Underfitting and Overfitting, Learning Curves

WS 2014/2015

The main aim of this tutorial is to illustrate the theoretical concepts from the lectures on simple examples. Specifically, we will be dealing with the underfitting and overfitting of classifiers and the connection to learning curves.

Classification Model, Learning Algorithm

1. In this tutorial, we work with k-conjunctions (k-conjunctions are conjunctions containing at most k literals). You have seen an algorithm for learning conjunctions in the lectures. However, this algorithm is not usable here. First, it assumes that there is no noise in the data and, second, it cannot work with k-conjunctions - just with general conjunctions with unbounded length (bounded only by the number of attributes in data). That is why we use an exhaustive algorithm based on the branch-and-bound method which search for a conjunction having length at most k which minimizes the training error.

   In both tasks, you will use preprepared code (you do not have to write your own code in this tutorial) and you will try to interpret the obtained results.

Underfitting and Overfitting

2. Matlab script `tutorial_8.m` consists of three main parts: data generation, construction of *bias-variance-trade-off* curve and construction of learning curves. Notice that there is a boolean variable `prefer_long` in function `conj_bb`. If this variable is set to 1 then the algorithm selects from all the conjunctions with minimum training error the one which is longest. **Construct *bias-variance-trade-off* curve with data containing noise and data not containing it and with the version of the learning algorithm preferring long conjunctions**

and for version with `prefer_long = 0`. Where is overfitting more pronounced and why?

Learning Curves

3. The third part of the script `tutorial_8.m` is meant for construction of learning curves. Study the code. **Explain how the displayed learning curves correspond to the *bias-variance-trade-off* curve. Test the effect of noise on the learning curves.**