

Banknote Authenticity

Machine Learning - Project Update - 12/12/2017

Miriam Riefel 11332549
Delanie van der Meulen 10729038
Francis Mocsin 11426896

Support Vector Machines - Miriam

The basic SVM algorithm with default parameters was first implemented. The algorithm had an accuracy of 1 on the training set.

Using a grid-search, different parameters were optimised on the training set

<- cross-validation set should be used

Tuned parameters & values (the values in bold turned out to be the optimal parameters):

Cs = [0.001, 0.01, 0.1, 1, **10**]

gammas = [0.001, 0.01, **0.1**, 1]

kernels = ["linear", "poly", "**rbf**", "sigmoid"]

This yielded an accuracy of 1 on the training set <- should be cross-validation set

This yielded an accuracy of 1 on the test set

The algorithm was also evaluated using a confusion matrix, accuracy, error, tp-rate, fp-rate and precision. Error and fp-rate are 0.0, Accuracy, tp-rate and precision are 1.0.

ROC-curve should still be added

In-code notes should still be added

Question: Is it possible that we have a perfect prediction algorithm, or was a mistake made during the optimisation or the division of data etc.

Logistic Regression - Delanie

First the logistic regression classifier was implemented using the default parameters of the Sklearn Logistic Regression function.

Then, using GridSearchCV the following parameters were optimized:

- 'C', the inverse of the regularization parameter lambda, with values ranging from 0.01 to 40 with 400 steps in between
- 'max_iter', the maximum number of iterations, with values ranging from 10 to 1000, with 100 steps in between

C = 20.2555639098 turned out to be the optimal C value, and max_iter = 100 was the optimal amount of iterations.

The accuracy on the training set was: 0.985350435164

The accuracy on the test set was: 0.990213791027

There was also a confusion matrix made for this classifier, for the evaluation process.

Neural Networks - Francis

For Neural Networks I first trained the training set on the default parameters of the MLPClassifier(). After this I tried training it using tuned parameters, where I tuned the amount of hidden layers ((100,5) to (500,5) with steps of 100), activation function ("identity", "logistic", "tanh", "relu"), regularization term (0.0001 to 1 in 10 evenly divided steps), and the maximum amount of iterations (100 to 500 with steps of 100).

As we speak, the classifier is still computing the scores for each of these tuned parameters, so I can't give a detailed report on the accuracy, precision and recall yet. Is this due to me trying to tune too many parameters? Is the step between each parameter too big?

As for the code, we still need to make sure the code is coherent and that we don't have conflicting names for the assignment of each classifier. We still need to add more comments explaining each step.

Lastly, we (Miriam) has made histograms for the features, as well as tables including information about the data and confusion matrices. Should we include these in the final report? (the written part)