Neural Processing Letters
https://doi.org/10.1007/s11063-022-10870-1

TSVMPath: Fast Regularization Parameter Tuning Algorithm for Twin Support Vector Machine

Kanglei Zhou[1] · **Qiyang Zhang**[2] · **Juntao Li**[3]

Accepted: 29 April 2022

Abstract
Twin support vector machine (TSVM) has attracted much attention in the field of machine learning with good generalization ability and computational performance. However, the conventional grid search method is very time-consuming to obtain the optimal regularization parameter. To address this problem, we develop a novel fast regularization parameter tuning algorithm for TSVM, named TSVMPath. After transforming the models of two sub-optimization problems, we divide the two classes of samples into different sets. Lagrangian multipliers are then proved to be piecewise linear concerning the corresponding regularization parameters, greatly extending the search space of the solution. By proving that the Lagrangian multipliers of two sub-optimization models are 1 when the regularization parameters approach infinity, we design a simple yet effective initialization. As a result, the entirely regularized solution path can be obtained without solving quadratic programming problems. Four types of events are finally defined to update the solution path. Experiments on 8 UCI datasets show that the prediction accuracy of TSVMPath is superior to the best competing methods, with up to four orders of magnitude speed-up for the computational overhead compared with the grid search method.

Keywords Statistical machine learning · Twin support vector machine · Parameter tuning algorithm · Regularized solution path

✉ Juntao Li
lijuntao@htu.edu.cn

Kanglei Zhou
zhoukanglei@buaa.edu.cn

Qiyang Zhang
qyzhang@bupt.edu.cn

[1] School of Computer Science and Engineering, Beihang University, 37 Xueyuan Road, Beijing 100191, China

[2] State Key Laboratory of Networking and Switching, Beijing University of Posts and Telecommunications, 10 Xitucheng Road, Beijing 100876, China

[3] College of Mathematics and Information Science, Henan Normal University, 46 Jianshe East Road, Xinxiang 453007, Henan, China

Published online: 07 June 2022

1 Introduction

Although deep learning is hot in recent years, machine learning algorithms [1–3] such as support vector machine (SVM) [4] are still not negligible due to their solid theoretical support and strong interpretability. SVM was born in 1964 and developed rapidly in the 1990s. Since then, a series of improved extensions have emerged, among which twin SVM (TSVM) [5] is one of the most powerful variants. TSVM has achieved brilliant achievements in many applications [6–8]. However, obtaining the optimal regularization parameter for TSVM is challenging. Therefore, it is vital to develop an efficient solution path algorithm of regularization parameters for improving the performance of TSVM.

In the field of machine learning, most algorithms essentially pre-define one or more parameters to solve quadratic programming problems (QPPs), dubbed parameter quadratic programming (PQP) [9]. Parameters in a PQP problem are typically tuned by cross-validation. TSVM requires multiple training under different parameter settings, so it is hard to explore the optimal parameter extensively. In practice, TSVM usually depends on training many times by the traditional grid search method to determine the optimal hyperparameter. However, especially for multi-parameter adjustment problems, it is computationally expensive and unworkable. To address this problem, researchers have proposed some fast parameter tuning methods [10–12].

Compared with SVM, TSVM solves two small-scale QPPs instead of a large one. The method alleviates the stability problem of SVM in solving large-scale high-dimensional data but increases the difficulty of designing the entirely regularized solution path algorithm of TSVM. Therefore, the entirely regularized solution path algorithm of SVM [10] has been proposed as early as 2004, while TSVM has not been fully solved yet. Several attempts [11, 12] have been made since TSVM was born. However, they will inhibit the performance of the algorithm itself to some extent and cannot fully explore the entirely regularized solution path. Unlike the previous works [11, 12], we develop an entirely regularized solution path algorithm by strengthening the role of regularization.

Aiming at solving the PQP problem for TSVM, this paper develops a novel entirely regularized solution path for TSVM, i.e., TSVMPath,[1] including four steps: (1) We first adopt a simple yet effective sample partition strategy after model transformation. (2) Lagrangian multipliers in two QPPs of TSVM are piecewise linear w.r.t. regularization parameters accordingly. (3) An efficient initialization is designed without solving QPPs. (4) Four types of events are defined to seek breaks points of the regularized path. TSVMPath has reduced the computational overhead of parameter adjustment compared with the traditional grid search method. Experiments on 8 UCI datasets verify that both the prediction accuracy and the training efficiency are superior to the baselines.

The main contributions of this work are summarized as:

- Lagrangian multipliers of two sub-optimization problems are proved to be piecewise linear w.r.t. regularization parameters, ensuring only solving the breakpoints of regularization parameters to obtain the entirely regularized solution path.
- Lagrangian multipliers are proved to be 1 when the regularization parameter approaches infinity. And we design a simple yet effective initialization process, so that the entirely regularized solution path can be obtained without solving QPPs.
- The fast regularization parameter tuning algorithm for TSVM is proposed, which largely reduces the computational overhead of parameters tuning and greatly extends the solution space of regularization parameters to $(0, +\infty)$.

[1] Code will be available at https://github.com/ZhouKanglei/TSVMPath.

The organization of the rest paper is as follows: Sect. 2 reviews the related work, Sect. 3 dwells the basic concepts of TSVM and proposes the sample partition strategy, Sect. 4 proves that Lagrangian multipliers are piecewise linear w.r.t. the regularization parameters accordingly, Sect. 5 initializes the two sub-optimization problems by proving Lagrangian multipliers to be 1 as the regularization parameters approach infinity, Sect. 6 designs the entirely regularized solution path algorithm of TSVM in detail, Sect. 7 gives the experimental results and verifies the effectiveness of the algorithm, and Sect. 8 concludes the whole paper.

2 Related Work

In this section, we first review different SVM extensions and then introduce parameter tuning methods.

SVM As a well-known statistical machine learning method, SVM [4] is first proposed by Vapnik et al., based on the principle of structured risk minimization and Vapnik-Chervonenkis dimension theory. SVM trains samples by solving a convex QPP and constructing a classification hyperplane to maximize the classification margin. Due to its good predictive performance and powerful generalization ability, SVM has been developed into a wide range of applications in solving many practical problems such as text classification [13–15], time series analysis [16–18] and face recognition [19–21]. However, with the unstoppable development of the Internet and information technology, a large amount of high-dimensional, distributed and dynamic complex data are generated increasingly, leading to unprecedented difficulties for SVM in processing these complicated data.

TSVM To improve the prediction accuracy and computational efficiency, Jayadeva et al. proposed TSVM [5] based on standard SVM. Unlike SVM, TSVM constructs two non-parallel hyperplanes by solving two small-scale QPPs and makes one class of samples approach one hyperplane and stay away from the other hyperplane. Since TSVM converts a large QPP into two small-scale QPPs and the number of constraints for each small-scale QPP is half that of the original problem, the training performance can be efficiently improved [22]. On account of the obvious advantages, TSVM has become a hot topic in the field of machine learning and has been successfully used in intrusion detection [6, 23, 24], speaker recognition [8, 25, 26], cancer diagnosis and prognosis [7, 27, 28] and many other fields.

TSVM Extensions To further improve the comprehensive performance of TSVM, many eminent improvements are made, e.g., the least-square TSVM (LSTSVM) [29–32], weighted TSVM (WTSVM) [33–36], projection TSVM (PTSVM) [37–39], etc. We refer the interested readers to recent surveys [40, 41] for a more in-depth treatment of the area.

To improve the solving speed, Kumar et al. introduced the concept of approximate SVM to the original problem of TSVM and then proposed LSTSVM [42]. LSTSVM also needs to generate two non-parallel hyperplanes, but it only considers linear equality constraints instead of inequality ones in the original problem of TSVM. It extremely improves the solving efficiency and prediction accuracy. To extend LSTSVM into multi-classification problems, Chen et al. proposed a multi-classification LSTSVM classifier [43] based on the idea of optimal directed acyclic graph.

Because TSVM cannot fully exploit the potential correlation or similar information between any pair of data points with the same label, Ye et al. proposed WTSVM with local information [44] to overcome this shortcoming. WTSVM retains the benefits of TSVM while being able to mine as much potentially similar information as possible from the sample.

The general idea of PTSVM [45] is to find two projection directions, one for each class. The projection sample of such a class is well separated from the projection sample of the other class in its subspace. More than one projection axis is generated for each class, which further improves the performance of the algorithm. To overcome the singularity, principal component analysis (PCA) is used to transform the data in the original space into a lower-dimensional subspace.

Parameter Tuning Method Aiming at several important PQP problems in machine learning, researchers have put forward corresponding solutions [10–12, 46–49].

Hastie et al. proposed an entire regularization solution path algorithm based on SVM, termed as SVMPath [10]. SVMPath does not require multiple retraining of the model, which greatly improves the computational performance of SVM parameter tuning. Pan et al. designed a safety screening rule [11] to solve the original problem of QPP, which is helpful to speed up the TSVM training process. However, the entirely regularized solution path is not fully explored. Yang et al. proposed the piecewise linear solution algorithm of TSVM based on Pinball loss [12], which can provide optimal precision for all possible parameter values. Without solving QPP, the starting point of the solution path can be solved analytically, and it achieves good flexibility and predictive performance. However, the role of regularization is loose.

Although researchers have proposed many techniques to solve the PQP problem [49], they all affect the performance of TSVM to some extent. This work strengthens the role of regularization and designs a better solution path algorithm for the PQP problem of TSVM.

3 Preliminaries

In this section, we first define necessary notations, and then briefly give two sub-optimization problems of TSVM. Finally, we propose a simple yet effective sample partition strategy after model transformation.

3.1 Notations

This paper denotes $\mathcal{T} = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_n, y_n)\}$ as the training set of samples, where n is the number of samples, $\boldsymbol{x}_i \in \mathbb{R}^{m\times 1}$ is the feature vector of the ith sample and $y_i \in \{-1, 1\}$ is its corresponding class label. $\mathcal{A}$ and $\mathcal{B}$ are used to represent the index sets of positive $(+1)$ and negative (-1) samples, respectively. Let $\mathbf{A} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_{n_A}]^T$ and $\mathbf{B} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_{n_B}]^T$, where n_A and n_B are the number of two classes respectively, s.t., $n = n_A + n_B$.

3.2 Twin Support Vector Machine

To distinguish the different sample categories, the basic idea is to find a partition hyperplane in the sample space based on the training set. TSVM solves the classification problem by constructing two non-parallel hyperplanes $f_1 : \boldsymbol{x}^{\mathrm{T}}\boldsymbol{w}_1 + b_1 = 0$ and $f_2 : \boldsymbol{x}^{\mathrm{T}}\boldsymbol{w}_2 + b_2 = 0$ instead of one hyperplane, where $\boldsymbol{w}_1 \in \mathbb{R}^{m\times 1}$ and $\boldsymbol{w}_2 \in \mathbb{R}^{m\times 1}$ are the normal vectors of the two hyperplanes, respectively.

As shown in Fig. 1, each hyperplane corresponds to a class of samples, and each class of samples is as close as possible to its corresponding hyperplane and away from the other

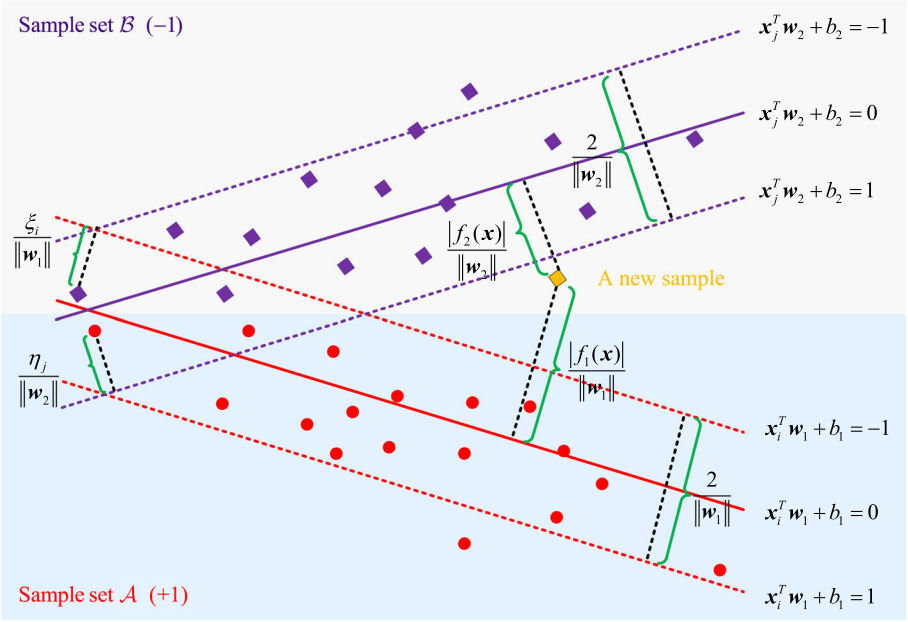


Fig. 1 Illustration of TSVM: the red circle samples is in the set $\mathcal{A}$, marked as $+1$, the violet square samples are in the set $\mathcal{B}$, marked as -1, and solid lines in red and violet represent two nonparallel hyperplanes, respectively. In addition, the distance from the sample to the hyperplane is also indicated in the figure

hyperplane. Accordingly, the label of a new sample is determined by the distance of the sample from two hyperplanes.

Compared with SVM [4], TSVM [5] solves two small-sized QPPs instead of a large one, as follows:

$$\begin{aligned} \min_{\boldsymbol{w}_1, b_1, \boldsymbol{\xi}} \ & \frac{1}{2}\|\mathbf{A}\boldsymbol{w}_1 + b_1\mathbf{e}_{n_A}\|^2 + c_1\mathbf{e}_{n_B}^{\mathrm{T}}\boldsymbol{\xi} \\ \text{s.t.} \ & -(\mathbf{B}\boldsymbol{w}_1 + b_1\mathbf{e}_{n_B}) + \boldsymbol{\xi} \geq \mathbf{e}_{n_B}, \\ & \boldsymbol{\xi} \geq 0\mathbf{e}_{n_B}, \end{aligned} \tag{1}$$

and

$$\begin{aligned} \min_{\boldsymbol{w}_2, b_2, \boldsymbol{\eta}} \ & \frac{1}{2}\|\mathbf{B}\boldsymbol{w}_2 + b_2\mathbf{e}_{n_B}\|^2 + c_2\mathbf{e}_{n_A}^{\mathrm{T}}\boldsymbol{\eta} \\ \text{s.t.} \ & (\mathbf{A}\boldsymbol{w}_2 + b_2\mathbf{e}_{n_A}) + \boldsymbol{\eta} \geq \mathbf{e}_{n_A}, \\ & \boldsymbol{\eta} \geq 0\mathbf{e}_{n_A}, \end{aligned} \tag{2}$$

where the penalty parameters satisfy $c_1 > 0$ and $c_2 > 0$, $\boldsymbol{\xi} \in \mathbb{R}^{n_B \times 1}$ and $\boldsymbol{\eta} \in \mathbb{R}^{n_A \times 1}$ are slack variables, and $\mathbf{e}_{n_A} \in \mathbb{R}^{n_A \times 1}$ and $\mathbf{e}_{n_B} \in \mathbb{R}^{n_B \times 1}$ are the unit vectors with different dimensions.

3.3 Partition Strategies

We first transform the two QPPs into their dual formats respectively so that the solution can be obtained, and then propose corresponding sample partition strategies.

3.3.1 The First QPP

Model Transformation For the first sub-optimization problem, let $\lambda_1 = 1/c_1$, the QPP (1) can be converted to (3).

$$\begin{aligned} \min_{\boldsymbol{w}_1, b_1, \boldsymbol{\xi}} \ & \frac{\lambda_1}{2}\|\mathbf{A}\boldsymbol{w}_1 + b_1\mathbf{e}_{n_A}\|^2 + \mathbf{e}_{n_B}^{\mathrm{T}}\boldsymbol{\xi} \\ \text{s.t.} \ & -(\mathbf{B}\boldsymbol{w}_1 + b_1\mathbf{e}_{n_B}) + \boldsymbol{\xi} \geq \mathbf{e}_{n_B}, \\ & \boldsymbol{\xi} \geq 0\mathbf{e}_{n_B}. \end{aligned} \tag{3}$$

Compared with Eq. (1), this transformation emphasizes the role of regularization [10].

The Lagrangian function of the QPP (3) can be constructed as follows:

$$\begin{aligned} \mathcal{L}_1(\boldsymbol{w}_1, b_1, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = & \frac{\lambda_1}{2}\|\mathbf{A}\boldsymbol{w}_1 + b_1\mathbf{e}_{n_A}\|^2 + \mathbf{e}_{n_B}^{\mathrm{T}}\boldsymbol{\xi} \\ & + \boldsymbol{\alpha}^{\mathrm{T}}[\mathbf{e}_{n_B} + (\mathbf{B}\boldsymbol{w}_1 + b_1\mathbf{e}_{n_B}) - \boldsymbol{\xi}] - \boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{\xi}. \end{aligned} \tag{4}$$

where $\boldsymbol{\alpha} \in \mathbb{R}^{n_B \times 1}$ and $\boldsymbol{\beta} \in \mathbb{R}^{n_B \times 1}$ are vectors of Lagrangian multipliers, and each of their components satisfies $\alpha_i \geq 0$ and $\beta_i \geq 0$ ($i \in \mathcal{B}$).

Let the partial derivative of $\mathcal{L}_1(\boldsymbol{w}_1, b_1, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta})$ w.r.t. $\boldsymbol{w}_1$, b_1 and $\boldsymbol{\xi}$ be equal to zero respectively, and we can obtain the following equations:

$$\frac{\partial \mathcal{L}_1}{\partial \boldsymbol{w}_1} = \lambda_1\mathbf{A}^{\mathrm{T}}(\mathbf{A}\boldsymbol{w}_1 + b_1\mathbf{e}_{n_A}) + \mathbf{B}^{\mathrm{T}}\boldsymbol{\alpha} = 0\mathbf{e}_m, \tag{5}$$

$$\frac{\partial \mathcal{L}_1}{\partial b_1} = \lambda_1\mathbf{e}_{n_A}^{\mathrm{T}}(\mathbf{A}\boldsymbol{w}_1 + b_1\mathbf{e}_{n_A}) + \mathbf{e}_{n_B}^{\mathrm{T}}\boldsymbol{\alpha} = 0, \tag{6}$$

$$\frac{\partial \mathcal{L}_1}{\partial \boldsymbol{\xi}} = \mathbf{e}_{n_B} - \boldsymbol{\alpha} - \boldsymbol{\beta} = 0\mathbf{e}_{n_B}. \tag{7}$$

From Eqs. (5) and (6), we have

$$\lambda_1\mathbf{H}^{\mathrm{T}}\mathbf{H}\boldsymbol{u} + \mathbf{G}^{\mathrm{T}}\boldsymbol{\alpha} = 0\mathbf{e}_{m+1}, \tag{8}$$

where $\mathbf{H} = [\mathbf{A}\ \mathbf{e}_{n_A}]$, $\mathbf{G} = [\mathbf{B}\ \mathbf{e}_{n_B}]$ and $\boldsymbol{u} = \begin{bmatrix} \boldsymbol{w}_1 \\ b_1 \end{bmatrix}$.

When the matrix $\mathbf{H}^{\mathrm{T}}\mathbf{H}$ is invertible, we can obtain

$$\boldsymbol{u} = -\frac{1}{\lambda_1}\left(\mathbf{H}^{\mathrm{T}}\mathbf{H} + \delta\mathbf{I}\right)^{-1}\mathbf{G}^{\mathrm{T}}\boldsymbol{\alpha}, \tag{9}$$

where the regularization term $\delta\mathbf{I}$ is to avoid the possible irreversible problem of $\mathbf{H}^{\mathrm{T}}\mathbf{H}$, δ is a minimal positive number, and $\mathbf{I} \in \mathbb{R}^{(m\times 1)\times(m\times 1)}$ is an unit matrix. By substituting Eq. (9) into the hyperplane $f_1(\boldsymbol{x})$, we can obtain

$$f_1(\boldsymbol{x}) = -\frac{1}{\lambda_1}\left[\boldsymbol{x}^{\mathrm{T}}\ 1\right]\left(\mathbf{H}^{\mathrm{T}}\mathbf{H} + \delta\mathbf{I}\right)^{-1}\mathbf{G}^{\mathrm{T}}\boldsymbol{\alpha}. \tag{10}$$

Partition Strategy for Samples in $\mathcal{B}$ In combination with Karush–Kuhn–Tucker (KKT) conditions [5], we can obtain

$$\boldsymbol{\alpha}^{\mathrm{T}}[\mathbf{e}_{n_B} + (\mathbf{B}\boldsymbol{w}_1 + b_1\mathbf{e}_{n_B}) - \boldsymbol{\xi}] = 0, \tag{11}$$

$$-(\mathbf{B}\boldsymbol{w}_1 + b_1\mathbf{e}_{n_B}) + \boldsymbol{\xi} - \mathbf{e}_{n_B} \geq 0\mathbf{e}_{n_B}, \tag{12}$$

$$\boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{\xi} = 0, \tag{13}$$

$$\boldsymbol{\xi} \geq 0\mathbf{e}_{n_B}. \tag{14}$$

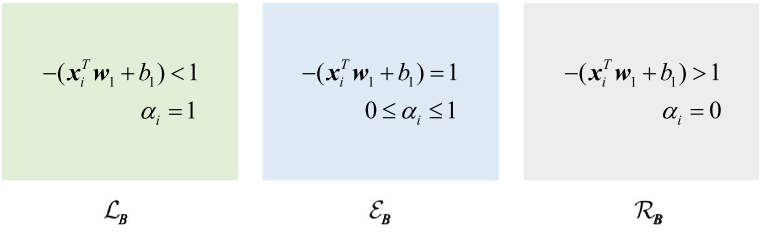


Fig. 2 Diagram of the partition of set $\mathcal{B}$: three different color-coded boxes represent three sample index sets $\mathcal{L}_B$, $\mathcal{E}_B$ and $\mathcal{R}_B$ respectively, in which each box indicates the conditions that the sample points in the index set meet

For $\forall i \in \mathcal{B}$, the following facts can be obtained from Eqs. (5)–(14).

- $-(\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{w}_1 + b_1) < 1 \overset{(12)}{\Longrightarrow} \xi_i > 0 \overset{(13)}{\Longrightarrow} \beta_i = 0 \overset{(7)}{\Longrightarrow} \alpha_i = 1.$
- $-(\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{w}_1 + b_1) = 1 \overset{(12)}{\Longrightarrow} \xi_i \ge 0 \overset{(13),(7)}{\Longrightarrow} 0 \le \beta_i \le 1 \overset{(7)}{\Longrightarrow} 0 \le \alpha_i \le 1.$
- $-(\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{w}_1 + b_1) > 1 \overset{(12)}{\Longrightarrow} \xi_i \ge 0 \overset{(12)}{\Longrightarrow} -(\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{w}_1 + b_1) + \xi_i - 1 > 0 \overset{(11)}{\Longrightarrow} \alpha_i = 0.$

Therefore, the set $\mathcal{B}$ can be ulteriorly divided into three index sets $\mathcal{L}_B$, $\mathcal{E}_B$ and $\mathcal{R}_B$ as shown in Fig. 2, where $\mathcal{L}_B = \{i \mid -(\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{w}_1 + b_1) < 1\}$, $\mathcal{E}_B = \{i \mid -(\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{w}_1 + b_1) = 1\}$ and $\mathcal{R}_B = \{i \mid -(\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{w}_1 + b_1) > 1\}$.

The Dual Problem Additionally, using Eq. (4) and the above KKT conditions in Eqs. (11), (12), (13), (14), we can obtain the Wolfe dual [5] of Eq. (3) as follows:

$$\begin{aligned} \max_{\boldsymbol{\alpha}} \ & \mathbf{e}_{n_B}^{\mathrm{T}}\boldsymbol{\alpha} - \frac{1}{2\lambda_1}\boldsymbol{\alpha}^{\mathrm{T}}\mathbf{G}\left(\mathbf{H}^{\mathrm{T}}\mathbf{H} + \delta\mathbf{I}\right)^{-1}\mathbf{G}^{\mathrm{T}}\boldsymbol{\alpha} \\ \text{s.t. } & 0\mathbf{e}_{n_B} \le \boldsymbol{\alpha} \le \mathbf{e}_{n_B}. \end{aligned} \tag{15}$$

3.3.2 The Second QPP

Model Transformation In exactly the similar way, let $\lambda_2 = 1/c_2$, the second QPP (2) can be converted to

$$\begin{aligned} \min_{\boldsymbol{w}_2, b_2, \boldsymbol{\eta}} \ & \frac{\lambda_2}{2}\|\mathbf{B}\boldsymbol{w}_2 + b_2\mathbf{e}_{n_B}\|^2 + \mathbf{e}_{n_A}^{\mathrm{T}}\boldsymbol{\eta} \\ \text{s.t. } & (\mathbf{A}\boldsymbol{w}_2 + b_2\mathbf{e}_{n_A}) + \boldsymbol{\eta} \ge \mathbf{e}_{n_A}, \\ & \boldsymbol{\eta} \ge 0\mathbf{e}_{n_A}. \end{aligned} \tag{16}$$

The Lagrangian function of the QPP (16) can be constructed as follows:

$$\begin{aligned} \mathcal{L}_2(\boldsymbol{w}_2, b_2, \boldsymbol{\eta}, \boldsymbol{\gamma}, \boldsymbol{\omega}) = & \frac{\lambda_2}{2}\|\mathbf{B}\boldsymbol{w}_2 + b_2\mathbf{e}_{n_B}\|^2 + \mathbf{e}_{n_A}^{\mathrm{T}}\boldsymbol{\eta} \\ & + \boldsymbol{\gamma}^{\mathrm{T}}[\mathbf{e}_{n_A} - (\mathbf{A}\boldsymbol{w}_2 + b_2\mathbf{e}_{n_A}) - \boldsymbol{\eta}] - \boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{\eta}, \end{aligned} \tag{17}$$

where $\boldsymbol{\gamma} \in \mathbb{R}^{n_A \times 1}$ and $\boldsymbol{\omega} \in \mathbb{R}^{n_A \times 1}$ are vectors of Lagrangian multipliers , and each of their components satisfies $\gamma_i \ge 0$ and $\omega_i \ge 0$ ($i \in \mathcal{A}$).

Let the partial derivative of $\mathcal{L}_2(\boldsymbol{w}_2, b_2, \boldsymbol{\eta}, \boldsymbol{\gamma}, \boldsymbol{\omega})$ w.r.t. $\boldsymbol{w}_2$, b_2 and $\boldsymbol{\eta}$ be equal to zero respectively, the following equations can be obtained.

$$\frac{\partial \mathcal{L}_2}{\partial \boldsymbol{w}_2} = \lambda_2 \mathbf{B}^{\mathrm{T}}(\mathbf{B}\boldsymbol{w}_2 + b_2\mathbf{e}_{n_B}) - \mathbf{A}^{\mathrm{T}}\boldsymbol{\gamma} = 0\mathbf{e}_m, \tag{18}$$

$$\frac{\partial \mathcal{L}_2}{\partial b_2} = \lambda_2 \mathbf{e}_{n_B}^{\mathrm{T}}(\mathbf{B}\boldsymbol{w}_2 + b_2\mathbf{e}_{n_B}) - \mathbf{e}_{n_A}^{\mathrm{T}}\boldsymbol{\gamma} = 0, \tag{19}$$

$$\frac{\partial \mathcal{L}_2}{\partial \boldsymbol{\eta}} = \mathbf{e}_{n_A} - \boldsymbol{\gamma} - \boldsymbol{\omega} = 0\mathbf{e}_{n_A}. \tag{20}$$

From Eqs. (18) and (19), we have

$$\lambda_2 \mathbf{Q}^{\mathrm{T}}\mathbf{Q}\boldsymbol{v} - \mathbf{P}^{\mathrm{T}}\boldsymbol{\gamma} = 0\mathbf{e}_{m+1}, \tag{21}$$

where $\mathbf{P} = [\mathbf{A}\ \mathbf{e}_{n_A}]$, $\mathbf{Q} = [\mathbf{B}\ \mathbf{e}_{n_B}]$ and $\boldsymbol{v} = \begin{bmatrix} \boldsymbol{w}_2 \\ b_2 \end{bmatrix}$.

When the matrix $\mathbf{Q}^{\mathrm{T}}\mathbf{Q}$ is invertible, we can obtain

$$\boldsymbol{v} = \frac{1}{\lambda_2}\left(\mathbf{Q}^{\mathrm{T}}\mathbf{Q} + \delta\mathbf{I}\right)^{-1}\mathbf{P}^{\mathrm{T}}\boldsymbol{\gamma}, \tag{22}$$

where the regularization term $\delta\mathbf{I}$ is to avoid the possible irreversible problem of $\mathbf{Q}^{\mathrm{T}}\mathbf{Q}$. By substituting Eq. (22) into the hyperplane $f_1(\boldsymbol{x})$, we can obtain

$$f_2(\boldsymbol{x}) = \frac{1}{\lambda_2}\left[\boldsymbol{x}^{\mathrm{T}}\ 1\right]\left(\mathbf{Q}^{\mathrm{T}}\mathbf{Q} + \delta\mathbf{I}\right)^{-1}\mathbf{P}^{\mathrm{T}}\boldsymbol{\gamma}. \tag{23}$$

Partition Strategy for Samples in $\mathcal{A}$ In combination with KKT conditions, we can obtain

$$\boldsymbol{\gamma}^{\mathrm{T}}[\mathbf{e}_{n_A} - (\mathbf{A}\boldsymbol{w}_2 + b_2\mathbf{e}_{n_A}) - \boldsymbol{\eta}] = 0, \tag{24}$$

$$(\mathbf{A}\boldsymbol{w}_2 + b_2\mathbf{e}_{n_A}) + \boldsymbol{\eta} - \mathbf{e}_{n_A} \geq 0\mathbf{e}_{n_A}, \tag{25}$$

$$\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{\eta} = 0, \tag{26}$$

$$\boldsymbol{\eta} \geq 0\mathbf{e}_{n_A}. \tag{27}$$

For $\forall i \in \mathcal{A}$, the following facts can be obtained.

- $\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{w}_2 + b_2 < 1 \overset{(25)}{\Longrightarrow} \eta_i > 0 \overset{(26)}{\Longrightarrow} \omega_i = 0 \overset{(20)}{\Longrightarrow} \gamma_i = 1.$
- $\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{w}_2 + b_2 = 1 \overset{(25)}{\Longrightarrow} \eta_i \geq 0 \overset{(26),(20)}{\Longrightarrow} 0 \leq \omega_i \leq 1 \overset{(20)}{\Longrightarrow} 0 \leq \gamma_i \leq 1.$
- $\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{w}_2 + b_2 > 1 \overset{(25)}{\Longrightarrow} \eta_i \geq 0 \overset{(25)}{\Longrightarrow} \boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{w}_2 + b_2 + \eta_i - 1 > 0 \overset{(24)}{\Longrightarrow} \gamma_i = 0.$

Then, we can get the similar partition results as shown in Fig. 3, i.e., the set $\mathcal{A}$ can be divided into three index sets $\mathcal{L}_A$, $\mathcal{E}_A$ and $\mathcal{R}_A$, where $\mathcal{L}_A = \{i \mid \boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{w}_2 + b_2 < 1\}$, $\mathcal{E}_A = \{i \mid \boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{w}_2 + b_2 = 1\}$ and $\mathcal{R}_A = \{i \mid \boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{w}_2 + b_2 > 1\}$.

The Dual Problem Similarly, using Eq. (17) and the above KKT conditions in Eqs. (24), (25), (26), (27), we can obtain the Wolfe dual [5] of Eq. (16) as follows:

$$\begin{aligned} \max_{\boldsymbol{\alpha}}\ & \mathbf{e}_{n_A}^{\mathrm{T}}\boldsymbol{\gamma} - \frac{1}{2\lambda_2}\boldsymbol{\gamma}^{\mathrm{T}}\mathbf{Q}\left(\mathbf{P}^{\mathrm{T}}\mathbf{P} + \delta\mathbf{I}\right)^{-1}\mathbf{Q}^{\mathrm{T}}\boldsymbol{\gamma} \\ \text{s.t. } & 0\mathbf{e}_{n_A} \leq \boldsymbol{\gamma} \leq \mathbf{e}_{n_A}. \end{aligned} \tag{28}$$

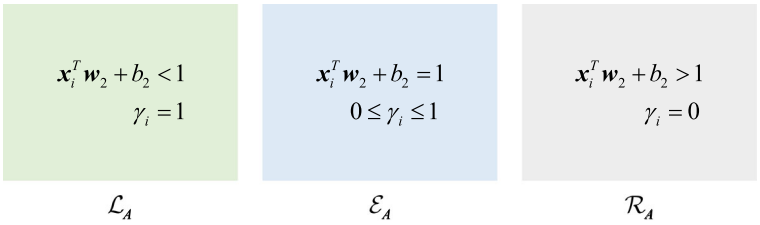


Fig. 3 Diagram of the partition of set $\mathcal{A}$: three different color-coded boxes represent three sample index sets $\mathcal{L}_A$, $\mathcal{E}_A$ and $\mathcal{R}_A$ respectively, in which each box indicates the conditions that the sample points in the index set meet

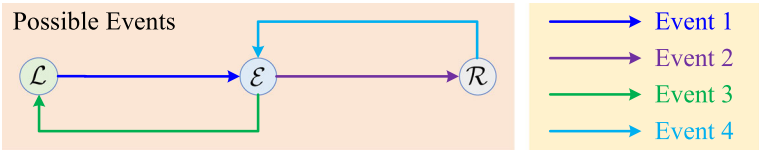


Fig. 4 Illustration of possible events: there are four different kinds of possible events for three sets $\mathcal{L}$, $\mathcal{E}$ and $\mathcal{R}$

4 Piecewise Linear Theory

In this section, the *event* is first defined and discussed. Then, the piecewise linear theory is established for the above two QPPs, i.e., the Lagrangian multipliers are proved to be piecewise linear w.r.t. the regularization parameters respectively.

Definition 1 When the regularization parameter changes, the index sets change accordingly. This paper defines the change of the sample point from one set $\mathcal{C}_1$ to another one C_2 w.r.t. the regularization parameter as an **event**, denoted as $\mathcal{C}_1 \to \mathcal{C}_2$.

4.1 Possible Events

For every sub-optimization problem, there are always four different kinds of possible events for three index sets, as shown in Fig. 4. We discuss two QPPs separately in the following.
The First QPP The QPP (3) mainly depends on the samples at the elbow $\mathcal{E}_B$. When the regularization parameter λ_1 changes, the sample index sets $\mathcal{L}_B$, $\mathcal{E}_B$ and $\mathcal{R}_B$ will change accordingly. We consider all the possible events from the following three scenarios.

i. If $\mathcal{E}_B \neq \varnothing$, then the sample $\boldsymbol{x}_i$ $(i \in \mathcal{E}_B)$ from the set $\mathcal{E}_B$ might go into sets $\mathcal{L}_B$ or $\mathcal{R}_B$, i.e.,

Event 1 $\mathcal{E}_B \to \mathcal{L}_B \Leftrightarrow 0 \le \alpha_i \le 1 \to \alpha_i = 1 \Leftrightarrow f_1(\boldsymbol{x}_i) = -1 \to f_1(\boldsymbol{x}_i) > -1$.
Event 2 $\mathcal{E}_B \to \mathcal{R}_B \Leftrightarrow 0 \le \alpha_i \le 1 \to \alpha_i = 0 \Leftrightarrow f_1(\boldsymbol{x}_i) = -1 \to f_1(\boldsymbol{x}_i) < -1$.

ii. If $\mathcal{L}_B \neq \varnothing$, then the sample $\boldsymbol{x}_i$ $(i \in \mathcal{L}_B)$ from the set $\mathcal{L}_B$ might go into the set $\mathcal{E}_B$, i.e.,

Event 3 $\mathcal{L}_B \to \mathcal{E}_B \Leftrightarrow \alpha_i = 1 \to 0 \le \alpha_i \le 1 \Leftrightarrow f_1(\boldsymbol{x}_i) > -1 \to f_1(\boldsymbol{x}_i) = -1$.

iii. If $\mathcal{R}_B \neq \varnothing$, then the sample $\boldsymbol{x}_i$ $(i \in \mathcal{R}_B)$ from the set $\mathcal{R}_B$ might go into the set $\mathcal{E}_B$, i.e.,

Event 4 $\mathcal{R}_B \to \mathcal{E}_B \Leftrightarrow \alpha_i = 0 \to 0 \le \alpha_i \le 1 \Leftrightarrow f_1(\boldsymbol{x}_i) < -1 \to f_1(\boldsymbol{x}_i) = -1$.

The Second QPP Likewise, the second QPP (16) mainly depends on the samples at the elbow $\mathcal{E}_A$. When the regularization parameter λ_2 changes, the sample index sets $\mathcal{L}_A$, $\mathcal{E}_A$ and $\mathcal{R}_A$ will change accordingly. In the same way, we consider all possible events from the following three scenarios.

i. If $\mathcal{E}_A \neq \varnothing$, then the sample $\boldsymbol{x}_i$ $(i \in \mathcal{E}_A)$ from the set $\mathcal{E}_A$ might go into sets $\mathcal{L}_A$ and $\mathcal{R}_A$, i.e.,

Event 1 $\mathcal{E}_A \rightarrow \mathcal{L}_A \Leftrightarrow 0 \leq \gamma_i \leq 1 \rightarrow \gamma_i = 1 \Leftrightarrow f_2(\boldsymbol{x}_i) = 1 \rightarrow f_2(\boldsymbol{x}_i) < 1.$
Event 2 $\mathcal{E}_A \rightarrow \mathcal{R}_A \Leftrightarrow 0 \leq \gamma_i \leq 1 \rightarrow \gamma_i = 0 \Leftrightarrow f_2(\boldsymbol{x}_i) = 1 \rightarrow f_2(\boldsymbol{x}_i) > 1.$

ii. If $\mathcal{L}_A \neq \varnothing$, then the sample $\boldsymbol{x}_i$ $(i \in \mathcal{L}_A)$ from the set $\mathcal{L}_A$ might go into the set $\mathcal{E}_A$, i.e.,

Event 3 $\mathcal{L}_A \rightarrow \mathcal{E}_B \Leftrightarrow \gamma_i = 1 \rightarrow 0 \leq \gamma_i \leq 1 \Leftrightarrow f_2(\boldsymbol{x}_i) < 1 \rightarrow f_2(\boldsymbol{x}_i) = 1.$

iii. If $\mathcal{R}_A \neq \varnothing$, then the sample $\boldsymbol{x}_i$ $(i \in \mathcal{R}_A)$ from the set $\mathcal{R}_B$ might go into the set $\mathcal{E}_B$, i.e.,

Event 4 $\mathcal{R}_A \rightarrow \mathcal{E}_A \Leftrightarrow \gamma_i = 0 \rightarrow 0 \leq \gamma_i \leq 1 \Leftrightarrow f_2(\boldsymbol{x}_i) > 1 \rightarrow f_2(\boldsymbol{x}_i) = 1.$

4.2 Piecewise Linear w.r.t. the Regularization Parameter

The First QPP For convenience, let $\mathcal{L}_B^l$, $\mathcal{E}_B^l$ and $\mathcal{R}_B^l$ respectively denote the sample index sets of the first QPP (3) after the occurrence of the lth event. The number of elements in each set is denoted by $|\cdot|$. We use $\boldsymbol{x}_i^{\mathcal{E}_B^l}$ and $\mathbf{e}_B^l \in \mathbb{R}^{m \times n_B^l}$ to represent ith sample and the matrix composed of samples from the corresponding index set $\mathcal{E}_B^l$, respectively. In particular, let $n_B^l = |\ \mathbf{e}_B^l\ |$ and $\mathbf{G}_E = [\mathbf{E}_B^l\ \mathbf{e}_{n_B^l}]$.

Theorem 1 (Piecewise Linear Theory of the First QPP) *For the first QPP* (3)*, when* $\lambda_1^{l+1} < \lambda_1 < \lambda_1^l$*, let*

$$\bar{\mathbf{A}}^l = \mathbf{G}_E^l \left(\mathbf{H}^{\mathrm{T}}\mathbf{H} + \delta\mathbf{I}\right)^{-1} \left(\mathbf{G}_E^l\right)^{\mathrm{T}},$$

and then we can get that the Lagrangian multipliers α_i $(i \in \mathcal{E}_B)$ *are piecewise linear w.r.t. the regularization parameter* λ_1*, i.e.,*

$$\alpha_i = \alpha_i^l - \left(\lambda_1^l - \lambda_1\right)\theta_i^l, \tag{29}$$

where θ_i^l *is the* i*th element of the vector* $\boldsymbol{\theta}^l$*.*

$$\boldsymbol{\theta}^l = \left(\bar{A}\right)^{-1} \mathbf{e}_{n_B^l}. \tag{30}$$

Proof The following is to prove Theorem 1, i.e., the Lagrangian multipliers α_i are piecewise linear w.r.t. the regularization parameter λ_1.

According to Eq. (10), its lth step function can be obtain

$$f_1^l(\boldsymbol{x}) = -\frac{1}{\lambda_1^l}\left[\boldsymbol{x}^{\mathrm{T}}\ 1\right]\left(\mathbf{H}^{\mathrm{T}}\mathbf{H} + \delta\mathbf{I}\right)^{-1}\mathbf{G}^{\mathrm{T}}\boldsymbol{\alpha}^l. \tag{31}$$

Then, it is easy to obtain

$$\begin{aligned} f_1(\boldsymbol{x}) &= \frac{\lambda_1^l}{\lambda_1} f_1^l(\boldsymbol{x}) + f_1(\boldsymbol{x}) - \frac{\lambda_1^l}{\lambda_1} f_1^l(\boldsymbol{x}) \\ &= \frac{1}{\lambda_1}\left[\lambda_1^l f_1^l(\boldsymbol{x}) + \left[\boldsymbol{x}^{\mathrm{T}}\ 1\right]\left(\mathbf{H}^{\mathrm{T}}\mathbf{H} + \delta\mathbf{I}\right)^{-1}\mathbf{G}^{\mathrm{T}}\left(\boldsymbol{\alpha}^l - \boldsymbol{\alpha}\right)\right]. \end{aligned} \tag{32}$$

For $\forall i \in \mathcal{B}$, there are three special scenarios without considering any possible events, i.e.,

- If $i \in \mathcal{L}_B^l$, then $\alpha_i = \alpha_i^l = 1$.
- If $i \in \mathcal{E}_B^l$, then $f_1(\boldsymbol{x}) = f_1^l(\boldsymbol{x}) = -1$.
- If $i \in \mathcal{R}_B^l$, then $\alpha_i = \alpha_i^l = 0$.

Therefore, Eq. (32) can be simplified

$$\begin{aligned} \mathbf{G}_E^l \left(\mathbf{H}^{\mathrm{T}}\mathbf{H} + \delta\mathbf{I}\right)^{-1} \left(\mathbf{G}_E^l\right)^{\mathrm{T}} \left(\boldsymbol{\alpha}_E{}^l - \boldsymbol{\alpha}_E\right) &= \bar{\mathbf{A}}^l \left(\boldsymbol{\alpha}_E{}^l - \boldsymbol{\alpha}_E\right) \\ &= \left(\lambda_1^l - \lambda_1\right) \mathbf{e}_{n_B^l}. \end{aligned} \tag{33}$$

If $\bar{\mathbf{A}}^l$ is invertible, the we can obtain

$$\begin{aligned} \boldsymbol{\alpha}_E &= \boldsymbol{\alpha}_E^l - \left(\lambda_1^l - \lambda_1\right) \left(\bar{\mathbf{A}}^l\right)^{-1} \mathbf{e}_{n_B^l} \\ &= \boldsymbol{\alpha}_E^l - \left(\lambda_1^l - \lambda_1\right) \boldsymbol{\theta}^l. \end{aligned} \tag{34}$$

To sum up, Theorem 1 is proved. □

Corollary 1 (Corollary to Theorem 1) *According to Theorem 1, the recursive expression of hyperplane function* $f_1(\boldsymbol{x})$ *is*

$$f_1(\boldsymbol{x}) = \frac{1}{\lambda_1} \left[\lambda_1^l f_1^l(\boldsymbol{x}) + \left(\lambda_1^l - \lambda_1\right) h^l(\boldsymbol{x})\right], \tag{35}$$

where

$$h^l(\boldsymbol{x}) = \left[\boldsymbol{x}^{\mathrm{T}}\ 1\right] \left(\mathbf{H}^{\mathrm{T}}\mathbf{H} + \delta\mathbf{I}\right)^{-1} \left(\mathbf{G}_E^l\right)^{\mathrm{T}} \boldsymbol{\theta}^l. \tag{36}$$

The Second QPP For convenience, let $\mathcal{L}_A^l$, $\mathcal{E}_A^l$ and $\mathcal{R}_A^l$ respectively represent the sample index sets of the second QPP (16) after the occurrence of the lth event. We use $\boldsymbol{x}_i^{\mathcal{E}_A^l}$ and $\mathbf{e}_A^l \in \mathbb{R}^{m \times n_A^l}$ to represent ith sample and the matrix composed of samples from the corresponding index set $\mathcal{E}_A^l$, respectively. In particular, let $n_A^l = \mid \mathcal{E}_A^l \mid$ and $\mathbf{P}_E = [\mathbf{E}_A^l\ \mathbf{e}_{n_A^l}]$.

Theorem 2 (Piecewise Linear Theory of the Second QPP) *For the second QPP* (16)*, when* $\lambda_2^{l+1} < \lambda_2 < \lambda_2^l$*, let*

$$\bar{\mathbf{B}}^l = \mathbf{P}_E^l \left(\mathbf{Q}^{\mathrm{T}}\mathbf{Q} + \delta\mathbf{I}\right)^{-1} \left(\mathbf{P}_E^l\right)^{\mathrm{T}},$$

and then we can get that the Lagrangian multipliers γ_i $(i \in \mathcal{E}_A)$ *are piecewise linear w.r.t. the regularization parameter* λ_2*, i.e.,*

$$\boldsymbol{\gamma}_i = \boldsymbol{\gamma}_i^l - \left(\lambda_2^l - \lambda_2\right) \vartheta_i^l, \tag{37}$$

where ϑ_i^l *is the* i*th element of the vector* $\boldsymbol{\vartheta}^l$*.*

$$\boldsymbol{\vartheta}^l = \left(\bar{\mathbf{B}}\right)^{-1} \boldsymbol{e}_{n_A}^l. \tag{38}$$

Proof The proof of Theorem 2 is similar to that of Theorem 1, and it is listed in Appendix A in detail. □

Corollary 2 (Corollary to Theorem 2) *According to Theorem 2, the recursive expression of hyperplane function* $f_2(\boldsymbol{x})$ *is*

$$f_2(\boldsymbol{x}) = \frac{1}{\lambda_2}\left[\lambda_2^l f_2^l(\boldsymbol{x}) - \left(\lambda_2^l - \lambda_2\right) g^l(\boldsymbol{x})\right], \tag{39}$$

where

$$g^l(\boldsymbol{x}) = \left[\boldsymbol{x}^{\mathrm{T}}\ 1\right]\left(\mathbf{Q}^{\mathrm{T}}\mathbf{Q} + \delta\mathbf{I}\right)^{-1}\left(\mathbf{P}_E^l\right)^{\mathrm{T}}\boldsymbol{\vartheta}^l. \tag{40}$$

Notably, the establishment of piecewise linear theory in Theorems 1 and 2 makes us only solve the breakpoints to obtain the entirely regularized solution path. It not only greatly extends the search space to $(0, +\infty)$, but also improves the solving efficiency compared with the grid search method.

5 Initialization

Before introducing the solution path algorithm, we propose a simple yet efficient initialization. It is first proved that when the regularization parameter approaches the positive infinity, the Lagrangian multiplier is 1. Thus, we can establish the initial state of the sets defined above. Then, the corresponding initialization of two sub-optimization problems is designed without solving QPPs.

Initialization of the First QPP Theorem 3 can be used to prove that when the regularization parameter λ_1 approaches positive infinity, the Lagrangian multiplier α_i is 1.

Theorem 3 *For the first QPP* (3)*, when* λ_1 *approaches infinity, the Lagrangian multipliers* α_i $(i \in \mathcal{B})$ *are always equal to 1, i.e., if* $\lambda_1 \to +\infty$*, then* $\alpha_i = 1$ $(i \in \mathcal{B})$.

Proof From Eq. (10), when λ_1 approaches infinity, it is easy to get $-f_1(\boldsymbol{x}_i) = 0 > -1$ $(i \in \mathcal{B})$. According to the definition of index set $\mathcal{L}_B$ in Fig. 2, we can obtain directly $\alpha_i = 1$ $(i \in \mathcal{B})$. Therefore, Theorem 3 can be proved. □

Initialization of the Second QPP Theorem 4 can be used to prove that when the regularization parameter λ_2 approaches the positive infinity, the Lagrangian multiplier γ_i is 1.

Theorem 4 *For the second QPP* (16)*, when* λ_2 *approaches infinity, the Lagrangian multipliers* γ_i $(i \in \mathcal{A})$ *are always equal to 1, i.e., if* $\lambda_2 \to +\infty$*, then* $\gamma_i = 1 (i \in \mathcal{A})$.

Proof The proof of Theorem 4 is similar to that of Theorem 3. From Eq. (23), when λ_2 approaches infinity, it is easy to get $f_2(\boldsymbol{x}_i) = 0 < 1$ $(i \in \mathcal{A})$. According to the definition of index set $\mathcal{L}_A$ in Fig. 3, we can obtain directly $\gamma_i = 1$ $(i \in \mathcal{A})$. Therefore, Theorem 4 can be proved. □

Initialization Algorithm We have got the initial Lagrangian multipliers $\alpha_i^0 = 1$ $(i \in \mathcal{B})$ and $\beta_i^0 = 1$ $(i \in \mathcal{A})$ when the regularization parameters are sufficiently large through Theorems 3 and 4. Additionally, all the sample points lie in the *left* of the elbow from Figs. 2 and 3. The general idea of the initialization process is to initialize regularization parameters λ_1^0 and λ_2^0 when the *first* sample point goes into the *elbow* from its left.

Specifically, we first iterate over all the sample points to obtain the corresponding initial candidate values for regularization parameters by assuming that they enter the elbow from the left set of the elbow. The largest regularization parameter candidate values are then assigned

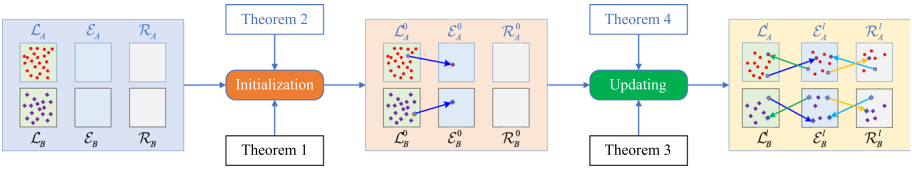


Fig. 5 Flowchart of the proposed fast regularization parameter tuning algorithm for TSVM (TSVMPath). It mainly consists of two steps: initialization as shown in Algorithm 1 and updating as shown in Algorithm 2, where the initialization aims to assign initial values to parameters by solving event 1 and the updating process aims to find out the entire solution path by reducing the value of the regularization parameter

to initial ones λ_1^0 and λ_2^0, respectively. Thus, we can extend the search space of regularization parameters to $(0, +\infty)$ without solving QPPs.

Algorithm 1 describes the initialization process of the first QPP (3) in detail. The initialization process of the second QPP (16) is exactly in the same way.

Algorithm 1: Initialization Algorithm of QPP (3)

Input: Training sample matrices $\mathbf{A}$ and $\mathbf{B}$ and the system parameter δ.
Output: Initial parameters λ_1^0, $\boldsymbol{\alpha}^0$, $\boldsymbol{u}^0$, $\mathcal{L}_B^0$, $\mathcal{E}_B^0$ and $\mathcal{R}_B^0$.

$\mathbf{H} \leftarrow [\mathbf{A}\ \mathbf{e}_{n_A}]$, $n_A \leftarrow \text{size}(\mathbf{A}, 1)$;
$\mathbf{G} \leftarrow [\mathbf{B}\ \mathbf{e}_{n_B}]$, $n_B \leftarrow \text{size}(\mathbf{B}, 1)$;
$\mathbf{I} \leftarrow \text{eye}(m+1)$, $m \leftarrow \text{size}(\mathbf{A}, 2)$;
$\mathcal{L}_B^0, \mathcal{E}_B^0, \mathcal{R}_B^0 \leftarrow \{1, 2, \ldots, n_B\}, \varnothing, \varnothing$; // By Theorem 3.
$\boldsymbol{\alpha}^0 \leftarrow \mathbf{e}_{n_B}$; // By Theorem 3.
$\lambda_1^0 \leftarrow 0$; // Initialize to a minimum.
foreach $i \in \mathcal{B}$ **do**
  $\lambda \leftarrow \left[\boldsymbol{x}_i^{\mathrm{T}}\ 1\right]\left(\mathbf{H}^{\mathrm{T}}\mathbf{H} + \delta\mathbf{I}\right)^{-1}\mathbf{G}^{\mathrm{T}}\boldsymbol{\alpha}$; // Solve $\mathcal{L}_B^0 \to \mathcal{R}_B^0$ using Eq. (10).
  if $\lambda_1^0 < \lambda$ **then**
    $\lambda_1^0 \leftarrow \lambda$; // Take the largest λ.
    $p \leftarrow i$; // Remember the sample point.
  end
end
$\boldsymbol{u}^0 \leftarrow -\frac{1}{\lambda_1^0}\left(\mathbf{H}^{\mathrm{T}}\mathbf{H} + \delta\mathbf{I}\right)^{-1}\mathbf{G}^{\mathrm{T}}\boldsymbol{\alpha}^0$; // By Eq. (9)
Take the sample point p out of $\mathcal{L}_B^0$ and put it in $\mathcal{E}_B^0$;
return λ_1^0, $\boldsymbol{\alpha}^0$, $\boldsymbol{u}^0$, $\mathcal{L}_B^0$, $\mathcal{E}_B^0$, $\mathcal{R}_B^0$;

6 Fast Regularization Parameter Tuning Algorithm

After initialization, we can get initial parameters using Algorithm 1. To design the fast regularization parameter tuning algorithm for TSVM, we need to update the $(l+1)$th step parameter for iteration. Furthermore, it is feasible to set the appropriate termination conditions.

Definition 2 When the regularization parameter changes, several events may occur. This paper defines the **first event** as the event that has the highest priority to occur.

6.1 Finding λ_1^{l+1}

For the first QPP (3), when the λ_1 reduces from $+\infty$ to 0, there are four types of events that can occur as described in Sect. 4.1. We consider these events separately below.

Event 1 If $\mathcal{E}_B^l \to \mathcal{L}_B^l$, then $\alpha_i = 1$. From Eq. (29), when this event occurs, the regularization parameter can be updated to

$$\lambda_1^{(1)} = \max_{i \in \mathcal{E}_B^l} \left\{ \lambda_1^l - \frac{\alpha_i^l - \alpha_i}{\theta_i^l} \right\} = \max_{i \in \mathcal{E}_B^l} \left\{ \lambda_1^l - \frac{\alpha_i^l - 1}{\theta_i^l} \right\}, \tag{41}$$

where $\theta_i^l < 0$.

Event 2 If $\mathcal{E}_B^l \to \mathcal{R}_B^l$, then $\alpha_i = 0$. From Eq. (29), when this event occurs, the regularization parameter can be updated to

$$\lambda_1^{(2)} = \max_{i \in \mathcal{E}_B^l} \left\{ \lambda_1^l - \frac{\alpha_i^l - \alpha_i}{\theta_i^l} \right\} = \max_{i \in \mathcal{E}_B^l} \left\{ \lambda_1^l - \frac{\alpha_i^l}{\theta_i^l} \right\}, \tag{42}$$

where $\theta_i^l > 0$.

Event 3 If $\mathcal{L}_B^l \to \mathcal{E}_B^l$, then $-f_1(\boldsymbol{x}_i) = 1$. From Eq. (35), when this event occurs, the regularization parameter can be updated to

$$\lambda_1^{(3)} = \max_{i \in \mathcal{L}_B^l} \left\{ \lambda_1^l \frac{f_1^l(\boldsymbol{x}_i) + h^l(\boldsymbol{x}_i)}{f_1(\boldsymbol{x}_i) + h^l(\boldsymbol{x}_i)} \right\} = \max_{i \in \mathcal{L}_B^l} \left\{ \lambda_1^l \frac{f_1^l(\boldsymbol{x}_i) + h^l(\boldsymbol{x}_i)}{-1 + h^l(\boldsymbol{x}_i)} \right\}. \tag{43}$$

Event 4 If $\mathcal{R}_B^l \to \mathcal{E}_B^l$, then $-f_1(\boldsymbol{x}_i) = 1$. From Eq. (35), when this event occurs, the regularization parameter can be updated to

$$\lambda_1^{(4)} = \max_{i \in \mathcal{R}_B^l} \left\{ \lambda_1^l \frac{f_1^l(\boldsymbol{x}_i) + h^l(\boldsymbol{x}_i)}{f_1(\boldsymbol{x}_i) + h^l(\boldsymbol{x}_i)} \right\} = \max_{i \in \mathcal{R}_B^l} \left\{ \lambda_1^l \frac{f_1^l(\boldsymbol{x}_i) + h^l(\boldsymbol{x}_i)}{-1 + h^l(\boldsymbol{x}_i)} \right\}. \tag{44}$$

Therefore, the first event e_1 can be selected and then the next step regularization parameter λ_1^{l+1} can be updated accordingly. At the same time, the Lagrangian multiplier α_i, index set $\mathcal{L}_B^l, \mathcal{E}_B^l, \mathcal{R}_B^l$ and other parameters are updated according to the first event e_1.

$$e_1 = \arg\max_i \left\{ \lambda_1^{(i)} \mid i = 1, 2, 3, 4 \right\}, \tag{45}$$

$$\lambda_1^{l+1} = \max \left\{ \lambda_1^{(i)} \mid i = 1, 2, 3, 4 \right\}. \tag{46}$$

6.2 Finding λ_2^{l+1}

Similarly, for the second QPP (16), when the λ_2 reduces from $+\infty$ to 0, there are also four types of events that can occur as described in Sect. 4.1. We consider these events separately below.

Event 1 If $\mathcal{E}_A^l \to \mathcal{L}_A^l$, then $\gamma_i = 1$. From Eq. (37), when this event occurs, the regularization parameter can be updated to

$$\lambda_2^{(1)} = \max_{i \in \mathcal{E}_A^l} \left\{ \lambda_1^l - \frac{\gamma_i^l - \gamma_i}{\vartheta_i^l} \right\} = \max_{i \in \mathcal{E}_A^l} \left\{ \lambda_2^l - \frac{\gamma_i^l - 1}{\vartheta_i^l} \right\}, \tag{47}$$

where $\vartheta_i^l < 0$.

Event 2 If $\mathcal{E}_A^l \to \mathcal{R}_A^l$, then $\gamma_i = 0$. From Eq. (37), when this event occurs, the regularization parameter can be updated to

$$\lambda_2^{(2)} = \max_{i \in \mathcal{E}_A^l} \left\{ \lambda_2^l - \frac{\gamma_i^l - \gamma_i}{\vartheta_i^l} \right\} = \max_{i \in \mathcal{E}_A^l} \left\{ \lambda_2^l - \frac{\gamma_i^l}{\vartheta_i^l} \right\}, \tag{48}$$

where $\vartheta_i^l > 0$.

Event 3 If $\mathcal{L}_A^l \to \mathcal{E}_A^l$, then $f_2(\boldsymbol{x}_i) = 1$. From Eq. (39), when this event occurs, the regularization parameter can be updated to

$$\lambda_2^{(3)} = \max_{i \in \mathcal{L}_A^l} \left\{ \lambda_2^l \frac{f_2^l(\boldsymbol{x}_i) - g^l(\boldsymbol{x}_i)}{f_2(\boldsymbol{x}_i) - g^l(\boldsymbol{x}_i)} \right\} = \max_{i \in \mathcal{L}_A^l} \left\{ \lambda_2^l \frac{f_2^l(\boldsymbol{x}_i) - g^l(\boldsymbol{x}_i)}{1 - g^l(\boldsymbol{x}_i)} \right\}. \tag{49}$$

Event 4 If $\mathcal{R}_A^l \to \mathcal{E}_A^l$, then $f_2(\boldsymbol{x}_i) = 1$. From Eq. (39), when this event occurs, the regularization parameter can be updated to

$$\lambda_2^{(4)} = \max_{i \in \mathcal{R}_A^l} \left\{ \lambda_2^l \frac{f_2^l(\boldsymbol{x}_i) - g^l(\boldsymbol{x}_i)}{f_2(\boldsymbol{x}_i) - g^l(\boldsymbol{x}_i)} \right\} = \max_{i \in \mathcal{R}_A^l} \left\{ \lambda_2^l \frac{f_2^l(\boldsymbol{x}_i) - g^l(\boldsymbol{x}_i)}{1 - g^l(\boldsymbol{x})} \right\}. \tag{50}$$

Therefore, the first event e_2 can be selected and then the next step regularization parameter λ_2^{l+1} can be updated accordingly. At the same time, the Lagrangian multiplier γ_i, index set $\mathcal{L}_A^l, \mathcal{E}_A^l, \mathcal{R}_A^l$ and other parameters are updated according to the first event e_2.

$$e_2 = \arg\max_i \left\{ \lambda_2^{(i)} \mid i = 1, 2, 3, 4 \right\}, \tag{51}$$

$$\lambda_2^{l+1} = \max \left\{ \lambda_2^{(i)} \mid i = 1, 2, 3, 4 \right\}. \tag{52}$$

Algorithm 2: Parameter Update Algorithm of QPP (3)

Input: The lth step parameters, regularized threshold t and the number of maximum iterations $l_{\max}$.
Output: The $(l+1)$th step parameters.

```
while λ_1^l ≤ t and l ≤ l_max do
    Obtain Ā^l, e^l, θ_0^l and θ_i^l (i ∈ E_B^l) according to Theorem 1;
    Calculate λ_1^(i) (i = 1, 2, 3, 4) using Eqs. (41), (42), (43), (44);
    Determine the first event e_1 by Eq. (45) and then obtain λ_1^{l+1} by Eq. (46);
    Obtain and update α^{l+1}, u^{l+1}, L_B^{l+1}, E_B^{l+1} and R_B^{l+1};
    l ← l + 1;
end
return (l + 1)th step parameters;
```

6.3 Process of TSVMPath

For a binary dataset $\mathcal{D}$, the two classes of samples are denoted by $\mathcal{A}$ and $\mathcal{B}$ and labeled by "+1" and "−1", respectively. As shown in Fig. 5, TSVMPath mainly consists of initialization and updating. Next, we train it through the following steps:

Step 1 We randomly divide the dataset into training set and test set accordingly to the partition ratio r. Then, divide the training set into ten parts to carry out 10-fold cross-validation.

Step 2 One fold is selected sequentially as the validation set and the rest as the training.

Step 3 Calculate the regularized solution path for the first QPP (3).

Step 3.1 Invoke Algorithm 1 to obtain the initial parameters λ_1^0, $\boldsymbol{u}^0$, $\boldsymbol{\alpha}^0$.

Step 3.2 Determine the $(l+1)$th step regularization parameter λ_1^{l+1} based on Sect. 6.1.

Step 3.3 Invoke Algorithm 2 to update the $(l+1)$th step parameters according to the first event e_1.

Step 3.4 If the λ_1^{l+1} is less than the threshold t or the maximum iterations exceeds the top, then stop the loop and go to **Step 3**, otherwise set $l = l+1$, go to **Step 3.2** and continue the next loop.

Step 4 Obtain the entirely regularized solution path for the second QPP (16) similar to **Step 2**.

Step 5 Test on the validation set to choose the optimal combination of parameters λ_1^* and λ_2^*. Obtain the hyperplanes f_1 and f_2 by the optimal parameters. We use the decision function to test any sample $\boldsymbol{x}$ in the test set and then obtain the classification accuracy.

$$f(\boldsymbol{x}) = \begin{cases} +1, \, c < d \\ -1, \, c > d \\ 0, \, \text{otherwise.} \end{cases} \tag{53}$$

where c and d denote the distance from the sample $\boldsymbol{x}$ to two hyperplanes respectively, i.e..

$$c = \frac{| \, f_1(\boldsymbol{x}) \, |}{\|\boldsymbol{w}_1\|}, \tag{54}$$

$$d = \frac{| \, f_2(\boldsymbol{x}) \, |}{\|\boldsymbol{w}_2\|}. \tag{55}$$

Step 6 Return **Step 2** for the next fold until training ten folds.

Step 7 The average value of the ten times classification accuracy on the same test set is taken as the final classification accuracy.

7 Experiments

To evaluate and analyze the performance of the proposed algorithm, we first verify the piecewise linear theory and then compare it with different baselines in terms of the prediction accuracy and the computational overhead.

7.1 Setup

Using MATLAB R2021a, all the experiments are performed on the personal computer equipped with Intel (R) Core (TM) i7-7500U 2.90GHz CPU and 8GB of RAM.

Table 1 Properties of 8 machine learning UCI datasets used in this paper

#	Datasets	Number of samples			Dimension
		Positive	Negative	Total	
1	Blood	570	178	748	4
2	Bupa	145	200	345	6
3	Cancer	444	239	683	9
4	Diabetes	268	500	768	8
5	Haberman	225	81	306	3
6	Heartstatlog	150	120	270	13
7	WBC	444	239	683	9
8	WDBC	357	212	569	30

7.1.1 Datasets

We evaluate TSVMPath on 8 binary UCI datasets,[2] i.e., Blood, Bupa, Cancer, Diabetes, Haberman, Heartstatlog, WBC and WDBC. The positive sample number, negative sample number, total sample number and feature dimension of the 8 UCI datasets are shown in Table 1. To ensure the diversity of datasets, we selected datasets with different feature dimensions. These datasets have different numbers of instances, ranging from hundreds to thousands. Among these benchmark datasets, the dataset with the smallest feature dimension is `Haberman`, whose feature dimension is 3, and the dataset with the largest feature dimension is `WDBC`, up to 30.

7.1.2 Implementation Details

To avoid the matrix irreversible problem, we set $\delta = 10^{-8}$. For the fast regularization parameter tuning algorithm, the corresponding solving loop stops when the regularization parameter λ is less than the threshold 10^{-4} or the maximum number of iterations exceeds 3000. For the contrast experiments, we set the initial parameters $\lambda_1^0 = 1000$ and $\lambda_2^0 = 1000$. The proposed algorithm is compared with TSVM [5], weighted linear loss TSVM (WLTSVM) [36] and least-square projection TSVM (LSPTSVM) [32]. The original TSVM is solved using `quadprog` toolbox of MATLAB.

In this paper, we randomly divide each dataset into the training set and test set according to the ratio $r = 3 : 1$. For each training set, the proposed algorithm is trained by tenfold cross-validation to select the optimal parameter pair (λ_1, λ_2) for testing.

7.2 Results and Analysis

We first visualize the entirely regularized solution path of two sub-optimization problems to verify the pairwise linear theory. Then, we analyze the first event and compare the prediction accuracy performance and training time with state-of-the-art methods. Finally, we discuss the computational overhead and time complexity of TSVMPath.

[2] Download UCI datasets at https://archive.ics.uci.edu/ml/index.php.

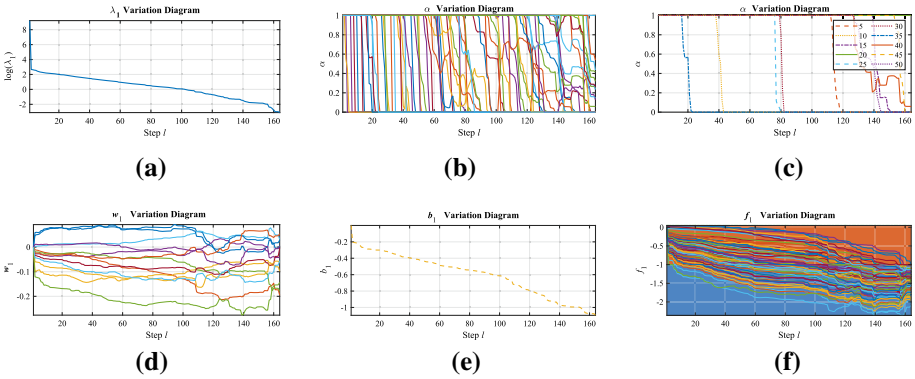


Fig. 6 Solution path diagrams of the first QPP (3) (Heartstatlog): **a**–**f** variation diagrams of λ_1, $\boldsymbol{\alpha}$, $\alpha_{(5,\,10,\,15,\ldots,50)}$, $\boldsymbol{w}_1$, b_1 and f_1 w.r.t. the step l, respectively

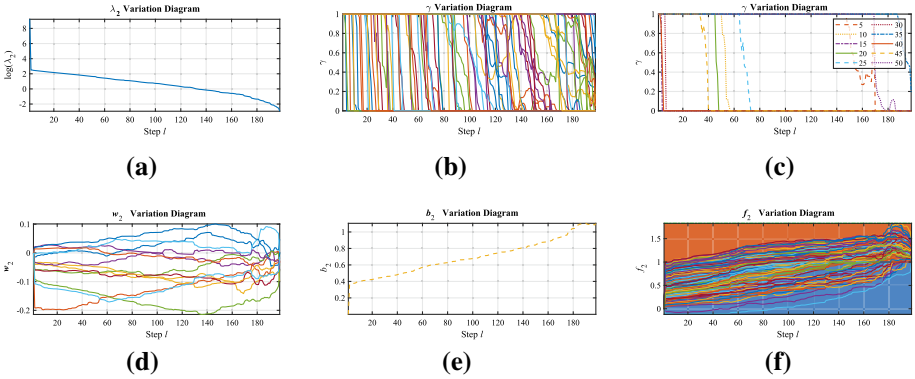


Fig. 7 Solution path diagrams of the second QPP (16) (Heartstatlog): **a**–**f** variation diagrams of λ_2, $\boldsymbol{\gamma}$, $\gamma_{(5,\,10,\,15,\ldots,50)}$, $\boldsymbol{w}_2$, b_2 and f_2 w.r.t. the step l, respectively

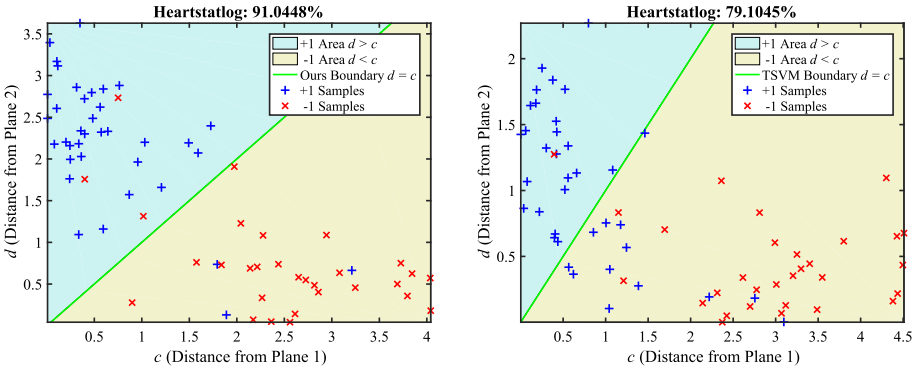
7.2.1 Regularized Solution Path

To test the piecewise linear theory, taking the dataset `Heartstatlog` as a verification example, we obtain the solution of two QPPs as shown in Figs. 6 and 7. Figures 6a and 7a show the regularization parameter changes of the two QPPs respectively, where the initialization parameters of two QPPs are $\lambda_1^0 = 14.4014$ and $\lambda_2^0 = 12.1583$ respectively. It is obvious that the regularization parameter gradually reduces to less than the threshold value t and then stops the iteration. It is also explicit that the Lagrangian multipliers are piecewise linear w.r.t. regularization parameters from Figs. 6b and 7b. Furthermore, to clearly show the piecewise linear solution path of the two QPPs, we select several sample points to show its corresponding Lagrangian multipliers in Figs. 6c and 7c. It can be unambiguously seen that the experimental results are consistent with piecewise linear theory in Sect. 4.

Form Figs. 6b and 7b, the Lagrangian multipliers can be vaguely seen that almost all of them tend to go from 1 to 0. In general, we can see that the Lagrangian multipliers tend to go from 1 to 0 in Fig. 7c. However, it is also clear from Fig. 6c that the Lagrangian multiplier may also tend to increase from 0. Indeed, this is related to the defined events in

Table 2 First events of the solution path on Heartstatlog

#	Event 1		Event 2		Event 3		Event 4		# Total
	#	Freq.	#	Freq.	#	Freq.	#	Freq.	
QPP 1	26	0.0844	123	0.3994	107	0.3474	52	0.1688	308
QPP 2	6	0.0222	124	0.4593	118	0.4370	22	0.0815	270



(a) TSVMPath **(b)** TSVM (Grid Search)

Fig. 8 Predictive decision diagram (Heartstatlog): **a**, **b** plots of ours and TSVM by solving QPPs, respectively. The blue "+" samples and the red "×" samples are labeled as "+1" and "−1" respectively; the horizontal and vertical axes represent the distance between the samples and the two hyperplanes respectively, and the green divider line indicates the equal distance between the two hyperplanes and the samples

Sect. 4. The experimental results are consistent with the algorithm design because we allow the Lagrangian multipliers to perform arbitrary changes between 0 and 1.

Additionally, the solution path w.r.t. $\boldsymbol{w}_1, b_1, f_1$ and $\boldsymbol{w}_2, b_2, f_2$ are shown in Figs. 6d–f and 7d–f. As shown in Fig. 6f, we can see that the value of f_1 has a trend from greater than -1 to less than -1. This is because at the beginning, all sample points are on the *left* set of the elbow, and as the algorithm iterates, the sample gradually moves from the *left* of the elbow to that of the *right*, as shown in Fig. 2.

7.2.2 First Events Analysis

The variation trend of Lagrangian multipliers can be intuitively reflected from the distribution of the first events. Similarly, Table 2 shows an example to count the number of first events for the two QPPs on the dataset `Heartstatlog`. From the data in the first row of Table 2, it can be seen that the frequencies of event 2 and event 3 are greater than that of event 1 and event 4. Therefore, it can be inferred that the samples in the first QPP tend to move closer to the index set $\mathcal{R}_B$ and the Lagrangian multipliers tend to move to decrease to 0 from the initial value 1. Moreover, it is just in the similar way for the second QPP from the data in the second row of Table 2.

The experimental results are consistent with the theory of the piecewise linear solution algorithm. According to Theorems 3 and 4, all the sample points are located in the set $\mathcal{L}_B$ and $\mathcal{L}_A$ respectively during initialization. Then, the sample points may go from $\mathcal{L}_B$ or $\mathcal{L}_A$ to the elbow, and the sample points at the elbow can go into $\mathcal{L}$ and $\mathcal{R}$, respectively. Similarly, sample

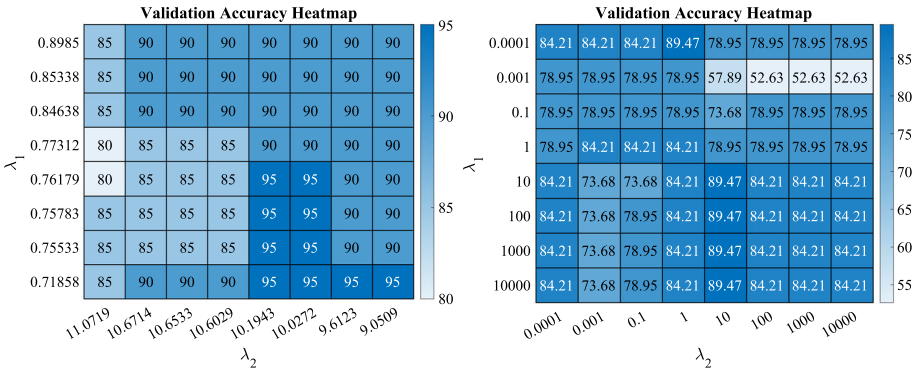


Fig. 9 Cross-validation accuracy heatmaps of **a** TSVMPath and **b** the grid search method for TSVM

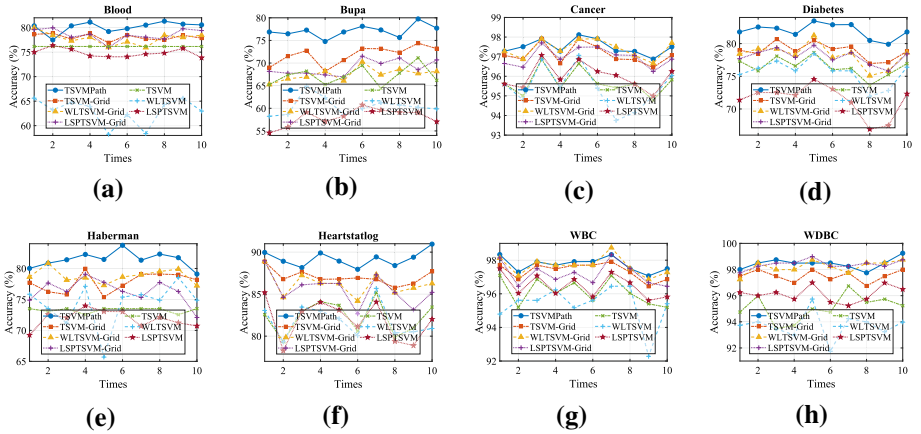


Fig. 10 Accuracy (%) curves on 8 UCI datasets: **a–l** the ten times predictive accuracy diagrams, in which blue circle lines, orange square lines, yellow triangle lines, purple plus lines and green cross lines, cyan plus lines and dark red pentagram lines are the prediction results of TSVMPath (ours), grid search methods for TSVM, WLTSVM [36] and LSPTSVM [32], and non-grid search methods for TSVM, WLTSVM [36] and LSPTSVM [32], respectively

points in $\mathcal{R}_A$ or $\mathcal{R}_B$ will have analogous events. We hope that all Lagrangian multipliers can change from 1 to 0 to obtain an entirely regularized solution path, which corresponds exactly to event 2 and event 3.

From Table 2, it can be inferred that the samples in two QPPs tend to move closer to the index set $\mathcal{L}_B$ and $\mathcal{L}_A$ respectively. At the same time, the Lagrangian multipliers tend to decrease to 0 from 1. Therefore, the results dovetailed with our expectations.

7.2.3 Prediction Accuracy

Similarly, taking the dataset `Heartstatlog` as an example, the predicted results of ours and TSVM by solving QPPs on the test dataset are shown in Fig. 8a, b in one experiment. In Fig. 8, the horizontal axis represents the distance between the predicted sample and the first hyperplane, while the vertical axis represents the distance between the predicted sample and

Table 3 Average prediction accuracy (%) of ten times on 8 UCI datasets

Dataset	Ours	Grid search methods			Non-grid search methods		
		TSVM	WLTSVM	LSPTSVM	TSVM	WLTSVM	LSPTSVM
Blood	**80.14** ± 2.66	78.08 ± 1.16	77.77 ± 1.84	78.55 ± 2.58	76.15 ± 0.00	62.92 ± 4.70	74.83 ± 0.97
Bupa	**77.08** ± 2.32	71.83 ± 3.68	67.52 ± 2.25	68.97 ± 2.04	67.20 ± 2.78	60.47 ± 2.24	58.05 ± 3.46
Cancer	**97.49** ± 0.62	97.12 ± 0.67	97.41 ± 0.75	96.99 ± 0.74	95.68 ± 0.90	95.41 ± 1.66	95.99 ± 1.00
Diabetes	**81.95** ± 2.05	78.91 ± 1.95	78.24 ± 3.15	78.05 ± 2.24	76.45 ± 2.83	75.60 ± 3.67	71.37 ± 4.47
Haberman	**81.43** ± 2.32	77.76 ± 2.38	78.77 ± 1.55	76.34 ± 4.25	73.27 ± 0.73	74.28 ± 8.58	71.87 ± 2.63
Heartstatlog	**89.20** ± 1.24	87.03 ± 1.27	86.20 ± 2.02	85.57 ± 2.91	82.27 ± 2.92	81.85 ± 2.53	81.91 ± 3.63
WBC	**97.74** ± 0.68	97.37 ± 0.92	97.66 ± 0.80	97.18 ± 0.74	96.12 ± 0.92	95.36 ± 3.09	96.47 ± 0.86
WDBC	**98.45** ± 0.70	97.52 ± 0.77	98.22 ± 0.98	98.40 ± 0.65	95.09 ± 1.61	93.71 ± 1.96	96.12 ± 0.88

Bold font indicates the best result

Table 4 Training time (s) on 8 UCI datasets

Dataset	Ours	Grid search methods			Non-grid search methods		
		TSVM	WLTSVM	LSPTSVM	TSVM	WLTSVM	LSPTSVM
Blood	**0.0038**	0.2530	0.0050	0.0157	0.3399	0.0057	0.0158
Bupa	**0.0040**	0.0588	0.0042	0.0052	0.0849	0.0048	0.0057
Cancer	**0.0055**	0.1826	0.0056	0.0182	0.2129	0.0060	0.0183
Diabetes	0.0080	0.2132	**0.0056**	0.0183	0.2398	0.0061	0.0195
Haberman	0.0048	0.0522	**0.0036**	0.0049	0.0869	0.0039	0.0049
Heartstatlog	0.0031	0.0383	**0.0030**	0.0038	0.0612	0.0034	0.0038
WBC	**0.0052**	0.1835	0.0057	0.0182	0.2068	0.0063	0.0190
WDBC	**0.0087**	0.1531	0.0144	0.0224	0.1638	0.0154	0.0230

Bold font indicates the best result

the second hyperplane. The positive samples are marked with a blue "+" and the negative samples with a red "×". For the prediction results, the positive sample area is cyan and the negative sample area is yellow. The two predicted sample areas are separated by green lines in the middle, and the predicted sample distances on the line are equal to the two hyperplanes. However, in the experiment, such sample points that are equidistant from two hyperplanes are rarely seen. In an experiment, the prediction accuracy of the proposed algorithm on the dataset Heartstatlog is 91.0448%, while that of SVM is 71.1045%. Therefore, it is proved that TSVMPath has better classification accuracy than TSVM. Additionally, the cross-validation results in Fig. 9 also show our superiority.

Figure 10 shows predicted accuracy on 8 UCI datasets in Table 1, where Fig. 10a–h are diagrams of ten times predicted accuracies on 8 UCI datasets using different methods, in which blue circle lines, orange square lines, yellow triangle lines, purple plus lines and green cross lines, cyan plus lines and dark red pentagram lines are the prediction results of TSVMPath (ours), grid search methods for TSVM, WLTSVM [36] and LSPTSVM [32], and non-grid search methods for TSVM, WLTSVM [36] and LSPTSVM [32], respectively. Table 3 shows the average accuracies on 8 UCI datasets using different methods. Our algorithm is not only optimal in all the datasets, but also the prediction performance of our algorithm is stable, as can be seen from the accuracy and standard deviation of 10 repeated experiments in Table 3.

From Fig. 10, both the grid search method and our solution path algorithm perform better than the non-parametric adjustment method, which shows the effectiveness of the former. On the datasets `Blood`, `Bupa`, `Diabates`, `Haberman` and `Heartstatlog`, TSVMPath shows great performance advantages. However, there is still a huge space to improve the performance on the other three datasets. We highlight that TSVMPath is a fast solution path algorithm for TSVM without solving QPPs. Notably, it achieves the best prediction performance, demonstrating the superiority of the proposed method.

7.2.4 Training Time Comparison

Table 4 shows the average training time comparison for 10 repeated experiments between the proposed algorithm and the other methods on 8 UCI datasets. For the fairness of experiments, we only count the average time to solve one programming problem. It can be seen that the training time of our algorithm is lower than others on five datasets, including `Blood`, `Bupa`, `Cancer`, `WBC` and `WDBC`. On the other three datasets, the training time of the proposed

method is comparable to that of WLTSVM. Neither of them (TSVMPath and WLTSVM) is solved by solving QPP, and the training time is shorter than that by solving QPP for TSVM. For example, the average time to adjust a parameter for TSVMPath is 0.0038s on the `Blood` dataset and 0.2530s with QPP for TSVM. Therefore, the significant advantage of our algorithm is the short training time for each solution, compared with solving QPP for TSVM.

7.2.5 Discussion

Computational Overhead For the grid search parameter optimization method, it is assumed that the regularization parameter decreases from 1000 and the step rate is 0.1, so it needs to fit TSVM 2×10^8 times to obtain optimal regularization parameters. However, the time required to fit a single TSVM using solving QPP will increase with the sample dimensions, as shown in Table 4. For some samples with higher dimensions, the training time can even reach several hours. The main factor restricting the efficiency of the algorithm is to solve the QPP problem. Therefore, the times of solving QPPs can be used to measure the computational overhead of the algorithm. Notably, the proposed algorithm is very effective without solving any QPP. Compared with solving a QPP, the total time for solving TSVMPath is about the same. However, the computational overhead of the grid search method increases exponentially with the adjustment of iteration step size and the change of initial point. As a result, grid search often cannot traverse the whole parameter space, resulting in suboptimal solutions. Notably, the proposed algorithm can sharply reduce the computational overhead of the grid search method without solving QPPs, and fully obtain the optimal solution by finding the break point based on the piecewise linear theory.
Time Complexity Since Algorithm 1 needs to solve linear equations of size n_B, its time complexity is $\mathcal{O}(n_B^2)$ at least. According to Hastie et al. [10], the time complexity of Algorithm 2 is $\mathcal{O}(cn_B^2 m + n_B m^2)$, where m is the average size of $\mathcal{E}$ and c is a small number. In summary, the time complexity of the whole algorithm is proportional to the square of the data size. Additionally, the total computation burden of the entire solution path algorithm is similar to that of a single TSVM fit. For the grid search method, we need to fit the TSVM n_{grid} times, and the corresponding time complexity is also n_{grid} times of TSVM fits, where n_{grid} is the granularity of the grid, e.g., n_{grid} is equal to 2×10^4 as analyzed above in this paper. Therefore, the solution path algorithm can greatly reduce the computational burden of parameter adjustment, with up to four orders of magnitude speed-up for the computational complexity compared with the grid search method.

8 Conclusion

In this paper, we develop a novel parameter tuning algorithm for TSVM. Two sub-optimization problems of TSVM are first transformed and the training samples are divided into different index sets. It is proved that the Lagrangian multiplier is piecewise linear w.r.t. the regularization parameter accordingly. Simulation results of 8 UCI datasets show that the Lagrangian multipliers in the two sub-models are piecewise linear w.r.t. regularization parameters, which lays a foundation for the further selection of regularization path algorithm and makes TSVM have stronger generalization performance. Experiments show that both the prediction accuracy and the training speed of TSVMPath are superior to that of the state-of-

the-art methods. Notably, since there is no need to solve QPPs, our computational overhead is greatly reduced compared with the grid search method.

In the future, the solution path algorithm for TSVM will be extremely generalized by extending the solution path algorithm of TSVM to multi-classification and nonlinear problems.

Acknowledgements This work was supported by the Natural Science Foundation of China (61203293, 61702164, 31700858), the Scientific and Technological Project of Henan Province (162102310461, 172102310535), Foundation of Henan Educational Committee (18A520015).

A Proof of Theorem 2

Proof The following is to prove Theorem 2, i.e., the Lagrangian multipliers γ_i are piecewise linear w.r.t. the regularization parameter λ_2.
According to Eq. 23, its lth step function can be obtain

$$f_2^l(\boldsymbol{x}) = \frac{1}{\lambda_2^l}\left[\boldsymbol{x}^{\mathrm{T}}\ 1\right]\left(\mathbf{Q}^{\mathrm{T}}\mathbf{Q}+\delta\mathbf{I}\right)^{-1}\mathbf{P}^{\mathrm{T}}\boldsymbol{\gamma}^l. \tag{56}$$

From Eq. 23 and Eq. (56), it is easy to obtain

$$\begin{aligned} f_2(\boldsymbol{x}) &= \frac{\lambda_2^l}{\lambda_2}f_1^2(\boldsymbol{x}) + f_2(\boldsymbol{x}) - \frac{\lambda_2^l}{\lambda_2}f_2^l(\boldsymbol{x}) \\ &= \frac{1}{\lambda_2}\left[\lambda_2^l f_2^l(\boldsymbol{x}) - \left[\boldsymbol{x}^{\mathrm{T}}\ 1\right]\left(\mathbf{Q}^{\mathrm{T}}\mathbf{Q}+\delta\mathbf{I}\right)^{-1}\mathbf{P}^{\mathrm{T}}\left(\boldsymbol{\gamma}^l - \boldsymbol{\gamma}\right)\right]. \end{aligned} \tag{57}$$

For $\forall i \in \mathcal{A}$, there are three special scenarios without considering any possible events, i.e.,

- If $i \in \mathcal{L}_A^l$, then $\gamma_i = \gamma_i^l = 1$.
- If $i \in \mathcal{E}_A^l$, then $f_2(\boldsymbol{x}) = f_2^l(\boldsymbol{x}) = 1$.
- If $i \in \mathcal{R}_A^l$, then $\gamma_i = \gamma_i^l = 0$.

Therefore, Eq. (57) can be simplified

$$\begin{aligned} \mathbf{P}_E^l\left(\mathbf{Q}^{\mathrm{T}}\mathbf{Q}+\delta\mathbf{I}\right)^{-1}\left(\mathbf{P}_E^l\right)^{\mathrm{T}}\left(\boldsymbol{\gamma}_{\boldsymbol{E}}{}^l - \boldsymbol{\gamma}_E\right) &= \bar{\mathbf{B}}^l\left(\boldsymbol{\gamma}_{\boldsymbol{E}}{}^l - \boldsymbol{\gamma}_E\right) \\ &= \left(\lambda_2^l - \lambda_2\right)\mathbf{e}_{n_A^l}. \end{aligned} \tag{58}$$

If $\bar{\mathbf{B}}^l$ is invertible, the we can obtain

$$\begin{aligned} \boldsymbol{\gamma}_E &= \boldsymbol{\gamma}_E^l - \left(\lambda_2^l - \lambda_2\right)\left(\bar{\mathbf{B}}^l\right)^{-1}\mathbf{e}_{n_A^l} \\ &= \boldsymbol{\gamma}_E^l - \left(\lambda_2^l - \lambda_2\right)\boldsymbol{\vartheta}^l. \end{aligned} \tag{59}$$

To sum up, Theorem 2 is proved. □

References

1. Kong L, He W, Yang C, Sun C (2020) Robust neurooptimal control for a robot via adaptive dynamic programming. IEEE Trans Neural Netw Learn Syst 32(6):2584–2594
2. Sun C, Li X, Sun Y (2020) A parallel framework of adaptive dynamic programming algorithm with off-policy learning. IEEE Trans Neural Netw Learn Syst 32:3578–3587

3. Liu J, Ran G, Wu Y, Xue L, Sun C (2021) Dynamic event-triggered practical fixed-time consensus for nonlinear multi-agent systems. IEEE Trans Circuits Syst II Express Briefs 69:2156–2160
4. Cortes C, Vapnik V (1995) Support-vector networks. Mach Learn 20(3):273–297
5. Khemchandani R, Chandra S et al (2007) Twin support vector machines for pattern classification. IEEE Trans Pattern Anal Mach Intell 29(5):905–910
6. Liu W, Ci L, Liu L (2020) A new method of fuzzy support vector machine algorithm for intrusion detection. Appl Sci 10(3):1065
7. de Lima MD, de Oliveira Roque e Lima J, Barbosa RM (2020) Medical data set classification using a new feature selection algorithm combined with twin-bounded support vector machine. Med Biol Eng Comput 58(3):519–528
8. Cong H, Yang C, Pu X (2008) Efficient speaker recognition based on multi-class twin support vector machines and GMMs. In: IEEE conference on robotics, automation and mechatronics. IEEE, pp 348–352
9. Best MJ (1996) An algorithm for the solution of the parametric quadratic programming problem. In: Applied mathematics and parallel computing. Springer, pp 57–76
10. Hastie T, Rosset S, Tibshirani R, Zhu J (2004) The entire regularization path for the support vector machine. J Mach Learn Res 5:1391–1415
11. Pan X, Yang Z, Xu Y, Wang L (2017) Safe screening rules for accelerating twin support vector machine classification. IEEE Trans Neural Netw Learn Syst 29(5):1876–1887
12. Yang Z, Pan X, Xu Y (2018) Piecewise linear solution path for pinball twin support vector machine. Knowl Based Syst 160:311–324
13. Asheghi Dizaji Z, Asghari Aghjehdizaj S, Soleimanian Gharehchopogh F (2020) An improvement in support vector machines algorithm with imperialism competitive algorithm for text documents classification. Signal Data Process 17(1):117–130
14. Adeleke A, Samsudin N, Othman Z, Khalid SA (2019) A two-step feature selection method for quranic text classification. Indones J Electr Eng Comput Sci 16(2):730–736
15. Kumaresan T, Saravanakumar S, Balamurugan R (2019) Visual and textual features based email spam classification using s-cuckoo search and hybrid kernel support vector machine. Clust Comput 22(1):33–46
16. Fu W, Wang K, Zhang C, Tan J (2019) A hybrid approach for measuring the vibrational trend of hydroelectric unit with enhanced multi-scale chaotic series analysis and optimized least squares support vector machine. Trans Inst Meas Control 41(15):4436–4449
17. Raj JS, Ananthi JV (2019) Recurrent neural networks and nonlinear prediction in support vector machines. J Soft Comput Paradig (JSCP) 1(01):33–40
18. Zhou Y, Chang F-J, Chang L-C, Kao I-F, Wang Y-S, Kang C-C (2019) Multi-output support vector machine for regional multi-step-ahead pm2.5 forecasting. Sci Total Environ 651:230–240
19. Al-Dabagh MZN, Alhabib M, Al-Mukhtar F (2018) Face recognition system based on kernel discriminant analysis, k-nearest neighbor and support vector machine. Int J Res Eng 5(3):335–338
20. Al-Shibli A, Abusham E (2017) Face recognition using local graph structure and support vector machine (LGS-SVM). Int J Comput Appl Sci (IJOCAAS) 2(2):68–72
21. Al-Dabagh MZN, Rashid SJ, Ahmad MI (2020) Face recognition system based on wavelet transform, histograms of oriented gradients and support vector machine. Int J Comput Digital Syst 10:1–4
22. Huang H, Wei X, Zhou Y (2018) Twin support vector machines: a survey. Neurocomputing 300:34–43
23. He J, Zheng S-H (2014) Intrusion detection model with twin support vector machines. J Shanghai Jiaotong Univ (Sci) 19(4):448–454
24. Gupta D, Richhariya B, Borah P (2019) A fuzzy twin support vector machine based on information entropy for class imbalance learning. Neural Comput Appl 31(11):7153–7164
25. Singh S (2018) Forensic and automatic speaker recognition system. Int J Electr Comput Eng 8(5):2804
26. Prasetio BH, Tamura H, Tanno K (2018) Ensemble support vector machine and neural network method for speech stress recognition. In: International workshop on big data and information security (IWBIS). IEEE, pp 57–62
27. Wang S, Lu S, Dong Z, Yang J, Yang M, Zhang Y (2016) Dual-tree complex wavelet transform and twin support vector machine for pathological brain detection. Appl Sci 6(6):169
28. Rustam Z, Rampisela TV (2018) Support vector machines and twin support vector machines for classification of schizophrenia data. Int J Eng Technol 7(4):6378–6877
29. Xu Y, Pan X, Zhou Z, Yang Z, Zhang Y (2015) Structural least square twin support vector machine for classification. Appl Intell 42(3):527–536
30. Gao Q-Q, Bai Y-Q, Zhan Y-R (2019) Quadratic kernel-free least square twin support vector machine for binary classification problems. J Oper Res Soc China 7(4):539–559
31. Azad-Manjiri M, Amiri A, Sedghpour AS (2020) ML-SLSTSVM: a new structural least square twin support vector machine for multi-label learning. Pattern Anal Appl 23(1):295–308

32. Shao Y-H, Deng N-Y, Yang Z-M (2012) Least squares recursive projection twin support vector machine for classification. Pattern Recogn 45(6):2299–2307
33. Xu Y, Wang L (2012) A weighted twin support vector regression. Knowl Based Syst 33:92–101
34. Ye Y-F, Bai L, Hua X-Y, Shao Y-H, Wang Z, Deng N-Y (2016) Weighted lagrange ε-twin support vector regression. Neurocomputing 197:53–68
35. Wang L, Gao C, Zhao N, Chen X (2019) A projection wavelet weighted twin support vector regression and its primal solution. Appl Intell 49(8):3061–3081
36. Shao Y-H, Chen W-J, Wang Z, Li C-N, Deng N-Y (2015) Weighted linear loss twin support vector machine for large-scale classification. Knowl Based Syst 73:276–288
37. Yang Z-M, Wu H-J, Li C-N, Shao Y-H (2016) Least squares recursive projection twin support vector machine for multi-class classification. Int J Mach Learn Cybern 7(3):411–426
38. Chen S, Wu X, Yin H (2019) A novel projection twin support vector machine for binary classification. Soft Comput 23(2):655–668
39. Chen W-J, Shao Y-H, Li C-N, Liu M-Z, Wang Z, Deng N-Y (2020) ν-projection twin support vector machine for pattern classification. Neurocomputing 376:10–24
40. Cervantes J, Garcia-Lamont F, Rodríguez-Mazahua L, Lopez A (2020) A comprehensive survey on support vector machine classification: applications, challenges and trends. Neurocomputing 408:189–215
41. Singla M, Shukla K (2020) Robust statistics-based support vector machine and its variants: a survey. Neural Comput Appl 32(15):11173–11194
42. Kumar MA, Gopal M (2009) Least squares twin support vector machines for pattern classification. Expert Syst Appl 36(4):7535–7543
43. Chen J, Ji G (2010) Multi-class LSTSVM classifier based on optimal directed acyclic graph. In: The 2nd international conference on computer and automation engineering (ICCAE), vol 3. IEEE, pp 100–104
44. Ye Q, Zhao C, Gao S, Zheng H (2012) Weighted twin support vector machines with local information and its application. Neural Netw 35:31–39
45. Chen X, Yang J, Ye Q, Liang J (2011) Recursive projection twin support vector machine via within-class variance minimization. Pattern Recogn 44(10–11):2643–2655
46. Rosset S, Zhu J (2007) Piecewise linear regularized solution paths. Ann Stat 35:1012–1030
47. Karasuyama M, Takeuchi I (2011) Nonlinear regularization path for quadratic loss support vector machines. IEEE Trans Neural Netw 22(10):1613–1625
48. Gu B, Wang J-D, Zheng G-S, Yu Y-C (2012) Regularization path for ν-support vector classification. IEEE Trans Neural Netw Learn Syst 23(5):800–811
49. Gu B, Sheng VS (2017) A solution path algorithm for general parametric quadratic programming problem. IEEE Trans Neural Netw Learn Syst 29(9):4462–4472