Machine Learning 2013: Project 2 - Classification Report

altrif@student.ethz.ch
bvancea@student.ethz.ch
draganm@student.ethz.ch

November 24, 2013

Experimental Protocol

Our experimental protocol was the following:

- We read the input training data into a matrix of feature values.
- We split the data from the training.csv file into two sets: a training set and a test set.
- We trained several classifiers on the training set and measured their generalization performance on the test set. We chose the best parameters for each of the classifier by doing cross-validation over the training set.
- We then submitted the results obtained from the classifiers with the best performance.
- We then tweaked some of the parameters of the chosen classifier by doing more submission and changing the parameters slightly.

1 Tools

For the project we used to following tools:

- The Python programming language
- The Scikit-learn Python library. We used this library for preprocessing data and prediction.
- The Pandas Python library. We used this library for parsing csv files and data management.

2 Algorithm

The algorithm we used for the classification task is SVM (Support Vector Machines). Support Vector Machines classifiers find the optimal separating hiperplane between class labels in the feature space. Due to the fact that the classes might not be separable through a hyperplane, it is possible to use the context of "slack" variables to allow classification errors when choosing the optimal separating hyperplane. It is also possible to reformulate the optimization problem in term of Lagrange multipliers, which is independent of the "slack" variables.

3 Features

On the final submission, the only feature processing is the normalization to zero mean and unit variance. We attempted to apply monomial transformation to features and even do feature selection, however that approach did not provide better results in our case. Features where then projected in a high dimensional space using the Gaussian kernel, as explained below.

4 Parameters

the parameters we needed to choose for the SVM classifier are:

- The kernel used (we chose the one which performed the best among: linear, polynomial, sigmoid and gaussian kernels).
- The penalty parameter C.
- The γ parameter (the radius of RBF) used for the gaussian kernel.
- The weights used for the classes.
- The *tolerance* parameter used by the scikit-learn implementation of the SVM as a stopping condition.

We obtained the best results for the kernel using k-fold cross-validation on the training data. The score used for the cross-validation was the asymmetric classification error CE used for scoring the results on the course webpage. The parameters we used on the final submissions are: a Gaussian kernel with the parameters $C = 4.2$, $\gamma = 1.2$, class weights: $-1 : 0.85, 1 : 0.15$ and *tolerance* 0.25.

5 Lessons Learned

We also attempted to train ensemble classifiers like Adaboost and Random Forests. However, the classifiers we have trained did not perform well enough to pass the hard baseline. We believe that this happened because the current problem is an imbalanced classification problem.