Research Article

Selecting the Optimal Combination Model of FSSVM for the Imbalance Datasets

Chuangdong Qin¹ and Huixia Zhao²

¹ School of Mathematics and Information Science, North National University, Yinchuan 750021, China

² Business School, North National University, Yinchuan 750021, China

Correspondence should be addressed to Chuangdong Qin; qcd369@163.com

Received 20 October 2013; Revised 27 January 2014; Accepted 9 February 2014; Published 16 March 2014

Academic Editor: Cheng Shao

Copyright © 2014 C. Qin and H. Zhao. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Imbalanced data learning is one of the most active and important fields in machine learning research. The existing class imbalance learning methods can make Support Vector Machines (SVMs) less sensitive to class imbalance; they still suffer from the disturbance of outliers and noise present in the datasets. A kind of Fuzzy Smooth Support Vector Machines (FSSVMs) are proposed based on the Smooth Support Vector Machine (SSVM) of O. L. Mangasarian. SSVM can be computed by the Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm or the Newton-Armijo algorithm easily. Two kinds of fuzzy memberships and three smooth functions can be chosen in the algorithms. The fuzzy memberships consider the contribution rate of each sample to the optimal separating hyperplane. The polynomial smooth functions can make the optimization problem more accurate at the inflection point. Those changes play the active effects on trials. The results of the experiments show that the FSSVMs can gain the better accuracy and the shorter time than the SSVMs and some of the other methods.

1. Introduction

Learning from imbalanced datasets is an important and ongoing issue in machine learning research. The classification problem with imbalanced training data corresponds to domains for which one class is represented by a large number of instances while the other is represented by only a few. There are many such problems in the real world [1–3]. Conventional classifiers, which are trained with an imbalanced dataset, can produce a model that is biased toward the majority class. It has a low performance on the minority class. These methods can be broadly divided into two categories, namely, external methods and internal methods. External methods involve preprocessing of training datasets in order to make them balanced, such as randomly undersampling [4], randomly oversampling [5], while internal methods deal with modifications of the learning algorithms in order to reduce their sensitiveness to class imbalance, such as Synthetic Minority Oversampling Technique (SMOTE) [6]. In addition, a genetic algorithm based sampling has been proposed in [7], and Z-SVM has been proposed in [8].

The general SVM considers all the training examples uniformly. It is sensitive to outliers and noise in the datasets [9]. They exist in most of the real world. A fuzzy membership technique [10] is introduced to SVM and assigned a different fuzzy membership values (weights) for the different examples. It can reflect the importance of each sample in the algorithms and reduce the effect of outliers and noise. However, the Fuzzy Support Vector Machine (FSVM) can still be influenced by the imbalanced problem. Considering those factors, we define the imbalanced adjustment factor and two kinds of fuzzy membership functions in the models. On the other hand, three smooth functions are applied to the SSVM models which can change the differentiability and make the model be computed by Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm [11] or the Newton-Armijo algorithm easily [12].

The rest of this paper is organized as follows. Section 2 briefly reviews the SSVM learning theory and its smooth functions, and Section 3 defines the two fuzzy membership functions. In Section 4, we present the FSSVM algorithm, and in Section 5 we discuss the two algorithms. Section 6

is the experiment results. Finally, Section 7 concludes the paper.

2. SSVM and Its Smooth Function

Given an unknown independent and identically distributed dataset: $S = \{(x_1, y_1), \dots, (x_n, y_n), x_i \in R^n, y_i \in \{-1, 1\}\}$, SSVM is a variant of the SVM learning algorithm which was originally proposed in [13]. The reformation of SVM can be expressing as follows:

$$\begin{aligned} \min_{(w, b, \xi) \in R^{n+1+m}} \quad & \frac{1}{2} (w^T w + b^2) + \frac{C}{2} \xi^2 \\ \text{subject to} \quad & D(Aw - eb) + \xi \geq e. \end{aligned} \quad (1)$$

Here ξ is given by $\xi = (e - D(Aw - eb))_+$, and the sign function $(\cdot)_+$ replaces negative components of a vector by zeros. Thus, we can replace ξ in (1) by $(e - D(Aw - e))_+$ and convert the SVM problem (1) into an equivalent SVM, which is an unconstrained optimization problem as follows:

$$\min_{(w, b, \xi) \in R^{n+1+m}} \quad \frac{\gamma}{2} \|(e - D(Aw - eb))_+\|_2^2 + \frac{1}{2} (w^T w + b^2). \quad (2)$$

This is a strongly convex minimization problem without any constraints and it has a unique solution. However, the objective function in (2) is not twice differentiable which precludes the use of the fast Newton method. We thus apply the smoothing techniques and approximately replace x_+ by an accurate sigmoid smooth function:

$$p_s(x, k) = x + \frac{1}{k} \log(1 + e^{-kx}), \quad k > 0. \quad (3)$$

In order to gain the more accurate smooth function, some researchers [14] proposed several polynomial smooth functions in his paper as follows:

$$\begin{aligned} p_2(x) &= \begin{cases} x, & x \geq \frac{1}{k}, \\ \frac{k}{4}x^2 + \frac{1}{2}x + \frac{1}{4k}, & \frac{1}{k} < x < \frac{1}{k}, \quad k > 0 \\ 0, & x \leq -\frac{1}{k}, \end{cases} \\ p_4(x) &= \begin{cases} x, & x \geq \frac{1}{k}, \\ -\frac{1}{16k}(kx+1)^3(kx+3), & -\frac{1}{k} < x < \frac{1}{k} \\ 0, & x \leq -\frac{1}{k}. \end{cases} \end{aligned} \quad (4)$$

Obviously, function $p_2(x, k)$ is a piecewise continuous function and first-order differentiable about x^2 . Function $p_4(x, k)$ is another piecewise continuous function and twice-order differentiable about x^4 . The sigmoid function is arbitrary-order differentiable. But the smoothness of those functions is different.

FIGURE 1: The graph of the three smooth functions and the sign function.

Lemma 1. For a given x and k , the features of the smooth functions can be gained:

$$p_s(x, k) \geq p_2(x, k) \geq p_4(x, k) \geq x_+. \quad (5)$$

Lemma 2. For a given x and k , the relation of the above smooth functions and the x_+ can be gained the following inequality:

$$\begin{aligned} p_s(x, k)^2 - x_+^2 &\leq ((\log 2)^2 + 2 \log 2) \frac{1}{k^2} \approx 0.6927 \frac{1}{k^2}, \\ p_2(x, k)^2 - x_+^2 &\leq \frac{1}{11k^2} \approx 0.0909 \frac{1}{k^2}, \\ p_4(x, k)^2 - x_+^2 &\leq \frac{1}{19k^2} \approx 0.0526 \frac{1}{k^2}. \end{aligned} \quad (6)$$

The proofs of Lemmas 1 and 2 can be found in paper [14]. From these features, we can gain the advantages and disadvantages of each smooth function: the sigmoid function is arbitrary-order differentiable, but it is inaccurate at the inflection point; function $p_2(x, k)$ is first-order differentiable. It is more accurate than that of the sigmoid function, but not as good as the function $p_4(x, k)$; function $p_4(x, k)$ is twice-order differentiable and it is the most accurate at the inflection point among the functions. But for the quadratic convergence, the speed of iterations of $p_4(x, k)$ is quick, but it may lead to the expensive computation. The convergence of the three functions are relative to the parameter k . In order to gain the better effectiveness, we apply those smooth functions and the fuzzy memberships to the SSVM. The comparison of those smooth functions graph can be seen from Figure 1, where e in the sigmoid function is the base of natural logarithm and k is the smooth factor. These SSVMs algorithms with smooth functions can be showed as follows generally:

$$\min_{(w, b, \xi)} \quad \frac{1}{2} (w^T w + b^2) + \frac{C}{2} \|p(e - D(Aw - eb), k)\|_2^2. \quad (7)$$

Formula (7) is an unconstrained optimization problem, which can be computed by the gradient descent algorithms, but the disturbing of some noise and outliers in datasets was not considered. In order to improve the accuracy of the SSVMs and increase the complexity of the algorithms not too much, we introduce a kind of fuzzy membership to the SSVMs.

3. Fuzzy Membership for the Imbalanced Dataset

In order to deal with the problem of outliers and noise, we introduce a kind of fuzzy membership technique which can consider the effect of the noise and outliers in the imbalanced datasets. The r^+ and r^- are assigned to reflect the unbalancedness. Therefore, a positive-class example is given a membership value in the $[0, r^+]$ interval, while a negative-class example is given a membership value in the $[0, r^-]$ interval. We assign $r^+ = 1$ and $r^- = r$ to show the imbalanced ratio, where r is the minority-to-majority class ratio. According to this assignment of values, a positive-class example can take a membership value in the $[0, 1]$ interval, and the negative-class example can take the value in the $[0, r]$ interval, where $r < 1$.

At the same time, we define the function $f(x_i)$ based on the distance from the actual separating hyperplane to x_i , which is found by training a normal SVM model on the imbalanced dataset. The examples closer to the actual separating hyperplane are treated as more informative and assigned higher membership values. The membership function can define as follows.

- (1) Train a normal SVM model with the original imbalanced dataset.
- (2) Find the functional margin d_i^h of each example x_i . The functional margin is proportional to the geometric margin of a training example with respect to the hyperplane:

$$d_i^h = y_i (w \cdot \phi(x_i) + b). \quad (8)$$

- (3) Define the linear-decaying function and the exponential-decaying functions as follows:

$$\begin{aligned} f_{\text{lin}}^h(x_i) &= 1 - \frac{d_i^h}{\max(d_i^h + \Delta)} \\ f_{\text{exp}}^h(x_i) &= \frac{2}{1 + \exp(\beta d_i^h)}, \quad \beta \in [0, 1]. \end{aligned} \quad (9)$$

Here Δ is a small positive number. For the imbalanced dataset, we define the fuzzy membership of an example as $s_i = f(x_i) \cdot r$ and apply the two fuzzy memberships to the SSVM.

4. SSVM with the Fuzzy Membership (FSSVM)

After preprocessing dataset, we can gain dataset with the fuzzy membership as follow:

$$T = \{(x_1, y_1, s_1), (x_2, y_2, s_2) \cdots (x_m, y_m, s_m)\}. \quad (10)$$

Based on the SSVM classifiers of (7), the optimization problem of FSSVM in the higher feature space F is given by the following model:

$$\begin{aligned} \min_{(w, b, \xi) \in R^{n+1+m}} \quad & \frac{1}{2} (w^T w + b^2) + \frac{C}{2} S^2 \xi^2 \\ \text{subject to} \quad & D(w\varphi(A) - eb) + \xi \geq e, \end{aligned} \quad (11)$$

where A is the dataset, ξ is the slack variable, and S is the fuzzy membership matrix. At a solution of problem (11), ξ is given by the following plus function:

$$\xi = (e - D(w\varphi(A) - eb))_+. \quad (12)$$

According to the smoothness and the differentiability, we replace the plus function with the above smooth function, respectively. At the same time, in order to find a better separation of classes, the data are first transformed into a higher dimensional feature space by a mapping function φ . As an important property of SVMs, it is not necessary to know the mapping function $\varphi(x)$ explicitly. By defining the kernel function $K := k(A, A^T) = \varphi(A) \cdot \varphi(A^T)$ in the feature space $F : x_i \rightarrow \varphi(x_i)$, we gain the FSSVMs model as the following:

$$\min_{(u, b, \xi)} \frac{1}{2} (u^T u + b^2) \quad (13)$$

$$+ \frac{CS^2}{2} \|p_s(e - D(K(A, A^T)Du - eb), k)\|_2^2,$$

$$\min_{(u, b, \xi)} \frac{1}{2} (u^T u + b^2) \quad (14)$$

$$+ \frac{CS^2}{2} \|p_2(e - D(K(A, A^T)Du - eb), k)\|_2^2,$$

$$\min_{(u, b, \xi)} \frac{1}{2} (u^T u + b^2) \quad (15)$$

$$+ \frac{CS^2}{2} \|p_4(e - D(K(A, A^T)Du - eb), k)\|_2^2,$$

where $K(A, A^T)$ is a kernel map from $R^{m \times n} \times R^{n \times m}$ to $R^{m \times m}$. We note that this problem, which is capable of generating highly nonlinear separating surfaces, still retains the strong convexity and differentiability properties for any arbitrary kernel. Hence we can apply the BFGS algorithm to solve the problems (13)~(15) and the Newton-Armijo algorithm to solve (13) and (15). We call them FS_sSVM , FS_2SVM , and FS_4SVM in the following experiments. We turn our attention now to the algorithms.

5. Algorithms

In this section, we introduce the BFGS algorithm and Newton-Armijo algorithm for the above unconstrained optimizations (13)~(15).

5.1. BFGS Algorithm. If the objection function is the first-order differentiable, we can use BFGS algorithm to compute

the unconstrained optimization problem according to the following algorithm.

- (i) Set $H = I$, x_0 , $\varepsilon_1 > 0$, let $k \leftarrow 0$. Set constant s , ρ , β where $s > 0$, $\sigma \in (0, 0.5)$, $\beta \in (0, 1)$.
- (ii) Compute $g_{(k)}$. If $\|g_{(k)}\| \leq \varepsilon_1$, stop, take $x^* = x_k$, otherwise compute the descent direction d_k from $d_k = -H_k g_{(k)}$.
- (iii) Compute the iteration step α_k with the linear search. Let $\alpha_k = \beta^{m_k} s$, m_k is the smallest positive integral which make the inequality: $f(x_{(k)} + \beta^m s d_{(k)}) \leq f(x_{(k)}) + \sigma \beta^m g_{(k)}^T d_{(k)}$ let: $x_{(k+1)} = x_{(k)} + \alpha_k d_{(k)}$, $\delta_{(k)} = \alpha_k d_{(k)}$, let: $s_k = s_{k+1} - s_k$, $y_{(k)} = g_{(k+1)} - g_{(k)}$.
- (iv) From H_k to H_{k+1} : If $y_k^T s_k \leq 0$, let $H_{k+1} = H_k$, otherwise: $H_{k+1} = H_k - H_k y_{(k)} H_k y_{(k)}^T / y_{(k)}^T H_k y_{(k)} + s_k s_k^T / s_{(k)}^T y_{(k)}$.
- (v) Let: $k \leftarrow k + 1$, go to step (ii).

5.2. Newton-Armijo Algorithm. If the objection function is the twice-order differentiable, we can use Newdon-Armijo algorithm to compute the unconstrained optimization problem according to the following algorithm.

- (i) Set the initial point x_1 , and $\varepsilon_1 > 0$, let $k \leftarrow 0$.
- (ii) Let $k \leftarrow k + 1$ compute $g_{(k)}$, if $\|g_{(k)}\| \leq \varepsilon$, then take $x^* = x_{(k)}$, stop, otherwise go to (iii).
- (iii) Compute $G_k = G(x_{(k)})$ and the descent direction d_k from $G_k d_k = -g_{(k)}$.
- (iv) Armijo step-size, choose a step-size $\lambda_i \in R$ such that $f(x_{(i)}) - f(x_{(i)} + \lambda_i d_{(i)}) \geq -\delta \lambda_i (g_{(i)})^T d_{(i)}$ where $\lambda_i = \{1, 1/2, 1/4, \dots\}$, $\delta \in (0, 0.5)$.
- (v) Set $x_{(i+1)} = x_{(i)} + \lambda_i d_{(i)}$, $i = i + 1$, go to step (ii).

6. Numerical Experiment

6.1. The Parameters Selection. Before our experiments, there are three important parameters to be selected. The first parameter is the smooth factor k . Some researchers [14] gave them the upper in the each optimization problem as follows:

$$\begin{aligned} k_{p_s}(m, \varepsilon) &\leq \sqrt{\frac{0.0927m}{2\varepsilon}}, \\ k_{p_2}(m, \varepsilon) &\leq \sqrt{\frac{0.0909m}{2\varepsilon}}, \\ k_{p_4}(m, \varepsilon) &\leq \sqrt{\frac{0.0526m}{2\varepsilon}}, \end{aligned} \quad (16)$$

where m is the number of sample and ε is the accuracy. After gaining the upper value, we select the weight parameters C and the kernel parameters ν by using 5-fold cross validation. This whole training and testing procedure is repeated 5 times with different training and testing partitions; finally, the results on the testing partitions are averaged and reported. The figure of 5-fold cross validation can be found in Figure 2. The optimal value of β is chosen from the range $\beta = \{0.1, \dots, 1\}$ and the Δ in linearly decaying function is set 10^{-6} .

FIGURE 2: The 5-fold cross validation.

6.2. Evaluation Criterion and Datasets. According to the proposed FSSVMs, the accuracy is not suitable for the high imbalanced datasets. We employ the geometric mean of the acc^+ , acc^- [15] to evaluate the performances of algorithms in our experiments:

$$G \text{ means} = \sqrt{\text{acc}^+ \times \text{acc}^-}, \quad (17)$$

where $\text{acc}^+ = TP/n^+$ and $\text{acc}^- = TN/n^-$ indicate the accuracy rate of the positives and the negative, respectively. The impact of the change of acc^+ or acc^- value on the G value depends on the size of acc^+ or acc^- value: the smaller acc^+ or acc^- value, the bigger the change of the G value. That means the more minority class samples have wrong points, the greater the cost of misclassification gain.

We demonstrate the effectiveness of the selected FSSVMs with five benchmark real-world imbalanced datasets from the UCI machine learning repository [16]. These real-world datasets contain some outliers and noisy examples and the features can be found in Table 1.

6.3. Experimental Results. For the differentiability of the smooth functions is different, some of the first-order differentiable FSSVMs or SSVMs are designed to test under the BFGS algorithm and the two-order differentiable ones are tested under the Newton-Armijo algorithm. Among the methods, there are three kinds of smooth functions and two kinds of decaying fuzzy membership functions to be chosen. For brevity, We use subscript operator l, e, s, p_2, p_4 to stand for the linear-decaying function, exponential-decaying function, $p_s(x, k)$, $p_2(x, k)$, and $p_4(x, k)$ in the FSSVMs. The result of experiments can be found in Tables 2 and 3. From the tables, it is clear that the G-means of the FSSVMs are better than those of the SSVMs and the normal SVM for the five datasets. Those show that the two fuzzy memberships and the imbalanced factor play an active role in the FSSVMs. On the other hand, the smooth functions change the constraints of the optimization into the unconstraint

TABLE 1: Details of the imbalanced selected for our study.

Dataset	Pos	Neg	Total	Imb. ratio	Total class.	Pos class.
Abanole	103	4074	4177	49	29	15
Yeast	51	1433	1484	32.3	10	5
Satimage	626	5089	6435	9	29	4
Pima	268	500	768	1.8	2	1
Haberman	81	225	306	2.5	2	2

TABLE 2: FSSVMs compare with SSVM and the normal SVM on five datasets for G-means (%) and time (s) (exclusive the time of the preprocessing normal SVM). All of the SSVMs are based on the BFGS algorithm and the two kinds of fuzzy membership functions and three kinds of smooth functions are used in the FSSVMs.

Data	SSVM					FSSVM						
	N-SVM	S_s -SVM	S_{p2} -SVM	S_{p4} -SVM	FS_s -SVM	Linear decaying		FSVM	Exponential decaying			
						FS_{p2} -SVM	FS_{p4} -SVM		FS_s -SVM	FS_{p2} -SVM	FS_{p4} -SVM	FSVM
Abanole	19.39	17.92	18.73	20.45	72.72	71.43	73.26	29.82	65.25	64.08	70.16	28.99
	14.32	1.03	1.16	1.29	1.03	1.78	1.94	4.06	1.18	2.02	2.82	4.56
Yeast	67.68	58.62	59.46	61.36	83.23	84.67	86.60	71.80	83.54	82.70	84.44	70.80
	56.23	11.56	12.24	9.56	11.24	13.46	10.20	20.48	7.88	9.08	6.26	23.48
Satimage	81.05	82.66	81.36	81.84	90.10	89.26	91.28	89.28	90.56	93.06	91.64	89.68
	25.91	14.16	20.24	21.62	22.90	23.02	23.03	26.24	16.86	21.40	21.71	28.24
Pima	68.86	69.08	68.92	70.04	72.64	73.20	71.46	69.88	72.80	71.58	72.47	69.64
	3.54	2.12	2.28	2.36	2.46	3.12	2.46	2.86	2.46	3.24	3.24	3.88
Haberman	42.46	48.66	49.24	45.88	61.56	62.21	62.46	62.47	64.28	63.48	63.48	62.86
	1.20	0.26	0.28	0.36	0.46	0.32	0.46	1.34	0.46	0.28	0.28	1.56

TABLE 3: FSSVMs compare with SSVM and the normal SVM on five datasets for G-means (%) and time (s) (exclusive the time of the preprocessing normal SVM). All of the SSVMs are based on the Newton-Armijo algorithm and the two kinds of fuzzy membership functions and two kinds of smooth functions are used in the FSSVMs.

Data	SSVM					FSSVM				
	N-SVM	S_s -SVN	S_{p4} -SVM	FS_s -SVM	FS_{p4} -SVM	Linear decaying		FSVM	Exponential decaying	
						FS_{p4} -SVM	FS_{p4} -SVM		FS_s -SVM	FS_{p4} -SVM
Abanole	19.39	17.80	19.84	71.86	72.96	29.92	70.45	70.16	31.24	
	14.32	0.89	0.98	0.56	0.28	1.56	0.28	0.24	1.26	
Yeast	67.68	58.46	62.06	80.40	85.54	72.06	82.54	84.20	71.46	
	56.23	8.06	4.56	8.24	5.20	14.48	3.88	3.26	12.48	
Satimage	81.05	81.86	82.54	90.32	91.46	90.28	93.72	94.40	91.28	
	25.91	12.24	16.45	15.56	17.28	28.24	11.86	11.71	26.42	
Pima	68.86	69.24	70.56	71.88	72.28	70.64	72.43	73.68	71.04	
	3.54	1.12	1.36	0.86	0.46	1.88	0.46	0.24	1.68	
Haberman	42.46	52.20	50.56	61.24	62.60	60.26	64.96	66.48	61.48	
	1.20	0.12	0.36	0.16	0.11	0.56	0.16	0.08	0.24	

ones, which make the BFGS algorithm and the Newton-Armijo algorithm be used in the computational process. Because the features and the imbalanced ratios of the datasets are discriminating, the effectiveness of tests is different: for the higher imbalanced ratio datasets, it is better to select the linear decaying FS_{p4} -SVM and the BFGS algorithm, and the fuzzy membership function can well weaken the effect of the outliers and noise; for the lower imbalanced ratio datasets, it is better to choose the exponential decaying FS_{p4} -SVM and the

Newton-Armijo algorithm. It seem that the smooth function $p_4(x, k)$ increases the complexity of the optimization; in fact, it makes the speed of convergence become faster in Newton-Armijo algorithm. Obviously, the running time (exclusive the time of the preprocessing normal SVM) of the Newton algorithm is much smaller than that of the other SVMs.

Table 4 presents some of the best FSSVMs which are compared with some other methods for the five datasets. From the results, we can observe that most of the FSSVMs methods

TABLE 4: The best G-means (%) of the proposed FSSVMs compare with other class imbalanced methods.

Methods	Abanole	Yeast	Satimage	Haberman	Pima
NSVM	19.39	58.80	81.05	42.46	68.86
Under.	73.19	83.31	89.13	62.06	72.51
Over.	72.73	83.77	88.75	64.49	72.77
SMOTE	72.69	83.23	87.64	62.33	73.08
ADASYN	62.91	82.46	88.21	63.76	66.25
Z-SVM	56.26	78.89	86.43	62.89	74.94
$F_{lS_{p4}}$ SVM	73.26	—	—	—	73.68
F_{eS_s} SVM	—	85.54	—	—	—
$F_{eS_{p4}}$ SVM	—	—	94.40	66.48	—

obtained better classification results than the results given by the undersampling, oversampling, SMOTE, ADASYN, Z-SVM, and the normal SVM training. But for the different datasets, the optimal method is different and it needs to select the best combination among the fuzzy memberships functions and smooth functions.

7. Conclusions

Choosing a proper fuzzy membership function and a proper smooth function are quite important to solve classification problem with FSSVM. A kind of FSSVM with two fuzzy functions and three smooth functions for nonlinear classification has been proposed in this paper, which can use the BFGS algorithm or Newton-Armijo algorithm to compute. Experiments results confirm that it is effective according to the datasets features to choose fuzzy membership function and smooth function. It can achieve better performance on reducing the disturbance of outliers and noise than some existing methods in imbalanced datasets.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

The project is supported by the Natural Science Ningxia Foundation (NZ13095).

References

- [1] G. Wu and E. Y. Chang, "KBA: kernel boundary alignment considering imbalanced data distribution," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 6, pp. 786–795, 2005.
- [2] I. Maglogiannis, E. Zafiroopoulos, and I. Anagnostopoulos, "An intelligent system for automated breast cancer diagnosis and prognosis using SVM based classifiers," *Applied Intelligence*, vol. 30, no. 1, pp. 24–36, 2009.
- [3] S. Cai, R. Zhang, L. Liu, and D. Zhou, "A method of salt-affected soil information extraction based on a support vector machine

with texture features," *Mathematical and Computer Modelling*, vol. 51, no. 11-12, pp. 1319–1325, 2010.

- [4] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 9, pp. 1263–1284, 2009.
- [5] J. M. Matías, J. Taboada, C. Ordóñez, and W. González-Manteiga, "Partially linear support vector machines applied to the prediction of mine slope movements," *Mathematical and Computer Modelling*, vol. 51, no. 3-4, pp. 206–215, 2010.
- [6] G. Weiss, "Mining with rarity: a unifying framework," *ACM SIGKDD Explorations Newsletter*, vol. 6, no. 1, pp. 7–19, 2004.
- [7] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.
- [8] S. Zou, Y. Huang, Y. Wang, J. Wang, and C. Zhou, "SVM learning from imbalanced data by GA sampling for protein domain prediction," in *Proceedings of the 9th International Conference for Young Computer Scientists (ICYCS '08)*, pp. 982–987, November 2008.
- [9] N. Chawla, N. Japkowicz, and A. Kolcz, "Editorial:Special issue on learning from imbalanced data sets," *ACM SIGKDD Explorations Newsletter*, vol. 6, no. 1, pp. 1–6, 2004.
- [10] H.-P. Huang and Y.-H. Liu, "Fuzzy support vector machines for pattern recognition and data mining," *International Journal of Fuzzy Systems*, vol. 4, no. 3, pp. 826–835, 2002.
- [11] Y. X. Yuan, "A modified BFGS algorithm for unconstrained optimization," *IMA Journal of Numerical Analysis*, vol. 11, no. 3, pp. 325–332, 1991.
- [12] Y. X. Yuan and R. H. Byrd, "Non-quasi-Newton updates for unconstrained optimization," *Journal of Computational Mathematics*, vol. 13, no. 2, pp. 95–107, 1995.
- [13] O. L. Mangasarian and D. R. Musicant, "Lagrangian support vector machines," *Journal of Machine Learning Research*, vol. 1, no. 3, pp. 161–177, 2001.
- [14] Y.-B. Yuan, J. Yan, and C.-X. Xu, "Polynomial smooth support vector machine (PSSVM)," *Chinese Journal of Computers*, vol. 28, no. 1, pp. 9–17, 2005.
- [15] H. He, Y. Bai, E. A. Garcia, and S. Li, "ADASYN: Adaptive synthetic sampling approach for imbalanced learning," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN '08)*, pp. 1322–1328, June 2008.
- [16] <http://archive.ics.uci.edu/ml/>.