Under-Sampling Approaches for Improving Prediction of the Minority Class in an Imbalanced Dataset

Show-Jane Yen and Yue-Shi Lee

Department of Computer Science and Information Engineering, Ming Chuan University
5 The-Ming Rd., Gwei Shan District, Taoyuan County 333, Taiwan
{sjyen, leeys}@mcu.edu.tw

Abstract. The most important factor of classification for improving classification accuracy is the training data. However, the data in real-world applications often are imbalanced class distribution, that is, most of the data are in majority class and little data are in minority class. In this case, if all the data are used to be the training data, the classifier tends to predict that most of the incoming data belong to the majority class. Hence, it is important to select the suitable training data for classification in the imbalanced class distribution problem. In this paper, we propose cluster-based under-sampling approaches for selecting the representative data as training data to improve the classification accuracy for minority class in the imbalanced class distribution problem. The experimental results show that our cluster-based under-sampling approaches outperform the other under-sampling techniques in the previous studies.

1 Introduction

Classification Analysis [5, 7] is a well-studied technique in data mining and machine learning domains. Due to the forecasting characteristic of classification, it has been used in a lot of real applications, such as flow-away customers and credit card fraud detections in finance corporations. Classification analysis can produce a class predicting system (or called a classifier) by analyzing the properties of a dataset having classes. The classifier can make class forecasts on new samples with unknown class labels. For example, a medical officer can use medical predicting system to predict if a patient have drug allergy or not. A dataset with given class can be used to be a training dataset, and a classifier must be trained by a training dataset to have the capability for class prediction. In brief, the process of classification analysis is included in the follow steps:

1. Sample collection.
2. Select samples and attributes for training.
3. Train a class predicting system using training samples.
4. Use the predicting system to forecast the class of incoming samples.

The classification techniques usually assume that the training samples are uniformly-distributed between different classes. A classifier performs well when the classification technique is applied to a dataset evenly distributed among different

D.-S. Huang, K. Li, and G.W. Irwin (Eds.): ICIC 2006, LNCIS 344, pp. 731 – 740, 2006.

classes. However, many datasets in real applications involve imbalanced class distribution problem [9, 11]. The imbalanced class distribution problem occurs while there are much more samples in one class than the other class in a training dataset. In an imbalanced dataset, the *majority class* has a large percent of all the samples, while the samples in *minority class* just occupy a small part of all the samples. In this case, a classifier usually tends to predict that samples have the majority class and completely ignore the minority class.

Many applications such as fraud detection, intrusion prevention, risk management, medical research often have the imbalanced class distribution problem. For example, a bank would like to construct a classifier to predict that whether the customers will have fiduciary loans in the future or not. The number of customers who have had fiduciary loans is only two percent of all customers. If a fiduciary loan classifier predicts that all the customers never have fiduciary loans, it will have a quite high accuracy as 98 percent. However, the classifier can not find the target people who will have fiduciary loans within all customers. Therefore, if a classifier can make correct prediction on the minority class efficiently, it will be useful to help corporations make a proper policy and save a lot of cost. In this paper, we study the effects of under-sampling [1, 6, 10] on the backpropagation neural network technique and propose some new under-sampling approaches based on clustering, such that the influence of imbalanced class distribution can be decreased and the accuracy of predicting the minority class can be increased.

2 Related Work

Since many real applications have the imbalanced class distribution problem, researchers have proposed several methods to solve this problem. As for re-sampling approach, it can be distinguished into *over-sampling approach* [4, 9] and *under-sampling approach* [10, 11]. The over-sampling approach increases the number of minority class samples to reduce the degree of imbalanced distribution. One of the famous over-sampling approaches is SMOTE [2]. SMOTE produces synthetic minority class samples by selecting some of the nearest minority neighbors of a minority sample which is named *S*, and generates new minority class samples along the lines between *S* and each nearest minority neighbor. SMOTE beats the random over-sampling approaches by its informed properties, and reduce the imbalanced class distribution without causing overfitting. However, SMOTE blindly generate synthetic minority class samples without considering majority class samples and may cause overgeneralization.

On the other hand, since there are much more samples of one class than the other class in the imbalanced class distribution problem, under-sampling approach is supposed to reduce the number of samples which have the majority class. Assume in a training dataset, MA is the sample set which has the majority class, and MI is the other set which has the minority class. Hence, an under-sampling approach is to decrease the skewed distribution of MA and MI by lowering the size of MA. Generally, the performances of under-sampling approaches are worse than that of under-sampling approaches.

One simple method of under-sampling is to select a subset of MA randomly and then combine them with MI as a training set, which is called *random under-sampling approach*. Several advanced researches are proposed to make the selective samples more representative. The under-sampling approach based on distance [11] uses distinct modes: the nearest, the farthest, the average nearest, and the average farthest distances between MI and MA, as four standards to select the representative samples from MA. For every minority class sample in the dataset, the first method "nearest" calculates the distances between all majority class samples and the minority class samples, and selects k majority class samples which have the smallest distances to the minority class sample. If there are n minority class samples in the dataset, the "nearest" approach would finally select $k \times n$ majority class samples ($k \geq 1$). However, some samples within the selected majority class samples might duplicate.

Similar to the "nearest" approach, the "farthest" approach selects the majority class samples which have the farthest distances to each minority class samples. For every majority class samples in the dataset, the third method "average nearest" calculates the average distance between one majority class sample and all minority class samples. This approach selects the majority class samples which have the smallest average distances. The last method "average farthest" is similar to the "average nearest" approach; it selects the majority class samples which have the farthest average distances with all the minority class samples. The above under-sampling approaches based on distance in [11] spend a lot of time selecting the majority class samples in the large dataset, and they are not efficient in real applications.

In 2003, J. Zhang and I. Mani [10] presented the compared results within four informed under-sampling approaches and random under-sampling approach. The first method "*NearMiss-1*" selects the majority class samples which are close to some minority class samples. In this method, majority class samples are selected while their average distances to three closest minority class samples are the smallest. The second method "*NearMiss-2*" selects the majority class samples while their average distances to three farthest minority class samples are the smallest. The third method "*NearMiss-3*" take out a given number of the closest majority class samples for each minority class sample. Finally, the fourth method "*Most distant*" selects the majority class samples whose average distances to the three closest minority class samples are the largest. The final experimental results in [10] showed that the *NearMiss-2* approach and random under-sampling approach perform the best.

3 Our Approaches

In this section, we present our approach *SBC* (under-Sampling Based on Clustering) which focuses on the under-sampling approach and uses clustering techniques to solve the imbalanced class distribution problem. Our approach first clusters all the training samples into some clusters. The main idea is that there are different clusters in a dataset, and each cluster seems to have distinct characteristics. If a cluster has more majority class samples and less minority class samples, it will behave like the majority class samples. On the opposite, if a cluster has more minority class samples and less majority class samples, it doesn't hold the characteristics of the majority class samples and behaves more like the minority class samples. Therefore, our

approach *SBC* selects a suitable number of majority class samples from each cluster by considering the ratio of the number of majority class samples to the number of minority class samples in the cluster.

3.1 Under-Sampling Based on Clustering

Assume that the number of samples in the class-imbalanced dataset is *N*, which includes majority class samples (MA) and minority class samples (MI). The size of the dataset is the number of the samples in this dataset. The size of MA is represented as $Size_{MA}$, and $Size_{MI}$ is the number of samples in MI. In the class-imbalanced dataset, $Size_{MA}$ is far larger than $Size_{MI}$. For our under-sampling method *SBC*, we first cluster all samples in the dataset into *K* clusters. The number of majority class samples and the number of minority class samples in the *i*th cluster ($1 \leqq i \leqq K$) are $Size^i_{MA}$ and $Size^i_{MI}$, respectively. Therefore, the ratio of the number of majority class samples to the number of minority class samples in the *i*th cluster is $Size^i_{MA} / Size^i_{MI}$. If the ratio of $Size_{MA}$ to $Size_{MI}$ in the training dataset is set to be *m*:1, the number of selected majority class samples in the *i*th cluster is shown in expression (1):

$$SSize^i_{MA} = (m \times SizeMI) \times \frac{Size^i_{MA} \big/ Size^i_{MI}}{\sum_{i=1}^{K} Size^i_{MA} \big/ Size^i_{MI}} \tag{1}$$

In expression (1), $m \times SizeMI$ is the total number of selected majority class samples that we suppose to have in the final training dataset. $\sum_{i=1}^{K} Size^i_{MA} \big/ Size^i_{MI}$ is the total ratio of the number of majority class samples to the number of minority class samples in all clusters. expression (1) determines that more majority class samples would be selected in the cluster which behaves more like the majority class samples. In other words, $SSize^i_{MA}$ is larger while the *i*th cluster has more majority class samples and less minority class samples. After determining the number of majority class samples which are selected in the *i*th cluster, $1 \leqq i \leqq K$, by using expression (1), we randomly choose majority class samples in the *i*th cluster. The total number of selected majority class samples is $m \times Size_{MI}$ after merging all the selected majority class samples in each cluster. At last, we combine the whole minority class samples with the selected majority class samples to construct a new training dataset. Table 1 shows the steps for our under-sampling approach.

For example, assume that an imbalanced class distribution dataset has totally 1100 samples. The size of MA is 1000 and the size of MI is 100. In this example, we cluster this dataset into three clusters. Table 2 shows the number of majority class samples $Size^i_{MA}$, the number of minority class samples $Size^i_{MI}$, and the ratio of $Size^i_{MA}$ to $Size^i_{MI}$ for the *i*th cluster.

Table 1. The structure of the under-sampling based on clustering approach *SBC*

Step1.	Determine the ratio of $Size_{MA}$ to $Size_{MI}$ in the training dataset.
Step2.	Cluster all the samples in the dataset into some clusters.
Step3.	Determine the number of selected majority class samples in each cluster by using expression (1), and then randomly select the majority class samples in each cluster.
Step4.	Combine the selected majority class samples and all the minority class samples to obtain the training dataset.

Table 2. Cluster descriptions

Cluster ID	Number of majority class samples	Number of minority class samples	$Size^i_{MA} / Size^i_{MI}$
1	500	10	500/10=50
2	300	50	300/50=6
3	200	40	200/40=5

Assume that the ratio of $Size_{MA}$ to $Size_{MI}$ in the training data is set to be 1:1, in other words, there are 100 selected majority class samples and the whole 100 minority class samples in this training dataset. The number of selected majority class samples in each cluster can be calculated by expression (1). Table 3 shows thenumber of selected majority class samples in each cluster. We finally select the majority samples randomly from each cluster and combine them with the minority samples to form the new dataset.

Table 3. The number of selected majority class samples in each cluster

Cluster ID	The number of selected majority class samples
1	1×100× 50 / (50+6+5) =82
2	1×100× 6 / (50+6+5) = 10
3	1×100× 5 / (50+6+5)= 8

3.2 Under-Sampling Based on Clustering and Distances

In *SBC* method, all the samples are clustered into several clusters and the number of selected majority class samples is determined by expression (1). Finally, the majority class samples are randomly selected from each cluster. In this section, we propose other five under-sampling methods, which are based on *SBC* approach. The difference between the five proposed under-sampling methods and *SBC* method is the way to select the majority class samples from each cluster. For the five proposed methods, the majority class samples are selected according to the distances between the majority class samples and the minority class samples in each cluster. Hence, the distances

between samples will be computed. For a continuous attribute, the values of all samples for this attribute need to be normalized in order to avoid the effect of different scales for different attributes. For example, suppose A is a continuous attribute. In order to normalize the values of attribute A for all the samples, we first find the maximum value Max_A and the minimum value Min_A of A for all samples. To lie an attribute value a_i in between 0 to 1, a_i is normalized to $\frac{a_i - Min_A}{Max_A - Min_A}$. For a categorical or discrete attribute, the distance between two attribute values x_1 and x_2 is 0 (i.e. x_1-x_2=0) while x_1 is not equal to x_2, and the distance is 1 (i.e. x_1-x_2=1) while they are the same.

Assume that there are N attributes in a dataset and V_i^X represents the value of attribute A_i in sample X, for $1 \leqq i \leqq N$. The Euclidean distance between two samples X and Y is shown in expression (2).

$$\text{distance}(X,Y) = \sqrt{\sum_{i=1}^{N} (V_i^X - V_i^Y)^2} \tag{2}$$

The five approaches we proposed in this section first cluster all samples into K ($K \geq 1$) clusters as well, and determine the number of selected majority class samples for each cluster by expression (1). For each cluster, the representative majority class samples are selected in different ways. The first method *SBCNM-1* (Sampling Based on Clustering with *NearMisss-1*) selects the majority class samples whose average distances to M nearest minority class samples ($M \geq 1$) in the ith cluster ($1 \leqq i \leqq K$) are the smallest. In the second method *SBCNM-2* (Sampling Based on Clustering with *NearMisss-2*), the majority class samples whose average distances to M farthest minority class samples in the ith cluster are the smallest will be selected.

The third method *SBCNM-3* (Sampling Based on Clustering with *NearMisss-3*) selects the majority class samples whose average distances to the closest minority class samples in the ith cluster are the smallest. In the forth method *SBCMD* (Sampling Based on Clustering with *Most Distant*), the majority class samples whose average distances to M closest minority class samples in the ith cluster are the farthest will be selected. For the above four approaches, we refer to [10] for selecting the representative samples in each cluster. The last proposed method, which is called *SBCMF* (Sampling Based on Clustering with Most Far), selects the majority class samples whose average distances to all minority class samples in the cluster are the farthest.

4 Experimental Results

For our experiments, we use three criteria to evaluate the classification accuracy for minority class: the precision rate P, the recall rate R, and the F-measure for minority class. Generally, for a classifier, if the precision rate is high, then the recall rate will be low, that is, the two criteria are trade-off. We cannot use one of the two criteria

to evaluate the performance of a classifier. Hence, the precision rate and recall rate are combined to form another criterion F-measure, which is shown in expression (3).

$$\text{MI's F-measure} = \frac{2 \times P \times R}{P + R} \tag{3}$$

In the following, we use the three criteria discussed above to evaluate the performance of our approaches *SBC*, *SBCNM-1*, *SBCNM-2*, *SBCNM-3*, *SBCMD*, and *SBCMF* by comparing our methods with the other methods *AT*, *RT*, and *NearMiss-2*. The method *AT* uses all samples to train the classifiers and does not select samples. *RT* is the most common-used random under-sampling approach and it selects the majority class samples randomly. The last method *NearMiss-2* is proposed by J. Zhang and I. Mani [10], which has been discussed in section 2. The two methods *RT* and *NearMiss-2* have the better performance than the other proposed methods in [10]. In the following experiments, the classifiers are constructed by using the artificial neural network technique in *IBM Intelligent Miner for Data V8.1*.

Table 4. The experimental results on *Census-Income Database*

Method	MI's Precision	MI's Recall	MI's F-measure	MA's Precision	MA's Recall	MA's F-measure
SBC	47.78	88.88	62.15	94.84	67.79	79.06
RT	30.29	99.73	46.47	99.63	23.92	38.58
AT	35.1	98.7	51.9	98.9	39.5	43.8
NearMiss-2	46.3	81.23	58.98	91.70	68.77	78.60
SBCNM-1	29.28	99.80	45.28	99.67	20.07	33.41
SBCNM-2	29.6	99.67	45.64	99.49	21.39	35.21
SBCNM-3	28.72	99.8	44.61	99.63	17.9	30.35
SBCMD	29.01	99.73	44.94	99.54	19.05	31.99
SBCMF	43.15	93.48	59.04	96.47	59.15	73.34

We compare our approaches with the other under-sampling approaches in two real datasets. One of the real datasets is named *Census-Income Database*, which is from *UCI Knowledge Discovery in Databases Archive*. *Census-Income Database* contains census data which are extracted from the 1994 and 1995 current population surveys managed by the U.S. Census Bureau. The binary classification problem in this dataset is to determine the income level for each person represented by the record. The total number of samples after cleaning the incomplete data is 30162, including 22654 majority class samples which the income level are less than 50K dollars and 7508 minority class samples which the income level are greater than or equal to 50K dollars. We use eighty percent of the samples to train the classifiers and twenty percent to evaluate the performances of the classifiers. The precision rate, recall rate, and F-measure for our approaches and the other approaches are shown in Table 4. Fig 1 shows

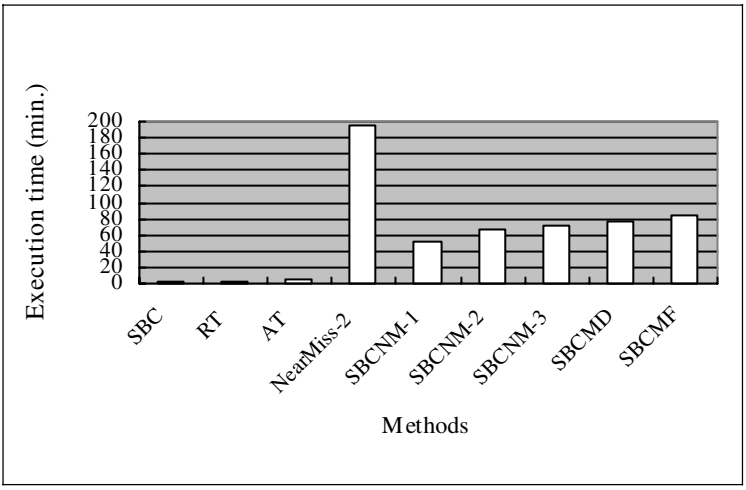

Fig. 1. The execution time on *Census-Income Database* for each method

the execution time for each method, which includes selecting the training data and training the classifier. In Table 4, we can observe that our method *SBC* has the highest MI's F-measure and MA's F-measure while comparing with other methods. Besides, *SBC* only need to take a short execution time which is shown in Fig 1.

The other real dataset in our experiment is conducted by a bank and is called *Overdue Detection Database*. The records in *Overdue Detection Database* contain the information of customers, the statuses of customers' payment, the amount of money in customers' bills, and so on. The purpose of this binary classification problem is to detect the bad customers. The bad customers are the minorities within all customers and they do not pay their bills before the deadline. We separate *Overdue Detection Database* into two subsets. The dataset extracted from November in 2004 are used for training the classifier and the dataset extracted from December in 2004 are used for testing task. The total number of samples in the training data of *Overdue Detection Database* is 62309, including 47707 majority class samples which represent the good customers and 14602 minority class samples which represent the bad customers. The total number of samples in the testing data of *Overdue Detection Database* is 63532, including 49931 majority class samples and 13601 minority class samples. Fig 2 shows the precision rate, the recall rate and the F-measure of minority class for each approach. From Fig 2, we can see that our approaches *SBC* and *SBCMD* have the best MI's F-measure. Fig 3 shows the execution times for all the approaches in *Overdue Detection Database*.

In the two real applications which involve the imbalanced class distribution problem, our approach *SBC* has the best performances on predicting the minority class samples. Moreover, *SBC* takes less time for selecting the training samples than the other approaches *NearMiss-2*, *SBCNM-1*, *SBCNM-2*, *SBCNM-3*, *SBCMD*, and *SBCMF*.

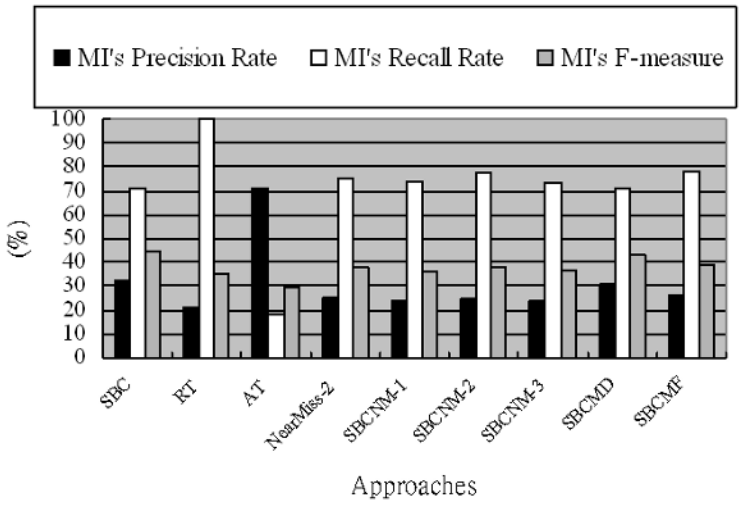

Fig. 2. The Experimental Results on *Overdue Detection Database*

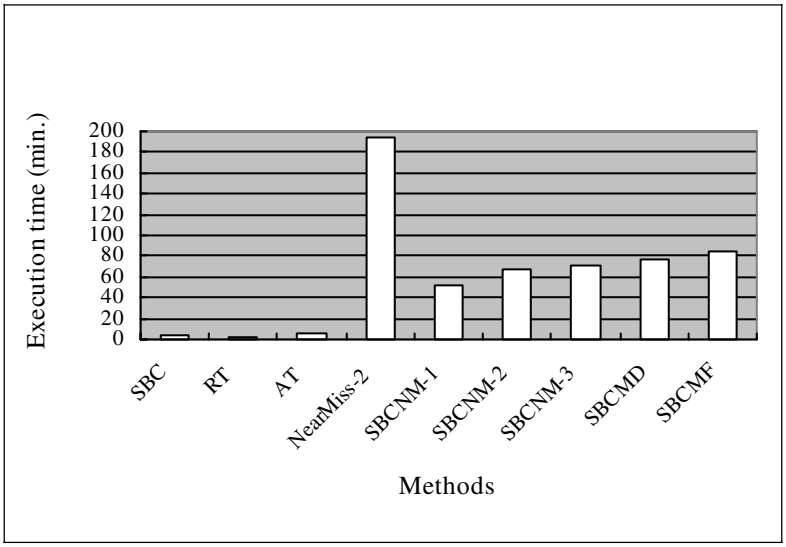

Fig. 3. Execution time on *Overdue Detection Database* for each method

5 Conclusion

In a classification task, the effect of imbalanced class distribution problem is often ignored. Many studies [3, 7] focused on improving the classification accuracy but did not consider the imbalanced class distribution problem. Hence, the classifiers which are constructed by these studies lose the ability to correctly predict the correct deci-

sion class for the minority class samples in the datasets which the number of majority class samples are much greater than the number of minority class samples. Many real applications, like rarely-seen disease investigation, credit card fraud detection, and internet intrusion detection always involve the imbalanced class distribution problem. It is hard to make right predictions on the customers or patients who that we are interested in.

In this study, we propose cluster-based under-sampling approaches to solve the imbalanced class distribution problem by using backpropagation neural network. The other two under-sampling methods, Random selection and *NearMiss-2*, are used to be compared with our approaches in our performance studies. In the experiments, our approach *SBC* has better prediction accuracy and stability than other methods. *SBC* not only has high classification accuracy on predicting the minority class samples but also has fast execution time. However, *SBCNM-1*, *SBCNM-2*, *SBCNM-3*, and *SBCMF* do not have stable performances in our experiments. The four methods take more time than *SBC* on selecting the majority class samples as well.

References

1. Chawla, N. V.: C4.5 and Imbalanced Datasets: Investigating the Effect of Sampling Method, Probabilistic Estimate, and Decision Tree Structure. Proceedings of the ICML'03 Workshop on Class Imbalances, (2003)
2. Chawla, N. V., Bowyer, K.W., Hall, L. O., Kegelmeyer, W. P.: SMOTE: Synthetic Minority Over-Sampling Technique. Journal of Artificial Intelligence Research, 16 (2002) 321–357
3. Caragea, D., Cook, D., Honavar, V.: Gaining Insights into Support Vector Machine Pattern Classifiers Using Projection-Based Tour Methods. Proceedings of the KDD Conference, San Francisco, CA (2001) 251-256
4. Chawla, N. V., Lazarevic, A., Hall, L. O., Bowyer, K. W.: Smoteboost: Improving Prediction of the Minority Class in Boosting. Proceedings of the Seventh European Conference on Principles and Practice of Knowledge Discovery in Databases, Dubrovnik, Croatia (2003) 107-119
5. Clark, P., Niblett, T.: The CN2 Induction Algorithm. Machine Learning, 3 (1989) 261-283
6. Drummond, C., Holte, R. C.: C4.5, Class Imbalance, and Cost Sensitivity: Why Under-Sampling Beats Over-Sampling. Proceedings of the ICML'03 Workshop on Learning from Imbalanced Datasets, (2003)
7. Del-Hoyo, R., Buldain, D., Marco, A.: Supervised Classification with Associative SOM. Lecture Notes in Computer Science, 2686 (2003) 334-341
8. Japkowicz, N.: Concept-learning in the Presence of Between-class and Within-class Imbalances. Proceedings of the Fourteenth Conference of the Canadian Society for Computational Studies of Intelligence, (2001) 67-77
9. Zhang, J., Mani, I.: KNN Approach to Unbalanced Data Distributions: A Case Study Involving Information Extraction. Proceedings of the ICML'2003 Workshop on Learning from Imbalanced Datasets, (2003).
10. Chyi, Y. M.: Classification Analysis Techniques for Skewed Class Distribution Problems. Master Thesis, Department of Information Management, National Sun Yat-Sen University, (2003)