CS6011 Term Project - Twin SVM

Sanjay Ganapathy
Department of Computer Science
IIT Madras
CS12B027
cs12b027@smail.iitm.ac.in

Krishnan R
Department of Computer Science
IIT Madras
CS12B016
cs12b016@smail.iitm.ac.in

Abstract

We analyse Twin SVM, a binary SVM classifier that determines two nonparallel planes by solving two related SVM-type problems, each of which is smaller than in a conventional SVM. Predictions are made based on the distances of the point from both the planes. On the selected benchmark data sets, Twin SVM is not only fast, but shows good generalization.

1 Introduction

The basic aim in twin SVMs is to find two non-parallel hyperplanes, one for each class, such that the sum of distances of points of corresponding class data points to the hyperplane is minimised, while also imposing the condition that the distance of the hyperplane from the points of the other class is at least 1. The second constraint is imposed using slack variables to account for points which may be at a closer distance than 1.

As we had proposed in the preliminary report, we implemented twin SVM and classical SVM in matlab using the cvx solver for solving the dual convex optimization problem and tested the models on 4 different datasets as follows[1]

- Ionosphere with linear kernel
- BUPA-liver with rbf kernel
- Heart-c with linear kernel
- Synthentic 2-D linearly separable dataset with linear kernel
- Synthentic 2-D non-linearly separable dataset with rbf kernel

Of these, ionosphere and liver datasets were used for testing the performance of Twin SVM with respect to classical SVM and a non-kernel method, Gaussian Mixture Model[2]

The synthetic datasets were used to visualise the hyperplanes and decison boundaries for twin SVM.

Ionosphere and heart-c datasets with linear kernel, were used to measure the training times for twin SVM with respect to classical SVM.

2 Synthetic 2D Dataset

The position of support vectors for each plane has been marked and we find it is roughly the closest point of the other class which becomes a support vector.

[1] All datasets except synthetic are from UCI ML repository

[2] We had proposed to use neural networks in the preliminary report but found GMM gave better performance for all datasets

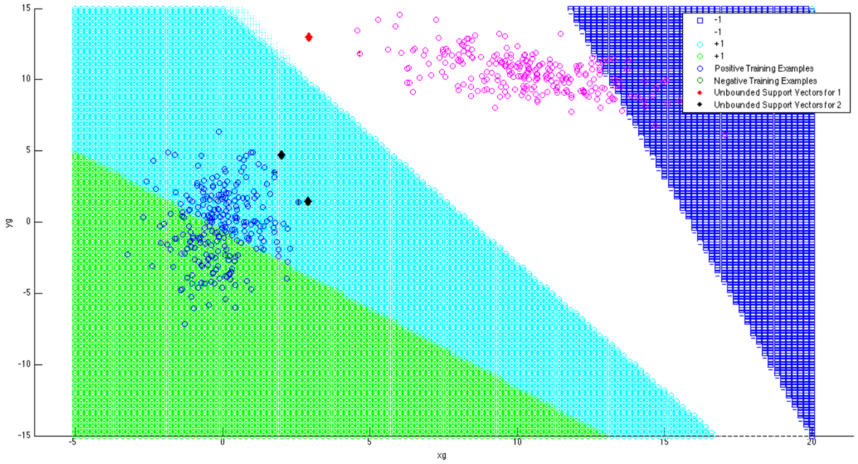

Figure 1: Twin SVM hyperplanes and decision boundary for linearly separable data

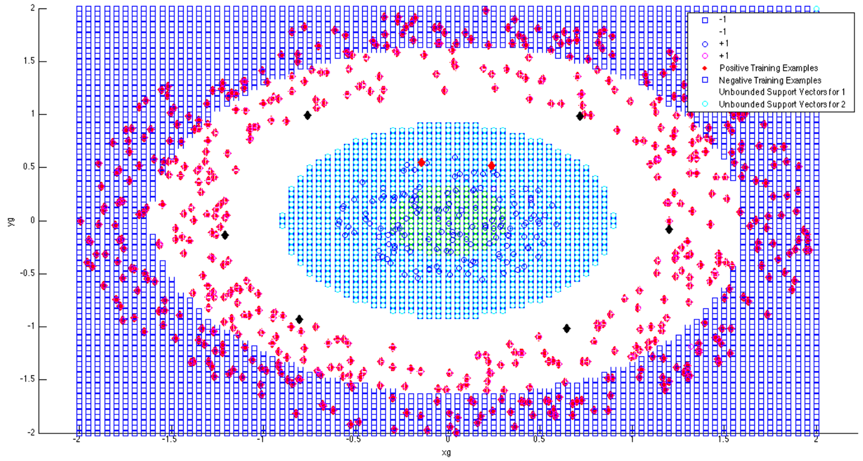

Figure 2: Twin SVM hyperplanes and decision boundary for non-linearly separable data

Table 1: Estimated Parameters for TSVM

Dataset(Kernel)	**Cost1**	**Cost2**	$\epsilon 1$	$\epsilon 2$
Ionosphere (Linear)	5	5	0.1	0.1
BUPA-liver (RBF)	100	100	10	10

Table 2: Estimated Parameters for SVM

Dataset(Kernel)	**Cost**
Ionosphere (Linear)	3
BUPA-liver (RBF)	100

3 Parameter Estimation

Parameter estimation was done by selecting 15% of the training data as a validation set for fine tuning the parameters instead of the 10% used in the paper. The main parameters to be estimated for twin SVM were Cost1, Cost2, $\epsilon 1$, $\epsilon 2$ and s in case of rbf kernel. The ϵ parameter is needed while calculating the inverse of a matrix, which is not guaranteed to be singular and hence we require a $\lambda * I$ term to be added. This parameter however does not greatly affect performance and is only required to ensure that the inverse exists. The main parameters are the costs and s. See Table 1 and Table 2. The s parameter for rbf kernel was found to be 10000.

4 Performance Comparison

The performance of all models reported in Table 3 were measured by standard 10 fold cross validation done on the training data, as used in the paper as well, using the parameters estimated as above.

4.1 Liver dataset with gaussian kernel

Primarily we find that the performance as reported by using the best parameters for twin SVM is almost identical to what is reported in the paper. However, we find that the best performance of classical SVM is actually significantly better than what is reported in the paper and marginally better than twin SVM.

We also find that GMM performs almost similar to TSVM in Liver dataset. We can see in figure 3 a projection of this dataset and TSVM's performance.

4.2 Ionosphere dataset with linear kernel

We find here that the best results obtained for twin SVM falls slightly short of what is reported in the paper, though this may be because the parameters require more fine-tuning or the split used by the paper for cross validation may have been different. But, we find that twin SVM is able to perform marginally better than classical SVM, whose performance is almost identical to the performance reported in the paper. Here, GMM with 3 gaussians for each class performs better than SVMs. For finding the accuracy with GMM, we remove the first 2 features as the fitgmdist function showed an error due to these features being constant in one of the classes. We can see in figure 4 a projection of this dataset and TSVM's performance.

5 Training Time Comparison

The training time comparison in the paper was done by using svm and twin svm as a exe file and a dll(dynamically linked library). For the purpose of this project we decided to measure the times

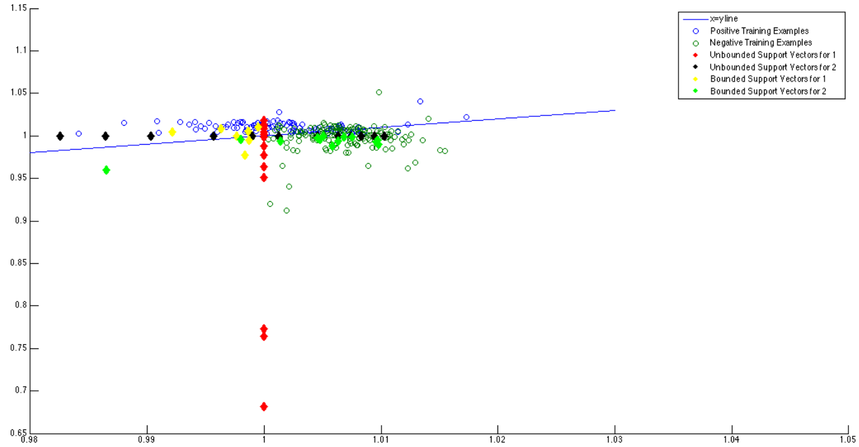

Figure 3: Points represented as 2D coordinates where each coordinate is the distance from one of the hyperplanes (Liver Dataset)

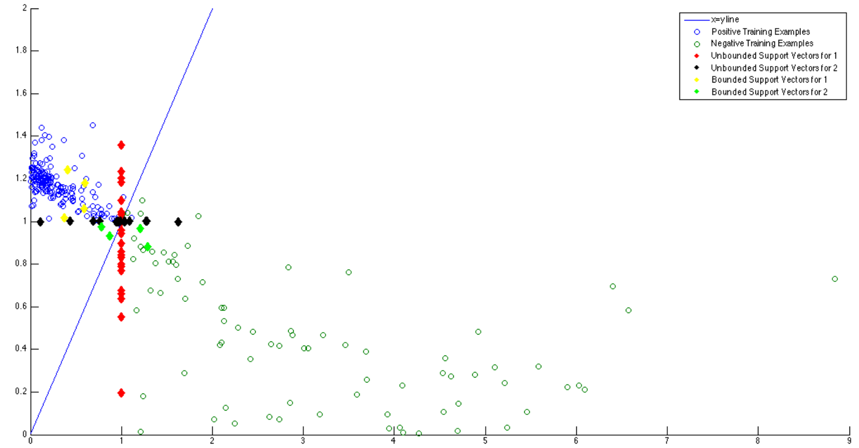

Figure 4: Points represented as 2D coordinates where each coordinate is the distance from one of the hyperplanes (Ionosphere Dataset)

Table 3: Test Set Accuracy

Dataset(Kernel)	TSVM	SVM	TSVM (paper)	SVM (paper)	GMM
Ionosphere (Linear)	86.83 ± 5.26	85.80 ± 8.44	88.03 ± 2.81	86.04 ± 2.37	91.24 ± 4.89
BUPA-liver (RBF)	67.55 ± 7.08	72.47 ± 4.17	67.83 ± 6.49	58.32 ± 8.20	68.43 ± 4.40

Table 4: Support Vector for Liver Dataset

Model	Bounded Support Vectors	Unbounded Support Vectors
SVM	152	66
TSVM positive plane	7	17
TSVM negative plane	13	13

Table 5: Support Vector for Ionosphere Dataset

Model	Bounded Support Vectors	Unbounded Support Vectors
SVM	56	29
TSVM positive plane	4	23
TSVM negative plane	4	16

using just our matlab implementations, which we found were comparable with the training times for exe file implementation of Tsvm. We have recorded out measurements of 2 different times - one is the time required by the solver to solve the dual optimization problem for tsvm and svm, which is supposed to be about four times less for tsvm. The other is the overall time required to 10 fold cross validation, which includes computing the gram matrix and setting up the parameters needed for the optimization. We found that in our matlab implementation, this time was comparable to the time required to solve the optimization problem, and this time is nearly the same for both SVM and twin SVM. So, the overall training time is skewed by this time.

Time for solving dual optimization in Table 6.

Table 6: Dual Optimization Solving Time

Dataset	TSVM	SVM
Ionosphere	2.2	8.6
Heart-c	1.9	6.9

Training time for 10 fold cross validation in Table 7.

Table 7: 10 fold cross validation Training Time

Dataset	TSVM	SVM	TSVM exe
Ionosphere	7.93	12.02	9.93
Heart-c	7.07	9.79	8.37

6 Concluding Remarks

From our studies of the twin SVM in this project, we find that :

- Twin SVM test set performance is very similar to classical SVM for the datasets analysed, which are in fact the datasets in which twin SVM gives significant improvement in performance as reported in the paper. However, its performance is only upper bounded by GMM's performance in the 2 datasets where we tested GMM.
- Twin SVM can also be used to obtain a 2D projection of the input dataset by taking the distance from the two hyperplanes to be the two components. Since our decision rule is based on which is greater, we are likely to obtain a linearly non-separable projection separated by the line y = x.
- Twin SVM does in fact take lesser time to train than classical SVM. The formulation of TWSVM is also attractive for handling preferential classification problems. A classifier may be obtained very rapidly by solving the smaller problem. This is particularly interesting for unbalanced data sets, e.g., in medical databases, where the number of disease-free examples may far out-number instances of the other class. We can solve the TSVM objective function for the plane corresponding to the minority class and use a plane passing through the unbounded support vectors as the decision boundary. This way we try to make sure that we classify the minority class with high precision.
- Since Twin SVM uses kernels, like normal SVM, this gives an option of a wide range of complexities to choose from.

References

[1] S.R. Gunn, *Support Vector Machines for Classification and Regression*, technical report, School of Electronics and Computer Science, Univ. of Southampton, Southampton, U.K., 1998, http://www.isis.ecs.soton.ac.uk/resources/svminfo/.

[2] K. Jayadeva, R. Khemchandani and S. Chandra *Twin support vector machines for pattern classification*, IEEE Trans. Pattern Anal. Mach. Intell., vol. 29, no. 5, pp.905 -910 2007