Neural Processing Letters (2021) 53:517–533
https://doi.org/10.1007/s11063-020-10393-7

Multi-class support vector machine based on the minimization of class variance

Zhiqiang Zhang[1] · Zeqian Xu[2] · Junyan Tan[2] · Hui Zou[2]

Accepted: 15 November 2020 / Published online: 3 January 2021

Abstract
Since the existing methods can not balance the sufficient use of information and the scale of the optimization problem, a new method for multi class classification problem is proposed, which is called multi-class support vector machine based on the minimization of class variance (MCVMSVM for short). MCVMSVM adopts the idea of semi-supervised learning and transfers the K-class problem to K(K − 1)/2 binary classification problems. For each binary classification problem, a new SVM with a mixed regularization term which considers the margin and the distribution of examples is proposed. MCVMSVM can utilize the information of all examples without increasing the scale of the optimization problem. The performance of MCVMSVM on UCI and NDC datasets is the best compared with other methods, that means MCVMSVM is more effective.

Keywords Multi-class problems · Support vector machines · Class variance

1 Introduction

Support vector machine (SVM) proposed byVapnik [1, 2] is a general learning method based on statistical learning theory (SLT). Support vector machines show many good properties on two classification problems [3]. While, in the real word, there are a lot of multi-class classification problems, such as text recognition, image classification, speech recognition, face recognition and so on. Therefore, more and more researchers devote to the study of multi-class classification problems. There are a lot of classification algorithms in data

✉ Junyan Tan
tanjunyan0@126.com

✉ Hui Zou
Huizou@cau.edu.cn

Zhiqiang Zhang
sunwithmoon@bit.edu.cn

Zeqian Xu
2583673301@qq.com

[1] School of Mechanical Engineering, Beijing Institute of Technology, Beijing 100081, China

[2] College of Science, China Agricultural University, Beijing 100083, China

mining, such as neural network, k-nearest neighbor, decision tree, support vector machine, etc. Although Neural network has better classification accuracy, it has some shortcomings. Neural network needs a large number of parameters, such as the initial value of network topology, weight and threshold; it can not observe the learning process between them, and the output results are difficult to explain, which will affect the credibility and acceptability of the results [16–18]. K-nearest neighbor is the easiest method, it is suitable for the data with large sample size, while it is easy to produce misclassifications for those data with small sample size. Decision tree is easy to understand and the output is more comprehensive, while over-fitting is easy to occur in decision tree and the decision tree ignores the correlation between attributes. Support vector machine (SVM) is more classical and representative. SVM can improve generalization performance and solve high dimensional, nonlinear problems, it can also avoid the problems of neural network structure selection and local minima.

Most of the currently existed multi classification methods based on support vector machine adopt the decomposition-reconstruction strategy. These kinds of methods transfer the K-class problem to a series of binary-classification problems and the discrimination functions can be obtained by solving a series of binary classification problems [4–8, 12]. There are totally three "decomposition reconstruction" strategies. The first one is "1-versus-rest" strategy [9] which constructs K classifiers. For the k-th problem, the examples belong to the k-th class are viewed as positive examples, the examples in the other classes are viewed as negative examples, then binary SVM is used to build the classifier. A new example is assigned a label according to the winner-takes-all scheme. Although "1-versus-rest" strategy uses the information of all examples in the training set, almost all the binary problems are unbalanced because the number of negative examples is much larger than the number of positive examples. The second one is "1-versus-1" strategy [10] which constructs binary classifiers and each classifier involves only two class examples with abandoning the other K-2 classes. For a new example, its label is decided by the voting scheme. In "1-versus-1" strategy, the information of the remaining examples is omitted when constructing each binary classifier. The third one is "1-versus-1-verus-rest" which is the improvement of "1-versus-1" strategy, called K-SVCR [11]. KSVCR constructs K(K−1)/2 sub-classifiers. In the process of constructing the sub-classifiers, the original training set is divided into three parts: positive examples (the examples belonging to i-th class in original training set), negative examples (the examples belonging to j-th class in original training set) and rest examples (all the other examples belonging to the original training set except the i-th class and the j-th class). KSVCR aims at finding the sub-classifier that can classify the positive and negative examples correctly and the rest examples are restricted within an ε-insensitive band. For a new example, its label is also decided by the voting scheme. Compared with "1-versus-1" support vector machine, KSVCR makes full use of the information of training samples. Although KSVCR does not loss any information of the examples, the scale of its optimization problems increase largely and thus it's difficult to be solved.

In order to overcome the shortcomings of "1-versus-1" SVM and KSVCR, a new optimal method which incorporates the idea of semi-supervised learning is proposed in this paper. Our method is based on 1-versus-1-versus rest framework, i.e. the K-class problem is transferred to K(K−1)/2 binary classification problems. When solving the k-th binary problem, the training set is divided into three parts: positive examples (the examples in the i-th class), negative examples (the examples in the j-th class) and rest examples (all examples except the positive and negative examples). For each sub-classifier, the classification hyperplane is required to separate the positive and negative examples correctly and the projections of rest examples on the normal vector of the classification hyperplane is focused around zero. So, a new SVM with a mixed regularization term is proposed. The new regularization term considers both the margin between positive and negative examples and the distribution of the rest examples. Compared with currently existed methods, the new method uses all the information of examples without abandon any examples.

This paper is organized as follows. In Sect. 2, we give a brief introduction of SVM, minimization class variance SVM (MCVSVM) and K-SVCR. Our new method MCVMSVM is proposed in Sect. 3. The numerical experiments on several real datasets are conducted in Sect. 4. We summarize this paper in Sect. 5.

2 Related Works

Some works closely related to this paper, such as SVM, MCVSVM and K-SVCR are introduced in this section.

2.1 Support Vector Machine

Consider the binary classification problem with the training set

$$T= \{(x_1, y_1), \dots, (x_l, y_l)\} \tag{1}$$

where $x_i \in R^n$ is an example,$y_i \in \{+1, -1\}$ is the corresponding label.

SVM constructs a hyperplane $w \cdot x + b = 0$ by maximizing the margin between two support hyperplanes ($w \cdot x + b = -1$ and $w \cdot x + b = 1$). Furthermore, the positive examples should be located above the positive hyperplane $w \cdot x + b = 1$ and the negative examples should be located under the negative hyperplane $w \cdot x + b = -1$. SVM gets the classification hyperplane by solving the following quadratic program:

$$\begin{aligned} &\min_{w,b,\xi} \frac{1}{2}||w||^2 + C\sum_{i=1}^{l} \xi_i \\ &y_i(w \cdot x_i + b) \ge 1 - \xi_i, i = 1, \dots, l, \\ &\xi_i \ge 0, i = 1, \dots, l, \end{aligned} \tag{2}$$

where $C > 0$ is the parameter that can balance the margin and classification error, $\xi_i \ge 0$ is slack variable measuring the classification loss of examples. More details of SVM can be found in the literature [13].

2.2 Minimization Class Variance Support Vector Machine

For the training set (1), the hyperplane generated by the traditional support vector machine is determined by a small number of support vectors, and it ignores the structural characteristics of the training examples. Therefore, the SVM can not play a good classification effect for some data, as shown in Fig. 1. Although the SVM can find a hyperplane to separates the positive and negative training examples correctly, the hyperplane derived by SVM does not represent the distribution trend of the data well. Therefore, new point "*" may be misclassified by SVM. So, minimization class variance support vector machine (MCVSVM) is proposed by

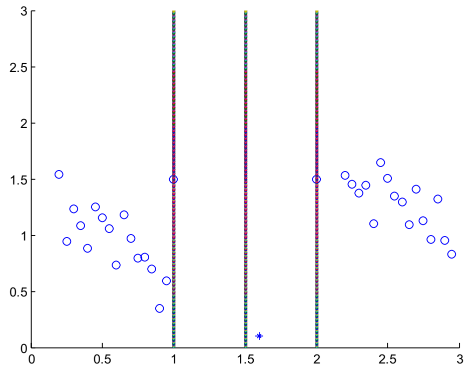


Fig. 1 The classification effect of SVM

the literature [14]. MCVSVM adopts an intra class divergence matrix to keep the distribution between data. The optimization problem is as follow:

$$\begin{aligned}&\min_{w,b}\frac{1}{2}w^TS_ww+C\sum_{i=1}^{l}\xi_i\\&y_i(x_i^Tw+b)\geq 1-\xi_i,\\&\xi_i\geq 0,\end{aligned} \tag{3}$$

where matrix S_w is the sum of the divergence matrix of the positive examples and negative examples. $C > 0$ is the penalty parameter, ξ_i is the slack variable measuring the classification loss of examples.

Now, let's give the geometric interpretation of the optimization problem (3). Minimizing the first term in the objective function means the intra-class examples' projections on the normal vector of the hyperplane should be as close as possible; minimizing the second term is to minimize the loss of misclassification; the constraints means the examples belonging to positive and negative class should be located on both sides of hyperplanes $(w\cdot x+b)=1$ and $(w\cdot x+b)=-1$. For a new example, it's label is decided by the following discriminant function:

$$f(x)=\operatorname{sgn}(S_w^{-1}\sum_{i=1}^{l}\alpha_iy_i\langle x_i\cdot x\rangle+b). \tag{4}$$

Since MCVSVM considers the distribution of examples, the hyperplane obtained by MCVSVM is better which is shown in Fig. 2. There is a serious shortcoming in MCVMSVM, when the dimension of examples is much larger than the number of examples, the inverse of matrix S_w may not exist.

2.3 Support Vector Classification-Regression Machine for K-Class (K-SVCR)

Given the training set

$$T=\{(x_1,y_1),\ldots,(x_l,y_l)\} \tag{5}$$

where $x_i\in R^n$ is an example,$y_i\in\{1,2,\ldots,K\}$ is the corresponding label of x_i.

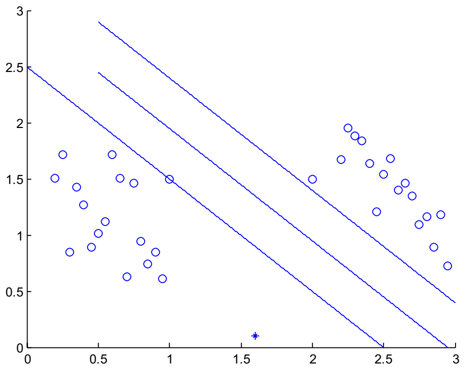


Fig. 2 The classification effect of MCVSVM

K-SVCR constructs K(K − 1)/2 sub-classifiers for the K-class classification problem. When constructing the sub-classifiers, the training set is divided into three parts: positive class, negative class and rest class. A mixed classification and regression machine is formulated. Firstly, the classification hyperplane $w \cdot x + b = 0$ should separates the training examples belonging to positive and negative classes as correct as possible. Secondly, the examples of the rest class should be in the ε-insensitive band (0 < ε < 1) of the hyperplane $w \cdot x + b = 0$. According to the idea mentioned above, the optimization problem of K-SVCR is formulated as follows:

$$\begin{aligned} &\min_{w,b,\xi,\varphi_i,\varphi_i^*} \frac{1}{2}||w||^2 + C\sum_{i=1}^{l_1+l_2}\xi_i + D\sum_{i=l_1+l_2+1}^{l}(\varphi_i + \varphi_i^*) \\ &s.t. y_i(w \cdot x_i + b) \geq 1 - \xi_i, i = 1, ..., l_1 + l_2, \\ &-\varepsilon - \varphi_i^* \leq w \cdot x_i + b \leq \varepsilon + \varphi_i, i = l_1 + l_2 + 1, ..., l, \\ &\xi_i \geq 0, i = 1, ..., l_1 + l_2, \end{aligned} \tag{6}$$

where ξ_i, φ_i, φ_i^* are slack variables measuring the loss of misclassification, $C>0$ and $D>0$ are parameters that can balance the margin, classification error and regression error. l_1 represents the number of examples in positive class. l_2 represent the number of examples in negative class.

Now we explain the optimization problem in detail. Minimizing the first term in (6) is to maximize the margin between two support hyperplanes $w \cdot x + b = -1$ and $w \cdot x + b = 1$. The second term is to minimize the sum of classification errors of the examples belonging to the positive and negative class, the third term is to minimizes the sum of regression errors of the examples belonging to the rest class. The first constraints mean the examples belonging to the positive class should be above the positive support hyperplane $w \cdot x + b = 1$ and the examples belonging to the negative class be under the negative support hyperplane $w \cdot x + b = -1$.The second constraints mean the examples belonging to the rest class should be in the ε-insensitive band of $w \cdot x + b = 0$. It is obviously that there are a lot of constraints in the optimization problem of (6). So, it takes a lot of time to be solved.

3 Minimization class variance SVM for multi classification SVM (MCVMSVM)

Now, we are in the position to put out our MCVMSVM including the linear MCVMSVM and nonlinear MCVMSVM.

3.1 The linear MCVMSVM

Our new method takes "1-versus-1" strategy and totally K(K − 1)/2 classifiers are constructed, and the final decision is made by the voting scheme. When constructing the classifiers, the training set (5) is firstly divided into three parts: the examples belonging to the i-th class are viewed as positive class with label 1, the examples belonging to the j-th class are viewed as negative class with label -1 and the examples belonging to the rest class except the i-th and j-th class are viewed as zero class with label 0. For the re-divided training set, MCVMSVM wishes to find a hyperplane so that the positive and negative examples are located on both sides of $w \cdot x + b = 1$ and $w \cdot x + b = -1$, and the projections of zero-class examples on the normal vector w are concentrated as close as possible to zero. The geometric explanation can be seen clearly in Fig. 3. The negative examples locate under the plane "$w \cdot x + b = -1$"; the positive examples locate above the plane $w \cdot x + b = 1$; the projections of zero-class examples on w are closed to zero. The hyperplane obtained by MCVMSVM should be the middle solid line H_1 but not H_0 which is obtained by 1-versus-1 SVM.

According to the training set (5), the matrix composed of class i is constructed as the positive sample matrix A_i:

$$A_i = [x_{i1}, x_{i2}, \dots, x_{il_i}], i = 1, \dots, K, A_i \in R^{l_i \times n}, \tag{7}$$

the matrix composed of class j is constructed as the positive sample matrix B_j:

$$B_j = [x_{j1}, x_{j2}, \dots, x_{jl_j}], j = 1, \dots, K, B_j \in R^{l_j \times n} \tag{8}$$

The matrix composed of zerois denoted as $C_{i,j}$ Define the set $C_{i,j}^k$ which is composed of the examples belonging to the k-th classas follows:

$$C_{i,j}^k = \{x_{k1}, x_{k2}, \dots, x_{kl_k}, k = 1, \dots, K, k \neq i, k \neq j\} \tag{9}$$

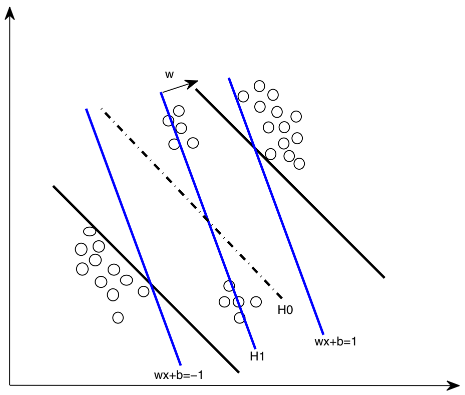


Fig. 3 An example learned by the linear MCVMSVM

Let $m_{i,j}^k$ denote the center point of $C_{i,j}^k$.

The intra class scatter matrix of the zero class is S_w:

$$S_w = \sum_{k=1}^{K-2} \sum_{x \in C_{i,j}^k} (x - m_{i,j}^k)(x - m_{i,j}^k)^T, \tag{10}$$

Suppose the i-th hyperplane is $(w_{i,j} \cdot x) + b_{i,j} = 0$, then the i-th problem is constructed as follows:

$$\begin{aligned} &\min_{w_{i,j}, b_{i,j}} \frac{1}{2}||w_{i,j}||^2 + De_{i,j}^T \xi_{i,j} + \frac{\lambda}{2} w_{i,j}^T S_w w_{i,j} \\ &(A_i w_{i,j} + e_i b_{i,j}) + \xi_{i,j} \geq e_i, \\ &-(B_j w_{i,j} + e_j b_{i,j}) + \xi_{i,j} \geq e_j, \\ &\xi_{i,j} \geq 0, \end{aligned} \tag{11}$$

where, $D > 0$, $\lambda > 0$ are the parameters, $e_{i,j} = [e_i, e_j]$ and e_i, e_j are the vector of ones with appropriate dimensions, $\xi_{i,j}$ are slack variables measuring the misclassification。

Minimizing the first term in the objective function of problem (11) is to maximizes the interval between the support hyperplanes. Minimizing the second term is to minimize the sum of classification errors. Minimizing the last term is to minimize the projection of examples in the zero class on the normal vector direction of hyperplane. The constraints guarantee the positive should be located above $w \cdot x + b = 1$ and the negative examples should be located under $w \cdot x + b = -1$. Next, we construct the dual problem of (11). The Lagrange function of problem (11) is formulated firstly,

$$\begin{aligned} &L(w_{i,j}, b_{i,j}, \xi_{i,j}, , \alpha_{i,j}, \mu_{i,j}, \beta_{i,j}, \eta_{i,j}) \\ &= De_{i,j}^T \xi_{i,j} + \frac{1}{2}||w_{i,j}||^2 + \frac{\lambda}{2} w_{i,j}^T S_w w_{i,j} \\ &-\alpha_{i,j}^T[(A_i w_{i,j} + e_i b_{i,j}) + \xi_{i,j} - e_i] - \mu_{i,j}^T \xi_{i,j} \\ &+\beta_{i,j}^T[(B_j w_{i,j} + e_j b_{i,j}) - \xi_{i,j} + e_j] - \eta_{i,j}^T \xi_{i,j} \end{aligned} \tag{12}$$

Then, by the Karuch-Kuhn-Tucker (KKT) conditions, we get

$$\frac{\partial L}{\partial w_{i,j}} = w_{i,j} + S_w w_{i,j} - A_i^T \alpha_{i,j} + B_j^T \beta_{i,j} = 0 \Rightarrow w_{i,j} = (I + \lambda S_w)^{-1}(A_i^T \alpha_{i,j} - B_j^T \beta_{i,j}), \tag{13}$$

$$\frac{\partial L}{\partial b_{i,j}} = -e_i^T \alpha_{i,j} + e_j^T \beta_{i,j} = 0, \tag{14}$$

$$\frac{\partial L}{\partial \xi_{i,j}} = \begin{bmatrix} De_i \\ De_j \end{bmatrix} - \begin{bmatrix} \alpha_{i,j}^T e_i \\ \beta_{i,j}^T e_j \end{bmatrix} + \begin{bmatrix} \mu_{i,j}^T e_i \\ \eta_{i,j}^T e_j \end{bmatrix} = 0 \Rightarrow 0 \leq \alpha_{i,j}, \beta_{i,j} \leq D, \tag{15}$$

Substituting the Eq. (13)–(15)s into (12), the dual problem of problem (11) can be abbreviated as:

$$\max_{\alpha_{i,j},\beta_{i,j}}[e_i^T,e_j^T][\alpha_{i,j}^T,\beta_{i,j}^T]^T-\frac{1}{2}[\alpha_{i,j}^T,\beta_{i,j}^T]^T[A_i^T,-B_j^T]^T(I+\lambda S_w)^{-1}[A_i^T,-B_j^T]^T[\alpha_{i,j}^T,\beta_{i,j}^T]^T$$
$$0\le\alpha_{i,j},\beta_{i,j}\le D, \tag{16}$$

where $\alpha_{i,j},\beta_{i,j}$ are slack variables. Solving the dual problem (16), we get the optimal solution $[\alpha_k^*,\beta_k^*]$, and randomly select a support vector label y_k, then the optimal solution of the primal problem (11) is

$$\begin{aligned}w_{i,j}^*&=(I+\lambda S_w)^{-1}[A_i^T\alpha_{i,j}^*-B_j^T\beta_{i,j}^*],\\ b_{i,j}^*&=y_k-A_i^k(I+\lambda S_w)^{-1}[A_i^T\alpha_{i,j}^*-B_j^T\beta_{i,j}^*],\end{aligned} \tag{17}$$

So, the function of the k-th hyperplane can be expressed as:

$$f_{ij}(x)=[x,1][w_{i,j}^{*T},b_{i,j}^{*T}]^T=0. \tag{18}$$

Totally $K(K-1)/2$ hyperplanes like (18) are constructed in our MCVMSVM and thus the $K(K-1)/2$ decision functions are defined as follows:

$$\mathrm{sgn}(f_{i,j}(x))=\begin{cases}+1,[x,1][w_{i,j}^{*T},b_{i,j}^{*T}]^T>0\,i,j=1,\dots,K;i<j,\\-1,[x,1][w_{i,j}^{*T},b_{i,j}^{*T}]^T<0\,i,j=1,\dots,K;i<j,\end{cases} \tag{19}$$

For a new example x, a vote is given to the class (i) or class (j) based on which condition is satisfied. Finally, the example x is assigned to the class label that gets the most votes.

3.2 The nonlinear classifier

In this section, the linear MCVMSVM is extended to the nonlinear case using kernel trick. The training set (5) is mapped into the high-dimensional feature space by introducing a nonlinear mapping ϕ:

$$x\to\phi(x) \tag{20}$$

Suppose that the normal vector of the hyperplane in the feature space can be expressed as:

$$w=\sum_{i=1}^{l_1+l_{12}}\mu_i\phi\left(x_i\right) \tag{21}$$

Then we define the intra-class divergence as:

$$\begin{aligned}w^TS_ww&=w^T\sum_{k=1}^{K-2}\sum_{x\in C_{i,j}}(\phi(x)-m_{i,j}^k)(\phi(x)-m_{i,j}^k)^Tw\\&=\sum_{k=1}^{K-2}[w^T\sum_{x\in C_{i,j}^k}(\phi(x)-m_{i,j}^k)(\phi(x)-m_{i,j}^k)^Tw].\end{aligned} \tag{22}$$

We rewrite the term in (22) as follows:

$$
\begin{aligned}
& w^T \sum_{x\in C_{i,j}^k} (\varphi(x) - m_{i,j}^k)(\varphi(x) - m_{i,j}^k)^T w \\
& = w^T \sum_{x\in C_{i,j}^k} \left[(\varphi(x)\varphi(x)^T - m_{i,j}^k \varphi(x)^T - \varphi(x) m_{i,j}^{kT} + m_{i,j}^k m_{i,j}^{kT})^T\right] w \\
& = w^T \left[\sum_{x\in C_{i,j}^k} \varphi(x)\varphi(x)^T - m_{i,j}^k \sum_{x\in C_{i,j}^k} \varphi(x)^T - \sum_{x\in C_{i,j}^k} (\varphi(x) m_{i,j}^{kT} + l_r^k m_{i,j}^k m_{i,j}^{kT})^T\right] w \\
& = w^T \left[\sum_{x\in C_{i,j}^k} \varphi(x)\varphi(x)^T - l_r^k m_{i,j}^k m_{i,j}^{kT}\right] w \\
& = \sum_{x\in C_{i,j}^k} w^T \varphi(x)\varphi(x)^T w - w^T l_r^k m_{i,j}^k m_{i,j}^{kT} w \\
& = \mu^T K(C_{i,j}^k, S) K(C_{i,j}^k, S)^T \mu - \mu^T K(C_{i,j}^k, S) L_{l_r}^k K(C_{i,j}^k, S)^T \mu \\
& = \mu^T K(C_{i,j}^k, S)(I_{l_r}^k - L_{l_r}^k) K(C_{i,j}^k, S)^T \mu
\end{aligned}
\tag{23}
$$

where $\mu = \left[\mu_1, \ldots \mu_{l1+l2}\right]^{\mathrm{T}}$, l_r^k is the number of $C_{i,j}^k$, $I_{l_r}^k$ is l_r^k-order identity matrix, $L_{l_r}^k$ is l_r^k-order matrix of all elements being equal to the square of $\frac{1}{l_r^k}$,$S = [A_i^T, B_j^T]$. The k-th optimization problem nonlinear MCVMSVM is:

$$
\begin{aligned}
& \min_{w,b,\xi_{i,j}} \frac{1}{2}\mu^T K(S,S)\mu + De_{i,j}^T \xi_{i,j} + \frac{\lambda}{2}\sum_{k=1}^{K-2} \mu^T K(C_{i,j}^k, S)(I_{i,j}^k - L_{i,j}^k) K(C_{i,j}^k, S)^T \mu \\
& (K(A_i^T, S^T)\mu + e_i b_{i,j}) + \xi_{i,j} \ge e_i, \\
& -(K(B_j^T, S^T)\mu + e_j b_{i,j}) + \xi_{i,j} \ge e_j, \\
& \xi_{i,j} \ge 0
\end{aligned}
\tag{24}
$$

where $D > 0, \lambda > 0$ are the parameters, $e_{i,j} = [e_i, e_j]$ and e_i, e_j are the vector of ones with appropriate dimensions, $\xi_{i,j}$ are slack variables measuring the misclassification. Similar to thelinear case, we get the optimal solution of (24) by solving the following dual problem:

$$
\begin{aligned}
& \max_{\alpha_{i,j},\beta_{i,j}} [e_i^T, e_j^T][\alpha_{i,j}^T, \beta_{i,j}^T]^T \\
& -\frac{1}{2}[\alpha_{i,j}^T, \beta_{i,j}^T]^T [K_i^T, -K_j^T](G + \lambda \sum_{k=1}^{K-2} K_{i,j}^k L_{i,j}^k K_{i,j}^{kT})^{-1} [K_i^T, -K_j^T]^T [\alpha_{i,j}^T, \beta_{i,j}^T] \\
& 0 \le \alpha_{i,j}, \beta_{i,j} \le D,
\end{aligned}
\tag{25}
$$

where $G = K(S^{\mathrm{T}}, S^{\mathrm{T}})$, $K_i = K(A_i^T, S^T)$, $K_j = K(B_j^T, S^T)$, $K_{ij}^k = K(C_{ij}^{kT}, S^T)$, $L_{ij}^k = I_{l_r}^k - L_{l_r}^k$. Solving the dual problem (25), we get the optimal solution $[\alpha_k^*, \beta_k^*]$, and y_k is the label of a support vectorx_k, then the optimal solution of the primal problem (24) is

$$
\mu^* = (G + \lambda \sum_{k=1}^{K-2} K_{i,j}^k L_{i,j}^k K_{i,j}^{kT})^{-1} (\alpha_{i,j}^{*T} K_i - \beta_{i,j}^{*T} K_j)
\tag{26}
$$

$$b^* = y_k^* - \mu^* K(x_k, S) \tag{27}$$

Then the function of the i-th hyperplane can be expressed as:

$$f_{i,j}(x) = \mu^* K(x^T, S^T) + b^* = 0 \tag{28}$$

Just as the linear case, the decision function that classifies class (i) and class (j) is designed as

$$\mathrm{sgn}(f_{i,j}(x)) = \begin{cases} +1, \mu^* K(x^T, S^T) + b^* > 0\, i,j = 1, \dots, K; i < j, \\ -1, \mu^* K(x^T, S^T) + b^* < 0\, i,j = 1, \dots, K; i < j, \end{cases} \tag{29}$$

For a new example x, a vote is given to the class (i) or class (j) based on which condition is satisfied. Finally, the example x is assigned to the class label that gets the most votes.

3.3 Comparing with other methods

Comparing with "1-versus-1″ SVM. Our method needs to construct K(K − 1)/2 hyperplanes, this is just the same as "1-versus-1″ SVM. And the computational complexity of our method is almost the same as "1-versus-1" SVM. The information of all examples is used in our method. While, only the positive and negative examples are used in "1-versus-1" SVM. It is clearly that our method makes full use of the information of rest class compared with "1-versus-1" SVM.

Comparing with "1-versus-rest" SVM. "1-versus-rest" SVM and our MCVMSVM all make full use of the information of all examples. MCVMSVM makes more detailed use of information because it divides all examples into three parts (positive, negative and rest), while "1-versus-rest" SVM makes rough use of information because it divides all examples into two parts (positive and negative).

Comparing with K-SVCR. The "1-versus-1-versus-rest" strategy is used in both our MCVMSVM and KSVCR. But the processing method is quite different. A square term with intra class divergence of rest class is introduced in MCVMSVM, while a lot of constraints containing the rest examples are added in KSVCR. So, the computational complexity of MCVMSVM is much less than the computational complexity of KSVCR, and the training time of MCVMSVM is shorter than that of KSVCR. The gap is more obvious when the number of categories is large which can be seen clearly in Fig. 4.

4 Numerical experiments

To evaluate the performance of the new algorithm, several UCI [14] datasets and large scale NDC datasets are used in numerical experiments (Table 3). All experiments have been implemented in Matlab 2015b on a PC with system configuration Intel core 7CPU at 3.5 GHz with 4 GB of RAM, and Windows 10 operating system. For the nonlinear case, RBF kernel is used. The parameters D are selected from the set $[2^{-8}, 2^{-7}, \dots, 2^{7}, 2^{8}]$ and γ

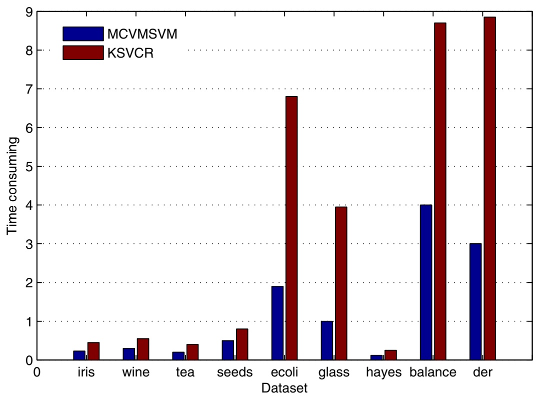
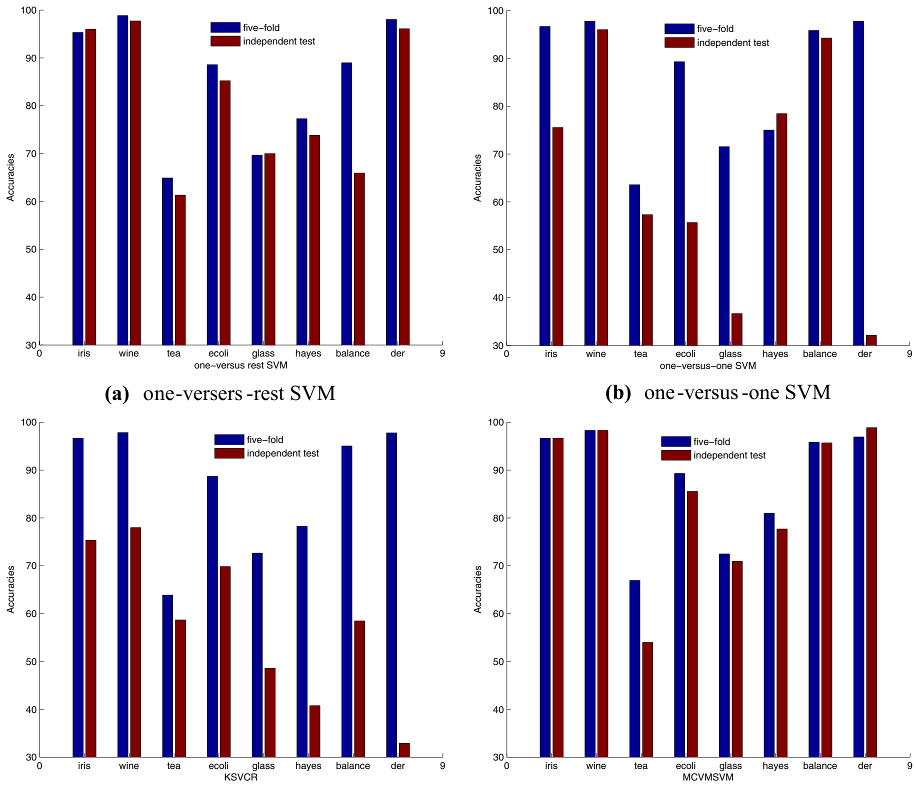


Fig. 4 The comparison of the time consuming of MCVMSVM and KSVCR on the nine experimental data sets



(a) one-versers -rest SVM **(b)** one -versus -one SVM

(c) KSVCR **(d)** MCVMSVM

Fig. 5 The comparison of generality on UCI data sets. "Blue column" represents the accuracies of five-fold validation, "red column" represents the accuracies of independent test

is selected from $\gamma \in [0.1, 0.2, \ldots, 2]$. Our MCVMSVM is compared the popular multiclass classification methods: 1-v-1 SVM,1-r SVM and KSVCR.

4.1 Benchmark datasets

In this section, we compare our MCVMSVM and other three multiclass classification methods on UCI data sets. The five-fold cross validations are conducted on 9 UCI datasets, i.e. the data set is randomly split int five parts, four of them is the training set that helps us to learn the optimal hyperplane, one of them is the validation data set that test the hyperplane got by the training set. The five-fold validation results are shown in Table 1. We can see that our MCVMSVM performs better than the other four methods on five data sets: ‘wine’, ‘tea’, ‘seeds’, ‘glass’, ‘hayes’. On these five data sets, the accuracy on the data sets‘tea’ and ‘hayes’ are improved by up to 3.3% and 6.8% using MCVMSVM. On the datasets‘iris’, ‘balance’ and ‘ecoli’, the accuracy of RBSVM is similar to that of the other method. Only on the data sets‘der’, the accuracy of MCVMSVM is 0.8% lower than other methods. On the whole, Table 1 shows that MCVMSVM has the best classification results on the seven datasets. In addition, the last two rows in the table are the average classification accuracy and mean standard deviation of the three classification models on the nine sets of datasets. Obviously, MCVMSVM has the highest accuracy and the lowest standard deviation, which indicates that MCVMSVM can not only improve the classification accuracy, but also have a more stable classification effect. This is because MCVMSVM not only considers the samples of positive and negative class, but also considers the structure information of the rest samples, it makes full use of the information provided by the dataset. So, MCVMSVM has a better classification accuracy.

The classification effect is also compared on the independent test set. 30% of the datasets are randomly selected as the test set, the remaining 70% are used as the trains set. We firstly select the optimal parameters and train the classifiers using the training set by five-fold validation. Then we test the classifiers on the test set. Table 2 shows the comparison results. It is clearly that MCVMSVM performs better than all the other methods. Figure 5 compares the accuracies between five-fold cross validation and independent test. It is clearly the accuracies drop more than 50% in other three methods. While the accuracy got by MCVMSVM almost is unchanged. So, the generalization of MCVMSVM is better.

Table 1 Five-fold cross validation comparison on UCI data sets

dataset	1-v-1SVM	1-rSVM	KSVCR	MCVMSVM
iris	96.67 ± 3.65	95.33 ± 4.52	96.67 ± 2.11	96.67 ± 3.26
wine	97.75 ± 2.09	98.86 ± 1.40	97.84 ± 2.37	98.3 ± 1.39
tea	63.58 ± 5.47	94.90 ± 4.31	63.87 ± 3.76	66.94 ± 5.71
seeds	91.9 ± 3.87	91.90 ± 3.56	92.12 ± 5.47	92.38 ± 1.78
ecoli	89.30 ± 2.55	88.58 ± 2.82	88.68 ± 1.35	89.30 ± 2.55
glass	71.55 ± 4.72	69.67 ± 4.09	72.64 ± 3.28	72.47 ± 3.91
hayes	75.01 ± 3.74	77.29 ± 4.05	78.24 ± 2.31	81.82 ± 1.43
balance	95.84 ± 2.23	88.96 ± 1.18	95.04 ± 1.87	95.84 ± 2.10
der	97.76 ± 1.41	98.04 ± 0.99	97.76 ± 1.43	96.92 ± 1.64
Average ACC	86.60	85.95	86.93	87.84
Average std	3.64	2.34	2.67	2.64

Table 2 The comparison on the independent test set

dataset	1-v-1SVM	1-r SVM	KSVCR	MCVMSVM
iris	75.33 ± 11.46	96 ± 2.49	75.33 ± 11.46	96.67 ± 2.11
wine	96 ± 2.28	97.71 ± 1.14	78 ± 4.64	98.29 ± 1.39
tea	57.33 ± 5.73	61.33 ± 6.53	58.67 ± 6.52	54.00 ± 10.5
ecoli	55.69 ± 7.24	85.23 ± 5.02	69.84 ± 0.75	85.54 ± 2.85
glass	36.67 ± 6.32	70 ± 7.91	48.57 ± 9.71	70.95 ± 5.08
hayes	78.46 ± 10.74	73.84 ± 4.48	40.77 ± 7.92	77.69 ± 8.21
balance	94.24 ± 4.59	65.92 ± 2.28	58.48 ± 3.81	95.68 ± 1.48
der	32.11 ± 6.13	96.09 ± 2.07	32.93 ± 5.44	98.87 ± 0.565
Average ACC	65.73	80.77	57.82	84.71
Average std	6.81	3.99	6.28	4.02

We also compare the time consuming by KSVCR and MCVMSVM on 9 UCI data sets, the results are shown in Fig. 4. It is clearly that MCVMSVM is much faster than KSVCR. Because the constraint condition of the MCVMSVM model is much less than that of KSVCR, the computational complexity of MCVMSVM is much less than that of KSVCR.

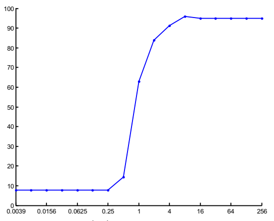
4.2 Parameter Analysis

This section will analyze the influence of model parameters on classification accuracy through numerical experiment. MCVMSVM has three parameters: D is penalty parameter, γ is kernel parameter, λ is a parameter which balance the divergence and classifier complexity in the class. Considering the influence of one parameter on accuracy, the other two parameters remain unchanged. The influence of parameters D, λ, γ on the classification accuracy are given in Figs. 6, 7 and 8.

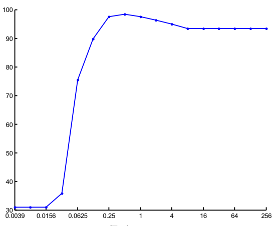
As shown in Fig. 6, when the parameter D is small, the precision of the data sets is very low, and then the classification accuracy increases significantly with the increase of the classification accuracy, and the final classification precision tends to be slow. From the point of view of the change of precision, the accuracy increases significantly at the beginning of the value 2^{-5} and the value becomes stable after 2^5. Therefore, the selection range of parameter D can be reduced between 2^{-3} and 2^5.

As shown in Fig. 7, for different data sets, the variation of γ vary greatly when MCVMSVM achieves the best accuracy. From the point of view of the change of precision, the accuracy increases significantly at the beginning of the value 0.3 and the value becomes stable after 1. Therefore, the selection range of parameter γ can be reduced between 0.3 and 1.

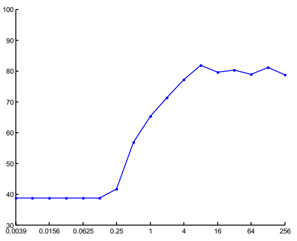
In Fig. 8, we found that except for the "iris" and "tea" datasets with the best parameters of $\lambda = 1$ and $\lambda = 4$, the other seven sets of data sets had the best accuracy at $\lambda = 0.05$. Because λ is a parameter that balances the degree of intra class divergence and the complexity of classifier, $\lambda = 0.05$ indicates that the intra class structure of the remaining class points has an effect on the classification model, so it is proved that the method we proposed is meaningful.



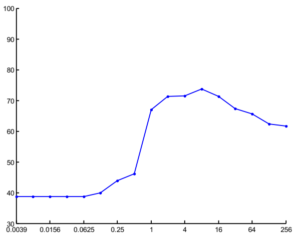
(a) Balance



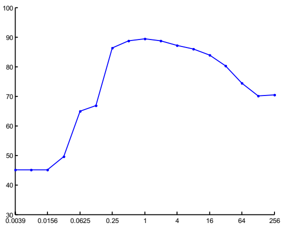
(b) Der



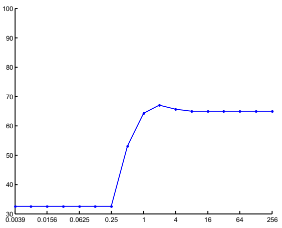
(c) Hayes



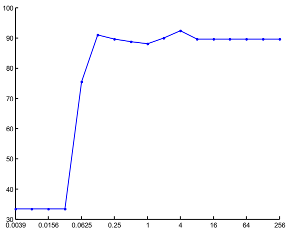
(d) Glass



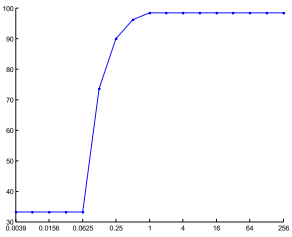
(e) Ecoli



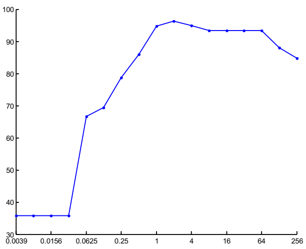
(f) Tea



(g) Seed



(h) Wine



(i) iris

Fig. 6 The influence of parameter *D* on classification accurac

4.3 NDC datasets

In order to verify the classification of MCVMSVM in large-scale data sets, NDC dataset is selected as the training samples and the data amount increases from 500 to 50,000. The parameters are fixed as $D = 2^8$, $P = 1$, $\lambda = 0.0039$. We compare the five-fold cross validation accuracies. For the five-fold cross validation, the data set is randomly split into five parts, four of them is the training set that is used to learn the hyperplane, one of them is the validation set to test the hyperplane learned by the training set. Table 3 shows the five-fold validation results. With the increase of the size of the dataset,

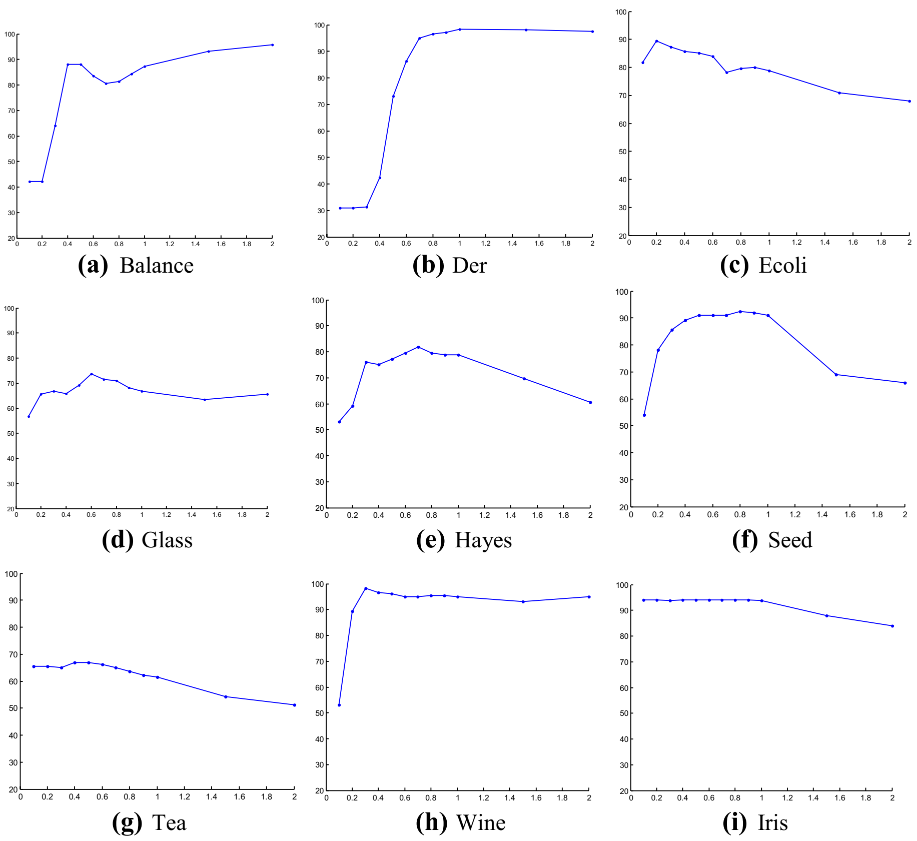


Fig. 7 The influence of parameter γ on classification accuracy

MCVMSVM has obvious advantages in classification accuracy. The average accuracy of MCVMSVM is 5% higher than other methods. Therefore, the proposed algorithm is applicable to the classification of large-scale data sets.

5 Conclusion

In this paper, the multi-class support vector machine based on the minimization of class variance (MCVMSVM) is proposed by introducing the class structure information into "1-versus-1" SVM. Comparing with "1-versus-1" SVM, we make full use of the information of rest class; Comparing with KSVCR, the computational complexity of MCVMSVM is much less than that of KSVCR. Numerical experiments demonstrate the effectiveness of our method.

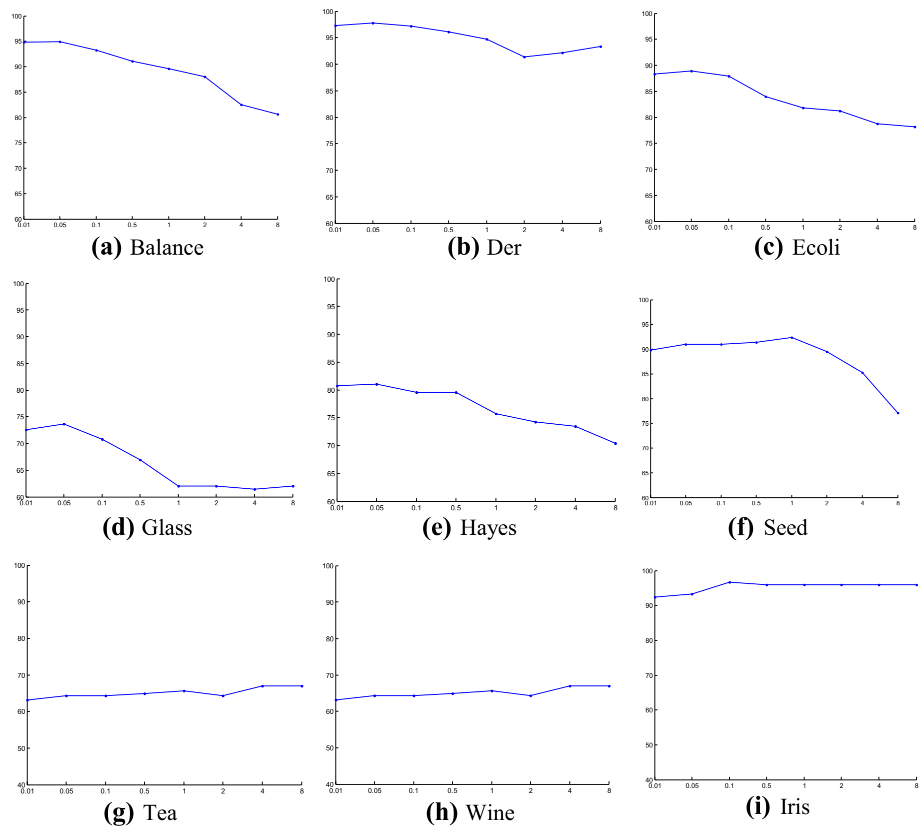


Fig. 8 The influence of parameter λ on classification accuracy

Table 3 Five fold cross validation comparison on NDC dataset

Dataset	1-1SVM	KSVCR	MCVMSVM
NDC-500	89.2 ± 3.8	66.4 ± 8.67	89.2 ± 3.8
NDC-1 k	91.59 ± 1.15	81.39 ± 8.27	91.3 ± 1.48
NDC-5 k	95.7 ± 1.26	a	96.8 ± 0.68
NDC-10 k	92.12 ± 6.99	a	96.8 ± 3.10
NDC-50 k	97.62 ± 3.23	a	96.60 ± 2.69
Average Acc	93.24	–	94.22
Average std	3.28	–	2.45

‘a’ means the experiment is terminated because of long time consuming

References

1. Cortes C, Vapnik V (1995) Support-vector network. Mach Learn 20:273–297
2. Vapnik V (1998) The nature of statistical learning, 2nd edn. Springer, New York

3. Bennett KP (1999) Combining support vector and mathematical programming methods for classification. In: Schälkopf B, Burges CJC, Smola AJ (eds) Advances in Kernel methods: support vector learning. MIT Press, Cambridge, MA, pp 307–326
4. Lee Y, Lin Y, Wahba G (2001) Multicategory support vector machines. Comput Sci Stat 33:498–512
5. Weston J, Watkins C (1998) Multi-class support vector machines, CSD-TR-98-04 Royal Holloway. University of London, Egham, UK
6. Graepel T, et al (1999) Classification on proximity data with LP-machines, In: Ninth International Conference on Artifical Neural Networks IEEE, London: Conference Publications, pp. 304–309.
7. Jayadeva R, Khemchandani R, Chandra S (2007) Twin support vector machine for pattern classification. IEEE Trans Pattern Anal Mach Intell 29(5):905–910
8. Crammer K, Singer Y (2002) On the algorithmic implementation of multiclass kernel-based vector machines. J Mach Learn Res 2:265–292
9. Bottou L, Cortes C, Denker JS, Drucher H, Guyon I, Jackel LD, LeCun Y, Muller UA, Sackinger E, Simard P, Vapnik V (1994) Comparison of classifier methods: a case study in handwriting digit recognition. In: International Conference on Pattern Recognition, Vol 2-conference B: Computer Vision & Image Processing
10. Kreb U (1999) Pairwise classification and support vector machines. In: Scholkopf B, Burges CJC, Smola AJ (eds) Advances in Kernel methods: support vector learning. MIT Press, Cambridge, pp 255–268
11. Angulo C, Parra X, Catala A (2003) K-SVCR. A support vector machine for multi-class classification. Nurocomputing 55(1):57–77
12. Yang X, Shao YH (2013) Multiple birth support vector machine for multi-class classification. Neural Comput Appl 22(Suppl 1):S153–S216
13. Deng NY, Tian YJ (2012) Support vector machines: optimization based theory, algorithms, and extensions. CRC Press, Boca Raton
14. Kotsia I, Zafeiriou S, Pitas I (2009) Novel multiclass classifiers based on the minimization of the within class variance. IEEE Trans Neural Netw 20(1):14–34
15. Blake C, Merz C (1998) UCI repository for machine learning databases. http://www.ics.uci.edu/mlearn/MLRepository.html
16. Lu J, Xuan JY, Zhang GP (2018) Structural property-aware multilayer network embedding for latent factor analysis. Pattern Recognit 76:228–241
17. Sun B, Wen SP, Wang SB (2019) Quantized synchronization of memristor-based neural networks via super twisting algorithm. Neurocomputing 380:133–140
18. Sun B, Cao YT, Guo ZY (2020) Synchronization of discrete-time recurrent neural networks with time-varying delays via quantized sliding mode control. Appl Math Comput 375:125093