Home assignment 3

Jan Viilma

May 2018

1 Data set

For data set generation I used an algorithm from "The Elements of Statistical Learning" [1]. It creates $X_1 ... X_{10}$ random features for each observation. These features have a Gaussian distribution. For each observation a corresponding output value is given:

$$Y = \begin{cases} 1 & \text{if } \sum_{j=1}^{10} X_j > \chi^2_{10}(0.5) \\ -1 & \text{otherwise.} \end{cases}$$

where $\chi^2_{10}(0.5) = 9.34$ is the median of a chi-squared random variable with 10 degrees of freedom. To keep a consistency between the data sets on each test run, a constant random seed was used.

2 AdaBoost

For my implementation of adaBoost, I used Matlab's *fitctree*, which generates a simple classification tree. This object also holds the information about the weights of every observation. I ran into quite a few problems when boosting the weak learners. For example when calculating the α_m it is possible that the model that's being used for iteration m, has a loss/error value of 0. In that case α_m will become infinitely large. To counter this problem α_m was calculated like this:

$$\alpha_m = \log\left(\frac{1 - err_m + \epsilon}{err_m + \epsilon}\right)$$

where $\epsilon = 10^-5$. α_m is used as a weight for the m'th weak learner.

We can see the efficiency of the algorithm on figure number 1.

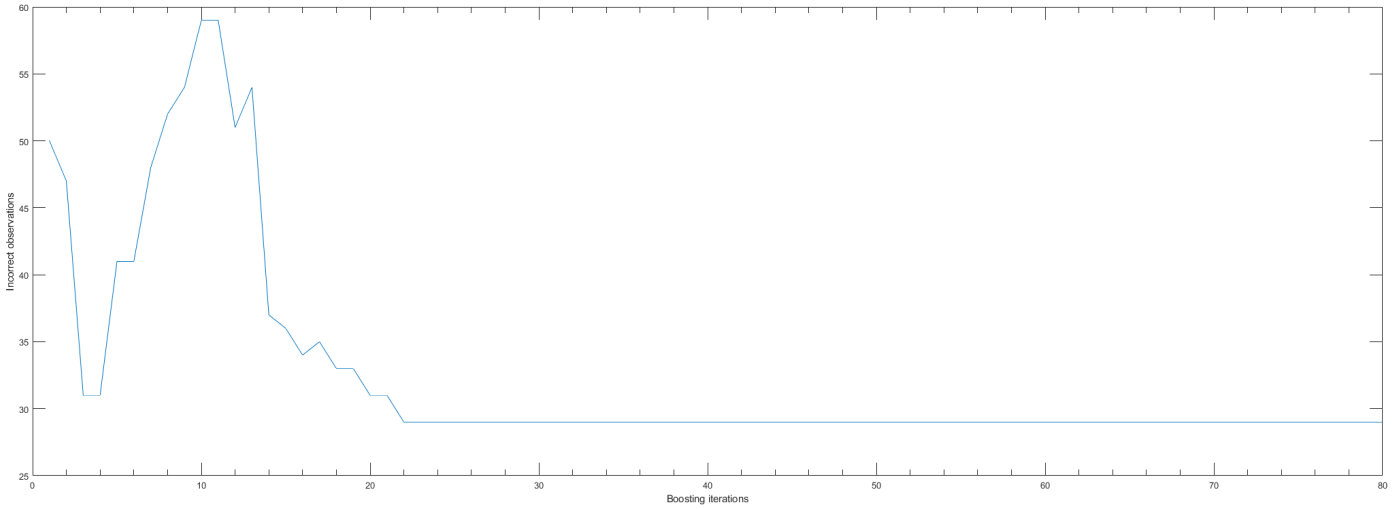

Figure 1: Simulated data

The x axis shows the number of weak learners that are used and the y axis shows how many incorrect classifications were made by the corresponding model. Since the making of these models takes some time, only a 1000 observations were used on each iterations. For the data set that was used here, it can be seen that after a few iterations the results actually get worse and eventually get slightly better than in the beginning. The outcome will of course vary a lot for different data sets.

References

[1] Hastie T., Tibshirani R., and Friedman J. *The Elements of Statistical Learning*. Springer Science and Business Media, 2017.