Ludwig-Maximilians-Universität München
Institut für Informatik
Prof. Dr. Thomas Seidl
Julian Busch, Evgeniy Faerman,
Florian Richter, Klaus Schmid

Knowledge Discovery in Databases

SS 2016

Exercise 11: Adaboost

We are looking for Tutors for „Einführung in die Programmierung“ for WS2016/17. If you are interested please write a short email to Florian Richter (richter@dbs.ifi.lmu.de).

Exercise 11-1 Adaboost

There are ten objects with the weight 0.01 and nine objects with the weight 0.1 as input for the classifier $m-1$. For following classification results of $m-1$, compute the normalized weighted error ϵ_{m-1}, the weighted coefficient α_{m-1} and the new data weights for the classifier m:

(a) Only one object with weight 0.01 is classified wrongly, the rest correctly

(b) Five objects with weight 0.1 are classified correctly, the rest wrongly

(c) Only one object with weight 0.01 is classified correctly, the rest wrongly

Which of the classifiers has the greatest influence on final prediction? How are the data weights affected?

Exercise 11-2 Nearest Neighbor

In this exercise you have to implement two types of k-NN classifiers and apply it to a real dataset. We use the Lending Club Loan dataset and try to predict whether a credit will be paid back or not. You find all relevant information and the link to the dataset in Exercise 2.
Feel free to try another dataset with your implementation. There are plenty of interesting datasets e.g. in the UCI machine learning repository.

(a) Load the dataset. You can reuse the code from exercise 2.

(b) Choose appropriate features for the k-NN classification with the Euclidean distance. We suggest the following features: *annual_inc, dti, total_rev_hi_lim, revol_util, int_rate*. Experiment with more or other feature sets.

(c) Preprocess the dataset. Ensure that all features are suitable for the Euclidean distance computation.

(d) Remove outliers.

(e) Normalize features. We used min-max normalization. Why is normalization important?

(f) Split dataset in training and test subsets. We took 30% of the instances for testing.

(g) Implement the k-NN classification with majority vote. Run it with k from 1 to 50. How is your classifier performing, is it better than random decisions?

(h) Implement the weighted k-NN classifier. Run it with k from 1 to 50. How is your classifier performing? Is it better than the previous one?