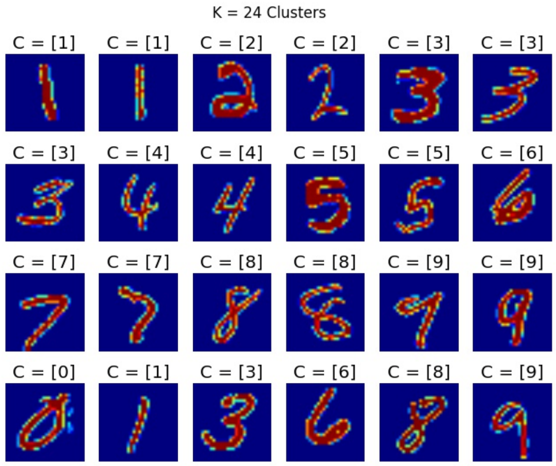
HW5: Support Vector Machines

In this exercise we will explore Support Vector Machines. Again, we will study how to select the optimal regularization parameters.

We will explore that same datasets as we did before, hand-written digits and newsgroup text. The MNIST dataset is comprised of 28x28 pixel images of hand-written digits from the U.S. Census Bureau. Each instance is made up of 784 features (28 x 28 pixel images), each representing a pixel color value between 0 and 1. There are 10 classes to classify, digits 0-9. Here is an example of the hand-written digits you will classify.



The 20 newsgroup dataset is comprised of class 0 (“comp.graphics”) and class 1 (“comp.windows.x”) new wire segments. Some examples are:

y=0 articl repli program work comput distribut system mail code softwar applic inform call internet fax point includ group number address type center id research current video usa develop engin david robert page design accept author contact gmt

y=0 nntp write program file graphic distribut time code find softwar call thing point d sourc includ give function ftp librari inc draw suggest lot wrote site implement stuff access handl object book place year fast output exist routin document robert limit ad pretti easi mac level

y=1 nntp nt articl program find ve make internet server point motif interest gener manag start inc type widget client lot id expolcsmitedu case put result memori xpert handl event back place termin long network design perform great

y=1 nntp write nt window distribut mail softwar ca motif unix manag usa engin david handl object ms design xt

The features are 2997 distinct words (bag of words model) where each news wire case’s feature space would be the words that are found in case. (word present = 1, word absent = 0).

HW5: Support Vector Machines

Again, some of the work has been done for you! You will implement the linear, polynomial, and RBF support vector machines using Scikit learn. As before, you will graph the performance results and determine the best hyper-parameters and model for each dataset. We will test our learners using 5-fold cross validation because it is fast but you know in practice you should use 10-fold cross-validation. The cross-validation code is already written and provided.

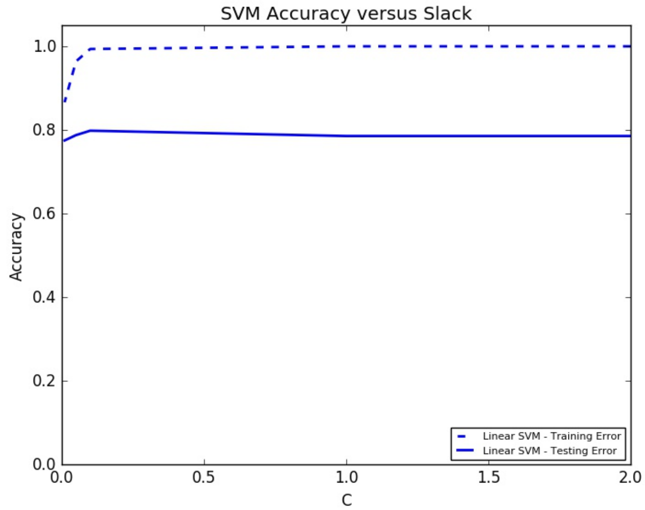
Run.py – I have provided some helper code to get you started. You do NOT have to use this code or you can use whatever snippets you like. But if you prefer writing your own code you are welcome to do so.

Hand-in all code along with a pdf document with the answers and graphs to the questions below. All submitted as one zip file.

HW5: Support Vector Machines

1. Linear SVM

The first learner that you will implement will be the Linear SVM with Scikit learn. You are to evaluate your learner's performance using the following slack hyper-parameters {2, 1, 0.1, 0.05, 0.01}. Your implementation should evaluate your training error, and test error using 5-fold cross validation. You will then graph your results on individual graphs for both the MNIST and 20NG datasets. Make sure whichever Scikit learn function you use, uses a hinge loss function and **not** the squared hinge loss error. It should look something like the graph below although this graph has been generated with a different data set.



(a) Record the test and training accuracy for each hyper-parameter setting for each dataset.

Hyper-Parameter Value	Training Accuracy	Test Accuracy
2		
1		
0.1		
0.05		
0.01		

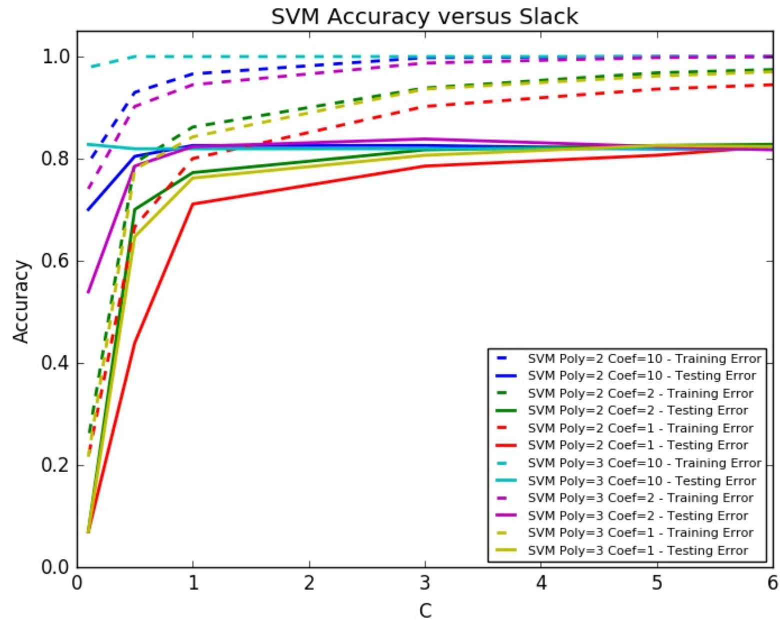
(b) Using your two graphs, determine the optimal regularization parameter value for each dataset. Explain why you think it is the optimal value?

(c) Again, looking at the graphs determine if you see evidence of overfitting or underfitting? If so, for which regularization value(s)? Explain if these results make sense or if they seem odd.

HW5: Support Vector Machines

2. Polynomial SVM

Now you will implement two polynomial SVM classifiers with degree = {2, 3}. You are to evaluate your learner's performance using the following hyper-parameters slack = {6, 5, 3, 1, 0.5, 0.1} and coef0 = {10, 2, 1}. Your implementation should evaluate your training error, and test error using 5-fold cross validation. You will then graph your results on individual graphs for both the MNIST and 20NG datasets. It should look something like the graph below although this graph has been generated with a different data set.



(a) Record the test and training accuracy in a table like the example table for the linear SVM. You should have 2 x 6 x 3 = 36 table rows for each combination of degree polynomial, slack, and coef0. Produce this for each of the two datasets.

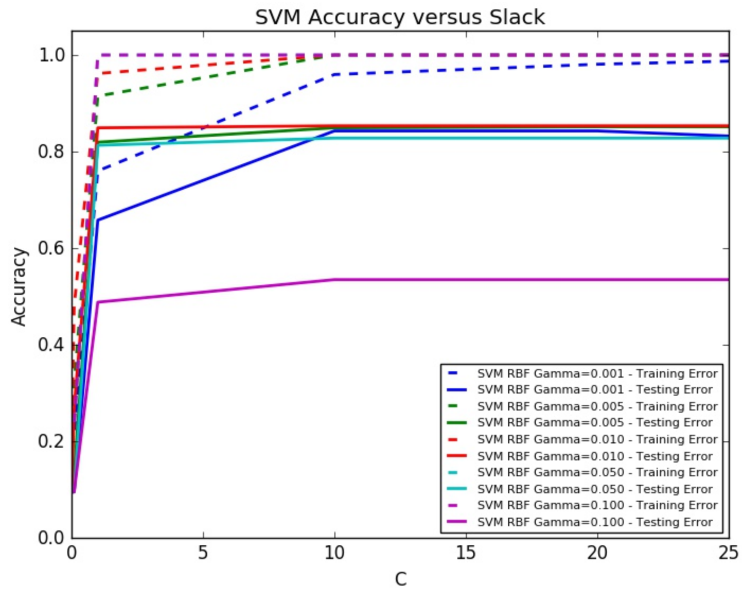
(b) Using your two graphs, determine the optimal degree polynomial model and its optimal regularization parameter values for each dataset. This means you should select 1 best model for each dataset and explain why you think it is the optimal model?

(c) Again, looking at the graphs for each dataset, determine if you see evidence of overfitting or underfitting for **each** of the 6 learned models? If so, for which regularization value(s) in each model? Explain if these results make sense or if they seem odd.

HW5: Support Vector Machines

3. RBF SVM

Now you will implement an RBF classifier. You are to evaluate your learner's performance using the following hyper-parameters slack = {25, 20, 10, 1, 0.1, 0.01} and γ = {0.001, 0.005, 0.01, 0.05, 0.1}. Your implementation should evaluate your training error, and test error using 5-fold cross validation. You will then graph your results on individual graphs for both the MNIST and 20NG datasets. It should look something like the graph below although this graph has been generated with a different data set.



(a) Record the test and training accuracy in a table like the example table for the linear SVM. You should have 6 x 5 = 30 table rows for each combination of hyper-parameters. Produce this for each of the two datasets.

(b) Using your two graphs, determine the optimal RBF model and its optimal regularization parameter values for each dataset. This means you should select 1 best model for each dataset and explain why you think it is the optimal model?

(c) Again, looking at the graphs for each dataset, determine if you see evidence of overfitting or underfitting for **each** of the 5 learned models? If so, for which regularization value(s) in each model? Explain if these results make sense or if they seem odd.

HW5: Support Vector Machines

4. Further Analysis of Results

(a) Which overall model is the best across linear, polynomial, and RBF models for each dataset? Explain why you selected the model that you did beyond just looking at the accuracy.

(b) For the linear SVM in which direction (higher or lower) do you vary slack to ensure less overfitting and better generalization?

(c) For the polynomial SVM in which direction (higher or lower) do you vary degree, corf0, and slack to ensure less overfitting and better generalization?

(d) For the RBF SVM in which direction (higher or lower) do you vary γ and slack to ensure less overfitting and better generalization?

What to Hand in

You should hand in a *.pdf containing all of the necessary write-up items along with your evaluation results and graphs. Also hand in your code. Zip this all up.