

# Upper Bounds for Classification Errors

You have learnt a method for deriving probabilistic upper bounds on training errors of classifiers (PAC-learning theory). In this tutorial, you will try to apply this theory to learning of  $k$ -conjunctions.

## Upper Bounds for Conjunctions

1. We will use the following inequality which holds with probability  $1 - \delta$  (in case the conditions presented at the lectures are satisfied).

$$e(f) \leq e(S, f) + \sqrt{\frac{1}{2|S|} \ln \frac{2|\mathcal{F}|}{\delta}}$$

Here,  $e(f)$  is the true error,  $\mathcal{F}$  is the concept space and  $e(S, f)$  is the error estimate on a training set  $S$ .

2. Your task is to compute the upper bound on the error for conjunctions with length at most  $k$ . You should display this upper bound in one figure with several learning curves. You should try this also for several  $k$ .

## Implementation details

3. You should construct learning curves using `learningCurve` which can be downloaded with the materials for this tutorial<sup>1</sup>. Function `conj_learningCurve` gets six parameters on its input.

```
function lc = conj_learningCurve(learningFunction, k, ...  
    train, test, train_labels, test_labels)
```

The first parameter is a handle to a function for learning conjunctions, the second parameter is the maximal length of the conjunctions, the third and fourth parameters are training and testing data respectively in the form

---

<sup>1</sup>This function differs from the function with the same name used in the previous tutorial in that it does not average several learning curves but constructs just one learning curve which it also returns.

of matrices (where 1 stands for *true* and -1 stands for *false*) and finally the fifth and sixth argument are class-labels of training and testing data (positive examples are denoted by 1, negative examples are denoted by -1). For generating the training and testing data, you should use the prepared codes in m-file `cviceni_10.m`. For learning the conjunctions, you should use the function `conj_bb` that you used in the previous tutorial. A learning curve can be generated by the following code.

```
lc = conj_learningCurve(@conj_bb, size(train,2), ...  
    train, test, train_labels, test_labels);
```