Multimed Tools Appl (2014) 71:699–715
DOI 10.1007/s11042-013-1526-5

Multi-task least-squares support vector machines

Shuo Xu · Xin An · Xiaodong Qiao · Lijun Zhu

Published online: 30 May 2013

Abstract There are often the underlying cross relatedness amongst multiple tasks, which is discarded directly by traditional single-task learning methods. Since multi-task learning can exploit these relatedness to further improve the performance, it has attracted extensive attention in many domains including multimedia. It has been shown through a meticulous empirical study that the generalization performance of Least-Squares Support Vector Machine (LS-SVM) is comparable to that of SVM. In order to generalize LS-SVM from single-task to multi-task learning, inspired by the regularized multi-task learning (RMTL), this study proposes a novel multi-task learning approach, multi-task LS-SVM (MTLS-SVM). Similar to LS-SVM, one only solves a convex linear system in the training phrase, too. What's more, we unify the classification and regression problems in an efficient training algorithm, which effectively employs the Krylow methods. Finally, experimental results on *school* and *dermatology* validate the effectiveness of the proposed approach.

Keywords Multi-task learning · Least-Square Support Vector Machine (LS-SVM) · Multi-Task LS-SVM (MTLS-SVM) · Krylow methods

S. Xu · X. Qiao · L. Zhu
Information Technology Supporting Center, Institute of Scientific and Technical Information of China, No. 15 Fuxing Rd., Haidian District, Beijing 100038, People's Republic of China

S. Xu
e-mail: xush@istic.ac.cn

X. Qiao
e-mail: qiaox@istic.ac.cn

L. Zhu
e-mail: zhulj@istic.ac.cn

X. An (✉)
School of Economics and Management, Beijing Forestry University, No. 35 Qinghua East Rd., Haidian District, Beijing 100083, People's Republic of China
e-mail: anxin927@gmail.com

1 Introduction

It is increasing important to learn multiple related tasks in modern applications, ranging from the prediction of test scores in social sciences [3, 6] and the classification of protein functions in systems biology [16] to the categorization of scenes in computer vision [42] and more recently to web and text-image search and ranking [15, 17], web information extraction [19] and labeling music tags [27]. A näve solution is to learn a model for each task separately and then to make predictions using the independent models, i.e., traditional single-task learning methods. This approach is simple and easy to implement, but its performance is unsatisfactory, since it disregards the underlying (potentially non-linear) cross relatedness amongst multiple tasks, that is to say, it does not take advantage of all the information contained in the data.

Intuitively, when there are relations between the tasks to learn, it can be advantageous to learn all tasks simultaneously. This motivated the introduction of the multi-task learning paradigm that exploits the correlations amongst multiple tasks by learning them simultaneously rather than individually [12, 41]. There has been abundant literature on multi-task learning showing that the performance indeed improves when the tasks are related [3, 4, 6, 12, 15, 16, 26, 42]. There have also been various attempts to theoretically study multi-task learning, see [6–10, 26].

Based on the minimization of regularization functionals, the kernel based learning methods, such as Support Vector Machine (SVM) [45, 46], have been successfully used in the past for single-task learning. In order to generalize the kernel based learning methods from single-task to multi-task learning, the regularized multi-task learning (RMTL) is proposed by Pontil & its co-workers [11, 20, 21, 32] by following the intuition of hierarchical Bayes [1, 5, 26], in which the kernel is a matrix-valued function. Similar to SVM, RMTL is also characterized by convex quadratic programming (QP) problem.

By changing the inequality constraints in the SVM by the equality ones, the Least-Squares SVM (LS-SVM) [36, 38, 40] replaces convex QP problem with convex linear system solving problem, thus largely speeding up training. With this advantage, certain problems become much more tractable, model selection using leave-one-out (LOO) procedure for example [13, 14]. Furthermore, it has been shown through a meticulous empirical study that the generalization performance of the LS-SVM is comparable to that of the SVM [44, 52]. Van Gestel et al. [43] also established the equivalence of LS-SVM with a particular form of regularized kernel Fisher discriminant (KFD) method [33]. Therefore, LS-SVM has been attracting extensive attentions during the past few years, such as [2, 49, 50] and references therein.

In this paper, we develop a multi-task learning method for LS-SVM, named as multi-task LS-SVM (MTLS-SVM), for both classification and regression problems. Similar to LS-SVM, one only solves a convex linear system in the training phrase, too. What's more, an efficient training algorithm, which effectively employs the Krylow methods, is given. Our previous work [50, 51] restricts us to (multi-output) regression setting, but in this study we unify the classification and regression problems in an algorithm.

The organization of the rest of this paper is as follows. After LS-SVM for classification and regression problems is briefly described in Section 2, a novel multi-task learning approach, MTLS-SVM, is proposed in Section 3, and some properties and an efficient training algorithm are also described in this section. In Section 4, experimental results on *school* and *dermatology* data sets show that MTLS-SVM performs better than existing multi-task learning methods and largely outperforms single-task LS-SVM, and Section 5 concludes this work.

Notations The following notations will be used in this study. Let $\mathbb{R}$ be the set of real numbers and $\mathbb{R}_+$ the subset of positive ones. For every $n \in \mathbb{N}$, the set of positive integers, we let $\mathbb{N}_n = \{1, 2, \cdots, n\}$. A vector will be written in lower-case letters $\mathbf{x} \in \mathbb{R}^d$ with x_i as its elements. The transpose of $\mathbf{x}$ is written as $\mathbf{x}^{\mathrm{T}}$. The vector $\mathbf{1}_d = [1, 1, \cdots, 1]^{\mathrm{T}} \in \mathbb{R}^d$ and $\mathbf{0}_d = [0, 0, \cdots, 0]^{\mathrm{T}} \in \mathbb{R}^d$. The inner product between vectors $\mathbf{x} \in \mathbb{R}^d$ and $\mathbf{z} \in \mathbb{R}^d$ is defined as $\mathbf{x}^{\mathrm{T}}\mathbf{z} = \sum_{k=1}^{d} x_k z_k$.

Matrices are denoted by capital letters $\mathbf{A} \in \mathbb{R}^{m \times n}$ with $a_{i,j}$ as its elements. The transpose of $\mathbf{A}$ is written as $\mathbf{A}^{\mathrm{T}}$. If $\mathbf{A}$ is an $m \times n$ matrix with all zeros or ones, it is denoted directly as $\mathbf{0}_{m \times n}$ or $\mathbf{1}_{m \times n}$. The identity matrix of dimension $m \times m$ is written as $\mathbf{I}_m$. The function $blockdiag(\mathbf{A}_1, \mathbf{A}_2, \cdots, \mathbf{A}_n)$ or $blockdiag(\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n)$ creates a block diagonal matrix, having $\mathbf{A}_1, \mathbf{A}_2, \cdots, \mathbf{A}_n$ or $\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n$ as main diagonal blocks, with all other blocks being zero matrices/vectors.

$\varphi : \mathbb{R}^d \to \mathbb{R}^{n_h}$ is a mapping to some higher (maybe infinite) dimensional Hilbert space $\mathcal{H}$ (also known as feature space) with n_h dimensions. $\kappa(\cdot, \cdot)$ is a kernel function meeting the Mercer's theorem [45, 46]. The indicator function $\mathrm{sgn}(x) = +1$ if $x \geq 0$, -1 otherwise.

2 Least-Squares Support Vector Machine (LS-SVM)

In this section, we give a brief summary on basic principles of LS-SVM for classification and regression problem. The classification or regression problem is regarded as finding the mapping between an incoming vector $\mathbf{x} \in \mathbb{R}^d$ and an observable output $y \in \{-1, +1\}$ or $y \in \mathbb{R}$ from a given set of independent and identically distributed (i.i.d.) samples, i.e., $\{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$ with $\mathbb{R}^d \times \{-1, +1\}$ or $(\mathbf{x}_i, y_i) \in \mathbb{R}^{d+1}$. For convenience, let $\mathbf{y} = (y_1, y_2, \cdots, y_n)^{\mathrm{T}}$.

2.1 Classification problem

LS-SVM solves the classification problem by finding $\mathbf{w} \in \mathbb{R}^{n_h}$ and $b \in \mathbb{R}$ that minimizes the following objective function with constraint [38, 40]:

$$\min \mathcal{J}(\mathbf{w}, \boldsymbol{\xi}) = \frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} + \gamma \frac{1}{2}\boldsymbol{\xi}^{\mathrm{T}}\boldsymbol{\xi} \tag{1}$$

$$\text{s.t.} \quad \mathbf{Z}^{\mathrm{T}}\mathbf{w} + b\mathbf{y} = \mathbf{1}_n - \boldsymbol{\xi} \tag{2}$$

where $\mathbf{Z} = (y_1\varphi(\mathbf{x}_1), y_2\varphi(\mathbf{x}_2), \cdots, y_n\varphi(\mathbf{x}_n)) \in \mathbb{R}^{n_h \times n}$, $\boldsymbol{\xi} = (\xi_1, \xi_2, \cdots, \xi_n)^{\mathrm{T}} \in \mathbb{R}^n$ is a vector consisting of slack variables, and $\gamma \in \mathbb{R}_+$ is a positive real regularized parameter.

The Lagrangian function for the problem (1) and (2) is

$$\mathcal{L}(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}) = \mathcal{J}(\mathbf{w}, \boldsymbol{\xi}) - \boldsymbol{\alpha}^{\mathrm{T}}(\mathbf{Z}^{\mathrm{T}}\mathbf{w} + b\mathbf{y} - \mathbf{1}_n + \boldsymbol{\xi}) \tag{3}$$

where $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \cdots, \alpha_n)^{\mathrm{T}} \in \mathbb{R}^n$ is a vector consisting of Lagrange multipliers. The Karush–Kuhn–Tucker (KKT) conditions for optimality yield the following set of linear equations:

$$\begin{cases} \dfrac{\partial \mathcal{L}}{\partial \mathbf{w}} = 0 \Rightarrow \mathbf{w} = \mathbf{Z}\boldsymbol{\alpha} \\ \dfrac{\partial \mathcal{L}}{\partial b} = 0 \Rightarrow \boldsymbol{\alpha}^{\mathrm{T}}\mathbf{y} = 0 \\ \dfrac{\partial \mathcal{L}}{\partial \boldsymbol{\xi}} = 0 \Rightarrow \boldsymbol{\alpha} = \gamma\,\boldsymbol{\xi} \\ \dfrac{\partial \mathcal{L}}{\partial \boldsymbol{\alpha}} = 0 \Rightarrow \mathbf{Z}^{\mathrm{T}}\mathbf{w} + b\,\mathbf{y} - \mathbf{1}_n + \boldsymbol{\xi} = \mathbf{0}_n \end{cases} \tag{4}$$

By eliminating $\mathbf{w}$ and $\boldsymbol{\xi}$, one can obtain the following linear system:

$$\begin{bmatrix} 0 & \mathbf{y}^{\mathrm{T}} \\ \mathbf{y} & \mathbf{H} \end{bmatrix} \begin{bmatrix} b \\ \boldsymbol{\alpha} \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{1}_n \end{bmatrix} \tag{5}$$

with the positive definite matrix $\mathbf{H} = \boldsymbol{\Omega} + \frac{1}{\gamma}\mathbf{I}_n \in \mathbb{R}^{n\times n}$. Here, $\boldsymbol{\Omega} = \mathbf{Z}^{\mathrm{T}}\mathbf{Z} \in \mathbb{R}^{n\times n}$ is defined by its elements $\omega_{i,j} = y_i y_j \varphi(\mathbf{x}_i)^{\mathrm{T}}\varphi(\mathbf{x}_j) = y_i y_j \kappa(\mathbf{x}_i, \mathbf{x}_j)$ for $\forall(i, j) \in \mathbb{N}_n \times \mathbb{N}_n$.

Let the solution of (5) be $\boldsymbol{\alpha}^* = (\alpha_1^*, \alpha_2^*, \cdots, \alpha_l^*)^{\mathrm{T}}$ and b^*. Then, the corresponding decision function is

$$\begin{aligned} f(\mathbf{x}) &= \operatorname{sgn}\left(\varphi(\mathbf{x})^{\mathrm{T}}\mathbf{w}^* + b^*\right) = \operatorname{sgn}\left(\varphi(\mathbf{x})^{\mathrm{T}}\mathbf{Z}\boldsymbol{\alpha}^* + b^*\right) \\ &= \operatorname{sgn}\left(\sum_{i=1}^{n} \alpha_i^* \varphi(\mathbf{x})^{\mathrm{T}}\varphi(\mathbf{x}_i) + b^*\right) = \operatorname{sgn}\left(\sum_{i=1}^{n} \alpha_i^* \kappa(\mathbf{x}, \mathbf{x}_i) + b^*\right) \end{aligned} \tag{6}$$

2.2 Regression problem

LS-SVM solves the regression problem by finding $\mathbf{w} \in \mathbb{R}^{n_h}$ and $b \in \mathbb{R}$ that minimizes the following objective function with constraints [36, 40]:

$$\min \mathcal{J}(\mathbf{w}, \boldsymbol{\xi}) = \frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} + \gamma\,\frac{1}{2}\boldsymbol{\xi}^{\mathrm{T}}\boldsymbol{\xi} \tag{7}$$

$$\text{s.t.} \quad \mathbf{y} = \mathbf{Z}^{\mathrm{T}}\mathbf{w} + b\,\mathbf{1}_n + \boldsymbol{\xi} \tag{8}$$

where $\mathbf{Z} = (\varphi(\mathbf{x}_1), \varphi(\mathbf{x}_2), \cdots, \varphi(\mathbf{x}_n)) \in \mathbb{R}^{n_h\times n}$, $\boldsymbol{\xi} = (\xi_1, \xi_2, \cdots, \xi_n)^{\mathrm{T}} \in \mathbb{R}^n$ is a vector consisting of slack variables, and $\gamma \in \mathbb{R}_+$ is a positive real regularized parameter.

The Lagrangian function for the problem (7) and (8) is

$$\mathcal{L}(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}) = \mathcal{J}(\mathbf{w}, \boldsymbol{\xi}) - \boldsymbol{\alpha}^{\mathrm{T}}(\mathbf{Z}^{\mathrm{T}}\mathbf{w} + b\,\mathbf{1}_n + \boldsymbol{\xi} - \mathbf{y}) \tag{9}$$

where $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \cdots, \alpha_n)^{\mathrm{T}} \in \mathbb{R}^n$ is a vector consisting of Lagrange multipliers. The KKT conditions for optimality yield the following set of linear equations:

$$\begin{cases} \dfrac{\partial \mathcal{L}}{\partial \mathbf{w}} = 0 \Rightarrow \mathbf{w} = \mathbf{Z}\boldsymbol{\alpha} \\ \dfrac{\partial \mathcal{L}}{\partial b} = 0 \Rightarrow \boldsymbol{\alpha}^{\mathrm{T}}\mathbf{1}_n = 0 \\ \dfrac{\partial \mathcal{L}}{\partial \boldsymbol{\xi}} = 0 \Rightarrow \boldsymbol{\alpha} = \gamma\,\boldsymbol{\xi} \\ \dfrac{\partial \mathcal{L}}{\partial \boldsymbol{\alpha}} = 0 \Rightarrow \mathbf{Z}^{\mathrm{T}}\mathbf{w} + b\,\mathbf{1}_n + \boldsymbol{\xi} - \mathbf{y} = \mathbf{0}_n \end{cases} \tag{10}$$

By eliminating $\mathbf{w}$ and $\boldsymbol{\xi}$, one can obtain the following linear system:

$$\begin{bmatrix} 0 & \mathbf{1}_n^{\mathrm{T}} \\ \mathbf{1}_n & \mathbf{H} \end{bmatrix} \begin{bmatrix} b \\ \boldsymbol{\alpha} \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{y} \end{bmatrix} \tag{11}$$

with the positive definite matrix $\mathbf{H} = \Omega + \frac{1}{\gamma} I_n \in \mathbb{R}^{n\times n}$. Here, $\Omega = \mathbf{Z}^{\mathrm{T}}\mathbf{Z} \in \mathbb{R}^{n\times n}$ is defined by its elements $\omega_{i,j} = \varphi(\mathbf{x}_i)^{\mathrm{T}}\varphi(\mathbf{x}_j) = \kappa(\mathbf{x}_i, \mathbf{x}_j)$ for $\forall(i, j) \in \mathbb{N}_n \times \mathbb{N}_n$.

Let the solution of (11) be $\boldsymbol{\alpha}^* = (\alpha_1^*, \alpha_2^*, \cdots, \alpha_l^*)^{\mathrm{T}}$ and b^*. Then, the corresponding decision function is

$$\begin{aligned} f(\mathbf{x}) &= \varphi(\mathbf{x})^{\mathrm{T}}\mathbf{w}^* + b^* = \varphi(\mathbf{x})^{\mathrm{T}}\mathbf{Z}\boldsymbol{\alpha}^* + b^* \\ &= \sum_{i=1}^{n} \alpha_i^* \varphi(\mathbf{x})^{\mathrm{T}}\varphi(\mathbf{x}_i) + b^* = \sum_{i=1}^{n} \alpha_i^* \kappa(\mathbf{x}, \mathbf{x}_i) + b^* \end{aligned} \tag{12}$$

2.3 Efficient training algorithm

On closer examination, one can easily find that it is very difficult to solve directly the linear system (5) or (11), since their coefficient matrix are not positive definite. This can be overcome by reformulating (5) or (11) into the following one [39, 40]

$$\begin{bmatrix} s & \mathbf{0}_n^{\mathrm{T}} \\ \mathbf{0}_n & \mathbf{H} \end{bmatrix} \begin{bmatrix} b \\ \boldsymbol{\alpha} + b\mathbf{H}^{-1}\mathbf{d}_1 \end{bmatrix} = \begin{bmatrix} \mathbf{d}_1^{\mathrm{T}}\mathbf{H}^{-1}\mathbf{d}_2 \\ \mathbf{d}_2 \end{bmatrix} \tag{13}$$

where $s = \mathbf{d}_1^{\mathrm{T}}\mathbf{H}^{-1}\mathbf{d}_1 \in \mathbb{R}_+$, $\mathbf{d}_1 = \mathbf{y}/\mathbf{1}_n$ and $\mathbf{d}_2 = \mathbf{1}_n/\mathbf{y}$ for the classification/regression problem. This new linear system (13) is positive definite, which opens many opportunities for using fast and efficient numerical optimization methods. In fact, the solution of the system (5) or (11) can be found in the following three steps [39, 40]:

1. Solve $\boldsymbol{\eta}$, $\boldsymbol{\nu}$ from $\mathbf{H}\boldsymbol{\eta} = \mathbf{d}_1$ and $\mathbf{H}\boldsymbol{\nu} = \mathbf{d}_2$, respectively. Let the corresponding solution be $\boldsymbol{\eta}^*$, $\boldsymbol{\nu}^*$;
2. Compute $s = \mathbf{d}_1^{\mathrm{T}}\boldsymbol{\eta}^*$;
3. Find solution: $b^* = \boldsymbol{\eta}^{*\mathrm{T}}\mathbf{d}_2/s$, $\boldsymbol{\alpha}^* = \boldsymbol{\nu}^* - b^*\boldsymbol{\eta}^*$.

Therefore, the solution of the training procedure can be found by solving two sets of linear equations with the same positive definite coefficient matrix $\mathbf{H} \in \mathbb{R}^{n\times n}$. Since $\mathbf{H}$ is symmetric positive-definite, one typically first finds the Cholesky decomposition $\mathbf{H} = \mathbf{L}\mathbf{L}^{\mathrm{T}}$ [24, 34, 35]. Then since $\mathbf{L}$ is lower triangular, solving the system is simply a matter of applying forward and backward substitution. Other commonly used methods include the conjugate gradient, single value decomposition (SVD) or eigendecomposition, etc.

3 Multi-Task LS-SVM (MTLS-SVM)

Suppose we have m learning tasks. For $\forall i \in \mathbb{N}_m$, we have n_i training data $\{\mathbf{x}_{i,j}, y_{i,j}\}_{j=1}^{n_i}$, where $\mathbf{x}_{i,j} \in \mathbb{R}^d$ and $y_{i,j} \in \{-1, +1\}$ for classification problem or $y_{i,j} \in \mathbb{R}$ for regression problem. Thus, we have $n = \sum_{i=1}^{m} n_i$ training data. For convenience, let $\mathbf{y} = (\mathbf{y}_1^{\mathrm{T}}, \mathbf{y}_2^{\mathrm{T}}, \cdots, \mathbf{y}_m^{\mathrm{T}})^{\mathrm{T}}$ with $\mathbf{y}_i = (y_{i,1}, y_{i,2}, \cdots, y_{i,n_i})^{\mathrm{T}}$ for $\forall i \in \mathbb{N}_m$.

In order to formulate the intuition of Hierarchical Bayes [1, 5, 26], we assume all $\mathbf{w}_i \in \mathbb{R}^{n_h} (\forall i \in \mathbb{N}_m)$ can be written as $\mathbf{w}_i = \mathbf{w}_0 + \mathbf{v}_i$, where the vectors $\mathbf{v}_i \in \mathbb{R}^{n_h}$ are "small" when the different tasks are similar to each other, otherwise the mean vector $\mathbf{w}_0 \in \mathbb{R}^{n_h}$ are "small". That is to say, $\mathbf{w}_0$ carries the information of the commonality and $\mathbf{v}_i (i \in \mathbb{N}_m)$ carries the information of the specialty. Figure 1 illustrates the intuition underling the MTLS-SVM.

3.1 Classification problem

MTLS-SVM solves the classification problem by finding $\mathbf{w}_0 \in \mathbb{R}^{n_h}$, $\{\mathbf{v}_i\}_{i=1}^{m} \in \mathbb{R}^{n_h \times m}$, and $\mathbf{b} = (b_1, b_2, \cdots, b_m)^{\mathrm{T}} \in \mathbb{R}^m$ simultaneously that minimizes the following objective function with constraints:

$$\min \mathcal{J}(\mathbf{w}_0, \{\mathbf{v}_i\}_{i=1}^{m}, \{\boldsymbol{\xi}_i\}_{i=1}^{m}) = \frac{1}{2}\mathbf{w}_0^{\mathrm{T}}\mathbf{w}_0 + \frac{1}{2}\frac{\lambda}{m}\sum_{i=1}^{m}\mathbf{v}_i^{\mathrm{T}}\mathbf{v}_i + \gamma\frac{1}{2}\sum_{i=1}^{m}\boldsymbol{\xi}_i^{\mathrm{T}}\boldsymbol{\xi}_i \tag{14}$$

$$\text{s.t.} \quad \mathbf{Z}_i^{\mathrm{T}}(\mathbf{w}_0 + \mathbf{v}_i) + b_i\mathbf{y}_i = \mathbf{1}_{n_i} - \boldsymbol{\xi}_i, i \in \mathbb{N}_m \tag{15}$$

where for $\forall i \in \mathbb{N}_m$, $\boldsymbol{\xi}_i = (\xi_{i,1}, \xi_{i,2}, \cdots, \xi_{i,n_i})^{\mathrm{T}} \in \mathbb{R}^{n_i}$, $\mathbf{Z}_i = (y_{i,1}\varphi(\mathbf{x}_{i,1}), y_{i,2}\varphi(\mathbf{x}_{i,2}), \cdots, y_{i,n_i}\varphi(\mathbf{x}_{i,n_i})) \in \mathbb{R}^{n_h \times n_i}$, and $\lambda, \gamma \in \mathbb{R}_+$ are two positive real regularized parameters. And we let $\mathbf{Z} = (\mathbf{Z}_1, \mathbf{Z}_2, \cdots, \mathbf{Z}_m) \in \mathbb{R}^{n_h \times n}$.

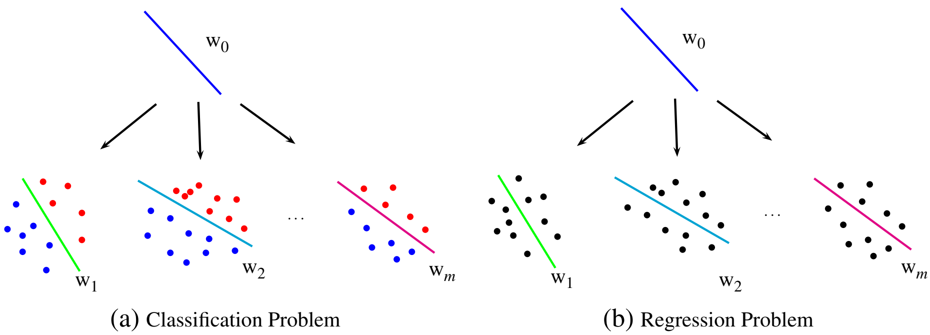

Fig. 1 Illustration of the intuition underlying the MTLS-SVM

The Lagrangian function for the problem (14) and (15) is

$$\begin{aligned}
&\mathcal{L}(\mathbf{w}_0, \{\mathbf{v}_i\}_{i=1}^m, \mathbf{b}, \{\boldsymbol{\xi}_i\}_{i=1}^m, \{\boldsymbol{\alpha}_i\}_{i=1}^m) \\
&\quad = \mathcal{J}(\mathbf{w}_0, \{\mathbf{v}_i\}_{i=1}^m, \{\boldsymbol{\xi}_i\}_{i=1}^m) - \sum_{i=1}^m \boldsymbol{\alpha}_i^{\mathrm{T}}(\mathbf{Z}_i^{\mathrm{T}}(\mathbf{w}_0 + \mathbf{v}_i) + b_i \mathbf{y}_i - \mathbf{1}_{n_i} + \boldsymbol{\xi}_i)
\end{aligned} \tag{16}$$

where $\forall i \in \mathbb{N}_m, \boldsymbol{\alpha}_i = (\alpha_{i,1}, \alpha_{i,2}, \cdots, \alpha_{i,n_i})^{\mathrm{T}}$ consists of Lagrange multipliers. And we let $\boldsymbol{\alpha} = (\boldsymbol{\alpha}_1^{\mathrm{T}}, \boldsymbol{\alpha}_2^{\mathrm{T}}, \cdots, \boldsymbol{\alpha}_m^{\mathrm{T}})^{\mathrm{T}} \in \mathbb{R}^n$. The KKT conditions for optimality yield the following set of linear equations:

$$\begin{cases}
\dfrac{\partial \mathcal{L}}{\partial \mathbf{w}_0} = 0 \Rightarrow \mathbf{w}_0 = \mathbf{Z}\boldsymbol{\alpha} \\
\dfrac{\partial \mathcal{L}}{\partial \mathbf{v}_i} = 0 \ \Rightarrow \mathbf{v}_i = \dfrac{m}{\lambda}\mathbf{Z}_i\boldsymbol{\alpha}_i, \forall i \in \mathbb{N}_m \\
\dfrac{\partial \mathcal{L}}{\partial b_i} = 0 \ \Rightarrow \boldsymbol{\alpha}_i^{\mathrm{T}}\mathbf{y}_i = 0, \forall i \in \mathbb{N}_m \\
\dfrac{\partial \mathcal{L}}{\partial \boldsymbol{\xi}_i} = 0 \ \Rightarrow \boldsymbol{\alpha}_i = \gamma\, \boldsymbol{\xi}_i, \forall i \in \mathbb{N}_m \\
\dfrac{\partial \mathcal{L}}{\partial \boldsymbol{\alpha}_i} = 0 \ \Rightarrow \mathbf{Z}_i^{\mathrm{T}}(\mathbf{w}_0 + \mathbf{v}_i) + b_i \mathbf{y}_i - \mathbf{1}_{n_i} + \boldsymbol{\xi}_i = \mathbf{0}_{n_i}, \forall i \in \mathbb{N}_m
\end{cases} \tag{17}$$

Similar to LS-SVM for the classification problem in Section 2.1, by eliminating $\mathbf{w}_0, \{\mathbf{v}_i\}_{i=1}^m$ and $\{\boldsymbol{\xi}_i\}_{i=1}^m$, one can obtain the following linear system:

$$\begin{bmatrix} \mathbf{0}_{m\times m} & \mathbf{A}^{\mathrm{T}} \\ \mathbf{A} & \mathbf{H} \end{bmatrix} \begin{bmatrix} \mathbf{b} \\ \boldsymbol{\alpha} \end{bmatrix} = \begin{bmatrix} \mathbf{0}_m \\ \mathbf{1}_n \end{bmatrix} \tag{18}$$

where $\mathbf{A} = blockdiag(\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_m) \in \{-1, +1\}^{n\times m}$, the positive definite matrix $\mathbf{H} = \Omega + \frac{1}{\gamma}\mathbf{I}_n + \frac{m}{\lambda}\mathbf{B} \in \mathbb{R}^{n\times n}$, $\Omega = \mathbf{Z}^{\mathrm{T}}\mathbf{Z} \in \mathbb{R}^{n\times n}$, and $\mathbf{B} = blockdiag(\Omega_1, \Omega_2, \cdots, \Omega_m) \in \mathbb{R}^{n\times n}$ with $\Omega_i = \mathbf{Z}_i^{\mathrm{T}}\mathbf{Z}_i \in \mathbb{R}^{n_i\times n_i}$.

Let the solution of (18) be $\boldsymbol{\alpha}^* = (\boldsymbol{\alpha}_1^{*\mathrm{T}}, \boldsymbol{\alpha}_2^{*\mathrm{T}}, \cdots, \boldsymbol{\alpha}_m^{*\mathrm{T}})^{\mathrm{T}}$ with $\boldsymbol{\alpha}_i^* = (\alpha_{i,1}^*, \alpha_{i,2}^*, \cdots, \alpha_{i,n_i}^*)^{\mathrm{T}}$ and $\mathbf{b}^* = (b_1^*, b_2^*, \cdots, b_m^*)^{\mathrm{T}}$. Then, the corresponding decision function for the task $i \in \mathbb{N}_m$ is

$$\begin{aligned}
f_i(\mathbf{x}) &= \operatorname{sgn}\left(\varphi(\mathbf{x})^{\mathrm{T}}(\mathbf{w}_0^* + \mathbf{v}_i^*) + b_i^*\right) \\
&= \operatorname{sgn}\left(\varphi(\mathbf{x})^{\mathrm{T}}(\mathbf{Z}\boldsymbol{\alpha}^* + \frac{m}{\lambda}\mathbf{Z}_i\boldsymbol{\alpha}_i^*) + b_i^*\right) \\
&= \operatorname{sgn}\left(\sum_{i'=1}^m \sum_{j=1}^{n_{i'}} \alpha_{i',j}^* \kappa(\mathbf{x}_{i',j}, \mathbf{x}) + \frac{m}{\lambda}\sum_{j=1}^{n_i} \alpha_{i,j}^* \kappa(\mathbf{x}_{i,j}, \mathbf{x}) + b_i^*\right)
\end{aligned} \tag{19}$$

3.2 Regression problem

MTLS-SVM solves the regression problem by finding $\mathbf{w}_0 \in \mathbb{R}^{n_h}$, $\{\mathbf{v}_i\}_{i=1}^m \in \mathbb{R}^{n_h \times m}$, and $\mathbf{b} = (b_1, b_2, \cdots, b_m)^{\mathrm{T}} \in \mathbb{R}^m$ simultaneously that minimizes the following objective function with constraints:

$$\min \mathcal{J}(\mathbf{w}_0, \{\mathbf{v}_i\}_{i=1}^m, \{\boldsymbol{\xi}_i\}_{i=1}^m) = \frac{1}{2}\mathbf{w}_0^{\mathrm{T}}\mathbf{w}_0 + \frac{1}{2}\frac{\lambda}{m}\sum_{i=1}^m \mathbf{v}_i^{\mathrm{T}}\mathbf{v}_i + \gamma\frac{1}{2}\sum_{i=1}^m \boldsymbol{\xi}_i^{\mathrm{T}}\boldsymbol{\xi}_i \tag{20}$$

$$\text{s.t.} \quad \mathbf{y}_i = \mathbf{Z}_i^{\mathrm{T}}(\mathbf{w}_0 + \mathbf{v}_i) + b_i\mathbf{1}_{n_i} + \boldsymbol{\xi}_i, i \in \mathbb{N}_m \tag{21}$$

where for $\forall i \in \mathbb{N}_m$, $\boldsymbol{\xi}_i = (\xi_{i,1}, \xi_{i,2}, \cdots, \xi_{i,n_i})^{\mathrm{T}} \in \mathbb{R}^{n_i}$, $\mathbf{Z}_i = (\varphi(\mathbf{x}_{i,1}), \varphi(\mathbf{x}_{i,2}), \cdots, \varphi(\mathbf{x}_{i,n_i})) \in \mathbb{R}^{n_h \times n_i}$, and $\lambda, \gamma \in \mathbb{R}_+$ are two positive real regularized parameters. And we let $\mathbf{Z} = (\mathbf{Z}_1, \mathbf{Z}_2, \cdots, \mathbf{Z}_m) \in \mathbb{R}^{n_h \times n}$.

The Lagrangian function for the problem (20) and (21) is

$$\begin{aligned} &\mathcal{L}(\mathbf{w}_0, \{\mathbf{v}_i\}_{i=1}^m, \mathbf{b}, \{\boldsymbol{\xi}_i\}_{i=1}^m, \{\boldsymbol{\alpha}_i\}_{i=1}^m) \\ &\quad = \mathcal{J}(\mathbf{w}_0, \{\mathbf{v}_i\}_{i=1}^m, \{\boldsymbol{\xi}_i\}_{i=1}^m) - \sum_{i=1}^m \boldsymbol{\alpha}_i^{\mathrm{T}}(\mathbf{Z}_i^{\mathrm{T}}(\mathbf{w}_0 + \mathbf{v}_i) + b_i\mathbf{1}_{n_i} + \boldsymbol{\xi}_i - \mathbf{y}_i) \end{aligned} \tag{22}$$

where $\forall i \in \mathbb{N}_m, \boldsymbol{\alpha}_i = (\alpha_{i,1}, \alpha_{i,2}, \cdots, \alpha_{i,n_i})^{\mathrm{T}}$ consists of Lagrange multipliers. And we let $\boldsymbol{\alpha} = (\boldsymbol{\alpha}_1^{\mathrm{T}}, \boldsymbol{\alpha}_2^{\mathrm{T}}, \cdots, \boldsymbol{\alpha}_m^{\mathrm{T}})^{\mathrm{T}} \in \mathbb{R}^n$. The KKT conditions for optimality yield the following set of linear equations:

$$\begin{cases} \dfrac{\partial \mathcal{L}}{\partial \mathbf{w}_0} = 0 \Rightarrow \mathbf{w}_0 = \mathbf{Z}\boldsymbol{\alpha} \\ \dfrac{\partial \mathcal{L}}{\partial \mathbf{v}_i} = 0 \Rightarrow \mathbf{v}_i = \dfrac{m}{\lambda}\mathbf{Z}_i\boldsymbol{\alpha}_i, \forall i \in \mathbb{N}_m \\ \dfrac{\partial \mathcal{L}}{\partial b_i} = 0 \Rightarrow \boldsymbol{\alpha}_i^{\mathrm{T}}\mathbf{1}_{n_i} = 0, \forall i \in \mathbb{N}_m \\ \dfrac{\partial \mathcal{L}}{\partial \boldsymbol{\xi}_i} = 0 \Rightarrow \boldsymbol{\alpha}_i = \gamma\boldsymbol{\xi}_i, \forall i \in \mathbb{N}_m \\ \dfrac{\partial \mathcal{L}}{\partial \boldsymbol{\alpha}_i} = 0 \Rightarrow \mathbf{Z}_i^{\mathrm{T}}(\mathbf{w}_0 + \mathbf{v}_i) + b_i\mathbf{1}_{n_i} + \boldsymbol{\xi}_i - \mathbf{y}_i = \mathbf{0}_{n_i}, \forall i \in \mathbb{N}_m \end{cases} \tag{23}$$

Similar to LS-SVM for the regression problem in Section 2.2, by eliminating $\mathbf{w}_0$, $\{\mathbf{v}_i\}_{i=1}^m$ and $\{\boldsymbol{\xi}_i\}_{i=1}^m$, one can obtain the following linear system:

$$\begin{bmatrix} \mathbf{0}_{m\times m} & \mathbf{A}^{\mathrm{T}} \\ \mathbf{A} & \mathbf{H} \end{bmatrix} \begin{bmatrix} \mathbf{b} \\ \boldsymbol{\alpha} \end{bmatrix} = \begin{bmatrix} \mathbf{0}_m \\ \mathbf{y} \end{bmatrix} \tag{24}$$

where $\mathbf{A} = blockdiag(\mathbf{1}_{n_1}, \mathbf{1}_{n_2}, \cdots, \mathbf{1}_{n_m}) \in \mathbb{R}^{n\times m}$, the positive definite matrix $\mathbf{H} = \Omega + \frac{1}{\gamma}\mathbf{I}_n + \frac{m}{\lambda}\mathbf{B} \in \mathbb{R}^{n\times n}$, $\Omega = \mathbf{Z}^{\mathrm{T}}\mathbf{Z} \in \mathbb{R}^{n\times n}$, and $\mathbf{B} = blockdiag(\Omega_1, \Omega_2, \cdots, \Omega_m) \in \mathbb{R}^{n\times n}$ with $\Omega_i = \mathbf{Z}_i^{\mathrm{T}}\mathbf{Z}_i \in \mathbb{R}^{n_i\times n_i}$.

Let the solution of (24) be $\boldsymbol{\alpha}^* = (\boldsymbol{\alpha}_1^{*\mathrm{T}}, \boldsymbol{\alpha}_2^{*\mathrm{T}}, \cdots, \boldsymbol{\alpha}_m^{*\mathrm{T}})^\mathrm{T}$ with $\boldsymbol{\alpha}_i^* = (\alpha_{i,1}^*, \alpha_{i,2}^*, \cdots, \alpha_{i,n_i}^*)^\mathrm{T}$ and $\mathbf{b}^* = (b_1^*, b_2^*, \cdots, b_m^*)^\mathrm{T}$. Then, the corresponding decision function for the task $i \in \mathbb{N}_m$ is

$$\begin{aligned} f_i(\mathbf{x}) &= \varphi(\mathbf{x})^\mathrm{T}(\mathbf{w}_0^* + \mathbf{v}_i^*) + b_i^* \\ &= \varphi(\mathbf{x})^\mathrm{T}\left(\mathbf{Z}\boldsymbol{\alpha}^* + \frac{m}{\lambda}\mathbf{Z}_i\boldsymbol{\alpha}_i^*\right) + b_i^* \\ &= \sum_{i'=1}^{m}\sum_{j=1}^{n_{i'}} \alpha_{i',j}^* \kappa(\mathbf{x}_{i',j}, \mathbf{x}) + \frac{m}{\lambda}\sum_{j=1}^{n_i} \alpha_{i,j}^* \kappa(\mathbf{x}_{i,j}, \mathbf{x}) + b_i^* \end{aligned} \tag{25}$$

3.3 Some properties

It is easy to see from (17) and (23) that the mean vector $\mathbf{w}_0 \in \mathbb{R}^{n_h}$ and the vectors $\{\mathbf{v}_i\}_{i=1}^m \in \mathbb{R}^{n_h \times m}$ meet the following relationship:

$$\mathbf{w}_0 = \frac{\lambda}{m}\sum_{i=1}^{m}\mathbf{v}_i \tag{26}$$

In other words, $\mathbf{w}_0$ is a linear combination of $\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_m$. As a 3-task learning example, Fig. 2 visualizes the relationship between $\mathbf{w}_0$ and $\{\mathbf{v}_i\}_{i=1}^3$. Since for $\forall i \in \mathbb{N}_m$, $\mathbf{w}_i$ is assumed to be $\mathbf{w}_i = \mathbf{w}_0 + \mathbf{v}_i$, $\mathbf{w}_i$ can also be expressed as a linear combination of $\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_m$. This suggests that one can obtain an equivalent optimization problem with constraints involving only the $\{\mathbf{v}_i\}_{i=1}^m$ and $\mathbf{b}$ for the respective classification and regression problems as follows.

$$\min \mathcal{J}(\{\mathbf{v}_i\}_{i=1}^m, \{\boldsymbol{\xi}_i\}_{i=1}^m) = \frac{1}{2}\frac{\lambda^2}{m^2}\sum_{i=1}^{m}\sum_{j=1}^{m}\mathbf{v}_i^\mathrm{T}\mathbf{v}_j + \frac{1}{2}\frac{\lambda}{m}\sum_{i=1}^{m}\mathbf{v}_i^\mathrm{T}\mathbf{v}_i + \gamma\frac{1}{2}\sum_{i=1}^{m}\boldsymbol{\xi}_i^\mathrm{T}\boldsymbol{\xi}_i \tag{27}$$

$$\text{s.t.} \begin{cases} \mathbf{Z}_i^\mathrm{T}\left(\dfrac{\lambda}{m}\displaystyle\sum_{i'=1}^{m}\mathbf{v}_{i'} + \mathbf{v}_i\right) + b_i\mathbf{y}_i = \mathbf{1}_{n_i} - \boldsymbol{\xi}_i, i \in \mathbb{N}_m, \text{ classification problem} \\ \mathbf{y}_i = \mathbf{Z}_i^\mathrm{T}\left(\dfrac{\lambda}{m}\displaystyle\sum_{i'=1}^{m}\mathbf{v}_{i'} + \mathbf{v}_i\right) + b_i\mathbf{1}_{n_i} + \boldsymbol{\xi}_i, i \in \mathbb{N}_m, \text{ regression problem} \end{cases} \tag{28}$$

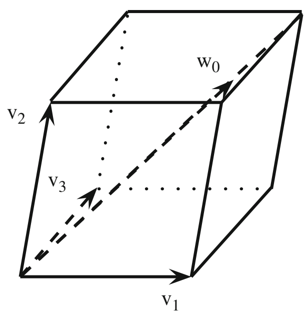

Fig. 2 The relationship between $\mathbf{w}_0$ and $\{\mathbf{v}_i\}_{i=1}^3$

From (27), one can see that our MTLS-SVM tries to find a trade off between small size vectors for each task, $\sum_{i=1}^{m} \mathbf{v}_i^{\mathrm{T}} \mathbf{v}_i$, and closeness of all vectors to the average vector, $\sum_{i=1}^{m} \sum_{j=1}^{m} \mathbf{v}_i^{\mathrm{T}} \mathbf{v}_j$. But (1) and (7) only tries to find small size vectors for each task, which results in decoupling between the different tasks.

Similar to LS-SVM again, one drawback of MTLS-SVM in comparison with RMTL is the lack of sparseness in the solution error, which is clear from the fact that $\boldsymbol{\alpha}_i = \gamma\, \boldsymbol{\xi}_i$ $(\forall i \in \mathbb{N}_m)$. However, there are several possible ways to can sparsify the MTLS-SVM. For example, the simple heuristic is to remove the samples corresponding to small $|\alpha_{i,j}|$, since it is very possible that these samples are less relevant for the construction of the model, in analogy with RMTL where zero $\alpha_{i,j}$ values do not contribute the model. For more elaborate and detailed surveys on sparseness by pruning, we refer the readers to [40].

3.4 Efficient training algorithm

The matrix in (18) or (24) is of dimension $(n+m) \times (n+m)$, and it is usually density. For large values of $n+m$, this matrix cannot be stored in memory, therefore an iterative solution method for solving (18) or (24) is preferred. For now, there are many iterative approaches to solve a set of linear equations [24, 34, 35], such as Cholesky factorization, successive overrelaxation (SOR), Krylow methods (conjugate gradient, block-conjugate gradient), and so on. It has been shown that Krylow methods show the best performance for large data sets [25]. However, Krylow methods are only applicable to solving $\mathcal{A}\mathbf{x} = \mathcal{B}$ with $\mathcal{A} \in \mathbb{R}^{n\times n}$ symmetric positive definite and $\mathcal{B} \in \mathbb{R}^n$. Since the matrix in (18) or (24) is symmetric, but not positive definite, it cannot be solved in this form by Krylow methods.

Both (18) and (24) are of the form

$$\begin{bmatrix} \mathbf{0}_{m\times m} & \mathbf{A}^{\mathrm{T}} \\ \mathbf{A} & \mathbf{H} \end{bmatrix} \begin{bmatrix} \mathbf{b} \\ \boldsymbol{\alpha} \end{bmatrix} = \begin{bmatrix} \mathbf{d}_1 \\ \mathbf{d}_2 \end{bmatrix} \tag{29}$$

where $\mathbf{d}_1 = \mathbf{0}_m$, and $\mathbf{d}_2 = \mathbf{1}_n/\mathbf{y}$ for the classification/regression problem. Equation (29) is equivalent to solving

$$\begin{bmatrix} \mathbf{S} & \mathbf{0}_{n\times n} \\ \mathbf{0}_{m\times m} & \mathbf{H} \end{bmatrix} \begin{bmatrix} \mathbf{b} \\ \mathbf{H}^{-1}\mathbf{A}\mathbf{b} + \boldsymbol{\alpha} \end{bmatrix} = \begin{bmatrix} -\mathbf{d}_1 + \mathbf{A}^{\mathrm{T}}\mathbf{H}^{-1}\mathbf{d}_2 \\ \mathbf{d}_2 \end{bmatrix} \tag{30}$$

with $\mathbf{S} = \mathbf{A}^{\mathrm{T}}\mathbf{H}^{-1}\mathbf{A} \in \mathbb{R}^{m\times m}$. It is very easy to show that $\mathbf{S}$ is a positive definite matrix. In this way, this new linear system (30) is positive definite, whose solution can be found in the following three steps:

1. Sovle $\boldsymbol{\eta}$, $\boldsymbol{\nu}$ from $\mathbf{H}\boldsymbol{\eta} = \mathbf{A}$ and $\mathbf{H}\boldsymbol{\nu} = \mathbf{d}_2$ with Krylow methods, respectively. Let the corresponding solution be $\boldsymbol{\eta}^*$, $\boldsymbol{\nu}^*$;
2. Calculate $\mathbf{S} = \mathbf{A}^{\mathrm{T}}\boldsymbol{\eta}^*$;
3. Find solution: $\mathbf{b}^* = \mathbf{S}^{-1}\boldsymbol{\eta}^{*\mathrm{T}}\mathbf{d}_2$, $\boldsymbol{\alpha}^* = \boldsymbol{\nu}^* - \boldsymbol{\eta}^*\mathbf{b}^*$.

Therefore, again similar to LS-SVM, the solution of the training procedure can be found by solving two sets of linear equations with the same positive coefficient matrix $\mathbf{H} \in \mathbb{R}^{n\times n}$. What's more, since the number of tasks m is usually very small relative to the number of samples n, one can easily obtain the inverse of $\mathbf{S} \in \mathbb{R}^{m\times m}$ just using matrix multiplications.

4 Experiments and discussions

Whether in (6) and (12) or (19) and (25), the kernel function, which should meet the Mercer's theorem, is involved. There are many many candidates, such as the linear kernel: $\kappa(\mathbf{x}, \mathbf{z}) = \mathbf{x}^{\mathrm{T}}\mathbf{z}$; the polynomial kernel: $\kappa(\mathbf{x}, \mathbf{z}) = \left(p_1\mathbf{x}^{\mathrm{T}}\mathbf{z} + p_2\right)^{p_3}$, $p_1 > 0$; the Gaussian (RBF, radial basis function) kernel: $\kappa(\mathbf{x}, \mathbf{z}) = \exp\left(-p \parallel \mathbf{x} - \mathbf{z} \parallel^2\right)$, $p > 0$; the Sigmoid kernel: $\kappa(\mathbf{x}, \mathbf{z}) = \tanh\left(p_1\mathbf{x}^{\mathrm{T}}\mathbf{z} + p_2\right)$ and so on.

Here, the linear and RBF kernel functions are adopted. The reasons are four-fold: (a) the linear kernel function is a special case of RBF kernel function [29], but the cost of calculation is the lowest, so it is suited to solve large scale problems; (b) The Sigmoid kernel function is not positive definite, and for certain parameters, and the Sigmoid kernel function behaves like RBF kernel function [31]; (c) Relatively, there are more parameters in the polynomial kernel function so that it is more difficult for model selection. In addition, the polynomial kernel function has also numerical difficulties, such as overflow or underflow; (d) The RBF kernel function possesses good smoothness properties, which are usually preferred in the case one does not have a prior knowledge about the problem at hand [23, 37].

Finally, in order to identify proper parameters, the grid search [48] is used. Let $\gamma \in \{2^{-5}, 2^{-3}, \cdots, 2^{15}\}$, $\lambda \in \{2^{-10}, 2^{-8}, \cdots, 2^{10}\}$ and $p \in \{2^{-15}, 2^{-13}, \cdots, 2^{3}\}$. For all possible combinations (γ, λ, p) with RBF kernel function or (γ, λ) with linear kernel function, the explained variance (EV) [6] for *school* data set[1] or average classification error for *dermatology* data set[2] is calculated using LOO procedure. Once the optimal value for γ, λ or p lies at the border of the search space, the search space for the parameter that is at the border is increased by the same multiplicative step as described above ($2^{\pm 2}$). Thus, an optimal triple $(\gamma^*, \lambda^*, p^*)$ or pair (γ^*, λ^*) can be determined. We have implemented all related approaches in MATLAB R2010a on an IBM 3850 M2. The corresponding toolbox can be available from the first author upon request for academic use.

4.1 *School* data set

This data set comes from the Inner London Education Authority (ILEA), consisting of examination records of 15,362 students from 139 secondary schools in years 1985, 1986 and 1987. The goal is to predict the exam scores of the students based on the following inputs in Table 1. The categorical variables are expressed with binary (dummy) variables, so the total number of inputs for each student in each of the schools was 27. Each school is considered to be "one task", hence we have 139 tasks in total.

We randomly split the data into training (75% of the data, hence around 70 students per school on average) and test (the remaining 25% of the data, hence around 40 students per school on average) data. This procedure is repeated 10 times. The EV of the test data is utilized to measure the generalization performance, so that we can have a direct comparison with RMTL [11, 20, 21, 32] and Bayesian multi-task learning (BMTL) [6]. The EV in [6] is defined to be the total variance of the data

[1] *School* data set can be available online from http://multilevel.ioe.ac.uk/intro/datasets.html.

[2] *Dermatology* data set can be available online from http://www.ics.uci.edu/ mlearn/MLRespository.html.

Table 1 Variables in *school* data set and their codings

ID	Description	Coding	Input/output
1	Year	1985 = 1; 1986 = 2; 1987 = 3	Input
2	Exam score	Numeric score	Output
3	% FSM	Percent of students eligible for free school meals	Input
4	% VR1 band	Percent of students in school in VR band 1	Input
5	Gender	Male = 0; Female = 1	Input
6	VR band of student	VR1 = 2; VR2 = 3; VR3 = 1	Input
7	Ethnic group of student	ESWI[a] = 1; African = 2; Arab = 3; Bangladeshi = 4; Caribbean = 5; Greek = 6; Indian = 7; Pakistani = 8; S.E. Asian = 9; Turkish = 10; Other = 11	Input
8	School gender	Mixed = 1; Male = 2; Female = 3	Input
9	School denomination	Maintained = 1; Church of England = 2; Roman Catholic = 3	Input

[a]ESWI: Students born in England, Scotland, Wales or Ireland

minus the sum-squared error on the test set as a percentage of the total data variance, which is a percentage version of the standard R^2 error measure for regression for the test data. Finally, the linear kernel function is used for each of the task in MTLS-SVM, but the RBF kernel function is used in LS-SVM, since LS-SVM with the linear kernel function gives the worse performance, which is not reported here.

The results for this experiment are reported in Table 2. Through comparing columns 1 and other columns in Table 2, one can see the obvious advantage of learning all tasks simultaneously instead of learning them one by one. Furthermore, even MTLS-SVM with the simple linear kernel significantly outperforms LS-SVM with the RBF kernel function. The results from last three columns in Table 2 show the efficiency of our proposed MTLS-SVM method.

4.2 *Dermatology* data set

This data set consists of 366 differential diagnosis of erythemato-squamous in dermatology. The goal is to diagnose one of six dermatological diseases (psoriasis, seboreic dermatitis, lichen planus, pityriasis rosea, cronic dermatitis, and pityriasis rubra pilaris) based on 33 clinical and histopathological attributes. That is to say, this is a multi-class (6-class) problem. As in [4, 22, 28], we convert this problem to 6 binary one-versus-rest classification problems, each of which is considered to be "one task". Hence we have six tasks in total. This data set is divided into ten random splits of 200 training and 166 testing samples. The classification error of the test data across these splits is utilized to measure the generalization performance.

Table 2 Performance of the methods for the *school* data set

LS-SVM	MTLS-SVM	RMTL [11, 20, 21, 32]	BMTL [6]
9.8 ± 0.6	38.16 ± 0.3	34.32 ± 0.4	34.37 ± 0.4

Table 3 Performance of the methods for the *dermatology* data set

LS-SVM	MTLS-SVM	MTL-FEAT (RBF) [4]	Independent (RBF) [4]
7.9 ± 3.2	8.2 ± 2.7	9.5 ± 3.0	9.8 ± 3.1

We report the misclassification error on test data in Table 3. From Table 3, one can find that the performance of MTLS-SVM is similar to that of LS-SVM. This phenomenon is also observed in [4] for MTL-FEAT (RBF) and independent (RBF). Hence Argyrious et al. [4] conjecture that these tasks are weakly related to each other or unrelated, and their experimental results reinforce their hypothesis. Table 3 also indicates that MTLS-SVM does not "hurt" the performance by simultaneously learning all tasks in such a case.

5 Conclusions

It has been shown through a meticulous empirical study that the generalization performance of LS-SVM is comparable to that of SVM. In order to generalize LS-SVM from single-task to multi-task learning, inspired by the regularized multi-task learning, this study proposes a novel multi-task learning approach, multi-task LS-SVM (MTLS-SVM). Similar to LS-SVM, one only solves a convex linear system in the training phrase, too. What's more, we unify the classification and regression problems in an efficient training algorithm, which effectively employs the Krylow methods.

As for large scale problem, Keerthi and Shevades [30] extends the well-known SMO (Sequential Minimal Optimization) algorithm of SVM to LS-SVM. With the help of Nyström method [47], Brabanter et al. [18] approximates the eigendecomposition of the Gram matrix, thus LS-SVM can be solved in input space rather than in feature space. All these methods can be directly applied to MTLS-SVM. Another way to say this is that most of the approaches for solving LS-SVM can be directly borrowed to solve MTLS-SVM.

Acknowledgements This work was funded partially by Beijing Forestry University Young Scientist Fund: Research on Econometric Methods of Auction with their Applications in the Circulation of Collective Forest Right under grant number BLX2011028, Key Technologies R&D Program of Chinese 12th Five-Year Plan (2011–2015): Key Technologies Research on Large Scale Semantic Computation for Foreign Scientific & Technical Knowledge Organization System, Application Demonstration of Knowledge Service based on STKOS, and Key Technologies Research on Data Mining from the Multiple Electric Vehicle Information Sources under grant number 2011BAH10B04, 2011BAH10B06 and 2013BAG06B01, respectively, National Natural Science Foundation: Multilingual Documents Clustering based on Comparable Corpus under grant number 70903032, Social Science Foundation of Jiangsu Province: Study on Automatic Indexing of Digital Newspapers under grant number 09TQC011, and MOE Project of Humanitites and Social Sciences: Research on Further Processing of e-Newspaper under grant number 09YJC870014. Our gratitude also goes to the anonymous reviewers for their valuable comments.

References

1. Allenby GM, Rossi PE (1998) Marketing models of consumer heterogeneity. J Econ 89(1–2):57
2. An X, Xu S, Zhang L, Su S (2009) Multiple dependent variables LS-SVM regression algorithm and its application in NIR spectral quantitative analysis. Spectrosc Spectr Anal 29(1):127

3. Ando RK, Zhang T (2005) A framework for learning predictive structures from multiple tasks and unlabeled data. J Mach Learn Res 6:1817
4. Argyriou A, Evgeniou T, Pontil M (2008) Convex multi-task feature learning. Mach Learn 73(3):243
5. Arora N, Allenby GM, Ginter JL (1998) A hierarchical Bayes model of primary and secondary demand. Mark Sci 17(1):29
6. Bakker B, Heskes T (2003) Task clustering and gating for Bayesian multitask learning. J Mach Learn Res 4:83
7. Baxter J (1997) A Bayesian/information theoretic model of learning to learn via multiple task sampling. Mach Learn 28(1):7
8. Baxter J (2000) A model of inductive bias learning. J Artif Intell Res 12(1):149
9. Ben-David S, Schuller R (2003) Exploiting task relatedness for multiple task learning. In: Proceedings of the 16th annual conference on computational learning theory, pp 567–580
10. Ben-David S, Gehrke J, Schuller R (2002) A theoretical framework for learning from a pool of disparate data sources. In: Proceedings of the 8th ACM SIGKDD international conference on knowledge discovery and data mining. ACM, New York, NY, pp 443–449
11. Caponnetto A, Micchelli CA, Pontil M, Ying Y (2008) Universal multi-task kernels. J Mach Learn Res 9:1615
12. Caruana R (1997) Multitask learning. Mach Learn 28(1):41
13. Cawley GC (2006) Leave-one-out cross-validation based model selection criteria for weighted LS-SVMs. In: Proceedings of the international joint conference on neural networks. Vancouver, BC, pp 1661–1668
14. Cawley GC, Talbot NLC (2004) Fast exact leave-one-out cross-validation of sparse least-squares support vector machine. Neural Netw 17(10):1467
15. Chapelle O, Shivaswamy P, Vadrevu S, Weinberger K, Zhang Y, Tseng B (2010) Multi-task learning for boosting with application to web search ranking. In: Proceedings of the 16th ACM SIGKDD international conference on knowledge discovery and data mining. ACM, New York, NY, pp 1189–1198
16. Chari R, Lockwood WW, Coe BP, Chu A, Mcacey D, Thomson A, Davies JJ, MacAulay C, Lam WL (2006) SIGMA: a system for integrative genomic microarray analysis of cancer genomes. BMC Bioinform 7:324
17. David B, Sabrina T, Patrick G (2012) A learning to rank framework applied to text-image retrieval. Multimed Tools Appl 60(1):161
18. De Brabanter K, De Brabanter J, Suykens JAK, De Moor B (2010) Optimized fixed-size kernel models for large data sets. Comput Stat Data Anal 54(6):1484
19. Dhillon PS, Sundararajan S, Keerthi SS (2011) Semi-supervised multi-task learning of structured prediction models for web information extraction. In: Proceedings of the 20th ACM international conference on information and knowledge management. ACM, New York, NY, pp 957–966
20. Evgeniou T, Pontil M (2004) Regularized multi-task learning. In: Proceedings of the 10th ACM SIGKDD international conference on knowledge discovery and data mining. Seattle, WA, pp 109–117
21. Evgeniou T, Micchelli CA, Pontil M (2005) Learning multiple tasks with kernel methods. J Mach Learn Res 6:615
22. Evgeniou T, Pontil M, Toubia O (2006) A convex optimization approach to modeling consumer heterogeneity in conjoint estimation. Tech. rep., technolgoy management and decision sciences, INSEAD
23. Girosi F (1998) An equivalence between sparse approximation and support vector machines. Neural Comput 10(6):1455
24. Golub GH, Van Loan CF (1996) Matrix computations, 3rd edn. The Johns Hopkins University Press, Baltimore and London
25. Hamers B, Suykens JA, De Moor B (2001) A comparison of iterative methods for least squares vector machine classifiers. Internal report 01-110, ESAT-SISTA, K.U. Leuven, Leuven, Belgium
26. Heskes T (2000) Empirical Bayes for learning to learn. In: Proceedings of the 17th international conference on machine learning. Morgan Kaufmann Publishers Inc., San Francisco, CA, pp 367–374
27. Hsu JL, Li YF (2012) A cross-modal method of labeling music tags. Multimed Tools Appl 58(3):521

28. Jebara T (2004) Multi-task feature and kernel selection for SVMs. In: Proceedings of the 21st international conference on machine learning. Banff, AB, pp 55–62
29. Keerthi SS, Lin CJ (2003) Asymptotic behaviors of support vector machines with Gaussian kernel. Neural Comput 15(7):1667
30. Keerthi SS, Shevade SK (2003) SMO algorithm for least squares SVM formulations. Neural Comput 15(2):487
31. Lin HT, Lin CJ (2003) A study on sigmoid kernels for SVM and the training of non-PSD kernels by SMO-type methods. Tech. rep., department of computer science, National Taiwan University
32. Micchelli CA, Pontil M (2005) Kernels for multi-task learning. In: Saul LK, Weiss Y, Bottou L (eds) Advances in neural information processing systems 18, vol 17. MIT Press, Cambridge, MA, pp 921–928
33. Minka S, Rätsch G, Müller KR (2001) A mathematical programming approach to the kernel fisher algorithm. In: Advances in Neural Information Processing Systems, vol 13. MIT Press, Cambridge, MA
34. Press WH, Teukolsky SA, Vetterling WT, Flannery BP (1992) Numerical recipes in C: the art of scientific computing, 2nd edn. Cambridge University Press, New York,
35. Saad Y (2003) Iterative methods for sparse linear systems, 2nd edn. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA
36. Saunders C, Gammerman A, Vovk V (1998) Ridge regression learning algorithm in dual variables. In: Shavlik JW (ed) Proceedings of the 15th international conference on machine learning. Morgan Kaufmann Publishers Inc., San Francisco, CA, pp 515–521
37. Smola AJ, Schölkopf B, Müller KR (1998) The connection between regularization operators and support vector kernels. Neural Netw 11(4):637
38. Suykens JAK, Vandewalle J (1999) Least squares support vector machine classifiers. Neural Process Lett 9(3):293
39. Suykens JA, Lukas L, Van Dooren P, De Moor B, Vandewalle J (1999) Least squares support vector machine classifiers: a large scale algorithm. In: Proceedings of the European conference on circuit theory and design. Stresa, Italy, pp 839–842
40. Suyken JAK, Van Gestel T, De Brabanter J, De Moor B, Vandewalle J (eds) (2002) Least Squares Support Vector Machines. World Scientific Pub. Co
41. Thrun S, Pratt LY (eds) (1997) Learning to learn. Kluwer Academic Press
42. Torralba A, Murphy KP, Freeman WT (2004) Sharing features: efficient boosting procedures for multiclass object detection. In: Proceedings of the 17th IEEE conference on computer vision and pattern recognition. IEEE Computer Society, pp 762–769
43. Van Gestel T, Suykens JAK, Lanckriet G, De Moor B, Vandewalle J (2002) Bayesian framework for least-squares support vector machine classifiers, Gaussian processes, and kernel fisher discriminant analysis. Neural Comput 14(5):1115
44. Van Gestel T, Suykens JAK, Baesens B, Viaene S, Vanthienen J, Dedene G, Moor BD, Vandewalle J (2004) Benchmarking least squares support vector machine classifiers. Mach Learn 54(1):5
45. Vapnik VN (ed) (1998) Statistical learning theory. Wiley & Sons, Inc., New York
46. Vapnik VN (ed) (1999) The nature of statistical learning theory, 2nd edn. Springer, New York
47. Williams CKI, Seeger M (2001) Using the Nyström method to speed up kernel machines. In: Leen TK, Dietterich TG, Tresp V (eds) Advances in neural information processing systems, vol 13. MIT Press, Cambridge, MA, pp 682–688
48. Xu S, Ma F, Tao L (2007) Learn from the information contained in the false splice sites as well as in the true splice sites using SVM. In: Proceedings of the international conference on intelligent systems and knowledge engineering. Atlantis Press, pp 1360–1366
49. Xu S, An X, Qiao X, Zhu L, Li L (2011) Semi-supervised least-squares support vector regression machines. J Inf Comput Sci 8(6):885
50. Xu S, Qiao X, Zhu L, An X, Zhang L (2011) Multi-task least-squares support vector regression machines and their applications in NIR spectral analysis. Spectrosc Spectr Anal 31(5):1208
51. Xu S, An X, Qiao X, Zhu L, Li L (2013) Multi-output least-squares support vector regression machines. Pattern Recogn Lett 34(9):1078
52. Ye J, Xiong T (2007) SVM versus least squares SVM. In: Meila M, Shen X (eds) Proceedings of the 11th international conference on artificial intelligence and statistics, pp 644–651

Shuo Xu works as associate professor and the manager of Text Mining Lab., Information Technology Support Center, at Institute of Scientific and Technical Information of China (ISTIC), China. He obtained his M.S. and Ph.D. from China Agriculture University (CAU). His current research interest includes text mining (TM), machine learning (ML), natural language processing (NLP), science and technology monitoring, and knowledge organization system (KOS), etc.

Xin An works as lecturer, School of Economics and Management, at Beijing Forestry University, China. She obtained her M.S. from China Agriculture University (CAU), and Ph.D. from University of International Business and Economics (UIBE). Her current research interest includes data mining (DM), keywords auction, quantitative economics, game theory with applications to economics, etc.

Xiaodong Qiao works as professor and chief engineer of Institute of Scientific Technical Information of China (ISTIC), China. He obtained his M.S. from university of Sheffield, United Kingdom. His current research interest includes knowledge service technology, digital library, and information resource management, etc.

Lijun Zhu works as professor and deputy director of Information Technology Support Center, at Institute of Scientific and Technical Information of China (ISTIC), China. He obtained his M.S. and Ph.D. from China University of Petroleum and China Agriculture University (CAU), respectively. His current research interest includes semantic web, web service and knowledge organization system (KOS), science and technology information service based knowledge technology, etc.