European Journal of Operational Research xxx (xxxx) xxx

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Computational Intelligence & Inform. Management

A kernel-free double well potential support vector machine with applications

Zheming Gao[a], Shu-Cherng Fang[b], Jian Luo[c,*], Negash Medhin[d]

[a] Graduate Program in Operations Research, North Carolina State University, Raleigh, NC 27695, USA
[b] Edward P. Fitts Department of Industrial and Systems Engineering, North Carolina State University, Raleigh, NC 27695, USA
[c] School of Management Science and Engineering, Dongbei University of Finance and Economics, Dalian 116025, China
[d] Department of Mathematics, North Carolina State University, Raleigh, NC 27695, USA

ARTICLE INFO

Article history:
Received 7 January 2020
Accepted 27 October 2020
Available online xxx

Keywords:
Data science
Support vector machine
Double well potential function
Kernel-free SVM
Binary classification

ABSTRACT

As a well-known machine learning technique, support vector machine (SVM) with a kernel function achieves much success in nonlinear binary classification tasks. Recently, some quadratic surface SVM models are proposed and studied by utilizing quadratic surfaces for nonlinear binary separations. In this paper, a kernel-free soft quartic surface SVM model is proposed by utilizing the double well potential function for highly nonlinear binary classification. Mathematical analysis on the theoretical properties of the proposed model, including the existence, uniqueness and support vector representation of optimal solutions, is shown. The sequential minimal optimization algorithm is adopted to implement the proposed model for computational efficiency. Numerical results on some artificial and public benchmark data sets demonstrate its effectiveness over well-known SVM models with or without kernel functions. The proposed model is extended to successfully handle some real-life corporate and personal credit data sets for applications.

1. Introduction

Binary classification has made a crucial impact in many fields. Various models and algorithms have been proposed in recent years. Support vector machine (SVM) was proposed in late 1990's (Cortes & Vapnik, 1995), and has been well studied with many successful applications. Given a data set, the classical linear SVM model (Cortes & Vapnik, 1995) separates the data points into two classes utilizing a hyperplane, while the margin between the two classes is maximized and the misclassification of data points is minimized. It works effectively for linearly separable data sets but it may fail for nonlinearly separable data sets. SVM models with kernel functions (Cortes & Vapnik, 1995) were proposed to fix this drawback. They first map all points into a higher dimensional feature space, and then the mapped points are classified by a linear hyperplane.

Although achieving a great success in many applications, SVMs with kernel functions may have some disadvantages (Blanquero, Carrizosa, Jimnez-Cordero, & Mart-n-Barragn, 2019; Chen, Fan, & Sun, 2016). First, there is no general principle to pre-select a suitable kernel function for a given data set. Besides, the performance of SVM models with a particular kernel function depends heavily on the parameters embedded in the kernel function (Scholkopf & Smola, 2001). Moreover, using some kernel functions may be computationally expensive since the inverse of a kernel matrix is needed for solving the dual problem, or a decomposition of the kernel matrix is needed for solving the primal problem (Cristianini & Shawe-Taylor, 2000). In addition, the singularity issue of a kernel matrix may influence the accuracy of classification. It should be noted that the developed sequential minimal optimization (SMO) algorithm in Platt (1998) helps avoid the decomposition of kernel matrix (Fan, Chen, & Lin, 2005) and improves the training efficiency of the kernel-based SVM models for supervised classification.

To overcome those drawbacks, Dagher (2008) proposed a kernel-free quadratic SVM model, which has been extended to a soft-margin quadratic surface SVM (SQSSVM) model (Luo, Fang, Deng, & Guo, 2016; Mousavi, Gao, Han, & Lim, 2019; Yan, Bai, Fang, & Luo, 2018). The main idea of SQSSVM is to seek a quadratic separation surface, which maximizes the sum of relative geometrical margins (Dagher, 2008; Luo et al., 2016) and penalizes the misclassifications of all data points. There are other kernel-free SVM models proposed in literature. Astorino and Fuduli (2015) di-

* Corresponding author.
E-mail addresses: zgao5@ncsu.edu (Z. Gao), fang@ncsu.edu (S.-C. Fang), luojian546@hotmail.com (J. Luo), ngmedhin@ncsu.edu (N. Medhin).

https://doi.org/10.1016/j.ejor.2020.10.040
0377-2217/

rectly constructed a spherical surface with no kernel for semi-supervised separation, but it might not be suitable for supervised binary classification problems. Moreover, the spherical separation surface is a special type of quadratic surfaces, which may have more limitations to handle highly nonlinear cases. Bai, Han, Chen, and Yu (2015) proposed a quadratic kernel-free least square SVM and applied it to target disease classification. Tian, Yong, and Luo (2018) proposed a kernel-free fuzzy quadratic surface SVM and applied it for the reject inference procedure in credit scoring. Gao, Bai, and Zhan (2019) proposed a quadratic kernel-free least square twin SVM. These quadratic kernel-free SVM models work well for some applications, but they still have some limitations: First, the used separating quadratic surfaces in these models may not well handle highly nonlinear separable data sets. In addition, these models adopt the relative geometrical margins of all points to measure the separability between two classes so that it requires extra time to calculate the coefficient matrix in the objective function of a corresponding model.

Double well potential (DWP) function attracts considerable attentions in quantum mechanics and solid mechanics (Gao & Yu, 2008; Heuer & Haeberlen, 1991). It is a special category of degree-4 multi-variate polynomial function, where a quadratic term is embedded in a quadratic function. The DWP function was utilized in the numerical approximation to the generalized Ginzburg-Landau functional (Gao & Yu, 2008). Some optimization properties of DWP function were studied in Fang, Gao, Lin, Sheu, and Xing (2017) and Xia, Sheu, Fang, and Xing (2017). Our motivation to investigate DWP function comes from the aim to build a kernel-free SVM model that can handle highly nonlinearly separable data. Recall that the linear SVM model was initially proposed for binary classification. Since not all data sets can be linearly separated, the kernel functions were equipped on SVM models for nonlinear binary classification. To overcome some drawbacks of kernel-based SVM models, the kernel-free QSSVM model was proposed by directly using quadratic surfaces for nonlinear separation. Although QSSVM works well on some data, it cannot well handle highly nonlinearly separable data. As a forth degree polynomial function, the DWP function is more "nonlinear" than a quadratic or a cubic function. Therefore, DWP function has a strong potential for highly nonlinear classifications.

In this paper we propose a DWP based kernel-free nonlinear SVM model, which is denoted as DWPSVM. Certain theoretical properties are studied, including the solution existence, uniqueness and support vector representation. Numerical experiments are conducted to investigate the effectiveness and efficiency of the proposed DWPSVM model. Besides, it is applied to credit scoring with real-life corporate and benchmark personal credit data. The main contributions of this paper to the field of binary classification include:

(1) To the best of our knowledge, this is the first study of proposing a kernel-free quartic surface SVM (i.e. DWPSVM) for binary classification. The proposed kernel-free DWPSVM model handles the drawbacks induced by kernel functions in classical SVM models, which may save the efforts used in selecting suitable kernel functions and tuning related kernel parameters.
(2) The proposed DWPSVM model is theoretically and numerically investigated in this paper. It outperforms those well-known kernel-based SVM models and kernel-free SVM models for binary classifications. The well-known SMO algorithm is adopted to implement the proposed model for computational efficiency. Numerical results indicate its increasing dominance in classification accuracy as data features increases. Moreover, DWPSVM has the potential to be applied to solve some real-life problems.

The rest of the paper is organized as follows. In Section 2, we briefly review some related works in binary classification using SVM models in the literature. The DWP function is introduced to propose a DWPSVM model based on a newly-derived margin (i.e., G-margin) in Section 3. Then we investigate the theoretical properties of DWPSVM in Section 4. Computational experiments are conducted using the artificial, public benchmark and real-life credit data sets in Section 5. Section 6 concludes the paper.

2. Preliminaries

In this section, we introduce some preliminary knowledge and briefly review some related SVM models for binary classification.

2.1. Mathematical notations

Throughout this article, we use lower case letters to denote scalars, bold lower case letters to denote vectors, and bold upper case letters to denote matrices. We denote the n-dimensional real space by $\mathbb{R}^n$, n-dimensional nonnegative orthant by $\mathbb{R}^n_+$, zero matrix of size $m\times n$ by $\mathbf{0}_{m\times n}$, $n\times n$ identity matrix by $\mathbf{I}_n$, all-one matrix of size $m\times n$ by $\mathbf{1}_{m\times n}$, diagonal matrix with vector $\boldsymbol{a}=[a_1,\ldots,a_n]^T$ on its diagonal by $\mathrm{Diag}(a_1,\ldots,a_n)$ and the set of all $n\times n$ real symmetric matrices by $\mathbb{S}^n$. For any $\mathbf{A}\in\mathbb{S}^n$, we write $\mathbf{A}\succ 0$ if $\mathbf{A}$ is positive definite and $\mathbf{A}\succeq 0$ if $\mathbf{A}$ is positive semidefinite. Given any matrix $\mathbf{B}\in\mathbb{R}^{m\times n}$, $\mathbf{B}_{i\bullet}$ stands for the ith row of $\mathbf{B}$, and $\mathbf{B}_{\bullet j}$ stands for the jth column of $\mathbf{B}$.

For any square matrix $\mathbf{A}=\left[A_{ij}\right]_{i,j=1,\ldots,n}\in\mathbb{R}^{n\times n}$, its vectorization is defined as a vector of size n^2 formed by stacking up the columns of $\mathbf{A}$, i.e., the vectorization of $\mathbf{A}$ is given by

$$\mathrm{vec}(\mathbf{A})\triangleq[A_{11},\ldots,A_{n1},A_{12},\ldots,A_{n2},\ldots,A_{1n},\ldots,A_{nn}]^T\in\mathbb{R}^{n^2}.$$

When $\mathbf{A}$ is symmetric, all information of $\mathbf{A}$ is included in the upper triangular $\frac{n(n+1)}{2}$ elements (Dagher, 2008; Luo et al., 2016; Mousavi et al., 2019). Hence, it is enough to consider the following half-vectorization of $\mathbf{A}\in\mathbb{S}^n$:

$$\mathrm{hvec}(\mathbf{A})\triangleq[A_{11},\ldots,A_{1n},A_{22},\ldots,A_{2n},\ldots,A_{n-1,n-1},A_{n-1,n},A_{nn}]^T\in\mathbb{R}^{\frac{n(n+1)}{2}}.$$

For any vector $\boldsymbol{a}=[a_1,\ldots,a_n]^T\in\mathbb{R}^n$, we use $\mathrm{lvec}(\boldsymbol{a})$ to denote the vector of the cross terms of its elements (Dagher, 2008; Luo et al., 2016; Mousavi et al., 2019):

$$\mathrm{lvec}(\boldsymbol{a})\triangleq\left[\frac{1}{2}a_1a_1,a_1a_2,\ldots,a_1a_n,\frac{1}{2}a_2a_2,a_2a_3,\ldots,a_2a_n,\ldots,\frac{1}{2}a_na_n\right]^T\in\mathbb{R}^{\frac{n(n+1)}{2}}.$$

Notice that $\frac{1}{2}\boldsymbol{a}^T\mathbf{A}\boldsymbol{a}=\mathrm{lvec}(\boldsymbol{a})^T\mathrm{hvec}(\mathbf{A})$, then a quadratic term with respective to $\boldsymbol{a}$ can be substituted by a linear term.

For any vector $\boldsymbol{a}$, denote its ℓ_2 norm as $\|\boldsymbol{a}\|_2\triangleq\sqrt{\boldsymbol{a}^T\boldsymbol{a}}$. For any symmetric matrix $\mathbf{A}$, denote its matrix ℓ_2-norm and Frobenious norm as the following:

$$\|\mathbf{A}\|_F\triangleq\left(\sum_{i=1}^n\sum_{j=1}^n|A_{ij}|^2\right)^{1/2}\qquad\|\mathbf{A}\|_2\triangleq\sqrt{\lambda_{\max}(\mathbf{A}^T\mathbf{A})}$$

where $\lambda_{\max}$ represents the largest eigenvalue of $\mathbf{A}$.

To connect Frobenious norm with $\mathrm{hvec}(\mathbf{A})$, define matrix $\mathbf{H}_n\in\mathbb{S}^{\frac{n(n+1)}{2}}$ such that

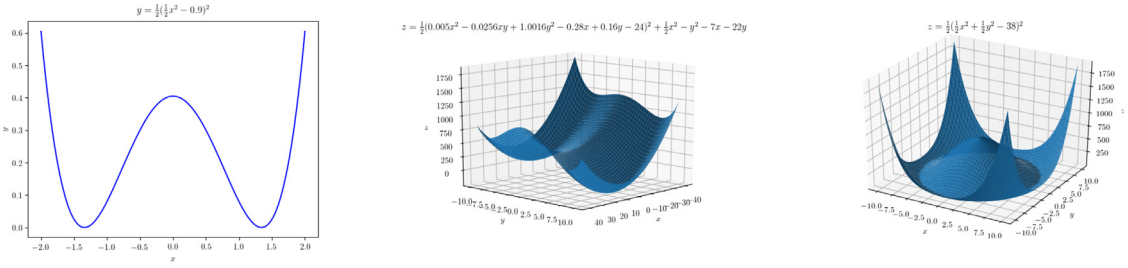
$$\mathrm{hvec}(\mathbf{A})^T\mathbf{H}_n\,\mathrm{hvec}(\mathbf{A})=\|\mathbf{A}\|_F^2$$

When $n=3$, $\mathbf{H}_3=\mathrm{Diag}(1,2,2,1,2,1)$. And in general,

$$\mathbf{H}_n\triangleq 2\mathbf{I}_{\frac{n(n+1)}{2}}-\mathrm{Diag}(\mathrm{hvec}(\mathbf{I}_n)).\tag{1}$$

Given a matrix $\mathbf{A}\in\mathbb{R}^{n\times n}$, the singular value decomposition (SVD) of $\mathbf{A}$ is represented as

$$\mathbf{A}=\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$$



(a) Example 1 (b) Example 2 (c) Example 3

Fig. 1. DWP function examples.

where $\mathbf{U} \in \mathbb{R}^{n\times n}$, $\mathbf{\Sigma} = \text{Diag}(\sigma_1, \ldots, \sigma_n) \in \mathbb{R}^{n\times n}$ and $\mathbf{V} \in \mathbb{R}^{n\times n}$. Throughout this article, the singular values are ordered in sequence such that $\sigma_1 \geqslant \cdots \geqslant \sigma_r \geqslant \sigma_{r+1} = \cdots = \sigma_n = 0$. If matrix $\mathbf{A} \in \mathbb{S}^n$, then its matrix ℓ_2-norm is related to its biggest singular value (Meyer, 2000).

Lemma 2.1. *Let* $\mathbf{A} \in \mathbb{S}^n$, *then* $\|\mathbf{A}\|_2 = \sigma_1(\mathbf{A})$ *and* $\|\mathbf{A}\|_F = \left(\sum_{i=1}^n \sigma_i^2(\mathbf{A})\right)^{1/2}$.

2.2. *Related SVM models for binary classification*

Given a data set of two classes, the goal of binary classification is to find a separation surface to separate them as accurate as possible. For any binary classification problem, the data set with two classes can be mathematically denoted by

$$\mathcal{D} = \left\{ \left(\boldsymbol{x}^{(i)}, y^{(i)}\right)_{i=1,\cdots,N} \;\middle|\; \boldsymbol{x}^{(i)} \in \mathbb{R}^n,\ y^{(i)} \in \{-1, 1\} \right\}, \tag{2}$$

where N is the data size, n is the number of features, $\boldsymbol{x}^{(i)} = [x_1^{(i)}, \ldots, x_n^{(i)}]^T \in \mathbb{R}^n$ is the vector of n feature values of point i, and $y^{(i)}$ is the label of point $\boldsymbol{x}^{(i)}$. Denote the positive and negative labeled index sets as $\mathcal{M}^+ \triangleq \{i \,|\, y^{(i)} = 1\}$ and $\mathcal{M}^- \triangleq \{i \,|\, y^{(i)} = -1\}$, and let the total index set be $\mathcal{M} \triangleq \mathcal{M}^+ \cup \mathcal{M}^-$. Denote their cardinalities as N^+ and N^-, respectively, and notice that $N = N^+ + N^-$. In this article we assume that both $\mathcal{M}^+$ and $\mathcal{M}^-$ are nonempty. The goal of binary classification is to actually separate the data by a classifier.

According to Deng, Tian, and Zhang (2012), a data set $\mathcal{D}$ is *linearly separable* if there exists $\boldsymbol{u} \in \mathbb{R}^n$, and $d \in \mathbb{R}$ such that

$$\boldsymbol{u}^T\boldsymbol{x}^{(i)} + d > 0 \quad (i \in \mathcal{M}^+), \quad \boldsymbol{u}^T\boldsymbol{x}^{(i)} + d < 0, \quad (i \in \mathcal{M}^-). \tag{3}$$

Given a linearly separable data set $\mathcal{D}$, the idea of SVM is to separate the data by a hyperplane while the margin of separation is maximized (Cortes & Vapnik, 1995). Denote the separation function as $f(\boldsymbol{x}) = \boldsymbol{u}^T\boldsymbol{x} + d$, then the width of margin equals to $\frac{2}{\|\boldsymbol{u}\|_2}$. An example on $\mathbb{R}^2$ is shown in Fig. 2a. If the data set $\mathcal{D}$ is not linearly separable, the soft-margin idea (Cortes & Vapnik, 1995) is adopted by introducing the slack vector $\boldsymbol{\xi} = [\xi_1, \ldots, \xi_N]^T \in \mathbb{R}^N$ to allow the location of points to violate constraints. The soft-margin SVM is formulated as the following model (SSVM):

$$\begin{aligned} \min \quad & \frac{1}{2}\|\boldsymbol{u}\|_2^2 + C\sum_{i=1}^N \xi_i \\ s.t. \quad & y^{(i)}\left(\boldsymbol{u}^T\boldsymbol{x}^{(i)} + d\right) \geqslant 1 - \xi_i \quad \forall i = 1, \ldots, N \\ & \boldsymbol{u} \in \mathbb{R}^n, d \in \mathbb{R}, \boldsymbol{\xi} \in \mathbb{R}_+^N. \end{aligned} \tag{SSVM}$$

where $C > 0$ is the penalty parameter for data points.

However, most data sets are not linearly separable and a nonlinear separation surface is more appropriate than a linear one. The nonlinear classification task could be done by an SVM model with a kernel function, which was proposed in Vapnik (2013) and equivalent to the formulation as the following:

$$\begin{aligned} \min \quad & \frac{1}{2}\|\boldsymbol{v}\|_2^2 + C\sum_{i=1}^N \xi_i \\ s.t. \quad & y^{(i)}\left(\boldsymbol{v}^T\phi(\boldsymbol{x}^{(i)}) + d\right) \geqslant 1 - \xi_i \quad \forall i = 1, \ldots, N \\ & \boldsymbol{v} \in \mathbb{R}^l, d \in \mathbb{R}, \boldsymbol{\xi} \in \mathbb{R}_+^N. \end{aligned} \tag{SSVM-kernel}$$

where ϕ maps data point $\boldsymbol{x}^{(i)}$ from $\mathbb{R}^n$ to $\mathbb{R}^l$ $(m < l)$ and $K(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)}) = \phi(\boldsymbol{x}^{(i)})^T\phi(\boldsymbol{x}^{(j)})$ is a kernel function for any $\boldsymbol{x}^{(i)}$ and $\boldsymbol{x}^{(j)}$. There are various kernel functions including the frequently used Gaussian (RBF) kernel and Quadratic (2nd order polynomial) kernel (Scholkopf & Smola, 2001). The idea of an SVM with a kernel function is to first map the data points into a higher dimensional feature space and then separate the mapped data points with a hyperplane in the higher dimensional space.

Notice that (SSVM-kernel) is also a convex quadratic programming (QP) problem, it is important to studied its dual problem, which can be formulated as the following:

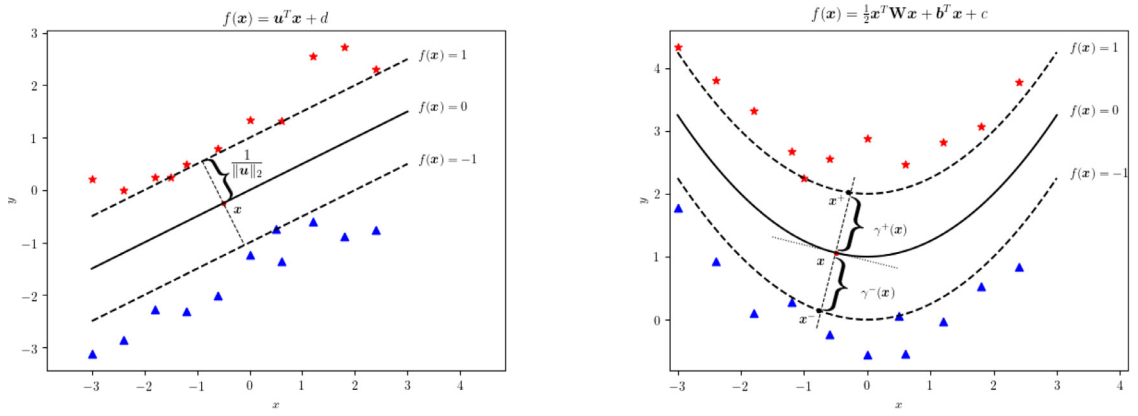
$$\begin{aligned} \min \quad & \frac{1}{2}\sum_{i=1}^N\sum_{j=1}^N K(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)})y^{(i)}y^{(j)}\alpha_i\alpha_j - \sum_{i=1}^N \alpha_i \\ s.t. \quad & \sum_{i=1}^N \alpha_i y^{(i)} = 0 \\ & 0 \leqslant \alpha_i \leqslant C, \quad i = 1, \ldots, N. \end{aligned} \tag{DSSVM-kernel}$$

where C is the parameter of penalties. The details of derivation for this dual problem can be found in Vapnik (2013). Even though the dual gap is zero from duality theory, the dual problem (DSSVM-kernel) is simpler as it has only one linear equality constraint with the upper bounds of variables. Hence, the (SSVM-kernel) model is usually trained from its dual side. One of the most popular approaches proposed in literature for solving its dual problem is the sequential minimal optimization (SMO) algorithm (Platt, 1998), which has been adopted in software packages such as LIBSVM (Chang & Lin, 2011).

Moreover, kernel-free nonlinear SVM models were proposed and developed in Dagher (2008) and Luo et al. (2016) by directly utilizing quadratic surfaces for separations. These kernel-free SVM models share the similar idea of separating the data in the original space instead of mapping the data onto a higher dimensional feature space. According to Luo et al. (2016), a data set $\mathcal{D}$ is *quadratically separable* if there exists $\mathbf{W} \in \mathbb{S}^n$, $\boldsymbol{b} \in \mathbb{R}^n$, and $c \in \mathbb{R}$ such that

$$\begin{aligned} &\frac{1}{2}\boldsymbol{x}^{(i)^T}\mathbf{W}\boldsymbol{x}^{(i)} + \boldsymbol{b}^T\boldsymbol{x}^{(i)} + c > 0 \quad \left(i \in \mathcal{M}^+\right), \\ &\frac{1}{2}\boldsymbol{x}^{(i)^T}\mathbf{W}\boldsymbol{x}^{(i)} + \boldsymbol{b}^T\boldsymbol{x}^{(i)} + c < 0 \quad \left(i \in \mathcal{M}^-\right). \end{aligned} \tag{4}$$

Given a quadratically sparable data set, the quadratic separation surface obtained from the quadratic surface SVM (QSSVM)



(a) Original margin of SVM. (b) G-margin.

Fig. 2. Original margin vs G-margin.

(Luo et al., 2016) is represented by $f(\boldsymbol{x}) \triangleq \frac{1}{2}\boldsymbol{x}^T\mathbf{W}\boldsymbol{x} + \boldsymbol{b}^T\boldsymbol{x} + c = 0$. A typical model is the following SQSSVM model, which not only minimizes the sum of relative geometrical margins of points but also adopts the soft-margin idea (Luo et al., 2016):

$$\begin{aligned} \min \quad & \sum_{i=1}^{N} \|\mathbf{W}\boldsymbol{x}^{(i)} + \boldsymbol{b}\|_2^2 + C\sum_{i=1}^{N}\xi_i \\ s.t. \quad & y^{(i)}\left(\frac{1}{2}{\boldsymbol{x}^{(i)}}^T\mathbf{W}\boldsymbol{x}^{(i)} + {\boldsymbol{x}^{(i)}}^T\boldsymbol{b} + c\right) \geqslant 1 - \xi_i, \quad i = 1, \cdots, N, \\ & \mathbf{W} \in \mathbb{S}^n,\ \boldsymbol{b} \in \mathbb{R}^n,\ c \in \mathbb{R},\ \boldsymbol{\xi} \in \mathbb{R}^N_+. \end{aligned} \tag{SQSSVM}$$

In summary, SSVM model yields a linear separation function $H(\boldsymbol{x}) = \boldsymbol{u}^T\boldsymbol{x} + d$, with $\boldsymbol{u} \in \mathbb{R}^n$, $d \in \mathbb{R}$, while SQSSVM yields a quadratic separation function $Q(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^T\mathbf{W}\boldsymbol{x} + \boldsymbol{b}^T\boldsymbol{x} + c$, with $\mathbf{W} \in \mathbb{S}^n$, $\boldsymbol{b} \in \mathbb{R}^n$, $c \in \mathbb{R}$. The SVM with Guassian kernel yields a highly nonlinear separation generated by the Gaussian kernel function. It was studied in Luo et al. (2016) and Mousavi et al. (2019) that quadratic surfaces work more effectively in classification than a hyperplane. In this paper, we will investigate the performance of utilizing a special quartic surface for binary classification.

3. Double well potential support vector machine

In this section, we first introduce the double well potential function and then propose a new type of margin for measuring the distance between two classes. Finally, based on the new type of margin, a double well potential SVM model is proposed.

3.1. *Double well potential function*

Double well potential function is a special quartic function of interest in quantum mechanics, field theories and other research areas. It is defined as follows.

Definition 3.1 (Double Well Potential (DWP) function)**.** Let F be a real-value function defined on $\mathbb{R}^n$ such that

$$F(\boldsymbol{x}) = \frac{1}{2}\left(\frac{1}{2}\|\mathbf{B}\boldsymbol{x} - \boldsymbol{c}\|_2^2 - d\right)^2 + \frac{1}{2}\boldsymbol{x}^T\mathbf{A}\boldsymbol{x} + \boldsymbol{b}^T\boldsymbol{x} + q. \tag{5}$$

where $\mathbf{B} \in \mathbb{R}^{m\times n}$, $\boldsymbol{c} \in \mathbb{R}^m$, $d \in \mathbb{R}$, $\mathbf{A} \in \mathbb{S}^n$, $\boldsymbol{b} \in \mathbb{R}^n$, $q \in \mathbb{R}$.

Three examples of DWP functions are given in Fig. 1.

Given a DWP function f, and any data point $(\boldsymbol{x}^{(i)}, y^{(i)}) \in \mathcal{D}$ denoted by (2), define

$$\boldsymbol{\xi}^{(i)} \triangleq \frac{1}{2}\|\mathbf{B}\boldsymbol{x}^{(i)} - \boldsymbol{c}\|_2^2 - d. \tag{6}$$

Define $\boldsymbol{s}^{(i)} \triangleq \mathrm{lvec}(\boldsymbol{x}^{(i)})$, $\boldsymbol{w}_B \triangleq \mathrm{hvec}(\mathbf{B}^T\mathbf{B})$, $\boldsymbol{w}_{Bc} \triangleq \boldsymbol{c}^T\mathbf{B}$ and $c_d \triangleq \frac{1}{2}\boldsymbol{c}^T\boldsymbol{c} - d$, then we have $\boldsymbol{\xi}^{(i)} = {\boldsymbol{s}^{(i)}}^T\boldsymbol{w}_B - {\boldsymbol{x}^{(i)}}^T\boldsymbol{w}_{Bc} + c_d$. Define $\boldsymbol{z}^{(i)} \triangleq [{\boldsymbol{s}^{(i)}}^T, {\boldsymbol{x}^{(i)}}^T, 1]^T$ and $\boldsymbol{w}_\xi \triangleq [\boldsymbol{w}_B^T, \boldsymbol{w}_{Bc}^T, c_d]^T$. Therefore,

$$F(\boldsymbol{x}^{(i)}) = \tilde{F}\left(\begin{bmatrix}\boldsymbol{z}^{(i)}\\ \boldsymbol{x}^{(i)}\end{bmatrix}\right) \triangleq \frac{1}{2}{\boldsymbol{z}^{(i)}}^T\boldsymbol{w}_\xi{\boldsymbol{w}_\xi}^T\boldsymbol{z}^{(i)} + \frac{1}{2}{\boldsymbol{x}^{(i)}}^T\mathbf{A}\boldsymbol{x}^{(i)} + \boldsymbol{b}^T\boldsymbol{x}^{(i)} + q. \tag{7}$$

where function $\tilde{F} : \mathbb{R}^{\frac{n(n+1)}{2}+2n+1} \to \mathbb{R}$ has a quadratic term with respect to $\boldsymbol{z}^{(i)}$ on $\mathbb{R}^{\frac{n(n+1)}{2}+n+1}$ and another quadratic term with respect to $\boldsymbol{x}^{(i)}$ on $\mathbb{R}^n$. With the similar vectorization procedure, denote $l \triangleq \frac{n(n+1)}{2} + n + 1$, and

$$\boldsymbol{w}_W \triangleq \mathrm{hvec}(\boldsymbol{w}_\xi\boldsymbol{w}_\xi^T) \in \mathbb{R}^{l(l+1)/2}, \quad \boldsymbol{w}_A \triangleq \mathrm{hvec}(\mathbf{A}) \in \mathbb{R}^{n(n+1)/2},$$
$$\boldsymbol{\eta}^{(i)} \triangleq \mathrm{lvec}(\boldsymbol{z}^{(i)}) \in \mathbb{R}^{l(l+1)/2}.$$

$$\boldsymbol{v} \triangleq \begin{bmatrix}\boldsymbol{w}_W\\ \boldsymbol{w}_A\end{bmatrix} \in \mathbb{R}^{\frac{l(l+1)+n(n+1)}{2}}, \quad \boldsymbol{r}^{(i)} \triangleq \begin{bmatrix}\boldsymbol{\eta}^{(i)}\\ \boldsymbol{s}^{(i)}\end{bmatrix} \in \mathbb{R}^{\frac{l(l+1)+n(n+1)}{2}}. \tag{8}$$

Consequently, $F(\boldsymbol{x}^{(i)})$ equals to a linear function F_l with respect to $\boldsymbol{r}^{(i)}$ and $\boldsymbol{x}^{(i)}$ in $\mathbb{R}^{\frac{l(l+1)+n(n+1)}{2}+n}$, i.e.,

$$F(\boldsymbol{x}^{(i)}) = F_l\left(\begin{bmatrix}\boldsymbol{r}^{(i)}\\ \boldsymbol{x}^{(i)}\end{bmatrix}\right) \triangleq {\boldsymbol{r}^{(i)}}^T\boldsymbol{v} + {\boldsymbol{x}^{(i)}}^T\boldsymbol{b} + q. \tag{9}$$

In other words, we have following result:

Theorem 3.1. *A DWP function in $\mathbb{R}^n$ is equivalent to a linear function in $\mathbb{R}^{\frac{l(l+1)+n(n+1)}{2}+n}$, where $l = \frac{n(n+1)}{2} + n + 1$.*

Remark. As a highly nonlinear function, the DWP function represents some types of 4th order polynomial functions, which are commonly seen in physics and maybe in some aspects of real life. Moreover, the form of DWP function is an embedded quadratic function of the quadratic term so that it is more tractable than other 4th order polynomial function. Hence, the DWP surface may be amenable for the highly nonlinear classifications.

3.2. *G-margin*

To the best of our knowledge, the only margin adopted by the kernel-free SVM models is the relative geometric margin (Dagher, 2008; Luo et al., 2016; Mousavi et al., 2019). However, this margin is calculated by utilizing the positions of all data points, so the efficiency of these models may be sensitive to the data size. In this

subsection, we propose a new way to measure the margin between two classes of data, which has a simpler form while only using the local information of a separating surface. It may help overcome the drawbacks of the relative geometric margin.

The G-margin between two classes, illustrated for an example in $\mathbb{R}^2$ in Fig. 2b, is proposed here to separate the data points in $\mathbb{R}^n$ with a quadratic separation surface $\mathcal{S}$, where

$$\mathcal{S} \triangleq \{\boldsymbol{x} \in \mathbb{R}^n | Q(\boldsymbol{x}) = 0\}; \quad Q : \mathbb{R}^n \to \mathbb{R}, \ \boldsymbol{x} \mapsto \frac{1}{2}\boldsymbol{x}^T\mathbf{W}\boldsymbol{x} + \boldsymbol{b}^T\boldsymbol{x} + c. \tag{10}$$

For any $\boldsymbol{x} \in \mathcal{S}$, denote its normal vector by $\hat{\boldsymbol{n}}(\boldsymbol{x}) \triangleq \frac{\nabla Q(\boldsymbol{x})}{\|\nabla Q(\boldsymbol{x})\|_2}$. Let the straight line in the direction of $\hat{\boldsymbol{n}}(\boldsymbol{x})$ intersect with $Q(\boldsymbol{x}) = 1$ and $Q(\boldsymbol{x}) = -1$ at $\boldsymbol{x}^+$ and $\boldsymbol{x}^-$, respectively. Define $\gamma^+(\boldsymbol{x}) \triangleq \|\boldsymbol{x}^+ - \boldsymbol{x}\|_2$ and $\gamma^-(\boldsymbol{x}) \triangleq \|\boldsymbol{x}^- - \boldsymbol{x}\|_2$. In this way,

$$\begin{aligned} \boldsymbol{x}^+ &= \boldsymbol{x} + \gamma^+(\boldsymbol{x})\hat{\boldsymbol{n}}(\boldsymbol{x}), \quad Q(\boldsymbol{x}^+) = 1 \\ \boldsymbol{x}^- &= \boldsymbol{x} - \gamma^-(\boldsymbol{x})\hat{\boldsymbol{n}}(\boldsymbol{x}), \quad Q(\boldsymbol{x}^-) = -1 \end{aligned} \tag{11}$$

Definition 3.2 (G-margin on a quadratic surface)**.** Given a quadratic surface $\mathcal{S}$ denoted by (10). For any $\boldsymbol{x} \in \mathcal{S}$, the G-margin at $\boldsymbol{x}$ is defined as

$$\gamma(\boldsymbol{x}) \triangleq \gamma(\boldsymbol{x})^- + \gamma(\boldsymbol{x})^-.$$

The next lemma shows that G-margin is highly related to the singular values of $\mathbf{W}$. Notice that $\mathbf{W}$ is a symmetric matrix, there exists an orthonormal matrix $\mathbf{U} \in \mathbb{R}^{n\times n}$ so that the singular value decomposition of $\mathbf{W}$ is

$$\mathbf{W} = \mathbf{U}\mathbf{\Sigma}\mathbf{U}^T \tag{12}$$

where $\mathbf{\Sigma} = \mathrm{Diag}(\sigma_1, \ldots, \sigma_n)$ is a diagonal matrix of all singular values of $\mathbf{W}$. Remember that σ_i's are ordered decreasingly, i.e., $\sigma_1 \geqslant \cdots \geqslant \sigma_n$.

Lemma 3.2. *Given a quadratic surface $\mathcal{S}$ denoted by (10), for any $\boldsymbol{x} \in \mathcal{S}$, there exists $R(\boldsymbol{x}) \in [\sigma_n, \sigma_1]$ such thats*

$$\frac{1}{\gamma(\boldsymbol{x})} \geqslant \frac{\sqrt{R(\boldsymbol{x})}}{2}. \tag{13}$$

where σ_1 and σ_n are the biggest and the smallest singular values of matrix $\mathbf{W}$, respectively.

The proof is in A.1.

Lemma 3.3. *Given matrix $A \in \mathbb{R}^{n\times n}$, there exist $0 < \gamma_1 \leqslant \gamma_2$ such that*

$$\gamma_1\|\mathbf{A}\|_2 \leqslant \|\mathbf{A}\|_F \leqslant \gamma_2\|\mathbf{A}\|_2. \tag{14}$$

(14) is a specific case of the equivalence of matrix norms. The proof can be found in Meyer (2000).

3.3. Double Well Potential SVM model

The basic idea of SVM models for binary classification is to separate the two classes with the maximum margin. To the best of our knowledge, there is no margin defined for a quartic separation surface in literature. Remember that DWP function can be transformed to a quadratic form in Section 3.1. Since G-margin is defined for the quadratic separation surface, it is possible to extend G-margin for the DWP separation surface. Here, we first introduce a new quadratic surface SVM model adopting G-margin, then propose a kernel-free DWPSVM model after extending G-margin to the DWP separation surface.

Given any quadratic surface $\mathcal{S}$ as defined in (10), denote the G-margin at $\boldsymbol{z} \in \mathcal{S}$ by $\gamma(\boldsymbol{z})$ as in Definition 3.2. In order to maximize the G-margin at $\boldsymbol{z}$, it is equivalent to minimize $1/\gamma(\boldsymbol{z})$. By reaching a similar goal, we minimize the maximum of the bound provided by Lemma 3.2 as follows.

$$\min_{\substack{\mathbf{W}\in\mathbb{S}^n \\ \boldsymbol{b}\in\mathbb{R}^n, c\in\mathbb{R}}} \max_{\boldsymbol{z}\in\mathcal{S}} \frac{\sqrt{R(\boldsymbol{z})}}{2} \tag{15}$$

Since $\sqrt{R(\boldsymbol{z})} \in [\sigma_n, \sigma_1]$ and $\sigma_1 = \|\mathbf{W}\|_2$, a similar goal can be reached by solving the following optimization problem (16):

$$\min_{\substack{\mathbf{W}\in\mathbb{S}^n \\ \boldsymbol{b}\in\mathbb{R}^n, c\in\mathbb{R}}} \frac{1}{2}\|\mathbf{W}\|_2^{1/2} \tag{16}$$

In this way, a quadratic surface SVM model can be formulated with G-margin. Given a data set $\mathcal{D}$ denoted by (2), a kernel-free model GQSVM is formulated as follows.

$$\begin{aligned} \min \quad & \frac{1}{2}\|\mathbf{W}\|_F^2 \\ s.t. \quad & y^{(i)}\left(\frac{1}{2}\boldsymbol{x}^{(i)^T}\mathbf{W}\boldsymbol{x}^{(i)} + \boldsymbol{b}^T\boldsymbol{x}^{(i)} + c\right) \geqslant 1, \quad i = 1, \ldots, N. \\ & \mathbf{W} \in \mathbb{S}^n, \boldsymbol{b} \in \mathbb{R}^n, c \in \mathbb{R} \end{aligned} \tag{GQSVM}$$

Notice that, the matrix ℓ_2-norm is substituted by the Frobenius norm in the objective. By Lemma 3.3, $\|\mathbf{W}\|_F$ is equivalent to $\|\mathbf{W}\|_2$. In addition, the F-norm helps vectorize the matrix variable into a vector variable, which makes it easier for implementation.

Based on a similar idea, we apply G-margin to propose a DWPSVM model. Moreover, a regularization term $\|\boldsymbol{b}\|_2^2$ is added to facilitate the design of an SMO algorithm for the following DWPSVM model. Notice that, other than the measurement of margin, GQSVM does not produce a soft margin as the SQSSVM model does. To allow the location of data points to violate constraints, the similar soft-margin idea is adopted by adding a slack vector $\boldsymbol{\zeta} = [\zeta_1, \ldots, \zeta_N]^T$. Denote the DWP surface by $\mathcal{S}_D$ as the following:

$$\mathcal{S}_D \triangleq \{\boldsymbol{x} \in \mathbb{R}^n | F(\boldsymbol{x}) = 0\}. \tag{17}$$

where F is a DWP function defined in (5). Notice that F has a quadratic form $\tilde{F}$ as defined in (7). Denote $\mathbf{W} = \boldsymbol{w}_\xi\boldsymbol{w}_\xi^T$ and use a similar idea of GQSVM, we apply G-margin on $\mathcal{S}_D$ and propose the following DWPSVM model with a soft margin:

$$\begin{aligned} \min \quad & \frac{1}{2}\left\|\begin{bmatrix}\mathbf{W} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}\end{bmatrix}\right\|_F^2 + \frac{1}{2}\|\boldsymbol{b}\|_2^2 + C\sum_{i=1}^N \zeta_i \\ \text{s.t.} \quad & y^{(i)}\left(\frac{1}{2}\boldsymbol{z}^{(i)^T}\mathbf{W}\boldsymbol{z}^{(i)} + \frac{1}{2}\boldsymbol{x}^{(i)^T}\mathbf{A}\boldsymbol{x}^{(i)} + \boldsymbol{b}^T\boldsymbol{x}^{(i)} + q\right) \\ & \geqslant 1 - \zeta_i, \quad i = 1, \ldots, N \\ & \mathrm{rank}(\mathbf{W}) = 1 \\ & \mathbf{W} \in \mathbb{S}^l, \mathbf{A} \in \mathbb{S}^n, \boldsymbol{b} \in \mathbb{R}^n, q \in \mathbb{R}, \boldsymbol{\zeta} \in \mathbb{R}_+^N. \end{aligned} \tag{DWPSVM}$$

The rank-1 constraint is non-convex, which makes problem (DWPSVM) difficult to be implemented. To make it computationally solvable, (DWPSVM) is relaxed to be (DWPSVM-relaxed) by dropping the rank-1 constraint as below. In literature, dropping the non-convex rank-1 constraint to make a hard problem easy to solve is a common practice. For example, in Luo, Ma, So, Ye, and Zhang (2010), the rank-1 constraint is dropped to convexify the problem during the process of semidefinite relaxation.

$$\begin{aligned} \min \quad & \frac{1}{2}\left\|\begin{bmatrix}\mathbf{W} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}\end{bmatrix}\right\|_F^2 + \frac{1}{2}\|\boldsymbol{b}\|_2^2 + C\sum_{i=1}^N \zeta_i \\ \text{s.t.} \quad & y^{(i)}\left(\frac{1}{2}\boldsymbol{z}^{(i)^T}\mathbf{W}\boldsymbol{z}^{(i)} + \frac{1}{2}\boldsymbol{x}^{(i)^T}\mathbf{A}\boldsymbol{x}^{(i)} + \boldsymbol{b}^T\boldsymbol{x}^{(i)} + q\right) \\ & \geqslant 1 - \zeta_i, \quad i = 1, \ldots, N \\ & \mathbf{W} \in \mathbb{S}^l, \mathbf{A} \in \mathbb{S}^n, \boldsymbol{b} \in \mathbb{R}^n, q \in \mathbb{R}, \boldsymbol{\zeta} \in \mathbb{R}_+^N. \end{aligned} \tag{DWPSVM-relaxed}$$

Besides, with definitions of (1) and (8), problem (DWPSVM-relaxed) can be reformulated as an equivalent convex QP problem (DWPSVM'):

$$\begin{aligned}\min\quad & \tfrac{1}{2}\boldsymbol{w}_W^T\mathbf{H}_l\boldsymbol{w}_W+\tfrac{1}{2}\boldsymbol{w}_A^T\mathbf{H}_n\boldsymbol{w}_A+\tfrac{1}{2}\|\boldsymbol{b}\|_2^2+C\sum_{i=1}^{N}\zeta_i\\ \text{s.t.}\quad & y^{(i)}\left(\boldsymbol{\eta}^{(i)^T}\boldsymbol{w}_W+\boldsymbol{s}^{(i)^T}\boldsymbol{w}_A+\boldsymbol{x}^{(i)^T}\boldsymbol{b}+q\right)\geqslant 1-\zeta_i,\quad i=1,\ldots,N\\ & \boldsymbol{w}_W\in\mathbb{R}^{\frac{l(l+1)}{2}},\boldsymbol{w}_A\in\mathbb{R}^{\frac{n(n+1)}{2}},\boldsymbol{b}\in\mathbb{R}^n,q\in\mathbb{R},\boldsymbol{\zeta}\in\mathbb{R}_+^N.\end{aligned}\tag{DWPSVM'}$$

4. Theoretical properties of DWPSVM

In this section we study some theoretical properties of the proposed DWPSVM model. For the convenience of analysis, we reformulate problem (DWPSVM') as the following equivalent problem (DWPSVM") by utilizing the definitions of (8) and (9):

$$\begin{aligned}\min\quad & \tfrac{1}{2}\boldsymbol{v}^T\mathbf{H}\boldsymbol{v}+\tfrac{1}{2}\|\boldsymbol{b}\|_2^2+C\sum_{i=1}^{N}\zeta_i\\ \text{s.t.}\quad & y^{(i)}\left(\boldsymbol{r}^{(i)^T}\boldsymbol{v}+\boldsymbol{x}^{(i)^T}\boldsymbol{b}+q\right)\geqslant 1-\zeta_i,\quad i=1,\ldots,N\\ & \boldsymbol{v}\in\mathbb{R}^{\frac{l(l+1)+n(n+1)}{2}},\boldsymbol{b}\in\mathbb{R}^n,q\in\mathbb{R},\boldsymbol{\zeta}\in\mathbb{R}_+^N.\end{aligned}\tag{DWPSVM"}$$

where $\mathbf{H}:=\begin{bmatrix}\mathbf{H}_l & \mathbf{0}\\ \mathbf{0} & \mathbf{H}_n\end{bmatrix}$, with $\mathbf{H}_l$ and $\mathbf{H}_n$ being defined according to (1). It is clear that $\mathbf{H}\succ 0$ so that problem (DWPSVM") is a convex quadratic program (QP). Given a data set $\mathcal{D}$ denoted as (2), any tuple $(\boldsymbol{v},\boldsymbol{b},q,\boldsymbol{\zeta})$ satisfying

$$\boldsymbol{v}\in\mathbb{R}^{\frac{l(l+1)+n(n+1)}{2}},\quad \boldsymbol{b}\in\mathbb{R}^n,\quad q\in\mathbb{R},\quad \zeta_i=\max\left\{0,1-y^{(i)}\left(\boldsymbol{r}^{(i)^T}\boldsymbol{v}+\boldsymbol{x}^{(i)^T}\boldsymbol{b}+q\right)\right\},\quad \forall i=1,\ldots,N.$$

is a feasible solution. Moreover, the objective function is bounded below by zero. Thus, problem (DWPSVM") achieves a finite optimum for any given data set $\mathcal{D}$ denoted as (2).

Next, we study the uniqueness of the optimal solution to problem (DWPSVM") with respect to $\boldsymbol{v}$ and $\boldsymbol{b}$.

Theorem 4.1. ***$(\boldsymbol{v}^*,\boldsymbol{b}^*)$-uniqueness*** *For any given data set $\mathcal{D}$ denoted by (2), let $(\boldsymbol{v}^*,\boldsymbol{b}^*,q^*,\boldsymbol{\zeta}^*)$ be an optimal solution to problem (DWPSVM"), then $(\boldsymbol{v}^*,\boldsymbol{b}^*)$ is unique.*

Proof. The proof is in A.2. □

Remark. The uniqueness of $(\boldsymbol{v}^*,\boldsymbol{b}^*)$ plays an important role in characterizing the pattern of the classifier, since various coefficients of non-constant terms of a polynomial give rise to various families of quartic surfaces. Even though the optimal solution may not be unique, different optimal solutions only vary by intercepts. Thus, for any given data set, the main characteristics of the separating DWP surface are uniquely determined by the optimal solution of the problem (DWPSVM") with respect to the variable $\boldsymbol{v}$ and $\boldsymbol{b}$.

Similar to other SVM models, DWPSVM has the property that its optimal solution is highly related to the support vectors. According to Vapnik (2013), the support vectors produced by model (SSVM) are the data points for which in inequality constraints equalities are achieved. Similarly, we bring the definition of support vectors of model (DWPSVM") as follows.

Definition 4.1 (Support Vector of DWPSVM")**.** Given a data set $\mathcal{D}$ denoted as (2) and assume that $\{\boldsymbol{x}\in\mathbb{R}^n|g(\boldsymbol{x})=0\}$ is the quartic surface that obtained by solving problem (DWPSVM"). Then $\boldsymbol{x}^{(k)}$ is called a support vector if $g(\boldsymbol{x}^{(k)})=y^{(k)}$ for any $(\boldsymbol{x}^{(k)},y^{(k)})\in\mathcal{D}$.

To study the property of support vectors representation, we write the dual of problem (DWPSVM")

$$\begin{aligned}\min\quad & -\sum_{i=1}^{N}\alpha_i+\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}y^{(i)}y^{(j)}\alpha_i\alpha_j\left(\boldsymbol{r}^{(i)^T}\mathbf{H}^{-1}\boldsymbol{r}^{(j)}+\boldsymbol{x}^{(i)^T}\boldsymbol{x}^{(j)}\right)\\ s.t.\quad & \sum_{i=1}^{N}y^{(i)}\alpha_i=0.\\ & 0\leqslant\alpha_i\leqslant C,\quad i=1,\ldots,N.\end{aligned}\tag{DDWPSVM"}$$

Since both of problem (DWPSVM") and problem (DDWPSVM") are convex QPs, no duality gap exists between them. The KKT optimality conditions for them are listed as the following:

$$\begin{aligned}&\alpha_i^*\left(1-\zeta_i^*-y^{(i)}\left(\boldsymbol{r}^{(i)^T}\boldsymbol{v}^*+\boldsymbol{x}^{(i)^T}\boldsymbol{b}^*+q^*\right)\right)=0,\quad i=1,\ldots,N\\ &(C-\alpha_i^*)\zeta_i^*=0,\quad i=1,\ldots,N\\ &1-\zeta_i^*-y^{(i)}\left(\boldsymbol{r}^{(i)^T}\boldsymbol{v}^*+\boldsymbol{x}^{(i)^T}\boldsymbol{b}^*+q^*\right)\leqslant 0,\quad i=1,\ldots,N\\ &\sum_{i=1}^{N}y^{(i)}\alpha_i^*=0,\quad \boldsymbol{\zeta}^*\in\mathbb{R}_+^N,\quad 0\leqslant\alpha_i^*\leqslant C,\quad i=1,\ldots,N.\end{aligned}\tag{18}$$

Recall that the index set for points in data set $\mathcal{D}$ is $\mathcal{M}=\{1,\ldots,N\}$, which can be split into the following four subsets:

$$\begin{aligned}&I_1=\left\{i\in\mathcal{M}|\alpha_i^*=0\right\},\quad I_2=\left\{i\in\mathcal{M}|0<\alpha_i^*<C\right\}\\ &I_3=\left\{i\in\mathcal{M}|\alpha_i^*=C,\ \zeta_i^*=0\right\},\quad I_4=\left\{i\in\mathcal{M}|\alpha_i^*=C,\ \zeta_i^*>0\right\}.\end{aligned}\tag{19}$$

It could be observed from (18) and (19) that

- If $i\in I_1$, then $\zeta_i^*=0$ and the ith data point satisfies $y^{(i)}(\boldsymbol{r}^{(i)^T}\boldsymbol{v}^*+\boldsymbol{x}^{(i)^T}\boldsymbol{b}^*+q^*)\geqslant 1$, which indicates that point $\boldsymbol{x}^{(i)}$ is inside the scope of the class.
- If $i\in I_2\cup I_3$, then $\zeta_i^*=0$ and $y^{(i)}(\boldsymbol{r}^{(i)^T}\boldsymbol{v}^*+\boldsymbol{x}^{(i)^T}\boldsymbol{b}^*+q^*)=1$. From Definition 4.1, $\boldsymbol{x}^{(i)}$ is a support vector.
- If $i\in I_4$, then $\zeta_i^*>0$ and $y^{(i)}(\boldsymbol{r}^{(i)^T}\boldsymbol{v}^*+\boldsymbol{x}^{(i)^T}\boldsymbol{b}^*+q^*)<1$. It indicates that $\boldsymbol{x}^{(i)}$ might be a misclassified data sample.

By Lagrangian duality theory, solving the optimality condition yields an relationship between primal and dual optimal solutions as the following:

$$\begin{aligned}&\boldsymbol{v}^*=\mathbf{H}^{-1}\sum_{i\in I_2\cup I_3\cup I_4}y^{(i)}\alpha_i^*\boldsymbol{r}^{(i)},\quad \boldsymbol{b}^*=\sum_{i\in I_2\cup I_3\cup I_4}y^{(i)}\alpha_i^*\boldsymbol{x}^{(i)}\\ &q^*=y^{(i)}-\boldsymbol{r}^{(i)^T}\boldsymbol{v}^*-\boldsymbol{x}^{(i)^T}\boldsymbol{b}^*,\quad i\in I_2\cup I_3.\end{aligned}\tag{20}$$

Remark. From (20), we notice that the primal optimal solution is only related to data points with indices in I_2, I_3 and I_4, which include most of the support vectors. It indicates that a new data point $(\boldsymbol{x},y)$ will not change the separation surface if it satisfies $yg(\boldsymbol{x})>1$. In other words, $(\boldsymbol{x},y)$ is not only correctly classified, but outside the margin area as well.

Therefore, it is not necessary to re-train the DWPSVM model every time after a new data point enters. A precise separation surface could last for a long time before re-training the DWPSVM model.

Since both the primal problem (DWPSVM") and the dual problem (DDWPSVM") are convex QP problems, they can be solved by applying QP solvers. Notice that, problem (DDWPSVM") has only one equality constraint with bounded variables, which is similar to

the structure of DSSVM-kernel. The SMO algorithm can be applied to solve problem (DDWPSVM") after preprocessing the data as follows.

Given a data set $\mathcal{D}$ as defined in (2), generate a data set as

$$\hat{\mathcal{D}} \triangleq \left\{ \left(\boldsymbol{t}^{(i)}, y^{(i)} \right) \middle| \boldsymbol{t}^{(i)} = \begin{bmatrix} \mathbf{H}^{-\frac{1}{2}} \boldsymbol{r}^{(i)} \\ \boldsymbol{x}^{(i)} \end{bmatrix}, (\boldsymbol{x}^{(i)}, y^{(i)}) \in \mathcal{D}, \ i = 1, \ldots, N \right\} \tag{21}$$

where $\boldsymbol{r}^{(i)}$ is defined as (8). Then problem (DDWPSVM") can be reformulated as the following problem (DDWPSVM"-SMO) to be solved by directly applying the SMO algorithm:

$$\begin{aligned} \min \quad & \tfrac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} y^{(i)} y^{(j)} \alpha_i \alpha_j \boldsymbol{t}^{(i)^T} \boldsymbol{t}^{(j)} - \sum_{i=1}^{N} \alpha_i \\ s.t. \quad & \sum_{i=1}^{N} y^{(i)} \alpha_i = 0, \\ & 0 \leqslant \alpha_i \leqslant C, \quad i = 1, \ldots, N. \end{aligned} \tag{DDWPSVM$''$ – SMO}$$

Denote the optimal solution of (DDWPSVM"-SMO) as $\boldsymbol{\alpha}^*$, and its corresponding primal optimal solution as $(\hat{\boldsymbol{v}}^*, \hat{\boldsymbol{b}}^*, \hat{q}^*, \hat{\boldsymbol{\zeta}}^*)$, then the optimal solution $(\boldsymbol{v}^*, \boldsymbol{b}^*, q^*, \boldsymbol{\zeta}^*)$ to problem (DWPSVM") is

$$\boldsymbol{v}^* = \mathbf{H}^{-\frac{1}{2}} \hat{\boldsymbol{v}}^*, \quad \boldsymbol{b}^* = \hat{\boldsymbol{b}}^*, \quad q^* = \hat{q}^*, \quad \boldsymbol{\zeta}^* = \hat{\boldsymbol{\zeta}}^*. \tag{22}$$

The convergence of the SMO algorithm is promised by Osuna's Theorem Osuna, Freund, and Girosi (1997) and the computational efficiency of the SMO algorithm on solving the soft-margin SVM model is investigated in Platt (1998). The efficiency of the SMO algorithm is tested by computational experiments in Section 5.

5. Computational experiments

In this section, we conduct computational experiments to investigate the performance of DWPSVM on some artificial, public benchmark and real-life data sets. We first introduce the settings of our experiments, then the proposed DWPSVM model and well-known SVM models are tested with some artificial and public benchmark data sets. Finally, the proposed DWPSVM model is extended and applied to some benchmark and real-life credit data sets, by comparing to well-known SVM models.

5.1. *Experiment settings*

The linear SVM model, SQSSVM model, QLSTSVM model, QLSSVM model and SVM models with Gaussian or Quadratic kernel are tested on the same data sets for comparisons with the proposed DWPSVM model. All numerical experiments are conducted on a computer with eight Intel(R) Core(TM) i7-2600 CPU @ 3.40gigahertz CPUs and 8gigabyte RAM. Moreover, Gurobi 8.1.1 and LIBSVM Chang and Lin (2011) are utilized to implement some of tested models, as shown in Table 1. Since QLSTSVM model and QLSSVM model have explicit solutions, no solver or package will be needed. Notice that, the DDWPSVM"-SMO model is implemented for DWPSVM by utilizing the SMO algorithm in LIBSVM package. Throughout all tables and figures of results in this paper, each model is denoted by its abbreviation name, as shown in Table 1.

All the data points are normalized into [0,1] to avoid the dominance of input features with greater numerical values over other smaller values. A 10-fold cross-validation procedure is applied for each experiment and each experiment is repeated ten times for each model to make it statistically meaningful.

All the possible parameters for each model listed in Table 1 are tuned by using grid method, such as $\log_2 C \in \{-6, -3, \ldots, 21, 22\}$, $\log_2 r \in \{-4, -3, \ldots, 3, 4\}$, $\log_2 \gamma \in \{-4, -3, \ldots, 3, 4\}$ and $\log_2 C_i \in \{-4, -3, \ldots, 3, 4\}$ $(i = 1, 2)$.

Table 1
Abbreviations and solvers of tested models.

Model	Abbreviation	Solver/Package	Parameters
Linear SVM	SVM	LIBSVM	C
Soft margin quadratic surface SVM	SQSSVM	Gurobi	C
SVM with Gaussian kernel	SVM-Gauss	LIBSVM	(C, γ)
SVM with quadratic kernel	SVM-Quad	LIBSVM	(C, r)
Double Well Potential SVM	DWPSVM	LIBSVM	C
Quadratic least square twin SVM	QLSTSVM	–	(C_1, C_2)
Quadratic least square SVM	QLSSVM	–	C

For all experiments on each data set, the mean and the standard deviation of accuracy scores, the average training CPU time of all models and the CPU time for testing each data point are recorded. Notice that, the SVM, SVM-Gauss and SVM-Quad are implemented by utilizing LIBSVM Chang and Lin (2011) python package, while the scripts of DWPSVM, SQSSVM, QLSTSVM and QLSSVM are written from scratch.

5.2. *Tests on artificial data*

In order to show and compare the flexibility of separation surfaces produced by the DWPSVM model and other benchmark SVM models, Fig. 3a to 3h are displayed. Each data set is plotted and separated by using different separation surfaces. Besides DWPSVM, the data in Fig. 3a, 3c, 3e, 3g and 3i is classified by the other four kernel-free benchmark SVM models (SQSSVM, QLSTSVM, QLSSVM and SVM), and the data in other figures is classified by the two benchmark kernel-based SVM models (SVM-Gauss and SVM-Quad).

From these figures, we have the following observations: SVM only works well when the data is linearly separable, as in Fig. 3a and 3b. SQSSVM, QLSTSVM, QLSSVM and SVM-Quad work well for both linearly and quadratically separable data sets, as in Fig. 3c and 3d, but their performance are not satisfying when the data sets have highly nonlinear patterns. In Fig. 3e to 3h, both of DWPSVM and SVM-Gauss show the flexibility to capture the highly nonlinear classifiers, but the performance of DWPSVM is better than that of SVM-Gauss. In summary, these figures indicate that the DWPSVM model, compared with other benchmark SVM models, has stronger potential and flexibility to classify highly nonlinearly-separable data sets.

Since the DWP separation surface produced by the proposed DWPSVM model is a special type of degree-4 polynomial surface, it may also be produced by the SVM model with a high-degree polynomial kernel,(e.g. the 4th order polynomial kernel). However, compared with DWPSVM, SVM with degree-4 polynomial kernel (SVM-Q4) has two more parameters to be tuned during the training process. Hence, the SVM-Q4 model may not be as practical as the DWPSVM model, because it takes much more effort to train SVM-Q4 than that to train DWPSVM.

To further compare the performance of DWPSVM to that of other SVM models, we generated some artificial data sets, including the mexican hat (MH) data, the wave data and the multi-petal data. For MH data, two classes of data points are generated respectively on each side of a MH-shaped surface on $\mathbb{R}^k$ $(k = 2, 3, 4, 5)$. An example of MH data in $\mathbb{R}^2$ is shown in Fig. 3e and 3f. Similarly, for the wave data, the data points are generated on the two sides of a sine curve in $\mathbb{R}^2$. In addition, we generate a six-petal data set in $\mathbb{R}^2$ and a eight-petal data set in $\mathbb{R}^3$. An example of the six-petal data set is shown in Fig. 3g and 3h. The description of all artificial data sets can be found in Table 2. For each data set, n is the num-

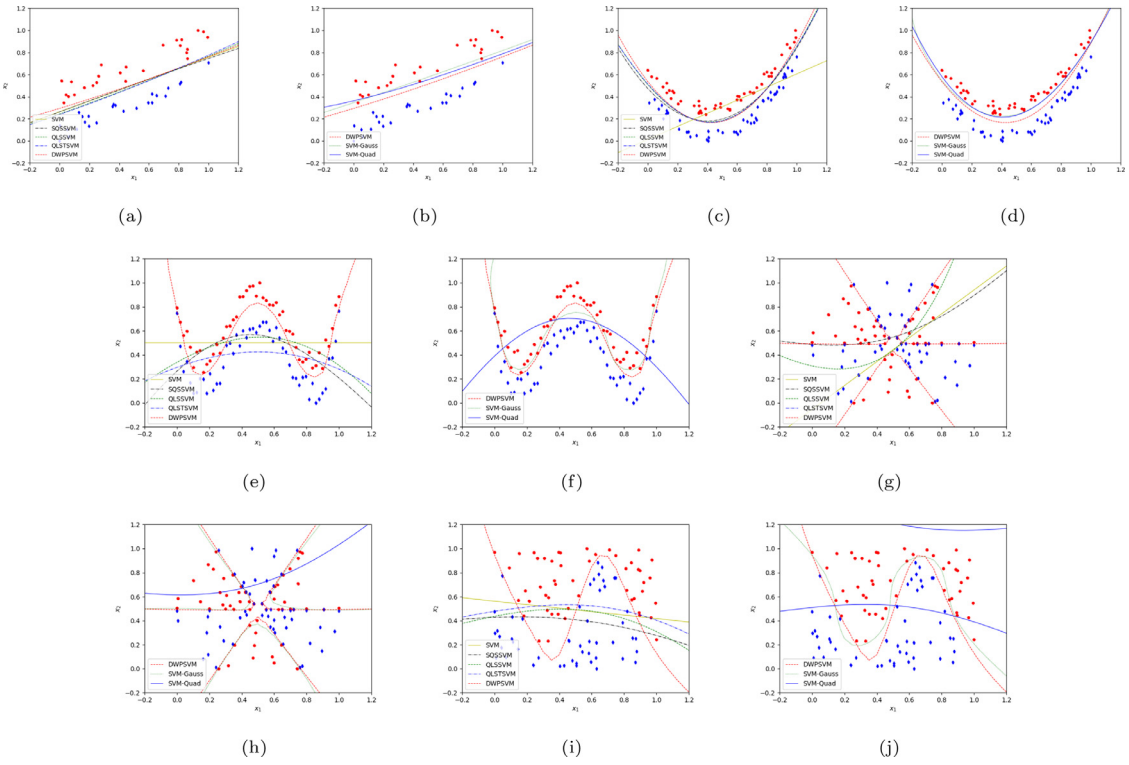


Fig. 3. DWPSVM vs kernel and kernel-free SVMs.

Table 2
Artificial data sets.

Data set	MH2d	MH3d	MH4d	MH5d	Six-petal 2d	Eight-petal 3d	Wave 2d
n	2	3	4	5	2	3	2
Sample size (N_1 vs N_2)	45 vs 45	100 vs 100	200 vs 200	450 vs 450	55 vs 55	120 vs 130	49 vs 51

Table 3
MH2d, MH3d, MH4d and MH5d results.

Model	Accuracy score %															
	MH2d				MH3d				MH4d				MH5d			
	mean	std	min	max	mean	std	min	max	mean	std	min	max	mean	std	min	max
DWPSVM	92.81	8.44	75.00	100.00	91.45	6.49	70.00	100.00	91.85	2.94	80.00	100.00	91.18	2.75	81.11	96.67
SQSSVM	44.38	13.85	25.00	75.00	47.00	11.12	20.00	75.00	51.15	8.35	25.00	70.00	51.72	4.37	43.33	57.78
SVM	47.19	15.63	12.50	87.50	44.00	8.04	25.00	60.00	50.65	7.67	32.50	70.00	51.75	3.65	44.44	61.11
SVM-Gauss	86.88	11.65	62.50	100.00	84.70	8.28	60.00	100.00	81.60	6.36	67.50	97.50	79.94	4.61	70.00	87.78
SVM-Quad	45.31	13.48	12.50	75.00	45.10	9.04	20.00	65.00	49.45	8.24	32.50	72.50	51.28	4.76	42.22	65.56
QLSTSVM	50.94	8.67	25.00	75.00	49.40	6.90	25.00	60.00	50.75	6.59	35.00	67.50	52.25	5.06	42.22	62.22
QLSSVM	47.50	13.03	12.50	75.00	46.00	11.01	20.00	75.00	49.45	7.32	25.00	70.00	53.00	5.14	43.33	68.89

ber of features. The number of data points in class 1 and class 2 are denoted as N_1 and N_2, respectively.

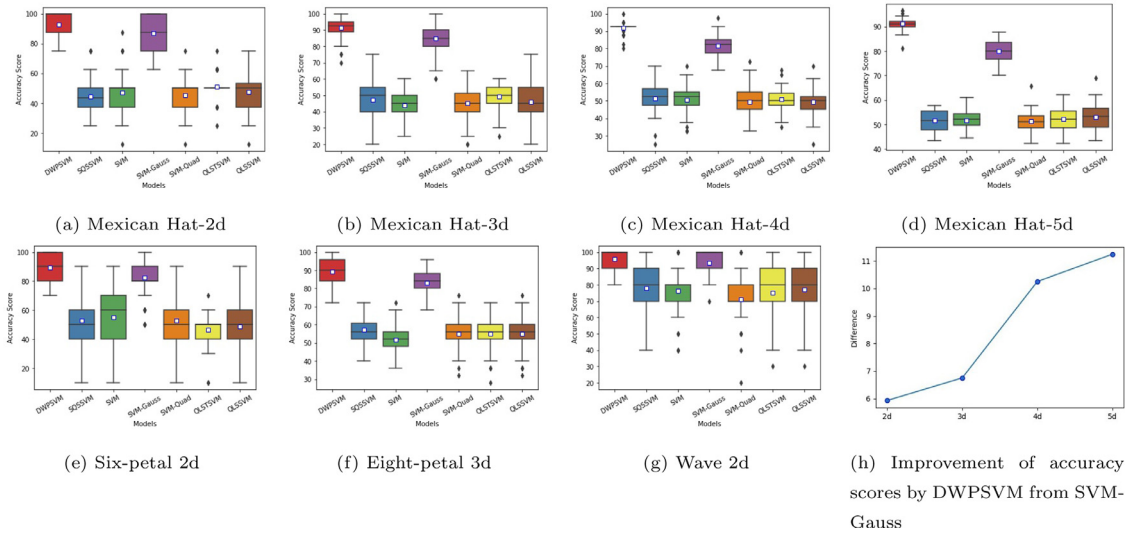
The mean and standard deviation of accuracy scores are shown in Tables 3 and 4 and the corresponding box plots are shown in Fig. 4. Besides, the training CPU time of all tested models is listed in Table 5. The testing CPU time on each data point is listed in Table 12.

From Tables 3 and 5 and Fig. 4, there are a few observations as the following:

- Comparing with all other tested SVM models, the proposed DWPSVM shows the dominant and the stabler performance on each artificial data set in terms of classification accuracy. Moreover, since all artificial data sets are highly nonlinearly-separable, the results from the experiments verify the flexibility of separation surfaces produced by the proposed DWPSVM model.
- All the SVM models are tested on mexican hat data sets in different dimensional Euclidean spaces. For these data sets, DWPSVM produces much more accurate classification than all other tested benchmark models. Moreover, Fig. 4h shows that, the accuracy improvement of DWPSVM over the second best model SVM-Gauss, is increasing as the dimension of data set increases. It suggests that, in terms of classification accuracy, the DWPSVM model is a better choice when the data set has more features.
- Table 5 shows that it takes both of the proposed DWPSVM model and SVM-Gauss more CPU time to generate highly nonlinear separation surfaces. Although the CPU time of DWPSVM is longer than that of SVM-Gauss, the accuracy advantage of DWPSVM over SVM-Gauss is obvious, which makes the sacrifice in efficiency of the proposed DWPSVM model acceptable.

Table 4
Six-petal 2d, Eight-petal 3d and Wave 2d results.

Model	Accuracy score %											
	Six-petal 2d				Eight-petal 3d				Wave 2d			
	mean	std	min	max	mean	std	min	max	mean	std	min	max
DWPSVM	89.00	10.01	70.00	100.00	89.20	6.47	72.00	100.00	95.75	6.36	80.00	100.00
SQSSVM	52.50	15.79	10.00	80.00	57.35	7.03	40.00	72.00	77.75	14.05	40.00	100.00
SVM	54.88	16.84	10.00	90.00	51.85	7.78	36.00	72.00	76.25	14.44	40.00	100.00
SVM-Gauss	82.63	11.99	50.00	100.00	82.95	6.36	68.00	96.00	93.25	8.59	70.00	100.00
SVM-Quad	53.25	14.91	10.00	90.00	55.05	7.54	32.00	76.00	71.25	15.22	20.00	100.00
QLSTSVM	46.50	10.08	10.00	70.00	54.95	7.84	28.00	72.00	75.00	15.53	30.00	100.00
QLSSVM	49.00	15.48	10.00	90.00	55.20	8.57	32.00	76.00	77.25	15.02	30.00	100.00



(a) Mexican Hat-2d (b) Mexican Hat-3d (c) Mexican Hat-4d (d) Mexican Hat-5d

(e) Six-petal 2d (f) Eight-petal 3d (g) Wave 2d (h) Improvement of accuracy scores by DWPSVM from SVM-Gauss

Fig. 4. Results on artificial data sets.

Table 5
Artificial data training CPU time.

Model	Training CPU time (s)						
	MH2d	MH3d	MH4d	MH5d	Six-petal 2d	Eight-petal 3d	Wave 2d
DWPSVM	2.721	4.864	17.633	39.010	1.493	3.040	0.034
SQSSVM	0.041	0.085	0.257	0.589	0.041	0.127	0.038
SVM	<0.001	<0.001	0.002	0.007	<0.001	<0.001	<0.001
SVM-Gauss	1.210	3.361	12.975	26.307	0.499	1.148	0.019
SVM-Quad	<0.001	0.002	0.008	0.047	<0.001	0.035	<0.001
QLSTSVM	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001
QLSSVM	0.041	0.124	0.327	0.941	0.050	0.156	0.047

Table 6
Public benchmark data sets.

Small-sized			Medium-sized			Large-sized		
data set	n	sample size (N_1 vs N_2)	data set	n	sample size (N_1 vs N_2)	data set	n	sample size (N_1 vs N_2)
Wine	13	71 vs 59	SVMguide3	20	947 vs 296	Cardiotocography	21	1655 vs 471
Glass	9	163 vs 51	Brain Tumor	17	1178 vs 97	Abalone	8	1407 vs 1323
Liver Disorders	6	199 vs 142	Car Evaluation	6	1210 vs 384	SVMguide1	4	2000 vs 1089
Wholesale	7	298 vs 142						
Tax Payer	9	336 vs 331						

5.3. *Tests on public benchmark data*

Besides the experiments on artificial data sets, the proposed DWPSVM model is applied to some public benchmark data sets (grouped by size). Some basic information of these benchmark data sets is listed in Table 6. Similar to that in Section 5.2, all the models are respectively tested on each benchmark data set and the numerical results are listed inTables 7–10. For each tested SVM model, the training CPU time and testing CPU time are listed in Tables 11 and 12, respectively. We also recorded the median of optimal parameters of corresponding models on each benchmark data set, which can be found in Appendix B.

Table 7
Wine, glass, liver disorders results.

Model	Accuracy score %											
	Wine				Glass				Liver Disorders			
	mean	std	min	max	mean	std	min	max	mean	std	min	max
DWPSVM	98.33	3.70	83.33	100.00	93.38	4.97	76.19	100.00	72.90	7.07	57.58	90.91
SQSSVM	97.64	4.63	83.33	100.00	92.90	5.35	76.19	100.00	71.28	6.95	54.55	90.91
SVM	97.36	4.20	83.33	100.00	92.81	5.31	80.95	100.00	69.60	7.25	42.42	84.85
SVM-Gauss	97.64	4.63	83.33	100.00	92.43	5.11	76.19	100.00	72.59	7.05	54.55	87.88
SVM-Quad	97.36	4.47	83.33	100.00	92.33	5.15	80.95	100.00	72.05	6.76	54.55	87.88
QLSTSVM	90.69	5.96	75.00	100.00	91.05	5.55	76.19	100.00	72.42	6.63	60.61	90.91
QLSSVM	96.94	5.08	83.33	100.00	92.86	5.00	80.95	100.00	72.46	6.87	57.58	87.88

Table 8
Wholesale, tax payer results.

Model	Accuracy score %							
	Wholesale				Tax Payer			
	mean	std	min	max	mean	std	min	max
DWPSVM	91.93	3.63	79.07	100.00	54.45	4.35	40.91	62.12
SQSSVM	90.42	3.97	81.40	100.00	50.47	4.48	39.39	63.64
SVM	91.40	3.82	79.07	97.67	50.94	4.77	39.39	63.64
SVM-Gauss	91.60	3.85	81.40	100.00	50.36	5.44	39.39	63.64
SVM-Quad	91.26	4.02	79.07	100.00	52.70	5.02	40.91	66.67
QLSTSVM	90.79	3.66	81.40	97.67	50.42	5.08	34.85	60.61
QLSSVM	89.65	4.34	79.07	100.00	51.17	5.81	36.36	68.18

Table 9
SVMguide3, brain tumor and car evaluation results.

Model	Accuracy score %											
	SVMguide3				Brain Tumor				Car Evaluation			
	mean	std	min	max	mean	std	min	max	mean	std	min	max
DWPSVM	84.82	3.13	78.05	92.68	97.56	1.40	92.06	100.00	98.63	0.87	96.86	100.00
SQSSVM	84.17	3.42	76.42	91.06	93.67	1.29	90.48	96.83	96.62	1.48	92.45	100.00
SVM	82.89	3.00	75.61	88.62	97.33	1.30	93.65	100.00	85.94	2.64	77.36	91.82
SVM-Gauss	84.55	3.12	76.42	91.06	96.06	1.84	91.27	99.21	98.52	0.87	96.23	100.00
SVM-Quad	84.23	3.05	77.24	91.06	95.93	2.08	90.48	99.21	93.46	1.86	88.68	98.74
QLSTSVM	82.70	2.82	75.61	87.80	96.58	1.77	91.27	100.00	94.39	1.80	90.57	98.11
QLSSVM	83.13	3.17	76.42	89.43	96.60	1.82	92.86	100.00	94.40	1.79	89.94	98.11

Table 10
Cardiotocography, abalone, SVMguide1 results.

Model	Accuracy score %											
	Cardiotocography				Abalone				SVMguide1			
	mean	std	min	max	mean	std	min	max	mean	std	min	max
DWPSVM	94.17	1.63	90.57	97.17	79.08	2.74	72.79	85.66	96.85	0.90	94.16	98.70
SQSSVM	92.78	2.06	88.68	96.23	77.89	3.08	69.49	84.56	96.64	0.96	93.51	99.03
SVM	90.59	2.13	84.91	93.87	77.73	3.14	68.38	85.29	95.39	1.03	93.18	97.73
SVM-Gauss	93.99	1.38	91.51	97.64	78.68	2.79	71.69	84.56	96.69	0.90	93.83	99.03
SVM-Quad	93.09	2.00	89.15	96.70	78.46	3.00	70.22	83.46	96.44	0.92	94.16	98.38
QLSTSVM	92.64	1.75	88.68	95.75	78.62	2.83	70.96	84.56	95.11	1.06	92.53	97.73
QLSSVM	93.02	1.66	89.15	95.28	78.61	2.81	70.96	85.66	94.77	1.03	91.23	97.08

Table 11
Benchmark data training CPU time.

Model	Small-sized					Medium-sized			Large-sized		
	Wine (13, 130)	Glass (9, 214)	Liver Disorders (6, 341)	Wholesale (7, 440)	Tax Payer (9, 667)	SVMguide3 (20, 1243)	Brain Tumor (17, 1275)	Car Evaluation (6, 1594)	Cardioto- cography (21, 2126)	Abalone (8, 2730)	SVMguide1 (4, 3089)
DWPSVM	0.036	0.010	0.040	0.035	0.583	19.991	4.535	0.190	18.270	7.165	0.192
SQSSVM	0.021	0.024	0.332	0.014	0.768	9.108	10.636	0.045	9.342	3.926	0.115
SVM	<0.001	<0.001	0.004	0.001	0.011	2.423	0.014	0.022	0.053	0.207	0.031
SVM-Gauss	<0.001	<0.001	0.006	0.001	0.014	0.456	0.013	0.034	0.228	0.687	0.083
SVM-Quad	<0.001	<0.001	0.018	0.004	0.010	0.165	0.009	0.065	0.069	2.053	0.047
QLSTSVM	0.007	0.002	<0.001	<0.001	0.002	0.031	0.018	0.001	0.021	0.001	0.001
QLSSVM	0.631	0.535	0.327	0.588	1.247	15.361	6.142	1.497	12.721	4.840	2.152

Table 12
Testing CPU time (both artificial and public benchmark data sets.).

	DWPSVM	SQSSVM	SVM	SVM-Gauss	SVM-Quad	QLSTSVM	QLSSVM
Testing CPU time	10^{-4}	10^{-5}	$< 10^{-5}$	$< 10^{-5}$	$< 10^{-5}$	$< 10^{-5}$	10^{-5}

From Tables 7–12, the following observations can be summarized:

- The proposed DWPSVM model produces more accurate classification than other tested SVM models on all tested benchmark data sets. It shows that the proposed DWPSVM model is able to classify the data sets in different sizes properly. In addition, the DWPSVM also produces the smallest standard deviation on most of the tested benchmark data sets, which implies that the proposed DWPSVM model is more stable than other tested models for binary classification.
- For some data sets (such as the Brain Tumor data), the second most accurate model is linear SVM; For some data sets (such as Wine, Glass and Tax Payer data), the second most accurate tested models produce the quadratic separation surfaces; For all other tested benchmark data sets, the second most accurate model is SVM-Gauss, which produces nonlinear classifiers. Nevertheless, they are all outperformed by DWPSVM, which indicates that the DWPSVM model may classify the data sets with different shapes of separation surfaces.
- From the training CPU time recorded in Table 11, the training efficiency of the proposed DWPSVM on small-sized data sets is satisfying, as the CPU time of DWPSVM is at least in the same order as that of SQSSVM. For medium-sized and large-sized data sets, even though the training CPU time of DWPSVM is longer than that of other tested models, it is still acceptable. Compared with other tested SVM model, the proposed DWPSVM model is only 1–2 orders of magnitude slower for most medium-sized and large-sized data sets. Moreover, for all tested benchmark data sets including large-sized ones, the CPU time of DWPSVM is still less than twenty seconds, and the mean accuracy score of DWPSVM is higher than that of the second most accurate tested SVM model by the value between 0.2% and 1.75% for most benchmark data sets. Although this accuracy advantage by DWPSVM is not very obvious, it may be competitive and valuable to some practical applications.
- In practical applications, the training process is usually completed before forecasting. The computational time (i.e. the testing CPU time) of forecasting procedure by using DWPSVM model is as short as that of other tested SVM model.

5.4. *Application to credit scoring*

The financial institutions suffered heavy losses from loan defaults during the financial crisis. To reduce the credit risk, effective credit scoring methods have been utilized by the financial institutions. As the widely-used technique, credit scoring helps the lender make better credit granting decisions. The objective of quantitative credit scoring models is to accurately distinguish the good applicants from applicants with potential loan defaults (Baesens et al., 2003), which can be achieved by binary classification methods. Hence, an accurate classifier is essential in credit scoring.

In this subsection, we first introduce the credit data sets. Then the proposed DWPSVM model is extended and applied to four preprocessed credit data sets, as well as all other tested SVM models.

5.4.1. Two different types of credit data

Two types of credit data sets are utilized. One type includes personal credit information, such as the German credit data (GCD) (Dua & Graff, 2017), the Japanese credit data (JCD) (Dua & Graff, 2017) and the customer credit applications (CCA) data (Quinlan, 1987). The other type includes corporate credit information, such as the Chinese corporate credit data (CCC) (Luo, Yan, & Tian, 2020). The basic information of these data sets is displayed in Table 13. More details about the credit data sets can be found in Appendix B.

5.4.2. Data preprocessing

To reduce the impact of irrelevant features, we adopt two types of feature weighting strategies: the t-test based weighting strategy and the entropy based weighting strategy (Zhou, Lai, & Yen, 2009). Given a data set, the t-test based weight and entropy based weight for feature j is defined by (23) and (24), respectively. Notice that, for the CCA data, there is no real-life meaning of data features from the source, so we convert each category to a unique integer value for the same categorical feature.

$$\bar{w}_j = \frac{|\mu_j^+ - \mu_j^-|}{\sqrt{\frac{\sigma_j^{+2}}{N^+} + \frac{\sigma_j^{-2}}{N^-}}}, \quad w_j = \frac{\bar{w}_j}{\sum_{k=1}^n \bar{w}_k}, \quad j = 1, \ldots, n. \tag{23}$$

$$\bar{w}_j = \frac{\frac{\sigma_j^+}{\sigma_j^-} + \frac{\sigma_j^-}{\sigma_j^+} - 2}{2} + \frac{\left(\mu_j^+ - \mu_j^-\right)^2 \left(\frac{1}{\sigma_j^{+2}} + \frac{1}{\sigma_j^{-2}}\right)}{2},$$
$$w_j = \frac{\bar{w}_j}{\sum_{k=1}^n \bar{w}_k}, \quad j = 1, \ldots, n. \tag{24}$$

where μ_j^+ and σ_j^+ are the mean and the standard deviation of the jth feature of data points with positive labels, respectively; μ_j^- and σ_j^- are the mean and the standard deviation of the those with negative labels, respectively. Recall that N^+ and N^- are the number of data points with positive and negative labels, respectively. Notice that, there are other strategies for handling credit data features including principal component analysis and other feature selection methods in Hajek and Michalak (2013) and Maldonado, Pérez, and Bravo (2017). According to Zhou et al. (2009) and Luo et al. (2020), t-test based feature weighting strategy is more effective for credit scoring than principal component analysis and the entropy-based feature weighting strategy. And in Tsai (2009), the t-test based feature selection strategy is shown to be preferable in the related field of credit score forecasting.

5.4.3. Numerical experiments on credit data sets

As we discussed before, an accurate classification tool is important for credit scoring. The area under the receiver operating characteristics (ROC) curve, denoted as AUC, is a measure that captures the general behavior of a classifier regardless of the classification threshold values (Zhou et al., 2009). It is an alternative measure that helps decision makers to select a proper classification tool. Given a separation surface $\{f(\boldsymbol{x}) = 0 | \boldsymbol{x} \in \mathbb{R}^n\}$, the AUC can be calculated by the following:

$$\text{AUC} = \frac{\sum_{i \in \mathcal{M}^+} \sum_{j \in \mathcal{M}^-} \mathbb{1}_{f(\boldsymbol{x}^{(i)}) > f(\boldsymbol{x}^{(j)})}}{N^+ N^-}. \tag{25}$$

where $\mathbb{1}$ is the indicator function.

In addition to the proposed DWPSVM model and all tested SVM models, the logistic regression (LR) model is also tested on four credit data sets through a similar procedure as described in Section 5.1. The AUC values and the classification accuracy scores are recorded inTables 14–17. Moreover, the training and testing CPU time of tested models is recorded in Table 18.

Table 13
World credit data.

data set	# of features	name of class	sample size
Chinese Corporate Credit (CCC)	6	Good credit/Bad credit	58/48
Customer Credit Applications (CCA)	15	Default/Non-default	296/357
German Credit Data (GCD)	20	Creditworthy/Non-creditworthy	700/300
Japanese Credit Data (JCD)	10	Positive/Negative[a]	84/40

[a] "Positive" indicates the credit was granted and "negative" indicates the credit was not granted.

Table 14
CCC Results.

Model	AUC %								accuracy score %							
	t-test Based				Entropy Based				t-test Based				Entropy Based			
	mean	std	min	max	mean	std	min	max	mean	std	min	max	mean	std	min	max
DWPSVM	96.16	5.98	77.78	100.00	94.81	11.55	50.00	100.00	91.75	8.39	66.67	100.00	90.48	8.52	66.67	100.00
SQSSVM	95.00	9.11	55.00	100.00	94.06	11.23	55.00	100.00	91.43	8.50	66.67	100.00	89.52	8.65	66.67	100.00
SVM	94.31	9.85	55.00	100.00	94.44	8.19	75.00	100.00	90.95	8.52	66.67	100.00	89.52	8.65	66.67	100.00
SVM-Gauss	95.31	8.70	60.00	100.00	93.31	10.73	55.00	100.00	90.95	8.52	66.67	100.00	89.52	8.65	66.67	100.00
SVM-Quad	92.94	11.28	55.00	100.00	93.69	9.62	65.00	100.00	89.52	8.65	66.67	100.00	89.52	8.65	66.67	100.00
QLSTSVM	94.94	9.72	55.00	100.00	93.56	11.24	55.00	100.00	90.48	8.73	66.67	100.00	89.68	9.14	66.67	100.00
QLSSVM	88.94	9.56	62.50	100.00	87.06	8.79	75.00	100.00	89.37	8.76	66.67	100.00	88.41	8.76	66.67	100.00
LR	91.75	8.97	62.50	100.00	92.38	9.09	62.50	100.00	93.02	8.06	66.67	100.00	93.02	8.06	66.67	100.00

Table 15
CCA Results.

Model	AUC %								accuracy score %							
	t-test Based				Entropy Based				t-test Based				Entropy Based			
	mean	std	min	max	mean	std	min	max	mean	std	min	max	mean	std	min	max
DWPSVM	93.20	3.48	77.93	99.11	93.01	4.29	84.43	99.21	86.38	3.85	78.13	93.75	86.31	3.76	78.13	92.19
SQSSVM	92.32	3.42	80.99	98.13	92.22	4.38	81.67	98.42	86.03	3.87	78.13	92.19	86.44	3.82	78.13	92.19
SVM	91.36	3.68	78.42	98.82	92.28	4.44	83.15	98.62	86.28	3.82	78.13	92.19	86.31	3.76	78.13	92.19
SVM-Gauss	93.05	3.66	78.92	98.92	92.73	4.63	80.69	99.11	86.13	3.82	78.13	92.19	86.44	3.82	78.13	92.19
SVM-Quad	92.27	3.92	76.55	98.92	92.58	4.68	81.18	99.21	86.28	3.82	78.13	92.19	86.38	3.72	78.13	92.19
QLSTSVM	91.96	3.92	77.73	98.82	91.50	4.30	83.74	99.11	85.91	3.87	76.56	95.31	84.88	4.22	76.56	93.75
QLSSVM	86.87	4.01	77.39	95.71	86.98	4.83	77.39	95.71	86.06	3.81	78.13	92.19	86.44	3.82	78.13	92.19
LR	86.91	4.11	77.39	95.71	87.19	4.97	78.82	95.71	86.41	3.97	76.56	93.75	86.50	3.93	76.56	93.75

Table 16
GCD Results.

Model	AUC %								accuracy score %							
	t-test Based				Entropy Based				t-test Based				Entropy Based			
	mean	std	min	max	mean	std	min	max	mean	std	min	max	mean	std	min	max
DWPSVM	79.67	5.20	68.00	87.33	79.55	5.28	67.57	86.81	77.37	2.92	70.00	83.00	76.73	3.08	69.00	83.00
SQSSVM	78.65	6.05	66.33	88.48	77.65	5.93	64.86	85.52	77.00	2.60	70.00	81.00	75.38	3.77	70.00	81.00
SVM	79.56	5.20	67.67	87.29	79.52	5.33	67.00	86.19	76.67	3.12	66.00	83.00	77.10	3.27	66.00	83.00
SVM-Gauss	79.61	5.11	68.05	87.05	79.41	5.45	67.29	86.90	76.77	2.60	71.00	82.00	76.73	2.80	71.00	82.00
SVM-Quad	79.27	5.36	68.71	87.62	78.15	4.70	69.29	83.76	75.37	3.03	69.00	82.00	73.83	3.30	69.00	82.00
QLSTSVM	72.09	6.81	57.38	85.86	72.76	6.95	58.86	89.19	73.60	3.40	63.00	79.00	74.28	2.81	68.00	80.00
QLSSVM	68.69	5.70	59.29	79.05	67.37	5.73	57.62	77.86	77.17	2.70	70.00	81.00	77.15	2.39	73.00	80.00
LR	69.04	5.50	60.95	80.48	68.71	5.79	58.57	78.81	77.10	3.49	67.00	84.00	77.05	3.34	70.00	84.00

Table 17
JAP Results.

Model	AUC %								accuracy score %							
	t-test Based				Entropy Based				t-test Based				Entropy Based			
	mean	std	min	max	mean	std	min	max	mean	std	min	max	mean	std	min	max
DWPSVM	84.33	10.48	58.33	100.00	83.94	14.74	34.38	100.00	77.86	9.71	58.33	100.00	77.78	11.25	58.33	100.00
SQSSVM	79.31	15.23	28.13	100.00	80.63	14.05	31.25	100.00	77.14	9.27	58.33	91.67	77.78	10.39	58.33	91.67
SVM	82.00	15.24	28.13	100.00	82.94	14.34	28.13	100.00	75.83	8.99	58.33	91.67	76.67	10.41	58.33	91.67
SVM-Gauss	81.63	15.23	28.13	100.00	82.75	14.88	28.13	100.00	76.07	8.85	58.33	91.67	76.39	10.56	58.33	91.67
SVM-Quad	80.81	15.01	25.00	100.00	81.19	14.42	28.13	100.00	75.24	8.51	58.33	91.67	74.72	9.00	58.33	91.67
QLSTSVM	73.94	16.53	34.38	100.00	71.13	14.13	28.13	100.00	69.76	12.38	41.67	91.67	71.11	11.73	50.00	91.67
QLSSVM	66.13	12.63	43.75	87.50	68.00	12.54	43.75	87.50	77.26	9.92	58.33	100.00	77.50	9.32	58.33	91.67
LR	66.88	14.02	31.25	87.50	65.25	14.29	31.25	87.50	73.69	11.49	41.67	91.67	73.21	10.72	50.00	91.67

Table 18
Credit data CPU time.

Model	training CPU time (s)				testing CPU time (s)
	CCC (6, 106)	CCA (15, 653)	GCD (20, 1000)	JAP (10, 124)	
DWPSVM	<0.001	0.653	14.758	0.011	10^{-4}
SQSSVM	0.079	1.998	11.361	0.161	10^{-4}
SVM	<0.001	1.530	0.047	<0.001	$< 10^{-5}$
SVM-Gauss	<0.001	0.013	0.054	<0.001	$< 10^{-5}$
SVM-Quad	<0.001	0.007	0.048	0.002	$< 10^{-5}$
QLSTSVM	<0.001	0.002	0.076	<0.001	$< 10^{-5}$
QLSSVM	0.152	2.557	7.857	0.359	10^{-4}
LR	0.024	0.047	0.048	0.036	$< 10^{-5}$

Some observations and conclusions from Tables 14–18 are as the following:

- Compared with the entropy based feature weighting strategy, the SVM models equipped with t-test based weighting strategy produce higher mean accuracy scores and AUC values. It indicates that the t-test based weighting might be preferred in the applications of credit score forecasting.
- The proposed DWPSVM model outperforms all other tested models on GCD and JAP data sets. The credit score data may not be highly nonlinear, so the accuracy improvement of DWPSVM is not very obvious. Although LR produces the highest mean of accuracy scores on CCC and CCA, the mean of AUC values produced by LR is much lower than that of DWPSVM. Since the AUC depicts a more general behavior of a chassifier, DWPSVM may be more preferable in classifying credit score data sets.
- The training CPU time consumed by DWPSVM on small-sized credit data (e.g. CCC, JAP) is satisfying. Although it takes more time to implement the proposed model for larger data sets, such as GCD, the training CPU time of proposed model is 1–2 orders of magnitude slower than those of other tested models. Moreover, the testing CPU time of DWPSVM is as short as those of other tested models.

6. Conclusion

In this paper, we have derived the G-margin to propose a kernel-free quartic surface SVM for classifying nonlinearly separable data by directly using the DWP surface. Certain theoretical properties of DWPSVM model have been studied, and numerical experiments have been conducted to investigate the effectiveness and efficiency of the proposed model. The SMO algorithm has been adopted to implement the proposed DWPSVM model. Besides, the proposed model has been applied to credit scoring with some real-life corporate and personal credit data sets. We summarize some major findings here.

- In terms of classification accuracy, the proposed DWPSVM model performs better than other well-known SVM models. The separation surface produced by DWPSVM is a quartic surface, which is much more flexible than a quadratic surface. Consequently, the DWPSVM model has better capabilities to capture the hidden high-degree nonlinearity inside the data.
- The proposed DWPSVM model showed dominant performance on most of the artificial and public benchmark data sets. The numerical results on artificial data also indicated the increasing dominance of DWPSVM over other tested models as the number of data features increases. After being applied to a real-life corporate data set and three benchmark personal credit data sets, the proposed DWPSVM showed its stable effectiveness and acceptable efficiency in credit scoring. This shows the potential of DWPSVM in handling real-life classification problems.
- Unlike other kernel based nonlinear SVM models, the proposed DWPSVM model does not require any kernel functions or tuning their relative parameters. It saves considerable effort in the training process.

Our investigation of the proposed DWPSVM model for binary classification indicates some additional research works as follows. First, compared with kernel-based SVM models, the training CPU time of the proposed DWPSVM model is bigger for large-sized data sets. In fact, the kernel-based SVM models were implemented by using LIBSVM, which has been customized professionally, but the codes for implementing the proposed DWPSVM with SMO algorithm were written from scratch. So an immediate future work is to optimize and customize the codes of DWPSVM model for rapid computation. Another interesting work is to reformulate (DWPSVM) as an SDP problem and compare the SMO algorithm with SDP solvers. Moreover, the proposed DWPSVM model can be extended for other real-world applications including electric load forecasting (Luo, Hong, & Fang, 2018) and cross-selling recommendations (Chen et al., 2016).

Acknowledgements

This work has been sponsored by the US Army Research Office Grant #W911NF-15-1-0223, the National Natural Science Foundation of China Grant #71701035 and the Key Program of the National Natural Science Foundation of China Grant #71831003.

Appendix A. Proofs

A1. *Proof of Lemma 3.2*

Proof. Given a quadratic surface $\mathcal{S}$ denoted by (10), for any point $\boldsymbol{x} \in \mathcal{S}$, recall the definitions of $\hat{\boldsymbol{n}}(\boldsymbol{x})$, $\gamma(\boldsymbol{x})$, $\gamma^+(\boldsymbol{x})$, $\gamma^-(\boldsymbol{x})$, $\boldsymbol{x}^+$ and $\boldsymbol{x}^-$ in Section 3.2. By (11), $Q(\boldsymbol{x}^+) = 1$ and $Q(\boldsymbol{x}^-) = -1$, which are equivalent to

$$\begin{aligned}&\frac{1}{2}(\boldsymbol{x}+\gamma^+(\boldsymbol{x})\hat{\boldsymbol{n}}(\boldsymbol{x}))^T\mathbf{W}(\boldsymbol{x}+\gamma^+(\boldsymbol{x})\hat{\boldsymbol{n}}(\boldsymbol{x}))\\&\quad+\boldsymbol{b}^T(\boldsymbol{x}+\gamma^+(\boldsymbol{x})\hat{\boldsymbol{n}}(\boldsymbol{x}))+c-1=0\\&\frac{1}{2}(\boldsymbol{x}-\gamma^-(\boldsymbol{x})\hat{\boldsymbol{n}}(\boldsymbol{x}))^T\mathbf{W}(\boldsymbol{x}-\gamma^-(\boldsymbol{x})\hat{\boldsymbol{n}}(\boldsymbol{x}))\\&+\boldsymbol{b}^T(\boldsymbol{x}-\gamma^-(\boldsymbol{x})\hat{\boldsymbol{n}}(\boldsymbol{x}))+c+1=0\end{aligned} \tag{A.1}$$

With $Q(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^T\mathbf{W}\boldsymbol{x}+\boldsymbol{b}^T\boldsymbol{x}+c=0$, (A.1) can be simplified as the following:

$$\begin{aligned}\tfrac{1}{2}\gamma^+(\boldsymbol{x})^2\hat{\boldsymbol{n}}(\boldsymbol{x})^T\mathbf{W}\hat{\boldsymbol{n}}(\boldsymbol{x})+\gamma^+(\boldsymbol{x})\hat{\boldsymbol{n}}(\boldsymbol{x})^T(\mathbf{W}\boldsymbol{x}+\boldsymbol{b})-1 &= 0\\ \tfrac{1}{2}\gamma^-(\boldsymbol{x})^2\hat{\boldsymbol{n}}(\boldsymbol{x})^T\mathbf{W}\hat{\boldsymbol{n}}(\boldsymbol{x})-\gamma^-(\boldsymbol{x})\hat{\boldsymbol{n}}(\boldsymbol{x})^T(\mathbf{W}\boldsymbol{x}+\boldsymbol{b})+1 &= 0\end{aligned} \tag{A.2}$$

Notice that (A.2) are second order equations with respect to $\gamma^+(\boldsymbol{x})$ and $\gamma^-(\boldsymbol{x})$, respectively. Therefore, we will be able to solve out the explicit solutions as the following:

$$\gamma^+(\boldsymbol{x})=\frac{-\hat{\boldsymbol{n}}(\boldsymbol{x})^T(\mathbf{W}\boldsymbol{x}+\boldsymbol{b})\pm\sqrt{\left[\hat{\boldsymbol{n}}(\boldsymbol{x})^T(\mathbf{W}\boldsymbol{x}+\boldsymbol{b})\right]^2+2\hat{\boldsymbol{n}}(\boldsymbol{x})^T\mathbf{W}\hat{\boldsymbol{n}}(\boldsymbol{x})}}{\hat{\boldsymbol{n}}(\boldsymbol{x})^T\mathbf{W}\hat{\boldsymbol{n}}(\boldsymbol{x})}$$

$$\gamma^-(\boldsymbol{x})=\frac{\hat{\boldsymbol{n}}(\boldsymbol{x})^T(\mathbf{W}\boldsymbol{x}+\boldsymbol{b})\pm\sqrt{\left[\hat{\boldsymbol{n}}(\boldsymbol{x})^T(\mathbf{W}\boldsymbol{x}+\boldsymbol{b})\right]^2-2\hat{\boldsymbol{n}}(\boldsymbol{x})^T\mathbf{W}\hat{\boldsymbol{n}}(\boldsymbol{x})}}{\hat{\boldsymbol{n}}(\boldsymbol{x})^T\mathbf{W}\hat{\boldsymbol{n}}(\boldsymbol{x})}$$

Since $\hat{\boldsymbol{n}}(\boldsymbol{x})^T(W\boldsymbol{x}+\boldsymbol{b}) = \|\mathbf{W}\boldsymbol{x}+\boldsymbol{b}\|$, the above equations can be simplified as the following:

$$\gamma^+(\boldsymbol{x}) = \frac{-\|\mathbf{W}\boldsymbol{x}+\boldsymbol{b}\|\pm\sqrt{\|\mathbf{W}\boldsymbol{x}+\boldsymbol{b}\|^2+2\hat{\boldsymbol{n}}(\boldsymbol{x})^T\mathbf{W}\hat{\boldsymbol{n}}(\boldsymbol{x})}}{\hat{\boldsymbol{n}}(\boldsymbol{x})^T\mathbf{W}\hat{\boldsymbol{n}}(\boldsymbol{x})},\quad \gamma^-(\boldsymbol{x})$$
$$= \frac{\|\mathbf{W}\boldsymbol{x}+\boldsymbol{b}\|\pm\sqrt{\|\mathbf{W}\boldsymbol{x}+\boldsymbol{b}\|^2-2\hat{\boldsymbol{n}}(\boldsymbol{x})^T\mathbf{W}\hat{\boldsymbol{n}}(\boldsymbol{x})}}{\hat{\boldsymbol{n}}(\boldsymbol{x})^T\mathbf{W}\hat{\boldsymbol{n}}(\boldsymbol{x})}$$

By eliminating two useless roots, we have the explicit solutions of $\gamma^+(\boldsymbol{x})$ and $\gamma^-(\boldsymbol{x})$.

$$\gamma^+(\boldsymbol{x}) = \frac{-\|\mathbf{W}\boldsymbol{x}+\boldsymbol{b}\| + \sqrt{\|\mathbf{W}\boldsymbol{x}+\boldsymbol{b}\|^2 + 2\hat{\boldsymbol{n}}(\boldsymbol{x})^T\mathbf{W}\hat{\boldsymbol{n}}(\boldsymbol{x})}}{\hat{\boldsymbol{n}}(\boldsymbol{x})^T\mathbf{W}\hat{\boldsymbol{n}}(\boldsymbol{x})}$$
$$\gamma^-(\boldsymbol{x}) = \frac{\|\mathbf{W}\boldsymbol{x}+\boldsymbol{b}\| - \sqrt{\|\mathbf{W}\boldsymbol{x}+\boldsymbol{b}\|^2 - 2\hat{\boldsymbol{n}}(\boldsymbol{x})^T\mathbf{W}\hat{\boldsymbol{n}}(\boldsymbol{x})}}{\hat{\boldsymbol{n}}(\boldsymbol{x})^T\mathbf{W}\hat{\boldsymbol{n}}(\boldsymbol{x})} \tag{A.3}$$

By (12), there exists an orthonormal matrix $\mathbf{U} \in \mathbb{R}^{n\times n}$ such that $\mathbf{W} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{U}^T$, where $\boldsymbol{\Sigma}$ is a diagonal matrix of the singular values of $\mathbf{W}$. Recall that the singular values of $\mathbf{W}$ are in a decreasing order: $\sigma_1 \geqslant \sigma_2 \geqslant \cdots \geqslant \sigma_r \geqslant \sigma_{r+1} = \cdots = \sigma_n = 0$. Denote $\mathbf{U} = [\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_n]$. Then $\{\boldsymbol{u}_i \in \mathbb{R}^n\}_{1\leqslant i\leqslant n}$ forms an orthonormal basis on $\mathbb{R}^n$. Hence, there exists $\boldsymbol{\alpha}(\boldsymbol{x}) = [\alpha_1(\boldsymbol{x}), \alpha_2(\boldsymbol{x}), \ldots, \alpha_n(\boldsymbol{x})]^T \in \mathbb{R}^n$ such that

$$\mathbf{W}\boldsymbol{x}+\boldsymbol{b} = \sum_{i=1}^{n}\alpha_i(\boldsymbol{x})\boldsymbol{u}_i = \mathbf{U}\boldsymbol{\alpha}(\boldsymbol{x}). \tag{A.4}$$

Taking (A.4) into (*A*.3) and we have

$$\gamma^+(\boldsymbol{x}) = \frac{-\|\mathbf{U}\boldsymbol{\alpha}(\boldsymbol{x})\| + \sqrt{\|\mathbf{U}\boldsymbol{\alpha}(\boldsymbol{x})\|^2 + 2\frac{\boldsymbol{\alpha}(\boldsymbol{x})^T\boldsymbol{\Sigma}\boldsymbol{\alpha}(\boldsymbol{x})}{\boldsymbol{\alpha}(\boldsymbol{x})^T\boldsymbol{\alpha}(\boldsymbol{x})}}}{\frac{\boldsymbol{\alpha}(\boldsymbol{x})^T\boldsymbol{\Sigma}\boldsymbol{\alpha}(\boldsymbol{x})}{\boldsymbol{\alpha}(\boldsymbol{x})^T\boldsymbol{\alpha}(\boldsymbol{x})}},$$
$$\gamma^-(\boldsymbol{x}) = \frac{\|\mathbf{U}\boldsymbol{\alpha}(\boldsymbol{x})\| - \sqrt{\|\mathbf{U}\boldsymbol{\alpha}(\boldsymbol{x})\|^2 - 2\frac{\boldsymbol{\alpha}(\boldsymbol{x})^T\boldsymbol{\Sigma}\boldsymbol{\alpha}(\boldsymbol{x})}{\boldsymbol{\alpha}(\boldsymbol{x})^T\boldsymbol{\alpha}(\boldsymbol{x})}}}{\frac{\boldsymbol{\alpha}(\boldsymbol{x})^T\boldsymbol{\Sigma}\boldsymbol{\alpha}(\boldsymbol{x})}{\boldsymbol{\alpha}(\boldsymbol{x})^T\boldsymbol{\alpha}(\boldsymbol{x})}}$$

Notice that $\frac{\boldsymbol{\alpha}(\boldsymbol{x})^T\boldsymbol{\Sigma}\boldsymbol{\alpha}(\boldsymbol{x})}{\boldsymbol{\alpha}(\boldsymbol{x})^T\boldsymbol{\alpha}(\boldsymbol{x})}$ is the Rayleigh quotient of $\boldsymbol{\Sigma}$ at $\boldsymbol{\alpha}(\boldsymbol{x})$. Denote $\frac{\boldsymbol{\alpha}(\boldsymbol{x})^T\boldsymbol{\Sigma}\boldsymbol{\alpha}(\boldsymbol{x})}{\boldsymbol{\alpha}(\boldsymbol{x})^T\boldsymbol{\alpha}(\boldsymbol{x})} = R(\Sigma, \boldsymbol{\alpha}(\boldsymbol{x}))$. By Parseval's identity, $\|\mathbf{U}\boldsymbol{\alpha}(\boldsymbol{x})\|_2^2 = \|\boldsymbol{\alpha}(\boldsymbol{x})\|_2^2$. Hence, the G-margin at $\boldsymbol{x}$ can be written as

$$\begin{aligned}\gamma(\boldsymbol{x}) &= \gamma^+(\boldsymbol{x}) + \gamma^-(\boldsymbol{x})\\ &= \frac{\sqrt{\|\boldsymbol{\alpha}(\boldsymbol{x})\|^2 + 2R(\boldsymbol{\Sigma}, \boldsymbol{\alpha}(\boldsymbol{x}))} - \sqrt{\|\boldsymbol{\alpha}(\boldsymbol{x})\|^2 - 2R(\boldsymbol{\Sigma}, \boldsymbol{\alpha}(\boldsymbol{x}))}}{R(\boldsymbol{\Sigma}, \boldsymbol{\alpha}(\boldsymbol{x}))}\end{aligned} \tag{A.5}$$

Notice that inequality $\sqrt{x} - \sqrt{y} \leqslant \sqrt{x-y}$ holds for any $x \geqslant y \geqslant 0$. Therefore, we have inequality

$$\frac{1}{\gamma(\boldsymbol{x})} = \frac{1}{\gamma^+(\boldsymbol{x}) + \gamma^-(\boldsymbol{x})} \geqslant \frac{\sqrt{R(\boldsymbol{\Sigma}, \boldsymbol{\alpha}(\boldsymbol{x}))}}{2}. \tag{A.6}$$

Since $\boldsymbol{\Sigma}$ is known when $\mathcal{S}$ is given, we proved Lemma 3.2. □

A2. ***Proof of theorem 4.1***

Proof. Assume $(\boldsymbol{v}^*, \boldsymbol{b}^*, q^*, \boldsymbol{\zeta}^*)$ and $(\hat{\boldsymbol{v}}, \hat{\boldsymbol{b}}, \hat{q}, \hat{\boldsymbol{\zeta}})$ are both optimal solutions to problem (DWPSVM"). For a convex program, its optimal solution set is convex. In other words, $(\forall \alpha \in (0,1))$ $\alpha(\boldsymbol{v}^*, \boldsymbol{b}^*, q^*, \boldsymbol{\zeta}^*) + (1-\alpha)(\hat{\boldsymbol{v}}, \hat{\boldsymbol{b}}, \hat{q}, \hat{\boldsymbol{\zeta}})$ is optimal as well. Denote the optimal value as $\bar{z}$.

Define function $p : \mathbb{R}^{\frac{l(l+1)+n(n+1)}{2}+N} \to \mathbb{R}$, such that

$$p(\boldsymbol{v}, \boldsymbol{b}, \boldsymbol{\zeta}) \triangleq \frac{1}{2}\boldsymbol{v}^T\mathbf{H}\boldsymbol{v} + \frac{1}{2}\|\boldsymbol{b}\|_2^2 + C\sum_{i=1}^{N}\zeta_i. \tag{A.7}$$

which leads to

$$\begin{aligned}\bar{z} &= \frac{1}{2}\big(\alpha\boldsymbol{v}^* + (1-\alpha)\hat{\boldsymbol{v}}\big)^T\mathbf{H}\big(\alpha\boldsymbol{v}^* + (1-\alpha)\hat{\boldsymbol{v}}\big) + \frac{1}{2}\big(\alpha\boldsymbol{b}^*\\ &\quad + (1-\alpha)\hat{\boldsymbol{b}}\big)^T\big(\alpha\boldsymbol{b}^* + (1-\alpha)\hat{\boldsymbol{b}}\big) + C\sum_{i=1}^{N}\Big(\alpha\zeta_i^* + (1-\alpha)\hat{\zeta}_i\Big)\\ &= \frac{\alpha^2}{2}\Big(\boldsymbol{v}^{*T}\mathbf{H}\boldsymbol{v}^* + \boldsymbol{b}^{*T}\boldsymbol{b}^*\Big) + \frac{(1-\alpha)^2}{2}\Big(\hat{\boldsymbol{v}}^T\mathbf{H}\hat{\boldsymbol{v}} + \hat{\boldsymbol{b}}^T\hat{\boldsymbol{b}}\Big)\\ &\quad + \alpha(1-\alpha)\Big(\boldsymbol{v}^{*T}\mathbf{H}\hat{\boldsymbol{v}} + \boldsymbol{b}^{*T}\hat{\boldsymbol{b}}\Big) + C\sum_{i=1}^{N}\Big(\alpha\zeta_i^* + (1-\alpha)\hat{\zeta}_i\Big)\\ &= \alpha\, p(\boldsymbol{v}^*, \boldsymbol{b}^*, \boldsymbol{\zeta}^*) + (1-\alpha)\, p(\hat{\boldsymbol{v}}, \hat{\boldsymbol{b}}, \hat{\boldsymbol{\zeta}}) + \alpha(\alpha-1)\\ &\quad \times \Big(\big(\boldsymbol{v}^* - \hat{\boldsymbol{v}}\big)^T\mathbf{H}\big(\boldsymbol{v}^* - \hat{\boldsymbol{v}}\big) + (\boldsymbol{b}^* - \hat{\boldsymbol{b}})^T(\boldsymbol{b}^* - \hat{\boldsymbol{b}})\Big).\end{aligned}$$

Since $\bar{z} = p(\boldsymbol{v}^*, \boldsymbol{b}^*, \boldsymbol{\zeta}^*) = p(\hat{\boldsymbol{v}}, \hat{\boldsymbol{b}}, \hat{\boldsymbol{\zeta}})$ forces $\big(\boldsymbol{v}^* - \hat{\boldsymbol{v}}\big)^T\mathbf{H}\big(\boldsymbol{v}^* - \hat{\boldsymbol{v}}\big) + (\boldsymbol{b}^* - \hat{\boldsymbol{b}})^T(\boldsymbol{b}^* - \hat{\boldsymbol{b}}) = 0$, which implies $\big(\boldsymbol{v}^* - \hat{\boldsymbol{v}}\big)^T\mathbf{H}\big(\boldsymbol{v}^* - \hat{\boldsymbol{v}}\big) = 0$ due to the positive definiteness of $\mathbf{H}$ and $(\boldsymbol{b}^* - \hat{\boldsymbol{b}})^T(\boldsymbol{b}^* - \hat{\boldsymbol{b}}) = 0$.

In conclusion, $\boldsymbol{v}^* = \hat{\boldsymbol{v}}$ and $\boldsymbol{b}^* = \hat{\boldsymbol{b}}$. □

Appendix B. Auxiliary Information

All the benchmark data sets come from three sources: UCI machine learning repository (Dua & Graff, 2017), Kaggle, and Hsu, Chang, Lin et al. (2003). For the credit data sets, the CCC data is collected from a Chinese credit reporting agency. The CCA data was used in Quinlan (1987). The GCD and JAP data sets are collected from UCI machine learning repository (Dua & Graff, 2017). Please find the auxiliary data information at at https://github.com/gorgeous1992/DWPbinary.

Besides, the median of optimal parameters of all tested models on benchmark data sets are recorded in the link above.

References

Astorino, A., & Fuduli, A. (2015). Semisupervised spherical separation. *Applied Mathematical Modelling, 39*(20), 6351–6358. https://doi.org/10.1016/j.apm.2015.01.044.

Baesens, B., Van Gestel, T., Viaene, S., Stepanova, M., Suykens, J., & Vanthienen, J. (2003). Benchmarking state-of-the-art classification algorithms for credit scoring. *Journal of the Operational Research Society, 54*(6), 627–635.

Bai, Y., Han, X., Chen, T., & Yu, H. (2015). Quadratic kernel-free least squares support vector machine for target diseases classification. *Journal of Combinatorial Optimization, 30*(4), 850–870.

Blanquero, R., Carrizosa, E., Jimnez-Cordero, A., & Mart-n-Barragn, B. (2019). Functional-bandwidth kernel for support vector machine with functional data: An alternating optimization algorithm. *European Journal of Operational Research, 275*(1), 195–207. https://doi.org/10.1016/j.ejor.2018.11.024.

Chang, C.-C., & Lin, C.-J. (2011). LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology, 2*, 27:1–27:27. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm

Chen, Z.-Y., Fan, Z.-P., & Sun, M. (2016). A multi-kernel support tensor machine for classification with multitype multiway data and an application to cross-selling recommendations. *European Journal of Operational Research, 255*(1), 110–120.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning, 20*(3), 273–297.

Cristianini, N., & Shawe-Taylor, J. (2000). *An introduction to support vector machines and other kernel-based learning methods*. Cambridge University Press. https://doi.org/10.1017/CBO9780511801389.

Dagher, I. (2008). Quadratic kernel-free non-linear support vector machine. *Journal of Global Optimization, 41*(1), 15–30.

Deng, N., Tian, Y., & Zhang, C. (2012). *Support vector machines: optimization based theory, algorithms, and extensions*. Chapman and Hall/CRC.

Dua, D., & Graff, C. (2017). UCI machine learning repository. http://archive.ics.uci.edu/ml.

Fan, R.-E., Chen, P.-H., & Lin, C.-J. (2005). Working set selection using second order information for training support vector machines. *Journal of Machine Learning Research, 6*(Dec), 1889–1918.

Fang, S.-C., Gao, D., Lin, G.-X., Sheu, R.-L., & Xing, W. (2017). Double well potential function and its optimization in the *n*-dimensional real space–part I. *Journal of Industrial and Management Optimization, 13*, 1291–1305. https://doi.org/10.3934/jimo.2016073.

Gao, D. Y., & Yu, H. (2008). Multi-scale modelling and canonical dual finite element method in phase transitions of solids. *International Journal of Solids and Structures, 45*(13), 3660–3673.

Gao, Q.-Q., Bai, Y.-Q., & Zhan, Y.-R. (2019). Quadratic kernel-free least square twin support vector machine for binary classification problems. *Journal of the Operations Research Society of China, 7*(4), 539–559.

Hajek, P., & Michalak, K. (2013). Feature selection in corporate credit rating prediction. *Knowledge-Based Systems, 51*, 72–84.

Heuer, A., & Haeberlen, U. (1991). The dynamics of hydrogens in double well potentials: The transition of the jump rate from the low temperature quantum-mechanical to the high temperature activated regime. *The Journal of Chemical Physics, 95*(6), 4201–4214. https://doi.org/10.1063/1.461795.

Hsu, C.-W., Chang, C.-C., Lin, C.-J. et al. (2003). A practical guide to support vector classification.

Luo, J., Fang, S.-C., Deng, Z., & Guo, X. (2016). Soft quadratic surface support vector machine for binary classification. *Asia-Pacific Journal of Operational Research, 33*(06), 1650046.

Luo, J., Hong, T., & Fang, S.-C. (2018). Benchmarking robustness of load forecasting models under data integrity attacks. *International Journal of Forecasting, 34*(1), 89–104. https://doi.org/10.1016/j.ijforecast.2017.08.004.

Luo, J., Yan, X., & Tian, Y. (2020). Unsupervised quadratic surface support vector machine with application to credit risk assessment. *European Journal of Operational Research, 280*(3), 1008–1017.

Luo, Z.-Q., Ma, W.-K., So, A. M.-C., Ye, Y., & Zhang, S. (2010). Semidefinite relaxation of quadratic optimization problems. *IEEE Signal Processing Magazine, 27*(3), 20–34.

Maldonado, S., Pérez, J., & Bravo, C. (2017). Cost-based feature selection for support vector machines: An application in credit scoring. *European Journal of Operational Research, 261*(2), 656–665.

Meyer, C. (2000). Matrix analysis and applied linear algebra. *Other Titles in Applied Mathematics*. Society for Industrial and Applied Mathematics (SIAM, 3600 Market Street, Floor 6, Philadelphia, PA 19104).

Mousavi, S., Gao, Z., Han, L., & Lim, A. (2019). Quadratic surface support vector machine with $\ell 1$ norm regularization. arXiv:1908.08616.

Osuna, E. E., Freund, R. M., & Girosi, F. (1997). An improved training algorithm for support vector machines. In *Proceedings of the IEEE signal processing society workshop on neural networks for signal processing VII* (pp. 276–285).

Platt, J. (1998). Sequential minimal optimization: A fast algorithm for training support vector machines. *Technical Report*.

Quinlan, J. R. (1987). Simplifying decision trees. *International Journal of Man-Machine Studies, 27*(3), 221–234.

Scholkopf, B., & Smola, A. J. (2001). *Learning with kernels: support vector machines, regularization, optimization, and beyond*. Cambridge, MA, USA: MIT Press.

Tian, Y., Yong, Z., & Luo, J. (2018). A new approach for reject inference in credit scoring using kernel-free fuzzy quadratic surface support vector machines. *Applied Soft Computing, 73*, 96–105. https://doi.org/10.1016/j.asoc.2018.08.021.

Tsai, C.-F. (2009). Feature selection in bankruptcy prediction. *Knowledge-Based Systems, 22*(2), 120–127.

Vapnik, V. (2013). *The nature of statistical learning theory*. Springer Science & Business Media.

Xia, Y., Sheu, R.-L., Fang, S.-C., & Xing, W. (2017). Double well potential function and its optimization in the *n*-dimensional real space–part II. *Journal of Industrial and Management Optimization, 13*, 1307–1328. https://doi.org/10.3934/jimo.2016074.

Yan, X., Bai, Y., Fang, S.-C., & Luo, J. (2018). A proximal quadratic surface support vector machine for semi-supervised binary classification. *Soft Computing, 22*(20), 6905–6919.

Zhou, L., Lai, K. K., & Yen, J. (2009). Credit scoring models with AUC maximization based on weighted SVM. *International Journal of Information Technology & Decision Making, 8*(04), 677–696.