International Journal of Control and Automation
Vol.7, No.10 (2014), pp.167-178
http://dx.doi.org/10.14257/ijca.2014.7.10.16

Research on Dynamic Cost-Sensitive SVM Classifier based on Chaos Particle Swarm Optimization Algorithm

Ruili Zhang

[1]*Department of Computer and Information Engineering, Heze University, Heze 274015, Shandong, China*
[2]*Key Laboratory of computer Information Processing, Heze University, Heze 274015, Shandong, China*
hzxy_zrl@163.com

Abstract

In order to improve the performance of Support Vector Machine (SVM) classifier for imbalanced data, this paper proposes dynamic cost-sensitive SVM classifier based on chaos particle swarm optimization (CPDC_SVM). Firstly, this paper introduces dynamic cost-sensitive thought to SVM classifier, and gives the method for structuring dynamic cost and cost-sensitive SVM model. Secondly, we propose the evaluation methodology performance for classifier, and adopts decimal base to code the particles. At last, chaos thought is introduced in particle swarm optimization algorithm, and the Algorithm of the dynamic cost-sensitive SVM classifier is given, which improves convergent speed and accuracy of particle swarm optimization, and can optimize dynamic cost-sensitive SVM well, so CPDC_SVM adds effectively the convergence speed and accuracy for the particle swarm optimization algorithm. Experimental results show CPDC_SVM has higher precision than traditional SVM classifier, and dynamic cost and chaos particle swarm optimization can improve the performance for classifier.

Keywords: *dynamic cost-sensitive; chaotic particle swarm optimization; SVM classifier*

1. Introduce

Support vector machine (SVM) based on statistical learning theory is a new machine learning method, and it can well resolve such practical problems as nonlinearity, high dimension and local minima. As the mainstream statistical machine learning methods, Support Vector Machine has good performance, and is applied in the image detection, biometrics, and information security. There are classic one-against-one SVM, one-against-rest SVM, the directed acyclic graph SVM, and the decision tree SVM. However, the existing classical SVM classifications are sensitive to noise and aren't suitable to imbalance dates, and it focus overmuch the performance of the majority class in data. How to design a new SVM model for fitting the imbalance data is a hot spot research. There are some improved SVM models for the unbalanced data classification. Gonzalez introduced a new support vector machine GSVM, which is specially designed for bi-classification problems where balanced accuracy between classes is the objective [1]. Mordelet proposed a new method for PU learning with a conceptually simple implementation based on bootstrap aggregating (bagging) techniques [2]. Zhao proposed a weighted maximum margin criterion to optimize the data-dependent kernel, which makes the minority class more clustered in the induced feature space [3]. The paper [4] proposed the structural imbalance SVM, [5] and [6] improved directly C- SVM model, and used the different penalty factors for the different classes. Liu adopt AUC to measure the performance of classifier. In order to overcome the noise to influence the classifier

ISSN: 2005-4297 IJCA

performance, the researchers put forward a fuzzy SVW model in the unbalanced data [7]. The paper [8] conducted a comparative study on the effectiveness of these strategies in the context of imbalanced text classification using Support Vector Machines classifier. The paper [9] presents a novel second-order cone programming (SOCP) formulation, based on the LP-SVM formulation principle: the bound of the VC dimension is loosened properly. The paper [10] propose an efficient weighted Lagrangian twin support vector machine (WLTSVM) for the imbalanced data classification based on using different training points for constructing the two proximal hyper planes.

The parameters selection of SVM has great influence on the classification accuracy, and the common methods for optimizing parameters of SVM are the genetic algorithm, the particle swarm optimization algorithm, etc [11]. The convergence speed of genetic algorithm is slow, and it is easy to fall into the local optimum. The convergence speed of Particle Swarm Optimization Algorithm is faster more than Genetic algorithm, and its principle is simple. However, Particle Swarm Optimization Algorithm is simple, but it is easy to fall into local optimum points, and the local search ability is weak. The chaotic motion is random and isn’t sensitivity to initial conditions, so the paper proposes the dynamic cost-sensitive SVM classifier based on chaos particle swarm optimization algorithm (CPDC-SVM). CPDC-SVM improves the diversity of population, and enhances the convergence speed and precision, so as to optimize better SVM. CPDC-SVM introduces the dynamic cost-sensitive to optimize the classifier, and uses cost-sensitive support vector machine to build a classifier. By finding the optimal cost, constructs a new dynamic cost-sensitive misclassification weights, and solves the classifier problem for the unbalance data.

2. Dynamic Cost

In many practical applications, the different misclassification error usually associates with unequal cost. In medical diagnosis, for example, the cost of the wrong diagnosis patient as possible health is much higher than that of the healthy person is diagnosed as a patient. The current cost-sensitive algorithms adopt usually the fixed static cost, and the classifiers are unstable. The generalization of static cost isn’t strong, so the static cost is used in experiments and prospective study, and can't really been applied to real. Aiming at the shortcomings of the static cost, this paper puts forward a dynamic misclassification cost, and it combines expert experience to forms an objective dynamic misclassification cost space. Misclassification cost space comes from different experts in application area, so its value has representativeness, universality, authenticity and objectivity, and can reflect truly the class distribution characteristics in data sets. On the basic of different data sets, the optimization algorithm selects the optimal value from the misclassification cost space, and the trained classifier can avoid over-fitting and less learning. According to the dynamic cost mechanism, need to form objective misclassification cost space for any data sets in application, and the specific algorithm can be applied in different domains. The optimization algorithm can search dynamically misclassification cost in different data sets, so the optimal misclassification cost combine with a specific algorithm to structure classifier, its performance will reach optimal state. Dynamic composition of misclassification cost first determine application domain data sets, and the application in the field of data sets a pointer to a. Then select for application data sets in different situations of different application situation. Finally combining with the application domain expert experience and background, and different cases of misclassification generation of value, and then to sort by different generations value in ascending order, get an ascending sequence.

Dynamic misclassification costs are constituted by the fixed data in different application domain and select the different values in various circumstances. Finally, the misclassification costs are gotten by the experience expert and knowledge background in the applications and its values are sorted in ascending order as follows. $\cos t_0(1,0)\cos t_1(1,0)\cos t_2(1,0)\ldots\cos t_{m-1}(1,0)$. If $\mathrm{C}_a = \cos t_0$, $C_b = \cos t_{m-1}(0,1)$. The closed interval $[C_a, C_b]$ is misclassification cost space, and there is reason to believe that the misclassification cost values in the interval are feasible.

3. Cost-sensitive SVM

Cost-sensitive learning has been hot research in machine learning [12, 13]. A lot of cost-sensitive methods have been in many real-world applications, such as diagnosis, fraud detection and successful application in the business decisions [14, 15]. In many practical applications, different misclassifications have disparity costs, for example, disease diagnosis, fraud detection, business decisions, and so on [16]. These vendors classification problem is often referred to cost-sensitive learning problems, which is designed to minimize the total cost of misclassification [17]. The traditional SVM bases on accurate algorithm, and ignores the cost in the classification process. Traditional SVM transforms the learning problem to a quadratic programming, in other word, the problem learning problems turns into minimizing the objective function.

$$R(w,\xi) = \frac{1}{2}\|w\|^2 + C\left(\sum_{i=1}^{n} \xi_i\right)$$

$$s.t.\ y_i(x_i \times w + b) \geq 1 - \xi_i,\ \xi_i \geq 0, i = 1, \ldots., n.$$

Cost-sensitive SVM Consider that misclassification cost is asymmetric, and the corresponding quadratic programming is expressed as follows.

$$R(w,\xi) = \frac{1}{2}\|w\|^2 + C\left(\sum_{i=1}^{n} \cos t_i \xi_i\right)$$

$$s.t.\ y_i(x_i \times w + b) \geq 1 - \xi_i,\ \xi_i \geq 0, i = 1, \ldots., n.$$

$\|w\|^2$ is the complexity of cost model, and $\cos t_i \xi$ is different misclassification costs for sample. $\sum_{i=1}^{n} \cos t_i \xi_i$ is experience cost, and it combines asymmetric misclassification costs of different categories. C is relaxation factor, and also called the penalty factor. Relaxation factor C controls the punishment degree for misclassification and the balance between the structure and experience cost. If relaxation factor C is larger, the minimal training error is emphasized. Otherwise, selecting larger C value means to emphasis on the maximum interval for classifications. Obviously, the above formula considers the different categories and misclassification costs. However, when data sets are imbalance, the traditional cost-sensitive SVM has higher bias for most classes, so traditional cost-sensitive SVM adopts the fixed misclassification cost value. In order to make the class SVM to solve the problem of heterogeneous distribution, the dynamic SVM misclassification cost mechanism is introduced, so that it can adapt to different subsets.

$\{x_i, y_i, \cos t_i\}$, $i = 1, 2, \ldots, \mathrm{m}$, $y_i \in \{-1, +1\}$, $x_i \in R^d$, $\cos t_i \succ 0$ is misclassification cost of training example. The above formula is transferred as follows after the misclassification is considered.

$$\min\left(\frac{1}{2}\|w\|^2+\left(\sum_{i=1}^{l}\left(C_i\xi_i\cos t_i\right)\right)\right)$$

$$s.t.\, y_i\left(x_i\times w+b\right)\geq 1-\xi_i,\ \xi_i\geq 0, i=1,\ldots.,\mathrm{m}\ .$$

C_i is misclassification parameter, and the above optimization can be turned into dual form by the introduction of the Lagrange multiplier. The above formula is changed as follow.

$$\max\left(\sum_{i=1}^{l}\alpha_i-\frac{1}{2}\sum_{i}^{l}\sum_{j=1}^{l}\alpha_i\alpha_j y_i y_j\left(x_i\cdot x_j\right)\right)$$

$$s.t\sum_{i=1}^{l}\alpha_i y_i=0,\ 0\leq\alpha_i\leq\cos t_i\times C_i, i=1,2,\ldots,\mathrm{m}\ .$$

By selecting the optimization under the linear separable, the optimal classification function of cost-sensitive SVM is as follows.

$$f\left(x\right)=\mathrm{sgn}\left\{\sum_{i=1}^{l}\alpha_i^{\ *}y_i\left(x_i\cdot x\right)+b^*\right\}$$

b^* is the offset of optimal hyperplane, according to the symbol $f\left(x\right)$ to determine the ownership. By the kernel function, the cost-sensitive SVM is as follow in linear inseparable.

$$f\left(x\right)=\mathrm{sgn}\left\{\sum_{i=1}^{l}\alpha_i^{\ *}y_i K\left(x_i\cdot x\right)+b^*\right\}$$

Experience cost $\left(\sum_{i=1}^{l}\left(C_i\xi_i\cos t_i\right)\right)$ considers different misclassification cost, and makes SVM with higher bias for misclassification cost. The bias makes the classification boundary to move the majority classes, and misclassification for minority class is greatly reduced. Equation (4-34) is cost-sensitive SVM, and α_i is cost-sensitive support vector in $0\prec\alpha_i\leq\cos t_i\times C$.

4. Performance of classifier

The misclassification cost isn’t the same in different subset of the same data sets, so this paper constitutes misclassification costs space by the viable cost of different data sets, and combines the experience and knowledge of the application specialists, it is key to find the optimal misclassification cost is key. A practical classification must ensure omission factor at the same time as much as possible to reduce false alarm rate, so two evaluating indicator are taken into account, and their interrelated and impact are also considered, so that the overall effect is optimal. G-mean function has compromise, and the evaluation parameters tend to be small or large at the same time. $G-mean=\sqrt{\alpha\times\beta}$ .If the parameters $\sqrt{\alpha}$ and $\sqrt{\beta}$ are geometrical mean, G-mean is extremum. This paper selects the response rate (Re) and the precision rate (PR) as parameters α andβ,and find the minimum by adjusting cost-sensitive weight. The response rate (Re) and the precision rate (Pr) is the following expression.

$$\mathrm{Re}=\frac{TP}{TP+FN}\times 100\%$$

$$\mathrm{Pr}=\frac{TP}{TP+FP}\times 100\%$$

FP and TP denote the positive samples number of correct and improper identification, and FN and FP is negative number.

In order to measure the cost that is searched in the cost space is the optimum in data subset, extends particularly the G-mean, and the dynamic cost function is defined as follows.

$$f(\cos t)=G\left(\mathrm{Re}_{\mathrm{rare}}(\cos t),\mathrm{Pr}_{\mathrm{rare}}(\cos t),\mathrm{Re}_{more}(\cos t),\mathrm{Pr}_{more}(\cos t)\right)$$
$$=\sqrt{\mathrm{Re}_{\mathrm{rare}}(\cos t)\times\mathrm{Pr}_{\mathrm{rare}}(\cos t)\times\mathrm{Re}_{more}(\cos t)\times\mathrm{Pr}_{more}(\cos t)}$$

$\cos t$ is the value in misclassification space C, a point value, and G represents G-mean. $\mathrm{Re}_{\mathrm{rare}}(\cos t)$ is the response rate of rare class after applying the dynamic cost in data sets, and $\mathrm{Pr}_{\mathrm{rare}}(\cos t)$ is the precision rate. $\mathrm{Re}_{more}(\cos t)$ is the response rate of majority class after applying the dynamic cost in data sets, and $\mathrm{Pr}_{more}(\cos t)$ is the precision rate. If the misclassification cost of positive class is too high, the majority instances will be classified as the positive samples, Re=TP/(TP+FN), Pr=TP/(TP+FP), FN is more small and Re is more large. Otherwise, if the misclassification cost of negative class is high, the majority instances will be classified as the negative, so that the classifier has low precision rate. The formula $f(c)$ aims at finding the cost of the positive misclassification, so that the classifier can balance the precision and response rate. In other word, the classifier increases the precision rate (Pr) and response rate (RE) of the positive. The function $f(c)$ is the best misclassification cost, and it is defined as follows.

$$f(\cos t)=\arg\max_{\cos t\in COST} f(\cos t)=\arg\max_{\cos t\in COST}\left(\sqrt{\mathrm{Re}_{\mathrm{rare}}(\cos t)\times\mathrm{Pr}_{\mathrm{rare}}(\cos t)\times\mathrm{Re}_{more}(\cos t)\times\mathrm{Pr}_{more}(\cos t)}\right).$$

The dynamic cost applied to data set is the best balance in the misclassification space, and it is the corresponding point when the precision and response rate have the optimal balance between rare and majority class.

5. Dynamic cost-sensitive SVM Classifier Based on Chaos Particle Swarm Optimization Algorithm

Particle Swarm Optimization (PSO) is a famous swarm intelligence algorithm, and each candidate solution is a particle, so the collection of all solutions is the particle swarm. Standard PSO algorithm selects randomly the primary particles, so it is easy to fall into local optimum and can't find the global optimal. This paper improves weakness of PSO algorithm, in order to improve initial unbalanced particle population, uses chaotic sequence to replace random choice and enhances the quality of initial particles. Furthermore, avoid precocity the precocity for particles by chaotic disturbance, and help particle escape from the local optimum.

5.1. Chaotic Initialization

Standard PSO algorithm uses random initial particles to lead to some particles away from the global optimum, so affects the convergence rate. Chaotic sequence initialization can make initial particle populations to get the global optimum, thus overcome the disadvantage of slow convergence. Firstly, Logistic mapping comes into being the chaotic sequence, and the expression is as follows.

$$z_{n+1} = \mu z_n (1 - z_n), \quad n = 1, 2, \ldots, N - 1$$

μ is the control parameter, n is the iteration number, z_n is the nth iteration of chaotic number. $z_1 \in [0,1]$, $\mu \in (2,4)$. Adopt real-coded in this paper, and each chromosome is composed by the following variables in d-dimensional space.

$C_1, C_2, \ldots, C_m, \cos t_1, \cos t_2, \ldots, \mathrm{cost}_m$.

Step1:Generate randomly n-dimensional vector $Z1 = (Z_{11}, Z_{12}, \ldots, Z_{1n})$,according to formula 9, $Z_1, Z_2, \ldots, Z_N$.

Step 2: Use chaotic sequence $z = (Z_1, Z_2, \ldots, Z_N)$, and set initial position by mapping each vector in chaotic sequences. The way is as follows. $x_{ij} = p_{\min} + (p_{\max} - p_{\min}) Z_{ij}$.

$p_{\min}$ and $p_{\max}$ are the minimum and maximum. $i = 1, 2, \ldots, N$, $j = 1, 2, \ldots, n$. x_i is the initial position of the particle.

Step 3: Calculate the fitness value of objective function for each particle, and select the best particles with number m. The particles are $x_i = (x_1, x_2, \ldots, x_m)$.

Step 4: generate randomly the initial velocity for particle x_i , $i = 1, 2, \ldots, x$.

5.2. Chaotic Disturbance

Chaotic disturbances avoids premature, and help the particles escape from the local optima, to avoid falling into the local optima. The steps are as follows:

Step1: Generate randomly n-dimensional vector $\eta_1 = (\eta_{11}, \eta_{12}, ..., \eta_{1n}), \eta_{1i} \in [0,1], i = 1, 2, ..., n$, according to the formula 9, get the chaotic sequence.

Step 2: Set chaotic disturbance range $[-\delta, \delta]$, and Mapping the particles offset by chaotic sequence $\Delta x_j = -\delta + 2\delta\tau_{1j}, j = 1, 2, ..., n$, offset position for chaotic disturbance can be expressed as follows. $\Delta x = (\Delta x_1, \Delta x_2, ...\Delta x_n)$

Step 3: Use the following formula to calculate new position for particles.

$$x_i^{k+1} = x_i^k + v_i^{k+1}$$

$$\Delta x_i^{k+1} = x_i^k + v_i^{k+1} + \Delta x$$

Step 4: calculate objective function for x_i^{k+1} and Δx_i^{k+1},if objective function for Δx_i^{k+1} is better than x_i^{k+1}, the position Δx_i^{k+1} replaces x_i^{k+1}.

5.3. Algorithm Description for Dynamic Cost-sensitive SVM Classifier

Input:

D : Initial data sets

$Cost$: Dynamic misclassification space

$N_{\max}$: Maximum iterations

w : Inertia weight factor

$c1\ c2$: Accelerating factor

n : Particle Swarm scale

Output:

M: Dynamic cost-sensitive SVM

$best\_cost$: The optimum misclassification cost

Steps of algorithm are as follows.

Step 1: Chaos initialized. $Z_1 = (Z_{11}, Z_{12}, ..., Z_{1n})$, generate randomly n-dimensional vector $Z_1, Z_2, ..., Z_N$, $Z1 = (Z_{11}, Z_{12}, ..., Z_{1n})$.

Step2: Fitness evaluation. Use each particle in data set D, and apply cost-sensitive SVM algorithm to structure classifier Mi（i=1,2,⋯,N. Get missed rate (MI) and false alarm rate (FA) after 10 - fold cross-validation, and use the fitness function to calculate value for each particle.

Step 3: Update the speed and position of each particle. The velocity v_{id}^{k+1} and position for particle i is as follows in the k iteration.

$$v_{id}^{k+1} = wv_{id}^{k} + c_1 \times r_1\left(pbest - x_{id}^{k}\right) + c_2 * r_2\left(gbest - x_{id}^{k}\right) \quad x_{id}^{k+1} = x_{id}^{k} + v_{id}^{k+1}$$

$$x_{id}^{k+1} = x_{id}^{k} + v_{id}^{k+1}$$

Step 4: If the fitness for each particle is superior to the individual extreme $pbest$, $pbest$ is set to current new location. If fitness for each particle is better than the global extreme $gbest$, $gbest$ is set .

Step 5: Disturb the optimal part particles for chaos. The particles with higher fitness is closer to the global optimal, so they are easy to get new optimal particles. Therefore, in order to speed up the search process, some particles search and update, and they are easy to get rid of the local extreme to increase the diversity.

Step 6: If iteration isn't turned to step 2, otherwise, go to step 7.

Step 7: Construct cost-sensitive SVM classifier M by the optimal misclassification $best\_\cos t$, and return $best\_\cos t$ and M. Output the optimal solution, and the algorithm end.

6. Simulation Experiment

Adopt UCI datasets for testing the algorithm, and merger the different class for getting unbalanced data sets. $c_1 = 2$, $c_2 = 2$,the particle scale $n = 20$,maximum iterations $N_{\max} = 150$ .This paper proposes dynamic cost-sensitive SVM classifier based on chaos particle swarm optimization algorithm (CPDC_SVM), and STA_SVM [1], C_SVM [3], FUSSY_SVM [4] are the other algorithms for unbalanced data classification. The positive and negative samples are set as 1: 9, and initial dynamic cost use the artificial interval. The average classification accuracy for 200 experiments is table 1.

Table 1 Average classification accuracy of CPDC_SVM, STA_SVM,C_SVM,FUSSY_SVM

Classification Algorithm	average classification accuracy			
	Lenses	Yeast	hepatitis	Sick
C-SVM	73%	78%	84%	88%
Fuzzy_SVM	75%	82%	87%	86%
STA_SVM	86%	89%	92%	92%
CPDC_SVM	91%	93%	95%	94%

As can be seen from Table 1, the classification accuracy of CPDC_SVM is superior to other algorithms for unbalanced data sets.

Select AUC (Area Under the ROC Curve) to measure the performance for classifier, and the area under ROC curve is more intuitive and understandable. CP_SVM is non-sensitive SVM classifier based on chaos particle swarm, and CPSS_SVM is static cost-sensitive SVM classifier based on chaos particle swarm. Select Lenses, Ecolin, Yeast, Sponge and sick as the test data in UCI. Set the same parameters, and the average AUC of CPDC_SVM , CP_SVM and CPSS is Figure 1 by 200 times tests.

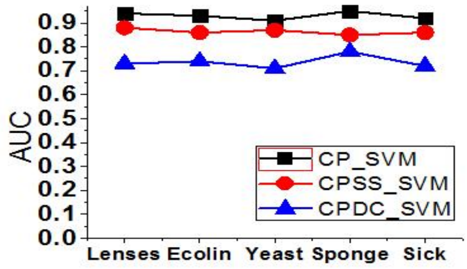

Figure 1 AUC of CP_SVM, CPSS_SVM,CPDC_SVM

If AUC is bigger, the classification accuracy is higher. As can be seen from figure 1, CPDC_SVM has higher classification accuracy. CPDC_SVM uses dynamic misclassification cost, and can search dynamically to get the optimal cost in datasets. CPSS_SVM selects static misclassification cost, and is inflexible and inefficiency.

In order to verify the performance of chaos particle swarm optimization algorithm, the dynamic cost-sensitive SVM that doesn’t introduce chaos particle swarm optimization is called DC_SVM. The cost-sensitive SVM that introduces only the particle swarm optimization is called PDC_SVM. CPDC_SVM this paper presents. Select Iris, Nursery, Yeast, Sponge and Servo as test data. Test in 200 times, the average accuracy of DC_SVM, PDC_SVM, CPDC_SVM is shown in table 2.

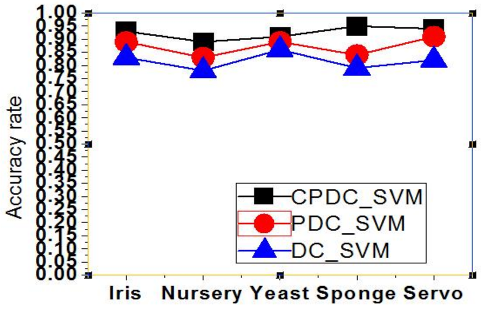

Figure 2 Average accuracy of DC_SVM, PDC_SVM, CPDC_SVM

As can be seen from the figure 2, the chaos particle swarm and ideas improve the classification accuracy for CPDC_SVM.

7. Conclusion

Support vector machine can solve the classification problem with small samples, nonlinear and high dimensions, which has strong practicability. However, Support Vector Machine classifier has bad performance for imbalanced data. This paper proposes dynamic cost-sensitive SVM classifier based on chaos particle swarm optimization, and introduces dynamic cost-sensitive thought to SVM classifier. Because the particle swarm optimization algorithm is easy to relapse into local extremism, the optimization result might be bad. Thus, in the study, chaos thought is introduced in particle swarm optimization algorithm, and chaos particle swarm optimization is presented to optimize SVM, chaos particle swarm optimization can improve diversity of swarm and ergodic property of particle, which improves convergent speed and accuracy of particle swarm optimization, and can optimize SVM well. Experimental results show CPDC_SVM has higher precision than traditional SVM classifier, and dynamic cost and chaos particle swarm optimization can improve the performance for classifier.

Acknowledgments

This work has been partially supported by grants from the research fund of heze university (No. XY12KJ03). This work was supported by the science and technology projects of the Shandong province universities (No. J12LN55, No. J13LN53).

References

[1] L. Gonzalez-Abril, H. Nuñez, C. Angulo and F. Velasco, “GSVM: An SVM for handling imbalanced accuracy between classes inbi-classification problems”, Applied Soft Computing, vol. 17, April **(2014)**, pp. 23-31.

[2] F. Mordelet and J. P. Vert, “A bagging SVM to learn from positive and unlabeled examples”, Pattern Recognition Letters, vol. 37, iss. 1, **(2014)** February, pp. 201-209.

[3] Z. Y. Zhao, P. Zhong and Y. H. Zhao, “Learning SVM with weighted maximum margin criterion for classification of imbalanced data”, Mathematical and Computer Modelling, vol. 54, iss. 3–4, August **(2011)**, pp. 1093-1099.

[4] Q. Q. Zhang, “Research on Anomaly Detection for Imbalanced Data”, Master Dissertation, Nanjing, China, Nanjing University of Aeronautics and Astronautics, **(2010).**

[5] B. Raskutti and A. Kowalczyk, “Extreme Rebalancing for SVMs: A Case Study,”, ACM SIGKDD Explorations Newsletter, vol. 6, iss. 1, **(2004),** pp. 60-69.

[6] R. Akbani, S. Kwek and N. Japkowicz, “Applying Support Vector Machines to Imbalanced Datasets”, Proc of the 15th European Conference on Machine Learning, **, (2004),** Pisa, Italy.

[7] Y. D. Ma, Z. Du and S. Y. Liu, “A New Noise-Immune Fuzzy SVM Algorithm for Imbalanced Data”, Journal of Xi'an Technological University, vol. 28, iss. 3, **(2008)**, pp. 297-300.

[8] A. X. Sun, E. P. Lim and Y. Liu, “On strategies for imbalanced text classification using SVM: A comparative study”, Decision Support Systems, vol. 48, iss. 1, December **(2009)**, pp. 191-201.

[9] S. Maldonado and J. L. López, “Imbalanced data classification using second-order cone programming support vector machines”, Pattern Recognition, vol. 47, iss. 5, **(2014)** May, pp. 2070-2079.

[10] Y. H. Shao, W. J. Chen, J. J. Zhang, Z, Wang *et.al*., “An efficient weighted Lagrangian twin support vector machine for imbalanced data classification”, Pattern Recognition, In Press, Corrected Proof **(2014)**.

[11] G. He and N. J. Huang, “A new particle swarm optimization algorithm with an application”, Applied Mathematics and Computation, vol. 232, iss. 1, **(2014)** April, pp. 521-528.

[12] X. Yu, D. Owens and D. Khazanchi, “Building socioemotional environments in metaverses for virtual teams in healthcare: A conceptual exploration, in: Health Information Science”, Springer, **(2012)**, pp. 4–12.

[13] H. Masnadi-Shirazi and N. Vasconcelos, “Risk minimization, probability elicitation, and cost-sensitive SVMs”, Proceedings of the 27th International Conference on Machine Learning (ICML-10), Omnipress, **(2010)**, pp. 759–766, Haifa, Israel.

[14] H. Masnadi-Shirazi and N. Vasconcelos, "Cost-Sensitive Boosting", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 33, iss. 2, **(2011)**, pp. 294–309.

[15] Z. Qi, Y. Xu, L. Wang and Y. Song, "Online multiple instance boosting for object detection", Neurocomputing, vol. 74, iss. 10, **(2011)**, pp. 1769–1775.

[16] J. G. Kim, K. Choi, G. Kim and Y. Suh, "Classification cost: An empirical comparison among traditional classifier, Cost-Sensitive Classifier, and MetaCost", Expert Systems with Applications, vol. 39, iss. 4, March **(2012)**, pp. 4013-4019.

[17] Z. Zidelmal, A. Amirou, D. Ould-Abdeslam and J. Merckle, "ECG beat classification using a cost sensitive classifier Computer Methods and Programs in Biomedicine", vol. 111, iss. 3, **(2013)** September, pp. 570-577.

Authors

Ruili Zhang, was born in 1980 at Shandong heze, and received the Master of computer application technology from the "Changchun University of Technology" in 2009. He is now a assistant of the Department of Computer and Information Engineering, Heze University. Her research interests include the data mining, Artificial intelligence.