Proceedings of the 5th International Conference on Information Technology and Application in Biomedicine, in conjunction with
The 2nd International Symposium & Summer School on Biomedical and Health Engineering
Shenzhen, China, May 30-31, 2008

The Generation Mechanism of Synthetic Minority Class Examples

Sheng TANG and Si-ping CHEN

Abstract—**The class imbalance problem, which exists in the field of medical image analysis universally, may cause a significant deterioration to the performance of the standard classifiers. In this paper, the related work on dealing with class imbalance is firstly reviewed, and then a proper generation mechanism of synthetic minority class examples is discussed. According to the analysis, a novel oversampling algorithm with synthetic examples, ADOMS, is proposed by generating synthetic examples along the first principal component axis of local data distribution. The experiments are arranged on 12 UCI datasets and the experimental results show that comparing with other relative methods, algorithm ADOMS is able to alleviate the deterioration of the classification performance effectively.**

I. Introduction

In the classification problem, a dataset is said to be imbalanced if it contains many more examples belonging to one of the classes than to the others [1]. The class imbalance problem exists in the real world universally, especially in the situation of detecting some rare but important things, such as medical diagnosis, text classification, and defective products monitoring. For instance, to most of medical image recognition problems, the positive images are always only a very small part of the total clinical images, compared with the wide-defined negative images.

Although it had been observed for a long time that class imbalance may cause a significant deterioration to the performance of the standard classifiers, lots of relative researchers did not notice that it was a systematic problem. After two workshops on class imbalance were hold at AAAI'00 and ICML'03 respectively and a special issue was published on the SIGKDD explorations in 2004 [2], it has been clear that not only the class imbalance problem itself, but also some complicating factors coming with it, e.g. rare cases and class overlapping, will cause the deterioration of the performance of the classifiers [3],[4].

Resampling approaches is a group of popular and flexible techniques to deal with class imbalance, which attempt to resample the original training dataset, by oversampling the minority class and/or undersampling the majority class [2], e.g. random oversampling, random undersampling, directed oversampling, directed undersampling, oversampling with synthetic examples and combinations of the above algorithms. Without adding any new information into the dataset, random oversampling may lead to overfitting, and random undersampling may discard potentially useful majority class examples. Directed resampling approaches attempt to handle the examples based on some information, but still suffer the same shortcomings as the random ones. In early published papers, undersampling approaches provided more accurate results than oversampling ones, but some recent research results are against the previous conclusions [5].

Manuscript received March 15, 2008. This work was supported in part by the national natural science fund of China No.60772147.

Sheng TANG is with the Department of Biomedical Engineering, Zhejiang University, Hangzhou, China. (e-mail: eric_king_china@hotmail.com).

Si-ping CHEN is with Shenzhen University, Shenzhen, China. (phone: 86-755-26745883; fax: 86-755-26745111; e-mail: chensiping@szu.edu.cn).

II. Adjusting The Direction of Synthetic Examples

A. Oversampling with Synthetic Examples

Oversampling with synthetic examples is introduced to alleviate the overfitting which caused by random and directed oversampling. New synthetic minority class examples will be inserted into the original training dataset obeying some specific rules, which can effectively expand the decision region of the minority class in the feature space, accompanied with populating the number of the minority class examples. However, it also can be predictable that it will also inevitably introduce additional noise into the dataset, since these generated synthetic examples are, at best, only a better approximation of the true distribution [4].

B. Algorithm SMOTE

Chawla introduced an important concrete algorithm of oversampling with synthetic examples, named SMOTE [6], which generated synthetic minority class examples by interpolating between minority class examples and its nearby neighbors. Briefly, for each selected minority class example $\boldsymbol{m}$, one of its K nearest neighbors $\boldsymbol{n}$ was randomly chosen and a *scaling* was also obtained from *Random*(0,1). And then a new synthetic minority class example $\boldsymbol{m'}$ will be inserted in the dataset by constructing its value in each dimension i in the feature space as $m_i' = m_i + scaling * (n_i - m_i)$.

C. Principal Components Analysis

Principal components analysis (PCA), an exploratory data analysis technique, is usually used as a common quantitative method for data reduction. PCA is concerned with explaining the variance structure of the original data. The entire principal components form an orthogonal basis in the feature space and each principal component is orthogonal to the others so there is no redundant information. The first principal component axis occupies the maximal amount of total variance in the feature space, and the second principal component axis occupies the most remaining variance *etc* [7].

D. A Proper Synthetic Examples Generation Mechanism

Obeying the true class distribution should be the ideal way to add examples into the training dataset, and when the specific rules for inserting synthetic examples are not consistent with the underlying class distribution, the noise will be introduced into the training dataset inevitably. In the uttermost condition, the space of the minority class in the feature space could be fully expanded by generating synthetic minority class examples completely randomly, nonetheless the awful classification results could be foreseen and the synthetic examples would be treated as noise directly since they are obviously the improper expression of the underlying class distribution. Considering the underlying class distribution is very hard to be disclosed in the real world dataset, local data distribution should be investigated to make better approximation of the true class distribution.

Analyzing algorithm SMOTE, the impact of synthetic examples to the original feature space is restricted into the local space by interpolating the synthetic one between the processing minority class example ("*center*" for short) and one of its nearby neighbors ("*neighbor*" for short). When the *neighbor* is far away from the *center*, which means that there are only a few examples in the local space near *center* and the true underlying class distribution will just be expressed coarsely, the synthetic minority class example should be inserted into the feature space farther from the *center* to occupy the comparatively vaguely defined space. On the contrary, when the *neighbor* is near the *center*, which means that there are already enough examples in the local space to finely express the underlying class distribution, the synthetic one should be inserted closer to the *center* to avoid disturbing the comparatively well defined space. The above generation mechanism of synthetic minority class examples ("*mechanism*" for short) sounds reasonable, however in SMOTE, only one of the neighbors of the *center* is randomly chosen to represent the local space, and the *mechanism* is even only realized in 1-Dimension which defined by the *center* and the selected neighbor. It is obviously that the *mechanism* should be realized in the entire space more properly.

It should be noted that the space of the real world data is hardly isotropic, and the *mechanism* must be considered in every direction in the feature space respectively. When a synthetic one is generated in the direction where the projections of the local data are sparse, the projection of the synthetic one should be farther from the *center*, and on the contrary, in the direction where the projections of the local data are dense, the projection of the synthetic one should be closer to the *center*.

The local space could be reconstructed using PCA. The first principal component axis occupies the maximal variance, and the next principal component axis occupies the maximum among the rest *etc*, therefore the synthetic one should be farthest from the *center* in the direction of the first principal component axis, and then should be closer in the direction where its occupied variance are smaller, till closest to the *center* in the direction of the last principal component axis. When the synthetic one is generated directly on the first principal component axis through the *center*, its projection on the first principal component axis should be itself and also the farthest from the *center*, and then its projection will be closer to the *center* in the projection which owns smaller variance, therefore it is easy to see that generating the synthetic minority class example along the first principal component axis of local data distribution would fit the *mechanism* most properly.

Figure 1 is an illustration in 2-Dimensional feature space. The square *M* represents the processing minority class example, and five triangles *N1*,*N2*,*N3*,*N4*,*N5* represent its nearby neighbors. When the number of the neighbors is chosen as 3, the local data distribution is composed of *N1*,*N2*,*N3* and *M*. And *N3* is supposed as the chosen neighbor. In SMOTE, the synthetic minority class example *M1* is generated along line *L1* from *M* to *N3*. Following the above analysis, the first principal component axis *L2* of local data distribution is obtained firstly by the Jacobi algorithm, and then, adopting the same distance as *M1*, *M2* is generated along *L2* from *M* to the projection of *N3* on *L2*. The projections of *M1* and *M2* on *L1* are *M1* itself and *M2p* respectively, meanwhile the projections of *M1* and *M2* on *L2* are *M1p* and *M2* itself respectively. According to the *mechanism*, the synthetic one should be farther from *M* on *L2* than on *L1*, since the projections of the local data distribution *N1*,*N2*,*N3* and *M* on *L2* are sparser than the projections on *L1*. It is obviously that *M2*, which is generated along the first principal component axis, fits the *mechanism* better than *M1*, and it is even the most fittable one on the circle *C1*, on which the points own the same distance to *M*.

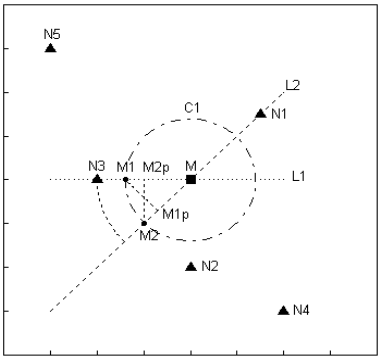

Fig. 1. An illustration of the generation mechanism of synthetic examples

E. Algorithm ADOMS

According to the aforementioned analysis, a novel oversampling algorithm with synthetic examples, named ADOMS (Adjusting the Direction Of the synthetic Minority clasS examples), is proposed, which generates synthetic minority class examples in the feature space by iterating the

following steps:

Step 1: Randomly select one of the minority class examples ***m*** in the original training data as the processing example;

Step 2: Define the neighbor number K (K=1,2,3,…), and calculate the K nearest neighbors of ***m*** in the feature space using the Euclidean distance;

Step 3: Calculate the first principal component axis of local data distribution which composed of ***m*** and its K neighbors in the feature space;

Step 4: Select one of its neighbors ***n*** randomly and calculate the Euclidean distance D between ***m*** and ***n***, then a *scaling* is obtained from *Random*(0,1);

Step 5: Generate a new synthetic minority class example ***m'*** in the feature space, where ***m'*** is generated along the direction from ***m*** to the projection of ***n*** on the first principal component axis through ***m***, and the Euclidean distance between ***m'*** and ***m*** is *scaling***D*.

It is worth noting that ADOMS is in essence equal to SMOTE when the neighbor number K is defined as 1.

III. Experiments

A. Performance Measure

Accuracy is widely used as the performance measure for the classifiers. However it is not proper when the classes are imbalanced, since accuracy will yield biased conclusions by favoring the majority class and cumbering the minority class. For instance, where the proportion of the majority class examples in a classification problem is overwhelming to 99%, you can easily create a classifier which having an accuracy of 99% by simply labeling every example to the majority class. The ROC curve and the geometric mean (GM) are good substitutions of the accuracy in the class imbalance situations [8], where the commonality of these means is they are independent of the distribution of examples between classes. GM is defined as the square root of the product of *TP* and *TN*, where *TP* denotes the accuracy on the positive examples and *TN* is the accuracy on the negative examples. This measure tries to maximize the accuracy on each of the two classes with keeping these accuracies balanced [1].

B. Experimental Datasets

12 datasets which derived from the UCI Repository [9] are chosen as our experimental datasets. The UCI Repository is commonly used as a standard platform for the performance evaluation in the field of machine learning. The experiments will be constrained in the two-class problem for simplicity. If the original UCI dataset owns more than two classes, the class with fewer examples is chosen to be the minority class with combining the others as the majority class. These 12 experimental datasets own different amounts of the examples, dimensions and class imbalance levels, which are all presented in Table 1. The names of the minority class are also introduced between the brackets after the name of the original UCI datasets if there are more than two classes in the original datasets.

It is worth noticing that the successive five experimental datasets from Index 3 to Index 7 in Table 1 are all derived from *Letter* in the UCI Repository. *Letter{M,N}* takes the letter *M* and *N* together as the minority class to get less imbalance situation. On the contrary, *Letter{Z}%2* and *Letter{Z}%4* take the letter *Z* as the minority class and then process the minority class using random undersampling with 2 and 4 times respectively to get higher imbalance situation.

TABLE I
THE MAJOR INFORMATION OF THE 12 EXPERIMENTAL DATASETS

Index	Dataset Name {Minority}	Amounts	Dim	Min%
1	Breast-WPBC	194	33	23.71
2	Heart-Disease {1}	297	13	18.18
3	Letter {A}	20000	16	3.95
4	Letter {M,N}	20000	16	7.88
5	Letter {Z}	20000	16	3.67
6	Letter {Z} %2	19639	16	1.9
7	Letter {Z} %4	19457	16	0.98
8	Pima-Indians-Diabetes	768	8	34.9
9	Segment {Brickface}	2310	19	14.29
10	Shuttle {!Rad Flow}	14500	9	20.84
11	Water-Treatment {Normal}	380	38	9.8
12	Wine {1}	178	13	33.15

C. Experimental Results

BP (Backpropagation) neural network is chosen as the experimental classifier and 10-fold cross validation is employed [10]. To each dataset, under certain parameters of BP network, the successive couples of *TP* and *TN* are collected by changing the decision threshold to calculate the average GM value (*GM*). And to diminish the impact of the specific classifier structure, the parameters of the network are gradually modified and the *GM*s are averaged. The modified parameters of BP network include the number of the hidden nodes, the epoch times and the learning rate.

The motive of the experiments is to compare the abilities of the proposed algorithm ADOMS with several relative oversampling algorithms to alleviate the deterioration of the classification performance of the experimental classifier in the class imbalance situations. The relative oversampling algorithms include random oversampling and SMOTE, and the original experimental dataset is also chosen as the benchmark. All of the datasets are performed normalization before learning. Testing under 10-fold cross validation, each dataset is partitioned into 10 equal-sized blocks. Each block in turn is used as the test data with the remaining 9 blocks as the training data, then the oversampling algorithms are employed on the training data while keeping the test data unchanged. All of three oversampling algorithms will

oversample the amounts of minority class examples to double times. Cross-validation results are repeated and averaged over 10 runs. And then, to alleviate the impact of the specific classifier structure, the parameters of the classifier will be gradually modified and the learning process will be repeated to obtain the average *GM*s. Moreover, the number of the neighbors *K*, which is an important parameter to SMOTE and ADOMS, is also modified from 1 to 3 successively and the synthetic examples under different *K* values will be generated to construct different training data, and then the average performance under different parameter values of SMOTE and ADOMS is obtained. Furthermore, to ease the uncertainty introduced by the randomization which used widely in three oversampling algorithms, the experiments for each oversampling algorithm under different classifier parameters and oversampling parameters will be repeated for three times and the experimental results will be averaged. The final average *GM*s are presented in Table 2 and the corresponding standard deviations are also shown between the brackets. The best *GM* of each dataset is highlighted in bold.

TABLE II
THE *GM*S OBTAINED BY DIFFERENT OVERSAMPLING ALGORITHMS ON THE 12 EXPERIMENTAL DATASETS

Index	Original	Rand Over	SMOTE	ADOMS
1	46.61(34.25)	51.61(24.59)	54.62(24.70)	**61.34(23.95)**
2	21.39(26.17)	59.35(17.32)	**60.07(18.92)**	53.56(15.39)
3	91.14(1.02)	92.79(1.78)	92.97(1.68)	**93.35(1.59)**
4	94.20(1.97)	94.83(1.79)	95.18(1.51)	**95.36(1.40)**
5	78.25(5.47)	85.41(4.06)	86.22(3.61)	**86.57(3.44)**
6	13.85(12.53)	19.09(18.47)	18.60(18.15)	**21.24(18.88)**
7	0(0)	0(0)	0(0)	0(0)
8	67.94(11.75)	72.43(5.83)	72.80(5.22)	**72.86(5.85)**
9	**98.45(1.86)**	97.97(1.75)	98.00(1.68)	98.04(1.63)
10	52.57(14.83)	**57.65(16.40)**	57.18(17.76)	56.91(16.14)
11	88.64(6.40)	**91.52(3.42)**	85.95(5.26)	88.11(4.36)
12	96.10(5.87)	95.84(5.00)	95.88(5.79)	**96.22(4.21)**

The experimental results clearly show that, to most of the datasets, the oversampling approaches can effectively alleviate the deterioration of the classification performance in the class imbalance situations, especially to the datasets on which the classifier performs badly, e.g. *Breast-WPBC*, *Heart-Disease* and *Letter{Z}%2*. The classification performance even sharply improves to almost three times on *Heart-Disease*. However it is frustrated to see that three oversampling methods all work in vain on *Letter{Z}%4*.

Paired student's *t* test is carried out for comparing SMOTE and ADOMS, two concrete oversampling algorithms with synthetic examples. The results with a *P* value less than 0.05 are thought of statistically significance and marked with the gray color in Table 2. Considering the experimental results which are verified statistically significance only, as two different concrete algorithms of oversampling with synthetic examples, ADOMS performs better on 6 of 7 datasets.

IV. CONCLUSION

In this paper, a proper generation mechanism of synthetic minority class examples was discussed intensively, and a novel oversampling algorithm with synthetic examples, ADOMS, was proposed, which generated synthetic minority class examples along the first principal component axis of local data distribution. It was proved by the experiments that ADOMS could effectively alleviate the deterioration of the classification performance in the class imbalance situations, comparing with other relative oversampling algorithms. Since the true underlying class distribution is very hard to be disclosed, it is worth doing more work on the local data distribution to reduce the negative impacts which introduced by the inserted synthetic examples. In the future, we will extend the study to the multi-class problem, and the dataset with the nominal features will also be handled. The number of the nearby neighbors *K* is an important parameter since it markedly influences the construction of local space distribution, thus the impact of *K* will be intensively investigated.

REFERENCES

[1] R. Barandela, J. S. Sanchez, V, Garcia. E. Rangel, "Strategies for learning in class imbalance problems," *Pattern Recogn*, vol. 36, pp. 849-851, 2003.

[2] N. V. Chawla, N. Japkowicz, A. Kolcz, "Editorial: Special issue on learning from imbalanced data sets," *SIGKDD Explor Newsl*, vol. 6, pp. 1-6, 2004.

[3] T. Jo, N. Japkowicz, "Class imbalance versus small disjuncts," *SIGKDD Explor Newsl*, vol 6, pp. 40-49, 2004.

[4] G. M. Weiss, "Mining with rarity: A unifying framework," *SIGKDD Explor Newsl*, vol 6, pp. 7-19, 2004.

[5] G. E. A. P. A. Batista, R. C. Prati, M. C. Monard, "A study of the behavior of several methods for balancing machine learning training data," *SIGKDD Explor Newsl*, vol 6, pp. 20-29, 2004.

[6] N. V. Chawla, K. W. Bowyer, L. O. Hall, W. P. Kegelmeyer, "SMOTE: Synthetic minority oversampling technique," *J Artif Intell Res*, vol 16, pp. 321-257, 2002.

[7] J. Cowe, D. H. Evans, "Automatic detection of emboli in the TCD RF signal using principle component analysis," *Ultrasound Med Biol*, vol 32, pp. 1853-1867, 2006.

[8] M. Kubat, S. Matwin, "Addressing the curse of imbalanced training sets: One-sided selection," *in 1997 Proc. ICML Conf.*, pp. 179-186.

[9] C. Blake, C. Merz (1998). UCI repository of machine learning databases [Online]. Available: http://www.ics.uci.edu/~mlearn/MLRepository.html

[10] A. R. Webb, *Statistical pattern recognition*. New York: John Wiley & Sons, 2002.