Machine Learning E16 - Handin 2
OCR with SVM and (Deep) Neural Nets

Mark Medum Bundgaard, Morten Jensen, Martin Sand Nielsen
Aarhus University

November 2, 2016

1 SVM with SciKit-Learn

Several Support Vector Machine-classifier has been trained on the *AUtrain*-digits with different kernels. See

1.1 Polynomial kernels

A linear(1st order polynomial), a 2nd and 3rd order polynomial has been trained with various values for the hyperparameter, C. This parameter defines how much it should cost for each support vector, that is not kept outside the SVM margins. Too large C and the SVM will tend to overfit the training data to avoid any support vectors in margins. Too low C results in no penalty for not fitting the training data well. The optimal value for this hyperparameter is found by evaluation with a separate validation set. See the comparison of these *in sample* and *out of sample* classification accuracies for various C values in Figure 1, 2 and 3.

FiXme Fatal: should we measure or mention training time? (mention multithreaded solution?)

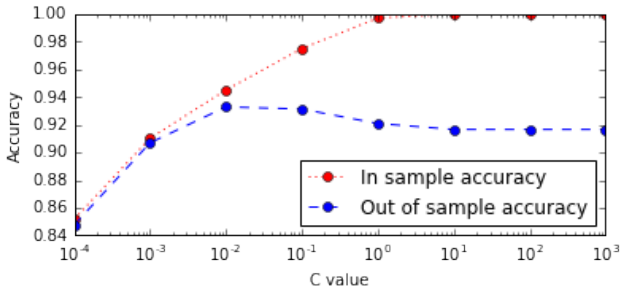

Figure 1: SVM performance with a simple linear(1st order polynomial) kernel for various C values(cost).

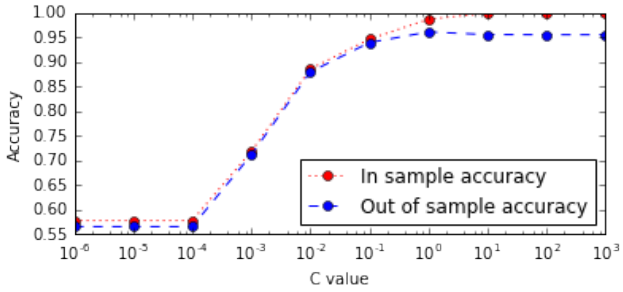

Figure 2: SVM performance with a second-order polynomial kernel for various C values(cost).

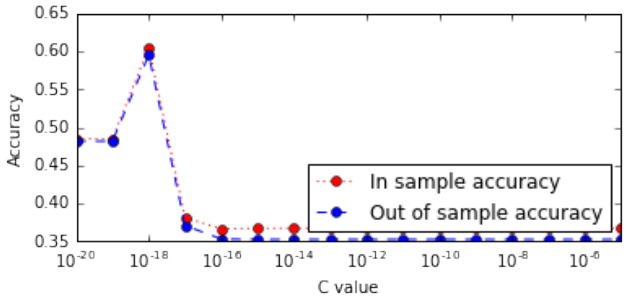

Figure 3: SVM performance with a third-order polynomial kernel for various C values(cost).

1.2 RBF kernel

A more general kernel that is often used, is the Radial Basis Function (RBF). Which expands the raw input feature dimensions (vector of pixel values) to infinite many feature dimensions, such that they results in a similarity measure between images. Beside the usual C cost hyper parameter, RBF also has a γ hyper parameter defining the width of the bell shape for each data point(image).

The results of a coarse grid search for the optimum combination of these parameters can be seen in Figure 5. For comparing the RBF kernel with the polynomial kernels, the optimal $\gamma = 0.01$ has been used in Figure 4.

FiXme Fatal: Comment on overfitting and the different 'areas' on the 3D plot, Morten

A comparison of the kernels can be found in Table 1. RBF is the best with a validation classification accuracy of 96.94%. The search for hyper parameters was only coarse with big step sizes, so there is clearly room for improvements. A better parameter search has to be evaluated by yet another separate validation set, so as to avoid overfitting the hyper parameters to the validation set.

The two dimensional hyper parameter fit for RBF results in many more SVM trainings. In some situations a less complex kernel with faster training time would be preferable.

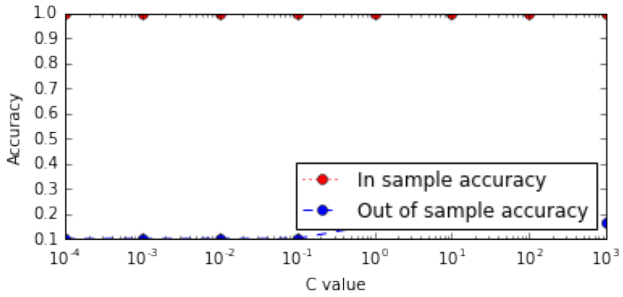

Figure 4: SVM classification accuracy with a RBF kernel with $\gamma = 0.01$ for various C values(cost).

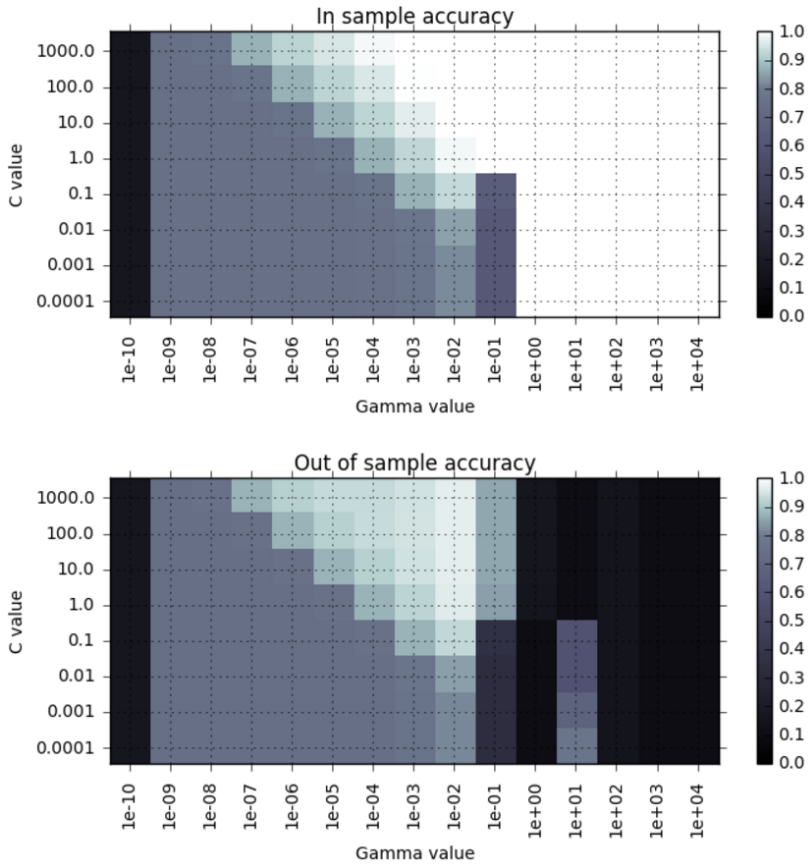

Figure 5: SVM classification accuracy with a RBF kernel with various γ- and C values(cost).

2 Neural Nets with TensorFlow

FiXme Fatal: Hyperparameters: learning rate, batchsize, epochs to run,

FiXme Fatal: training time

FiXme Fatal: results

FiXme Fatal: describe combination of datasets to achieve larger training set

FiXme Fatal:

3 Making the best classifier in 2016 ML Class

Table 1: Classification accuracy of digits with different kernels for the found optimal hyperparameters.

SVM kernel	Validation accuracy	Training accuracy	C-value
Linear	93.29%	94.52%	0.01
2nd order poly.	96.16%	98.74%	1
3rd order poly.	0%	0%	??
RBF, $\gamma = 0.01$	96.94%	99.98%	10

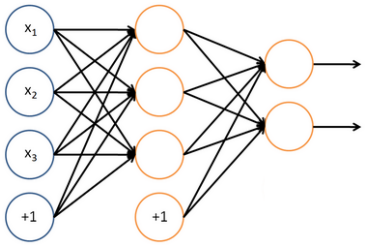

Figure 6: Illustration of a small NN, with one hidden layer. In our case the input vector consists of the 784 pixel values for an image, the hidden layer has 1024 nodes, and the output layer has 10 nodes, one for each digit-class. Biases are added for both computational layers.