TU München
Fakultät für Informatik
PD Dr. Rudolph Triebel
John Chiotellis, Maximilian Denninger

Machine Learning for Computer Vision
Winter term 2018

December 21, 2018
Topic: Bagging and Boosting

Exercise 1: Bootstrap Aggregation

a) What is the core idea in bagging? How does it differ from boosting?

Bagging:

Is a meta-algorithm to improve the accuracy and stability of machine learning algorithms. It works by randomly sampling points *with replacement* from a training set, which is then used as a training set for one of the classifiers. This is repeated several times to generate M different models/classifiers. The predictions of the different models are in the end averaged.

Boosting:

In contrast to bagging, boosting methods use all samples of the training set in each model exactly once and the models are being *stacked* on top of each other. Important here is that the data points are weighted by each model according to their misclassification rate according to the previous model in the *stack*, thereby *focusing* on the harder samples. In the end, the meta-estimator weights each model according to their final training accuracy.

b) Does bagging reduce the bias of the predictions, the variance or both? Why?

Bagging reduces the variance, by averaging over the learned models. The bias might be even increased, but the variance reduction is relatively larger and therefore the total error is reduced.

c) What is the out-of-bag error and why is it useful?

Out-of-bag error:

The OOB error is calculated on the data points, which were not used during the training of the current specific classifier. The advantage is that these points, have not been seen by the model and are therefore perfect for testing the generalization powers of the current model. If the test would be performed on the training data, the accuracy could be high, but the generalization could be bad. This is an alternative to cross-validation that can only be used with bagging.

Exercise 2: Adaboost (Programming)

Download the file 'banknote_auth.zip' available at the course's website. The data are features of banknotes and the labels indicate whether a banknote is forged or not. The dataset is taken from `https://archive.ics.uci.edu/ml/datasets/banknote+authentication` with some duplicate entries removed. Implement the AdaBoost algorithm with decision stumps as weak classifiers.

a) To begin train on 50% of the data with 20 weak classifiers.

b) Generate a plot of the classification error with respect to the number of weak classifiers. What do you observe?

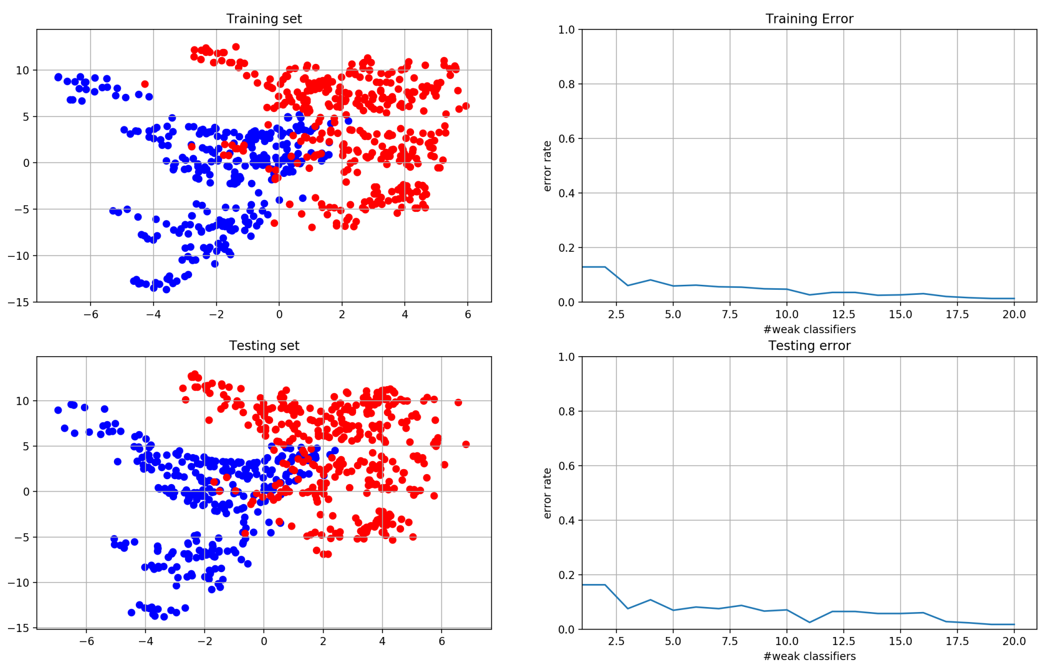

Figure 1: The result for 50 % of the data with 20 weak classifiers.

The test error falls, which each new weak classifier, not constantly, because sometimes the results gets worse, but in general it goes down.

c) Add more weak classifiers. Does the error still change? What's the optimal number of weak classifiers to use?

The optimal number depends on the needs of the problem, if a fast prediction time is more important than a high accuracy, 20 weak classifiers is a good value. If however the accuracy is more important than everything the best value is 68, because at this point the accuracy on the test set is not reduced any more.

d) Now keep the number of weak classifiers fixed and try different training/testing set sizes. How does it affect the classification accuracy?

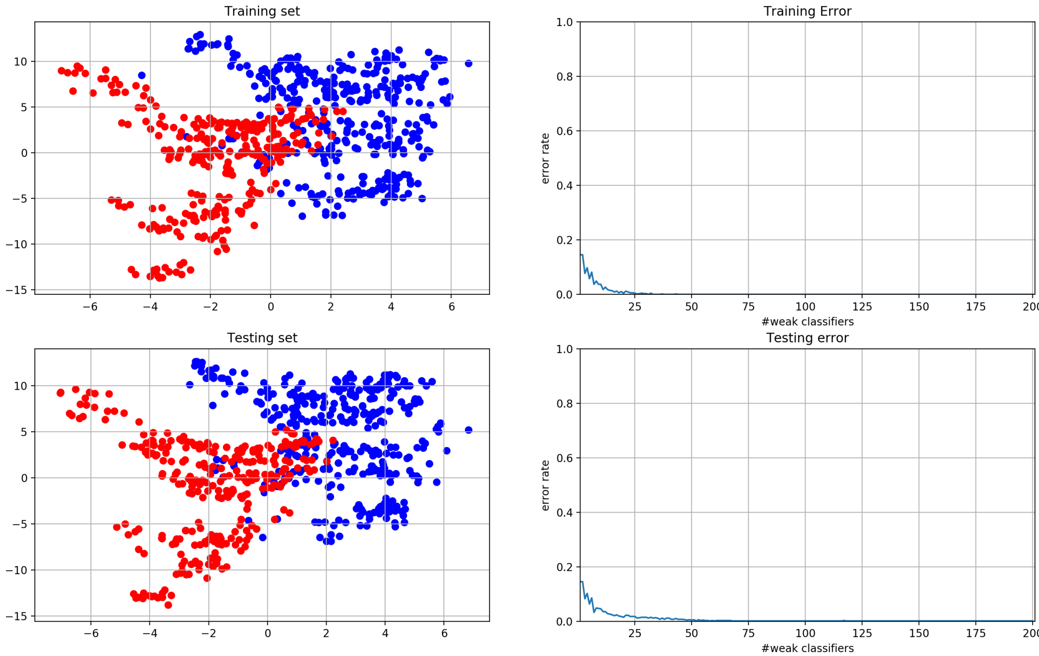

Figure 2: The result for 50 % of the data with 200 weak classifiers.

Trainings amount:	0.1	0.25	0.5	0.75	0.8	0.9	0.95
Test error:	0.198	0.178	0.104	0.0089	0.0074	0.0224	0.0303

In the beginning increasing the training size decreases the test error. However, there is a point where it turns and the test error goes up again, because too many points are used for training and too few for testing.