Robust Multi-Task Least Squares Twin Support Vector Machines for Classification

Reshma Rastogi[1] and Mustaffa Hussain[2]

[1] South Asian University, New Delhi 110021, India,
reshma.khemchandani@sau.ac.in,
[2] South Asian University,New Delhi 110021,India
mustaffahussain4734@gmail.com

Abstract. Lately, there has been a shift to Multi-Task learning. Multi-task learning performs better than the classical single task learning by learning from the training signals inherent in all the tasks. Inspired by Multi-task least squares twin support vector machine, we propose a Robust Multi task least squares twin support vector machine. In the proposed work we introduce an error factor which successfully handles the noise. The proposed model is easy to implement and fast. This allows the model to be of direct application to larger and real-world data sets. In addition, the model deals with nonlinear data patterns by using kernel trick.

Keywords: Multi-task, Twin support vector machine, Robust

1 Introduction

Multi-task learning(MTL) is also known as inductive transfer or inductive bias learning [4]. In contrast to the classical single task learning, multi-task learning aims to improve the generalisation accuracy by leveraging the task specific information contained in the training signals of the other tasks. It works on the idea that similar task share similar properties and structures which can be helpful for the other tasks in hand. Given the dynamacity of the changing problems, it also works as method of data augmentation.

Multi-task learning was well established and widely applied in deep learning and reinforcement learning. Some of benchmark work for attribute prediction and structure learning of tasks are [8],[1]. Inspired from the convex multi task learning idea, many researchers worked upon and proposed several multi-task learning methods [2],[7] tries discussing multi-task feature learning. Authors of [22],[12] discusses learning multi-task relationships in detail. Multi-task clustering was discussed in [16]. A comprehensive study on multi-task learning was presented in [25].

SVMs have been a huge success given the mathematical understanding and wide applicability. Given the success in SVMs, [5] introduced the idea of regularized multi-task learning, where the main assumption was, all the tasks share a common representation. One-class classification was extended to multi-task learning

framework in [24]. This idea has been discussed in detail by researchers in [23],[9] and [24]. Fung and Mangasarian [6] proposed the idea of proximal support vector machines. Proximal SVM was extended to multi task proximal support vector machines in[13]. Utilising the advantage of multi-task learning, multi-task asymmetric least square support vector machines (MTL-aLS-SVM) was proposed in [14].
In particular most of the work belonging to multi-task learning in SVM was done in regularised framework, where the main idea was all tasks share a common representation. Few other approaches in SVMs are discussed in [18] & [8] which are graph and tree based Multi-task learning approaches.
However there have not been much work in Twin support vector machine on Multi-task learning. TWSVM was first proposed in [10] by Jayadeva et.al. A comprehensive study of TWSVM was presented in [17]. The main idea of TWSVM was to use two non parallel hyperplanes to identify samples belonging to two different classes. Authors of [3] proposed least square TWSVM for Multi-label learning and [19] explores multi-task learning in TWSVM using pinball loss. Inspired from TWSVM and Multi-task learning, Multi-task twin support vector machine (DMTSVM) was proposed in [20]. DMTSVM uses the regularized method. In order to deal with the short coming of DMTSVM, multi-task centroid twin support vector machine (MCTSVM) was put forth in [21]. It successfully handles the outliers in all the tasks. Inspired from these models and the efficiency of least squares twin support vector machines[11]; multi-task least squares twin support vector machines was proposed. MT-LS-TWSVM proposed in [15] greatly exploits the advantages of least squares solutions and improves upon the efficiency. From the above work and on the lines of regularized framework we propose Regularised Multi-task least squares twin support vector machines. The contributions of our method are as follows:
(i) We propose a novel Robust MT-LS-TSVM which takes over the merits of Multi-task learning and improves the classification accuracy.
(ii) The proposed model solves a pair of small linear equations. Thus, it is faster in comparison to DMTSVM, MCTSVM.
(iii) The calculation of decision hyperplanes involves a single matrix inversions , which makes it fast and easy to implement.
(iv) it successfully handles multi-task noise present in all the tasks.

The rest of the paper is organised as; section(2) presents the primal of DMTSVM and MT-LS-TWSVM. Our proposed model introduced in section(3). The formulation of our multi-task model is presented in section(3.1). In section(3.2) we extend the model to deal with nonlinear patterns. Section(4) deals with the optimising the computational complexity of the model. Section(5) describes the experiments and the data sets used. The conclusion and future work comprises section(6).

2 Related Work

In this section, we present an overview of DMTSVM and MT-LS-TSVM. These algorithms provide a base for our proposed method.

2.1 Multi-Task twin support vector machine

Multi-task twin support vector machine was proposed in [20]. It carries on the idea of regularised multi-task learning into the twin support vector machines. Suppose given a data set D with n samples of m dimensional real space R^m. Each sample point relates to binary output $y \in \{-1, 1\}$. Let all the samples belonging to positive class from 't'-th task be represented by X_{pt} and all the samples belonging to negative class from 't'-th task be represented as X_{nt}. Let all the positive samples be represented as X_p and all the samples belonging to negative class be represented as X_n. From the above representations, we introduce some notations
$A_t = [X_{pt}, e_t],\ A = [X_p, e],\ B_t = [X_{nt}, e_t],\ B = [X_n, e]$,
where e_t and e are one vectors of appropriate dimensions.
All the task share two mean hyperplanes, $u = [w_1\ b_1]'$ and $v = [w_2\ b_2]'$. Then the positive hyperplane belonging to the 't'-th task can be represented as $[w_{1t}\ b_{1t}]' = (u + u_t)$. The negative hyperplane corresponding to t-th task can be represented as $[w_{2t}\ b_{2t}]' = (v + v_t)$. Here u_t and v_t are the biases from the different tasks. The primal of the DMTSVM can be written as

$$\begin{aligned} \underset{u,u_t,\xi_{1t}}{\text{minimize}} \quad & \frac{1}{2} \parallel Au \parallel_2^2 + \frac{1}{2}\sum_{t=1}^{T} \rho_t \parallel A_t u_t \parallel_2^2 + c_1 \sum_{t=1}^{T} e_t' \xi_{1t} \\ \text{subject to} \quad & \forall t : -B_t(u + u_t) + |xi_{1t} \geq e_{2t}, \ \ \xi_{1t} \geq 0, \end{aligned} \tag{1}$$

and

$$\begin{aligned} \underset{v,v_t,\xi_{2t}}{\text{minimize}} \quad & \frac{1}{2} \parallel Bv \parallel_2^2 + \frac{1}{2}\sum_{t=1}^{T} \lambda_t \parallel B_t v_t \parallel_2^2 + c_2 \sum_{t=1}^{T} e_t' \xi_{2t} \\ \text{subject to} \quad & \forall t : A_t(v + v_t) + \xi_{1t} \geq e_{1t}, \ \ \xi_{2t} \geq 0, \end{aligned} \tag{2}$$

where c_1, c_2, p_t,and λ_t are positive trade-off parameters, $e_{1t},\ e_{2t}$are one vectors of appropriate dimension. p_t and q_t are the slack variables. If $\xi_{1t} \ \rightarrow 0$ and $\xi_{2t} \ \rightarrow 0$ then all the tasks will be learnt unrelated. The class label of any new point x is determined by

$$f(x) = arg \min_{r=1,2} \mid {W_{rt}}'.x + b_{rt} \mid, \tag{3}$$

2.2 Multi-Task least squares twin support vector machine

The definition of A, A_t, B, B_t are the same as used in the section 2.1. Then the primal of MTLS-TWSVM can be written as

$$\begin{aligned} \underset{u,u_t,\xi_{1t}}{\text{minimize}} \quad & \frac{1}{2} \parallel Au \parallel_2^2 + \frac{1}{2}\rho \sum_{t=1}^{T} \parallel A_t u_t \parallel_2^2 + c_1 \sum_{t=1}^{T} \xi_{1t}^T \xi_{1t} \\ \text{subject to} \quad & \forall t : -B_t(u+u_t) + \xi_{1t} = e_{2t}, \end{aligned} \tag{4}$$

and

$$\begin{aligned} \underset{v,v_t,\xi_{2t}}{\text{minimize}} \quad & \frac{1}{2} \parallel Bv \parallel_2^2 + \frac{1}{2}\lambda \sum_{t=1}^{T} \parallel B_t v_t \parallel_2^2 + c_2 \sum_{t=1}^{T} \xi_{2t}^T \xi_{2t} \\ \text{subject to} \quad & \forall t : A_t(v+v_t) + \xi_{2t} = e_{1t}, \end{aligned} \tag{5}$$

where c_1, c_2, ρ, λ are non negative parameters and e_{1t},e_{2t} are one vectors of appropriate dimensions. Each task is given equal role by ρ and λ.Larger values of ρ and λ result is smaller u_t and v_t. This leads to T similar learnt models. Meanwhile smaller values of ρ and λ would lead to difference among tasks to be large. The decision function for any new sample point x from the t-th task is determined by

$$f(x) = arg \min_{r=1,2} \mid {W_{rt}}'.x + b_{rt} \mid, \tag{6}$$

3 Robust Multi-task least squares twin support vector machines

We work on the lines of Regularised Multi-Task learning framework. The model assumes that all the tasks share a common representation. As learning is based on all the tasks, there is a similarity in the samples of classes from all the tasks. As data of each task differs on the basis of origin, multi- task learning acts as a very good setup for noisy data. More over multi-task learning involves errors specific to different tasks. On this basis we introduce Robust multi-task least squares twin support vector machine (R-MT-LS-TWSVM) to tackle the noise involved.

3.1 Linear robust multi-Task least squares twin support vector machine

We carry on all the notations used in the section(2.1).The proposed model can be written as

$$\begin{aligned} \underset{u,u_t,\delta_1,\xi_1}{\text{minimize}} \quad & \frac{1}{2} \parallel Au \parallel_2^2 + \frac{1}{2}\rho \sum_{t=1}^{T} \parallel A_t u_t \parallel_2^2 + c_1 \parallel \delta_1 \parallel_2^2 + c_2 \xi_1' \xi_1 \\ \text{subject to} \quad & \forall t : -B_t(u+u_t) = e_{2t}(1-\delta_1) - \xi_1, \end{aligned} \tag{7}$$

and

$$\begin{aligned} \underset{v,v_t,\delta_2,\xi_2}{\text{minimize}} \quad & \frac{1}{2} \parallel Bv \parallel_2^2 + \frac{1}{2}\lambda \sum_{t=1}^{T} \parallel B_t v_t \parallel_2^2 + c_3 \parallel \delta_2 \parallel_2^2 + c_4 \xi_2' \xi_2 \\ \text{subject to} \quad & \forall t : A_t(v + v_t) = e_{1t}(1 - \delta_2) - \xi_2, \end{aligned} \tag{8}$$

where $c_1, c_2, c_3, c_4, \lambda, \rho, \delta_1, \delta_2$ are positive trade-off parameters and ξ_1 and ξ_2 are error variable; e_{1t} and e_{2t} are vectors of one of appropriate dimensions.

Substituting the value of ξ_1 from the constraint into the objective function(7), results in the following unconstrained optimisation problem:

$$\underset{u,u_t,\delta_1,\xi_1}{\text{minimize}} \quad F = \frac{1}{2} \parallel Au \parallel_2^2 + \frac{1}{2}\rho \sum_{t=1}^{T} \parallel A_t u_t \parallel_2^2 + c_1 \parallel \delta_1 \parallel_2^2 + c_2 \sum_{t=1}^{T} \parallel B_t(u + u_t) + (1 - \delta_1)e_{2t} \parallel_2 \tag{9}$$

Taking gradient of (9) and setting them to zero with respect to u, u_t, δ_1 yields the following equations,

$$\frac{\partial F}{\partial u} = A'Au + \sum_{t=1}^{T} c_2 B_t'(B_t(u + u_t) + (1 - \delta_1)e_{2t}), \tag{10}$$

$$\frac{\partial F}{\partial u_t} = \frac{\rho}{2T} A_t' A_t u_t + c_2 B_t'(B_t(u + u_t) + (1 - \delta_1)e_{2t}), \tag{11}$$

$$\frac{\partial F}{\partial \delta_1} = c_1 \delta_1 - c_2 e_{2t}' \sum_{t=1}^{T} (B_t(u + u_t) + (1 - \delta_1)e_{2t}), \tag{12}$$

rearranging these equations and solving for u, u_t, δ_1 gives
we define :

$$P = A'A + \sum_{t=1}^{T} c_2 B_t' B_t, \quad S = -c_2 \sum_{t=1}^{T} B_t' e_{2t}, \quad K = -c_2 \sum_{t=1}^{T} e_{2t}' B_t, \quad G = c_1 + c_2 e_{2t}' e_{2t},$$

$$Q_t = c_2 B_t' B_t, \qquad R_t = \frac{\rho}{2T} A_t' A_t + c_2 B_t B_t', \qquad J_t = -c_2 B_t' e_{2t}, \qquad L_t = -c_2 e_{2t}' B_t;$$

where $t \in \{1, 2, 3, ..., T\}$ *represents the task number..*

$$\begin{bmatrix} P & Q_1 & Q_2 & Q_3 & \dots & Q_T & S \\ Q_1 & R_1 & 0 & 0 & \dots & 0 & J_1 \\ Q_2 & 0 & R_2 & 0 & \dots & 0 & J_2 \\ Q_3 & 0 & 0 & R_3 & \dots & 0 & J_3 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ Q_T & 0 & 0 & 0 & \dots & R_T & J_T \\ K & L_1 & L_2 & L_3 & \dots & L_T & G \end{bmatrix} \begin{bmatrix} u \\ u_1 \\ u_2 \\ u_3 \\ \dots \\ u_T \\ \delta_1 \end{bmatrix} = \begin{bmatrix} -\sum_{t=1}^{T} c_2 B_t' e_{2t} \\ -c_2 B_1' e_{21} \\ -c_2 B_2' e_{22} \\ -c_2 B_3' e_{23} \\ \dots \\ -c_2 B_T' e_{21} \\ c_2 e_2' e_2 \end{bmatrix} \tag{13}$$

which leads to the solution

$$\begin{bmatrix} u \\ u_1 \\ u_2 \\ u_3 \\ \dots \\ u_T \\ \delta_1 \end{bmatrix} = \begin{bmatrix} P & Q_1 & Q_2 & Q_3 & \dots & Q_T & S \\ Q_1 & R_1 & 0 & 0 & \dots & 0 & J_1 \\ Q_2 & 0 & R_2 & 0 & \dots & 0 & J_2 \\ Q_3 & 0 & 0 & R_3 & \dots & 0 & J_3 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ Q_T & 0 & 0 & 0 & \dots & R_T & J_T \\ K & L_1 & L_2 & L_3 & \dots & L_T & G \end{bmatrix}^{-1} \begin{bmatrix} -\sum_{t=1}^{T} c_2 B_t' e_{2t} \\ -c_2 B_1' e_{21} \\ -c_2 B_2' e_{22} \\ -c_2 B_3' e_{23} \\ \dots \\ -c_2 B_T' e_{2T} \\ c_2 e_2' e_2 \end{bmatrix} \tag{14}$$

The positive classifier parameters u and u_t are determined. Similarly from equation(8) for the negative hyperplane parameters , the solution can be obtained from
we define :

$$P = B'B + \sum_{t=1}^{T} c_4 A_t' A_t, \quad S = c_4 \sum_{t=1}^{T} A_t' e_{2t}, \quad K = c_4 \sum_{t=1}^{T} e_{2t}' A_t, \quad G = c_3 + c_4 e_{2t}' e_{2t},$$

$$Q_t = c_4 A_t' A_t, \qquad R_t = \frac{\rho}{2T} B_t' B_t + c_4 A_t A_t', \qquad J_t = c_4 A_t' e_{2t}, \qquad L_t = c_4 e_{2t}' A_t;$$

$$where \;\; t \in \{1, 2, 3, ..., T\} \;\; represents \;\; the \;\; task \;\; number..$$

$$\begin{bmatrix} P & Q_1 & Q_2 & Q_3 & \dots & Q_T & S \\ Q_1 & R_1 & 0 & 0 & \dots & 0 & J_1 \\ Q_2 & 0 & R_2 & 0 & \dots & 0 & J_2 \\ Q_3 & 0 & 0 & R_3 & \dots & 0 & J_3 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ Q_T & 0 & 0 & 0 & \dots & R_T & J_T \\ K & L_1 & L_2 & L_3 & \dots & L_T & G \end{bmatrix} \begin{bmatrix} v \\ v_1 \\ v_2 \\ v_3 \\ \dots \\ v_T \\ \delta_2 \end{bmatrix} = \begin{bmatrix} \sum_{t=1}^{T} c_4 A_t' e_{1t} \\ c_4 A_1' e_{11} \\ c_4 A_2' e_{12} \\ c_4 A_3' e_{13} \\ \dots \\ c_4 A_T' e_{1T} \\ c_4 e_1' e_1 \end{bmatrix} \tag{15}$$

which leads to the solution

$$\begin{bmatrix} v \\ v_1 \\ v_2 \\ v_3 \\ \dots \\ v_T \\ \delta_2 \end{bmatrix} = \begin{bmatrix} P & Q_1 & Q_2 & Q_3 & \dots & Q_T & S \\ Q_1 & R_1 & 0 & 0 & \dots & 0 & J_1 \\ Q_2 & 0 & R_2 & 0 & \dots & 0 & J_2 \\ Q_3 & 0 & 0 & R_3 & \dots & 0 & J_3 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ Q_T & 0 & 0 & 0 & \dots & R_T & J_T \\ K & L_1 & L_2 & L_3 & \dots & L_T & G \end{bmatrix}^{-1} \begin{bmatrix} \sum_{t=1}^{T} c_4 A_t' e_{1t} \\ c_4 A_1' e_{11} \\ c_4 A_2' e_{12} \\ c_4 A_3' e_{13} \\ \dots \\ c_4 A_T' e_{1T} \\ c_4 e_1' e_1 \end{bmatrix} \tag{16}$$

From the above equation v and v_t's for all the tasks are obtained. The label for a new sample point x from the t-th task is decided by :

$$f(x) = arg \min_{r=1,2} \mid {W_{rt}}'.x + b_{rt} \mid, \tag{17}$$

3.2 Nonlinear robust multi-Task least squares twin support vector machine

It is very much logical that a linear classifier may not be able to learn well from samples that are linearly inseparable. In such a scenario, kernel trick is used. We define the kernel as :
$M = [K(A, Z'), e], \quad M_t = [K(A_t, Z'), e_t],$
$N = [K(B, Z'), e], \quad N_t = [K(B_t, Z'), e_t],$

$K(.)$ stands for the specific kernel function used. Z is defined as all the training samples from all the tasks. Then the kernelized version of R-MT-LS-TWSVM can be written as :

$$\begin{aligned} \underset{u,u_t,\delta_1}{\text{minimize}} \quad & \frac{1}{2} \parallel Mu \parallel_2^2 + \frac{1}{2}\rho \sum_{t=1}^{T} \parallel M_t u_t \parallel_2^2 + c_1 \parallel \delta_1 \parallel_2^2 + c_2 \xi_1' \xi_1 \\ \text{subject to} \quad & \forall t : -N_t(u + u_t) = e_{2t}(1 - \delta_1) - \xi_1, \end{aligned} \tag{18}$$

and

$$\begin{aligned} \underset{v,v_t,\delta_2}{\text{minimize}} \quad & \frac{1}{2} \parallel Nv \parallel_2^2 + \frac{1}{2}\lambda \sum_{t=1}^{T} \parallel N_t v_t \parallel_2^2 + c_3 \parallel \delta_2 \parallel_2^2 + c_4 \xi_2' \xi_2 \\ \text{subject to} \quad & \forall t : M_t(v + v_t) = e_{1t}(1 - \delta_2) - \xi_2, \end{aligned} \tag{19}$$

where $c_1, c_1, c_3, c_3, \delta_1, \delta_2$ are non-negative parameters. ξ is the error variable and e_{1t}, e_{2t} are ones of appropriate dimensions. The decision function for any new data point x from the 't-th' task is

$$f(x) = arg \min_{r=1,2} \mid {W_{rt}}'.K(x, Z') + b_{rt} \mid, \tag{20}$$

4 Computational Complexity

We can note from equation(14) and (16) in the proposed model, the hyperplane parameters are dependent on matrix inversion. Matrix inversion is a computationally heavy operation dependent on the dimension of matrix. When the size and dimension of data set is of large magnitude, it directly affects the speed of the proposed model.

Here we use the partition method of matrix inversion to speed up the process. The general formula for a matrix A is :

$$A = \left[\begin{array}{c|c} B & C \\ \hline E & F \end{array}\right] \quad \rightarrow \quad A^{-1} = \left[\begin{array}{c|c} X & Y \\ \hline W & V \end{array}\right] \tag{21}$$

where $V = [F - EB^{-1}C]^{-1}$, $\quad Y = -B^{-1}CV$, $\quad W = -VEB^{-1}$,
$X = B^{-1} - B^{-1}CW$. From the equation (14) and (21), we can write

$$B = \begin{bmatrix} P & Q_1 & Q_2 & Q_3 & \dots & Q_T \\ Q_1 & R_1 & 0 & 0 & \dots & 0 \\ Q_2 & 0 & R_2 & 0 & \dots & 0 \\ Q_3 & 0 & 0 & R_3 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ Q_T & 0 & 0 & 0 & \dots & R_T \end{bmatrix}, \quad C = \begin{bmatrix} S \\ J_1 \\ J_2 \\ J_3 \\ \dots \\ J_T \end{bmatrix}, \quad E = \begin{bmatrix} K \\ L_1 \\ L_2 \\ L_3 \\ \dots \\ L_T \end{bmatrix}^T, \quad F = [G]$$

As X, Y, W, V require B^{-1} is to be computed, we use equation (21) to compute B^{-1} recursively. The effective matrix B progresses as a series of sparse matrix inversions, which speeds up the process. Similarly the the inversion in equation (16) can be optimised by using equation (21) recursively.

5 Experimental Results

In this section, we present comparative experimental results on five traditional single task learning methods and five multi-task learning methods. The experiments are conducted on 3 benchmark multi-task learning data sets. For all the algorithms, the parameters are tuned using grid search strategy. The parameters λ, ρ are selected from a set of $\{2^i | i = -3, -2, ..., 8\}$. The parameters c_1, c_2, c_3, c_4 are selected from a set of $\{2^i | i = -12, -11, ..., 2\}$. In additions to the parameters, Gaussian kernel function is employed on the model. The evaluation parameters used are multi-task average classification accuracy and training time. We employ three-fold cross-validation on these data sets to get average accuracy. All the experiments are performed in Matlab R2018B on ubuntu 16.4 running on a PC with the configuration of Intel Xeon(R) CPU E5-2650 0 @ 2.00GHz with 64.0 GB of RAM.

5.1 Benchmark Data Sets

We select 3 benchmark multi-task data sets to conduct our experiments and test our proposed work. The data sets used are Monk, Isolet and Landmine data

sets. These data sets have been used in the past to assess multi-task learning techniques in the past. The comparative results are shown in table 1.
1. Monk data set- the Monk data set contains 3 sub-data sets. Each sub data set named as Monk 1, Monk 2 and Monk 3 each relate to an individual task. The domain of each task is the same, which effectively means they can be seen as related. We select 150 random yet balanced observations for our experiment. We use the whole data set for further evaluation.
2. Isolet data set- isolet is a spoken recognition data set. It is structured as 150 speakers speaking each alphabet of English twice. Effectively each speaker contributes 56 recorded observations. All the speakers are grouped into groups of 30. Thus there are 5 groups which are highly related to each other. For experimentation point of view, we select two alphabets spoken by all the 150 speakers. Each task differs from the speakers in a different group, but similar to the accent of the alphabets spoken by all of them. We use dimension reduction technique to reduce the dimensionality from 617 to 252 by capturing 97% of the information. This help in improving computational efficiency.
3. Landmine data set- contains 29 sub-tasks. Each task is of the binary classification type. Each observation is a 9-dimensional feature vector extracted from radar images. The information given is that of the presence of landmine or the absence on the basis of the observations in each of the tasks. Out of the 29 tasks, first, 15 of them relate to the observations captured from regions that are foliated. The remaining are captured from areas that are bare earth/deserted. The tasks differ by the geographical locations and are at the same time related by the similarity of the type of geographical area from which the observations are recorded. The data set is highly unbalanced. We conduct our experiment by randomly balancing the data set and as well as the whole data set and see that our algorithm performs well in both the situations.

Table (1) illustrates that R-MT-LS-TWSVM performs well on all the datasets. Fig.1. compares the average accuracy of MT-LS-TWSVM and R-MT-LS-TWSVM for changing size of the tasks. It can be noted that R-MT-LS-TWSVM performs better than MT-LS-TWSVM even when the the size of training data is relatively small. With the increases in data size we observe that our R-MT-LS-TWSVM outperforms MT-LS-TWSVM. We have referred to paper [15] for accuracy and time comparison of other reported algorithms discussed in our paper. The proposed R-MT-LS-TWSVM performs at par with least square version of TWSVM. Further, to investigate robustness of our proposed method and analyse the effect of noise in our data, we have reported standard error of mean. It considers the average accuracy mean along with average accuracy standard deviation. The reducing magnitude of standard deviation with increase in data size supports the validity of our proposed learning algorithm in terms of noise handling. The table (3) suggests that R-MT-LS-TWSVM exhibits consistency over mean accuracies as compared to MT-LS-TWSVM.

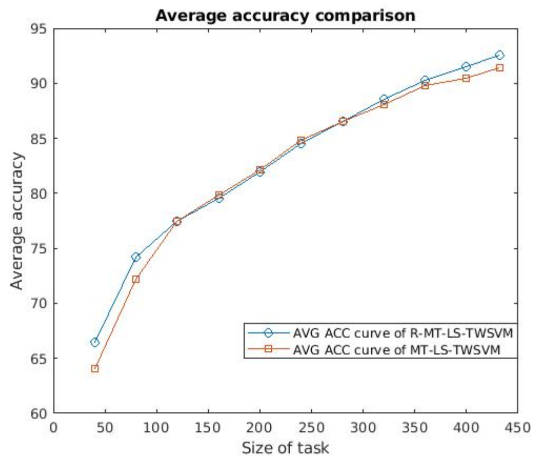


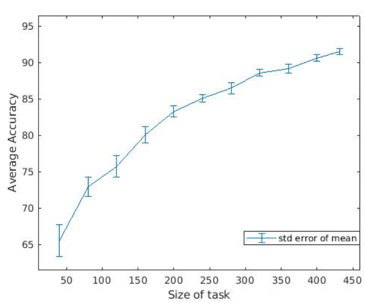
Fig. 1: This is the comparison of Average Accuracy Curve of the MT-LS-TWSVM and R-MT-LS-TWSVM .

Table 1: Comparison of the methods over benchmark data sets.

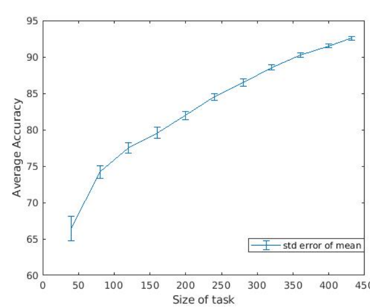
Type	Algorithm	Landmine *Accuracy*	Isolet *Accuracy*	Monk *Accuracy*
	SVM	76.19±5.37	87.17±10.35	94.75±5.58
	PSVM	71.67±7.09	82.50±0.91	78.63±4.20
STL	LS-SVM	75.25±6.49	98.50±0.91	89.51±7.40
	TWSVM	78.35±4.88	98.50±1.09	97.69±3.24
	LSTWSVM	78.44±5.42	98.67±1.12	92.98±9.76
	MTPSVM	76.73±5.61	99.50±0.46	89.20±7.75
	MTLSSVM	76.82±5.73	99.50±0.46	89.20±7.75
	MTL-aLS-SVM	79.14±4.82	**99.83±0.37**	93.29±10.63
MTL	DMTSVM	79.32±5.61	99.67±0.46	**98.23±2.88**
	MCTSVM	**80.00±6.24**	99.67±0.41	98.30±2.94
	MT-LS-TWSVM	79.41±5.83	**99.83±0.37**	92.90±9.89
Proposed	R-MT-LS-TWSVM	78.48±3.67	83.25±0.40	92.41±0.88

Table 2: Comparison of the computation time over benchmark data sets.

Type	Algorithm	Landmine $Time(ms)$	Isolet $Time(ms)$	Monk $Time(ms)$
	SVM	53.1	17.68	49.57
	PSVM	2.08	2.12	9.28
STL	LS-SVM	2.34	2.34	11.21
	TWSVM	98.96	34.29	64.58
	LSTWSVM	4.63	4.09	22.83
	MTPSVM	79.96	9.78	42.15
	MTLSSVM	46.86	11.68	58.45
	MTL-aLS-SVM	766.81	109.50	923.45
MTL	DMTSVM	646.66	68.49	352.06
	MCTSVM	731.93	83.50	377.51
	MT-LS-TWSVM	647.83	73.68	330.08
Proposed	R-MT-LS-TWSVM	650.56	74.77	333.36



(a) MT-LS-TWSVM.



(b) R-MT-LS-TWSVM.

Fig. 2: The comparison plots of standard error of mean.

Table 3: Comparison between MT-LS-TW-SVM and R-MT-LS-TWSVM on standard error on mean.

size of task	MT-LS-TWSVM std. error of mean	R-MT-LS-TWSVM std. error of mean
40	2.1883	1.6556
80	1.3181	0.8730
120	1.4555	0.7423
160	1.1233	0.7493
200	0.7613	0.5346
240	0.5156	0.4684
280	0.7677	0.4942
320	0.4362	0.3773
360	0.5953	0.3042
400	0.4777	0.2406
432	0.4097	0.2626

6 Conclusions

In this paper, we have proposed a novel Robust Multi-Task Least Square Twin Support Vector Machine which performs better than most of the existing single task learning and multi-task learning algorithms. Empirical results prove that it handles noise better than existing variant of multi task algorithms. Moreover, the proposed model is easier to implement and computationally effective which makes its usability when dealing with large data sets. A line of future research could be multi-task learning regularised framework.

References

[1] Abdulnabi, A.H., Wang, G., Lu, J., Jia, K.: Multi-task cnn model for attribute prediction. IEEE Transactions on Multimedia **17**(11), 1949–1959 (2015)

[2] Argyriou, A., Evgeniou, T., Pontil, M.: Convex multi-task feature learning. Mach. Learn. **73**(3), 243–272 (2008). DOI 10.1007/s10994-007-5040-8. URL http://dx.doi.org/10.1007/s10994-007-5040-8

[3] Azad-Manjiri, M., Amiri, A., Sedghpour, A.S.: Ml-slstsvm: a new structural least square twin support vector machine for multi-label learning. Pattern Analysis and Applications **23**(1), 295–308 (2020)

[4] Caruana, R.: Multitask learning. Machine Learning **28**(1), 41–75 (1997). DOI 10.1023/A:1007379606734. URL https://doi.org/10.1023/A:1007379606734

[5] Evgeniou, T., Pontil, M.: Regularized multi–task learning pp. 109–117 (2004). DOI 10.1145/1014052.1014067. URL http://doi.acm.org/10.1145/1014052.1014067

[6] Fung, G.M., Mangasarian, O.L.: Multicategory proximal support vector machine classifiers. Machine learning **59**(1-2), 77–97 (2005)

[7] Gong, P., Ye, J., Zhang, C.: Robust multi-task feature learning. KDD : proceedings. International Conference on Knowledge Discovery & Data Mining **2012**, 895–903 (2012)

[8] Han, L., Zhang, Y.: Learning tree structure in multi-task learning. In: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '15, pp. 397–406. ACM, New York, NY, USA (2015). DOI 10.1145/2783258.2783393. URL http://doi.acm.org/10.1145/2783258.2783393

[9] He, X., Mourot, G., Maquin, D., Ragot, J., Beauseroy, P., Smolarz, A., Grall-Maës, E.: Multi-task learning with one-class svm. Neurocomputing **133**, 416–426 (2014)

[10] Jayadeva, Khemchandani, R., Chandra, S.: Twin support vector machines for pattern classification. IEEE Trans. Pattern Anal. Mach. Intell. **29**(5), 905–910 (2007). DOI 10.1109/TPAMI.2007.1068. URL https://doi.org/10.1109/TPAMI.2007.1068

[11] Kumar, M.A., Gopal, M.: Least squares twin support vector machines for pattern classification. Expert Systems with Applications **36**(4), 7535–7543 (2009)

[12] Li, Y., Tian, X., Liu, T., Tao, D.: On better exploring and exploiting task relationships in multitask learning: Joint model and feature learning. IEEE transactions on neural networks and learning systems **29**(5), 1975–1985 (2018)

[13] Li, Y., Tian, X., Song, M., Tao, D.: Multi-task proximal support vector machine. Pattern Recognition **48**(10), 3249–3257 (2015)

[14] Lu, L., Lin, Q., Pei, H., Zhong, P.: The als-svm based multi-task learning classifiers. Applied Intelligence **48**(8), 2393–2407 (2018)
[15] Mei, B., Xu, Y.: Multi-task least squares twin support vector machine for classification. Neurocomputing (2019)
[16] Qi, K., Liu, W., Yang, C., Guan, Q., Wu, H.: Multi-task joint sparse and low-rank representation for the scene classification of high-resolution remote sensing image. Remote Sensing **9**(1) (2017). DOI 10.3390/rs9010010. URL http://www.mdpi.com/2072-4292/9/1/10
[17] Tanveer, M., Rajani, T., Rastogi, R., Shao, Y.: Comprehensive review on twin support vector machines. arXiv preprint arXiv:2105.00336 (2021)
[18] Widmer, C., Kloft, M., Görnitz, N., Rätsch, G.: Efficient training of graph-regularized multitask svms. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, pp. 633–647. Springer (2012)
[19] Xie, F., Pang, X., Xu, Y.: Pinball loss-based multi-task twin support vector machine and its safe acceleration method. Neural Computing and Applications pp. 1–17 (2021)
[20] Xie, X., Sun, S.: Multitask twin support vector machines pp. 341–348 (2012). DOI 10.1007/978-3-642-34481-7_42. URL http://dx.doi.org/10.1007/978-3-642-34481-7_42
[21] Xie, X., Sun, S.: Multitask centroid twin support vector machines. Neurocomputing **149**, 1085–1091 (2015)
[22] Xu, Z., Kersting, K.: Multi-task learning with task relations. In: 2011 IEEE 11th International Conference on Data Mining, pp. 884–893. IEEE (2011)
[23] Xue, Y., Beauseroy, P.: Multi-task learning for one-class svm with additional new features. In: 2016 23rd International Conference on Pattern Recognition (ICPR), pp. 1571–1576. IEEE (2016)
[24] Yang, H., King, I., Lyu, M.R.: Multi-task learning for one-class classification. In: The 2010 International Joint Conference on Neural Networks (IJCNN), pp. 1–8. IEEE (2010)
[25] Zhang, Y., Yang, Q.: A survey on multi-task learning. arXiv preprint arXiv:1707.08114 (2017)