B.Tech Project

Semester-VI

Robust Transductive Support Vector Machines (using stochastic gradient and dual form)

Khushi Agrawal (17075060) and Janhavi Gupta (17075061)

Under the guidance of: Prof. KK Shukla

Introduction

Obtaining labels of data is often a costly and error-prone procedure as it often requires human work. To this end, we worked towards exploring semi-supervised learning which combines a small amount of labelled data with a large amount of unlabelled data during training. The classical Support Vector Machines (SVM) work only towards classifying the test data with respect to the labelled data samples which were made available during training. However, Transductive Support Vector Machines (TSVM) take into consideration the unlabelled samples which were present in the training data set and thus lead to better results. Our main work involved a study of the application of TSVM to large-scale data and working towards improving the robustness so as to achieve better results even with datasets having noise.

Literature study

For the study of transductive support vector machines on large scale data, we studied the paper mentioned here [1] which deals with TSVM applied to large-scale data and also describes an algorithm for making them robust using the stochastic gradient method. The paper employs CCCP (Concave-Convex Procedure) to solve the non-convex optimization problem which decomposes the non-convex function into a convex and concave part and converges iteratively to the optimal value. The paper suggested replacement of the classical Hinge Loss function ($H_1(t)=max(0,1-t)$) with a more robust non-convex function, Ramp Loss ($R_s(t)=H_1(t)-H_s(t)$) for the labelled samples as it is less sensitive to outliers thereby providing robustness.

The paper discusses various algorithms for the final computation of the optimization problem and then employs a Stochastic Gradient-Based Solver to achieve robustness for large scale data. It further compares its results with other methods (such as primal and dual form) to show that this method performs comparative to the other methods for clean data but significantly outperforms others when noise is added progressively.

Classifiers studied

- *Support vector machines (SVMs)*

The support vector machine classifier finds a hyperplane to separate the data points into two classes with a large margin. It maximises the margin between the data-points and the hyperplane.

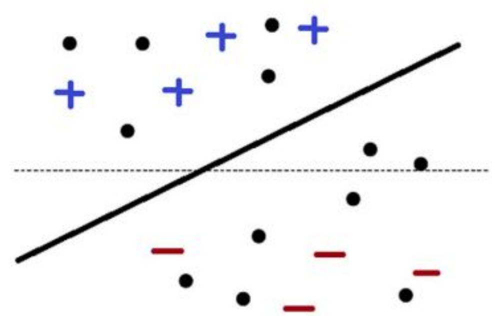
$$\arg\min_{\mathbf{w},b} \frac{1}{2} ||\mathbf{w}||^2 + C \sum_{i=1}^{L} \xi_i \cdot$$

$$\text{s. t. } y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 - \xi_i,$$

- *Transductive Support Vector Machines (TSVMs)*

  The aim of transductive support vector machines is to find a hyperplane to separate the data points into two classes as well as to classify the unlabelled samples, i.e. unlabelled samples should also be at a large margin from the hyperplane.

$$\arg\min_{\mathbf{w},b} \frac{1}{2} ||\mathbf{w}||^2 + C \sum_{i=1}^{L} \xi_i + C^* \sum_{i=L+1}^{L+U} \xi_i \text{s. t. } y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 - \xi_i,$$
$$i = 1, \ldots, L, |\mathbf{w}^\top \mathbf{x}_i + b| \geq 1 - \xi_i, \;\; i = L + 1, \ldots, L + U.$$



The figure above illustrates the difference between SVM and TSVM. The dotted line represents the hyperplane by SVM classifier, which divides the labelled samples well. The solid line represents the hyperplane by TSVM classifier, which also considers the unlabelled samples while classifying the data, yielding better results.

The TSVM proposed in the paper solves the following optimization problem:

$$\arg\min_{\mathbf{w},b} \frac{1}{2} ||\mathbf{w}||^2 + C \sum_{i=1}^{L} R_s(y_i(\mathbf{w}^\top \mathbf{x}_i + b)) + C^* \sum_{i=L+1}^{L+U} SR_s(\mathbf{w}^\top \mathbf{x}_i + b)$$
$$\text{s. t. } \frac{1}{U} \sum_{i=L+1}^{L+U} (\mathbf{w}^\top \mathbf{x}_i + b) = \frac{1}{L} \sum_{i=1}^{L} y_i.$$

We used the cvxopt.solvers package of python to solve for the dual form. For the stochastic gradient method, we employed the formula described in the paper which converged iteratively to yield the optimal value. The code for both methods can be found here.

Experiments and results

We tested our code against multiple datasets for a comparative analysis of our robust TSVM. Further, we also added noise incrementally by switching the labels of samples in the training data set.

The following tables illustrate our results:
Each column represents the percentage of data samples for which we switched labels, i.e., the percentage of noise added, while each row depicts the result of each classifier for various noise percentages.

- *Aust.csv*

	0%	**10%**	**20%**	**30%**	**40%**	**50%**
SVM	82.41	79.31	76.21	79.66	78.62	78.62
TSVM (dual form)	63.45	64.14	59.66	44.14	63.45	63.79
Robust TSVM (sgd)	84.14	85.17	80.35	84.48	81.72	85.52

- *Bupa1.csv*

	0%	**10%**	**20%**	**30%**	**40%**	**50%**
SVM	71.72	61.38	63.45	62.07	61.38	66.21
TSVM (dual form)	60.0	60.69	43.45	53.10	57.93	60.0
Robust TSVM (sgd)	64.83	62.76	63.45	60.69	62.07	57.24

- *Pima1.csv*

	0%	**10%**	**20%**	**30%**	**40%**	**50%**
SVM	80.97	80.22	80.60	77.61	79.10	73.51
TSVM (dual form)	66.79	65.30	62.31	49.63	56.72	35.45

Robust TSVM (sgd)	75.0	72.02	66.79	66.79	67.91	64.18

- *Fertility.csv*

(size of the dataset quite small, only 80 training examples and 20 testing examples)

	0%	**10%**	**20%**	**30%**	**40%**	**50%**
SVM	90.0	90.0	90.0	90.0	90.0	65.0
TSVM (dual form)	90.0	90.0	90.0	90.0	90.0	90.0
Robust TSVM (sgd)	90.0	90.0	90.0	90.0	90.0	65.0

- *Cleve.csv*

	0%	**10%**	**20%**	**30%**	**40%**	**50%**
SVM	80.58	87.38	76.70	76.70	76.70	67.0
TSVM (dual form)	58.25	58.25	57.28	55.34	49.51	58.25
Robust TSVM (sgd)	78.64	87.38	76.70	71.85	76.70	72.81

Future work

We plan to further add robustness to our TSVM. To this end, we are exploring various other robust loss functions such as rescaled hinge loss function [2], pinball loss function [3], huber loss function, etc.

References

[1] Hakan Cevikalp, Vojtech Franc, Large-scale robust transductive support vector machines

[2] Guibiao Xu, Zheng Cao, Bao-Gang Hu, Jose C. Principe, Robust support vector machines based on the rescaled hinge loss function

[3] Xin Shen, Lingfeng Niu, Zinquan Qi, Yingijie Tian, Support vector machine classifier with truncated pinball loss