



# Least Squares K-SVCR Multi-class Classification

Hossein Moosaei<sup>1,2</sup>(✉)  and Milan Hladík<sup>2</sup> 

<sup>1</sup> Department of Mathematics, Faculty of Science, University of Bojnord, Bojnord, Iran

[hmoosaei@gmail.com](mailto:hmoosaei@gmail.com)

<sup>2</sup> Department of Applied Mathematics, School of Computer Science, Faculty of Mathematics and Physics, Charles University, Prague, Czech Republic  
[hladik@kam.mff.cuni.cz](mailto:hladik@kam.mff.cuni.cz)

**Abstract.** The support vector classification-regression machine for K-class classification (K-SVCR) is a novel multi-class classification method based on “1-versus-1-versus-rest” structure. In this paper, we propose a least squares version of K-SVCR named as LSK-SVCR. Similarly as the K-SVCR algorithm, this method assess all the training data into a “1-versus-1-versus-rest” structure, so that the algorithm generates ternary output  $\{-1, 0, +1\}$ . 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 data set show that the LSK-SVCR has better performance in the aspects of predictive accuracy and learning speed.

**Keywords:** SVM · K-SVCR · Multi-class classification · Least squares

## 1 Introduction

Support vector machines (SVM) were proposed for binary classification problems by Vapnik and his colleagues [3, 4]. The idea of this method is based on finding the maximum margin between two hyperplanes, which leads to solving a constraint convex quadratic programming problem (QPP).

Whereas there are many methods for binary classification [2, 7–9, 12], multi-class classification is often accrued in practical problems and real life [11]. Due to its wide range of applications, Angulo et al. [1] introduced a new method for multi-class classification based on “1-versus-1-versus-rest” structure with ternary output  $\{-1, 0, +1\}$  for K-class classification. This method constructs  $\frac{k(k-1)}{2}$  K-SVCR classifiers. It should be noted that, since all samples are given for construction of classifiers, the K-SVCR provides better performance than SVM methods for multi-class problems.

---

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

© Springer Nature Switzerland AG 2020

I. S. Kotsireas and P. M. Pardalos (Eds.): LION 14 2020, LNCS 12096, pp. 117–127, 2020.

[https://doi.org/10.1007/978-3-030-53552-0\\_13](https://doi.org/10.1007/978-3-030-53552-0_13)

In this study, we propose a least squares version of K-SVCR, named as the least squares K-class support vector classification-regression machine (LSK-SVCR). In LSK-SVCR, we need to solve only one system of linear equations rather than solving a QPP in K-SVCR.

Numerical experiments on several benchmark data set indicate that the suggested LSK-SVCR has higher accuracy with lower computational time than K-SVCR.

*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  $f$  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 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

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

The rest of this paper is organized as follows: Sect. 2 briefly describes SVM, and K-SVCR is then introduced in Sect. 3. Section 4 presents our LSK-SVCR in linear and non-linear cases as well as a classification decision rule. We analyse the computational complexity of the methods in Sect. 5. Section 6 presents experimental results on benchmark data set to show the efficiency of the proposed algorithm, and concluding remarks are given in Sect. 7.

## 2 Support Vector Machine for Classification

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 canonical form, the optimal hyperplanes are found by solving the following primal optimization problem [13]:

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

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  $\xi$  is a slack variable. As for the primal problem, SVM solves its Lagrangian dual problem as follows:

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

where  $\alpha_i$ s are the Lagrange multipliers.

### 3 Support Vector Classification-Regression Machine for K-Class Classification

K-SVCR, which is a new method of multi-class classification with ternary output  $\{-1, 0, +1\}$ , has been proposed in [1]. 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 classification and regression support vector machine (SVM). Figure 1 illustrates the K-SVCR method graphically.

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

$$\begin{aligned} \min_{w, b, \zeta_1, \zeta_2, \phi, \phi^*} \quad & \frac{1}{2} \|w\|^2 + c_1 (e_1^T \zeta_1 + e_2^T \zeta_2) + c_2 e_3^T (\phi + \phi^*) \\ \text{subject to} \quad & Aw + e_1 b \geq e_1 - \zeta_1, \\ & -(Bw + e_2 b) \geq e_2 - \zeta_2, \\ & -\delta e_3 - \phi^* \leq Cw + e_3 b \leq \delta e_3 + \phi, \\ & \zeta_1, \zeta_2, \phi, \phi^* \geq 0. \end{aligned} \quad (4)$$

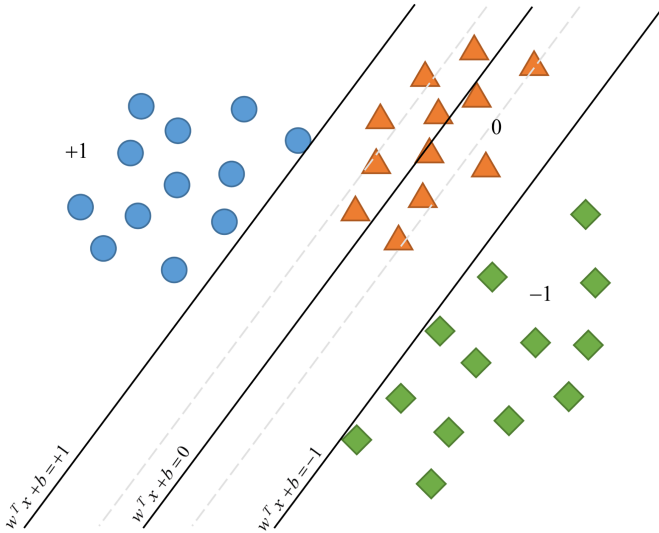
Where  $c_1 > 0$  and  $c_2$  are the regularization parameters,  $\zeta_1$ ,  $\zeta_2$ ,  $\phi$  and  $\phi^*$  are positive slack variables, and  $e_1$ ,  $e_2$ , and  $e_3$  are vectors of ones with proper dimensions. To avoid overlapping, positive parameter  $\delta$  must be lower than 1.

The dual formulation of (4) can be expressed as

$$\begin{aligned} \max_{\gamma} \quad & q^T \gamma - \frac{1}{2} \gamma^T H \gamma, \\ \text{subject to} \quad & 0 \leq \gamma \leq F, \end{aligned} \quad (5)$$

where  $Q = [A^T \ -B^T \ C^T \ -C^T]$ ,  $H = Q^T Q$ ,  $q = [e_1^T \ e_2^T \ -\delta e_3^T \ -\delta e_3^T]$ , 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  $f(x) = w^T x + b$  and the decision function can be written as:

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



**Fig. 1.** Geometric representation of K-SVCR method

## 4 Least Square K-SVCR

In this section, we propose a least squares type of K-SVCR method called LSK-SVCR in both linear and nonlinear cases.

### 4.1 Linear Case

We modify the primal problem (4) of K-SVCR as (6), which uses the square of 2-norm of slack variables  $\zeta_1$ ,  $\zeta_2$ ,  $\phi$  and  $\phi^*$  instead of 1-norm slack variables in the objective function and uses equality constraint instead of inequality constraint in K-SVCR. Then, the following minimization problem can be considered:

$$\begin{aligned}
& \min_{w,b,\zeta_1,\zeta_2,\phi,\phi^*} \frac{1}{2} \|w\|^2 + c_1 (\|\zeta_1\|^2 + \|\zeta_2\|^2) + c_2 \|\phi\|^2 + c_3 \|\phi^*\|^2 \quad (6) \\
& \text{subject to } e_1 - (Aw + e_1b) = \zeta_1, \\
& \quad e_2 + (Bw + e_2b) = \zeta_2, \\
& \quad Cw + e_3b - \delta e_3 = \phi^*, \\
& \quad -Cw - e_3b - \delta e_3 = \phi.
\end{aligned}$$

Where  $\zeta_1$ ,  $\zeta_2$ ,  $\phi$  and  $\phi^*$  are positive slack variables, and  $c_1$ ,  $c_2$  and  $c_3$  are penalty parameters and positive parameter  $\delta$  is a lower than 1.

Now by substituting the constraint into the objective function, we have the following unconstrained QPP

$$\begin{aligned}
& \min_{w,b} \frac{1}{2} \|w\|^2 + c_1 \|e_1 - Aw - e_1b\| + c_1 \|e_2 + Bw + e_2b\| \\
& \quad + c_2 \|-Cw - e_3b - \delta e_3\|^2 + c_3 \|Cw + e_3b - \delta e_3\|. \quad (7)
\end{aligned}$$

The objective function of problem (7) 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

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

The above equation can be displayed in the matrix form as

$$\begin{aligned}
& 2c_1 \begin{bmatrix} A^T A & A^T e_1 \\ e_1^T A & e_1^T e_1 \end{bmatrix} \begin{bmatrix} w \\ b \end{bmatrix} + 2c_1 \begin{bmatrix} B^T B & B^T e_2 \\ e_2^T B & e_2^T e_2 \end{bmatrix} \begin{bmatrix} w \\ b \end{bmatrix} + 2(c_2 + c_3) \begin{bmatrix} C^T C & C^T e_3 \\ e_3^T C & e_3^T e_3 \end{bmatrix} \begin{bmatrix} w \\ b \end{bmatrix} \\
& \quad + \begin{bmatrix} 2c_1(-A^T)e_1 + 2c_1B^Te_2 + 2c_2C^T\delta e_3 + 2c_3C^T(-\delta e_3) \\ 2c_1(-e_1^Te_1) + 2c_1e_2^Te_2 + 2c_2\delta e_3^Te_3 + 2c_3e_3^T(-\delta e_3) \end{bmatrix} = 0.
\end{aligned}$$

Therefore  $w$  and  $b$  can be computed as follows:

$$\begin{bmatrix} w \\ b \end{bmatrix} = \begin{bmatrix} c_1(A^T A + B^T B) + (c_2 + c_3)C^T C & c_1(A^T e_1 + B^T e_2) + (c_2 + c_3)C^T e_3 \\ c_1(e_1^T A + e_2^T B) + (c_2 + c_3)e_3^T C & c_1(e_1^T e_1 + e_2^T e_2) + (c_2 + c_3)e_3^T e_3 \end{bmatrix}^{-1} \begin{bmatrix} c_1(-A^T)e_1 + c_1B^Te_2 + c_2C^T\delta e_3 + c_3C^T(-\delta e_3) \\ c_1(-e_1^Te_1) + c_1e_2^Te_2 + c_2\delta e_3^Te_3 + c_3e_3^T(-\delta e_3) \end{bmatrix}.$$

We rewrite it as

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

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 as follows:

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

## 4.2 Nonlinear Case

In the 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, and we would like to find the following kernel surface:

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

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 (4) becomes:

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

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

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

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

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

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.,  $nS \gg nF$ , and in this

case the inverse matrix can be inverted by inverting a smaller  $n^F \times n^F$  matrix by using the Sherman–Morrison–Woodbury (SMW) [5] 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  $A + UV^T$ , where  $A \in R^{n \times n}$  and  $U, V \in R^{n \times K}$ , as follows:

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

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

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

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

where  $Z = \left( c_1 N^T N + (c_2 + c_3) P^T P \right)^{-1}$ . 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  $\alpha I$ , ( $\alpha > 0$  and small enough). Then we have  $Y = \frac{1}{\alpha} (I_{m_3} - P^T (\alpha I + PP^T)^{-1} P)$ .

### 4.3 Decision Rule

The multi-class classification techniques evaluate all training points into the “1-versus-1-versus-rest” structure with ternary output  $\{-1, 0, 1\}$ . For a new testing point  $x_i$ , we predict its class label by the following decision functions:

For linear K-SVCR and LSK-SVCR :

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

For nonlinear K-SVCR and LSK-SVCR :

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

For  $k$ -class classification problem, the “1-versus-1-versus-rest” constructs  $K(K-1)/2$  classifiers in total, and for decision about final class label of testing sample  $x_i$  we get a total votes 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).

## 5 Computational Complexity

In this subsection, we discuss computational complexity of K-SVCR, and LSK-SVCR. In three-class classification problems, suppose the total size of each class is equal to  $m/3$  (where  $m = m_1 + m_2 + m_3$ ). Since samples in the third class are used twice in the constraints of K-SVCR problem, there are  $4m/3$  inequality constraints in total. Therefore 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 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  $m_3 \times m_3$ .

## 6 Numerical Experiments

To assess performance of the proposed method, we apply LSK-SVCR on several UCI benchmark data sets [10] and compare our method with the K-SVCR. All experiments were carried out in Matlab 2019b on a PC with Intel core 2 Quad CPU (2.50 GHZ) and 8 GB RAM. For solving the dual problem of K-SVCR, we used “quadprog.m” function in Matlab. Also, we used 5-fold cross-validation to assess the performance of the algorithms in aspect of accuracy and training time. Note that in 5-fold cross-validation, the dataset 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.

### 6.1 Parameter Selection

It must be noted that the performance of the algorithms depends on the choice of parameters. In the experiments, we opt for the Gaussian kernel function  $k(x_i, x_j) = \exp(-\frac{\|x_i - x_j\|^2}{\gamma^2})$ . The best parameters are then obtained by the grid search method [6, 7].

In this paper, the optimal value for  $c_1, c_2, c_3$ , were selected from the set  $\{2^i | i = -8, -7, \dots, 7, 8\}$ , the parameters of the Gaussian kernel  $\gamma$  were selected from the set  $\{2^i | i = -6, -5, \dots, 5, 6\}$ , and parameter  $\delta$  in K-SVCR and LSK-SVCR was chosen from set  $\{0.1, 0.3, \dots, 0.9\}$ .



**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	150	5	3
Thyroid	215	6	3

**Table 2.** Performance of K-SVCR and LSK-SVCR with Gaussian kernel.

Data set	K-SVCR	K-SVCR	LSK-SVCR	LSK-SVCR
	Acc $\pm$ std	Time (s)	Acc $\pm$ std	Time (s)
Iris	96.54 $\pm$ 2.04	5.27	<b>98.67</b> $\pm$ 1.82	<b>1.54</b>
Balance	94.21 $\pm$ 2.04	89.56	<b>94.89</b> $\pm$ 2.01	<b>4.48</b>
Soybean	<b>100.00</b> $\pm$ 0.00	1.65	<b>100.00</b> $\pm$ 0.00	<b>1.59</b>
Wine	98.81 $\pm$ 2.51	7.06	<b>99.45</b> $\pm$ 1.24	<b>0.79</b>
Breast tissue	<b>47.06</b> $\pm$ 9.73	2.91	46.59 $\pm$ 15.39	<b>1.12</b>
Hayes-Roth	46.33 $\pm$ 12.86	12.70	<b>75.72</b> $\pm$ 8.81	<b>0.58</b>
Ecoli	77.36 $\pm$ 4.28	21.23	<b>89.01</b> $\pm$ 5.89	<b>5.57</b>
Teaching	63.68 $\pm$ 5.58	10.63	<b>70.19</b> $\pm$ 7.46	<b>0.96</b>
Thyroid	83.25 $\pm$ 6.02	21.26	<b>93.49</b> $\pm$ 2.55	<b>1.38</b>

## 6.2 UCI Data Sets

In this subsection, to compare the performance of K-SVCR with LSK-SVCR, we ran these algorithms on several benchmark data sets from UCI machine learning repository [10], which are described in Table 1.

To analyse the performance of the K-SVCR and LSK-SVCR algorithms, Table 2 shows a comparison of classification accuracy and computational time for K-SVCR and LSK-SVCR on nine benchmark datasets available at the UCI machine learning repository. This table indicates that for Iris dataset, the accuracy of LSK-SVCR (accuracy: 98.67, time: 0.03 s) was higher than K-SVCR (accuracy: 96.54, time: 3.51 s), so our proposed method was more accurate and faster than original K-SVCR. A similar discussion can be made for Balance, Soybean, Wine, Brest Tissue, Hayes-Roth, Ecoli, Teaching, and Thyroid datasets.

The analysis of experimental results on nine UCI datasets revealed that the performance of LSK-SVCR was slightly better than the original K-SVCR. We should note that for Brest Tissue, although the K-SVCR is a little more accurate than LSK-SVCR, the LSK-SVCR is faster. Therefore, according to the experimental results in Table 2, LSK-SVCR not only yielded higher prediction accuracy but also had lower computational times.

## 7 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 as LSK-SVCR for multi-class classification. Our proposed method leads to solving a simple system of linear equations instead of solving a hard QPP in K-SVCR. Kindly provide the page range for Ref. [2]. The K-SVCR and LSK-SVCR evaluates all training data into “1-versus-a-versus-rest” structure with ternary output  $\{-1, 0, +1\}$ .

The computational results performed on several UCI data set demonstrate that, compared to K-SVCR, the proposed LSK-SVCR has better efficiency in terms of accuracy and training time.

## References

1. Angulo, C., Català, A.: K-SVCR. a multi-class support vector machine. In: López de Mántaras, R., Plaza, E. (eds.) ECML 2000. LNCS (LNAI), vol. 1810, pp. 31–38. Springer, Heidelberg (2000). [https://doi.org/10.1007/3-540-45164-1\\_4](https://doi.org/10.1007/3-540-45164-1_4)
2. Bazikar, F., Ketabchi, S., Moosaei, H.: DC programming and DCA for parametric-margin  $\nu$ -support vector machine. *Appl. Intell.* 1–12 (2020)
3. Boser, B.E., Guyon, I.M., Vapnik, V.N.: A training algorithm for optimal margin classifiers. In: Proceedings of the fifth annual workshop on Computational learning theory. COLT 1992, pp. 144–152, Association for Computing Machinery, New York (1992). <https://doi.org/10.1145/130385.130401>
4. Cortes, C., Vapnik, V.: Support-vector networks. *Mach. Learn.* **20**(3), 273–297 (1995). <https://doi.org/10.1007/BF00994018>
5. Golub, G.H., Van Loan, C.F.: *Matrix Computations*. Johns Hopkins University Press, Baltimore (2012)
6. Hsu, C.W., Chang, C.C., Lin, C.J., et al.: *A practical guide to support vector classification* (2003)
7. Ketabchi, S., Moosaei, H., Razzaghi, M., Pardalos, P.M.: An improvement on parametric  $\nu$ -support vector algorithm for classification. *Ann. Oper. Res.* **276**(1–2), 155–168 (2019)
8. Kumar, M.A., Gopal, M.: Least squares twin support vector machines for pattern classification. *Expert Syst. Appl.* **36**(4), 7535–7543 (2009). <https://doi.org/10.1016/j.eswa.2008.09.066>
9. Lee, Y.J., Mangasarian, O.: SSVN: a smooth support vector machine for classification. *Comput. Optim. Appl.* **20**(1), 5–22 (2001). <https://doi.org/10.1023/A:1011215321374>

10. Lichman, M.: UCI machine learning repository (2013). <http://archive.ics.uci.edu/ml>
11. Tang, L., Tian, Y., Pardalos, P.M.: A novel perspective on multiclass classification: regular simplex support vector machine. *Inf. Sci.* **480**, 324–338 (2019)
12. Tang, L., Tian, Y., Yang, C., Pardalos, P.M.: Ramp-loss nonparallel support vector regression: robust, sparse and scalable approximation. *Knowl.-Based Syst.* **147**, 55–67 (2018)
13. Vapnik, V.N., Chervonenkis, A.J.: *Theory of Pattern Recognition*. Nauka (1974)