




A novel method for solving universum twin bounded support vector machine in the primal space

Hossein Moosaei^{1,2,3} · Saeed Khosravi⁴ · Fatemeh Bazikar⁵ · Milan Hladík^{6,7} · Mario Rosario Guarracino^{8,9} 

Published online: 2 November 2023

© The Author(s), under exclusive licence to Springer Nature Switzerland AG 2023

Abstract

In supervised learning, the Universum, a third class that is not a part of either class in the classification task, has proven to be useful. In this study we propose (N \mathcal{U} TBSVM), a Newton-based approach for solving in the primal space the optimization problems related to Twin Bounded Support Vector Machines with Universum data (\mathcal{U} TBSVM). In the N \mathcal{U} TBSVM, the constrained programming problems of \mathcal{U} TBSVM are converted into unconstrained optimization problems, and a generalization of Newton's method for solving the unconstrained problems is introduced. Numerical experiments on synthetic, UCI, and NDC data sets show the ability and effectiveness of the proposed N \mathcal{U} TBSVM. We apply the suggested method for gender detection from face images, and compare it with other methods.

Keywords Twin bounded support vector machine · Universum · Newton's method · Unconstrained optimization problem

Mathematics Subject Classification (2000) 90C20 · 90C25 · 68T10

1 Introduction

The use of machine learning methods covers a wide range of fields, including medical diagnosis [1], computer vision [2], text categorization [3], computational biology [4], bioinformatics [5], and many others [6–8]. Support Vector Machine (SVM) is a *de facto* standard machine learning approach for binary classification problems. Vapnik et al. [9] introduced SVM in the early 1990s. The standard SVM identifies the hyperplane that divides two classes by solving a convex quadratic programming problem [10]. The separating hyperplanes is the middle of two parallel hyperplanes that leave the points of each class in different half-spaces and are as far apart as possible.

To incorporate prior information about the distribution of data into the classifier, Weston et al. [11] proposed the idea of Universum Support Vector Machine (\mathcal{U} SVM). The name derives

✉ Mario Rosario Guarracino
mario.guarracino@unicas.it

Extended author information available on the last page of the article

from Universum data, which are unlabeled observations not present in either class of the original data set, still belonging to the same domain of the original problem. They empirically demonstrated that \mathcal{U} SVM gives superior generalization performance to the traditional SVM.

Motivated by \mathcal{U} SVM, additional studies have been conducted to present an Universum-based model for other supervised methods.

This is the case for Twin Support Vector Machine (TSVM) [12], a generalization of SVM, aiming to construct two nonparallel hyperplanes, with each hyperplane being as close as possible to one class and, at the same time, as far as possible from the other class. Following \mathcal{U} SVM and TSVM, Qi et al. [13] presented the Twin Support Vector Machine using Universum data (\mathcal{U} TSVM). More recently, Richhariya et al. [14] introduced an improvement of \mathcal{U} TSVM, which we call \mathcal{U} TBSVM, to improve the performance of \mathcal{U} TSVM. They solved the optimization problems of \mathcal{U} TBSVM in the dual space. Nevertheless, from an optimization point of view, some mathematical programming problems can be solved efficiently in the primal space.

This paper provides a novel approach to solve the \mathcal{U} TBSVM in the primal space. We convert the constrained problems of \mathcal{U} TBSVM into unconstrained optimization problems. We employ a generalized Newton's approach to solve the unconstrained optimization problem because the objective function is only once-differentiable and not twice-differentiable. To show the performance superiority of the proposed method, a computational comparison of the proposed method with TSVM, and \mathcal{U} TBSVM is reported, in terms of classification accuracy and computing time for synthetic data set, several UCI data sets, NDC data sets and also gender recognition data sets. Some critical aspects of the proposed methods are outlined below:

- Solving the \mathcal{U} TBSVM method in the primal space instead of the dual space.
- Modifying the fast Newton method due to the fact that unconstrained minimization problems are only once differentiable.
- Empirically proving the viability and efficacy of the suggested strategy on different synthetic and real world data sets.

The remaining sections of the paper are laid out as follows. The basic principles of TSVM, TBSVM, and \mathcal{U} TBSVM are reviewed in Section 2. N \mathcal{U} TBSVM and its theoretical analysis are proposed in Section 3. Section 4 contains the findings of all numerical experiments, and Section 5 concludes the study.

2 Related work

In this section, we provide an overview of the TSVM, TBSVM, and \mathcal{U} TBSVM formulations.

2.1 Twin support vector machine

To identify two hyperplanes for binary classification that are as close as possible to one of the two classes and as far from the other class, Jayadeva et al. [12] introduced the Twin Support Vector Machine (TSVM). The TSVM identifies the two not necessarily parallel hyperplanes shown below:

$$w_1^T x + b_1 = 0, \text{ and } w_2^T x + b_2 = 0, \quad (1)$$

where $w_1, w_2 \in \mathbb{R}^n$ and $b_1, b_2 \in \mathbb{R}$. Assume that all of the data points that belong to class $+1$ and -1 are represented by matrix $A \in \mathbb{R}^{m_1 \times n}$ and matrix $B \in \mathbb{R}^{m_2 \times n}$, respectively. Solving the following two quadratic programming problems yields the TSVM classifiers:

$$\begin{aligned} \min_{w_1, b_1, \xi_1} \quad & \frac{1}{2} \|Aw_1 + e_1 b_1\|^2 + \frac{c_1}{2} e_2^T \xi_1 \\ \text{s.t.} \quad & -(Bw_1 + e_2 b_1) + \xi_1 \geq e_2, \\ & \xi_1 \geq 0, \end{aligned} \quad (2)$$

and

$$\begin{aligned} \min_{w_2, b_2, \xi_2} \quad & \frac{1}{2} \|Bw_2 + e_2 b_2\|^2 + \frac{c_2}{2} e_1^T \xi_2 \\ \text{s.t.} \quad & (Aw_2 + e_1 b_2) + \xi_2 \geq e_1, \\ & \xi_2 \geq 0, \end{aligned} \quad (3)$$

where $c_1, c_2 > 0$ are penalty parameters, ξ_1, ξ_2 are slack vectors, and e_1, e_2 are vectors of ones of appropriate dimension. Using the KKT conditions, we obtain the Wolfe dual formulations of (2) and (3):

$$\begin{aligned} \max_{\alpha} \quad & -\frac{1}{2} \alpha^T G (H^T H)^{-1} G^T \alpha + e_2^T \alpha \\ \text{s.t.} \quad & 0 \leq \alpha \leq c_1 e_2, \end{aligned} \quad (4)$$

and

$$\begin{aligned} \max_{\alpha^*} \quad & -\frac{1}{2} \alpha^{*T} H (G^T G)^{-1} H^T \alpha^* + e_1^T \alpha^* \\ \text{s.t.} \quad & 0 \leq \alpha^* \leq c_2 e_1, \end{aligned} \quad (5)$$

where $H = [A \ e_1]$, $G = [B \ e_2]$ and α, α^* are the Lagrangian multipliers. Then, solving (4) and (5), the separating hyperplanes may be found by

$$\begin{bmatrix} w_1 \\ b_1 \end{bmatrix} = -(H^T H)^{-1} G^T \alpha, \quad (6)$$

and

$$\begin{bmatrix} w_2 \\ b_2 \end{bmatrix} = (G^T G)^{-1} H^T \alpha^*. \quad (7)$$

The following decision rule is used to assign a new data point $x \in \mathbb{R}^n$ to class $i \in \{+1, -1\}$:

$$\text{class } i = \arg \min \frac{|w_i^T x + b_i|}{\|w_i\|^2}, \quad i = 1, 2. \quad (8)$$

2.2 Twin bounded support vector machine

Twin Bounded Support Vector Machine (TBSVM) was proposed in [15]. Similarly to TSVM, TBSVM finds two not necessarily parallel hyperplanes $w_1^T x + b_1 = 0$ and $w_2^T x + b_2 = 0$.

TBSVM has an additional regularization term compared to TSVM, which results in the following optimization problems:

$$\begin{aligned} \min_{w_1, b_1, \xi_1} \quad & \frac{1}{2} \|Aw_1 + e_1 b_1\|^2 + \frac{c_1}{2} e_2^T \xi_1 + \frac{c_2}{2} (\|w_1\|^2 + b_1^2) \\ \text{s.t.} \quad & -(Bw_1 + e_2 b_1) + \xi_1 \geq e_2, \\ & \xi_1 \geq 0, \end{aligned} \quad (9)$$

and

$$\begin{aligned} \min_{w_2, b_2, \xi_2} \quad & \frac{1}{2} \|Bw_2 + e_2 b_2\|^2 + \frac{c_3}{2} e_1^T \xi_2 + \frac{c_4}{2} (\|w_2\|^2 + b_2^2) \\ \text{s.t.} \quad & (Aw_2 + e_1 b_2) + \xi_2 \geq e_1, \\ & \xi_2 \geq 0, \end{aligned} \quad (10)$$

where c_i , $i = 1, \dots, 4$ are positive penalty parameters, and ξ_i are slack vectors. Using the KKT necessary and sufficient conditions on the Lagrangian function of problems (9) and (10), the Wolfe dual problems are:

$$\begin{aligned} \max_{\alpha} \quad & -\frac{1}{2} \alpha^T G (H^T H + c_2 I)^{-1} G^T \alpha + e_2^T \alpha \\ \text{s.t.} \quad & 0 \leq \alpha \leq c_1 e_2, \end{aligned} \quad (11)$$

and

$$\begin{aligned} \max_{\alpha^*} \quad & -\frac{1}{2} \alpha^{*T} H (G^T G + c_4 I)^{-1} H^T \alpha^* + e_1^T \alpha^* \\ \text{s.t.} \quad & 0 \leq \alpha^* \leq c_3 e_1, \end{aligned} \quad (12)$$

where I is a dimension-appropriate identity matrix, and α, α^* are the Lagrangian multipliers. The nonparallel hyperplanes $w_1^T x + b_1 = 0$, and $w_2^T x + b_2 = 0$, may be derived by utilizing the parameters w_1, w_2, b_1 and b_2 in the following equations:

$$\begin{bmatrix} w_1 \\ b_1 \end{bmatrix} = -(H^T H + c_2 I)^{-1} G^T \alpha, \quad (13)$$

and

$$\begin{bmatrix} w_2 \\ b_2 \end{bmatrix} = (G^T G + c_4 I)^{-1} H^T \alpha^*. \quad (14)$$

Similar to the decision function of TSVM, a new data point $x \in \mathbb{R}^n$ is allocated to class $i \in \{+1, -1\}$ by applying the decision rule (8).

2.3 Twin bounded support vector machine with Universum data

The Twin Bounded Support Vector Machine with Universum (\mathcal{U} TBSVM) was proposed in [14] to enhance the classification performance of the TBSVM. The \mathcal{U} TBSVM was constructed by using Universum data in the TBSVM model. We now suppose that there are two subgroups in the training data \tilde{T} :

$$\tilde{T} = T \cup \mathcal{U},$$

where

$$T = \{(x_1, y_1), \dots, (x_n, y_n)\} \in (\mathbb{R}^m \times \{\pm 1\})^n,$$

$$\mathfrak{U} = \{x_1^*, \dots, x_u^*\}.$$

Here, the Universum class is denoted by $U \in \mathbb{R}^{u \times m}$, and each row of the matrix U stands for a Universum instance. The \mathfrak{U} TBSVM can be formulated as the following QPPs:

$$\begin{aligned} \min_{w_1, b_1, \xi_1, \psi_1} \quad & \frac{1}{2} \|Aw_1 + e_1 b_1\|^2 + \frac{c_1}{2} e_2^T \xi_1 + \frac{c_2}{2} (\|w_1\|^2 + b_1^2) + \frac{c_3}{2} e_u^T \psi_1 \\ \text{s.t.} \quad & -(Bw_1 + e_2 b_1) + \xi_1 \geq e_2, \\ & (Uw_1 + e_u b_1) + \psi_1 \geq (-1 + \varepsilon)e_u, \\ & \xi_1, \psi_1 \geq 0, \end{aligned} \quad (15)$$

and

$$\begin{aligned} \min_{w_2, b_2, \xi_2, \psi_2} \quad & \frac{1}{2} \|Bw_2 + e_2 b_2\|^2 + \frac{c_4}{2} e_1^T \xi_2 + \frac{c_5}{2} (\|w_2\|^2 + b_2^2) + \frac{c_6}{2} e_u^T \psi_2 \\ \text{s.t.} \quad & (Aw_2 + e_1 b_2) + \xi_2 \geq e_1, \\ & -(Uw_2 + e_u b_2) + \psi_2 \geq (-1 + \varepsilon)e_u, \\ & \xi_2, \psi_2 \geq 0, \end{aligned} \quad (16)$$

where c_i , $i = 1, \dots, 6$ are positive penalty parameters, $\varepsilon \in (0, 1)$ is the tolerance value for Universum class, ξ_1 , ξ_2 , ψ_1 and ψ_2 are measures of the violation of constraints associated, and e_u is a vector of ones of appropriate dimension. We may derive their dual problems by applying Lagrangian functions, as shown below.

$$\begin{aligned} \max_{\alpha, \beta} \quad & -\frac{1}{2} (G^T \alpha - O^T \beta)^T (H^T H)^{-1} (G^T \alpha - O^T \beta) + e_2^T \alpha + (\varepsilon - 1) e_u^T \beta \\ \text{s.t.} \quad & 0 \leq \alpha \leq c_1 e_2, \\ & 0 \leq \beta \leq c_3 e_u, \end{aligned} \quad (17)$$

and

$$\begin{aligned} \max_{\alpha^*, \beta^*} \quad & -\frac{1}{2} (H^T \alpha^* - O^T \beta^*)^T (G^T G)^{-1} (H^T \alpha^* - O^T \beta^*) + e_1^T \alpha^* + (\varepsilon - 1) e_u^T \beta^* \\ \text{s.t.} \quad & 0 \leq \alpha^* \leq c_4 e_1, \\ & 0 \leq \beta^* \leq c_6 e_u, \end{aligned} \quad (18)$$

where $H = [A \ e_1]$, $G = [B \ e_2]$, and $O = [U \ e_u]$. Once the QPPs (17) and (18) are solved, we can obtain the following parameters:

$$\begin{bmatrix} w_1 \\ b_1 \end{bmatrix} = -(H^T H)^{-1} (G^T \alpha - O^T \beta), \quad (19)$$

and

$$\begin{bmatrix} w_2 \\ b_2 \end{bmatrix} = (G^T G)^{-1} (H^T \alpha^* - O^T \beta^*). \quad (20)$$

Once vectors $(w_1^T, b_1)^T$ and $(w_2^T, b_2)^T$ are obtained from (19) and (20), the separating hyperplanes

$$w_1^T x + b_1 = 0, \text{ and } w_2^T x + b_2 = 0, \quad (21)$$

are known. A new data point $x \in \mathbb{R}^n$ is allocated to class $i \in \{+1, -1\}$ by a similar rule to the TSVM.

3 Newton's method for TBSVM with Universum data

In this section, we introduce a new method for solving the problems (15) and (16) in the primal space.

$$\begin{aligned}
 \min_{w_1, b_1, \xi_1, \psi_1} \quad & \frac{1}{2} \|Aw_1 + e_1 b_1\|^2 + \frac{c_1}{2} \|\xi_1\|^2 + \frac{c_2}{2} (\|w_1\|^2 + b_1^2) + \frac{c_3}{2} \|\psi_1\|^2 \\
 \text{s.t.} \quad & -(Bw_1 + e_2 b_1) + \xi_1 \geq e_2, \\
 & (Uw_1 + e_u b_1) + \psi_1 \geq (-1 + \varepsilon)e_u, \\
 & \xi_1, \psi_1 \geq 0,
 \end{aligned} \tag{22}$$

and

$$\begin{aligned}
 \min_{w_2, b_2, \xi_2, \psi_2} \quad & \frac{1}{2} \|Bw_2 + e_2 b_2\|^2 + \frac{c_4}{2} \|\xi_2\|^2 + \frac{c_5}{2} (\|w_2\|^2 + b_2^2) + \frac{c_6}{2} \|\psi_2\|^2 \\
 \text{s.t.} \quad & (Aw_2 + e_1 b_2) + \xi_2 \geq e_1, \\
 & -(Uw_2 + e_u b_2) + \psi_2 \geq (-1 + \varepsilon)e_u, \\
 & \xi_2, \psi_2 \geq 0.
 \end{aligned} \tag{23}$$

For the optimal solution of problems (22) and (23), we have

$$\xi_1 = (e_2 + (Bw_1 + e_2 b_1))_+, \tag{24}$$

$$\psi_1 = ((-1 + \varepsilon)e_u - (Uw_1 + e_u b_1))_+, \tag{25}$$

$$\xi_2 = (e_1 - (Aw_2 + e_1 b_2))_+, \tag{26}$$

$$\psi_2 = ((-1 + \varepsilon)e_u + (Uw_2 + e_u b_2))_+, \tag{27}$$

where, $(\cdot)_+$ replaces negative components of a vector by zeros. Thus, we can replace ξ_1, ξ_2, ψ_1 and ψ_2 in (22) and (23) by (24)–(27) and convert the \mathcal{U} TBSVM problems (22) and (23) into an equivalent \mathcal{U} TBSVM which are an unconstrained optimization problems as follows:

$$\begin{aligned}
 \min_{w_1, b_1} \quad \varphi_1(w_1, b_1) = & \frac{1}{2} \|Aw_1 + e_1 b_1\|^2 + \frac{c_1}{2} \|(e_2 + (Bw_1 + e_2 b_1))_+\|^2 + \frac{c_2}{2} (\|w_1\|^2 + b_1^2) \\
 & + \frac{c_3}{2} \|((-1 + \varepsilon)e_u - (Uw_1 + e_u b_1))_+\|^2
 \end{aligned} \tag{28}$$

and

$$\begin{aligned}
 \min_{w_2, b_2} \quad \varphi_2(w_2, b_2) = & \frac{1}{2} \|Bw_2 + e_2 b_2\|^2 + \frac{c_4}{2} \|(e_1 - (Aw_2 + e_1 b_2))_+\|^2 + \frac{c_5}{2} (\|w_2\|^2 + b_2^2) \\
 & + \frac{c_6}{2} \|((-1 + \varepsilon)e_u + (Uw_2 + e_u b_2))_+\|^2
 \end{aligned} \tag{29}$$

The functions $\varphi_1(w_1, b_1)$ and $\varphi_2(w_2, b_2)$ are differentiable and their gradients can be computed as follows:

$$\begin{aligned}
 \nabla \varphi_1(w_1, b_1) = & \begin{bmatrix} \frac{\partial \varphi_1}{\partial w_1} \\ \frac{\partial \varphi_1}{\partial b_1} \end{bmatrix} = \begin{bmatrix} A^T (Aw_1 + e_1 b_1) \\ e_1^T (Aw_1 + e_1 b_1) \end{bmatrix} + c_1 \begin{bmatrix} B^T (e_2 + (Bw_1 + e_2 b_1))_+ \\ e_2^T (e_2 + (Bw_1 + e_2 b_1))_+ \end{bmatrix} \\
 & + c_2 \begin{bmatrix} w_1 \\ b_1 \end{bmatrix} + c_3 \begin{bmatrix} -U^T ((-1 + \varepsilon)e_u - (Uw_1 + e_u b_1))_+ \\ -e_u^T ((-1 + \varepsilon)e_u - (Uw_1 + e_u b_1))_+ \end{bmatrix},
 \end{aligned}$$

and

$$\begin{aligned}\nabla\varphi_2(w_2, b_2) &= \begin{bmatrix} \frac{\partial\varphi_2}{\partial w_2} \\ \frac{\partial\varphi_2}{\partial b_2} \end{bmatrix} = \begin{bmatrix} B^T(Bw_2 + e_2b_2) \\ e_2^T(Bw_2 + e_2b_2) \end{bmatrix} + c_4 \begin{bmatrix} -A^T(e_1 - (Aw_2 + e_1b_2))_+ \\ e_1^T(e_1 - (Aw_2 + e_1b_2))_+ \end{bmatrix} \\ &\quad + c_5 \begin{bmatrix} w_2 \\ b_2 \end{bmatrix} + c_6 \begin{bmatrix} U^T((-1 + \varepsilon)e_u + (Uw_2 + e_ub_2))_+ \\ e_u^T((-1 + \varepsilon)e_u + (Uw_2 + e_ub_2))_+ \end{bmatrix}.\end{aligned}$$

Functions $\nabla\varphi_1(w_1, b_1)$ and $\nabla\varphi_2(w_2, b_2)$ are not differentiable, since they contain the terms

$$\begin{aligned}(e_2 + (Bw_2 + e_2b_1))_+, \quad &((-1 + \varepsilon)e_u - (Uw_1 + e_ub_1))_+, \\ (e_1 - (Aw_2 + e_1b_2))_+, \quad &((-1 + \varepsilon)e_u + (Uw_2 + e_ub_2))_+.\end{aligned}$$

Therefore, the objective functions of the problems (28) and (29) are not twice differentiable and we can not apply the standard Newton's method for solving these problems. Motivated by [8, 16, 17], we want to define the generalized Hessian for $\varphi_1(w_1, b_1)$ and $\varphi_2(w_2, b_2)$.

The generalized Hessian of f at x is the set $\partial^2 f(x)$ of $n \times n$ matrices is defined by

$$\partial^2 f(x) = \text{conv} \{ H \in \mathbb{R}^{n \times n} : \exists x_k \rightarrow x \text{ with } \nabla f \text{ differentiable at } x_k \text{ and } \partial^2 f(x) \rightarrow H \},$$

where $\text{conv}(\cdot)$ denotes the convex hull. Function $\frac{\partial\varphi_1}{\partial w_1}$ can be expressed

$$\frac{\partial\varphi_1}{\partial w_1} = T_1 z_1 + c_1 B^T(e_2 + T_2 z_1)_+ + c_2 T_3 z_1 - c_3 U^T((-1 + \varepsilon)e_u - T_4 z_1)_+, \quad (30)$$

where $T_1 = [A^T A \ A^T e_1]$, $T_2 = [B \ e_2]$, $T_3 = [I_{n \times n} \ 0_{n \times 1}]$, $T_4 = [U \ e_u]$ and $z = (w_1^T, b_1)^T$. Note that (30) is not differentiable, but it satisfies the Lipschitz conditions.

Theorem 1 Function $\frac{\partial\varphi_1}{\partial w_1}$ is globally Lipschitz.

Proof From (30) we have that

$$\begin{aligned}\left\| \frac{\partial\varphi_1}{\partial w_1}(s_1) - \frac{\partial\varphi_1}{\partial w_1}(s_2) \right\| &= \| T_1 s_1 + c_1 B^T(e_2 + T_2 s_1)_+ + c_2 T_3 s_1 \\ &\quad - c_3 U^T((-1 + \varepsilon)e_u - T_4 s_1)_+ - T_1 s_2 - c_1 B^T(e_2 + T_2 s_2)_+ \\ &\quad - c_2 T_3 s_2 + c_3 U^T((-1 + \varepsilon)e_u - T_4 s_2)_+ \|,\end{aligned}$$

we bound it from above by

$$\begin{aligned}&\| T_1(s_1 - s_2) \| + \| c_2 T_3(s_1 - s_2) \| + \| c_1 \| B^T \| (e_2 + T_2 s_1) - (e_2 + T_2 s_2) \| \\ &\quad + \| c_3 \| U^T \| ((-1 + \varepsilon)e_u - T_4 s_1) - ((-1 + \varepsilon)e_u - T_4 s_2) \| \\ &\leq \left(\| T_1 \| + c_2 \| T_3 \| + c_1 \| \| B \|^T \cdot \| T_2 \| + c_3 \| \| U \|^T \cdot \| T_4 \| \right) \| s_1 - s_2 \|.\end{aligned}$$

Thus we infer that $\frac{\partial\varphi_1}{\partial w_1}$ is globally Lipschitz with constant

$$K = \| T_1 \| + c_2 \| T_3 \| + c_1 \| \| B \|^T \cdot \| T_2 \| + c_3 \| \| U \|^T \cdot \| T_4 \|.$$

□

As a result, we can prove the following theorem:

Theorem 2 $\nabla\varphi_1(z_1)$ is globally Lipschitz continuous and the generalized Hessian of $\varphi_1(z_1)$ is $\partial^2\varphi_1(z_1) = T_1 + c_1 B^T D_1 T_2 + c_2 T_3 + c_3 U^T D_2 T_4$, where D_1 and D_2 stand for the diagonal matrices

$$D_1 = \text{diag}(\text{sgn}(e_2 + (Bw_1 + e_2 b_1))_+),$$

$$D_2 = \text{diag}(\text{sgn}((-1 + \varepsilon)e_u - (Uw_1 + e_u b_1))_+),$$

and $\text{sgn}(\cdot)$ is the standard sign function.

Remark 1 We can obtain similar results for $\frac{\partial\varphi_1}{\partial b_1}$, $\frac{\partial\varphi_2}{\partial w_2}$, and $\frac{\partial\varphi_2}{\partial b_2}$.

We know that $\nabla\varphi_1$ and $\nabla\varphi_2$ are differentiable almost everywhere and that the generalized Hessian of φ_1 and φ_2 exists everywhere based on the prior discussion and the aforementioned theorem. Therefore, to solve the unconstrained problems (28) and (29), we can use the generalized Newton's method [8, 16, 17]. Algorithm 1 describes our methodology.

Algorithm 1 Linear NLTBSVM.

Input: The training set $\tilde{T} = T \cup \mathcal{U}$ consisting of $T = \{(x_1, y_1), \dots, (x_n, y_n)\} \in (\mathbb{R}^m \times \{\pm 1\})^n$ and $\mathcal{U} = \{x_1^*, \dots, x_u^*\} \in \mathbb{R}^{u \times m}$.

- 1: Select penalty parameters c_i , for $i = 1, \dots, 6$ and parameter $\varepsilon = (0, 1)$.
 - 2: Determine parameters (w_1, b_1) and (w_2, b_2) of the hyperplanes by solving problems (28) and (29) by using the generalized Newton's method.
 - 3: By applying the decision function (8), assign a data point to either class +1 or -1.
-

For the nonlinear case, the LTBSVM considers the following two nonparallel kernel-generated hyperplanes:

$$K(x, C^T)w_1 + b_1 = 0 \text{ and } K(x, C^T)w_2 + b_2 = 0, \quad (31)$$

where $C^T = [A^T \ B^T]^T$ and $K(\cdot, \cdot)$ is a chosen appropriate kernel function. The resulting nonlinear optimization problems read as

$$\begin{aligned} \min_{w_1, b_1, \xi_1, \psi_1} \quad & \frac{1}{2} \|K(A, C^T)w_1 + e_1 b_1\|^2 + \frac{c_1}{2} \|\xi_1\|^2 + \frac{c_2}{2} (\|w_1\|^2 + b_1^2) + \frac{c_3}{2} \|\psi_1\|^2 \\ \text{s.t.} \quad & - \left(K(B, C^T)w_1 + e_2 b_1 \right) + \xi_1 \geq e_2, \\ & \left(K(U, C^T)w_1 + e_u b_1 \right) + \psi_1 \geq (-1 + \varepsilon)e_u, \\ & \xi_1, \psi_1 \geq 0, \end{aligned} \quad (32)$$

and

$$\begin{aligned} \min_{w_2, b_2, \xi_2, \psi_2} \quad & \frac{1}{2} \|K(B, C^T)w_2 + e_2 b_2\|^2 + \frac{c_4}{2} \|\xi_2\|^2 + \frac{c_5}{2} (\|w_2\|^2 + b_2^2) + \frac{c_6}{2} \|\psi_2\|^2 \\ \text{s.t.} \quad & \left(K(A, C^T)w_2 + e_1 b_2 \right) + \xi_2 \geq e_1, \\ & - \left(K(U, C^T)w_2 + e_u b_2 \right) + \psi_2 \geq (-1 + \varepsilon)e_u, \\ & \xi_2, \psi_2 \geq 0. \end{aligned} \quad (33)$$

As in the linear case of \mathcal{UTBSVM} , the unconstrained of problems (32) and (33) can be expressed as follows,

$$\min_{w_1, b_1} \frac{1}{2} \|K(A, C^T)w_1 + e_1 b_1\|^2 + \frac{c_1}{2} \| (e_2 + (K(B, C^T)w_1 + e_2 b_1))_+ \|^2 + \frac{c_2}{2} (\|w_1\|^2 + b_1^2) + \frac{c_3}{2} \| (-1 + \varepsilon)e_u - K(U, C^T)w_1 + e_u b_1 \|_+^2, \quad (34)$$

and

$$\min_{w_2, b_2} \frac{1}{2} \|K(B, C^T)w_2 + e_2 b_2\|^2 + \frac{c_4}{2} \| (e_1 - (K(A, C^T)w_2 + e_1 b_2))_+ \|^2 + \frac{c_5}{2} (\|w_2\|^2 + b_2^2) + \frac{c_6}{2} \| (-1 + \varepsilon)e_u + (K(U, C^T)w_2 + e_u b_2) \|_+^2. \quad (35)$$

Finally, by using the generalized Newton's method, we solve the above optimization problems and then obtain two nonparallel classifiers. Algorithm 2 summarizes the nonlinear case.

4 Experimental results

In this section, we compare TSVM, \mathcal{UTBSVM} and the proposed $\mathcal{NUTBSVM}$ algorithms on different data sets. The performance of these methods are assessed using five-fold cross-validation. We discussed two different performance parameters, i.e., accuracy and learning speed. All experiments were carried out in Python 3.6 on a notebook with a Core(TM) i5 CPU @ 1.6 GHz and 4GB of RAM operating system. We used the CVXOPT package in

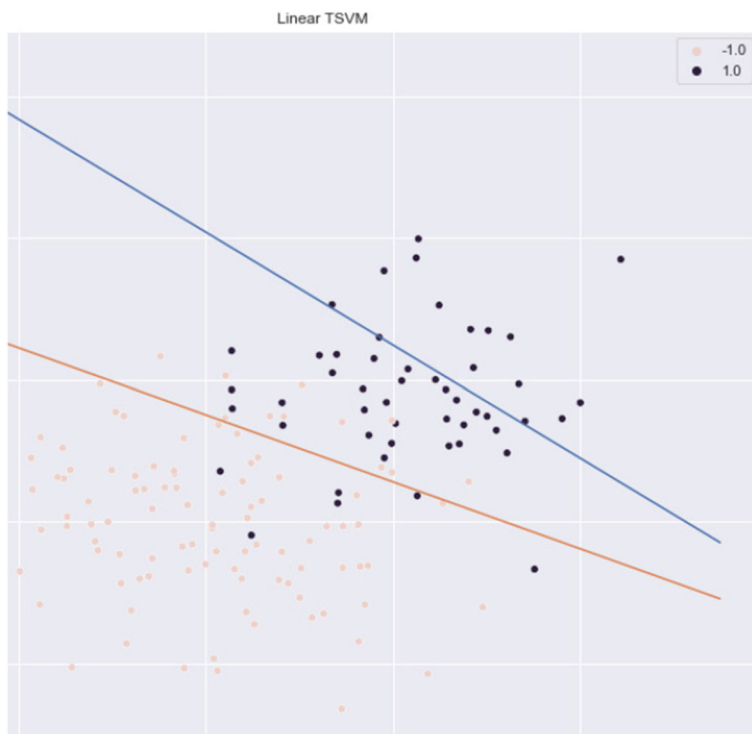


Fig. 1 The linear TSVM results on the generated data set

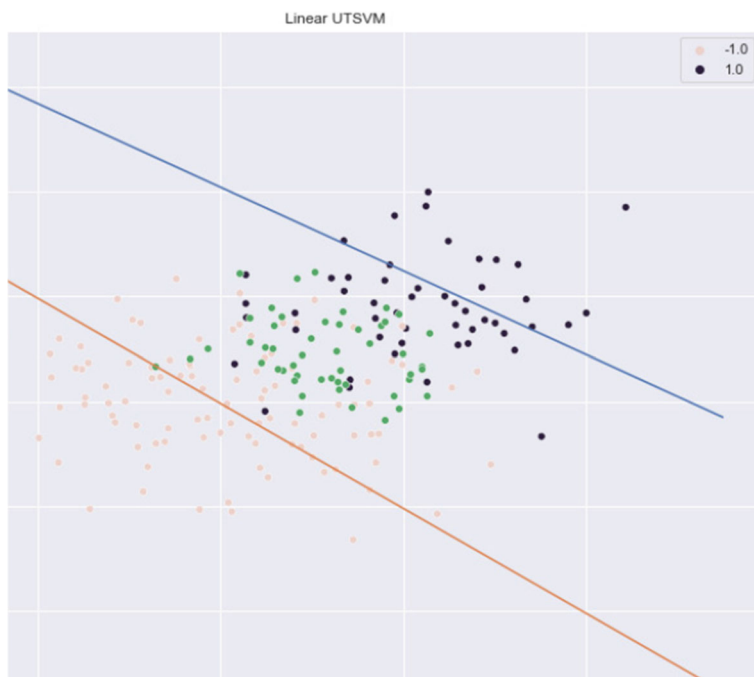


Fig. 2 The linear \mathcal{U} TBSVM results on the generated data set

Algorithm 2 Nonlinear $\mathcal{N}\mathcal{U}$ TBSVM.

Input: The training set $\tilde{T} = T \cup \mathcal{U}$ consisting of $T = \{(x_1, y_1), \dots, (x_n, y_n)\} \in (\mathbb{R}^m \times \{\pm 1\})^n$ and $\mathcal{U} = \{x_1^*, \dots, x_u^*\} \in \mathbb{R}^{u \times m}$.

- 1: Choose a Gaussian kernel function K .
 - 2: Select penalty parameters c_i , for $i = 1, \dots, 6$ and parameter $\varepsilon \in (0, 1)$ and the parameter γ of the Gaussian kernel.
 - 3: Determine parameters (w_1, b_1) and (w_2, b_2) of the hyperplanes by solving problems (34) and (35) by using the generalized Newton's method.
 - 4: By applying the decision function, assign a data point to either class $+1$ or -1 .
-

Python to solve the dual optimization problems of TSVM, \mathcal{U} TBSVM, and \mathcal{U} SVM. Notice that CVXOPT is a package developed specifically for convex optimization [18]. In addition, we utilized the Numpy library [19] for all necessary matrix operations. To build the Universum data, we randomly selected an equal amount of training data from each class. Then, pairs of data points from the two classes are averaged to obtain the Universum data [13].

Also, in all tables, the highest accuracy and lowest time are shown in bold.

4.1 Synthetic data sets

We test $\mathcal{N}\mathcal{U}$ TBSVM and the other two algorithms on synthetic data sets in \mathbb{R}^2 to visually demonstrate their efficacy. We randomly produced 50 points of class -1 and 100 points

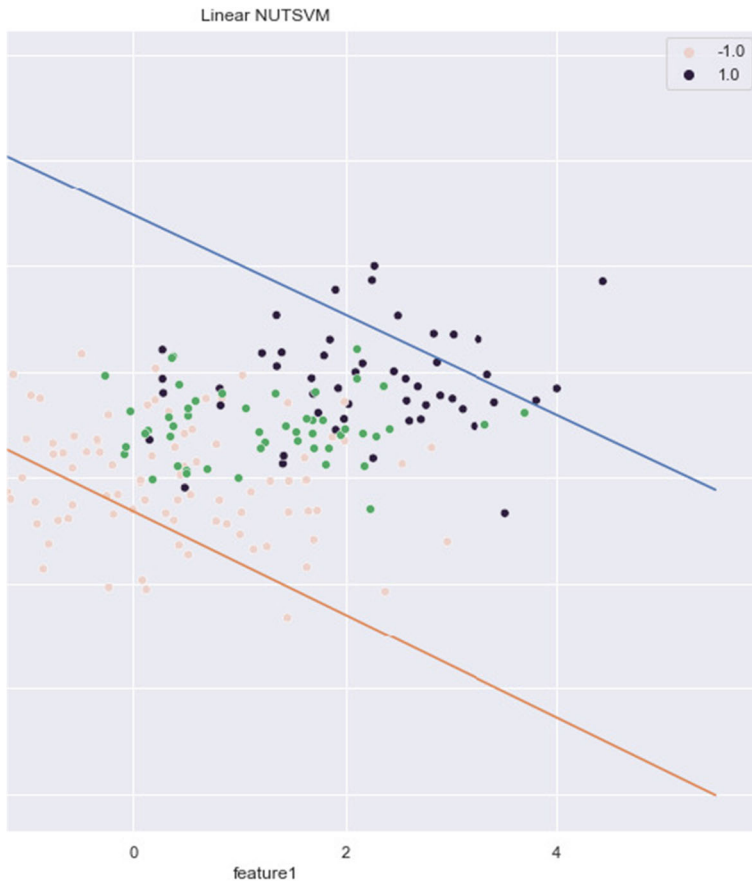


Fig. 3 The linear NUTSVM results on the generated data set

of class +1 for the simulations. By randomly matching samples from the two classes, fifty Universum data points are created from training data and depicted in green. Figures 1, 2, 3, and Table 1 show the results of NUTSVM, \mathcal{U} TBSVM and TSVM algorithms on the generated data sets for linear state. The results of the simulation demonstrate that algorithms are accurate and efficient, and they support the usefulness of the strategy that we have suggested.

4.2 Parameters selection

Obtaining the appropriate parameter values is a crucial preparatory step in performing experiments. Figures 4, 5, 6, 7, 8 and 9 illustrate the sensitivity of classification accuracy to changes

Table 1 Performance comparison of linear TSVM, \mathcal{U} TBSVM and NUTSVM on generated data sets

	TSVM	\mathcal{U} TBSVM	NUTSVM
Acc (%) \pm Std	77.92 \pm 4.98	70.19 \pm 5.26	95.57 \pm 2.74
Time (ms)	24	28	4

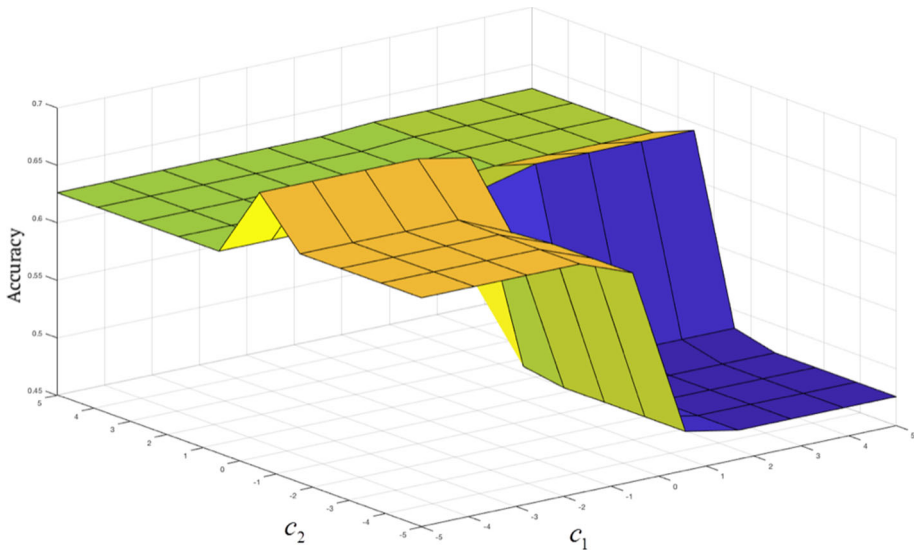


Fig. 4 The effects of parameter settings on the accuracy of the Bupa data sets in the linear NLTBSVM

in the selected parameters c_1 , c_2 , ε and γ for the proposed method on three data sets (Bupa, Ionosphere and Pima) in both linear and nonlinear cases. Therefore, selecting the appropriate parameter can play a significant role in increasing the accuracy of the classification method, and tuning the parameters to make sense. A grid search [20] has been used to tune the parameters c_i ($i = 1, \dots, 6$), ε , and kernel parameter γ . For this work, we have selected values of c_1 to c_6 from $\{2^{-5}, \dots, 2^5\}$ and value of γ from $\{2^{-7}, \dots, 2^7\}$. The parameter ε is tuned in the range $\{0.1, 0.3, 0.5, 0.7, 0.9\}$.

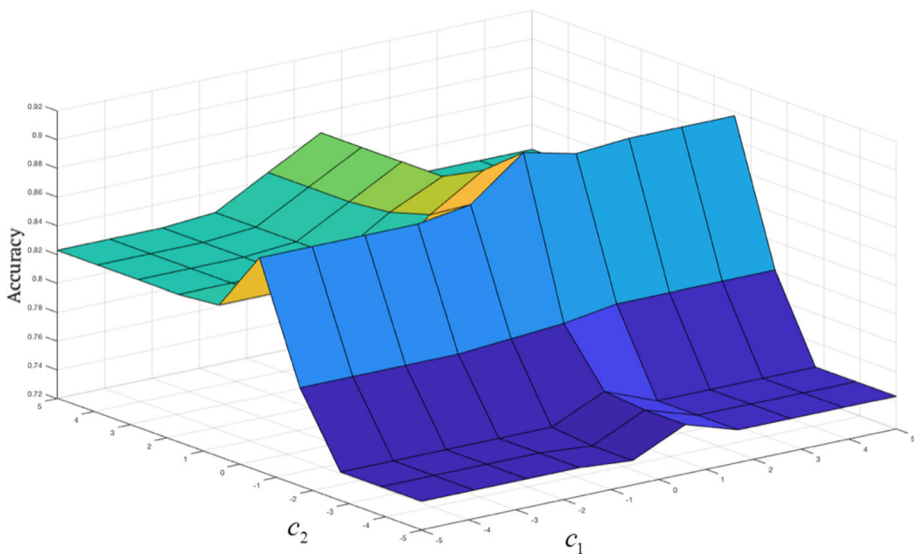


Fig. 5 The effects of parameter settings on the accuracy of Ionosphere data sets in the linear NLTBSVM

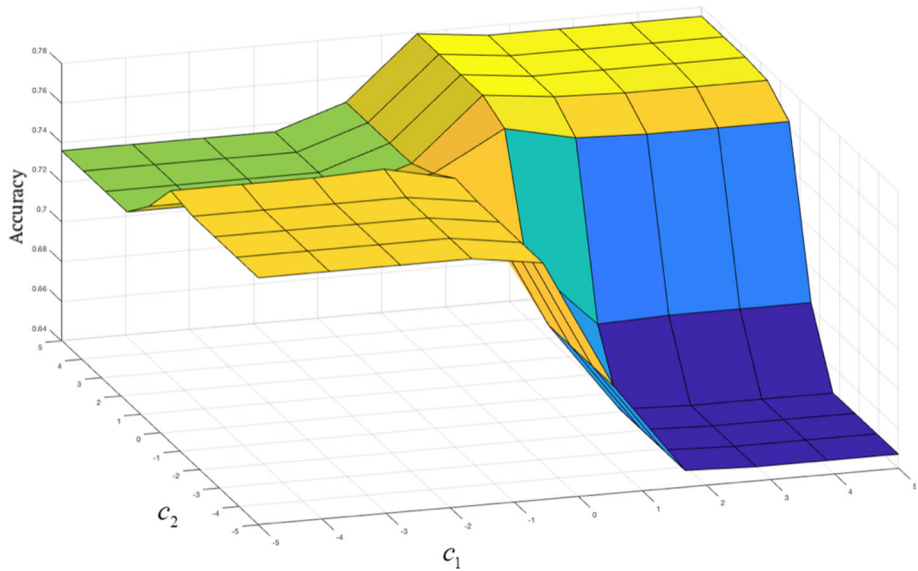


Fig. 6 The effects of parameter settings on the accuracy of Pima data sets in the linear NUTBSVM

4.3 UCI data sets

We select ten benchmark data sets from the UCI repository to test our algorithms. We utilize five-fold cross-validation to assess the effectiveness of the three methods. The experimental results on ten UCI data sets are summarized in Tables 2 and 3 for linear and nonlinear states, respectively. We compare the performance of the proposed method with TSVM and NUTBSVM on these data set.

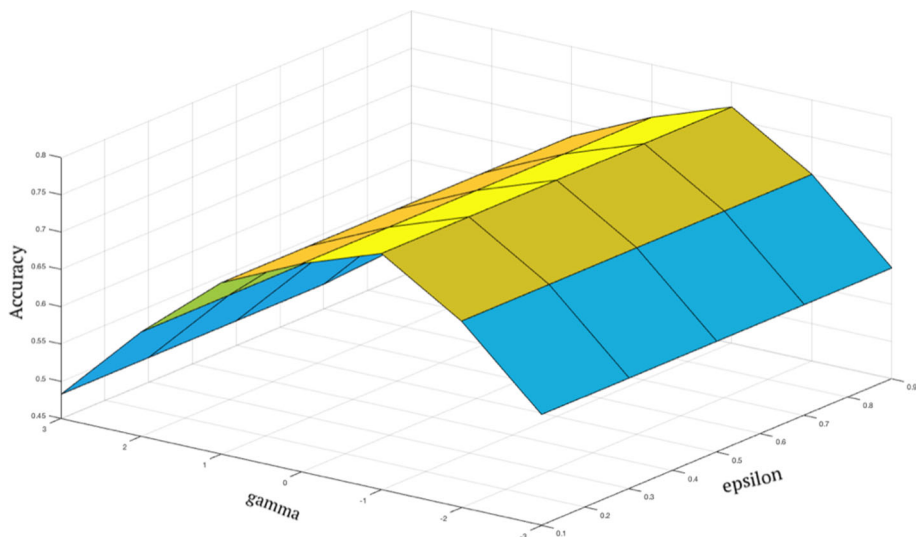


Fig. 7 The effects of parameter settings on the accuracy of Bupa data sets in the nonlinear NUTBSVM

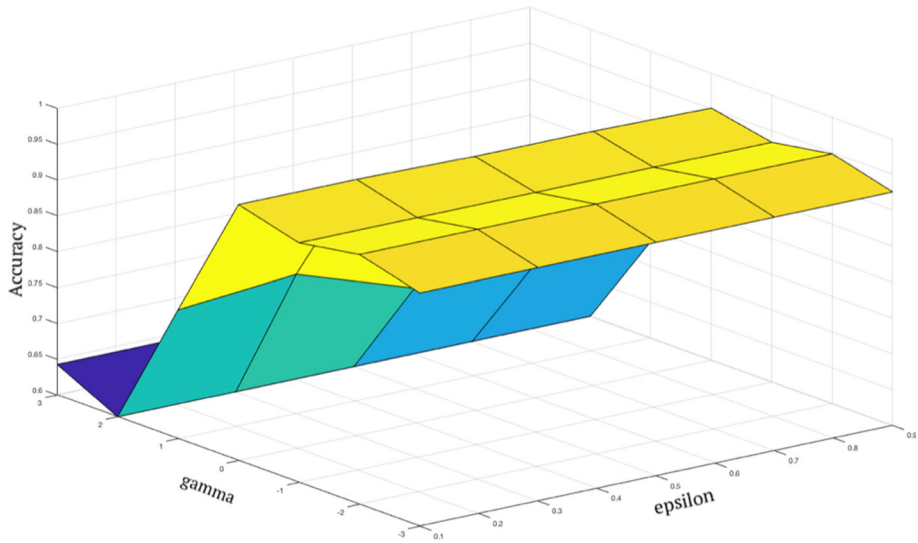


Fig. 8 The effects of parameter settings on the accuracy of Ionosphere data sets in the nonlinear NUTBSVM

In terms of running time, it should be seen that TSVM has less time to separate data than \mathcal{UTBSVM} and $N\mathcal{UTBSVM}$. The explanation is that TSVM does not face Universum data sets and, unlike the other methods, training samples' size is not increased. However, the suggested technique has the shortest running time compared to the other methods, despite using Universum samples in the training procedure and expanding the number of training samples. This is because in the proposed method, we used the fast method (the generalized Newton) to solve the unconstrained primal problems. This makes the performance of the

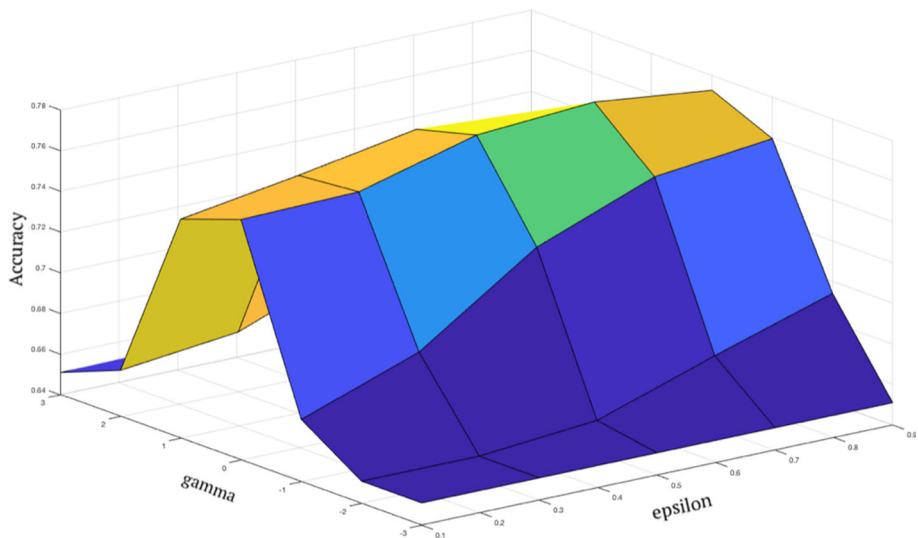


Fig. 9 The effects of parameter settings on the accuracy of Pima data sets in the nonlinear NUTBSVM

Table 2 Performance comparison of linear TSVM, \mathcal{U} TBSVM, and $\mathcal{N}\mathcal{U}$ TBSVM on UCI data sets

Data set Size	TSVM Acc (%) \pm Std Time (ms)	\mathcal{U} TBSVM Acc (%) \pm Std Time (ms)	$\mathcal{N}\mathcal{U}$ TBSVM Acc (%) \pm Std Time (ms)
Bupa	68.12 \pm 5.42	69.57 \pm 5.10	68.99 \pm 5.07
345 \times 6	28	40	7
Haberman	74.67 \pm 6.09	75.67 \pm 3.43	76.00 \pm 3.59
306 \times 3	39	65	3
Heart	72.08 \pm 2.50	72.45 \pm 4.56	73.58 \pm 4.77
270 \times 16	13	36	3
Ionosphere	86 \pm 1.89	87.43 \pm 2.46	90.29 \pm 2.78
351 \times 34	22	99	9
Pima	77.12 \pm 4.69	77.78 \pm 3.38	77.91 \pm 3.24
768 \times 9	340	882	11
Sonar	76.10 \pm 1.82	76.10 \pm 4.20	79.02 \pm 6.83
208 \times 60	17	25	6
Spect	78.50 \pm 8.40	79.24 \pm 8.69	80.00 \pm 7.97
237 \times 22	18	69	4
Trans	76.24 \pm 4.58	76.24 \pm 4.58	76.37 \pm 4.44
10 \times 32	153	324	9
House Votes	96.05 \pm 2.90	96.51 \pm 2.43	96.04 \pm 2.81
435 \times 16	79	104	7
Wpbc	96.28 \pm 1.71	97 \pm 1.64	97.17 \pm 1.31
198 \times 33	139	778	13

proposed method better than the other two methods, in spite of the addition of Universum samples in the training process.

In terms of accuracy, as shown in the Tables 2 and 3, our proposed method achieves the highest classification accuracy for most of the UCI data sets through linear and nonlinear states. Therefore, we conclude that our proposed approach has the highest classification accuracy and the lowest learning time among the three methods.

4.4 NDC data sets

David Musicant's NDC Data Generator [21] is used to create NDC data. Moosaei et al. [22] presented an enhanced version of this generator that may include an arbitrary number of samples, futures, and classes. In this subsection, the feature dimension of NDC data is chosen to be ten and the size of samples are 100, 500, 1000, 2000, and 5000. Table 4 describes the results obtained from the NDC data sets. It is fascinating to investigate the learning time of the methods by increasing the number of data points, and as can be seen, the proposed method is the fastest compared to TSVM and \mathcal{U} TBSVM.

4.5 Gender detection by using face images

In 2008, Bai [23] showed that Universum data help to detect the gender from face images via SVM. Following [23], Shen [24] showed that Universum data improves the performance

Table 3 Performance comparison of nonlinear TSVM, \mathcal{L} TSVM, and $\mathcal{N}\mathcal{L}$ TSVM on UCI data sets

Data set Size	TSVM Acc (%) \pm Std Time (ms)	\mathcal{L} TSVM Acc (%) \pm Std Time (ms)	$\mathcal{N}\mathcal{L}$ TSVM Acc (%) \pm Std Time (ms)
Bupa	66.09 \pm 8.02	71.59 \pm 2.98	73.62 \pm 4.43
345 \times 6	48	70	11
Haberman	74.34 \pm 3.89	77.00 \pm 2.2	76.00 \pm 2.26
306 \times 3	115	191	26
Heart	64.15 \pm 9.24	71.30 \pm 5.14	70.94 \pm 7.32
270 \times 16	26	73	6
Ionosphere	93.14 \pm 1.67	94.29 \pm 3.94	95.71 \pm 0.9
351 \times 34	62	276	18
Pima	75.16 \pm 4.27	76.99 \pm 4.20	78.17 \pm 4.81
768 \times 9	1151	2985	45
Sonar	81.95 \pm 7.65	84.88 \pm 4.20	86.34 \pm 2.49
208 \times 60	41	61	4
Spect	78.87 \pm 8.30	80.00 \pm 6.92	80.00 \pm textbf6.92
237 \times 22	47	162	14
Trans	76.38 \pm 4.44	76.24 \pm 4.58	76.64 \pm 4.43
10 \times 32	159	334	34
House Votes	87.67 \pm 3.17	88.14 \pm 0.87	90.47 \pm 3.56
435 \times 16	127	168	6
Wpbc	96.64 \pm 0.66	97.52 \pm 1.03	98.05 \pm 0.35
198 \times 33	243	1323	26

Table 4 Performance comparison of linear TSVM, \mathcal{L} TSVM, and $\mathcal{N}\mathcal{L}$ TSVM on NDC data sets

Data set Size	TSVM Acc (%) \pm Std Time (ms)	\mathcal{L} TSVM Acc (%) \pm Std Time (ms)	$\mathcal{N}\mathcal{L}$ TSVM Acc (%) \pm Std Time (ms)
100 \times 10	84.00 \pm 5.63	94.41 \pm 2.07	94.81 \pm 3.93
	29	44.5	5.7
500 \times 10	73.87 \pm 2.64	93.05 \pm 5.51	93.47 \pm 4.84
	78.50	102.8	11.5
1000 \times 10	70.25 \pm 2.29	87.73 \pm 2.44	88.02 \pm 1.15
	85.2	320.9	26.9
2000 \times 10	70.07 \pm 2.00	86.88 \pm 0.95	86.88 \pm 0.72
	175.5	1411.8	48
5000 \times 10	71.03 \pm 0.32	87.08 \pm 1.09	87.26 \pm 0.73
	965.8	14501.9	231.6

of \mathcal{U} Boost for gender classification. In this subsection, we use the research's process mentioned in gender classification to evaluate the performance of the proposed method by using Universum data.

We perform our experiments on the face images for detecting gender. Each image is converted to gray scale at the 256-level and down-sampled to 45×50 pixels to form a 2250 dimensional vector (see [23] for more details on this data set). We collect facial images of 40 men and 10 women and consider 5 images for each person. Figure 10 depicts a few face photos of both men and women, including their facial expressions, hairstyles, and eyeglasses. According to the last column, we created the Universum data by averaging pairs of male and female face photos. We selected randomly 2 women and 10 men for training and the remaining individuals for testing (of which 12 subjects are randomly selected for training, and the remaining 38 for testing). A single picture among five is randomly chosen for each individual.

Results are presented in Table 5. The mean accuracy and standard deviation are reported on 5 independent runs and the best results are shown in bold. Table 5 clearly shows that in gender classification our method performed better than the other three methods.

5 Conclusion and future work

In recent years, the new concept of Universum has been proposed and defined as a set of instances not belonging to any class. Previous researches have shown that Universum

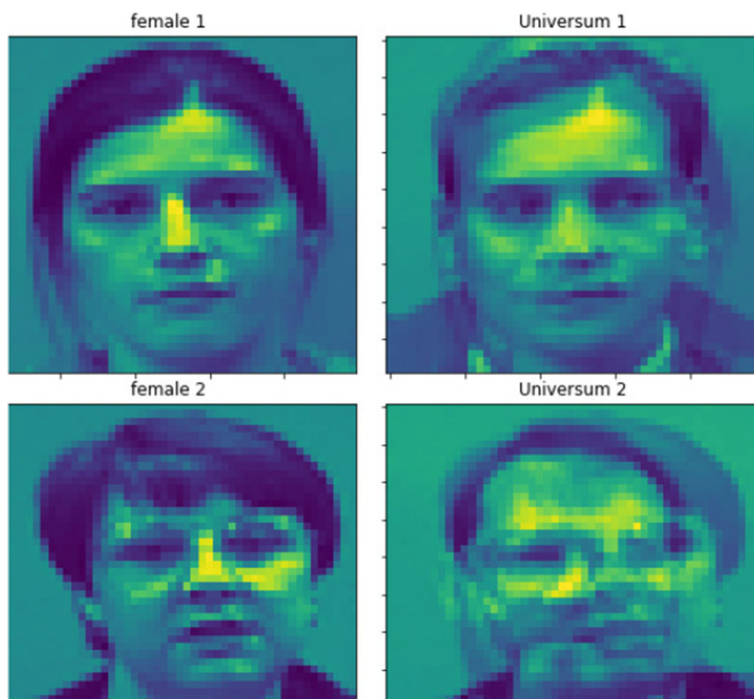


Fig. 10 The first column shows two men and the second column shows two women and the third column illustrates the Universum of a man and a woman in each row

Table 5 Gender recognition using TSVM, \mathcal{L} SVM, \mathcal{L} TBSVM and N \mathcal{L} TBSVM

TSVM	\mathcal{L} SVM	\mathcal{L} TBSVM	N \mathcal{L} TBSVM
Acc (%) \pm Std (C1, C2, ε)	Acc (%) \pm Std (C, Cu, ε)	Acc (%) \pm Std (C1, C2, Cu, ε)	Acc (%) \pm Std (C1, C2, Cu, ε)
73.83 \pm 5.00 (0.03125, 32.0, 0.1)	80.28 \pm 4.93 (0.039, 16, 0.1)	82.50 \pm 3.62 (0.5, 0.03125, 0.5, 0.5)	83.33 \pm 2.40 (2.0, 0.03125, 0.03125, 0.5)

data are very useful for supervised learning. This research introduced a novel approach to solve the optimization problems of \mathcal{L} TBSVM. By leveraging the 2-norm of the slack vectors in the objective functions of two quadratic problems, we transformed the constrained quadratic programming problems (\mathcal{L} TBSVMs) into unconstrained quadratic problems. Then, we suggested an extension of Newton's approach for addressing unconstrained quadratic problems, which makes solving the related problems fast and simple.

In order to demonstrate the superior efficiency of the suggested approach in both the linear and nonlinear cases, numerical tests were carried out on a synthetic data set, UCI data sets, NDC data sets, and face images data sets. The results shown that the proposed method in both the linear and nonlinear cases has higher efficiency than TSVM and \mathcal{L} TBSVM.

Although the randomly generated Universum data positively affects the data set classification, for future work, we will propose an Universum data selection method to increase the classification accuracy and reduce the learning time. Furthermore, the proposed method can be applied to other real-world applications involving binary data sets, such as disease diagnosis.

Funding H. Moosaei's research was funded by the Center for Foundations of Modern Computer Science (Charles Univ. project UNCE/SCI/004) and the Czech Science Foundation Grant 22-19353S. The work of M. Hladík was supported by the Czech Science Foundation Grant P403-22-11117S. M.R. Guarracino's work has been partially funded by the BiBiNet project (H35F21000430002) within POR-Lazio FESR 2014-2020, and conducted within the framework of the Basic Research Program at the National Research University Higher School of Economics (HSE).

Data Availability The data that support the findings of this study are available from the UCI machine learning repository, MC-NDC data sets, Face image data sets, associated with the following links: <https://archive.ics.uci.edu/ml/index.php>, <https://github.com/dmusican/ndc>, <http://cswww.essex.ac.uk/mv/allfaces/index.html>.

Disclosure statement

No potential conflict of interest was reported by the authors.

References


1. Arabasadi, Z., Alizadehsani, R., Roshanzamir, M., Moosaei, H., Yarifard, A.A.: Comput. Methods Programs Biomed. **141**, 19 (2017)
2. Wang, X.Y., Wang, T., Bu, J.: Pattern Recognit. **44**(4), 777 (2011)
3. Tong, S., Koller, D.: J. Mach. Learn. Res. **2**(Nov), 45 (2001)
4. Guarracino, M.R., Cuciniello, S., Pardalos, P.M.: J. Optim. Theory Appl. **141**(3), 533 (2009). <https://doi.org/10.1007/s10957-008-9496-x>
5. Cai, Y.D., Ricardo, P.W., Jen, C.H., Chou, K.C.: J. Theor. Biol. **226**(4), 373 (2004)
6. Javadi, S.H., Moosaei, H.D.: Ciunzo, Sensors **19**(3), 635:1 (2019)
7. Bazikar, F., Ketabchi, S., Moosaei, H.: Appl. Intell. **50**(6), 1763 (2020)
8. Ketabchi, S., Moosaei, H., Razzaghi, M., Pardalos, P.M.: Ann. Oper. Res. **276**(1–2), 155 (2019)
9. Cortes, C., Vapnik, V.: Machine Learning **20**(3), 273 (1995)

10. Vapnik, V.: The Nature of Statistical Learning Theory (Springer, 2013)
11. Weston, J., Collobert, R., Sinz, F., Bottou, L., Vapnik, V.: In Proceedings of the 23rd international conference on Machine learning, pp. 1009–1016 (2006)
12. Jayadeva, R., Khemchandani, S.: Chandra: IEEE Trans. Pattern Anal. Mach. Intell. **29**(5), 905 (2007)
13. Qi, Z., Tian, Y., Shi, Y.: Neural Netw. **36**, 112 (2012)
14. Richhariya, B., Sharma, A., Tanveer, M.: In 2018 IEEE Symposium Series on Computational Intelligence (IEEE SSCI 2018), ed. by S. Sundaram (IEEE, 2018), pp. 2045–2052
15. Shao, Y.H., Zhang, C.H., Wang, X.B., Deng, N.Y.: IEEE Trans. Neural Netw. **22**(6), 962 (2011)
16. Mangasarian, O.: J. Optim. Theory Appl. **121**(1), 1 (2004)
17. Pardalos, P.M., Ketabchi, S., Moosaei, H.: Optimization **63**(3), 359 (2014)
18. Andersen, M.S., Dahl, J., Vandenberghe, L., et al.: Available at cvxopt. org 54 (2013)
19. Harris, C.R., Millman, K.J., van der Walt, S.J., et al.: Nature **585**(7825), 357 (2020)
20. Hsu, C.W., Chang, C.C., Lin, C.J., et al: A practical guide to support vector classification (2003). <https://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf>
21. Musicant, D.R.: NDC: normally distributed clustered datasets (1998). <https://research.cs.wisc.edu/dmi/svm/ndc/>
22. Moosaei, H., Musicant, D., Khosravi, S., Hladík, M.: Carleton College, University of Bojnord (2020). <https://github.com/dmusicant/ndc>
23. Bai, X., Cherkassky, V.: In 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence) (IEEE, 2008), pp. 746–750
24. Shen, C., Wang, P., Shen, F., Wang, H.: IEEE Trans. Pattern Anal. Mach. Intell. **34**(4), 825 (2011)

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

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.

Authors and Affiliations

Hossein Moosaei^{1,2,3} · Saeed Khosravi⁴ · Fatemeh Bazikar⁵ · Milan Hladík^{6,7} · Mario Rosario Guarracino^{8,9} 

Hossein Moosaei
hmoosaei@gmail.com; hossein.moosaei@ujep.cz

Saeed Khosravi
saeedkhosravi72@gmail.com

Fatemeh Bazikar
f.bazikar@gmail.com; fatemeh_bazikar@phd.guilan.ac.ir

Milan Hladík
hladik@kam.mff.cuni.cz

¹ Department of Informatics, Faculty of Science, Jan Evangelista Purkyně University, Ústí nad Labem, Czech Republic

² Department of Applied Mathematics, School of Computer Science, Faculty of Mathematics and Physics, Charles University, Prague, Czech Republic

³ Prague University of Economics and Business, Prague, Czech Republic

⁴ Department of Computer Science, Faculty of Science, University of Bojnord, Bojnord, Iran

⁵ Department of Applied Mathematics, Faculty of Mathematical Sciences, University of Guilan, Rasht, Iran

⁶ Department of Applied Mathematics, Faculty of Mathematics and Physics, Charles University, Prague, Czech Republic

⁷ Department of Econometrics, University of Economics, Prague, Czech Republic

⁸ Department of Economics and Law, University of Cassino and Southern Lazio Campus Folcara, Cassino, Italy

⁹ Laboratory of Algorithms and Technologies for Networks Analysis, National Research University Higher School of Economics, Nizhny Novgorod, Russia