Mohammad Shahriar Hooshmand
Ph.D. Candidate
Department of Materials Science and Engineering
The Ohio State University

Neural Network
Project 3: SVM

The purpose of this project is to implement the SVM package named **libsvm** and use it to build a pretty outstanding prediction model. I have used the python interface of this program available on github. In the first part of project, we classify the whole training data using linear SVMs. In order to evaluate the effect of "C" variable, we varied C=(2^{-4},2^{-3},…,2^{7},2^{8}) for each training attempt. Then, using this model, we compared the accuracy of model with respect of "C" value. Fig.1 shows the result of this section. In order to magnify the effect, X scale is in log2 scale. It can be inferred from the figure that as "C" increases the accuracy is improved and using roughly 2 or higher gives us the accuracy of about 90-93%. The parameter "C" controls the tradeoff between minimizing the classification error and maximizing the margin of separation. Our evaluation here shows that for higher accuracy, the former overcomes the latter in this particular case study.

In the second part, we use RBF kernel to build our model. In order to make our model well-predictive, 5-fold cross validation on the randomly chosen 50% of data is performed, as explained in the instruction. The goal of this cross-validation analysis is to choose the best "C" and "α" parameters for our model. Fig.2 shows the result of this analysis on 13x13=169 pair combination of (c, α) values and it is concluded that the choice of C=256 and α=0.0625 gives the accuracy of about 95.2% for cross-validation analysis. Then, we use these optimized (c, α) values to train the whole set and predict the accuracy over non-observed test set. This analysis leads to the accuracy of about 94.2% (fig.3) which is quite outstanding. This analysis is of importance in the sense that the model can be trained pretty well on the portion of data and adjusts the kernel variables and then evaluate a test set which has not been observed before with an incredible accuracy. It is also worth mentioning that the result of these type analysis highly depends on the dataset and can differ case by case. In this particular case, although the accuracy of model is improved a bit (not significantly) by using RBF kernel compared to the linear SVM, but this is not a general conclusion. In fact, depending on the data and how tight/dispersed are distributed, the choice of kernel can improve the accuracy a lot. In conclusion, this project appreciates the powerful feature of SVM kernel tied with the 5(or higher) cross-validation analysis in machine learning prediction.

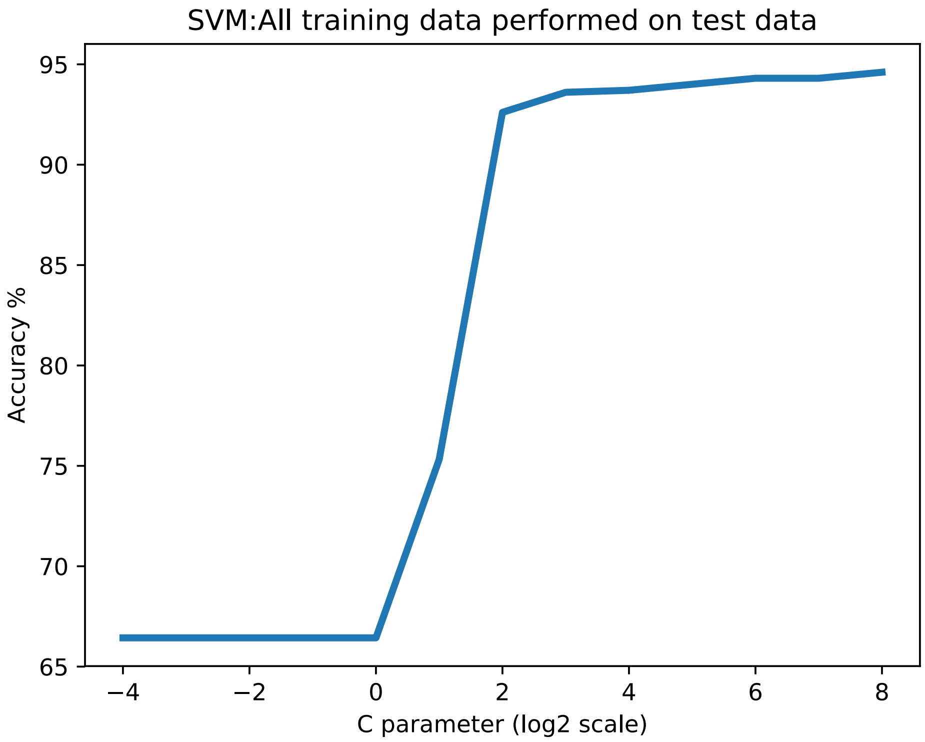

Fig.2: 5-fold cross validation. Optimum **C= 256** and Optimum alpha = **0.0625** with the accuracy of **95.2%**

α

C

	0.0625	0.125	0.25	0.5	1	2	4	8	16	32	64	128	256
0.0625	67.9	67.9	67.9	67.9	67.9	67.9	67.9	67.9	67.9	67.9	67.9	67.9	67.9
0.125	67.9	67.9	67.9	67.9	67.9	67.9	67.9	67.9	67.9	67.9	67.9	67.9	67.9
0.25	67.9	67.9	67.9	67.9	67.9	67.9	67.9	68.3	71.1	71.2	69.9	67.9	67.9
0.5	67.9	67.9	67.9	67.9	67.9	68.1	73.1	78.4	79.8	77.5	74.2	71.5	68.1
1	67.9	67.9	67.9	67.9	71.2	83	88.3	88.3	85.8	82.6	78.5	74.2	71.5
2	67.9	67.9	67.9	75.7	90.3	92.2	92.7	91.6	88.6	85.5	82.1	76.6	72.6
4	67.9	67.9	78.8	92.8	94.2	94.5	93.7	92.3	90.9	86.4	82.3	76.9	72.5
8	67.9	79.4	93.9	94.7	94.5	94.2	93.5	93.6	90.5	86.1	81.5	77	72.6
16	80.4	94.1	95	95	94.6	94.2	93.8	93	89.9	85.5	81.2	77	72.6
32	94	94.7	95.1	94.9	94.7	94.4	93.8	91.8	89.3	84.7	81.3	77	72.6
64	94.7	94.9	95	94.7	94.5	94.5	92.7	91.8	88.8	84.6	81.3	77	72.6
128	95.1	95.1	94.8	94.9	94.2	93.6	92.4	90.2	87.6	84.6	81.4	77	72.6
256	95.2	95.2	94.9	94.4	94.1	92.7	91.1	89.5	87.1	84.5	81.5	77	72.6

Fig.3: SVM prediction based on the training on the whole .train data and prediction on the .test data using optimized **C=256** and ***α*=0.0625**. The accuracy is **94.2058%.**

```
optimization finished, #iter = 2683
nu = 0.205522obj = -89698.750063, rho = -11.555905
nSV = 418, nBSV = 400
Total nSV = 418
Accuracy = 94.2058% (943/1001) (classification)
```