Machine Learning 2013: Project 2 - SVM Report

anufer@student.ethz.ch
elmerl@student.ethz.ch
nivo@student.ethz.ch

November 24, 2013

Experimental Protocol

Usage:
Download the csv files to /data/2/....csv (... = training, testing, validation)
Run svm.m
Results are in /data/2/....out (... = training, testing, validation)

1 Tools

- Matlab (code is in /code/ directory)
- Git / Github Repository [1]

2 Algorithm

For training the Support Vector Machine we used the *svmtrain*[2] function from the Statistics Toolbox of Matlab. And for classification of new data we used *svmclassify*[3], which is also a function of the Statistics Toolbox.

These two built-in function form the core of the algorithm. To improve the results, we added an additional pre process step to alter the input data before we give them to *svmtrain*, see 3 Features.

3 Features

Because we use the Gaussian Radial Basis Function as a kernel for the SVM, it is difficult to interpret the features and their importance for the prediction. However, experiments showed that the features

[1] https://github.com/lukaselmer/ethz-machine-learning
[2] http://www.mathworks.ch/ch/help/stats/svmtrain.html
[3] http://www.mathworks.ch/ch/help/stats/svmclassify.html

can be transformed before using the SVM. The formula to transform the features is the following: For the features f_i we do the following transformations (i starts at 1):

- $f_i' = log(log(f_i + 2.00001) + 0.22)$ where $i \ \% \ 3 = 1$
- $f_i' = log(log(f_i + 2.00001) + 0.22)$ where $i \ \% \ 3 = 2$
- $f_i' = log(log(f_i + 1000.0))$ where $i \ \% \ 3 = 0$

Ignoring features f_i where $i \ \% \ 3 = 1$, the estimated error is 0.4715. Using the same procedure, for $i \ \% \ 3 = 2$ and $i \ \% \ 3 = 0$ the estimated error is 0.2683 and 0.1829 respectively. Thus, the features f_i where $i \ \% \ 3 = 1$ seem to be the most important ones.

4 Parameters

The support vector machine with the Gaussian Radial Basis Function as kernel, takes 2 parameters: The box constraint C for the soft margin, and a scaling factor σ.
To find the optimal values for these parameters we first used grid search, to narrow down the range to $\sigma \in [0.5, 0.6]$ and $C \in [1, 1.2]$.
Second, we randomly searched in these ranges for the best values which resulted in following values:

- $\sigma = 0.556201641$
- $C = 1.316157273$

5 Lessons Learned

Even tough many classification algorithms exist, it is difficult to use them out of the box. Even tough there is some documentation, it is often brief, doesn't cover corner cases, or is out of date. For this reason, after playing with Weka[4], and scikit-learn[5], Spider[6], Shogun[7], libsvm[8] and many others, we chose to use the Matlab Statistics Toolbox[9].

Also, many different algorithms have been tried, and the SVM with the Gaussian Radial Basis Function as kernel seems to be the best one. However, some more things which could be done would be:

- Preprocess datapoints and preclassify them, then use different classifiers for each class.
- Use multiple classifiers and then use ensemble methods (voting).
- Test more / combined kernel functions.

[4] http://www.cs.waikato.ac.nz/ml/weka/

[5] http://scikit-learn.org/stable/index.html

[6] http://people.kyb.tuebingen.mpg.de/spider/

[7] http://www.shogun-toolbox.org/

[8] http://www.csie.ntu.edu.tw/ cjlin/libsvm/

[9] http://www.mathworks.ch/ch/help/stats/support-vector-machines.html