- `densityB(x, y)`, which returns $f_{XY}(x, y \mid B)$, i.e., the value of the conditional probability density function of class $B$ evaluated at point $(x, y)$.

b) Visualize the densities $f_{XY}(x, y)$, $f_{XY}(x, y \mid A)$, and $f_{XY}(x, y \mid B)$ in the $xy$-plane as a surface plot.