# Project 2: Support Vector Machines

Steven M. Hernandez
Department of Computer Science
Virginia Commonwealth University
Richmond, VA USA
hernandezsm@vcu.edu

***Abstract*— Support Vector Machines (SVM) provide a method of classification that by definition ensure decision boundaries between classifications are maximized, thus enforcing generality within models produced.**

## I. INTRODUCTION

Support Vector Machines (SVM) provide a method of classification that by definition ensure decision boundary margins between classifications are maximized, thus enforcing generality within models produced. Through this project, we explore a two class, linearly separable dataset in which we employ a Hard Margin Linear SVM in addition to exploring multiclass soft classification using a 1-vs-All Soft Margin Non-Linear SVM classifier using both polynomial and Gaussian kernels. Applying cross-validation ensure the highest quality parameters for each of these kernel methods.

## II. HARD MARGIN LINEAR SVM

Starting from the most basic, we exploring SVM classifiers through a hard margin linear SVM which is able to model linearly separable datasets. In this first case, we begin with only two classes, which we can identify as positive and negative. Figure 1 presents our dataset along with the SVM model. For this model, the alpha values for the three support vectors are $0.4362$, $0.8538$ and $0.4177$, bias $b = 3.1700$ and Margin $M = 1/||\mathbf{w}|| = 0.7652$.

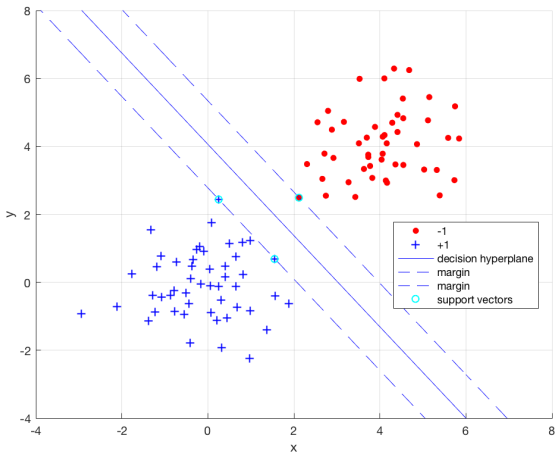


Fig. 1. Linearly Separable Dataset with two classes separated by a Hard Margin Linear SVM where margin M = 0.7652.

Using the calculated values for $\mathbf{w}$ and $b$, we can calculate $o = sign(\mathbf{w}^T\mathbf{x}+b)$ for the following values of $x$: $[3\ 4]^T$ and $[6\ 6]^T$ which both result in $-1$ as can be expected.

## III. MULTICLASS SOFT CLASSIFICATION

Hard margin linear SVMs only handle cases where classes are linear seperable, however data in the real world may appear noisy or naturally appear with data overlap, thus we must find a method where our model can handle this noise and potential overlapping of classes. In this section, we look to employ a soft margin SVM which allows for a certain amount of overlap as defined by a parameter $C$. As $C$ approaches infinity, our model allows less overlap or fewer misclassified data points. Hard margin linear SVMs can be considered Soft margin linear SVMs where $C = \infty$.

For this section, we use the *glass* dataset. As a result, in addition to the requirement of soft-margin classification, we must also consider the *glass* dataset contains data points from seven unique classes; additionally, each input data point has dimension $dim = 9$, thus we are unable to visualize the dataset as a whole. Finally, instead of linear kernels, we try classification using both polynomial and Gaussian kernels. While we are not able to visualize the whole 9-dimensional space that the *glass* dataset resides in, Figure 2 gives us a slight idea of the purpose for using a soft-margin classifier as well as the use of non-linear kernels.

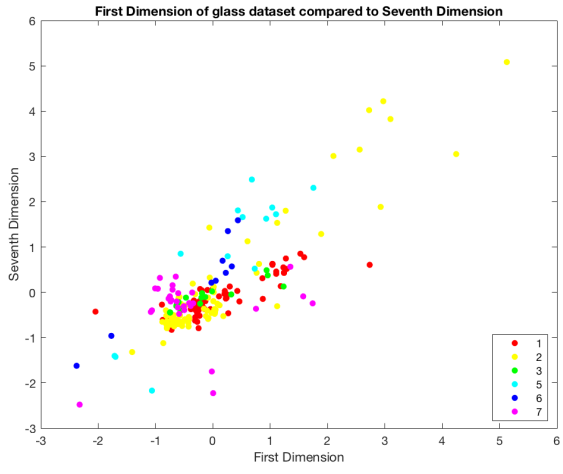


Fig. 2. Plot of arbitrarily selected dimensions (1 and 7) from the glass dataset shows the need to allow for overlap in SVM classifiers.

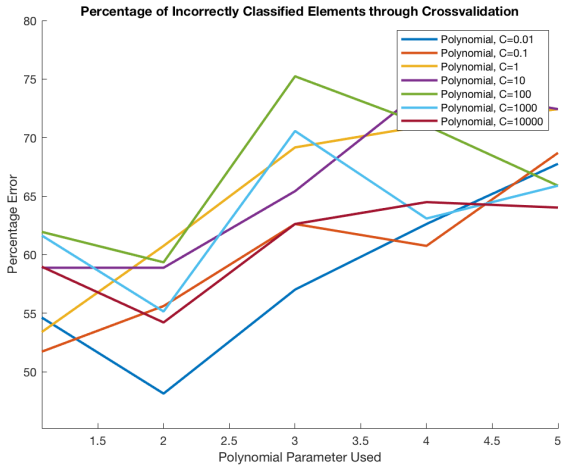


Fig. 3. Percent error in classification of elements when using a polynomial multiclass soft classifier SVM.

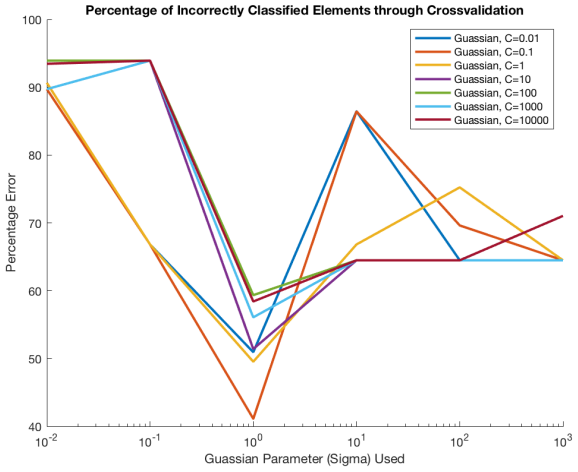


Fig. 4. Percent error in classification of elements when using a Gaussian multiclass soft classifier SVM.

In finding a suitable model, we look to 5-fold cross-validation which provides a method for validating our model uses the best parameters. For polynomial SVM, we test each the following parameters for the *degree of the polynomial* $param \in [1\ 2\ 3\ 4\ 5]$, while for the Gaussian SVM, we test each parameter for sigma $param \in [1e^{-2}\ 1e^{-1}\ 1\ 1e^{1}\ 1e^{2}\ 1e^{3}]$. On top of this parameter, for both polynomial SVM and Gaussian SVM, we test the $C \in [1e^{-2}\ 1e^{-1}\ 1\ 1e^{1}\ 1e^{2}\ 1e^{3}\ 1e^{4}]$.

### A. Polynomial SVM and Gaussian SVM Comparison

Figure 3 and Figure 4 present the percent error in classifying elements when using cross validation for both polynomial and Gaussian kernels. We can see in Fig. 2 when degree $d < 2$, percent error increases to between $65 - 75\%$ for all values of $C$. This indicates the model at these high polynomials are overfitting to our dataset, thus using these parameters will likely result in poorly generalizing models. On the other hand, when $d = 2$ results in better accuracy with the polynomial SVM where $C = 0.01$ producing the lowest percent error. Results for the Gaussian soft margin SVM (Fig. 3) show that $\sigma = 1$ results in the best accuracy for the given values of $sigma$ for all values of $C$ with $C = 0.1$ providing the lowest percent error $41\%$.

Using optimal parameters determined through cross-validation, we produce two SVM models, one polynomial where $C = 0.01, d = 2$ and one Gaussian where $C = 0.1, \sigma = 1$. Polynomial SVM results in $53.2710\%$ percent error while the Gaussian SVM results in $32.7103\%$ when trained and tested on the full **glass** dataset.

## IV. TECHNICAL NOTES

### A. MATLAB Performance

A result of the high number of parameters tested through cross validation along with the time taken for computing the **quadprog** function, we end up with a total computation time of 321.841874 seconds. Page 43 of [1] indicates when the Hessian matrix is positive defined (which both polynomial kernel and Gaussian Radial Basis Function produce), the quadratic programming optimization problem can be solved without the equality constrained $y^T\alpha = 0$, resulting in quicker computations. Removing this equality constraint within the MATLAB script results in a total computation time of 283.563615 seconds, a difference of 38.2783 seconds of a speedup of 1.1350. Not a huge speedup, but worth considering. For the figures and values within this paper however, the original method is taken.

### B. MATLAB Issues

When developing our Hessian Matrix $H$ for use in the MATLAB function *quadprog* in the case of the Gaussian Radial Basis Function, issues arise. The kernel when creating $H$ uses the supplied **grbf_fast.m** sub_routine. Through this method, occasionally the produced $H$ is not symmetrical as in $H \neq H'$. For these cases, reviewing the difference between non-symmetrical elements reveals only a small difference between the elements (i.e. below $1e^{-5}$). One option explored was to ensure symmetry for all $H$ before passing to **quadprog**. To decrease overhead in computations however, we simply opted to suppress this specific warning.

## V. CONCLUSION

The unique property of Support Vector Machine to maximize margins surrounding decision boundaries between classes ensure the generalizability of future predictions of SVM models. In section $II$ we explored this property of SVMs presenting a Hard Margin Classifier requiring only three Support Vectors. Further; in section $III$, we presented the issue of data overlap, thus presenting a Soft Margin Polynomial and Gaussian SVM Classifier which allows for certain amount of incorrectly classified elements in non-linear spaces.

## REFERENCES

[1] T. M. Huang, V. Kecman, I. Kopriva (2001). Kernel Based Algorithms for Mining Huge Data Sets. Springer, p.11-48.