Ludwig-Maximilians-Universität München
Institut für Informatik
Dr. Eirini Ntoutsi
PD Dr. Matthias Schubert

Knowledge Discovery in Databases II

WS 2015/2016

Übungsblatt 1: Feature Selection

Aufgabe 1-1 Python/Numpy/Scipy Exercises

In this exercise some basic python/numpy/scipy methods and data types are introduced which will be required for the tutorial.

(a) Write a python script which generates numpy matrixes of size 3×5 and 5×3 which are filled with random values, random integers, zeros and ones. Select pairs of these matrices and perform the following operations on them: add, matrix multiplication, add/multiply a scalar, transpose. Generate a 5×3 random matrix and select the second column, the first row, the upper left 2×2 matrix. Reshape your matrix to a 1×20 vector. What is the order of the matrix elements in the resulting vector?

(b) Write a method *arff_to_ndarray* for reading an arff-file containing numerical feature vectors and one nominal class attribute and returns a numpy matrix D and a nominal label vector Y. You should use the package *scipy.io.arff* for your solution. To transform a numpy record array to a numerical *ndarray* use the method view and then reshape the result to the wanted shape.

Aufgabe 1-2 Why Feature Selection?

Feature selection is the task of selecting an informative subset from a given set of features. Answer the following questions:

(a) What is the importance of feature selection from an *experimental* perspective?

(b) What is the importance of feature selection from a *statistical* perspective?

(c) What is the importance of feature selection from a *scientific* perspective?

Aufgabe 1-3 Greedy Forward Selection

The code template FS_template.py contains python code to read labeled feature vectors from an ARFF file (e.g. iris.arff) and compute the l best features either using Information Gain or χ^2-statistics.

(a) Download *FS_template.py* and the data set *iris.arff* from the homepage and analyse the code.

(b) Implement the method *class_counter* building up a dictionary containing the number of occurrences of the elements in *label_list* in *labels*.

(c) Implement the function *compute_entropy* for a given dictionary of labels and counts, and the sum over all counts *all* which computes the entropy in the dictionary.

(d) Implement the method *x2_statistics* for calculating this metric for a given split. The input consists of class dictionaries for both sides of the splits (*counter_l*, *counter_r*) and the number of elements of each side of the split (*all_l*, *all_r*).

(e) Now change the code, so that the feature selection is based on information gain instead of the χ^2-statistics.