SVM

1. What’s the objective?

The objective is to use the SVM tool from scikit-learn Python machine learning package to train an SVM classifier from the dataset and test it. To be specific, we use the training dataset which contains handwritten digit ranging from 0 to 9, with labels from 0 to 9, to train a classifier, and then test the classifierwith test set to find a good set of hyperparameters.

1. what’s the dataset?

The dataset is MNIST database of handwritten digits. The training set contains 60000 examples, and the test set 10000 examples. It is a subset of a larger set available from NIST. The digits have been size-normalized and centered in a fixed-size image.

1. (how) did you modify it (subtract mean, normalize)?

Since the vector machine algorithms are not scale invariant, we modify the dataset by limiting the features in the range of [-1,1].

1. what method do you use and why?

This is a multi classification problem, scikit-learn provides “one-against-one“ approach by SVC and “one-vs-the-rest“ strategy by LinearSVC.

Wihtout any preprocessing the default svm (SVC), using the radial basis function as kernel, predicts very poorly. The score by applying on a test set consisting of 2000 samples was only 0.1205. We blame wrong parameters (gamma and C) for this result. The linear and polynomial kernel gives significant better scores (0.8425 and 0.8515). However, after some experiments we could increase the efficiency of the svm with rbf kernel by changing the parameters.

1. what experiments do you run?

First of all, we tried all the different already defined kernels of scikit-learn. Secondly we preprocessed the data scaling it to [-1,1] and looked how that changed the scores of the different kernels. In order to find better parameters for the rbf kernel we used GridSearchCV.

1. what are the results?

Without preprocessing:

rbf (0.1205), linear (0.8425), sigmoid (0.1025), poly (0.8515)

With preprocessing and parameter change we got for rbf: 0.8845

1. compare them with the state-of-the-art results

The state-of-the-art linear methods have small error rate, the biggest of which is 12% and the smallest 7.6%. Our method can get the accuracy score of 88.45%.

1. give reasons why your method performed well / poorly

Improvement could be achieved by a different kernel function, parameters or preprocessing.

Reference:

http://yann.lecun.com/exdb/mnist/

http://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html#sklearn.svm.LinearSVC

http://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html#sklearn.svm.SVC