




Least squares approach to K-SVCR multi-class classification with its applications

Hossein Moosaei^{1,2}  · Milan Hladík³

Accepted: 22 April 2021 / Published online: 21 June 2021
© The Author(s), under exclusive licence to Springer Nature Switzerland AG 2021

Abstract

The support vector classification-regression machine for K-class classification (K-SVCR) is a novel multi-class classification method based on the “1-versus-1-versus-rest” structure. In this paper, we propose a least squares version of K-SVCR named LSK-SVCR. Similarly to the K-SVCR algorithm, this method assesses all the training data into a “1-versus-1-versus-rest” structure, so that the algorithm generates ternary outputs $\{-1, 0, +1\}$. In LSK-SVCR, the solution of the primal problem is computed by solving only one system of linear equations instead of solving the dual problem, which is a convex quadratic programming problem in K-SVCR. Experimental results on several benchmark, MC-NDC, and handwritten digit recognition data sets show that not only does the LSK-SVCR have better performance in the aspects of classification accuracy to that of K-SVCR and Twin-KSVC algorithms but also has remarkably higher learning speed.

Keywords Support vector machine · Twin-KSVC · K-SVCR · Multi-class classification · Least squares

1 Introduction

Support vector machines (SVM) were proposed for binary classification problems by Vapnik and his colleagues [1, 2]. They were used in many application areas including face recognition [3], heart disease detection [4], energies prediction [5–7], Raman spectroscopy [8], biomedicine [9], diagnosis of Alzheimer’s disease [10] and so on. The idea of this method is based on finding the maximum margin between two hyperplanes, which leads to

✉ Hossein Moosaei
hmoosaei@gmail.com; moosaei@ub.ac.ir; hmoosaei@kam.mff.cuni.cz

Milan Hladík
hladik@kam.mff.cuni.cz

¹ Department of Mathematics, Faculty of Science, University of Bojnord, Bojnord, Iran

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

³ Department of Applied Mathematics, Faculty of Mathematics and Physics, Charles University, Prague, Czech Republic

solving a constraint convex quadratic programming problem (QPP). Inspired by the generalized eigenvalue proximal support vector machine (GEP-SVM) [11], which seeks for two nonparallel hyperplanes such that each of them is as close as possible to its class and as far as possible to the other class, Jayadeva et al. [12] proposed twin support vector machine (TWSVM). Indeed, TWSVM obtains two non-parallel hyperplanes by solving two small-sized quadratic programming problems (QPPs) instead of one large QPP in classical SVM. In the past decades, many variants, extensions, and applications of SVM and TWSVM were proposed [13–17].

The variants and extensions of SVM and TWSVM can only solve binary classification problems, whereas the multi-class classification often occurs in practical problems [18]. For the multi-class classification problems in the SVM framework, two strategies are typically used. The first strategy “1-versus-1” constructs $\frac{k(k-1)}{2}$ binary classifiers [19]. This method may obtain unfavorable results due to omitting the remaining training samples in the training process of each classifier. The second strategy “1-versus-rest” constructs K binary classifiers so that each of them is involving all of the training samples [20]. Therefore in this case the class imbalance problem may occur.

Angulo et al. [21] proposed a new and effective method based on “1-versus-1-versus-rest” structure with ternary output $\{-1, 0, +1\}$ for K -class classification problems, called K -SVCR. This method constructs $\frac{k(k-1)}{2}$ classifiers so that each classifier is trained with all of the training data, which overcomes the risk of information loss and class imbalance problems, thus the K -SVCR provides better performance than SVM methods for multi-class classification problems.

By using the less computational time of the TWSVM and the structural advantage of K -SVCR, Xu et al., [22] proposed a new method for multi-class classification problems and termed it Twin-KSVC. Some improvements in this method were proposed by researchers. For instance, Nasiri et al. [23] proposed a version of least squares for Twin-KSVC and named as LSTKSVC, and also as another example, Tanveer et al. [24] suggested a least squares version of K -nearest neighbor based weighted Twin-KSVC.

In this paper, which is a revised and expanded version of our conference paper [25], we propose a least squares version of K -SVCR, named the least squares K -class support vector classification-regression machine (LSK-SVCR). Indeed, we replaced the inequality constraints with equality constraints and used 2-norm instead of 1-norm for minimizing the slack variables in the primal problem of K -SVCR. This smart modification leads to a fast algorithm with powerful generalization performance. Therefore in LSK-SVCR, we need to solve only one system of linear equations rather than solving a QPP in K -SVCR. Also in this paper, motivated by Lee and Huang [26], for large-scale data set, we propose the reduced LSK-SVCR. This method aims to reduce kernel matrix from $N \times N$ to a much smaller $N \times \tilde{N}$. It is worth to mention that the reduced kernel technique affects on computational time and the final results.

Numerical experiments on several benchmark data sets, USPS handwriting data set, and MC-NDC [27] indicate that the suggested LSK-SVCR has higher accuracy with lower computational time than the original K -SVCR and Twin-KSVC.

For classifying multi-class problems, most papers focused on solving QPP problems in primal or dual spaces. The main contribution of the paper is classifying multi-class classification problems by solving only a system of linear equations instead of solving QPP problems so that this leads to a fast and tailored algorithm that enjoys a satisfactory generalization performance. The experiments carried out on several artificial and real-world data sets verify the effectiveness of the proposed method.

The following are the highlights of our proposed LSK-SVCR method:

- The multi-class classification method based on the K-SVCR method is suggested.
- For finding the solution of our proposed method, LSK-SVCR, we just solve a system of linear equations which is different from K-SVCR and Twin-KSVC that require solving QPPs.
- The Sherman–Morrison–Woodbury (SMW) formulation is proposed to reduce the complexity of nonlinear LSK-SVCR.
- LSK-SVCR, similar to K-SVCR and Twin-KSVC, evaluates all training data into a “1-versus-1-versus-rest” structure with ternary outputs $\{-1, 0, +1\}$.
- We assess performance of our proposed method on real well-known UCI problems as well as on the US Postal (USPS) data set, which is a handwritten digit recognition data set, and MC-NDC
- In some experiments, for solving large-scale nonlinear LSK-SVCR a rectangular kernel technique is proposed.
- The results are analyzed by a well-known statistical method.

The rest of this paper is organized as follows: Section 2 briefly describes SVM, Twin SVM, Twin-KSVC, and K-SVCR. Section 3 presents our LSK-SVCR method in linear and non-linear cases as well as a classification decision rule and time complexity. Section 4 presents experimental results on UCI, USPS handwriting data sets, and MC-NDC, to show the efficiency of the proposed algorithm, and concluding remarks are given in Section 5.

Notations Let $a = [a_i]$ be a vector in R^n . If f is a real-valued function defined on the n -dimensional real space R^n , the gradient of f with respect to x is denoted by $\frac{\partial f}{\partial x}$, which is a column vector in R^n . By A^T we mean the transpose of a matrix A . For two vectors x and y in the n -dimensional real space, $x^T y$ denotes the scalar product. For $x \in R^n$, $\|x\|$ denotes 2-norm. A column vector of ones of arbitrary dimension is indicated by e . For $A \in R^{m \times n}$ and $B \in R^{n \times l}$, the kernel $k(A, B)$ is an arbitrary function which maps $R^{m \times n} \times R^{n \times l}$ into $R^{m \times l}$. In particular, if x and y are column vectors in R^n and $A \in R^{m \times n}$, then $k(x^T, y)$ is a real number, $k(x^T, A^T)$ is a row vector in R^m , and $k(A, A^T)$ is an $m \times m$ matrix. The identity $n \times n$ matrix is denoted by I_n , and $[A; B]$ stands for the matrix operation

$$[A; B] = \begin{bmatrix} A \\ B \end{bmatrix}.$$

2 Background

2.1 Support vector machine

For a classification problem, a data set (x_i, y_i) is given for training with the input $x_i \in R^n$ and the corresponding target value or label $y_i = 1$ or -1 , i.e.,

$$(x_1, y_1), \dots, (x_m, y_m) \in R^n \times \{\pm 1\}. \quad (1)$$

The two parallel supporting hyperplanes are defined as follow:

$$w^T x - b = +1 \quad \text{and} \quad w^T x - b = -1.$$

In the canonical form, the optimal hyperplanes are found by solving the following primal optimization problem [28]:

$$\begin{aligned} & \min_{w,b,\xi} \quad \frac{1}{2} w^T w + c e^T \xi \\ & \text{subject to} \quad \tilde{D}(Aw - eb) \geq e - \xi, \\ & \quad \quad \quad \xi \geq 0, \end{aligned} \tag{2}$$

where the matrix $A \in R^{m \times n}$ records the whole data, the diagonal matrix $\tilde{D} \in R^{m \times m}$ (with ones or minus ones along its diagonal) is according to membership of each point in the classes $+1$ or -1 , $c > 0$ is the regularization parameter, and ξ is a slack variable.

As for the primal problem, SVM solves its Lagrangian dual problem as follows:

$$\begin{aligned} & \min_{\alpha} \quad \sum_{i=1}^m \sum_{j=1}^m \alpha_i \alpha_j y_i y_j x_i x_j - \sum_{i=1}^m \alpha_i \\ & \text{subject to} \quad \sum_{i=1}^m y_i \alpha_i = 0, \\ & \quad \quad \quad 0 \leq \alpha_i \leq c, \quad i = 1, \dots, m, \end{aligned} \tag{3}$$

where α_i are the Lagrange multipliers, for $1 \leq i \leq m$.

2.2 Twin support vector machine

Twin support vector machine (TWSVM), suggested by Jayadeva et al. [29], finds two non-parallel hyperplanes for binary classification such that each plane is close to one of the two classes and as far as possible from the other class. The main idea of this method was inspired by GEPSVM [11]. However, TWSVM has a formulation similar to SVM formulation except that not all the patterns appear in the constraints of either problem at the same time. This makes TWSVM faster than standard SVM [29].

In fact TWSVM finds two nonparallel hyperplanes as follows:

$$f_1(x) = w_1^T x + b_1 \quad \text{and} \quad f_2(x) = w_2^T x + b_2, \tag{4}$$

where $w_1 \in R^n$, $w_2 \in R^n$, $b_1 \in R$ and $b_2 \in R$.

Suppose that all the data points in class $+1$ are associated with a matrix $A \in R^{m_1 \times n}$ and the data points of class -1 are associated with a matrix $B \in R^{m_2 \times n}$. The TWSVM classifiers are obtained by solving the following pair of quadratic programming problems:

$$\begin{aligned} & \min_{w_1, b_1, q_1} \quad \|Aw_1 + e_1 b_1\|^2 + c_1 e_2^T q_1, \\ & \text{subject to} \quad -(Bw_1 + e_2 b_1) + q_1 \geq e_2, \\ & \quad \quad \quad q_1 \geq 0. \end{aligned} \tag{5}$$

$$\begin{aligned} & \min_{w_2, b_2, q_2} \quad \|Bw_2 + e_2 b_2\|^2 + c_2 e_1^T q_2, \\ & \text{subject to} \quad (Aw_2 + e_1 b_2) + q_2 \geq e_1, \\ & \quad \quad \quad q_2 \geq 0, \end{aligned} \tag{6}$$

where $c_1, c_2 > 0$ are penalty parameters, e_1, e_2 are vectors of ones of appropriate dimensions, and q_1 and q_2 are slack vectors.

By using the KKT conditions, we can derive the Wolfe dual formulations of (5) and (6), respectively as follows:

$$\begin{aligned} & \max_{\alpha} e_2^T \alpha - \frac{1}{2} \alpha^T G (H^T H)^{-1} G^T \alpha, & (7) \\ & \text{subject to} & 0 \leq \alpha \leq c_1. \end{aligned}$$

$$\begin{aligned} & \max_{\gamma} e_1^T \gamma - \frac{1}{2} \gamma^T H (G^T G)^{-1} H^T \gamma, & (8) \\ & \text{subject to} & 0 \leq \gamma \leq c_2, \end{aligned}$$

where α and γ are the Lagrangian coefficients, $H = [A \ e_1]$ and $G = [B \ e_2]$. The non-parallel hyperplanes can be obtained from the solution of (7) and (8) by

$$[w_1; \ b_1] = -(H^T H)^{-1} G^T \alpha,$$

and

$$[w_2; \ b_2] = (G^T G)^{-1} H^T \gamma,$$

respectively.

To avoid the possible ill-conditioning, when $G^T G$ or $H^T H$ are (nearly) singular, the inverse matrices $(G^T G)^{-1}$ and $(H^T H)^{-1}$ are approximately replaced by $(G^T G + \delta I_{n+1})^{-1}$ and $(H^T H + \delta I_{n+1})^{-1}$, where δ is a small positive scalar.

2.3 Twin k -class support vector classification

Twin k -class support vector classification (Twin-KSVC), proposed in [22], is a new multi-class classification based on TWSVM. This method evaluates all the training data in a “1-versus-1-versus-rest” structure with ternary outputs $\{-1, 0, +1\}$ and solves two quadratic programming problems to obtain two non-parallel hyperplanes for classes $+1$ and -1 , and the remaining sample data is labeled by 0. Figure 1 illustrates a graphical representation of the Twin-KSVC method. The Twin-KSVC seeks two nonparallel hyperplanes:

$$x^T w_1 + b_1 = 0, \quad x^T w_2 + b_2 = 0. \tag{9}$$

Throughout this paper, we suppose without loss of generality that there are three classes $A_{m_1 \times n}$, $B_{m_2 \times n}$, and $C_{m_3 \times n}$ marked by class labels $+1$, -1 , and 0, respectively.

The Twin-KSVC classifiers are obtained by solving the following pair of QPPs:

$$\begin{aligned} & \min_{w_1, b_1, q_1, q_2} \frac{1}{2} \|Aw_1 + e_1 b_1\|^2 + c_1 e_2^T q_1 + c_2 e_3^T q_2, & (10) \\ & \text{subject to} & -(Bw_1 + e_2 b_1) + q_1 \geq e_2, \\ & & -(Cw_1 + e_3 b_1) + q_2 \geq e_3(1 - \epsilon), \\ & & q_1 \geq 0, \ q_2 \geq 0, \end{aligned}$$

and

$$\begin{aligned} & \min_{w_2, b_2, q_3, q_4} \frac{1}{2} \|Bw_2 + e_2 b_2\|^2 + c_3 e_1^T q_3 + c_4 e_3^T q_4, & (11) \\ & \text{subject to} & (Aw_2 + e_1 b_2) + q_3 \geq e_1, \\ & & (Cw_2 + e_3 b_2) + q_4 \geq e_3(1 - \epsilon), \\ & & q_3 \geq 0, \ q_4 \geq 0. \end{aligned}$$

Where $c_1, c_2, c_3, c_4 \geq 0$ are regularization parameters, e_1, e_2, e_3 , and e_4 are vectors of ones with appropriate dimensions, q_1, q_2, q_3 , and q_4 are slack variables, and ϵ is a positive

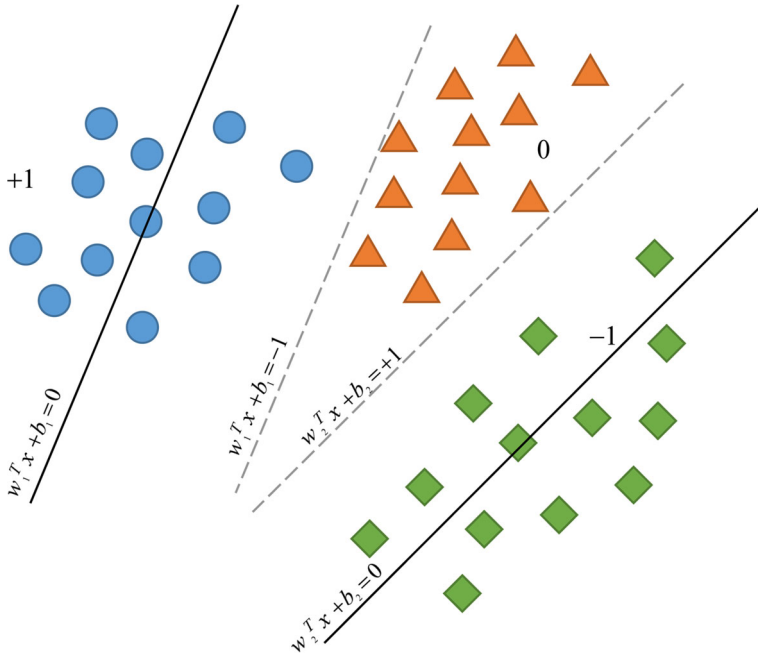


Fig. 1 Geometric representation of Twin-KSVC method

parameter. By introducing the Lagrangian function, the dual formulations of (10) and (11) can be represented as follows:

$$\begin{aligned} \max_{\gamma} \quad & e_4^T \gamma - \frac{1}{2} \gamma^T N (H^T H)^{-1} N^T \gamma, \\ \text{subject to} \quad & 0 \leq \gamma \leq F, \end{aligned} \tag{12}$$

where $H = [A \ e_1]$, $G = [B \ e_2]$, $M = [C \ e_3]$, $N = [G; M]$, $F = [c_1 e_2; c_2 e_3]$ and $e_4 = [e_2; e_3(1 - \epsilon)]$.

$$\begin{aligned} \max_{\alpha} \quad & e_5^T \alpha - \frac{1}{2} \alpha^T P (G^T G)^{-1} P^T \alpha, \\ \text{subject to} \quad & 0 \leq \alpha \leq F^*, \end{aligned} \tag{13}$$

where $P = [H; M]$, $F^* = [c_3 e_1; c_4 e_3]$ and $e_5 = [e_1; e_3(1 - \epsilon)]$. Now by solving the above quadratic problems, the separating hyperplanes (9) are given by the formula

$$[w_1; b_1] = -(H^T H + \delta I_{n+1})^{-1} N^T \gamma \quad \text{and} \quad [w_2; b_2] = (G^T G + \delta I_{n+1})^{-1} P^T \alpha.$$

Where the term δI_{n+1} (δ is a small positive number) is introduced in the case when the matrix is (nearly) singular.

2.4 K-support vector classification regression

K-SVCR, which is a new method of multi-class classification with ternary outputs $\{-1, 0, +1\}$, proposed in [21]. This method introduces the support vector classification-regression machine for K -class classification. This new machine evaluates all the training data into a “1-versus-1-versus-rest” structure during the decomposing phase using a mixed

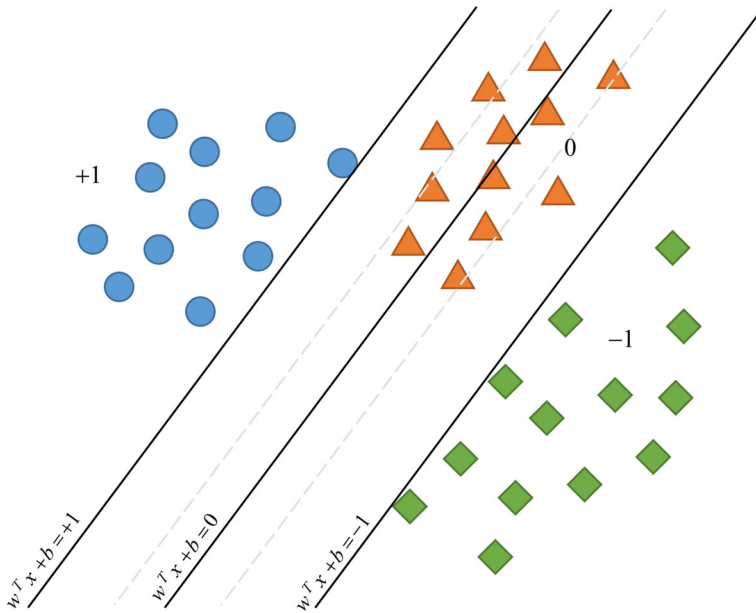


Fig. 2 Geometric representation of K-SVCR method

classification and regression support vector machine (SVM). Figure 2 illustrates the K-SVCR method graphically.

K-SVCR can be formulated as a convex quadratic programming problem as follows:

$$\begin{aligned}
 & \min_{w, b, \zeta_1, \zeta_2, \phi, \phi^*} \frac{1}{2} \|w\|^2 + c_1(e_1^T \zeta_1 + e_2^T \zeta_2) + c_2 e_3^T (\phi + \phi^*) \quad (14) \\
 & \text{subject to } Aw + e_1 b \geq e_1 - \zeta_1, \\
 & \quad \quad \quad -(Bw + e_2 b) \geq e_2 - \zeta_2, \\
 & \quad \quad \quad -\delta e_3 - \phi^* \leq Cw + e_3 b \leq \delta e_3 + \phi, \\
 & \quad \quad \quad \zeta_1, \zeta_2, \phi, \phi^* \geq 0.
 \end{aligned}$$

Where $c_1 > 0$ and c_2 are the regularization parameters, ζ_1 , ζ_2 , ϕ and ϕ^* are positive slack variables, and e_1 , e_2 , and e_3 are vectors of ones with proper dimensions. To avoid overlapping, the positive parameter δ must be lower than 1.

The dual formulation of (14) can be expressed as

$$\begin{aligned}
 & \max_{\gamma} q^T \gamma - \frac{1}{2} \gamma^T H \gamma, \quad (15) \\
 & \text{subject to } 0 \leq \gamma \leq F,
 \end{aligned}$$

where $Q = [A^T \ -B^T \ C^T \ -C^T]$, $H = Q^T Q$, $q = [e_1; e_2; -\delta e_3; -\delta e_3]$, and $F = [c_1 e_1; c_1 e_2; c_2 e_3; c_2 e_3]$. By solving this quadratic box constraint optimization problem, we can obtain the separating hyperplane $f(x) = w^T x + b$.

3 Least squares K-SVCR

In this section, we propose a least squares type of K-SVCR method, called LSK-SVCR, in both linear and nonlinear cases. This algorithm evaluates the training points in a structure “1-versus-1-versus-rest” with ternary outputs $\{-1, 0, +1\}$.

3.1 Linear case

We modify the primal problem (14) of K-SVCR as (16), which uses the square of the 2-norm of slack variables ζ_1, ζ_2, ϕ and ϕ^* instead of the 1-norm of the slack variables in the objective function, and it uses equality constraints instead of inequality constraints in K-SVCR. Then, the following minimization problem can be considered:

$$\begin{aligned} \min_{w,b,\zeta_1,\zeta_2,\phi,\phi^*} & \frac{1}{2} \|w\|^2 + c_1(\|\zeta_1\|^2 + \|\zeta_2\|^2) + c_2\|\phi\|^2 + c_3\|\phi^*\|^2 & (16) \\ \text{subject to} & e_1 - (Aw + e_1b) = \zeta_1, \\ & e_2 + (Bw + e_2b) = \zeta_2, \\ & Cw + e_3b - \delta e_3 = \phi^*, \\ & -Cw - e_3b - \delta e_3 = \phi. \end{aligned}$$

Where ζ_1, ζ_2, ϕ , and ϕ^* are positive slack variables, c_1, c_2 , and c_3 are penalty parameters, and the positive parameter δ is restricted to be lower than 1 to avoid overlapping.

In fact, the LSK-SVCR seeks for two parallel hyperplanes with maximum margin to separate classes A and B and at the same time, the middle separating hyperplane defines a δ -band that includes class C. Now by substituting the constraints into the objective function, we have the following unconstrained QPP:

$$\begin{aligned} \min_{w,b} & \frac{1}{2} \|w\|^2 + c_1\|e_1 - Aw - e_1b\| + c_1\|e_2 + Bw + e_2b\| \\ & + c_2\|-Cw - e_3b - \delta e_3\|^2 + c_3\|Cw + e_3b - \delta e_3\|. & (17) \end{aligned}$$

The objective function of problem (17) is convex, so for obtaining the optimal solution, we set the gradient of this function with respect to w and b to zero. Then we have:

$$\begin{aligned} \frac{\partial f}{\partial w} &= w + 2c_1(-A^T)(e_1 - Aw - e_1b) + 2c_1B^T(e_2 + Bw + e_2b) \\ &+ 2c_2(-C^T)(-Cw - e_3b - \delta e_3) + 2c_3C^T(Cw + e_3b - \delta e_3) = 0, \\ \frac{\partial f}{\partial b} &= 2c_1(-e_1^T)(e_1 - Aw - e_1b) + 2c_1e_2^T(e_2 + Bw + e_2b) \\ &+ 2c_2(-e_3^T)(-Cw - e_3b - \delta e_3) + 2c_3e_3^T(Cw + e_3b - \delta e_3) = 0. \end{aligned}$$

The above equation can be displayed in the matrix form as

$$\begin{aligned} 2c_1 \begin{bmatrix} A^T A & A^T e_1 \\ e_1^T A & e_1^T e_1 \end{bmatrix} \begin{bmatrix} w \\ b \end{bmatrix} + 2c_1 \begin{bmatrix} B^T B & B^T e_2 \\ e_2^T B & e_2^T e_2 \end{bmatrix} \begin{bmatrix} w \\ b \end{bmatrix} + 2(c_2 + c_3) \begin{bmatrix} C^T C & C^T e_3 \\ e_3^T C & e_3^T e_3 \end{bmatrix} \begin{bmatrix} w \\ b \end{bmatrix} \\ + \begin{bmatrix} 2c_1(-A^T)e_1 + 2c_1B^Te_2 + 2c_2C^T\delta e_3 + 2c_3C^T(-\delta e_3) \\ 2c_1(-e_1^T e_1) + 2c_1e_2^T e_2 + 2c_2\delta e_3^T e_3 + 2c_3e_3^T(-\delta e_3) \end{bmatrix} = 0. \end{aligned}$$

Algorithm 1 Linear LSK-SVCR.

Input: $A \in \mathbb{R}^{m_1 \times n}$ of class +1, $B \in \mathbb{R}^{m_2 \times n}$ of class -1, and the rest of sample data $C \in \mathbb{R}^{m_3 \times n}$ of class 0.

- 1: Set $E = [A \ e_1]$, $F = [B \ e_2]$, and $G = [C \ e_3]$.
- 2: Select penalty parameters c_1, c_2 , and c_3 and parameter δ .
- 3: Determine parameters of hyperplane (w, b) by using (18).
- 4: Assign a data point to class +1, -1, or 0 by using the decision function (22).

Therefore w and b can be computed as follows:

$$\begin{bmatrix} w \\ b \end{bmatrix} = \begin{bmatrix} c_1(A^T A + B^T B) + (c_2 + c_3)C^T C & c_1(A^T e_1 + B^T e_2) + (c_2 + c_3)C^T e_3 \\ c_1(e_1^T A + e_2^T B) + (c_2 + c_3)e_3^T C & c_1(e_1^T e_1 + e_2^T e_2) + (c_2 + c_3)e_3^T e_3 \end{bmatrix}^{-1} \begin{bmatrix} c_1(-A^T)e_1 + c_1 B^T e_2 + c_2 C^T \delta e_3 + c_3 C^T (-\delta e_3) \\ c_1(-e_1^T e_1) + c_1 e_2^T e_2 + c_2 \delta e_3^T e_3 + c_3 e_3^T (-\delta e_3) \end{bmatrix}.$$

We rewrite it as

$$\begin{bmatrix} w \\ b \end{bmatrix} = -[c_1 [A^T \ e_1^T] [A \ e_1] + c_1 [B^T \ e_2^T] [B \ e_2] + c_2 [C^T \ e_3^T] [C \ e_3] + c_3 [C^T \ e_3^T] [C \ e_3]]^{-1} \left(-c_1 \begin{bmatrix} A^T e_1 \\ m_1 \end{bmatrix} + c_1 \begin{bmatrix} B^T e_2 \\ m_2 \end{bmatrix} + c_2 \delta \begin{bmatrix} C^T e_3 \\ m_3 \end{bmatrix} - c_3 \delta \begin{bmatrix} C^T e_3 \\ m_3 \end{bmatrix} \right).$$

Denote $E = [A \ e_1]$, $F = [B \ e_2]$, and $G = [C \ e_3]$, then we can obtain the separating hyperplane by solving a system of linear equations and the solution becomes as follows:

$$\begin{bmatrix} w \\ b \end{bmatrix} = -[c_1 E^T E + c_1 F^T F + c_2 G^T G + c_3 G^T G]^{-1} \quad (18) \\ (-c_1 E^T e_1 + c_1 F^T e_2 + c_2 \delta G^T e_3 - c_3 \delta G^T e_3).$$

For clarity, the overall process for finding a label of a new sample is summarized in Algorithm 1.

3.2 Nonlinear case

In real-world problems, a linear kernel cannot always separate most of the classification tasks. To make the nonlinear types of problems separable, the samples are mapped to a higher dimensional feature space. Thus, in this subsection, we extend the linear case of LSK-SVCR to the nonlinear case. We would like to find the following kernel surface:

$$k(x^T, D^T)w + b = 0,$$

where $k(\cdot, \cdot)$ is an appropriate kernel function and $D = [A; B; C]$. After a careful selection of the kernel function, the primal problem of (14) becomes

$$\begin{aligned} \min_{w, b, \zeta_1, \zeta_2, \phi, \phi^*} \quad & \frac{1}{2} \|w\|^2 + c_1 (\|\zeta_1\|^2 + \|\zeta_2\|^2) + c_2 \|\phi\|^2 + c_3 \|\phi^*\|^2, \quad (19) \\ \text{subject to} \quad & e_1 - (k(A, D^T)w + e_1 b) = \zeta_1, \\ & e_2 + (k(B, D^T)w + e_2 b) = \zeta_2, \\ & k(C, D^T)w + e_3 b - \delta e_3 = \phi^*, \\ & -k(C, D^T)w - e_3 b - \delta e_3 = \phi. \end{aligned}$$

By substituting the constraints into the objective function, the problem takes the form

$$\min_{w,b} \frac{1}{2} \|w\|^2 + c_1 \|e_1 - k(A, D^T)w - e_1 b\| + c_1 \|e_2 + k(B, D^T)w + e_2 b\| + c_2 \| - k(C, D^T)w - e_3 b - \delta e_3 \|^2 + c_3 \|k(C, D^T)w + e_3 b - \delta e_3\|. \quad (20)$$

Similarly to the linear case, the solution of this convex optimization problem can be derived as follows:

$$\begin{bmatrix} w \\ b \end{bmatrix} = - \left[c_1 M^T M + c_1 N^T N + c_2 P^T P + c_3 P^T P \right]^{-1} (-c_1 M^T e_1 + c_1 N^T e_2 + c_2 \delta P^T e_3 - c_3 \delta P^T e_3),$$

where $M = [k(A, D^T) e_1] \in R^{m_1 \times (m+1)}$, $N = [k(B, D^T) e_2] \in R^{m_2 \times (m+1)}$, $P = [k(C, D^T) e_3] \in R^{m_3 \times (m+1)}$, $D = [A; B; C]$ and $m = m_1 + m_2 + m_3$.

The solution to the nonlinear case requires the inversion of a matrix of size $(m + 1) \times (m + 1)$. In general, a matrix has a special form if the number of features (nF) is much less than the number of samples (nS), i.e., $nS \gg nF$; in this case, the inverse matrix can be inverted by inverting a smaller $nF \times nF$ matrix by using the Sherman–Morrison–Woodbury (SMW) [30] formula. Therefore, in this paper, to reduce the computational cost, the SMW formula is applied.

More concretely, the SMW formula gives a convenient expression for the inverse matrix of $A + UV^T$, where $A \in R^{n \times n}$ and $U, V \in R^{n \times K}$, as follows:

$$(A + UV^T)^{-1} = A^{-1} - A^{-1}U(I_K + V^T A^{-1}U)^{-1}V^T A^{-1}.$$

Herein, A and $I_K + V^T A^{-1}U$ are nonsingular matrices.

By using this formula, we can reduce the computational cost and rewrite the above formula for the hyperplane as follows:

$$\begin{bmatrix} w \\ b \end{bmatrix} = - \left(Z - ZM^T \left(\frac{1}{c_1} I_{m_1} + MZM^T \right)^{-1} MZ \right) \begin{pmatrix} -c_1 M^T e_1 \\ +c_1 N^T e_2 + (c_2 - c_3) \delta P^T e_3 \end{pmatrix}, \quad (21)$$

where $Z = (c_1 N^T N + (c_2 + c_3) P^T P)^{-1}$. When we apply SMW formula on Z again, then we have

$$Z = \frac{1}{c_2 + c_3} \left(Y - YN^T \left(\frac{c_2 + c_3}{c_1} I_{m_2} + NYN^T \right)^{-1} NY \right),$$

where $Y = (P^T P)^{-1}$. Due to possible ill-conditioning of $P^T P$, we use a regularization term αI , ($\alpha > 0$ and small enough). Then we have $Y = \frac{1}{\alpha} (I_{m_3} - P^T (\alpha I + P P^T)^{-1} P)$. We now give an explicit statement of our nonlinear LSK-SVCR in Algorithm 2.

3.3 Decision rule

The multi-class classification techniques evaluate all training points into the “1-versus-1-versus-rest” structure with ternary outputs $\{-1, 0, +1\}$. For a new testing point x_i , we

Algorithm 2 Non linear LSK-SVCR.

- Input:** $A \in \mathbb{R}^{m_1 \times n}$ of class +1, $B \in \mathbb{R}^{m_2 \times n}$ of class -1, the rest of sample data $C \in \mathbb{R}^{m_3 \times n}$ of class 0, and $D = [A; B; C]$.
- 1: Choose a kernel function K .
 - 2: Set $M = [k(A, D^T) e_1] \in \mathbb{R}^{m_1 \times (m+1)}$, $N = [k(B, D^T) e_2] \in \mathbb{R}^{m_2 \times (m+1)}$, $P = [k(C, D^T) e_3] \in \mathbb{R}^{m_3 \times (m+1)}$.
 - 3: Select parameters c_1, c_2, c_3, δ and also the parameter of the Gaussian kernel γ .
 - 4: Determine parameters of hyperplane (w, b) by using (21).
 - 5: Assign a data point to class +1, -1, or 0 by using the decision function (23).

predict its class label by the following decision functions: For linear Twin-KSVC :

$$f(x_i) = \begin{cases} +1, & x_i^T w_1 + b_1 > -1 + \epsilon, \\ -1, & x_i^T w_2 + b_2 < 1 - \epsilon, \\ 0, & \text{otherwise.} \end{cases}$$

For nonlinear Twin-KSVC :

$$f(x_i) = \begin{cases} +1, & K(x_i^T, D^T)w_1 + b_1 > -1 + \epsilon, \\ -1, & K(x_i^T, D^T)w_2 + b_2 < 1 - \epsilon, \\ 0, & \text{otherwise.} \end{cases}$$

For linear K-SVCR and LSK-SVCR:

$$f(x_i) = \begin{cases} +1, & x_i^T w + b \geq \delta, \\ -1, & x_i^T w + b \leq -\delta, \\ 0, & \text{otherwise.} \end{cases} \tag{22}$$

For nonlinear K-SVCR and LSK-SVCR:

$$f(x_i) = \begin{cases} +1, & k(x_i^T, D^T)w + b \geq \delta, \\ -1, & k(x_i^T, D^T)w + b \leq -\delta, \\ 0, & \text{otherwise.} \end{cases} \tag{23}$$

For k -class classification problem, the “1-versus-1-versus-rest” constructs $\frac{K(K-1)}{2}$ classifiers in total, and for decision about final class label of testing sample x_i we get a total vote of each class. So the given testing sample will be assigned to the class label that gets the most votes (i.e., max-voting rule).

3.4 Time complexity

In this subsection, we discuss the time complexity of Twin-KSVC, K-SVCR, and LSK-SVCR. In three-class classification problems, suppose the total size of each class is equal to $m/3$ (where $m = m_1 + m_2 + m_3$).

In the K-SVCR problem, samples in the third class (i.e. C) are used twice in the constraints so this problem has $4m/3$ inequality constraints in total.

Twin-KSVC requires solving two box-constrained QPPs, each of them in $2m/3$ variables.

The computational complexity of K-SVCR is the complexity of solving one convex quadratic problem in dimension $n + 1$ and with $4m/3$ constraints, where n is the dimension of the input space.

In our proposed methods for the linear LSK-SVCR, we need to compute only one square system of linear equation of size $n + 1$.

In nonlinear LSK-SVCR, the inverse of a matrix of size $(m + 1) \times (m + 1)$ must be computed. The Sherman–Morrison–Woodbury (SMW) formula reduces the computational cost by finding the inverses of three matrices of smaller sizes $m_1 \times m_1$, $m_2 \times m_2$, and $m_3 \times m_3$.

4 Numerical experiments

To assess the performance of the proposed method, we apply LSK-SVCR on several UCI benchmark data sets [31] as well as handwriting data set and MC-NDC and compare our method with the K-SVCR and Twin-KSVC. All experiments were carried out in Matlab 2019b on a PC with Intel(R) CORE(TM) i7-7700HQ CPU@2.80GHz machine with 16 GB of RAM. For solving the dual problems of K-SVCR and Twin-KSVC, we used the `quadprog.m` function in Matlab. Also, we used 5-fold cross-validation to assess the performance of the algorithms in the aspect of accuracy and training time. Note that in 5-fold cross-validation, the data set is split randomly into five almost equal-size subsets, and one of them is reserved as a test set and the others play the role of a training set. This process is repeated five times, and the average accuracy of five testing results was used as the classification performance measure. Notice that the accuracy is defined as the number of correct predictions divided by the total number of predictions; to display it into a percentage we multiplied it by 100.

4.1 Parameter selection

The classification accuracy depends on the choice of parameters. Figure 3a and b show the influence of penalty and kernel parameters on the classification accuracy of K-SVCR and LSK-SVCR, respectively, for the Glass data set. In our computations for K-SVCR and

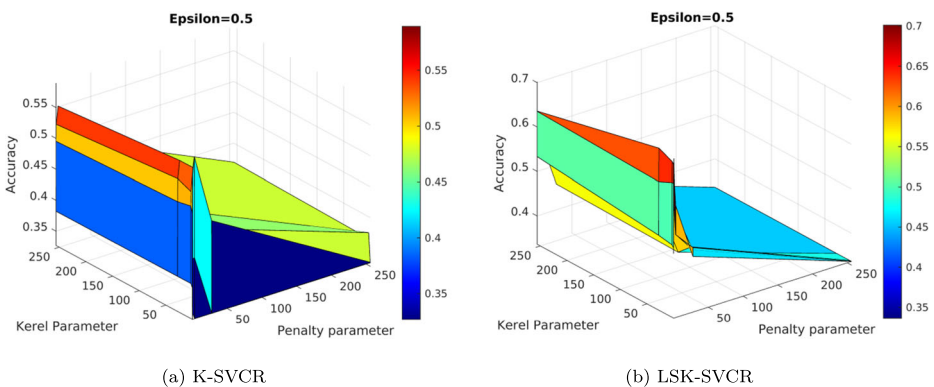


Fig. 3 Effect of penalty and kernel parameters on accuracy (K-SVCR and LSK-SVCR) for Glass data ($\epsilon = 0.5$)

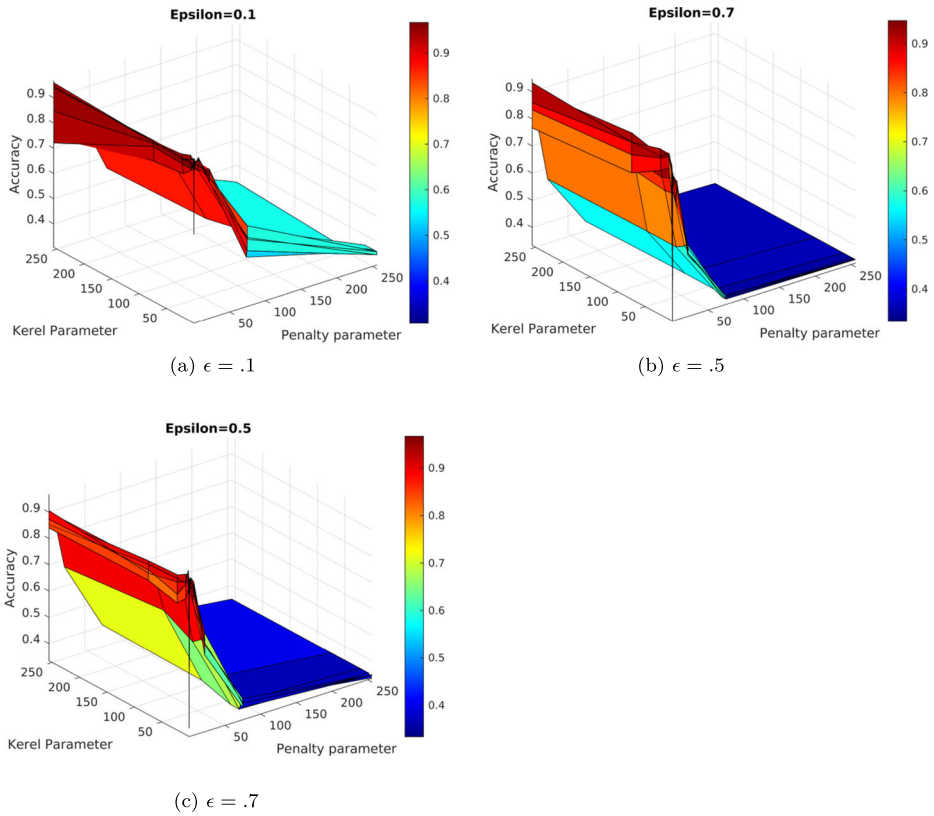


Fig. 4 Effect of epsilon parameter on accuracy of LSK-SVCR for Iris data set

LSK-SVCR, we set $c_1 = c_2$ and $c_1 = c_2 = c_3$, respectively and $\epsilon = 0.5$. Figure 4 shows the effect of ϵ parameter on the classification accuracy of LSK-SVCR for the Iris data set. Fig. 3 and Fig. 4 illustrate that the accuracy highly depends on the parameters. Therefore choosing the parameters is very important for the performance of classifiers. In other words, the classification performance is a function of parameter selection in these algorithms and we adopted the grid search method to choose the optimal values of the parameters [16, 32].

In the experiments, we opt for the Gaussian kernel function $k(x_i, x_j) = \exp\left(\frac{-\|x_i - x_j\|^2}{\gamma^2}\right)$. In this paper, to reduce the computational cost of the parameter selection, we set the regularization parameter values $c_1 = c_3$ and $c_2 = c_4$ in Twin-KSVC and the optimal value for c_1, c_2, c_3, c_4 , were selected from the set $\{2^i | i = -8, -7, \dots, 7, 8\}$, the parameters of the Gaussian kernel γ were selected from the set $\{2^i | i = -8, -7, \dots, 7, 8\}$, parameter δ in K-SVCR and LSK-SVCR was chosen from set $\{0.1, 0.3, \dots, 0.9\}$ and parameter ϵ in Twin-KSVC was selected from $\{0.1, 0.2, 0.3, 0.4\}$.

4.2 Results comparisons and discussion for UCI data sets

In this subsection, to compare the performance of K-SVCR, Twin-KSVC, and LSK-SVCR, we ran these algorithms on several benchmark data sets from the UCI machine learning repository [31]; they are described in Table 1.

Table 1 The characterization of data sets

Data set	Number of instances	Number of attributes	Number of classes
Iris	150	4	3
Balance	625	4	3
Soybean	47	35	4
Wine	178	13	3
Breast Tissue	106	10	4
Hayes-Roth	132	5	3
Ecoli	327	7	5
Teaching	151	5	3
Thyroid	215	5	3
Car	1728	6	4
Glass	214	9	6
Satimage	6435	36	6
PageBlock	5473	10	5
Contraceptive	1473	9	3

To analyze the performance of the Twin-KSVC, K-SVCR, and LSK-SVCR algorithms, Tables 2 and 3 show a comparison of classification accuracy and training time, respectively for Twin-KSVC, K-SVCR, and LSK-SVCR on several data sets. The bold value shows the best accuracy and time of algorithms. These tables indicate that for the Iris data set, the performance of LSK-SVCR (accuracy: 98.67, time: 0.03 s) was better than Twin-KSVC (accuracy: 94.46, time: 10.15 s) and K-SVCR (accuracy: 96.54, time: 1.72 s), so our proposed method was more accurate and faster than original K-SVCR and also Twin-KSVC. A similar discussion can be made for Balance, Soyabean, Wine, Brest Tissue, Hayes-Roth,

Table 2 Classification accuracy of Twin-KSVC, K-SVCR, and LSK-SVCR with Gaussian kernel

Data set	Twin-KSVC	K-SVCR	LSK-SVCR
	Acc \pm std	Acc \pm std	Acc \pm std
Iris	94.46 \pm 5.79	96.54 \pm 2.04	98.67 \pm 1.82
Balance	95.52 \pm 1.58	94.21 \pm 2.04	94.89 \pm 2.01
Soybean	100.00 \pm 0.00	100.00 \pm 0.00	100.00 \pm 0.00
Wine	97.12 \pm 3.14	98.81 \pm 2.51	99.45 \pm 1.24
Breast Tissue	46.49 \pm 16.65	47.06 \pm 9.73	46.59 \pm 15.39
Hayes-Roth	59.94 \pm 10.05	46.33 \pm 12.86	75.72 \pm 8.81
Ecoli	84.26 \pm 5.13	77.36 \pm 4.28	89.01 \pm 5.89
Teaching	70.91 \pm 6.93	63.68 \pm 5.58	70.19 \pm 7.46
Thyroid	91.62 \pm 5.57	83.25 \pm 6.02	93.49 \pm 2.55
Car	72.80 \pm 4.57	77.55 \pm 5.28	97.85 \pm 2.08
Glass	69.64 \pm 5.27	72.41 \pm 5.40	73.46 \pm 4.88
Satimage	85.83 \pm 6.59	80.04 \pm 5.09	90.68 \pm 5.42
PageBlock	79.09 \pm 6.28	90.57 \pm 6.14	93.29 \pm 5.92
Contraceptive	39.91 \pm 4.70	45.82 \pm 4.26	54.85 \pm 5.64

Table 3 Training time of Twin-KSVC, K-SVCR, and LSK-SVCR with Gaussian kernel

Data set	Twin-KSVC Time(s)	K-SVCR Time(s)	LSK-SVCR Time(s)
Iris	10.15	1.72	0.03
Balance	200.31	3.39	0.57
Soybean	10.92	1.44	0.07
Wine	10.91	0.27	0.03
Breast Tissue	13.13	1.37	0.09
Hayes-Roth	15.83	0.42	0.06
Ecoli	22.38	3.71	0.66
Teaching	10.07	0.32	0.04
Thyroid	8.69	0.60	0.09
Car	336.95	45.28	15.79
Glass	44.81	3.03	0.26
Satimage	16277.32	7148.50	1247.40
PageBlock	4598.50	8990.50	1496.10
Contraceptive	568.55	23.12	6.61

Ecoli, Teaching, and Thyroid, Car, Glass, Satimage, PageBlock, and Contraceptive data sets. The analysis of experimental results on 14 UCI data sets revealed that the performance of LSK-SVCR was better than the original K-SVCR and Twin-KSVC. We should note that for Balance, Breast Tissue, and Teaching, although the other methods are a bit more accurate than LSK-SVCR, the LSK-SVCR is always faster. Therefore, according to the experimental results in Tables 2 and 3, LSK-SVCR not only yielded higher classification accuracy but also had lower computational times.

4.3 USPS handwriting data set and discussion

The US Postal (USPS) data set, which is a handwritten digit recognition of 10 categories, contains 16×16 gray-level images from 0 to 9. The USPS is derived from a project on recognizing handwritten digits on envelopes [33]. Figure 5 shows some samples of this data set about digits 0, 3, 9, 2 in each row, respectively.

We want to classify some digits and compute accuracy and running time, here for example 0-vs-1-vs-2 illustrate classifying three classes 0, 1, and 2. The bold values show the best accuracy and running time of the algorithms.

Fig. 5 Examples of digits 0, 3, 9, and 2 from the USPS data set

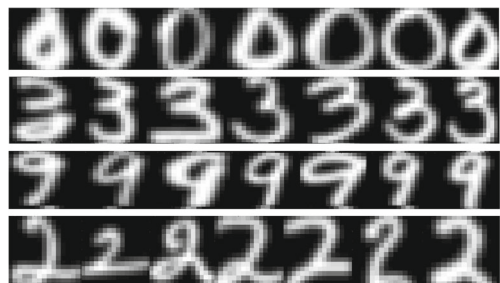


Table 4 Accuracy comparison on USPS data set with linear classifier

Data	Twin-KSVC	K-SVCR	LSK-SVCR
	Acc \pm std	Acc \pm std	Acc \pm std
0-vs-1-vs-2	51.02 \pm 1.70	65.52 \pm 0.73	98.13 \pm 0.38
4-vs-6-vs-8	48.90 \pm 0.46	62.86 \pm 2.08	96.49 \pm 0.68
5-vs-7-vs-9	49.46 \pm 0.91	65.35 \pm 2.08	96.43 \pm 0.83
1-vs-3-vs-5	68.38 \pm 1.25	70.34 \pm 2.77	96.79 \pm 0.91

Table 4 shows that our proposed method achieves better classification accuracy than the other two algorithms in all cases. Also from Table 5, we can see that LSK-SVCR costs the shortest training time among all algorithms.

4.4 MC-NDC data sets

In this subsection, we deeper analyze the advantage of the LSK-SVCR in the aspect of training time. Also, a reduced version of LSK-SVCR is proposed here.

The NDC data sets are generated by using David Musicants NDC Data Generator [34] and are normally distributed. Recently, Moosaei et al. proposed an extended version of it with an arbitrary number of samples, features, and classes; it was termed as MC-NDC [27]. Here we generated several 3 class data sets by using this code so that the size of samples increased from 1000 to 100000 with a fixed 32 number of features.

In our experiments, the parameters of all algorithms were fixed in advance ($c_1 = c_2 = c_3 = c_4 = 1$, $\gamma = 0.1$, $\epsilon = 0.1$, and $\delta = 0.1$). For the MC-NDC data sets which have more than 50000 samples, we used rectangular kernel [26] with 0.1% present of total data points. Table 6 shows the comparison of computing times for all three methods.

From Table 6, we see that when the number of samples is increasing, the Twin-KSVC and K-SVCR cannot solve the problem due to occurring out-of-memory error or very high computational time, while our proposed method works very well. So it is powerful to face high-dimensional data sets.

4.5 Statistical analysis

To further analyze the performance of the Twin-KSVC, K-SVCR, and LSK-SVCR algorithms on the UCI data set, this paper employs the Friedman test [35]. The Friedman test,

Table 5 Training time on USPS data set with linear classifier

Data	Twin-KSVC	K-SVCR	LSK-SVCR
	Time(s)	Time(s)	Time(s)
0-vs-1-vs-2	1799.54	261.04	2.69
4-vs-6-vs-8	5292.47	86.34	1.09
5-vs-7-vs-9	460.72	95.45	0.95
1-vs-3-vs-5	1135.12	124.50	1.45

Table 6 Comparison training times of Twin-KSVC, K-SVCR, and LSK-SVCR on MC-NDC data sets with Gaussian kernel

Data set	Twin-KSVC Time(s)	K-SVCR Time(s)	LSK-SVCR Time(s)
1000 × 32	2.52	23.89	.05
10000 × 32	138.39	b	1.99
50000 × 32 ^a	c	c	143.92
70000 × 32 ^a	c	c	410.59
100000 × 32 ^a	c	c	1273.40

^a The rectangular kernel with a reduction rate of 0.1% of data. ^b Experiment was stopped due to very high computing time. ^c Experiments terminated as the system was out of memory

which is commonly used by researchers, ranks algorithms for each data set separately such that the best performing algorithm gets rank 1, and the second-best performing algorithm gets rank 2. In case two algorithms perform similarly, the average ranks are assigned to them [13, 36, 37]. The Friedman test is calculated as:

$$\chi_F^2 = \frac{12N}{k(k+1)} \left[\sum_{j=1}^k R_j^2 - \frac{k(k+1)^2}{4} \right],$$

where $R_j = \frac{1}{N} \sum_i r_i^j$, k is the total number of algorithms, N is the number of data sets used in the study, and r_i^j denotes the rank of the j -th classifier of k algorithms on the i -th data set.

The Friedman's χ_F^2 is undesirably conservative and in [38] it is proposed a better statistic as follows:

$$F_f = \frac{(N-1)\chi_F^2}{N(k-1)-\chi_F^2},$$

which is distributed according to the F -distribution with $(k-1, (k-1)(N-1))$ degrees of freedom.

Table 7 Rank of accuracy for Twin-KSVC, K-SVCR, and LSK-SVCR with Gaussian kernel

Data set	Twin-KSVC	K-SVCR	LSK-SVCR
Iris	3	2	1
Balance	1	3	2
Soybean	2	2	2
Wine	3	2	1
Breast Tissue	3	1	2
Hayes-Roth	2	3	1
Ecoli	3	3	1
Teaching	1	3	2
Thyroid	2	3	1
Car	3	2	1
Glass	3	2	1
Satimage	2	3	1
PageBlock	3	2	1
Contraceptive	3	2	1
Average rank	2.42	2.35	1.28

Table 8 Rank of time for Twin-KSVC, K-SVCR, and LSK-SVCR with Gaussian kernel

Data set	Twin-KSVC	K-SVCR	LSK-SVCR
Iris	3	2	1
Balance	3	2	1
Soybean	3	2	1
Wine	3	2	1
Breast Tissue	3	2	1
Hayes-Roth	3	2	1
Ecoli	3	2	1
Teaching	3	2	1
Thyroid	3	2	1
Car	3	2	1
Glass	3	2	1
Satimage	3	2	1
PageBlock	2	3	1
Contraceptive	3	2	1
Average rank	2.92	2.07	1

Table 7 shows the rank of each algorithm in terms of classification accuracy for each data set. The bold value shows the best average rank of algorithms.

Now the χ^2_F and F_f are calculated as follows:

$$\chi^2_F = \frac{12 \times 14}{3(3+1)} [(2.42)^2 + (2.35)^2 + (1.28)^2 - \frac{3(3+1)^2}{4}] = 14.24,$$

$$F_f = \frac{(14-1) \times 14.24}{14(3-1) - 14.24} = 13.45.$$

With our three algorithms and 14 UCI data sets for the nonlinear case, F_f is distributed according to the F -distribution with $((k-1), (k-1)(N-1)) = (2, 36)$ degrees of freedom. The critical values of $F(2, 26)$ are $F(2, 26) = 3.37$ and $F(2, 26) = 5.53$ for $\alpha = 0.05$ and $\alpha = 0.01$, respectively. Table 7 and the critical value of $F(2, 26)$ when $\alpha = 0.05$ and $\alpha = 0.01$ show a significant difference between the performance of the algorithms in the aspect of classification accuracy. Regarding this point that the LSK-SVCR algorithm has the highest Friedman score (lowest average rank) among all the algorithms and the value of F_f is much larger than the critical values, we can conclude that there is a significant difference between these three algorithms and therefore the LSK-SVCR algorithm is more accurate.

For Table 8, χ^2_F and F_f are calculated as follows:

$$\chi^2_F = \frac{12 \times 14}{3(3+1)} [(1)^2 + (2.07)^2 + (2.92)^2 - \frac{3(3+1)^2}{4}] = 25.35,$$

$$F_f = \frac{(14-1) \times 25.35}{14(3-1) - 25.35} = 124.35.$$

Given three algorithms and 14 UCI data sets for the non linear case, F_f is distributed according to the F -distribution with $((k-1), (k-1)(N-1)) = (2, 26)$ degrees of freedom. The critical values of $F(2, 26)$ are $F(2, 26) = 3.37$ and $F(2, 26) = 5.53$ for $\alpha = 0.05$ and $\alpha = 0.01$, respectively. Table 8 and the critical value of $F(2, 26)$ when $\alpha = 0.05$ and $\alpha = 0.01$ show that there is a very high significant difference between the performance of the algorithms in the aspect of training time. Regarding this point that the LSK-SVCR algorithm has the highest Friedman score (lowest average rank) among all the algorithms

and the values of F_f are too much larger than the critical values, we can conclude that there is the significant difference between these three algorithms. Therefore the LSK-SVCR algorithm is the fastest in terms of learning speed.

5 Conclusion

The support vector classification-regression machine for K-class classification (K-SVCR) is a novel multi-class method. In this paper, we proposed a least squares version of K-SVCR named LSK-SVCR for multi-class classification. Our proposed method leads to solving a simple system of linear equations instead of solving a QPP in K-SVCR. The LSK-SVCR, similar to K-SVCR and Twin-KSVC, evaluates all training data into the “1-versus-a-versus-rest” structure with ternary outputs $\{-1, 0, +1\}$.

The computational results performed on several UCI data sets, the USPS handwriting data set, and MC-NDC data sets demonstrate that, compared to K-SVCR and Twin-KSVC, the proposed LSK-SVCR has better efficiency in terms of accuracy and training time. Therefore the proposed method can be used for solving multi-class classification problems, involving disease detection, handwritten digits recognition and many other real-world problems.

As future work, an adaptation of the proposed method can be considered to obtain sparse solutions for multi-class classification problems.

Acknowledgements The authors were supported by the Czech Science Foundation Grant P403-18-04735S.

References

1. Boser, B.E., Guyon, I.M., Vapnik, V.: A training algorithm for optimal margin classifiers. In: Proceedings of the fifth annual workshop on Computational learning theory, COLT '92, pp. 144–152. Association for Computing Machinery, New York (1992)
2. Cortes, C., Vapnik, V.: Support-vector networks. *Mach. Learn.* **20**(3), 273–297 (1995). <https://doi.org/10.1007/BF00994018>
3. Déniz, O., Castrillon, M., Hernández, M.: Face recognition using independent component analysis and support vector machines. *Pattern Recogn. Lett.* **24**(13), 2153–2157 (2003)
4. Arabasadi, Z., Alizadehsani, R., Roshanzamir, M., Moosaei, H., Yarifard, A.A.: Computer aided decision making for heart disease detection using hybrid neural network-genetic algorithm. *Comput. Methods Programs Biomed.* **141**, 19–26 (2017)
5. Ahmad, A.S., Hassan, M.Y., Abdullah, M.P., Rahman, H.A., Hussin, F., Abdullah, H., Saidur, R.: A review on applications of ANN and SVM for building electrical energy consumption forecasting. *Renew. Sustain. Energy Rev.* **33**, 102–109 (2014)
6. Shao, M., Wang, X., Bu, Z., Chen, X., Wang, Y.: Prediction of energy consumption in hotel buildings via support vector machines. *Sustain. Cit. Soc.*, 102128 (2020)
7. Zhao, H., Magoulès, F.: A review on the prediction of building energy consumption. *Renew. Sustain. Energy Rev.* **16**(6), 3586–3592 (2012)
8. Fenn, M.B., Xanthopoulos, P., Pyrgiotakis, G., Grobmyer, S.R., Pardalos, P.M., Hench, L.L.: Raman spectroscopy for clinical oncology. *Adv. Opt. Technol.* 2011 (2011)
9. Pardalos, P.M., Boginski, V.L., Vazacopoulos, A.: Data mining in biomedicine. *Springer Optimization and Its Applications*, vol. 7. Springer (2007)
10. Tanveer, M., Richhariya, B., Khan, R.U., Rashid, A.H., Khanna, P., Prasad, M., Lin, C.T.: Machine learning techniques for the diagnosis of alzheimers disease: A review. *ACM Trans. Multimed. Comput. Commun. Appl. (TOMM)* **16**(1s), 1–35 (2020)
11. Mangasarian, O.L., Wild, E.W.: Multisurface proximal support vector machine classification via generalized eigenvalues. *IEEE Trans. Pattern Anal. Mach. Intell.* **28**(1), 69–74 (2005)

12. Khemchandani, R., Chandra, S., et al.: Twin support vector machines for pattern classification. *IEEE Trans. Pattern Anal. Mach. Intell.* **29**(5), 905–910 (2007)
13. Bazikar, F., Ketabchi, S., Moosaei, H.: DC programming and DCA for parametric-margin ν -support vector machine. *Appl. Intell.* **50**(6), 1763–1774 (2020)
14. Ding, S., Shi, S., Jia, W.: Research on fingerprint classification based on twin support vector machine. *IET Image Process.* **14**(2), 231–235 (2019)
15. Ding, S., Zhang, N., Zhang, X., Wu, F.: Twin support vector machine: theory, algorithm and applications. *Neural Comput. Appl.* **28**(11), 3119–3130 (2017)
16. Ketabchi, S., Moosaei, H., Razzaghi, M., Pardalos, P.M.: An improvement on parametric ν -support vector algorithm for classification. *Ann. Oper. Res.* **276**(1-2), 155–168 (2019)
17. Trafalis, T.B., Ince, H.: Support vector machine for regression and applications to financial forecasting. In: Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium, vol. 6, pp. 348–353. IEEE (2000)
18. Tang, L., Tian, Y., Pardalos, P.M.: A novel perspective on multiclass classification: regular simplex support vector machine. *Inform. Sci.* **480**, 324–338 (2019)
19. Kressel, U.: Pairwise classification and support vector machines. In: Scholkopf, B., et al. (eds.) *Advances in Kernel Methods: Support Vector Learning*, pp. 255–268. MIT Press (1998)
20. Hsu, C.-W., Lin, C.-J.: A comparison of methods for multiclass support vector machines. *IEEE Trans. Neural Netw.* **13**(2), 415–425 (2002)
21. Angulo, C., Català, A.: K-SVCR. A multi-class support vector machine. In: López de Mántaras, R., Plaza, E. (eds.) *Machine Learning: ECML 2000, LNCS*, vol. 1810, pp. 31–38. Springer, Berlin (2000)
22. Xu, Y., Guo, R., Wang, L.: A twin multi-class classification support vector machine. *Cogn. Comput.* **5**(4), 580–588 (2013). <https://doi.org/10.1007/s12559-012-9179-7>
23. Nasiri, J.A., Charkari, N.M., Jalili, S.: Least squares twin multi-class classification support vector machine. *Pattern Recogn.* **48**(3), 984–992 (2015). <https://doi.org/10.1016/j.patcog.2014.09.020>
24. Tanveer, M., Sharma, A., Suganthan, P.N.: Least squares KNN-based weighted multiclass twin SVM. *Neurocomputing* (2020)
25. Moosaei, H., Hladík, M.: Least squares K-SVCR multi-class classification. In: Kotsireas, I.S., Pardalos, P.M. (eds.) *Learning and Intelligent Optimization, LNCS*, vol. 12096, pp. 117–127. Springer, Cham (2020)
26. Lee, Y.-J., Huang, S.-Y.: Reduced support vector machines: A statistical theory. *IEEE Trans. Neural Netw.* **18**(1), 1–13 (2007)
27. Moosaei, H., Musicant, D.R., Khosravi, S., Hladík, M.: MC-NDC: multi-class normally distributed clustered datasets. Carleton College, University of Bojnord. (2020)
28. Vapnik, V., Chervonenkis, A.J.: *Theory of pattern recognition*. Nauka (1974)
29. Jayadeva, Khemchandani, R., Chandra, S.: Twin support vector machines for pattern classification. *IEEE Trans. Pattern Anal. Mach. Intell.* **29**(5), 905–910 (2007). <https://doi.org/10.1109/TPAMI.2007.1068>
30. Golub, G.H., Van Loan, C.F.: *Matrix computations*. Johns Hopkins University Press, Baltimore (2012)
31. Lichman, M.: UCI machine learning repository. <http://archive.ics.uci.edu/ml> (2013)
32. Hsu, C.-W., Chang, C.-C., Lin, C.-J., et al.: A practical guide to support vector classification, Taipei. <https://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf> (2003)
33. Hastie, T., Tibshirani, R., Friedman, J.: *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media (2009)
34. Musicant, D.: NDC: normally distributed clustered datasets. Computer Sciences Department, University of Wisconsin, Madison (1998)
35. Friedman, M.: A comparison of alternative tests of significance for the problem of m rankings. *Ann. Math. Stat.* **11**(1), 86–92 (1940)
36. Tanveer, M., Khan, M.A., Ho, S.-S.: Robust energy-based least squares twin support vector machines. *Appl. Intell.* **45**(1), 174–186 (2016)
37. Wang, H., Zhou, Z., Xu, Y.: An improved ν -twin bounded support vector machine. *Appl. Intell.* **48**(4), 1041–1053 (2018)
38. Iman, R.L., Davenport, J.M.: Approximations of the critical region of the fbietkan statistic. *Commun. Stat. - Theory Methods* **9**(6), 571–595 (1980)