Support Vector Networks

Sarthak Gupta(2015CSB1029), Manas Gupta(2015CSB1018)
Third Year Undergraduates
B.Tech Computer Science and Engineering
IIT Ropar

Abstract

Support Vector Network is a learning machine for two group classification problems. In this project we tried to implement the paper "Support Vector Networks" [1]. Support Vector Network can efficiently implement linear and non-linear classification of the data. We were able to implement the quadratic optimization technique and perform classification on many two class datasets as discussed in the paper.

Introduction

The main task involved in this project was to implement the paper "Support Vector Machines" [1] through which we had to make a model which performed two class classification on linear and non-linear standard benchmark datasets and compare its efficiency against available toolkit solvers like LIBSVM.

The task of making a machine to learn to classify data has been an area of interest for many data scientists and statisticians. Many algorithms have been developed to solve this "classification" problem like Nearest Neighbors, Decision trees and Naïve Bayes' classifier and Support Vector Machine. Although we know that there is no single algorithm that is better than all others in all the problems. The efficiency of algorithm depends on the dataset used. But it can be observed that SVM perform good on most of the datasets provided we use the correct kernel and tune the parameters. So we can say that the implementation of SVM is one of the most important aspects that a person should know if he/she is trying to classify the data.

The main idea of our project is to implement a two class Support Vector Machine on non-linearly separable data. To get a decision boundary for non-linear data, the trick that we use is Kernels. Using a kernel, the input vectors are non-linearly mapped to a very high dimension feature space. In this feature space a linear decision boundary is constructed. SVMs aim at minimizing an upper bound on the generalization through maximizing the margin between the separating hyperplane and the data. After going through a lot of reading material we decided that since the data is non-linearly separable the we will not use the Linear Kernel but only test our implementation using the Gaussian Kernel. After some parameter tuning the Gaussian Kernel gave some good results. So finally we decided to implement the Gaussian Kernel as the only Kernel in our implementation of the Support Vector Machine.

Related Work

In the last few years, there has been a surge of interest in Support Vector Machines [1]. SVMs have empirically been shown to give good generalization performance on a wide variety of problems such as handwritten character recognition, face detection, pedestrian decision and text categorization [3]. The main problem in SVM is the problem of solving the quadratic optimization problem. The first approach to splitting large SVM learning problems into a series of smaller optimization tasks was proposed was known as the "Chunking Algorithm" [4]. The algorithm starts with a random subset of the data, solves this problem, and iteratively adds example which violate the optimality conditions. One disadvantage of this algorithm is that it is necessary to solve QP- problems scaling with the number of Support Vectors [5]. Another optimization technique was suggested in 1997 by Freund and Girosi [6]. The main technique of this algorithm is that a large Quadratic Programming problem can be broken down into a series of smaller Quadratic Programming sub-problems.

Then in 1998 John Platt invented Sequential Minimal Optimization [2] which was better than the previously available methods for SVM training. SMO did not require any third party Quadratic Programming solvers and it is much more simple then the previously available algorithms which were more complex and very hard to implement. It is an iterative coordinate ascent algorithm. In this algorithm we optimize with respect to the lagrange multipliers.

Model

Overview: The aim of our project is to implement Support Vector Machine on non-linearly separable data, which performs binary classification. Since the data is non-linearly separable so we need to use kernels in order to get good classification accuracy. The problem of Quadratic Optimization is solved through the implementation of Sequential Minimal Optimization technique. At every step of the optimization implementation the model chooses two Lagrange multipliers and jointly optimizes them. Once the model is learned using SMO, we can predict the class label of the validation set very easily.

SVM: A support vector machines are supervised learning models which are used for classification and regression. SVM is a non-probabilistic binary classifier. In its simplest form, linear form, an SVM is a hyperplane that separates a set of positive examples from a set of negative examples with the maximum margin. To find this hyperplane, as discussed in the class we minimize, $\frac{1}{2}\, ||\mathbf{w}||^2$ subject to the constraints $\mathbf{y_{i(w^T x_i + w0)}} >= 1$ for all data points (for all i). We will rewrite this optimization problem using Lagrange multipliers we will solve for the dual formulation using the Karush-Kuhn-Tucker conditions as were discussed in the machine learning lectures. The dual formulation gives

$\mathbf{L_d = max(\alpha_i) - \frac{1}{2} \sum\sum \alpha_i\, \alpha_j\, y_i\, y_j\, x_i^T x_j + \sum \alpha_i}$ subject to the conditions $\mathbf{\sum \alpha_i\, y_i \geq 0}$ **and** $\mathbf{\alpha_i \geq 0}$ **(**for all i**).** Time complexity of this formulation is $O(N^3)$. Also the size of the dual depends on the sample size.

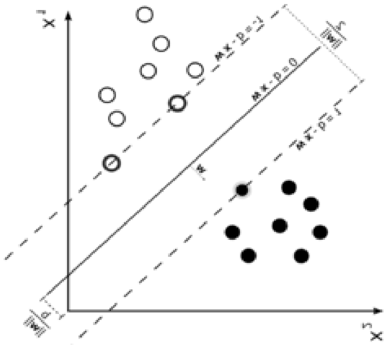

Figure 1: Hyperplane with maximum margin.

Kernels: All the formulation done above assumes the data is linearly separable. But if the data is not linearly separable in the current space perhaps if we map it to a high dimensional space, it may become linearly separable. The main function of the kernel is to map the data from a low dimensional space to a high dimensional space. Suppose the kernel maps the data like x -> $\phi(x)$ then to get equations of the dual formulation in this space all we need to do is replace all the instances of x in the previous equations with $\phi(x)$. This will give us the dual formulation in this space.

SMO: This is the optimization technique used to solve the quadratic optimization problem. We optimize the dual formulation using the Lagrange multipliers and KKT conditions. SMO chooses to solve the smallest possible optimization problem at every step. The smallest optimization problem here involves two Lagrange Multipliers, because they obey linearity equality constraint. We select two alphas to optimize the dual formulation with, while holding the other alphas constant. We repeat this step until convergence. As discussed in the lecture and implemented in the project the equations involved in the optimization steps and all the details of the algorithm are very nicely described in the paper by Platt [2]. After the convergence of the optimization problem the SMO will give a trained model which is used to classify the data.

Datasets: We used three data sets, all non-linearly separable**,** one of them we generated on our own the other two were taken from LIBSVM data repository [7]. We performed the experiments on all three datasets and got good results.

Results and Discussions

When we ran our implementation of the Support Vector Machine on the generated dataset we got a training accuracy of around of 96%. The dataset is shown above. We used

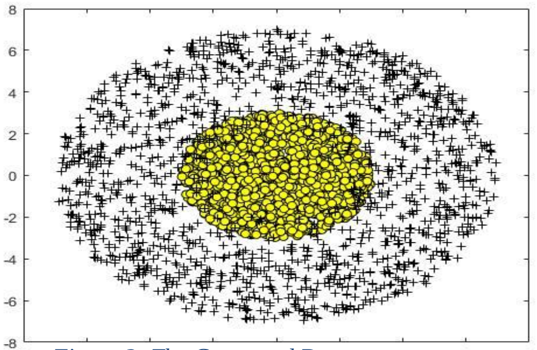

Figure 2: The Generated Dataset.

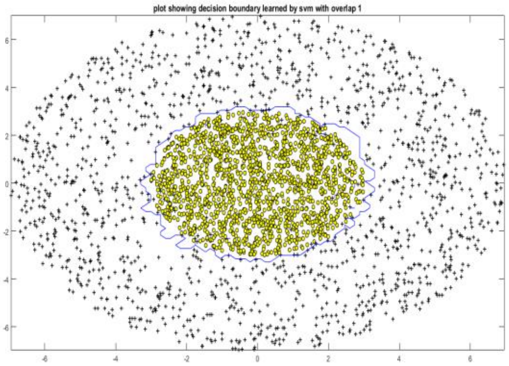

Figure 3: The Decision Boundary Learned.

the Gaussian kernel throughout and since we wanted a good accuracy on the training data set, we set the value of sigma to be 0.1 and the value of C to be 10, which gave a high variance model and gave a good accuracy.

We also plotted the decision boundary learned by our model. We can observe from the decision boundary that the model tries to over-fit the data to some extent.

On the dataset svmguide1 we plotted two graphs one shows the variation of accuracy on the training accuracy on training and test data with the box parameter C and the other one shows the variation of accuracy with sigma. As we can observe from both the graphs that as the value of sigma increase the bias of the model learned increases and the training accuracy decreases as the value of sigma increases. There is no visible change in the accuracy with the change in C.

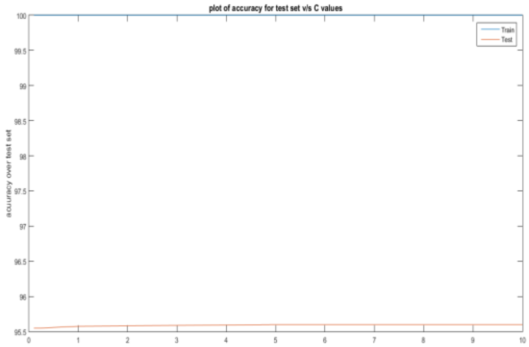

Figure 4: Variation with respect to C.

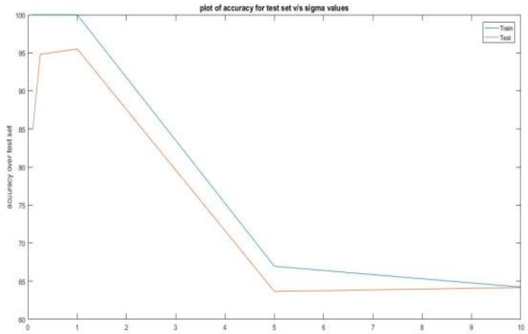

Figure 5: Variation with respect to sig-

On the dataset splice_scale we plotted the same two graphs that we did above and we got some good results. As we can observe that as the value of C increases the accuracy increase on both training and test data but afterwards it becomes constant as the model starts overfitting the data. In the second graph we can observe that as the value of sigma increases the model starts under-fitting the data and the accuracy on both the training and test data decreases. As we can see when the sigma increases to very high value we can see the accuracy becomes very low as the model learned neglects some important features of the data.

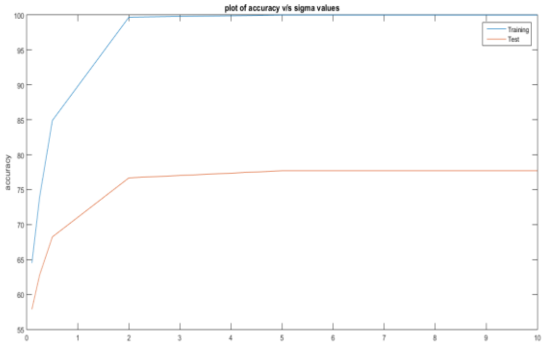

Figure 6: Variation with respect to C.

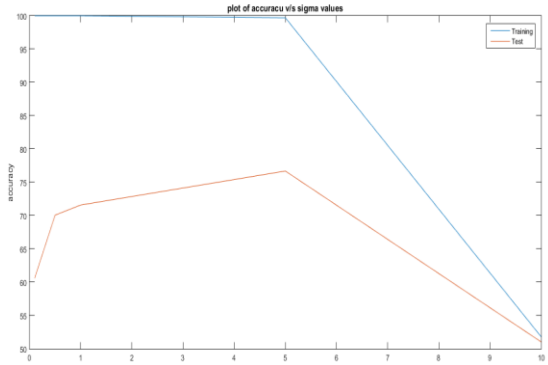

Figure 7: Variation with respect to sigma.

We can confirm that the observations in the variation of accuracy with respect to C and sigma are consistent with the theoretical observations and the text available.

Summary and Future Work

In the end we were able to implement support vector machines using sequential minimal optimization technique. The SVM was easily able to classify non linearly separable data using Gaussian Kernel with good accuracies if the parameters were given the right values. We successfully tested the implementation on three non-linearly separable datasets. SMO is one of the toughest algorithm to implement that we have come across but using the Platt paper as a guideline made the work very easy and efficient.

In future, we can take this implementation and try to make it classify a multi-class dataset. The other thing we can do is, we can extend the model to output probability of belonging to a class and apply an activation function for output.

References

[1] C Cortes V. Vapnik - Support Vector Networks 1995. *P*

[2] John C. Platt 1998. Sequential Minimal Optimization. *Technical Report MSR-TR-98-14.*

[3] Joachim, T 1997. Text Categorization with Support Vector Machine. *LS VIII Technical Report No. 23.*

[4] Boser, B.E.; Guyon, I. M.; Vapinak, V.N. 1992. A Training algorithm for optimal margin classifier. *Proceedings of the fifth annual workshop on Computational Learning Theory. COLT'92.*

[5]https://en.wikipedia.org/wiki/Sequential_minimal_optimization

[6] Osuna, E.Freud, R.; Girosi, F (1997). An improved training algorithm for support vector machines. *Proceedings of the 1997 IEEE workshop. Pp 276-285.*

[7] https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/