



Multi-task ν -twin support vector machines

Benshan Mei¹ · Yitian Xu²

Received: 23 March 2019 / Accepted: 22 November 2019 / Published online: 29 November 2019
© Springer-Verlag London Ltd., part of Springer Nature 2019

Abstract

Twin support vector machine (TWSVM) is proved to be better than support vector machine (SVM) in most cases, since it only deals with two smaller quadratic programming problems, which leads to high computational efficiency. It is proposed to solve a single-task learning problem, just like many other machine learning algorithms. However, a learning task may have relationships with other tasks in many practical problems. Training those tasks independently may neglect the underlying information among all tasks, while such information may be useful to improve the overall performance. Inspired by the multi-task learning theory, we propose two novel multi-task ν -TWSVMs. Both models inherit the merits of multi-task learning and ν -TWSVM. Meanwhile, they overcome the shortcomings of other multi-task SVMs and multi-task TWSVMs. Experimental results on three benchmark datasets and two popular image datasets also clearly demonstrate the effectiveness of our methods.

Keywords Multi-task learning · Twin support vector machine · Pattern recognition · Kernel method · Upper and lower bound · Image classification

1 Introduction

As a milestone in the development of support vector machine (SVM) [1], twin support vector machines (TWSVMs) attract much attention during recent years. It is first introduced in [2]. After a decade of research, there are many variants appeared, such as least squares twin support vector machine (LS-TWSVM) [3], twin bounded support vector machine (TBSVM) [4], robust twin support vector machine (robust-TWSVM) [5] and improved twin support vector machine (ITWSVM) [6]. A classical variant of TWSVM is ν -TWSVM [7]. It is motivated by the classical ν -SVM [8] and is proved to be more effective and efficient than ν -SVM. Experiments on both synthetic and real datasets also demonstrate the effectiveness and efficiency of TWSVMs when compared to SVMs [9, 10]. TWSVM has also been applied into many machine learning areas,

such as multi-view learning [11], domain adaptation [12] and clustering (TWSVC) [13]. Based on the PAC-Bayes theory, the generalization ability of TWSVM is analyzed [14]. Some novel safe screening rules are also proposed to speed up TWSVM without performance degradation [15, 16]. More advances of TWSVM can be found in recent survey [17, 18].

We should note that most machine learning algorithms belong to single-task learning, such as support vector machine, linear discriminant analysis, decision tree and so on. Many variants of TWSVM also belong to single-task learning. Actually, we usually train multiple tasks independently. In other words, one task is trained at one time. However, researchers point out that we may neglect the shared information among these tasks, which may be useful to improve the overall performance of these learning algorithms. The multi-task learning theory is thus proposed and has been studied extensively during the past two decades [19, 20]. It aims at improving the overall performance of several related tasks. Compared to single-task learning, it suggests that related tasks may share underlying knowledge, which should be learned jointly so as to take full advantages of the underlying information behind all tasks. Empirical works have demonstrated the effectiveness

✉ Yitian Xu
xytshuxue@126.com

¹ College of Information and Electrical Engineering, China Agricultural University, Beijing 100083, China

² College of Science, China Agricultural University, Beijing 100083, China

of multi-task learning and have also pointed out the mechanism of this machine learning paradigm [21].

A multi-task learning problem may be composed of several single-label learning problems regardless of how these tasks are related. One prerequisite is that all the samples in these tasks share the same feature space, which is also termed as homogeneous multi-task learning [22]. A special case of multi-task learning is multi-label learning, which studies the problem where each sample is associated with a set of labels simultaneously. The relation between these two machine learning paradigms has been clarified in [23]. Suppose the prediction of each label is a task, the multi-label learning problem can be transformed into a multi-task learning problem. By modeling the correlation of all tasks, the relations among multiple labels can be captured as well.

The research on multi-task learning in the early days can be found in [21, 24]. It mainly focused on neural network-based multi-task learning methods and also discussed k -nearest neighbors (k -NN) and decision tree-based multi-task learning algorithms. The generalization bound of multi-task learning was also discussed in early research [25]. At present, many multi-task learning methods appeared, such as Bayesian multi-task learning [26] and multi-task Gaussian process [27]. Recent survey on multi-task learning categorizes these methods into several types, including multi-task feature learning, multi-task relation learning, low-rank approach, dirty approach-based methods, task-clustering methods and other methods [22]. More surveys on multi-task learning can be found in [23, 28].

Recent success in multi-task support vector machines is interesting as well. The first practice is regularized multi-task learning (RMTL) [29–31], which suggests all tasks share a common separating hyperplane and belongs to mean-regularized multi-task learning. It has been used in human action recognition [32]. A generalized sequential minimal optimization (GSMO) is proposed for SVM + MTL [33]. Some other multi-task SVMs are also proposed recently, including multi-task least square support vector machine (MTLS-SVM) [34], multi-task proximal support vector machine (MTPSVM) [35] and multi-task asymmetric least squares support vector machine (MT-aLS-SVM) [36], all of which are based on a certain single-task learning method. Some other variants, such as multi-task infinite latent support vector machines (MT-iLSVM) [37], multi-task multi-class support vector machine (MTMCSVM) [38] and multi-view multi-task support vector machine (MVMTSVM) [39], are also inspiring. Based on SVM, an online multi-task learning algorithm is proposed for semantic concept detection in video [40]. A multi-task ranking SVM is proposed for image co-segmentation [41]. A least squares support vector

machine for semi-supervised multi-task learning is also proposed recently [42].

In contrast to the extensive research on multi-task SVMs, few attention is focused on multi-task TWSVMs. Recent works on multi-task TWSVMs have directed multi-task twin support vector machine (DMTSVM) [43], multi-task centroid twin support vector machine (MTCTSVM) [44] and multi-task least squares twin support vector machine (MTLS-TWSVM) [45]. Compared to their single-task learning counterparts, these models show better generalization performance. They suppose all tasks share two mean hyperplanes, one for the positive and the other for the negative. Inspired by the multi-task learning, we propose two novel multi-task ν -twin support vector machines (MT- ν -TWSVMs) to take full advantage of the regularized multi-task learning and ν -TWSVM. Both models inherit the merits of ν -TWSVM and multi-task learning and overcome the shortage of TWSVM, DMTSVM and MTCTSVM. Thus, our model can perform better than DMTSVM and MTCTSVM. The main contributions of our paper are as follows:

1. We propose two novel multi-task ν -twin support vector machines based on different assumptions. They are natural extensions of ν -TWSVM in multi-task setting.
2. Both models inherit the merits of ν -TWSVM. The fraction of support vectors is thus easier to control in our models than other multi-task SVMs and TWSVMs.
3. The task relation is easier to control in our models and more flexible than other methods.
4. Our models achieve better performance than other multi-task SVMs and TWSVMs.

The remainder of this paper is organized as follows. After a brief review of ν -TWSVM and DMTSVM in Sect. 2, we give a detailed derivation of the proposed MT- ν -TWSVMs in Sects. 3 and 4. Analysis of algorithms is shown in Sect. 5. The numerical experimental results are shown in Sect. 6. Finally, we show the conclusions and future work in Sect. 7.

2 Related work

Here we introduce ν -twin support vector machine (ν -TWSVM) and the primal multi-task twin support vector machine (DMTSVM). It would be better to clarify the original inspiration of these methods, since they lay a solid foundation for our proposed methods.

2.1 ν -Twin support vector machine

For a standard TWSVM model, it aims at finding two nonparallel hyperplanes rather than one hyperplane in

SVM. It is also proved to be faster than SVM for four times on training large datasets. The v -TWSVM is similar to TWSVM. Suppose X_p represents all the positive samples, and X_n stands for the negative. For simplicity, denote $A = [X_p e]$, $B = [X_n e]$, $u = [w_1, b_1]^T$ and $v = [w_2, b_2]^T$, and this model generates two nonparallel hyperplanes by solving the following problems,

$$\begin{aligned} \min_{u, p, \rho_+} \quad & \frac{1}{2} \|Au\|^2 - v_1 \rho_+ + \frac{1}{l_-} e_2^T p \\ \text{s.t.} \quad & -Bu + p \geq \rho_+, \\ & \rho_+, p \geq 0, \end{aligned} \tag{1}$$

and

$$\begin{aligned} \min_{v, q, \rho_-} \quad & \frac{1}{2} \|Bv\|^2 - v_2 \rho_- + \frac{1}{l_+} e_1^T q \\ \text{s.t.} \quad & Av + q \geq \rho_-, \\ & \rho_-, q \geq 0. \end{aligned} \tag{2}$$

where v_1 and v_2 are positive parameters. l_+ and l_- denote the numbers of positive samples and negative samples, respectively. Both e_1 and e_2 are vectors of ones of appropriate dimensions. Then, a new point $x \in R^n$ is assigned to class $i(i = +1, -1)$ by

$$f(x) = \arg \min_{r=\pm 1} |x^T w_r + b_r|. \tag{3}$$

This model is modeled after the v -SVM. It can adjust the fraction of support vectors and is proved to be more efficient and effective than traditional SVMs and TWSVMs. However, just like many other single-task learning models, it is not designed to deal with the commonality and individuality of multiple tasks.

2.2 Multi-task twin support vector machine

This model introduces TWSVM into multi-task learning setting, is modeled after the RMTL and also is called direct multi-task twin support vector machine (DMTSVM) [43], unlike multi-task support vector machines, which supposes all tasks share two mean hyperplanes. Suppose the positive (negative) samples in the t th task are represented by $X_{pt}(X_{nt})$. Meanwhile, X_p represents the positive samples, while X_n stands for the negative. Now, we let

$$A_t = [X_{pt} e_t], B_t = [X_{nt} e_t], A = [X_p e], B = [X_n e],$$

where e_t and e are one vectors of appropriate dimensions.

Suppose there are two mean hyperplanes $u = [w_1, b_1]^T$ and $v = [w_2, b_2]^T$ shared by all tasks, the two hyperplanes in the t th task are $(u + u_t) = [w_{1t}, b_{1t}]^T$ and $(v + v_t) = [w_{2t}, b_{2t}]^T$, respectively. The bias between the hyperplanes in the t th task and the common hyperplanes u

and v is captured by u_t and v_t . Then, the primal problem of DMTSVM is illustrated as follows:

$$\begin{aligned} \min_{u, u_t, p_t} \quad & \frac{1}{2} \|Au\|^2 + \frac{1}{2} \sum_{t=1}^T \rho_t \|A_t u_t\|^2 + c_1 \sum_{t=1}^T e_{2t}^T p_t \\ \text{s.t.} \quad & -B_t(u + u_t) + p_t \geq e_t, \quad p_t \geq 0, \end{aligned} \tag{4}$$

and

$$\begin{aligned} \min_{v, v_t, q_t} \quad & \frac{1}{2} \|Bv\|^2 + \frac{1}{2} \sum_{t=1}^T \lambda_t \|B_t v_t\|^2 + c_2 \sum_{t=1}^T e_{1t}^T q_t \\ \text{s.t.} \quad & A_t(v + v_t) + q_t \geq e_t, \quad q_t \geq 0, \end{aligned} \tag{5}$$

where $t \in \{1, 2, \dots, T\}$, e_{1t} and e_{2t} are one vectors. c_1 and c_2 are nonnegative trade-off parameters. The relationships of all tasks can be adjusted by parameters ρ_t and λ_t . Both p_t and q_t are slack variables. Then all tasks can be modeled unrelated when $\rho_t \rightarrow 0$ and $\lambda_t \rightarrow 0$ simultaneously. On the contrary, these models will be learned the same when $\rho_t \rightarrow \infty$ and $\lambda_t \rightarrow \infty$. Finally, the label of a new point x in the t th task can be determined by

$$f(x) = \arg \min_{r=\pm 1} |x^T w_{rt} + b_{rt}|. \tag{6}$$

3 Multi-task v -twin support vector machine I

3.1 Linear case

In this section, based on the regularized multi-task learning and the v -TWSVM, we propose a primal multi-task learning problems as follows:

$$\begin{aligned} \min_{u_0, u_t, \rho_+, p_t} \quad & \frac{1}{2} \|Au_0\|^2 + \frac{\mu_1}{2T} \sum_{t=1}^T \|A_t u_t\|^2 - v_1 \rho_+ \\ & + \frac{1}{l_-} \sum_{t=1}^T e_{2t}^T p_t \\ \text{s.t.} \quad & -B_t(u_0 + u_t) + p_t \geq \rho_+, \\ & \rho_+, p_t \geq 0, \end{aligned} \tag{7}$$

and

$$\begin{aligned} \min_{v_0, v_t, \rho_-, q_t} \quad & \frac{1}{2} \|Bv_0\|^2 + \frac{\mu_2}{2T} \sum_{t=1}^T \|B_t v_t\|^2 - v_2 \rho_- \\ & + \frac{1}{l_+} \sum_{t=1}^T e_{1t}^T q_t \\ \text{s.t.} \quad & A_t(v_0 + v_t) + q_t \geq \rho_-, \\ & \rho_-, q_t \geq 0, \end{aligned} \tag{8}$$

where $t \in \{1, 2, \dots, T\}$, l_+ and l_- are the numbers of positive and negative samples, separately. Note that, $w_{rt}(r \in \{+1, -1\})$ are the weight vectors of the

hyperplanes for each task. Here, $u_0(v_0)$ and $u_t(v_t)$ indicate the commonality and personality of each task, separately. The difference between all tasks is controlled by μ . However, we take ideas from v -TWSVM. Its merits may be different from the DMTSVM and MTCTSVM. Two additional variables ρ_{\pm} in (7) and (8) need to be optimized. Before analyzing the effect of v , we take the dual problem of (7) and (8). The Lagrangian function of problem (7) is given by

$$L_1 = \frac{1}{2} \|Au_0\|^2 + \frac{\mu_1}{2T} \sum_{t=1}^T \|A_t u_t\|^2 - v_1 \rho_+ + \frac{1}{l_-} \sum_{t=1}^T e_{2t}^\top p_t - \sum_{t=1}^T \alpha_t^\top (-B_t(u_0 + u_t) + p_t - \rho_+) - \sum_{t=1}^T \beta_t^\top p_t - \eta \rho_+, \tag{9}$$

where α_t , β_t and η are the Lagrangian multipliers. The Karush–Kuhn–Tucker (KKT) conditions are given below

$$\begin{aligned} \frac{\partial L}{\partial u_0} &= A^\top A u_0 + B^\top \alpha = 0, \\ \frac{\partial L}{\partial u_t} &= \frac{\mu_1}{T} A_t^\top A_t u_t + B_t^\top \alpha_t = 0, \\ \frac{\partial L}{\partial \rho_+} &= -v_1 + e_2^\top \alpha - \eta = 0 \Rightarrow e_2^\top \alpha \geq v_1, \\ \frac{\partial L}{\partial p} &= \frac{e_2}{l_-} - \alpha - \beta = 0 \Rightarrow 0 \leq \alpha \leq \frac{1}{l_-}, \end{aligned} \tag{10}$$

where $\alpha = [\alpha_1^\top, \alpha_2^\top, \dots, \alpha_T^\top]^\top$ and $p = [p_1^\top, p_2^\top, \dots, p_T^\top]^\top$. Then, we have

$$\begin{aligned} u_0 &= -(A^\top A)^{-1} B^\top \alpha, \\ u_t &= -\frac{T}{\mu_1} (A_t^\top A_t)^{-1} B_t^\top \alpha_t. \end{aligned} \tag{11}$$

Then, substituting u_0, u_t into function (9)

$$L_1 = \frac{1}{2} u_0^\top A^\top A u_0 + \frac{\mu_1}{2T} \sum_{t=1}^T u_t^\top A_t^\top A_t u_t + \sum_{t=1}^T \alpha_t^\top B_t (u_0 + u_t), \tag{12}$$

and using below equations

$$\begin{aligned} Q &= B(A^\top A)^{-1} B^\top, \\ P_t &= B_t(A_t^\top A_t)^{-1} B_t^\top, \\ P &= \text{blkdiag}(P_1, P_2, \dots, P_T), \end{aligned} \tag{13}$$

the dual problem of (7) can be simplified as

$$\begin{aligned} \max_{\alpha} \quad & -\frac{1}{2} \alpha^\top \left(Q + \frac{T}{\mu_1} P \right) \alpha \\ \text{s.t.} \quad & e_2^\top \alpha \geq v_1, \\ & 0 \leq \alpha \leq \frac{e_2}{l_-}. \end{aligned} \tag{14}$$

Similarly, we introduce below equations

$$\begin{aligned} R &= A(B^\top B)^{-1} A^\top, \\ S_t &= A_t(B_t^\top B_t)^{-1} A_t^\top, \\ S &= \text{blkdiag}(S_1, S_2, \dots, S_T). \end{aligned} \tag{15}$$

The dual problem of (8) can be written as

$$\begin{aligned} \max_{\gamma} \quad & -\frac{1}{2} \gamma^\top \left(R + \frac{T}{\mu_2} S \right) \gamma \\ \text{s.t.} \quad & e_1^\top \gamma \geq v_2, \\ & 0 \leq \gamma \leq \frac{e_1}{l_+}. \end{aligned} \tag{16}$$

Finally, the label of a new sample x in the t th task can be determined by

$$f(x) = \arg \min_{r=\pm 1} |x^\top w_{rt} + b_{rt}|. \tag{17}$$

3.2 Nonlinear case

Since linear classifier may not be appropriate for linear nonseparable cases, the kernel trick can be used in such case. Now, we introduce the kernel function $K(\cdot)$ and define

$$\begin{aligned} E &= [K(A, X^\top), e], E_t = [K(A_t, X^\top), e_t], \\ F &= [K(B, X^\top), e], F_t = [K(B_t, X^\top), e_t]. \end{aligned}$$

where X represents training samples from all tasks, i.e., $X = [A_1^\top, B_1^\top, A_2^\top, B_2^\top, \dots, A_T^\top, B_T^\top]^\top$. By substituting A and B in (7) and (8) with E and F , respectively, we can obtain the kernel version of this model. The primal problems of the nonlinear model are

$$\begin{aligned} \min_{u_0, u_t, \rho_+, \rho_t} \quad & \frac{1}{2} \|E u_0\|^2 + \frac{\mu_1}{2T} \sum_{t=1}^T \|E_t u_t\|^2 - v_1 \rho_+ \\ & + \frac{1}{l_-} \sum_{t=1}^T e_{2t}^\top p_t \\ \text{s.t.} \quad & -F_t(u_0 + u_t) + p_t \geq \rho_+, \\ & \rho_+, p_t \geq 0, \end{aligned} \tag{18}$$

and

$$\begin{aligned} \min_{v_0, v_t, \rho_-, q_t} \quad & \frac{1}{2} \|Fv_0\|^2 + \frac{\mu_2}{2T} \sum_{t=1}^T \|F_tv_t\|^2 - v_2\rho_- \\ & + \frac{1}{l_+} \sum_{t=1}^T e_{1t}^\top q_t \\ \text{s.t.} \quad & E_t(v_0 + v_t) + q_t \geq \rho_-, \\ & \rho_-, q_t \geq 0. \end{aligned} \tag{19}$$

Then the corresponding decision function of the t th task is

$$f(x) = \arg \min_{r=\pm 1} |K(x, X^\top)w_{rt} + b_{rt}|. \tag{20}$$

4 Multi-task v -twin support vector machine II

Although it is easy to understand MT- v -TWSVM I, this model may have some disadvantages. Because the range of parameters μ_1 (μ_2) is $(0, +\infty)$, it is hard for us to adjust the relationship among multiple tasks. Thus we propose another multi-task v -TWSVM to address this problem in this section.

4.1 Linear case

Suppose the hyperplane of the t th task can be expressed as a linear convex combination of the common vectors $u_0(v_0)$ and the task specific vectors $u_t(v_t)$, we propose another problem as follows:

$$\begin{aligned} \min_{u_0, u_t, \rho_+, p_t} \quad & \frac{\mu_1}{2} \|Au_0\|^2 + \frac{1-\mu_1}{2T} \sum_{t=1}^T \|A_t u_t\|^2 - v_1\rho_+ \\ & + \frac{1}{l_-} \sum_{t=1}^T e_{2t}^\top p_t \\ \text{s.t.} \quad & -B_t(\mu_1 u_0 + (1-\mu_1)u_t) + p_t \geq \rho_+, \\ & \rho_+, p_t \geq 0, \end{aligned} \tag{21}$$

and

$$\begin{aligned} \min_{v_0, v_t, \rho_-, q_t} \quad & \frac{\mu_2}{2} \|Bv_0\|^2 + \frac{1-\mu_2}{2T} \sum_{t=1}^T \|B_tv_t\|^2 - v_2\rho_- \\ & + \frac{1}{l_+} \sum_{t=1}^T e_{1t}^\top q_t \\ \text{s.t.} \quad & A_t(\mu_2 v_0 + (1-\mu_2)v_t) + q_t \geq \rho_-, \\ & \rho_-, q_t \geq 0, \end{aligned} \tag{22}$$

where $t \in \{1, 2, \dots, T\}$.

Similar to DMTSVM, the differences between all tasks are controlled by parameters μ_1 and μ_2 . But it is different from DMTSVM, that is, the task relation is captured by a linear convex combination of the common hyperplane

$u_0(v_0)$ and a specific vector $u_t(v_t)$ for the positive (negative). If we set $\mu_1 = 0$ and $\mu_2 = 0$, it means that u_0 and v_0 have no effect on the t th task. Then T completely different tasks will be learned, and the hyperplanes of the t th task will be far away from the common hyperplanes. When $\mu_1 = 1$ and $\mu_2 = 1$, our model reduces to an enlarged v -TWSVM, and it means all tasks have the same two hyperplanes. Therefore, the difference among all tasks can be easily captured by two parameters μ_1 and μ_2 . It is more flexible than DMTSVM and MTCT SVM.

However, both models are based on v -TWSVM. Their merits may be different from the DMTSVM and MTCT SVM. Two additional variables ρ_\pm in (21) and (22) need to be optimized. Before analyzing the effect of v , we take the dual problem of (21). The Lagrangian function of problem (21) is given by

$$\begin{aligned} L_1 = & \frac{\mu_1}{2} \|Au_0\|^2 + \frac{1-\mu_1}{2T} \sum_{t=1}^T \|A_t u_t\|^2 + \frac{1}{l_-} \sum_{t=1}^T e_{2t}^\top p_t \\ & - \sum_{t=1}^T \alpha_t^\top (-B_t(\mu_1 u_0 + (1-\mu_1)u_t) + p_t - \rho_+) \\ & - \sum_{t=1}^T \beta_t^\top p_t - \eta\rho_+ - v_1\rho_+. \end{aligned} \tag{23}$$

Taking the partial derivatives of Lagrangian function (23) with respect to (w_0, w_t, ρ_+, p) , we obtain the following KKT conditions:

$$\begin{aligned} \frac{\partial L}{\partial u_0} = & \mu_1(A^\top Au_0 + B^\top \alpha) = 0, \\ \frac{\partial L}{\partial u_t} = & (1-\mu_1) \left(\frac{1}{T} A_t^\top A_t u_t + B_t^\top \alpha_t \right) = 0, \\ \frac{\partial L}{\partial \rho_+} = & -v_1 + e_2^\top \alpha - \eta = 0 \Rightarrow e_2^\top \alpha \geq v_1, \\ \frac{\partial L}{\partial p} = & \frac{e_2}{l_-} - \alpha - \beta = 0 \Rightarrow 0 \leq \alpha \leq \frac{1}{l_-}, \end{aligned} \tag{24}$$

where $\alpha = [\alpha_1^\top, \alpha_2^\top, \dots, \alpha_T^\top]^\top$.

Then, we have the following equalities with respect to primal problem variables (u_0, u_t)

$$\begin{aligned} u_0 = & -(A^\top A)^{-1} B^\top \alpha, \\ u_t = & -T \cdot (A_t^\top A_t)^{-1} B_t^\top \alpha_t. \end{aligned} \tag{25}$$

Then, we substitute u_0 and u_t into function (23)

$$\begin{aligned} L_1 = & \frac{\mu_1}{2} u_0^\top A^\top Au_0 + \frac{1-\mu_1}{2T} \sum_{t=1}^T u_t^\top A_t^\top A_t u_t \\ & + \sum_{t=1}^T \alpha_t^\top B_t(\mu_1 u_0 + (1-\mu_1)u_t). \end{aligned} \tag{26}$$

The dual problem of (21) can be simplified as

$$\begin{aligned} \max_{\alpha} \quad & -\frac{1}{2}\alpha^{\top}(\mu_1 Q + (1 - \mu_1)T \cdot P)\alpha \\ \text{s.t.} \quad & e_2^{\top}\alpha \geq v_1, \\ & 0 \leq \alpha \leq \frac{e_2}{l_-}. \end{aligned} \tag{27}$$

Similarly, the dual problem of (22) can be written as

$$\begin{aligned} \max_{\gamma} \quad & -\frac{1}{2}\gamma^{\top}(\mu_2 R + (1 - \mu_2)T \cdot S)\gamma \\ \text{s.t.} \quad & e_1^{\top}\gamma \geq v_2, \\ & 0 \leq \gamma \leq \frac{e_1}{l_+}. \end{aligned} \tag{28}$$

Finally, the label of a new sample x in the t th task can be determined by

$$f(x) = \arg \min_{r=\pm 1} |x^{\top} w_{rt} + b_{rt}|. \tag{29}$$

4.2 Nonlinear case

A linear classifier may not be suitable for training samples that are linear inseparable. The kernel trick can be used to deal with such problems. Similarly, we introduce the kernel function $K(\cdot)$ and define

$$\begin{aligned} E &= [K(A, X^{\top}), e], \quad E_t = [K(A_t, X^{\top}), e_t], \\ F &= [K(B, X^{\top}), e], \quad F_t = [K(B_t, X^{\top}), e_t], \end{aligned}$$

where X represents training samples from all tasks, i.e., $X = [A_1^{\top}, B_1^{\top}, A_2^{\top}, B_2^{\top}, \dots, A_T^{\top}, B_T^{\top}]^{\top}$. The $K(\cdot)$ is a kernel function. The primal problems of the nonlinear case are given as

$$\begin{aligned} \min_{u_0, u_t, \rho_+, p_t} \quad & \frac{\mu_1}{2} \|Eu_0\|^2 + \frac{1 - \mu_1}{2T} \sum_{t=1}^T \|E_t u_t\|^2 - v_1 \rho_+ \\ & + \frac{1}{l_-} \sum_{t=1}^T e_{2t}^{\top} p_t \\ \text{s.t.} \quad & -F_t(\mu_1 u_0 + (1 - \mu_1)u_t) + p_t \geq \rho_+, \\ & \rho_+, p_t \geq 0, \end{aligned} \tag{30}$$

and

$$\begin{aligned} \min_{v_0, v_t, \rho_-, q_t} \quad & \frac{\mu_2}{2} \|Fv_0\|^2 + \frac{1 - \mu_2}{2T} \sum_{t=1}^T \|F_t v_t\|^2 - v_2 \rho_- \\ & + \frac{1}{l_+} \sum_{t=1}^T e_{1t}^{\top} q_t \\ \text{s.t.} \quad & E_t(\mu_2 v_0 + (1 - \mu_2)v_t) + q_t \geq \rho_-, \\ & \rho_-, q_t \geq 0. \end{aligned} \tag{31}$$

Then the decision function of the t th task is

$$f(x) = \arg \min_{r=\pm 1} |K(x, X^{\top})w_{rt} + b_{rt}|. \tag{32}$$

5 Analysis of algorithms

5.1 Equivalent form of model

The dual problems of MT- v -TWSVM are similar to that of v -TWSVM. The difference lies in the Hessian matrix. Besides, these models share similar features with v -TWSVM. Similar to v -SVM and v -TWSVM, to compute ρ_{\pm} , we select samples x_i (or x_j) with $0 < \alpha_i < \frac{1}{l_-}$ (or $0 < \gamma_j < \frac{1}{l_+}$) from all tasks, which means that $p_t = 0$ (or $q_t = 0$) and $w_1^{\top} x_j + b_+ = -\rho_+$ (or $w_2^{\top} x_i + b_2 = \rho_-$). According to the KKT conditions, the ρ_{\pm} can be calculated by

$$\begin{aligned} \rho_+ &= -\frac{1}{l_-} \sum_{t=1}^T \sum_{j=1}^{N_m} (w_{1t}^{\top} x_j + b_{1t}), \\ \rho_- &= \frac{1}{l_+} \sum_{t=1}^T \sum_{i=1}^{N_p} (w_{2t}^{\top} x_i + b_{2t}), \end{aligned} \tag{33}$$

where N_m and N_p represent the number of negative and positive samples satisfying above constraints in the t th task.

Here we show an equivalent form of QPP (14). However, the optimal value of parameter ρ_+ (ρ_-) is actually larger than zero. According to previous conclusions, we have the following Proposition 1.

Proposition 1 QPP (14) can be transformed into the following QPP.

$$\begin{aligned} \max_{\alpha} \quad & -\frac{1}{2}\alpha^{\top} \left(Q + \frac{T}{\mu_1} P \right) \alpha \\ \text{s.t.} \quad & 0 \leq \alpha \leq \frac{e_2}{l_-}, \\ & e_2^{\top} \alpha = v_1. \end{aligned} \tag{34}$$

The difference between (14) and (34) lies in the second constraint. The second inequality constraint $e_2^{\top} \alpha \geq v_1$ can be transformed into an equality constraint $e_2^{\top} \alpha = v_1$.

Proof According to the KKT conditions $\eta \rho_+ = 0$ and the assumption $\rho_+ > 0$, we have that $\eta = 0$. Then we obtain the equality constraint $e_2^{\top} \alpha = v_1$ by substituting η into (10). Thus we prove Proposition 1.

Similar to v -TWSVM, dual problems (14) and (15) of MT- v TWSVM I can be seen as minimizing the generalized Mahalanobis norm. This norm is defined as $\|u\|_{GM} = \sqrt{u^{\top} S u}$. Here, we set $S = Q + \frac{T}{\mu_1} P$, and problem (14) can be written as a standard generalized Mahalanobis norm minimizing problem as follows,

$$\begin{aligned} \min_{\alpha} \quad & \frac{1}{2} \alpha^T S \alpha \\ \text{s.t.} \quad & 0 \leq \alpha \leq \alpha_m, \\ & e_2^T \alpha = 1, \end{aligned} \tag{35}$$

where $\alpha_m = \frac{e_2}{v_1 l_+}$. Further analysis of this property can be found in [7], and the only difference lies in the Hessian matrix. Similar conclusions can be obtained for QPP (15) as well. The MT- ν -TWSVM II also has these features. \square

5.2 Property of parameter ν

As in ν -TWSVM, parameter ν in our multi-task ν -TWSVMs also has these properties. They are discussed in the following propositions.

Proposition 2 *Suppose we run both MT- ν -TWSVM I and II with n samples on dataset \mathcal{D} , obtaining the result that $\rho_{\pm} \geq 0$. Then*

1. ν_2 (or ν_1) is an upper bound on the fraction of positive (or negative) margin errors of the common task.
2. ν_2 (or ν_1) is a lower bound on the fraction of positive (or negative) support vectors of the common task.

Proof The proof of Proposition 2 is similar to that of Proposition 5 in [8]. These results can be extended to the nonlinear case by introducing the kernel function. \square

5.3 Complexity analysis

We analyze the training time complexity of our proposed algorithms. Clearly, both models need solving two smaller quadratic programming problems. It is the same as training original single-task learning ν -TWSVM on all the samples in these tasks. Although one may notice that there is $2T + 2$ times matrix inversion in a training process, we note that it can be better optimized by carefully organizing the training procedure of the grid process. It will not affect the overall time complexity theoretically. Therefore, the training time complexity of our proposed algorithms is the same as that of ν -TWSVM. Suppose the number of training samples in all the tasks is l , the time complexity of our algorithms is also $O(\frac{l^3}{4})$.

According to the analysis, we know that training such a multi-task learning model needs additional computation when compared to training a unify model on all the samples. But in one aspect, the personality and commonality can be modeled to improve the overall performance. In other aspect, the training tasks could help each other in a multi-task learning scenario. This is what a single-task learning method cannot achieve practically.

6 Numerical experiments

In this section, we present experimental results on both single-task learning methods and multi-task learning algorithms. The single-task learning algorithms are consisted of SVM, PSVM, LSSVM, TWSVM, LSTWSVM and ν -TWSVM, while the multi-task learning methods are MTPSVM, MTLs-SVM, MTL-aLS-SVM, DMTSVM, MCTSVM and our proposed MT- ν -TWSVM I and II. The numerical experiments are first conducted on three benchmark datasets. To further evaluate these methods, we have also made comparisons on popular Caltech 101 and 256 datasets.

For each algorithm, all parameters, such as λ , γ and ρ , are turned by grid-search strategy. Without specification, all parameters are selected from set $\{2^i | i = -3, -2, \dots, 8\}$. The parameter p in MTL-aLS-SVM is selected from set $\{0.82, 0.86, 0.90, 0.95\}$. The parameter ν in ν -TWSVM and MT- ν -TWSVMs is selected from set $\{0.1, 0.2, \dots, 1.0\}$. The parameter μ in MT- ν -TWSVM II is selected from set $\{0, 0.1, \dots, 0.9, 1\}$. Then, we use fivefold cross-validation to obtain average performance. Finally, all experiments are conducted in MATLAB R2018b on Windows 8.1 running on a PC with system configuration of Intel(R) Core(TM) i3-6100 CPU (3.90 GHz) with 12.00 GB of RAM.

We note one special operation we had done to handle multi-task learning problems when conducting simulations. Since training a group of unrelated tasks may have negative impact on the performance of our proposed multi-task learning models, all the training tasks should be conceptually positive related. In our work, the training tasks satisfy such requirement to a certain extent. Thus it can better utilize the generalization ability of our proposed multi-task learning methods.

6.1 Benchmark datasets

In this subsection, we conduct experiments on three datasets. The general information is show in (Table 1). The details of these datasets are as follows.

Monk This dataset comes from a first international comparison of learning algorithms and contains three Monk’s problems corresponding to three tasks [46]. The domains of all tasks are the same. Thus, these tasks can be seen as related. We select different number of samples to test these methods.

Emotions This is a multi-label dataset in Mulan library [47] and is used to recognize different emotions. There are six kinds of labels for all samples. Each sample may have more than one label (or emotion). Suppose the recognition tasks of different emotions share similar features and can

be seen as related tasks. We cast it into a multi-task classification problem, and each task is to recognize one type of emotion. We select 100 to 200 samples from this dataset to evaluate these multi-task learning algorithms.

Flags This is a multi-label dataset in Mulan library [47] as well. Each sample may have seven labels. Since the recognition task of each label can be seen as related. Thus, we also consider it as a multi-task learning problem. Then we select different number of samples from this dataset to compare the performance of these multi-task learning methods.

Finally, we use Gaussian kernel function on *Monk* dataset only. But considering the feature mapping of the Gaussian kernel, the data could be more linear separable in high-dimensional space, causing the classification performance of each model cannot be easily distinguished on limited testing samples. Therefore, a polynomial kernel function is applied in our experiments, i.e.,

$$K(x_i, x_j) = (\langle x_i, x_j \rangle + c)^d. \quad (36)$$

In our experiments, we set the kernel parameter $c = 1$ and $d = 2$. By the kernel trick, the input data are mapped into a high-dimensional feature space. In the feature space, a linear classifier is implemented which corresponds to a nonlinear separating surface in the input space.

Figures 1 and 2 show the performance comparison on *Monk* dataset with RBF kernel function. We can learn from them that our algorithms clearly outperform these single-task learning algorithms. The performance of each algorithm increases when increasing the size of each task. With the task size increases, the performance gap between single-task learning methods and our methods decreases. It can be explained as follows. Since our models train all tasks simultaneously, it can take advantage of the underlying information among all tasks when there are few samples in each task. The performance of single-task learning algorithms also becomes better when increasing the number of samples. Thus our multi-task learning methods are suitable for training small tasks. Meanwhile, the performance of our models in Fig. 2 is clearly better than other multi-task learning algorithms when there are few samples in each task. With the increase in task size, the performance of each algorithm also increases. In addition,

we note that the average training time of these four multi-task TWSVMs is almost the same, while the training time of MTPSVM and MTL-SVM is lower than the other algorithms.

In the following, the performance comparison on *Monk* dataset with polynomial kernel function is shown in Figs. 3 and 4. We point out that our algorithms also outperform other single tasks at varying task size. In addition, since our methods train all tasks simultaneously, the training time is surely larger than those single-task learning algorithms. But we also note that the training time of SVM, TWSVM and v -TWSVM is close to our methods when there are few training samples in each task, since these algorithms need to solve one or two quadratic programming problems. In addition, our algorithms also perform better than other multi-task learning methods at varying task size in terms of the mean accuracy. Meanwhile, the average training time of the last four algorithms is almost the same. The training time of PSVM, LSSVM and their multi-task learning extensions MTPSVM and MTL-SVM is the lowest among all algorithms. Finally, our algorithms perform better than these single-task learning and multi-task learning methods on *Monk* dataset in our experimental results in terms of the mean accuracy.

The experimental results on *Flags* dataset between multi-task learning algorithms with polynomial kernel function are illustrated in Fig. 5. Our algorithms perform better than other multi-task learning algorithms when the number of samples in each task is larger than 100. Since there are 7 tasks in this dataset, we cannot suppose these tasks are really correlated. The performance of these multi-task learning algorithms may not be so well. But our algorithms still perform better than these three multi-task SVMs in terms of mean accuracy. In addition, the training time of MPT SVM and MTL-SVM is clearly lower than other algorithms. However these two algorithms just solve a larger linear equation problem, while our algorithms need to solve two smaller quadratic programming problems and several small matrix inversions. The computational costs of our methods are naturally high.

The comparison between multi-task learning algorithms on *Emotions* dataset with polynomial kernel function is shown in Fig. 6. In this group of experiments, our algorithms perform better than other similar methods in terms of the mean accuracy. The MTPSVM and MTL-SVM are faster than other methods. The average training time of the last five algorithms is almost the same.

Table 1 The statistics of these three datasets

Name	#Tasks	#Samples	#Features
Monk	3	432	6
Flags	7	194	19
Emotions	6	593	72

6.2 Image datasets

To further evaluate the effectiveness of MT- v -TWSVMs, we conduct experiments on two image datasets. The images are selected from the Caltech 101 [48, 49] and the

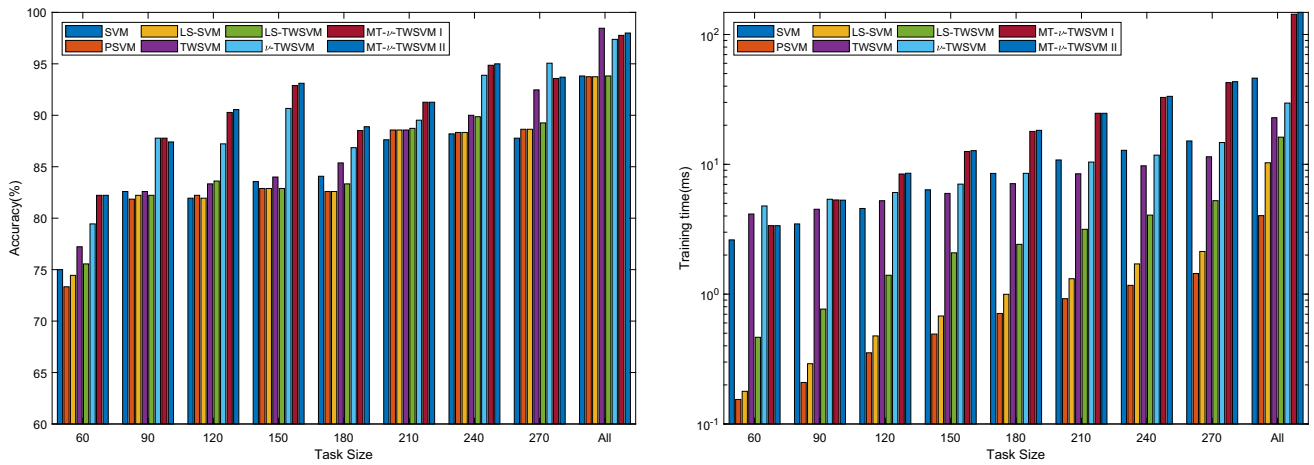


Fig. 1 Performance comparison between our methods and six STL methods on *Monk* dataset (RBF kernel)

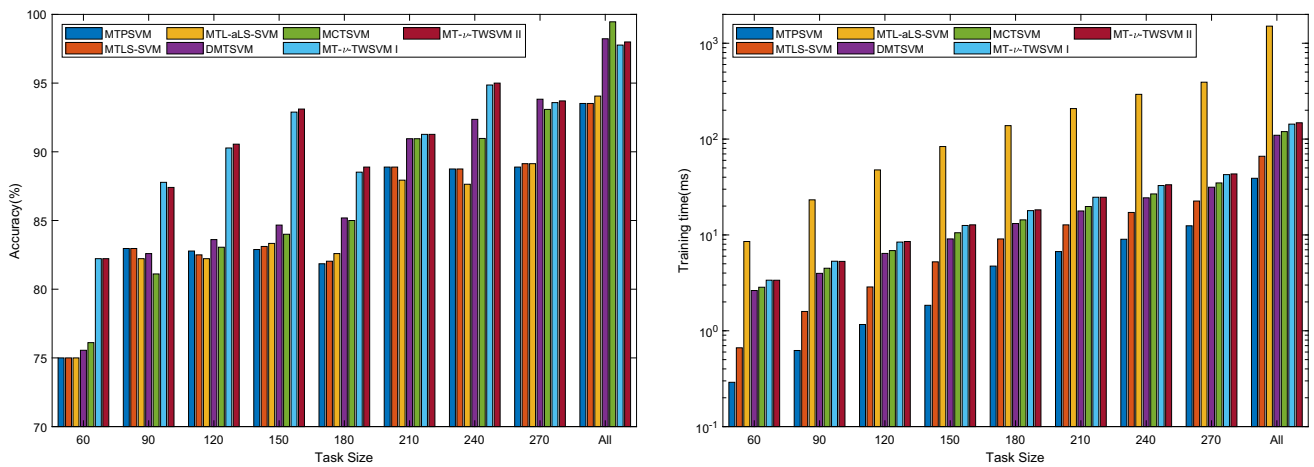


Fig. 2 Performance comparison between MTL methods on *Monk* dataset (RBF kernel)

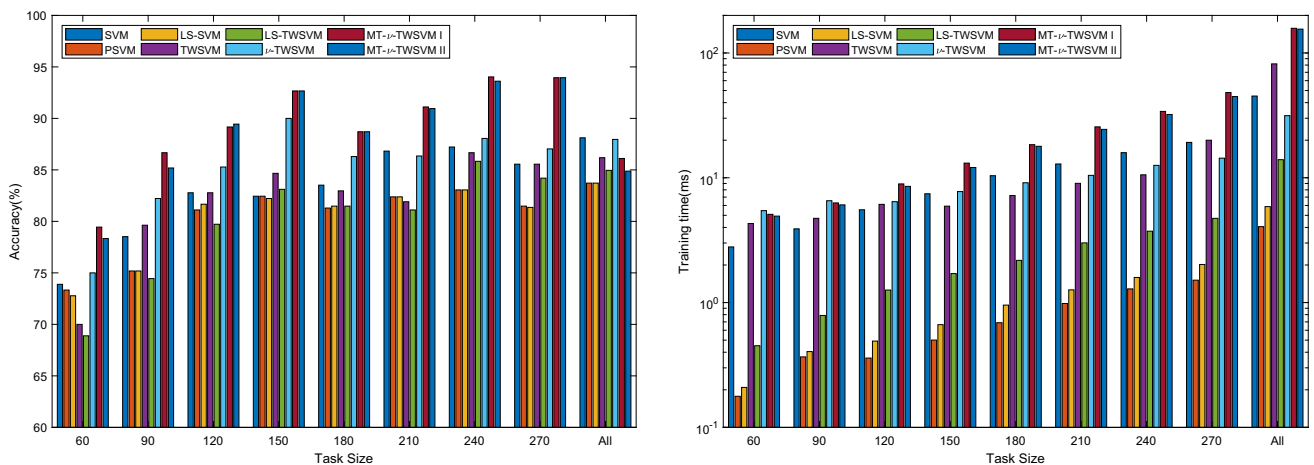


Fig. 3 Performance comparison between our methods and six STL methods on *Monk* dataset (polynomial kernel)

Caltech 256 datasets [50], which have been widely used in computer vision researches. There are 102 categories in Caltech 101 dataset, and each category has more than 50

samples. Each image has about 300×200 pixels [48]. We select about 50 samples from each category in our experiments. Caltech 256 dataset has 256 categories of images in

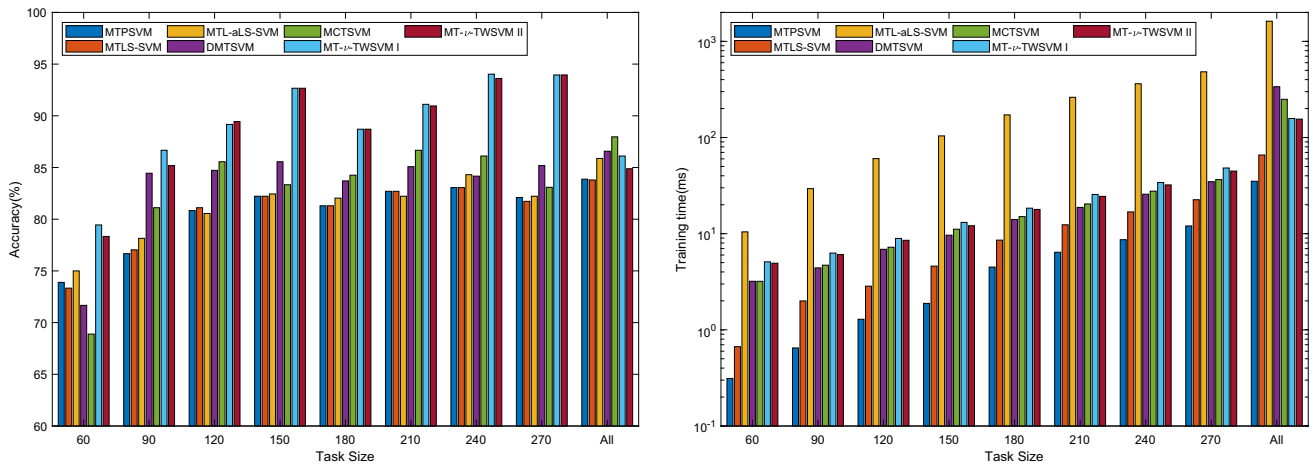


Fig. 4 Performance comparison between MTL methods on *Monk* dataset (polynomial kernel)

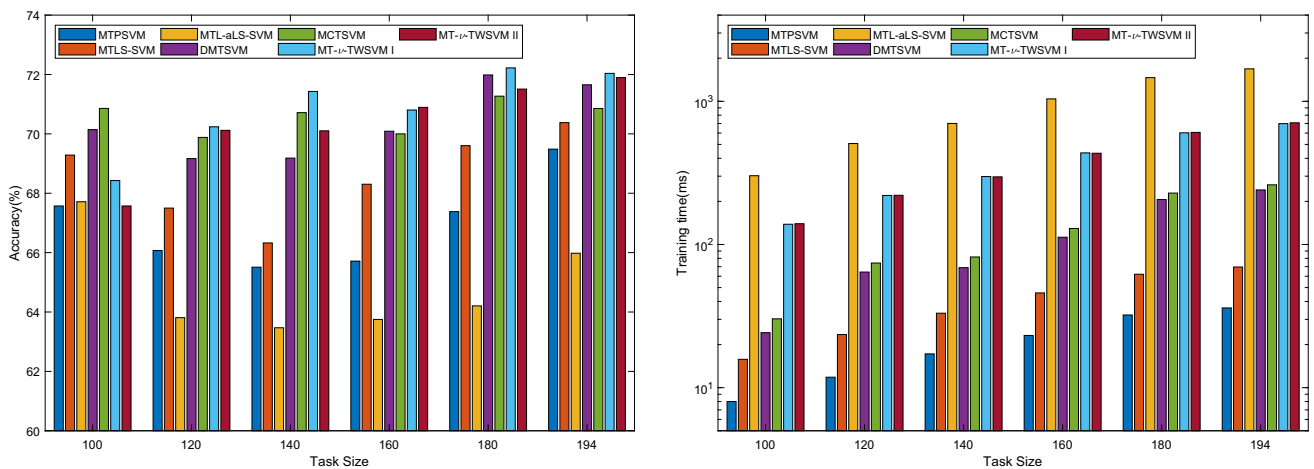


Fig. 5 Performance comparison between MTL methods on *Flags* dataset (polynomial kernel)

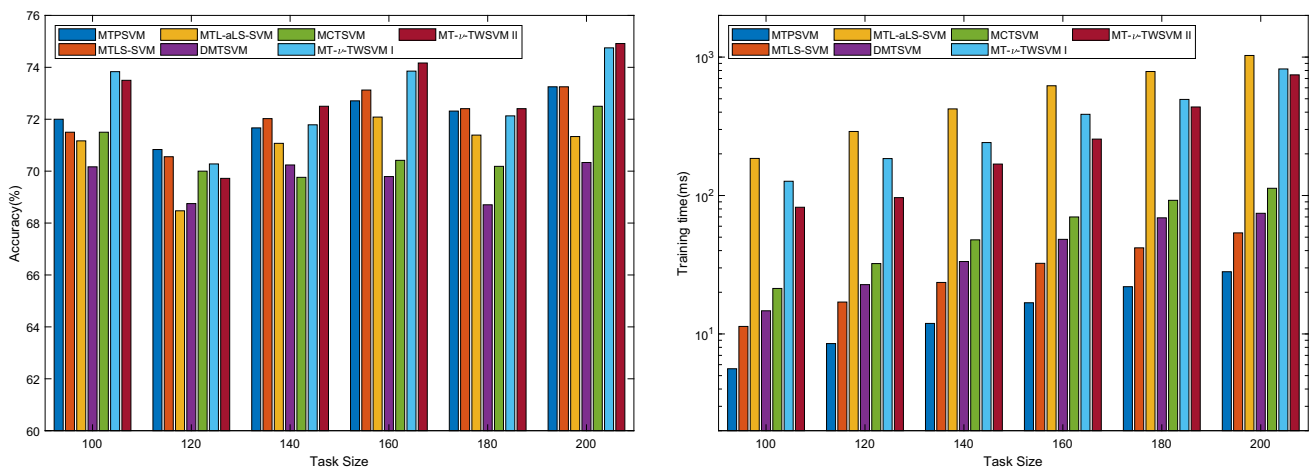


Fig. 6 Performance comparison between MTL methods on *Emotions* dataset (polynomial kernel)

total, such as *mammals*, *birds*, *insects* and *flowers*. There is a clutter category in this dataset, which can be seen as negative samples. The number of images in each category

ranges from 80 to 827. We select no more than 80 samples from each category. Then, we manually cluster these images into 15 main categories according to the hierarchy



Fig. 7 Samples selected from ten categories in Caltech datasets. Each column of samples belongs to the same main category, but the features of image differ in rows

of category, each category contains three to ten classes of images. Some images are shown in Fig. 7. We note that the images in one column have similar features. But each row belongs to different subclass. Therefore, the recognition tasks of different subclasses belonging to the same category can be regarded as a group of related tasks. Then, we train these tasks simultaneously to evaluate these multi-task learning methods.

As a classical image feature extractor, scale invariant feature transform (SIFT) algorithm [51] is widely used in many computer vision researches before [52–54]. Until few years ago, hand crafted features such as SIFT represented the state of the art for visual content analysis. In particular, SIFT is widely regarded as the gold standard in the context of local feature extraction [55]. In this paper, a fast and dense version of SIFT, called dense-SIFT¹, is used in accompany with Bag of Visual Words (BoVW) method to obtain the vector representation of the images. It is a fast algorithm for the calculation of a large number of SIFT descriptors of densely sampled features of the same scale and orientation.² It not only runs faster than original SIFT feature extractor, but also can generate more feature descriptors. Thus it can provide more information of an image. It is especially important in building the feature vector of an image with BoVW method. The feature vector of a preprocessed image is 1000 dimensions in our experiments. Afterward, the dimensions of those feature vectors are reduced with PCA to capture 97% of the variance. Thus, to reduce the training complexity, the task is to recognize those samples in each subclass. Finally, considering the high dimensional of samples, all

¹ http://www.vlfeat.org/matlab/vl_dsift.html.

² <http://www.vlfeat.org/api/dsift.html>.

experiments on these two datasets are conducted with a polynomial kernel function as described in previous experiments.

Figures 8 and 9 illustrate our experimental results on the Caltech 101 and Caltech 256 datasets. We find that our methods perform better than MTPSVM, MTL-SVM and MTL-aLS-SVM on four categories in Caltech 101 dataset. However, from the training time, multi-task TWSVMs are almost the same. But our methods outperform the DMTSVM and MTCTSVM. In comparison, the MTL-aLS-SVM performs badly on these two metrics in most cases. In addition, our algorithms perform better than other two multi-task TWSVMs on seven categories in Caltech 256 dataset. Our algorithms also perform better than MTL-SVM on six categories. We notice that the training time of the last five algorithms is almost the same in this group of experiments. However, they all need to solve one or two quadratic programming problems instead of one larger linear equation. The MTPSVM and MTL-SVM perform well in terms of the average training time. The reason has been clarified in the previous section. Although the feature vector has been reduced to a low dimension, the dimensions are still high when compared to the number of all sample in most cases. We point out that the number of features is about 300–600 dimensions in these two groups of experiments. In contrast to previous results on benchmark datasets, the ability of our algorithms in dealing with such case may not be so well.

After showing our experimental results, we then have an overview of the accuracy levels other researchers reported when they used Caltech 101 and 256 datasets in Table 2. As we can see, the SVM is applied with a specific feature extractor to evaluate the performance in these researches.

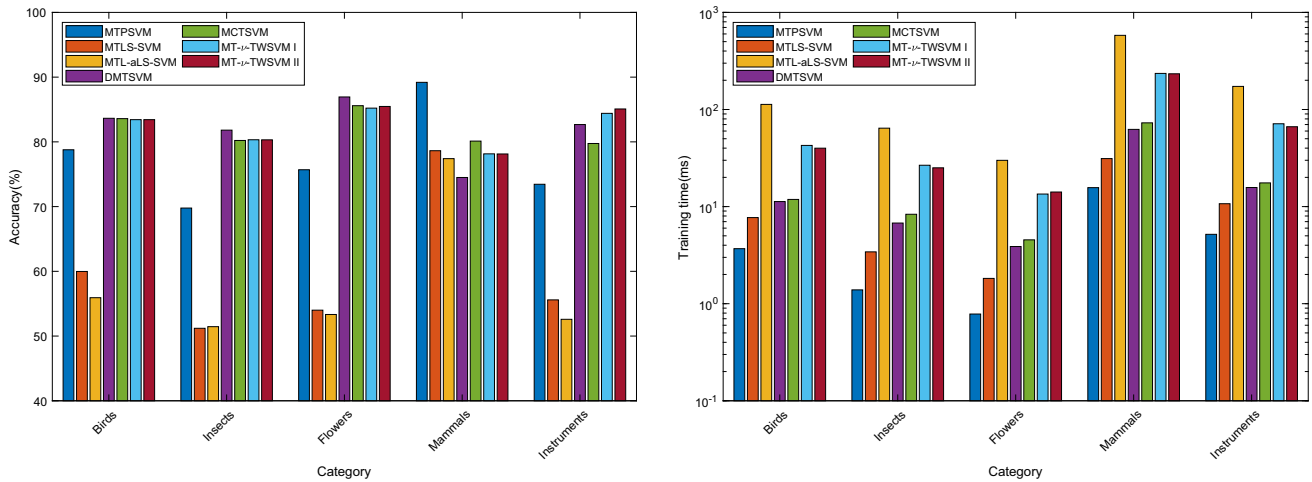


Fig. 8 Performance comparison between multi-task methods on *Caltech 101* image dataset (polynomial kernel)

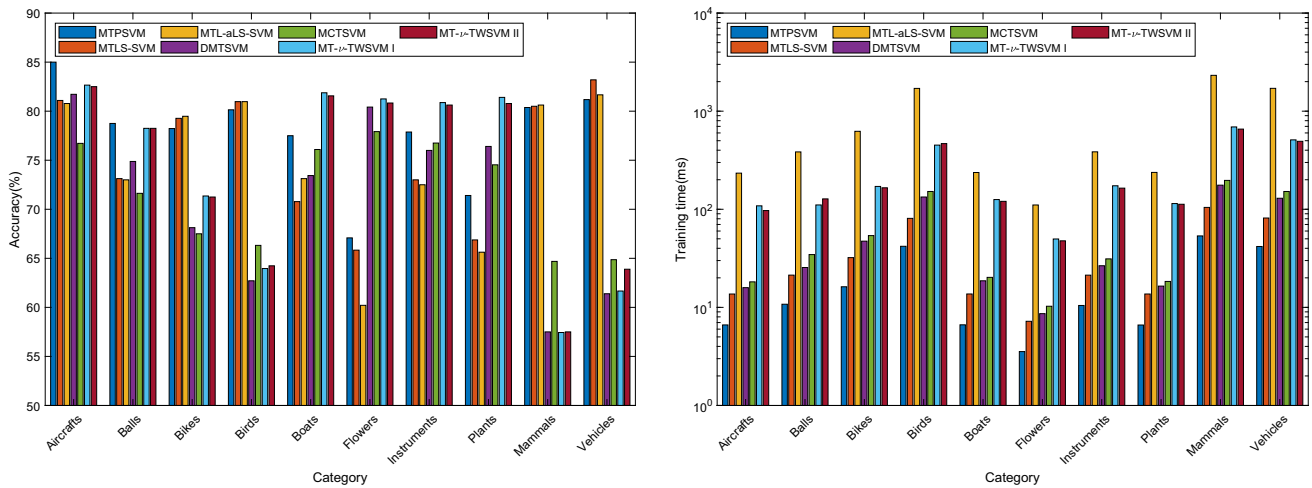


Fig. 9 Performance comparison between multi-task methods on *Caltech 256* image dataset (polynomial kernel)

Table 2 The accuracy levels recently proposed image recognition methods obtained on Caltech 101 and 256 datasets

Method	Caltech 101	Caltech 256
Pyramid SIFT + Linear SVM [56]	80.13	52.59
DRLTP + Linear SVM [57]	80.41	81.89
SC-MCASP + Linear SVM [58]	75.07	37.09
ResFeats-152 + PCA-SVM [59]	94.70	79.50
LEFSI + Linear SVM [60]	85.78	52.90

We note that recently proposed ResFeats-152 + PCA-SVM achieves the best accuracy on both datasets. It uses deep neural network as an image feature extractor and then feeds the preprocessed feature vectors into the SVM. In contrast, the other methods are manually designed feature extractors. The Pyramid SIFT is a feature extractor based on SIFT. In a word, the main difference of above

researches on Caltech datasets lies in the feature extraction method. Compared to previous results, the accuracy level of our methods is comparable to the other method on Caltech 101 dataset. Meanwhile, the accuracy level of our methods is better than the above methods in most cases. The comparison shows the effectiveness of our methods on Caltech datasets.

Finally, to verify our hypothesis that recognizing images belongs to different subclasses but in a common category can be trained simultaneously, Fig. 10 shows the trend of mean accuracy with respect to the parameters μ and ν around the best parameters when the kernel parameters are fixed. The raw data come from the result of MT- ν -TWSVM II on *Caltech 256* image dataset with a RBF kernel function. According to this figure, we can directly know whether the performance is largely affected by the choice of μ or ν . This figure indicates that the performance has strong correlation with the value of μ rather than ν . Our model

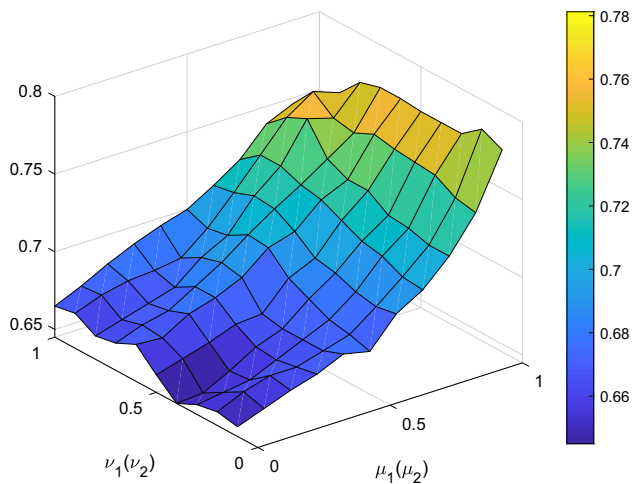


Fig. 10 The effect of parameters μ and ν on the overall performance of MT- ν -TWSVM II

achieves the highest accuracy at a larger value of μ . According to the previous analysis of our model, it means all tasks share two mean hyperplanes and have high correlation. Thus, we should choose a larger parameter μ in the range of $[0, 1]$. This result is in consistent with our hypothesis. It means the tasks selected from *Caltech 256* dataset are related and should be learned jointly rather than separately. However, this is not the case on all the datasets. But we can find the relationships of all tasks according to this figure. Therefore, it provides a better way to choose the best parameters.

7 Conclusion and future work

In this paper, we propose two novel multi-task classifiers, termed as MT- ν -TWSVM I and II, which are natural extension of ν -TWSVM in multi-task learning. Both models inherit the merits of ν -TWSVM and multi-task learning. Our analysis shows that both models share similar properties with ν -TWSVM. The main difference lies in the two Hessian matrices, which model the personality and commonality of all tasks. Unlike original ν -TWSVM, it is the fraction of support vectors of the common task that can be bounded by parameter ν . It overcomes the shortage of DMTSVM and MTCTSVM. The multi-task relationship can be modeled from completely irrelevant to fully relevant in the second model. Therefore, it is more flexible. Experimental results on three benchmark datasets and two image datasets demonstrate the effectiveness and efficiency of our algorithms. Meanwhile, the accuracy levels other researchers obtained on these two image datasets are also discussed. This comparison also clearly confirms that our proposed methods are powerful and consistently outperform the other image classification algorithms.

Finally, our future work will focus on speeding up the training process of multi-task SVM and TWSVMs on large datasets.

Acknowledgements The authors gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation. This work was supported in part by the National Natural Science Foundation of China (No. 11671010), Beijing Natural Science Foundation (No. 4172035) and Chinese People's Liberation Army General Hospital (No. 2017MBD-002).

Compliance with ethical standards

Conflict of Interest The authors declare that they have no conflict of interest.

References

- Burges CJ (1998) A tutorial on support vector machines for pattern recognition. *Data Min Knowl Discov* 2(2):121–167
- Jayadeva, Khemchandani R, Chandra S (2007) Twin support vector machines for pattern classification. *IEEE Trans Pattern Anal Mach Intell* 29(5):905–910
- Kumar MA, Gopal M (2009) Least squares twin support vector machines for pattern classification. *Expert Syst Appl* 36(4):7535–7543
- Shao YH, Zhang CH, Wang XB, Deng NY (2011) Improvements on twin support vector machines. *IEEE Trans Neural Netw* 22(6):962–968
- Qi Z, Tian Y, Shi Y (2013) Robust twin support vector machine for pattern classification. *Pattern Recogn* 46(1):305–316
- Tian Y, Ju X, Qi Z, Shi Y (2014) Improved twin support vector machine. *Sci China Math* 57(2):417–432
- Peng X (2010) A ν -twin support vector machine (ν -TWSVM) classifier and its geometric algorithms. *Inf Sci* 180(20):3863–3875
- Schlkopf B, Smola AJ, Williamson RC, Bartlett PL (2000) New support vector algorithms. *Neural Comput* 12(5):1207–1245
- Xu Y, Yang Z, Pan X (2017) A novel twin support-vector machine with pinball loss. *IEEE Trans Neural Netw* 28(2):359–370
- Xu Y, Li X, Pan X, Yang Z (2018) Asymmetric ν -twin support vector regression. *Neural Comput Appl* 30(12):3799–3814
- Xie X (2018) Regularized multi-view least squares twin support vector machines. *Appl Intell* 48(9):3108–3115
- Xie X, Sun S, Chen H, Qian J (2018) Domain adaptation with twin support vector machines. *Neural Process Lett* 48:1213–1226
- Wang Z, Shao YH, Bai L, Deng NY (2015) Twin support vector machine for clustering. *IEEE Trans Neural Netw* 26(10):2583–2588
- Xie X (2017) Pac-bayes bounds for twin support vector machines. *Neurocomputing* 234(19):137–143
- Pan X, Yang Z, Xu Y, Wang L (2018) Safe screening rules for accelerating twin support vector machine classification. *IEEE Trans Neural Netw* 29(5):1876–1887
- Wang H, Xu Y (2018) Scaling up twin support vector regression with safe screening rule. *Inf Sci* 465:174–190
- Ding S, Zhang N, Zhang X, Wu F (2017) Twin support vector machine: theory, algorithm and applications. *Neural Comput Appl* 28(11):3119–3130
- Ding S, Zhao X, Zhang J, Zhang X, Xue Y (2019) A review on multi-class TWSVM. *Artif Intell Rev* 52(2):775–801

19. Qi K, Liu W, Yang C, Guan Q, Wu H (2017) Multi-task joint sparse and low-rank representation for the scene classification of high-resolution remote sensing image. *Remote Sens* 9(1):10
20. Jeong JY, Jun CH (2018) Variable selection and task grouping for multi-task learning. In: *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery and data mining*, pp 1589–1598
21. Caruana R (1998) Multitask learning. In: *Learning to learn*, pp 95–133
22. Zhang Y, Yang Q (2017) A survey on multi-task learning. *arXiv preprint arXiv:1707.08114*
23. Thung KH, Wee CY (2018) A brief review on multi-task learning. *Multimed Tools Appl* 77(22):29705–29725
24. Caruana R (1993) Multitask learning: a knowledge-based source of inductive bias. In: *Proceedings of the tenth international conference on machine learning (ICML)*, pp 41–48
25. Baxter J (2000) A model of inductive bias learning. *J Artif Intell Res* 12(1):149–198
26. Bakker B, Heskes T (2003) Task clustering and gating for bayesian multitask learning. *J Mach Learn Res* 4:83–99
27. Yu K, Tresp V, Schwaighofer A (2005) Learning Gaussian processes from multiple tasks. In: *Proceedings of the 22nd international conference on machine learning (ICML)*, pp 1012–1019
28. Zhang Y, Yang Q (2018) An overview of multi-task learning. *Natl Sci Rev* 5(1):30–43
29. Evgeniou T, Pontil M (2004) Regularized multi-task learning. In: *Proceedings of the tenth ACM SIGKDD international conference on knowledge discovery and data mining*, pp 109–117
30. Jebara T (2004) Multi-task feature and kernel selection for SVMs. In: *Proceedings of the 21st international conference on machine learning (ICML)*, p 55
31. Micchelli CA, Pontil M (2004) Kernels for multi-task learning. In: *Advances in neural information processing systems (NIPS)*, pp 921–928
32. Liu A, Xu N, Su Y, Lin H, Hao T, Yang Z (2015) Single/multi-view human action recognition via regularized multi-task learning. *Neurocomputing* 151:544–553
33. Cai F, Cherkassky VS (2012) Generalized SMO algorithm for SVM-based multitask learning. *IEEE Trans Neural Netw* 23(6):997–1003
34. Xu S, An X, Qiao X, Zhu L (2014) Multi-task least-squares support vector machines. *Multimed Tools Appl* 71(2):699–715
35. Li Y, Tian X, Song M, Tao D (2015) Multi-task proximal support vector machine. *Pattern Recogn* 48(10):3249–3257
36. Lu L, Lin Q, Pei H, Zhong P (2018) The ALS-SVM based multi-task learning classifiers. *Appl Intell* 48(8):2393–2407
37. Zhu J, Chen N, Xing EP (2011) Infinite latent SVM for classification and multi-task learning. In: *Advances in neural information processing systems (NIPS)*, vol 24, pp 1620–1628
38. Ji Y, Sun S, Lu Y (2012) Multitask multiclass privileged information support vector machines. In: *Proceedings of the 21st international conference on pattern recognition (ICPR)*, pp 2323–2326
39. Zhang J, He Y, Tang J (2018) Multi-view multi-task support vector machine. In: *International conference on computational science (ICCS)*, pp 419–428
40. Markatopoulou F, Mezaris V, Patras I (2016) Online multi-task learning for semantic concept detection in video. In: *IEEE international conference on image processing (ICIP)*, pp 186–190
41. Liang X, Zhu L, Huang D (2017) Multi-task ranking SVM for image cosegmentation. *Neurocomputing* 247:126–136
42. Jia X, Wang S, Yang Y (2018) Least-squares support vector machine for semi-supervised multi-tasking. In: *IEEE 16th international conference on software engineering research, management and applications (SERA)*, pp 79–86
43. Xie X, Sun S (2012) Multitask twin support vector machines. In: *Proceedings of the 19th international conference on neural information processing (ICONIP)*, pp 341–348
44. Xie X, Sun S (2015) Multitask centroid twin support vector machines. *Neurocomputing* 149:1085–1091
45. Mei B, Xu Y (2019) Multi-task least squares twin support vector machine for classification. *Neurocomputing* 338:26–33
46. Dua D, Graff C (2019) UCI machine learning repository. University of California, School of Information and Computer Science, Irvine. <http://archive.ics.uci.edu/ml>
47. Tsoumakas G, Spyromitros-Xioufis E, Vilcek J, Vlahavas I (2011) Mulan: a Java library for multi-label learning. *J Mach Learn Res* 12:2411–2414
48. Li F, Fergus R, Perona P (2004) Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 object categories. In: *2004 Conference on computer vision and pattern recognition workshop*, pp 178–178
49. Li F, Fergus R, Perona P (2006) One-shot learning of object categories. *IEEE Trans Pattern Anal Mach Intell* 28(4):594–611
50. Griffin G, Holub AD, Perona P. The Caltech 256. Caltech Technical Report
51. Lowe D (2004) Distinctive image features from scale-invariant keypoints. *Int J Comput Vis* 60(2):91–110
52. Li F, Fergus P (2005) A Bayesian hierarchical model for learning natural scene categories. In: *IEEE computer society conference on computer vision and pattern recognition (CVPR)*, vol 2, pp 524–531
53. Ehab S, Qasaimeh M (2017) Recent advances in features extraction and description algorithms: a comprehensive survey. In: *IEEE international conference on industrial technology (ICIT)*, pp 1059–63
54. Zheng L, Yang Y, Tian Q (2018) SIFT meets CNN: a decade survey of instance retrieval. *IEEE Trans Pattern Anal Mach Intell* 40(5):1224–1244
55. Baroffio L, Redondi A, Tagliasacchi M, Tubaro S (2016) A survey on compact features for visual content analysis. *APSIPA Trans Signal Inf Process* 5:e13
56. Seidenari L, Serra G, Bagdanov AD, Bimbo AD (2014) Local pyramidal descriptors for image recognition. *IEEE Trans Pattern Anal Mach Intell* 36(5):1033–1040
57. Satpathy A, Jiang X, Eng H-L (2014) LBP-based edge-texture features for object recognition. *IEEE Trans Image Process* 23(5):1953–1964
58. Kim J, Tahboub K, Delp EJ (2017) Spatial pyramid alignment for sparse coding based object classification. In: *2017 IEEE international conference on image processing (ICIP)*, pp 1950–1954
59. Mahmood A, Bennamoun M, An S, Sohel FA (2017) Resfeats: residual network based features for image classification. In *2017 IEEE international conference on image processing (ICIP)*, pp 1597–1601
60. Pan Y, Xia Y, Song Y, Cai W (2018) Locality constrained encoding of frequency and spatial information for image classification. *Multimed Tools Appl* 77(19):24891–24907

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.