Applied Mathematics & Information Sciences
An International Journal

Classification for Imbalanced and Overlapping Classes Using Outlier Detection and Sampling Techniques

Zeping Yang[1,*] *and Daqi Gao*[1]

[1]School of Information Science and Engineering, East China University of Science and Technology, Shanghai 200237 P.R. China

Received: 20 Jul. 2012, Revised: 11 Aug. 2012, Accepted: 25 Aug. 2012
Published online: 1 Feb. 2013

Abstract: In many real world applications, the example data among different pattern classes are imbalanced and overlapping, which hinder the classification performance of many learning algorithms. In this paper, data cleaning techniques based BNF (the borderline noise factor) is proposed to remove the borderline noise and three under-sampling methods are studied to select the representative majority class examples and remove the distant samples which are useless to form the decision boundary. BNF shows the degree of being a borderline noise and the outlier detection algorithm is improved to clean the whole dataset. Here G-mean (Geometric Mean) is used to define the threshold, which can improve the classification accuracy of minority classes while achieving better performance on the overall classification. The experimental results demonstrate the effectiveness of sampling method with data cleaning techniques based on BNF.

Keywords: Under-sampling, outlier detection, overlapping, imbalanced data, artificial neural network (ANN)

1. Introduction

Pattern classification has a wide range of applications including text document classification, handwritten digit recognition [1] and face recognition [2]. However, in many applications, traditional machine learning algorithms may have difficulty in learning from imbalanced datasets [3]. Imbalanced datasets mean that some classes have much more instances than others. As a result, most traditional classifiers are biased to the majority class and tend to ignore the minority class, even trend the minority class as noise. Severe overlaps between different classes due to various reasons also hinder traditional classifiers. The overlap problem means that a region of data space contains a similar number of training data for each class [4].

Several works suggest that the loss of performance of learning systems is not only caused by class imbalances, but also related to the degree of overlapping among the classes[4][5]. Experiment which has been performed over the sick dataset provided good results (99.65% AUC), and only 6.5% of the examples belong to the minority class in this imbalanced datasets [5].So imbalance and overlap which are factors influencing classifiers performance are not considered in isolation. Currently, most studies consider the two problems separately. Data cleaning techniques have often been used to remove the overlapping that is introduced from sampling methods. These existing methods[2][3][10]can reduce the effects of overlapping on the classifiers, but the samples which are not in the overlapping region have been moved, and some borderline noises have not been removed because of stringent restrictions.

In this paper, BNF (the borderline noise factor) is proposed to show the degree of being a borderline noise. So sampling method with BNF-based data cleaning algorithm can deal with the two issues. Here G-mean (Geometric Mean) [3] is used to define the threshold, which can effectively improve the classification accuracy of minority class while maintaining better performance on the overall classification. The details are included in section3.

The rest of this paper is organized as follows. Section2 describes the related works about methods for imbalanced and overlapping classification. Section3 gives an overview of the proposed approach. Experiment results will be presented in Section4. Finally, we conclude in Section5.

* Corresponding author e-mail: yangzeping@ecust.edu.cn

2. Related Work

2.1. Methods for Imbalanced Learning

Most of the current research activities on imbalanced learning are focus on data sampling methods and algorithm improved [6] [7]. Typically, sampling methods consists of the modification of an imbalanced dataset in order to provide a balanced distribution. The synthetic minority oversampling technique (SMOTE) is a simple and effective method applied in many fields [8]. The SMOTE algorithm selects K-nearest neighbors for each example in minority class firstly, and creates the synthetic samples along the lines between the examples in minority class and their K-nearest neighbors. But the SMOTE algorithm also has its drawbacks including generating the same number of synthetic data samples for each minority examples and increasing the occurrence of overlapping between classes [9]. Some under-sampling methods are usually used to deal with the imbalance problem, such as the one-sided selection (OSS) method, the EasyEnsemble and BalanceCascade algorithms [2]. Sometimes these methods are limited to their overall accuracy, and generalization.

2.2. Data Cleaning Techniques

In this area, some representative works, such as the OSS method [2], the neighborhood cleaning rule (NCL) [10] based on the edited nearest neighbor rule (ENN) and SMOTE with Tomek links (SMOTE+Tomek). SMOTE method adds the artificial samples into the original dataset and causes the overlapping problem, so it needs data cleaning techniques. Given two samples, $x_{\min}$ in the minority class and x_{maj} in the majority class, and the distance between $x_{\min}$ and x_{maj} is denoted as $d(x_{\min},x_{maj})$. Then the pair $(x_{\min},x_{maj})$ is called a Tomek link if there is no sample y such that $d(x_{\min},y) < d(x_{\min},x_{maj})$ or $d(x_{\min},y) < d(x_{\min},x_{maj})$. The neighborhood cleaning rule based on ENN removes samples that differ from two of its three nearest neighbors [10]. The one-sided selection method removes the redundant negative examples and achieves the final training dataset without the borderline and noisy negative examples by using Tomek links. These methods can reduce the effects of overlapping on the classifiers, but the samples which are not in the overlapping region have been moved.

2.3. Outlier Detection

In data mining and machine learning, outliers refer to those samples which are different from the most of samples in datasets. Intuitively, outliers can be defined as given by Hawkins[11], 'an outlier is an observation that deviates so much from other observations as to arouse suspicion that it was generated by a different mechanism'[11]. Outlier detection refers to the task of finding outliers in some certain datasets according to specific rules. Finding such exceptions and outliers is more interesting and useful than identifying the common samples.

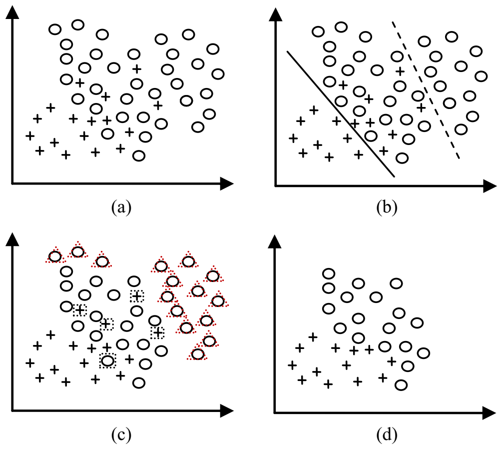

Figure. 1 (a) Original dataset. (b) Different decision boundaries according to different classification methods. (c)The identified useless borderline examples and majority class samples. (d) The dataset after removing useless borderline examples and majority samples.

Recently, most studies which have been applied in a large number of domains were conducted on outlier detection. Some methods which assign to each object a degree of becoming an outlier are more useful than the others which identify outliers as a binary property. The local outlier factor (LOF) is introduced by Breunig [12] to indicate the degree depends on how isolated the object is with respect to the surrounding neighborhood. The larger LOF, the higher the degree of isolation is. In order to avoid computation of density and distance, Neighborhood-based outlier factor (NOF) is proposed by Keping Zhao [13]. Given a sample p, NOF is defined as:

$$NOF(p) = (|kNB(p)| + 1)/(|R - kNB(p)| + 1). \quad (1)$$

Where $|kNB(p)|$is the number of $k$nearest neighbors and $|R\text{-}kNB(p)|$is the number of k neighbors which regard p as neighbor. This method is improved in this paper to handle the imbalance and overlap problem effectively.

3. The Proposed Approach

Imbalance and overlap problems exist in training dataset just like Fig.1a, there are 17 minority samples represented by plus and 34 majority examples denoted by circle in the original dataset. If the classifier is design for higher overall prediction accuracy and tends to classify most of samples in the majority class, the decision boundary which is represented by the solid line (in Fig.1b) may be achieved. 8 minority class samples are misclassified. However, sometimes minority samples are important for some cases. Then the decision boundary which is represented by the dashed line is gotten when the classifier is designed for higher prediction accuracy for minority class. 22 majority class samples are misclassified, just only for identifying 8 minority class samples correctly.

Obviously, it is a challenge to improve the classification accuracy of minority classes while maintaining better performance on the overall classification. If we get rid of the distant majority class samples which are represented by the dashed triangles just like Fig.1c, the overall dataset distribution is changed and the degree of imbalance is decreased. Then the useless borderline samples denoted by the dashed boxes (in Fig.1c) are removed and Fig.1d shows the final clean dataset which is more suitable for traditional classifiers. With these main ideas in mind, we designed the approach that is present in this paper. The whole learning algorithm is divided into four processes.

(1) Get a small-scale training dataset by under-sampling.

Considering the two-class classification problem, the original training datasetΩ is composed of two sets,$\Omega = \Omega^i \cup \Omega^j$, where Ω^icontains all the minority class i examples and Ω^jcontains all the examples belonging to the majority class j, respectively.

Here three under-sampling methods, OSS, NCL and OSS5, are studied. The main idea of OSS5 is the same as OSS and NCL: keep all the samples inΩ^i, and combine the minority class samples and one randomly selected example in Ω^j to the C dataset. Identify noisy data in $\Omega^j$and remove them from the original datasetsΩ. But methods of removing majority class samples are different. OSS classifies the majority class samples with 1-NN rule using the examples in C, while NCL uses ENN which removes examples whose class differs from the majority class of the three nearest neighbors. So OSS5 tries to remove examples based on 5-NN.

(2) Clean the small-scale training dataset by improving the outlier detection method.

The main idea of outlier detection methods is to give a certain degree to each object, depending on how 'isolated' the object is [11] [12]. However, whether the borderline samples are useless or useful for the correct classification is not only related to the degree of isolation, but also the number of neighbors in other classes. Such borderline samples may be minority class or majority class. Based the main idea, NOF [13] is improved to remove the borderline noise in the small-scale training dataset. Here the details should be given.

(a) Find the samples in the overlapping regions as follows.

Set the data set S $= \varnothing$.
For each example x_f in the training dataset
do
Find the nearest neighbor x_{fn} based on the Euclidean distance in n dimensional space.
If x_{fn} belongs to the different class of the example x_f
do
Add $x_{fn}$to the data set S.
End If
End For

(b) For each example x_r in the training dataset S, find K_s nearest neighbors of the same class in the training dataset based on the Euclidean distance. Draw a hyper-dimensional ball Θ_s, which center is x_r and the axes is the largest distance $d_{\max}^{(r)}$ between the example x_r and the sample in the K_s nearest neighbors. $\boldsymbol{kND}(K_s)$ refers to those samples belonging to the different class of the example x_r in the ball Θ_s, and $|kND(x_r)|$ is the number of $\boldsymbol{kND}(x_r)$.

(c) In the training dataset, $\boldsymbol{kNS}(x_r)$ refers to those samples belonging to the same class of the example x_r and their K_s nearest neighbors include the example x_r. $|\boldsymbol{kNS}(x_r)|$ is the number of $\boldsymbol{kNS}(x_r)$.

(d)So *the borderline noise factor* of the example x_r is defined as:

$$BNF(x_r) = \alpha\left(\frac{K_s+\delta}{|kNS(x_r)|+\delta}\right) + \beta|kND(x_r)|. \quad (2)$$

Where$\delta > 0$, is a small positive number, and $(|kNS(x_r)| + \delta) \neq 0$, $0 \le \alpha \le 1$, $0 \le \beta \le 1$, $\alpha+\beta = 1$.

For each example in the training data set S, order the **BNF**. Finally, move the borderline noise from the training dataset and get the clean and small-scale training dataset Ω^s.

(3) Train on the clean and small-scale training dataset.

When getting a clean and small-scale training dataset Ω^s, learning begins. The learning abilities of multilayer feed-forward neural networks depend on such factors as the types of activation functions, the learning rate and the initial weights, etc. Here the activation functions of the hidden and the output neurons are all sigmoid. Suppose the real and the target outputs of the jth output neuron for the pth sample x_p are $y_{pj}(\tau)$ and d_{pj}, the root-mean-squared (RMS) error is

$$E(\tau) = \sqrt{\frac{1}{2Nn}\sum_{p=1}^{N}\sum_{j=1}^{n}(d_{pj}-y_{pj}(\tau))^2}. \quad (3)$$

Where, τ is the number of iterative steps, n is the number of output neurons, and N is the total number of the final training data Ω^s. The weight vector for each iteration program is updated as

$$w(\tau+1) = w(\tau) + \eta\Delta w(\tau) + \alpha(w(\tau) - w(\tau-1)). \quad (4)$$

Here, η is the learning rate, and α is the moment factor. The gradient vector between the hth hidden node and the

Table 1 Confusion matrix for a two-class problem

	Positive prediction	Negative prediction
Actual Positive class	True positive(TP)	False negative(FN)
Actual Negative class	False positive(FP)	True negative(TN)

jth output unit is

$$\begin{aligned}\Delta w_{jh}(\tau) &= -\frac{\partial E(\tau)}{\partial w_{jh}(\tau)} = -\sum_{p=1}^{N}\frac{\partial E(\tau)}{\partial y_{pj}(\tau)}\frac{\partial y_{pj}(\tau)}{\partial \phi_{pj}(\tau)}\frac{\partial \phi_{pj}(\tau)}{\partial w_{jh}(\tau)} \\ &= \sum_{p=1}^{N}(d_{pj}-y_{pj}(\tau))y_{pj}(\tau)(1-y_{pj}(\tau))z_{ph}(\tau).\end{aligned} \tag{5}$$

Where $\phi_{pj}(\tau)$ is the sum of the pth input vector for the jth output unit, and $z_{ph}(\tau)$ is the output of the hth hidden node for the pth input vector. The gradient vector between the ith input node and the hth hidden node is

$$\begin{aligned}\Delta w_{hi}(\tau) &= -\frac{\partial E(\tau)}{\partial w_{hi}(\tau)} \\ &= -\sum_{p=1}^{N}\sum_{j=1}^{n}\frac{\partial E(\tau)}{\partial y_{pj}(\tau)}\frac{\partial y_{pj}(\tau)}{\partial \phi_{pj}(\tau)}\frac{\partial \phi_{pj}(\tau)}{\partial z_{ph}(\tau)}\frac{\partial z_{ph}(\tau)}{\partial \phi_{ph}(\tau)}\frac{\partial \phi_{ph}(\tau)}{\partial w_{hi}(\tau)} \\ &= \sum_{p=1}^{N}\sum_{j=1}^{n}(d_{pj}-y_{pj}(\tau))y_{pj}(\tau)(1-y_{pj}(\tau))w_{jh}(\tau) \qquad (6) \\ &\qquad z_{ph}(\tau)(1-z_{ph}(\tau))x_{pi}.\end{aligned}$$

In order to improve the learning speeds of networks, the activation functions of the hidden and the output neurons are set to be $f(\phi)=3(1+\exp(-\phi/3))^{-1}$[1].

(4) Get a threshold for large-scale training dataset.

There are many performance measures [2], and the most usually used is the whole accuracy (WA). Here G-mean is used to define the threshold θ, which is one of evaluation method to define the classifier's performance for imbalanced dataset. The G-Mean can be defined as follows:

$$G-mean=\sqrt{\frac{TP}{TP+FN}\times\frac{TN}{TN+FP}}. \tag{7}$$

Where TP(True Positive) is correctly predicted of minority class, FP(False Positive) means that the majority class is wrongly classified to the minority class, TN(True Negative) is correctly predicted of majority class, FN(False Negative) means that the minority class is wrongly classified to the majority class. Table1 shows a confusion matrix for a two-class problem. So the whole accuracy (WA) is defined as follows:

$$WA=\frac{TP+TN}{TP+TN+FP+FN}. \tag{8}$$

$$TPR=recall=MIA=\frac{TP}{TP+FN}. \tag{9}$$

$$MAA=\frac{TN}{TN+FP}. \tag{10}$$

Where, MIA refers to the accuracy of minority class and MAA means the accuracy of majority class, respectively. TPR means true positive rate. Positive means minority and negative means majority. G-mean is the square of MIA and MAA. So the larger G-mean value, the better the performance of classifier is. Based on Table 1, some evaluation metrics are defined as:

$$precision=\frac{TP}{TP+FP}. \tag{11}$$

$$F-Measure=\frac{(1+\beta^2)\times recall\times precision}{\beta^2\times recall+precision}. \tag{12}$$

Where β is a coefficient to adjust the relative importance of precision versus recall (usually $\beta=1$).

The above learning algorithm is applied to two class classification, but multi-class pattern recognition has a wide range of applications in real life. One of most popular multi-class modeling approaches, One-Against-All (OAA) [14], is employed in our experiment. OAA method constructs K binary classifiers for K-class pattern classification, and a classifier f_i is trained using the samples of class cl^i against all samples of the other classes. The classifier decision function is defined as:

$$f(x)=\underset{j\in\{1,\ldots,K\}}{\arg\max}\,(f_j(x)-\theta). \tag{13}$$

Then an example is classified in the class whose corresponding classifier has the highest output.

4. Experimental Results

4.1. experimental setting

Experiments are performed to demonstrate that the proposed approach indeed improves classification accuracy of minority classes while maintaining better performance on the overall classification. 10-fold cross-validation technique was used to obtain reliable results. The training dataset was divided into 10 subsets of equal size and each of them had the same proportion of minority and majority class samples. 9 subsets were combined to form the training data and the remaining subset was tested for all possible choices. The results were averaged. The experiments were performed on the

Pentium (R) Dual-Core 2.30GHz computer, with 2GB of RAM.

The performance of seven methods should be compared and included, and seven methods are defined as follows.

1) NCL: the neighborhood cleaning rule based on ENN [10].

2) NCL based on BNF (abbreviated as NCLB): The dataset after using NCL will be removed the borderline noise based on BNF.

3) OSS: the one-sided selection method removes the majority class samples based on 1-NN and gets rid of noise using Tomek links [2].

4) OSS1 based on BNF (abbreviated as OSS1B): OSS1B removes the borderline noise based on BNF, instead of using Tomek links, and the rest processes are the same as OSS.

5) SMOTE with Tomek links (abbreviated as SMTK). SMOTE method creates the artificial samples along the lines between the minority class samples and their K-nearest neighbors. Here $K = 5$. The overlap problem is resolved by using Tomek links.

6) SMOTE based on BNF (abbreviated as SMTB). The borderline noise is removed based on BNF, not using Tomek links. Here $K = 5$.

7) OSS5 based on BNF (abbreviated as OSS5B). OSS5B removes the majority class samples based on 5-NN, and gets rid of the borderline noise based on BNF.

4.2. Data Analysis

The proposed approach is tested on several real-world datasets. Table 2 gives the description for the used datasets, where size refers to the number of samples in the whole training dataset. The new training datasets are constructed by modifying the original datasets to test the learning capabilities from two-class imbalanced problems. *Glass*, *Vehicle* and *Ionosphere* datasets are obtained from UCI repository of machine learning databases [15].

The *Glass* dataset has seven classes, 214 examples in the dataset and contains 10 attributes all of which are numerical. But class4 has no samples in this dataset. Class7 is chosen as the minority class and the rest of the samples as the majority class, and then the new dataset contains 29 minority class samples and 185 majority class examples.

The *Ionosphere* dataset is only consist of two classes (good radar and bad radar), and has 351 samples. Each example is represented by 34 attributes. Instances labeled as “bad radar” form the minority class and instances labeled as “good radar” constitute the majority class, respectively.

The *Vehicle* dataset has 846 data samples, 4 classes (opel, saab, bus and van) and contains 18 attributes. “Van” is chosen as the minority class and the remainder of the whole dataset is chosen as the majority class as suggest in [9].

Table 2 The description for the used datasets

Dataset	Attribute	Size	#min	#maj
vehicle	18	846	199	647
ionosphere	34	351	126	225
glass	10	214	29	185
satimage	36	6435	626	5809

The *Satimage* dataset has 7 classes originally which is obtained from [16]. But class6 has no samples in this dataset and class4 has fewer samples than other classes. Class4 is chosen as the minority class and the rest of the classes are collapsed into one. This gave us a 2-class dataset, with 5809 majority class samples and 626 minority class samples.

4.3. Influence of the parameters

The parameters $\alpha$and βplay a balance role between the degrees of isolated by the same class samples and surrounded by the different class samples. Thus, both α and β would influence the performance of the related algorithms. In our experiments, NCLB, OSS1B, SMTB, OSS5B use bothα andβ. The G-mean values of these methods with varying αon the datasets showed in Table 2 are given to show the role of αandβ. From Fig. 2, when $\alpha$is initialized be 0.3, the best performance might be got on most datasets. Usually, parameters δ =0.1 and K=5.

4.4. Performance comparison

In this section, a comparison of seven methods is given in terms of different evaluation metrics. For each method, the best performance is denoted by bold and italic in each category. From Table 3, it can be found that: 1) sampling methods based on BNF provides the best performance in terms of G-mean and F-measure for all datasets. This means that these methods based on BNF improved accuracy for both minority and majority classes. Furthermore, these methods based on BNF attain better G-mean values compared to them. 2) Sampling methods based on BNF give the better performance than them with Tomek links in terms of G-mean, for example, OSS1B vs. OSS, and SMTB vs. SMTK. So sampling methods based on BNF can handle imbalanced and overlapping datasets, and improve the classification accuracy of minority class without sacrificing majority class. However, some parameters still need to be tuned artificially using sampling methods based on BNF, and an automatic computation method for these parameters should be work out in the future. 3) Three under-sampling methods perform well on all of the datasets, but NCL method

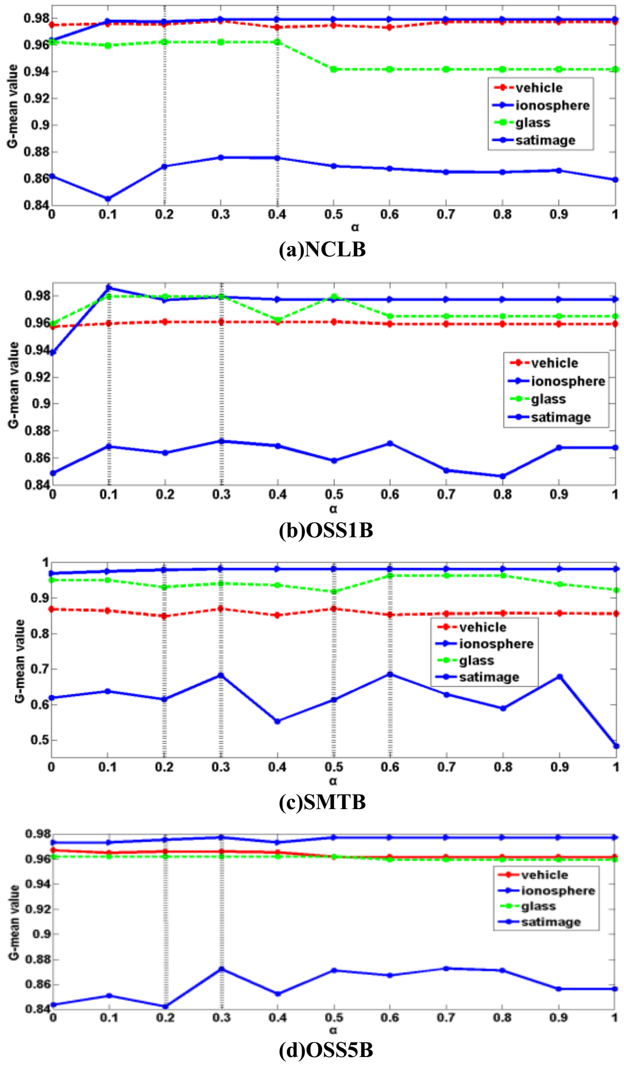

Figure. 2 G-mean values of 5 methods with varying αon the datasets showed in Table2

performed better on most of datasets. In fact, it is difficult to choose a perfect algorithm for all of the datasets. Selecting representative majority class samples without loss of informative ones is still worthy to think about.

5. Conclusion

The imbalance and overlap problems often complicate the traditional classifiers. So it is worthy to save the samples near the borderline and get rid of the distant ones automatically. Whether the borderline data is noise or not is very important for forming the correct decision boundary. BNF is proposed to show the degree of being a noise and the dataset is cleaned by improving the outlier detection method. G-mean is employed to define the threshold, and the G-mean value is larger, the performance of classifier is better. In order to improve the learning speeds of networks and get good generalization performance, the real outputs with suitable factors are amended. The experimental results on bench datasets show that the proposed method can effectively improve the classification accuracy of minority classes while maintaining better performance on the overall classification. In the future, various issues, such as high dimension and multiclass classification should be considered in detail.

Table 3 evaluation metrics and performance comparison of different methods

dataset	methods	WA	MIA	MAA	G-mean	F-measure
Vehicle	NCL	0.9678	0.9598	0.9703	0.9650	0.9335
	NCLB	***0.9707***	***0.9618***	0.9710	***0.9676***	***0.9392***
	OSS	0.9665	0.9492	***0.9719***	0.9605	0.9303
	OSS1B	0.9678	0.9593	0.9705	0.9649	0.9335
	SMTK	0.7703	0.9246	0.7229	0.8100	0.6690
	SMTB	0.8144	0.8980	0.7887	0.8375	0.7031
	OSS5B	0.9668	0.9603	0.9688	0.9645	0.9315
Ionosphere	NCL	0.9809	0.9730	0.9853	0.9791	0.9734
	NCLB	0.9809	0.9810	0.9809	0.9809	0.9736
	OSS	0.9789	0.9730	0.9822	0.9776	0.9707
	OSS1B	0.9809	0.9746	0.9844	0.9795	0.9734
	SMTK	0.9937	0.9905	***0.9956***	0.9930	0.9913
	SMTB	***0.9943***	***0.9921***	***0.9956***	***0.9938***	***0.9921***
	OSS5B	0.9786	0.9667	0.9853	0.9759	0.9701
Glass	NCL	0.9757	0.8966	0.9881	0.9412	0.9091
	NCLB	0.9832	0.9379	0.9903	0.9637	0.9379
	OSS	0.9561	0.9207	0.9616	0.9408	0.8505
	OSS1B	0.9841	***0.9655***	0.9870	***0.9762***	***0.9434***
	SMTK	0.9664	0.9000	0.9768	0.9372	0.8788
	SMTB	0.9664	0.9103	0.9751	0.9421	0.8811
	OSS5B	***0.9846***	0.9310	***0.9930***	0.9615	0.9425
Satimage	NCL	0.8594	0.8652	0.8588	0.8618	0.5468
	NCLB	***0.8694***	***0.8720***	***0.8691***	***0.8705***	***0.5654***
	OSS	0.8635	0.8717	0.8626	0.8671	0.5545
	OSS1B	0.8690	0.8688	0.8690	0.8689	0.5640
	SMTK	0.6003	0.7546	0.5837	0.6622	0.2708
	SMTB	0.6375	0.7492	0.6255	0.6840	0.2869
	OSS5B	0.8499	0.8580	0.8490	0.8533	0.5293

References

[1] Gao Daqi, Li Chunxia, and Yang Yunfan, Pattern Recognition **40**, 2226-2236 (2007).
[2] Jiangguo Wang, Appl. Math. Inf. Sci. **6**, 81-85 (2012).
[3] H.He and E.A.Garcia, IEEE Transactions On Knowledge And Data Engineering **21**, 1263-1282 (2009).
[4] Misha Denil and Thomas Trappenberg, Canadian Conference on AI **1**, 220-231 (2010).

[5] R.C.prati, G.E.A.P.A. Batista, and M.C. Monard, Proc. Mexican Int'l Conf. Artificial Intelligence **1**, 312-321 (2004).
[6] S.Ertekin, J. Huang, L. Bottou, and L. Giles, Proc. ACM Conf. Information and Knowledge Management **1**, 127-136 (2007).
[7] S.Ertekin, J.Huang, and C.L. Giles, Proc. Int'l SIGIR Conf. Research and Development in Information Retrieval **1**, 823-824 (2007).
[8] N.V.Chawla, K.W. Bowyer, L.O. Hall, and W.P. Kegelmeyer, Artificial Intelligence Research **16**, 321-357 (2002).
[9] H.He, Y. Bai, E.A. Garcia, and S. Li, Proc. Int'l J. Conf. Neural Networks **1**, 1322-1328 (2008).
[10] J. Laurikkala, Proc. Conf. AI in Medicine in Europe: Artificial Intelligence Medicine **1**, 63-66 (2001).
[11] D.Hawkins, *Identification of outliers,* Chapman and Hall 1980.
[12] M.M. Breunig, H.P.Kriegel, R.T.Ngand J.Sander, In Proceedings of ACM SIGMOD International Conference **1**, 93-104 (2000).
[13] Keping Zhao, Shuigeng Zhou, Jiehong Guan, and Aoying Zhou, CAAI-10 **1**, 470-475 (2003).
[14] Guobin Ou and Yi Lu Murphey, Pattern Recognition **40**, 4-18(2007).
[15] http://www.ics.uci.edu/˜mlearn/MLRepository.html.
[16] http://www.csie.ntu.edu.tw/˜cjlin/libsvmtools/datasets/multiclass.html.

Zeping Yang received the M.S. degree in School of Information Science and Engineering, East China University of Science and Technology, China, 2005. She is currently a Ph.D. student in East China University of Science and Technology. Her research interests include neural networks, pattern recognition, and machine olfactory.

Daqi Gao received the Ph. D. degree from Zhejiang University, China, in 1996. Currently, he is a professor in East China University of Science and Technology. He is a member of the International Neural Network Society. He has published over 50 scientific papers. His research interests are pattern recognition, neural networks, and machine olfactory.