

Machine Learning 2013: Project 2 - Classification Report

fregli@student.ethz.ch
ganzm@student.ethz.ch
sandrofe@student.ethz.ch

November 22, 2013

Experimental Protocol

This project was performed using Matlab only. To reproduce test results presented in this report the following steps have to be taken:

- Unzip the sourcefolder containing Matlab code and data sets
- Run the function `learn()` located in `learn.m` to obtain both files *testing_predicted.csv* and *validation_predicted.csv*

1 Tools

As stated in Section Experimental Protocol the only tool which is needed is Matlab - no fancy special commands, no additional libraries are required.

2 Algorithm

The algorithm performs the following steps and can be started by running `learn()`.

Read csv files All 3 files read in a first step.

Split training set Training data are split into feature vectors and the expected result vector.

Preprocess features In a next step some features are transformed due to some insights gained through extensive plotting.

Compute result To train our model and to compute the results we simply used the Matlab build in function `svmtrain` and `svmclassify`. This led to quite good results.

3 Parameters

Since we used a rbf kernel, we had to identify the two hyperparameters c (cost) and γ (scaling factor in the rbf kernel). We initially used simple try and error as a starting point and achieved quite impressive results already. Afterwards we applied some simpler version of cross validation to fine tune the parameters. Note that each parameter was identified on its own and not through grid search as it is usually recommended. Therefore it is likely that there are still better combinations of c and γ .

4 Lessons Learned

We tried numerous algorithms but none of them led us to success. We played around with libsvm, but even with the exact same parameters as for Matlab svm we didn't get similar results. Libsvm performed worse in each and every configuration compared to Matlab svm. This might be the cause of some implementational differences of those libraries or due to lack of knowledge from our side.

Overall we learned how to address classification problems with svm. It turned out that Matlab has an impressive implementation of svm, which performs very well out of the box.