International Journal of Signal Processing, Image Processing and Pattern Recognition
Vol.9, No.3 (2016), pp.219-230
http://dx.doi.org/10.14257/ijsip.2016.9.3.19

A Fast Bounded Parametric Margin Model for Support Vector Machine

Jingjing Zhang [1], Yuaihai Shao [2], Zhen Wang [3] and Wei Chen [1*]

[1] *College of Science, China Agricultural University, 100083 Beijing, China*
[2] *Zhijiang College, Zhejiang University of Technology, Hangzhou, 310024, China*
[3] *School of Mathematical Sciences, Inner Mongolia University, Hohehot, 010021,China*
zhjingjing09@163.com, shaoyuanhai21@163.com, wangzhen1882@126.com
[*] *Corresponding author: chenwei@cau.edu.cn*

Abstract

In this paper, a fast bounded parametric margin ν-support vector machine (BP-ν-SVM) for classification is proposed. Different from the parametric margin ν-support vector machine (par-ν-SVM), the BP-ν-SVM maximizes a bounded parametric margin, and consequently the successive overrelaxation (SOR) technique could be used to solve our dual problem as opposed solving the standard quadratic programming problem (QPP) in par-ν-SVM. Numerical experiments on several benchmark data sets and NDC data sets demonstrate the feasibility and effectiveness of the proposed algorithm.

***Keywords**: Support vector machine, Non-parallel hyperplanes, Parametric margin, Successive overrelaxation*

1. Introduction

Support vector machine (SVM) introduced by Vapnik and co-worker [1, 2] is an excellent kernel-based tool for classification and regression. Within a few years after its introduction, the SVM has been used in a wide variety of applications [3- 6]. The central idea of the basic SVM is to construct two parallel support hyperplanes, such that two parallel hyperplanes separate the two classes well and the margin between the two hyperplanes is maximized. Since the excellent performance of the SVM, many researches have been made to reduce the time complexity or to raise the classification accuracy, such as Chunking [1], SMO [20], LSSVM [21], fuzzy SVM [22].

Traditionally, the above algorithms and models rely on the assumption that the noise level is uniform throughout the domain, or at least, its functional dependency is known beforehand. However, it is not always satisfied, while the noise depends on location. In order to address this problem, Hao [9] proposed a parametric margin ν-support vector machine (par-ν-SVM). It maximizes a parametric margin by two non-parallel hyperplanes. Experimental results have shown that par-ν-SVM has some excellent results [9, 10] and some achievements have made based on par-ν-SVM [13, 14, 15]. However, par-ν-SVM and its improvement methods are solved by a time-consuming quadratic programming problem (QPP) in the training stage.

In this paper, we propose a fast bounded parametric margin ν-support vector machine (BP-ν-SVM). Similar with par-ν-SVM, the proposed algorithm searches for two non-parallel hyperplanes to adapt the heteroscedastic structure noise. Different from par-ν-SVM, the proposed BP-ν-SVR achieves upper bound of the maximum some margin. In addition, a versatile iterative technique, successive overrelaxation (SOR) [16, 17] is

[*] Corresponding Author

applied to our BP- ν -SVM, which improves the speed in the training process. Computational comparisons of the two algorithms on several benchmark data sets for linear kernel case and nonlinear kernel case indicate that our BP-ν -SVM is not only fast, but also obtains comparable accuracy.

This paper is organized as follows. Section 2 briefly dwells on par-ν -SVM. Our BP-ν -SVM for linear kernel and nonlinear kernel cases are formulated in Section 3. An SOR technology for BP-ν -SVM is given in Section 4. Experiments are presented in Section 5. Section 6 concludes this paper.

2. Parametric Margin ν -Support Vector Machine

Consider of binary classification problem with the training set $T=\{(x_1,y_1),...,(x_l,y_l)\}$, where $x_i \in R^n$ is the input sample labelled by $y_i \in \{+1,-1\}$ for $i=1,2,...,l.$ Par- ν -SVM [10] considers to find the parametric margin model $g(x)=c^T x+d$ and $f(x)=w^T x+b,$ where $w,c \in R^n$ are the weighted vectors, while $b,d \in R$ are the bias terms. It separates samples by two non-parallel hyperplanes $w^T x+b=\pm(c^T x+d),$ while the separated hyperplane is half of them. The par-ν -SVM considers the following constrained QPP

$$\begin{aligned}&\min_{w,c,b,d,\xi_i} \frac{1}{2}\|w\|^2+C\left(-\nu\left(\frac{1}{2}\|c\|^2+d\right)+\frac{1}{l}\sum_{i=1}^{l}\xi_i\right)\\ &s.t.\ \ y_i(w^T x_i+b)\ge\left(c^T x_i+d\right)-\xi_i \\ &\qquad \xi_i\ge 0,\ d\ge 0,\ i=1,...,l\end{aligned} \tag{1}$$

where C and ν are positive regularization factors, $\frac{1}{2}\|w\|^2$ reflects the model complexity and ξ_i is the slack vector.

By introducing the Lagrangian function for (1), its dual QPP is expressed as follows.

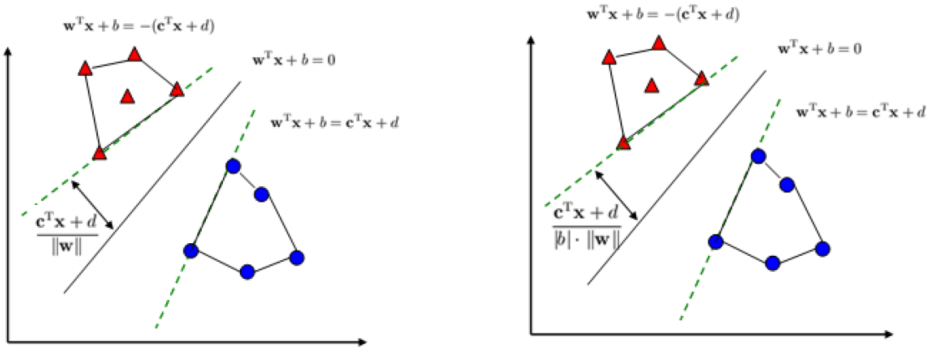
$$\begin{aligned}&\max_{\alpha_i} -\frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}y_i y_j\alpha_i\alpha_j x_i^T x_j+\frac{1}{2C\nu}\sum_{i=1}^{l}\sum_{j=1}^{l}\alpha_i\alpha_j x_i^T x_j\\ &s.t.\ \ \sum_{i=1}^{l}y_i\alpha_i=0,\ 0\le\alpha_i\le C\nu \\ &\qquad \alpha_i\in\left[0,\frac{C}{l}\right]\end{aligned} \tag{2}$$

Once the solution $\{\alpha_1,...,\alpha_l\}$ is obtained, the decision function and the parametric margin function can be shown

$$f(x)=\text{sgn}(\sum_{i=1}^{l}y_i\alpha_i x^T x_i+b), \text{and } g(x)=\frac{1}{C\nu}(\sum_{i=1}^{l}\alpha_i x^T x_i+d) \tag{3}$$

where $w=\sum_{i=1}^{l}y_i\alpha_i x_i$, $c=\frac{1}{C\nu}\sum_{i=1}^{l}\alpha_i x_i$. For some $\alpha_i,\alpha_j\in\left(0,\frac{C}{l}\right),$ according to the KKT conditions, b and d are determined by exploiting the KKT conditions

$$b = -\frac{1}{2}\left(w^T x_i + w^T x_j - c^T x_i + c^T x_j\right)$$
$$d = \frac{1}{2}\left(w^T x_i - w^T x_j - c^T x_i - c^T x_j\right)$$



(a) (b)

Figure 1. Geometric Interpretations of par-ν -SVM (a) and our BP-ν -SVM (b)

3. Bounded Parametric Margin ν -Support Vector Machine

3.1. Linear BP-ν -SVM

In this section, based on the par-ν -SVM, we propose the BP-ν -SVM. The proposed method searches for the parametric margin function $g(x) = c^T x + d$ and the function $f(x) = w^T x + b$. Its optimization problem can be described as

$$\min_{w,c,b,d,\xi_i} \frac{1}{2}(\|w\|^2 + b) + C\left(-\frac{\nu}{2}\left(\|c\|^2 + d\right) + \frac{1}{l}\sum_{i=1}^{l}\xi_i\right)$$
$$s.t.\ \ y_i(w^T x_i + b) \geq \left(c^T x_i + d\right) - \xi_i \tag{4}$$
$$\xi_i \geq 0,\ \ i = 1,\ldots,l$$

where the parameter $C > 0$ is the regularization parameter which determines the penalty weight, the parameter $\nu > 0$ controls the width of the parametric margin, ξ_i is the slack variable.

Now, we discuss the differences of the primal problems between par-ν -SVM and the proposed BP-ν -SVM by comparing (1) and (4). On the one hand, similar with [11, 12], there is an extra term b^2 in (4), which makes the QPP stable and the bias term b has a global optimal solution. On the other hand, to approximate the maximizing $\left|c^T x_i + d\right|$, $\frac{1}{2}\left(\|c\|^2 + d^2\right)$ represents the size of the parametric margin in (4). Correspondingly, the parametric margin in (1) is approximated by $\left|c^T x_i + d\right|$, $\frac{1}{2}\left(\|c\|^2 + d\right)$, which lead to a small value of d . Therefore, BP-ν -SVM has bigger margin to the parametric margin than that of par-ν -SVM. Specifically, in BP-ν -SVM, the margin between two classes can be measured by maximizing the margin between the hyperplane $f(x)$ and the bounded hyperplane $g(x)$, here, i.e., $\sum_{i=1}^{l}\frac{|c^T x_i + d|}{|b|\cdot\|w\|}$, where x_i is the support vector. And in par-ν-SVM, the margin by maximizing $f(x)$ and $g(x)$ is $\sum_{i=1}^{l}\frac{|c^T x_i + d|}{\|w\|}$ (see Figure 1).

To solve problem (4), we construct its Lagrangian function

$$
\begin{aligned}
L &= \frac{1}{2}(\|w\|^2 + b) + C\left(-\frac{\nu}{2}\left(\|c\|^2 + d\right) + \frac{1}{l}\sum_{i=1}^{l}\xi_i\right) \\
&-\sum_{i=1}^{l}\alpha_i(y_i(w^T x_i + b) - c^T x_i - d + \xi_i) - \sum_{i=1}^{l}\beta_i\xi_i
\end{aligned} \tag{5}
$$

where α_i and β_i are the nonnegative Lagrangian multipliers. The KKT conditions are given by

$$\partial L / \partial w = w - \sum_{i=1}^{l}\alpha_i y_i x_i = 0 \Rightarrow w = \sum_{i=1}^{l}\alpha_i y_i x_i \tag{6}$$

$$\partial L / \partial b = b - \sum_{i=1}^{l}\alpha_i y_i = 0 \Rightarrow b = \sum_{i=1}^{l}\alpha_i y_i \tag{7}$$

$$\partial L / \partial c = -C\nu c + \sum_{i=1}^{l}\alpha_i x_i = 0 \Rightarrow c = \frac{\sum_{i=1}^{l}\alpha_i x_i}{C\nu} \tag{8}$$

$$\partial L / \partial d = -C\nu d + \sum_{i=1}^{l}\alpha_i = 0 \Rightarrow d = \frac{\sum_{i=1}^{l}\alpha_i}{C\nu} \tag{9}$$

$$\partial L / \partial \xi_i = \frac{C}{l} - \alpha_i - \beta_i = 0 \Rightarrow \alpha_i + \beta_i = \frac{C}{l} \tag{10}$$

$$\alpha_i \geq 0, \beta_i \geq 0 \tag{11}$$

Utilizing the above equations into the Lagrangian function, we get its dual problem

$$
\begin{aligned}
&\max_{\alpha_i} -\frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}\alpha_i\alpha_j(y_i y_j + \frac{1}{C\nu})(x_i^T x_j + 1) \\
&s.t. \quad 0 \leq \alpha_i \leq \frac{C}{l}, \quad i = 1,...,l
\end{aligned} \tag{12}
$$

Once the solution of $(\alpha_1,...,\alpha_l)$ is obtained, the decision function $\text{sgn}(f(x))$ and the parametric margin function $g(x)$ can be shown as follows.

$$f(x) = \text{sgn}(w^T x + b) = \text{sgn}(\sum_{i=1}^{l}\alpha_i y_i x^T x_i + \sum_{i=1}^{l}\alpha_i y_i) \tag{13}$$

$$g(x) = c^T x + d = \frac{\sum_{i=1}^{l} \alpha_i x^T x_i}{C\nu} + \frac{\sum_{i=1}^{l} \alpha_i}{C\nu} \tag{14}$$

3.2. Nonlinear BP-ν-SVM

In order to extend our results to the nonlinear case, similar to [13], we consider the kernel-generated surface and the parametric margin hyperplane as $f(x) = w^T \varphi(x) + b$ and $g(x) = c^T \varphi(x) + d$, where $\varphi(\cdot)$ is to map a data set χ to high dimensional feature space H, and $k(\cdot,\cdot)$ is the kernel function. We need to solve the following QPP

$$\begin{aligned} &\min_{w,c,b,d,\xi_i} \frac{1}{2}(\|w\|^2 + b) + C\left(-\frac{\nu}{2}\left(\|c\|^2 + d\right) + \frac{1}{l}\sum_{i=1}^{l} \xi_i \right) \\ &s.t. \;\; y_i(w^T \varphi(x_i) + b) \ge \left(c^T \varphi(x_i) + d\right) - \xi_i \\ &\quad\; \xi_i \ge 0, \;\; i = 1, \ldots, l \end{aligned} \tag{15}$$

By introducing the Lagrangian function and utilizing the KKT conditions for problem (15), we obtain its dual QPP

$$\begin{aligned} &\max_{\alpha_i} -\frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l} \alpha_i \alpha_j (y_i y_j + \frac{1}{C\nu})(k(x_i, x_j) + 1) \\ &s.t. \;\; 0 \le \alpha_i \le \frac{C}{l}, \;\; i = 1, \ldots, l \end{aligned} \tag{16}$$

weighted vectors w and c can be obtained according to (16), $w = \sum_{i=1}^{l} \alpha_i y_i \varphi(x_i)$, $c = \frac{\sum_{i=1}^{l} \alpha_i \varphi(x_i)}{C\nu}$. As for the bias terms b and d, we have $b = \sum_{i=1}^{l} \alpha_i y_i$, $d = \frac{\sum_{i=1}^{l} \alpha_i}{C\nu}$.

Similar with the linear case, the decision surface $f(x)$ and the parametric margin surface $g(x)$ can be shown

$$f(x) = \text{sgn}(\sum_{i=1}^{l} \alpha_i y_i k(x, x_i) + \sum_{i=1}^{l} \alpha_i y_i) \tag{17}$$

$$g(x) = \frac{\sum_{i=1}^{l} \alpha_i k(x, x_i)}{C\nu} + \frac{\sum_{i=1}^{l} \alpha_i}{C\nu} \tag{18}$$

4. A SOR Algorithm for BP-ν-SVM

In our BP-ν-SVM, optimization problems (12) and (16) need to be solved. We rewrite the optimization problem as

$$\min_{\alpha} \frac{1}{2}\alpha^T A\alpha - d^T\alpha$$
$$s.t.\quad \alpha \in S = \{0 \le \alpha \le ce\} \tag{19}$$

where $A = (A_{i,j})_{l\times l}$ is a positive definite matrix with

$$A_{ij} = (y_i y_j + \frac{1}{C\nu})(k(x_i, x_j) + 1) \tag{20}$$

where $i = 1,...,l,\ j = 1,...,l.\ c = \frac{C}{l},\ d = (0,...,0)^T_{l\times l},\quad \alpha = \left(\alpha_1,...,\alpha_l\right)^T,\ e = (1,...,1)^T_{1\times l}$.
The above problem (19) can be solved efficiently by the following successive overrelaxation (SOR) technique [17, 12]. The SOR technique, which is a matrix splitting method that converges linearly to a sample a satisfying (12) or (16), leads to the following algorithm.

Algorithm 1. Choose $\omega \in \left(0,2\right)$. Start with any $\alpha^0 \in R^n$. Having α^i compute α^{i+1} as follows

$$\alpha^{i+1} = \left(\alpha^i - \omega D^{-1}\left(A\alpha^i - d + L\left(\alpha^{i+1} - \alpha^i\right)\right)\right) \tag{21}$$

until $\left\|\alpha^{i+1} - \alpha^i\right\|$ is less than some prescribed tolerance, where the nonzero elements of $L \in R^{n\times n}$ constitute the strictly lower triangular part of the symmetric matrix A, and the nonzero elements of $D \in R^{n\times n}$ constitute the diagonal of A.

SOR is an efficient solver for our BP-ν-SVM, since it do not need reside in memory. Moreover, it has been proved that this algorithm converges linearly to a solution [16]. The experimental results in the next section shows the acceleration effect on our BP-ν-SVM.

5. Experiments

To test the performance of the proposed BP-ν-SVM, we compare our implementation with par-ν-SVM and ν-SVM on several UCI [18] benchmark data sets and David Musicant's NDC Data Generator [19]. All methods are implemented by using Matlab 7.0 on a PC. We utilize "qp.m" for par-ν-SVM, libsvm [8] for ν-SVM, respectively, and the SOR technique is used in our BP-ν-SVM. For the nonlinear model, the Gaussian kernel $K(x, y) = \exp(-q\|x - y\|^2)$ is considered in our experiments. For the values of parameter C and kernel parameter q in these algorithms, we select them from the sets of values $\{10^i \mid i = -5, -4, ..., 6\}$ and $\{2^i \mid i = -9, -8, ..., 5\}$. If no particular claim, we apply the ten-fold cross-validation method on the whole training data set to estimate the generalized accuracy. In the comparisons, we consider the test accuracies, the numbers of support vector and the training time. The "Accuracy" used to evaluate methods is defined as: Accuracy = TP /(TP + FP), where TP and FP are the predicted correctly number of positive and negative samples, respectively.

Table 1 and Table 2 report the comparison results of these algorithms with linear and nonlinear kernels on UCI data sets, respectively. They contains the accuracy, the training time and the number of support vectors (SVs) as performance on six data sets. The best results of performance metrics are shown by bold figures. In Table 1, for the accuracy index, the proposed method has better or comparable values on all data sets except House-vote data set. As for the number of the support vectors, it also shows that the number of support vectors obtained by BP-ν-SVM is smallest than those of par-ν-SVM, and ν-SVM except for WPBC data set. It implies that our BP-ν-SVM has a better sparsity than par-ν-SVM and ν-SVM. Concerned with the nonlinear kernel BP-ν-SVM, par-ν-SVM and ν-SVM, we observed that for the accuracy our method has comparable values on all data sets except House-vote data set. As for the numbers of support vectors, our method obtains the smallest values on four of six data sets. The training time in Table 1 and 2 show that our BP-ν-SVM is the fastest algorithm on almost data sets. Especially, it is almost hundreds of times faster than par-ν-SVM.

We also conduct experiments on large data sets. For experiments with NDC data sets, we selected the samples from 10^2 to 10^4 as the training set and fixed penalty parameters of all methods to be one (*i.e.*, $C=1$, $\nu=0.1$ and $q=0.0001$). Table 3 (a) and (b) show the comparison of training time for linear and nonlinear kernel par-ν-SVC and our BP-ν-SVC on NDC data sets, respectively. It indicated that our method has a much faster learning speed compared with par-ν-SVM. Par-ν-SVM spends too much time to get a classifier and is terminated when learning more than 4,000 samples because of out of the memory.

Table 1. Results of linear ν-SVM, par-ν-SVM and BP-ν-SVM, Respectively

Data sets	BP-ν-SVC Accuracy (C,ν,q) Time (s) SVs	par-ν-SVC Accuracy (C,ν,q) Time (s) SVs	ν-SVC Accuracy (ν) Time (s) SVs
WPBC (198×34)	**0.7626** $(10,0.3)$ **0.0095** **114.6**	**0.7576** $(0.01,0.9)$ **0.4486** **178.2**	**0.7727** (0.3) **0.2030** **91.0**
Heart-statlog (270×13)	**0.8333** $(1,0.6)$ **0.0031** **90.0**	**0.7556** $(100000,0.8)$ **82.75** **213.9**	**0.8370** (0.5) **0.0410** **111.0**
Haberman (306×3)	**0.7255** $(0.01,0.1)$ **0.0094** **145.3**	**0.7255** $(10,0.9)$ **1.514** **275.4**	**0.7059** (0.5) **0.1710** **159.0**
Spect (267×44)	**0.7865** $(0.01,0.1)$ **0.0062** **83.6**	**0.7303** $(10^5,0.1)$ **0.8657** **240.0**	**0.7862** (0.3) **0.1200** **114.0**

CMC (1473×9)	**0.7739** $(100,0.1)$ **1.4370** **83.4**	**0.7632** $(0.1,0.1)$ **917.6** **1250.1**	**0.5696** (0.1) **0.5320** **159**
House-vote (435×16)	**0.8782** $(10,0.9)$ **0.0084** **127.1**	**0.8115** $(100000,0.8)$ **1.0174** **212.6**	**0.9586** (0.1) **0.2697** **136.4**

Table 2. Results of Nonlinear ν -SVM, par- ν -SVM and BP- ν -SVM, Respectively

Data sets	**BP- ν -SVC** **Accuracy** (C,ν) **Time (s)** **SVs**	**par- ν -SVC** **Accuracy** (C,ν) **Time (s)** **SVs**	**ν -SVC** **Accuracy** (C,ν) **Time (s)** **SVs**
WPBC (198×34)	**0.7626** $(10,0.6,2)$ **0.0140** **197.1**	**0.7980** $(10^6,0.3,2^{-7})$ **3.3590** **105.2**	**0.8030** $(0.3,2^{-9})$ **0.0860** **100**
Heart-statlog (270×13)	**0.8296** $(1,0.6,0.25)$ **0.0031** **103.8**	**0.8407** $(1,1,2^{-6})$ **1.5733** **185.6**	**0.8296** $(0.8,2^{-6})$ **0.0780** **169.0**
Haberman (306×3)	**0.7386** $(0.1,0.4,1)$ **0.0122** **133.7**	**0.7386** $(10^5,0.1,2^{-9})$ **8.8720** **185.6**	**0.7255** $(0.5,2^{-9})$ **0.1126** **169.0**
Spect (267×44)	**0.7940** $(0.1,0.1,0.25)$ **0.0110** **113.1**	**0.8052** $(10^2,0.7,2^{-7})$ **0.8480** **240.3**	**0.8030** $(0.4,2^{-7})$ **0.1055** **110.0**

CMC (1473×9)	**0.7766** $(0.01,1,1)$ **0.1236** **632.8**	**0.7760** $(10^6,0.9,2^4)$ **55.0** **1325.7**	**0.7591** $(0.4,2^{-2})$ **4.0600** **792**
House-vote (435×16)	**0.8677** $(0.1,0.8,0.125)$ **0.0303** **175.5**	**0.9011** $(10^3,0.2,2^{-6})$ **65.62** **213.3**	**0.9609** $(0.2,2^{-4})$ **0.1610** **192.1**

Table 3. CPU Time Cost for Par-ν -SVM, BP-ν-SVM, Respectively.
[a] Terminate because of Out of Memory

(a) Linear Case

NDC data $m\times n$	**BP-ν-SVM Time (s)**	**par-ν-SVM Time (s)**
100×32	**0.0016**	**1.613**
1000×32	**0.0334**	**140.4**
3000×32	**0.0312**	**4657**
4000×32	**0.6243**	[a]
$10k\times 32$	**15.72**	[a]

(b) Nonlinear Case

NDC data $m\times n$	**BP-ν-SVM Time (s)**	**par-ν-SVM Time (s)**
100×32	**0.0007**	**0.8010**
1000×32	**0.0296**	**170.9**
3000×32	**0.03581**	**5143**
4000×32	**0.6229**	[a]
$10k\times 32$	**19.16**	[a]

5. Conclusion

In this paper, we have proposed an improved version model of par-ν-SVM, called BP-ν-SVM. Similar with par-ν-SVM, BP-ν-SVM adjusts a flexible margin for classification, which is suitable for the heteroscedastic noise structure. The main contribution is that, our proposed algorithm maximized a bounded parametric margin and the SOR technique is used in our BP-ν-SVM. Computational comparisons among par-ν-SVM, ν-SVM and the proposed BP-ν-SVM have been made on several data sets, indicating that our BP-ν-SVM is not only fast, but also show comparable generalization. However, the parameter ν losses the some geometric meaning. It will be our future work.

Acknowledgment

This work is partially supported by National Natural Science Foundation of China (Nos. 11171346 and 11201426).

References

[1] C. Cortes and V. N. Vapnik, "Support vector networks", Machine Learning vol. 20, **(1995)**, pp. 273-297.
[2] V.N. Vapnik, "The nature of statistical learning theory", Springer Verlag, **(1995)**, New York.
[3] E. Osuna, R. Freund and F. Girosi, "Training support vector machines: An application to face detection", proceedings of IEEE computer vision and pattern recognition, San Juan, Puerto Rico, **(1997)**, pp.130-136.
[4] T. Joachims, C. Ndellec and C. Rouveriol, "Text categorization with support vector machines: Learning with many relevant features", European conference on machine learning, **(1998)**, pp.137-142.
[5] I. El-Naqa, Y. Yang, M. Wernik, N.P. Galatsanos and R. M. Nishikawa, "A support vector machine approach for detection of microclassification", IEEE transactions on medical imaging, vol. 21, no. 12, **(2002)**, pp. 1552-1563.
[6] T. Ebrahimi, G.N. Garcia and J.M. Vesin, "Joint time-frequency-space classification of EEG in a brain-computer interface application", Journal of applied signal processing, vol. 1, no. 7, **(2003)**, pp. 713-729.
[7] B. Schökopf, A.J. Smola, R.C. Williamson and P.L. Bartlett, "New support vector algorithms", neural computation, vol. 12, **(2000)**, pp. 1207-1245.
[8] C. Chang and C. Lin, "LIBSVM: a library for support vector machines", Technical report,Department of computer science and information engineering, **(2001)**.
[9] P.Y Hao, L.B. Tsai and M.S. Lin, "A new support vector classification algorithm with parametric-margin model", IEEE Int Jt Conf Neural Net, **(2008)**, pp. 420-425.
[10] P.Y. Hao, "New support vector algorithms with parametric insensitive/margin model", EEE transactions on neural networks, vol. 23, **(2010)**, pp. 60-73.
[11] G. Fung and O. L. Mangasarian, "Proximal support vector machine classifiers", In: Proceedings of seventh international conference on knowledge and data discovery, San Francisco, **(2001)**, pp. 77-86.
[12] Y. H. Shao, C. H. Zhang, X. B. Wang and N. Y. Deng, "Improvements on twin support vector machines", IEEE transactions on neural networks, vol. 22, no. 6, **(2011)**, pp. 962-968.
[13] X. J. Peng, "TPMSVM: A novel twin parametric-margin support vector machine for pattern recognition", Pattern Recognition, vol. 44, **(2011)**, pp. 2678-2692.
[14] Z. Wang, Y. H. Shao, T. R. Wu, "Proximal parametric-margin support vector classifier and its applications", Neural Computing and Applications, vol. 24, **(2014)**, pp.755-764.
[15] W. Chen, Y. Shao and H. Wang, "Successive overrelaxation for twin parametric-margin support vector machines", Journal of information computational science, vol. 10, **(2013)**, pp. 791-799.
[16] Z. Q. Luo and P. Tseng, "Error bounds and convergence analysis of feasible descent methods: A general approach", Annals of operations research, vol. 46, **(1993)**, pp. 157-178.
[17] O. L. Mangasarian and D.R. Musicant, "Successive overrelaxation for support vector machines", IEEE transactions on neural networks, vol. 10, no. 5, **(1999)**, pp. 1032-1037.
[18] C. L. Blake and C. J. Merz, "UCI repository for machine learning databases", Dept. of information and computer sciences, Univ. of California, Irvine, http://www.ics.uci.edu/ ~ mlearn/MLRepository.html, **(1998)**.
[19] D. R. Musicant, "NDC: normally distributed clustered data sets", www. cs. wisc. edu /dmi /svm/ndc/, **(1998)**.
[20] J. Platt, "Fast training of support vector machines using sequential minimal optimization", Advances in Kernel Methods-support vector learning, MIT Press, Cambridge, MA, **(1999)**, pp.185–208.
[21] A. K. Suykens, J. Vandewalle, "Least squares support vector machine classifiers", Neural Process Lett, vol. 9, no. 3, **(1999)**, pp. 293–300.
[22] C. F. Lin and S. D. Wang, "Fuzzy support vector machines", IEEE Trans. Neural Networks, vol. 13, no. 2, **(2002)**, pp. 464-471.

Authors

Jingjing Zhang, She has received her B.S. degree in College of Mathematics and Information Science from Shaanxi Normal University, 2010, and received. Master's degree in College of Science from China Agricultural University, China, in 2012. Currently, she currently pursuing the Ph.D. degree at theCollege of Science of China Agricultural University. Her research interests include optimization methods ,machine learning and data mining.

Yuanhai Shao, He received his B.S. degree in information and computing science in College of Mathematics from Jilin University, the master's degree in applied mathematics, and Ph.D. degree in operations research and management in College of Science from China Agricultural University, China, in 2006, 2008, and 2011, respectively. Currently, he is an Associate Professor at the Zhijiang College, Zhejiang University of Technology. His research interests include optimization methods, machine learning, and data mining. He has published over 50 refereed papers on these areas.

Zhen Wang, He received his Doctor's degree in College of Mathematics from Jilin University, China, in 2014. Currently, he is lecturer in School of Mathematical Sciences from Inner Monggolia University. His research interests include pattern recognition, text categorization, and data mining.

Wei Chen, He is a professor of China Agricultural University. She is also a part-time researcher at Institute of extension engineering, Guangdong University of Technology. Her research interest includes Statistics, machine learning and support vector machines.

