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 $M = 1/||\mathbf{w}|| = 0.7652$.

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.

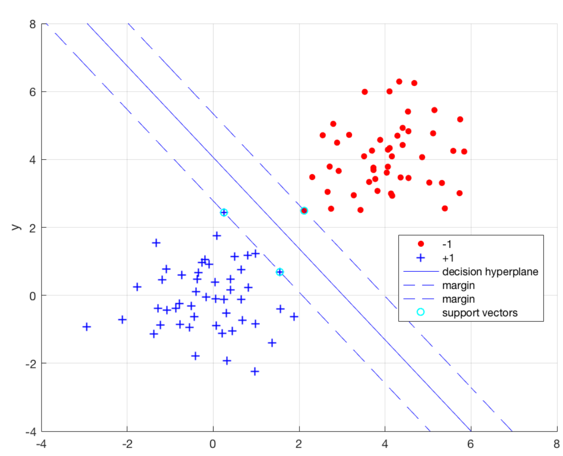

Fig. 1. Linearly Separable Dataset with two classes separated by a Hard Margin Linear SVM where margin M = 0.7652.

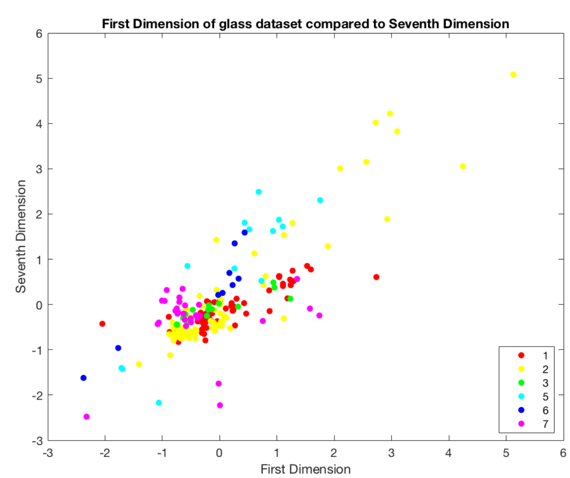

Fig. 2. Plot of arbitrarily selected dimensions (1 and 7) from the glass dataset shows the need to allow for overlap in SVM classifiers.

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

When running the SVM with both Polynomial and Gaussian kernels, we find that the Gaussian kernel does better with an accuracy of 92.5234% while the Polynomial kernel's accuracy is 88.7850%. We can see in table I the best parameters learned through cross-validation for each individual class of the polynomial SVM. We can see that classes $1-2$ do not have quite as good accuracy as the other classes with accuracy at and above 90%. Table II shows on the other hand the best parameters per class-classifier for the Gaussian SVM. We can see that the Gaussian kernel has troubles with classes $1-2$ as well.

TABLE I
BEST PARAMETERS PER CLASS FOR POLYNOMIAL SVM

Class	C	Degree	Accuracy %
1	0.1	4	69.63
2	0.01	5	72.43
3	0.01	1	89.25
5	100	4	97.20
6	1	2	97.20
7	0.1	2	96.73
Multi-class Accuracy: 88.7850%			

TABLE II
BEST PARAMETERS PER CLASS FOR GAUSSIAN SVM

Class	C	Degree	Accuracy %
1	10	1	67.29
2	1	1	64.49
3	10	1	92.06
5	1000	1	94.39
6	10000	10	95.79
7	10	10	96.26
Multi-class Accuracy: 92.5234%			

Rather surprisingly, at a per-class level, most Polynomial kernel classifiers individually achieve higher accuracy than the Gaussian kernel classifier counterparts. These classes are combined to create the multi-class classifier such that we run our test data (in this case the full dataset) on each individual classifier, then returning the class of the classifier which produced the highest value. By this, we applied a **MAX** operator across all the classifiers's results. With the combined multi-classifier using these individual classifiers, this is surprising that Gaussian would perform better, but this may point to the fact that while the Gaussian does not do quite as well per-class, the per-class classifiers actual result in much higher values when applying the **MAX** function as a result of the classifiers being more sure of the class of individual records.

IV. 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.