Expert Systems with Applications 36 (2009) 5718–5727

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Cluster-based under-sampling approaches for imbalanced data distributions

Show-Jane Yen [*], Yue-Shi Lee

Department of Computer Science and Information Engineering, Ming Chuan University, 5 The-Ming Road, Gwei Shan District, Taoyuan County 333, Taiwan

ARTICLE INFO

Keywords:
Classification
Data mining
Under-sampling
Imbalanced data distribution

ABSTRACT

For classification problem, the training data will significantly influence the classification accuracy. 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 belongs 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 and investigate the effect of under-sampling methods in the imbalanced class distribution environment. The experimental results show that our cluster-based under-sampling approaches outperform the other under-sampling techniques in the previous studies.

© 2008 Elsevier Ltd. All rights reserved.

1. Introduction

Classification analysis (del-Hoyo, Buldain, & Marco, 2003; Lee & Chen, 2005; Li, Ying, Tuo, Li, & Liu, 2004) 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 with 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 following 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 classes. However, many datasets in real applications involve imbalanced class distribution problem (Chawla, 2003; Chyi, 2003; Japkowicz, 2000, 2001; Jo & Japkowicz, 2004; Maloof, 2003; Zhang & Mani, 2003). 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 percentage for 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 2% 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%. 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 (Zhang & Mani, 2003) on the neural network technique and propose some new under-sampling methods 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.

[*] Corresponding author. Tel.: +886 3 3507001; fax: +886 3 3593874.
E-mail address: sjyen@mail.mcu.edu.tw (S.-J. Yen).

0957-4174/$ - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2008.06.108

2. Related work

Since many real applications have the imbalanced class distribution problem, researchers have proposed several methods to solve this problem. These methods try to solve the class distribution problem both at the algorithmic level and data level. At the algorithmic level, developed methods include cost-sensitive learning (Drummond & Holte, 2003; Elkan, 2001; Turney, 2000) and recognition-based learning (Chawla, Bowyer, Hall, & Kegelmeyer, 2002; Manevitz & Yousef, 2001).

Cost-sensitive learning approach assumes the misclassification costs are known in a classification problem. A cost-sensitive classifier tries to learn more characteristics of samples with the minority class by setting a high cost to the misclassification of a minority class sample. However, misclassification costs are often unknown and a cost-sensitive classifier may result in overfitting training. To ensure learning the characteristics of whole samples with the minority class, the recognition-based learning approach attempts to overfit by one-class (minority class) learning. One-class learning is more suitable than two-class approaches under certain conditions such like very imbalanced data and high dimensional noisy feature space (Elkan, 2001).

At the data level, methods include multi-classifier committee (Argamon-Engelson & Dagan, 1999; Freund, Sebastian Seung, Shamir, & Tishby, 1997) and re-sampling (Chawla, Lazarevic, Hall, & Bowyer, 2003; Chawla et al., 2002; Chyi, 2003; Drummond & Holte, 2003; Japkowicz, 2001; Zhang & Mani, 2003) approaches. Multi-classifier committee approach (Argamon-Engelson & Dagan, 1999; Freund et al., 1997) makes use of all information on a training dataset. Assume in a training dataset, MA is the sample set with majority class, and MI is the other set with minority class. Multi-classifier committee approach divides the samples with majority class (i.e. MA) randomly into several subsets, and then takes every subset and all the samples with minority class (i.e. MI) as training dataset, respectively. The number of the subsets depends on the ratio of MA's size to MI's size. For example, suppose in a dataset, the size of MA is 48 (samples) and the size of MI is 2 (samples). If we think the best ratio of MA's size to MI's size is 1:1 in a training dataset, then the number of training subsets will be 48/2 = 24. Each of these 24 subsets contains MI and a subset of MA that both sizes are 2, and the ratio of them is exactly 1:1.

After training these training datasets separately, several classifiers are available as committees. Multi-classifier committee approach uses all the classifiers to predict a sample and decides the final class to it by the prediction results of the classifiers. Voting is one simple method for making a final class decision to a sample, in which a minimum threshold is set up. If the number of classifiers that predict the same class "C" for a sample exceeds the minimum threshold, then the final class prediction of this sample will be "C". Though multi-classifier committee approach does not abandon any sample from MA, it may be inefficient in the training time for all the committees and can not ensure the quality for every committee. Further selection of the committees will make the predictions more correct and more efficient.

As for re-sampling approach, it can be distinguished into *over-sampling approach* (Chawla et al., 2002, 2003; Japkowicz, 2001) and *under-sampling approach* (Chyi, 2003; Zhang & Mani, 2003). 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 (Chawla et al., 2002). 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 with the majority class. Assume in a training dataset, MA is the sample set with 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 over-sampling approaches are worse than that of under-sampling approaches (Drummond & Holte, 2003).

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 (Chyi, 2003) 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" method would finally select $k \times n$ majority class samples ($k \geqslant 1$). However, some samples within the selected majority class samples might duplicate.

Similar to the "nearest" method, the "farthest" method 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 distances between one majority class sample and all minority class samples. This method selects the majority class samples which have the smallest average distances. The last method "average farthest" is similar to the "average nearest" method; 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 Chyi (2003) spend a lot of time selecting the majority class samples in the large dataset, and they are not efficient in real applications.

Zhang and Mani (2003) 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 Zhang and Mani (2003) showed that the *NearMiss-2* method and random under-sampling method perform the best.

3. Our approaches

In this section, we present our cluster-based under-sampling approach. 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 other hand, 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 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* (under-sampling based on clustering), we first cluster all samples in the dataset into K clusters. In the experiments, we will study the performances for the under-sampling methods on different number of clusters.

Let the number of majority class samples and the number of minority class samples in the ith cluster ($1 \leqq i \leqq K$) be $\text{Size}^i_{\text{MA}}$ and $\text{Size}^i_{\text{MI}}$, respectively. Therefore, the ratio of the number of majority class samples to the number of minority class samples in the ith cluster is $\text{Size}^i_{\text{MA}}/\text{Size}^i_{\text{MI}}$. Suppose the ratio of Size_{MA} to Size_{MI} in the training dataset is set to be m:1 ($m \geqq 1$). The number of selected majority class samples in the ith cluster is shown in expression (1):

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

In expression (1), $m \times \text{Size}_{\text{MI}}$ is the total number of selected majority class samples that we suppose to have in the final training dataset. $\sum_{i=1}^{K} \text{Size}^i_{\text{MI}}/\text{Size}^i_{\text{MI}}$ 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, $\text{SSize}^i_{\text{MA}}$ is larger while the ith cluster has more majority class samples and less minority class samples.

If there is no minority class samples in the ith cluster, then the number of minority class samples in the ith cluster (i.e., $\text{Size}^i_{\text{MI}}$) is regarded as one, that is, we assume that there is at least one minority class sample in a cluster. After determining the number of majority class samples which are selected in the ith cluster ($1 \leqq i \leqq K$) by using expression (1), we randomly choose majority class samples in the ith cluster. The total number of selected majority class samples is about $m \times \text{Size}_{\text{MI}}$ after merging all the selected majority class samples in each cluster. Finally, we combine the whole minority class samples with the selected majority class samples to construct a new training dataset. The ratio of Size_{MA} to Size_{MI} is about m:1 in the new training dataset. Table 1 shows the steps for our cluster-based under-sampling method *SBC*.

Table 1
The structure of *SBC*

Step 1	Determine the ratio of Size_{MA} to Size_{MI} in the training dataset
Step 2	Cluster all the samples in the dataset into some clusters
Step 3	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
Step 4	Combine the selected majority class samples and all the minority class samples to obtain the training dataset

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 $\text{Size}^i_{\text{MA}}$, the number of minority class samples $\text{Size}^i_{\text{MI}}$, and the ratio of $\text{Size}^i_{\text{MA}}$ to $\text{Size}^i_{\text{MI}}$ for the ith cluster.

Table 2
Cluster descriptions

Cluster ID	Number of majority class samples	Number of minority class samples	$\text{Size}^i_{\text{MA}}/\text{Size}^i_{\text{MI}}$
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 about 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 the number of selected majority class samples in each cluster. We finally select the majority class samples randomly from each cluster and combine them with the minority class 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	$1 \times 100 \times 50/(50+6+5) = 82$
2	$1 \times 100 \times 6/(50+6+5) = 10$
3	$1 \times 100 \times 5/(50+6+5) = 8$

3.2. Under-sampling based on clustering and distances between samples

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 and 1, a_i is normalized to $\frac{a_i - \text{Min}_A}{\text{Max}_A - \text{Min}_A}$. For a categorical or discrete attribute, the distance between two attribute values x_1 and x_2 is 1 (i.e. $x_1 - x_2 = 1$) while x_1 is not equal to x_2, and the distance is 0 (i.e. $x_1 - x_2 = 0$) while they are the same.

Assume that there are N attributes in a dataset and V^X_i 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):

$$\text{Distance}(X, Y) = \sqrt{\sum_{i=1}^{N} (V^X_i - V^Y_i)^2} \tag{2}$$

The five approaches we proposed in this section first cluster all samples into K ($K \geqslant 1$) 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 ($M \geqslant 1$) 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 Distance*), 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 (Zhang & Mani, 2003) 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

In this section, we evaluate the performances for our proposed under-sampling approaches on synthetic datasets and real datasets. In the following, we first describe the method of generating class imbalanced datasets. And then we compare the classification accuracies of our methods for minority class with the other methods by performing neural network classification algorithm (Sondak & Sondak, 1989) on synthetic datasets. Finally, the classification accuracies for minority class on real datasets by applying our proposed methods and the other methods are also evaluated.

4.1. Generation of synthetic datasets

In this subsection, we present the synthetic dataset generation method to simulate the real-world dataset. This method is implemented with a user interface such that the parameters can be set for generating the synthetic dataset from the user interface, which is called synthetic dataset generator.

A synthetic dataset includes a set of attributes and each sample in the dataset has a set of particular attribute values. In real-world, the samples in the same class should have similar attribute values and the samples in different class should have different characteristics. Even though the samples in the same class, these samples may have different characteristics and can be clustered into some clusters. The samples in a cluster may have the similar attribute values and may belong to different classes. Besides, there may be some noises or exceptions in a dataset, that is, some samples in one class may have the similar attribute values with the samples in the other class or may be not similar to any other samples with the same class. According to the above observations, the following parameters need to be set for generating the synthetic dataset: number of samples, number of attributes and number of clusters.

Because the samples in a cluster may belong to different classes, in a cluster, the samples are separated into two groups: the samples in one group are assigned a class and the samples in the other group are assigned to the other class. The attribute values for the samples are more similar to the samples in the same group, because they are in the same cluster and the same class. Fig. 1 shows the distribution of samples in a dataset which has three clusters inside.

In order to make the synthetic datasets more like real datasets, the noisy data are necessary. The synthetic datasets have two kinds of noisy data: disordered samples and exceptional samples. A dataset which does not have any noisy data is like the one in Fig. 1. The disordered samples are illustrated with Fig. 2 in which some majority class samples (or minority class samples) lie to the area of minority class samples (or majority class samples). As for exceptional samples, they distribute irregularly in a dataset. The samples outside the clusters in Fig. 3 are exceptional samples.

4.2. Evaluation criteria

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. The precision rate for minority class is the correct-classified percentage of samples which are predicted as minority class by the classifier. The recall rate for minority class is the correct-classified percentage of all the minority class samples. 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

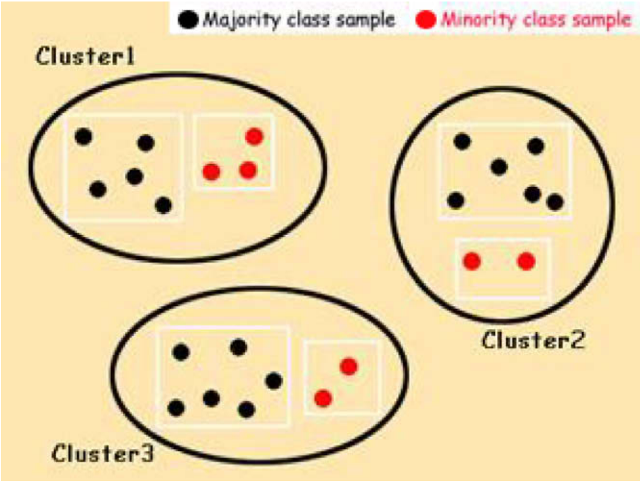

Fig. 1. The distribution of samples in a dataset.

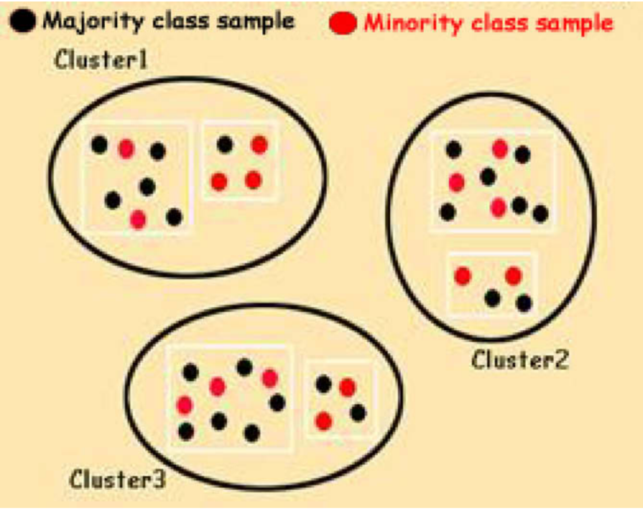

Fig. 2. Example for disordered samples.

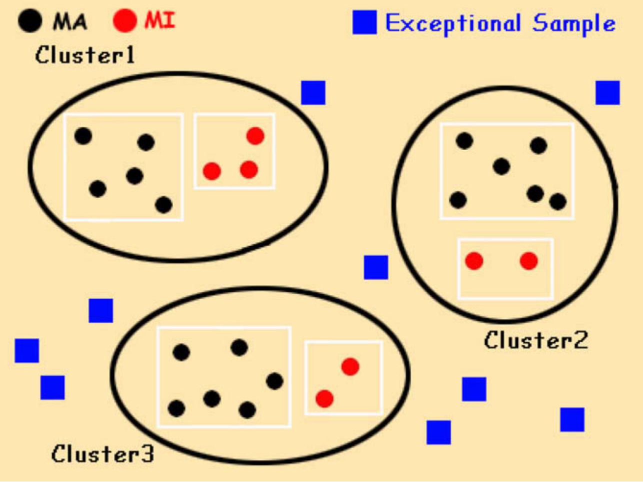

Fig. 3. Example for exceptional samples.

recall rate are combined to form another criterion *F*-measure, which is shown in expression (3).

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

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 Zhang & Mani, 2003, which has been discussed in Section 2. The two methods *RT* and *NearMiss-2* have the better performance than the other proposed methods in Zhang and Mani (2003). In the following experiments, the classifiers are constructed by using the artificial neural network technique in *IBM Intelligent Miner for Data V8.1*, and the *k*-means

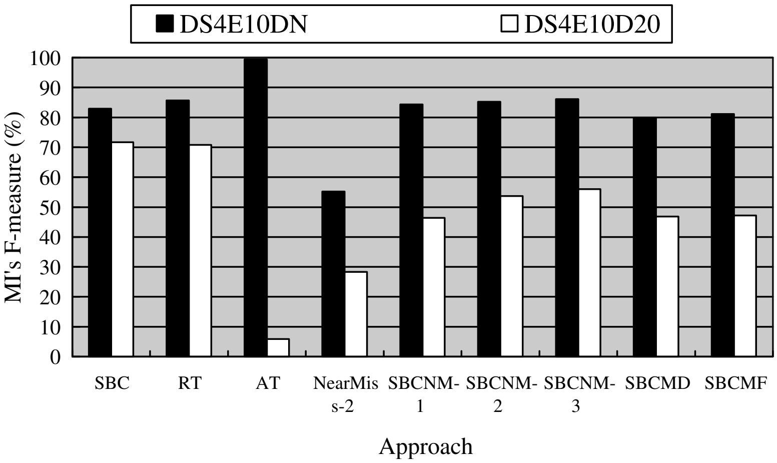

Fig. 4. The effect of disordered samples.

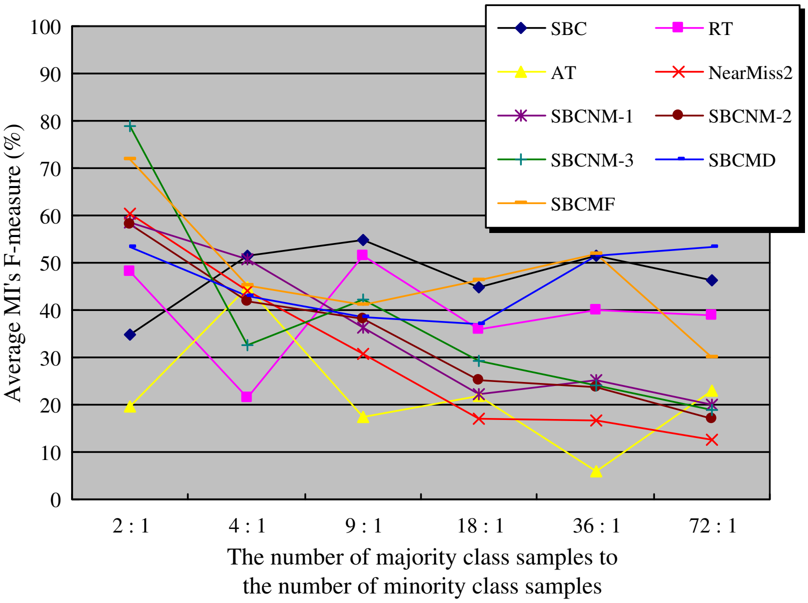

Fig. 5. Average MI's *F*-measure for datasets DSiE10D20.

clustering algorithm is used for our methods. In our experiments, the clustering algorithms would not influence the performances of our methods.

4.3. Experimental results on synthetic datasets

For each generated synthetic dataset, the number of samples is set to 10,000, the number of numerical attributes and categorical attributes are set to 5, respectively. The dataset DSi means that the dataset potentially can be separated into *i* clusters, and our methods also cluster the dataset DSi into *i* clusters. Moreover, a dataset DSi with *j*% exceptional samples and *k*% disordered samples is represented as DSiEjDk. If there is no disordered sample in the synthetic dataset, the dataset is represented as DSiEjDN.

Fig. 4 shows the MI's *F*-measures for our method and the other methods on datasets DS4E10DN and DS4E10D20. The ratio of the number of majority class samples to the number of minority class samples is 9–1 in the two datasets for this experiment. In Fig. 4, the method *AT* has the highest MI's *F*-measure in DS4E10DN because *AT* puts all the samples in the dataset into training and there is no disordered samples and just few exceptional samples in the dataset. The data distribution and characteristics can be completely represented from all the samples if there is no noise in the dataset. Hence, the classifier on DS4E10DN has the best classification accuracy when the method *AT* is applied. However, the method *AT* has to put all the samples into training, which is very time-consuming. Our method *SBC* and *RT* just need to put 20% of all samples into training since the ratio of Size_{MA} to Size_{MI} is set to be 1:1, and the MI's *F*-measures are above 80%. The method *AT* on dataset DS4E10D20 becomes worst and the classification accuracy is below 10%, because the dataset includes some noises, that is, 10% exceptional samples and 20% disordered samples for all the samples and all the noises are put into training. The classification accuracy for our method *SBC* and *RT* are significantly better

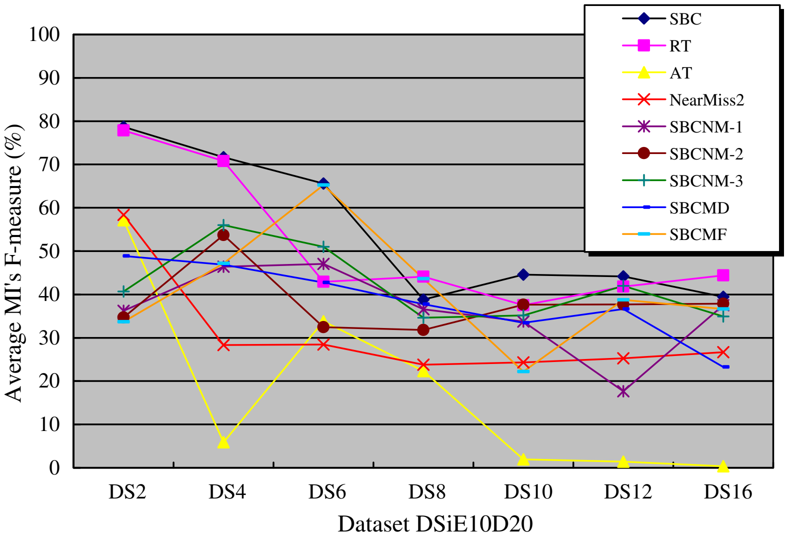

Fig. 6. MI's *F*-measure for each method on the datasets with 10% exceptional samples and 20% disordered samples.

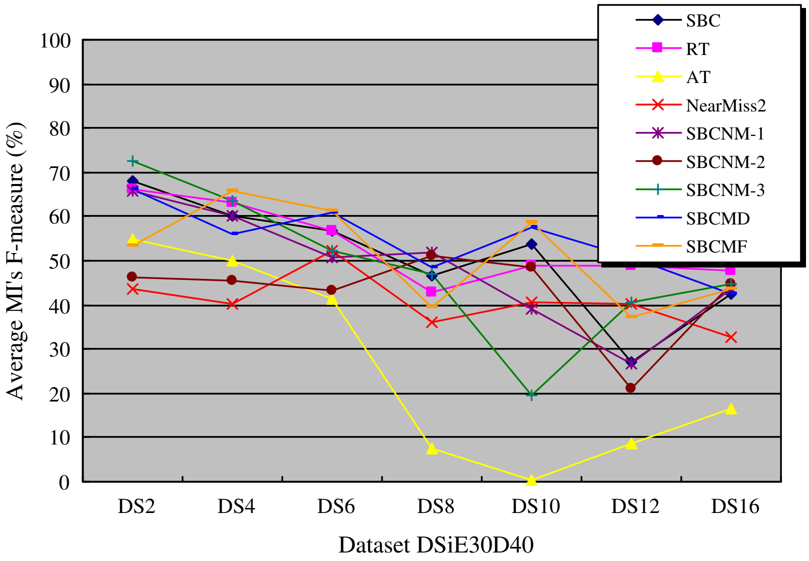

Fig. 7. MI's *F*-measure for each method on the datasets with 30% exceptional samples and 40% disordered samples.

than *AT*, since some noises can be ignored by applying *SBC* and *RT*. In this experiment, the performances of classification by using *SBC* and *RT* are better than the other methods.

Fig. 5 shows the experimental results in the datasets in which the ratios of the number of majority class samples to the number of minority class samples are 2:1, 4:1, 9:1, 18:1, 36:1, and 72:1, respectively. For each specific ratio, we generate several synthetic datasets DSiE10D20 in which *i* is from 2 to 16. Hence, the average MI's *F*-measures are computed from all the datasets for each specific ratio. In Fig. 5, we can see that the average MI's *F*-measure for *SBC* is higher than the other methods in most cases. Fig. 6 shows the performances of our approaches and other approaches on datasets DSiE10D20, in which *i* is from 2 to 16. In these synthetic datasets, the ratio of the number of majority class samples to the number of minority class samples is 9–1. In Fig. 6, we can see that the average MI's *F*-measure for *SBC* and *RT* are better than the other methods and our method *SBC* outperforms *RT* in most cases.

We raise the percentage of exceptional samples and disordered samples to 30% and 40%, respectively. And then we continue to raise the percentage of exceptional samples and disordered samples to 50% and 60%, respectively. Figs. 7 and 8 show the experimental results in DSiE30D40 and DSiE50D60, respectively, in which *i* is from 2 to 16. The experimental results show that *SBCMD* is the most stable method and has high MI's *F*-measure in each synthetic dataset. *RT* is also a stable method in the experiments, but the performance for *SBCMD* is better than *RT* in most cases.

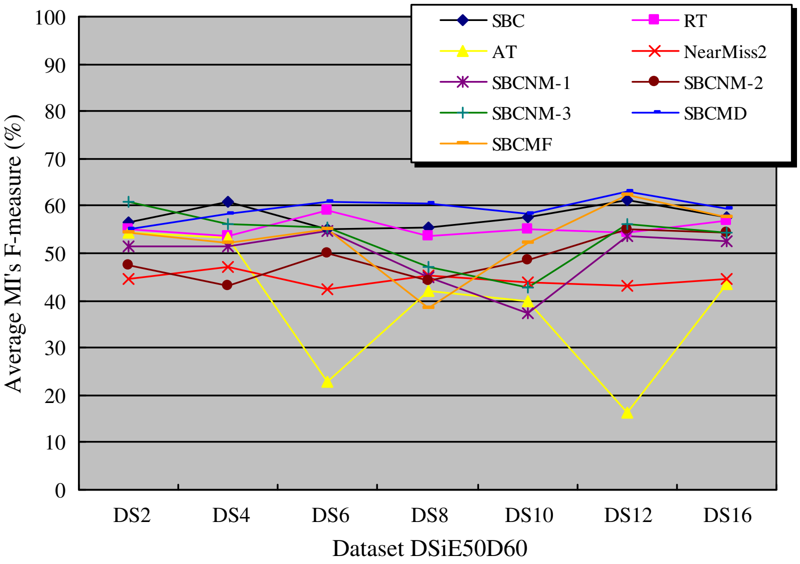

Fig. 8. MI's *F*-measure for each method on the datasets with 50% exceptional samples and 60% disordered samples.

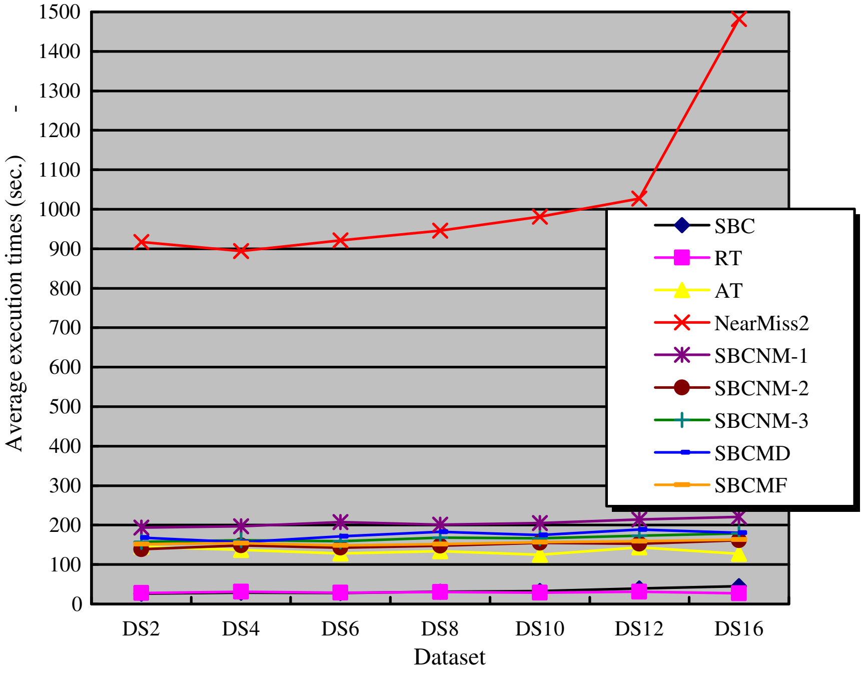

Fig. 9. Average execution time for each method.

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

Although the MI's *F*-measure for *SBCMF* is higher than the other methods in some cases, the performance for *SBCMF* is not stable. Hence, the performance for *SBCMD* is the best in most of the cases when the datasets contain more exceptional samples and disordered samples, and *SBC* is stable and performs well in any case.

The average execution time for each method is shown in Fig. 9. The execution time includes the time for executing the under-sampling method and the time for training the classifiers. According to the results in Fig. 9, both *SBC* and *RT* are most efficient among all the methods, and *NearMiss-2* spends too much time for selecting the majority class samples.

4.4. Experimental results on real datasets

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 US 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 30,162, including 22,654 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 80% of the samples to train the classifiers and 20% to evaluate the performances of the classifiers. The precision rates, recall rates, and *F*-measures for our approaches and the other approaches are shown in Table 4. Fig. 10 shows 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 needs to take a short execution time which is shown in Fig. 10.

The other real dataset in our experiment is conducted by a bank and is called *Overdue Detection Database*. The records in *Overdue*

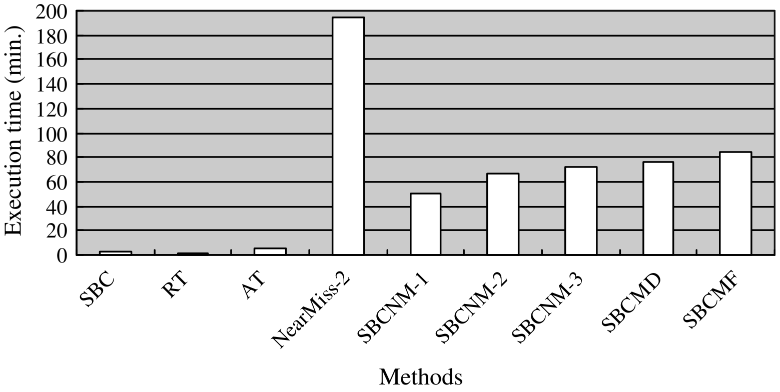

Fig. 10. The execution times on *Census-Income Database* for each method.

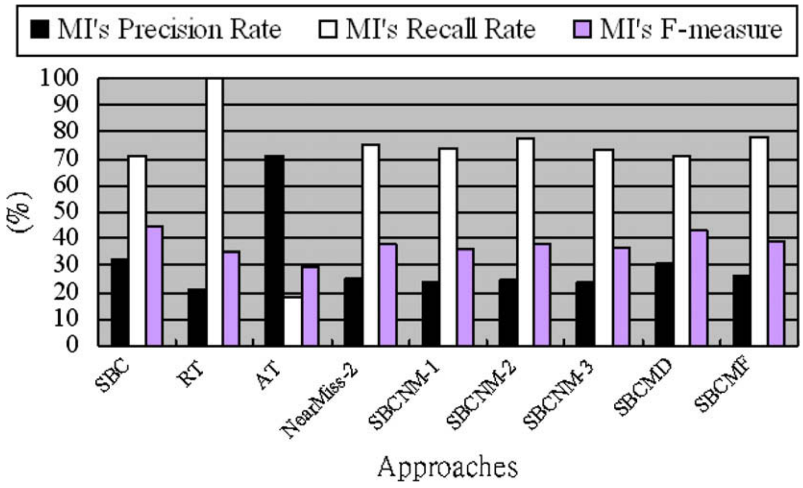

Fig. 11. The experimental results on *Overdue Detection Database*.

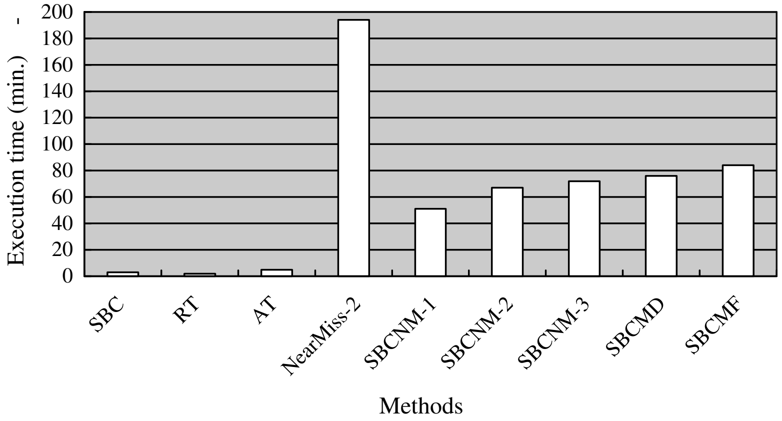

Fig. 12. Execution times on *Overdue Detection Database* for each method.

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 62,309, including 47,707 majority class samples which represent the good customers and 14,602 minority class samples which represent the bad customers. The total number of samples in the testing data of *Overdue Detection Database* is 63,532, including 49,931 majority class samples and 13,601 minority class samples. Fig. 11 shows the precision rate, recall rate and *F*-measure of minority class for each approach. From Fig. 11, we can see that our approaches *SBC* and *SBCMD* have the best MI's *F*-measure. Fig. 12 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*.

5. Conclusions

In a classification task, the effect of imbalanced class distribution problem is often ignored. Many studies (Japkowicz, 2001; Lee & Chen, 2005; Li et al., 2004) 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 decision 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. Our another approach *SBCMD* has better prediction accuracy and stability when the datasets contain more exceptional samples and disordered samples. However, our other approaches *SBCNM-1*, *SBCNM-2*, *SBCNM-3*, and *SBCMF* do not have stable performances in our experiments. The five methods take more time than *SBC* on selecting the majority class samples as well.

References

Argamon-Engelson, S., & Dagan, I. (1999). Committee-based sample selection for probabilistic classifiers. *Journal of Artificial Intelligence Research (JAIR), 11*, 335–360.

Chawla, N. V. (2003). C4.5 and imbalanced datasets: Investigating the effect of sampling method, probabilistic estimate, and decision tree structure. In *Proceedings of the ICML'03 workshop on class imbalances, August*.

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research, 16*, 321–357.

Chawla, N. V., Lazarevic, A., Hall, L. O., & Bowyer, K. W. (2003). Smoteboost: Improving prediction of the minority class in boosting. In *Proceedings of the 7th European conference on principles and practice of knowledge discovery in databases, Dubrovnik, Croatia* (pp. 107–119).

Chyi, Y.-M. (2003). Classification analysis techniques for skewed class distribution problems. Master thesis, Department of Information Management, National Sun Yat-Sen University.

del-Hoyo, R., Buldain, D., & Marco, A. (2003). Supervised classification with associative SOM. In *Seventh international work-conference on artificial and natural neural networks, IWANN 2003. Lecture notes in computer science* (Vol. 2686, pp. 334–341).

Drummond, C., & Holte, R. C., (2003). C4.5, class imbalance, and cost sensitivity: Why under-sampling beats over-sampling. In *Proceedings of the ICML'03 workshop on learning from imbalanced datasets*.

Elkan, C. (2001). The foundations of cost-sensitive learning. In Proceedings of the 17th international joint conference on artificial intelligence (pp. 973–978).

Freund, Y., Sebastian Seung, H., Shamir, E., & Tishby, N. (1997). Selective sampling using the query by committee algorithm. *Machine Learning, 28*(2–3), 133–168.

Japkowicz, N. (Ed.). (2000). *Proceedings of the AAAI'2000 Workshop on Learning from Imbalanced Data Sets, AAAI Tech Report WS-00-05*. AAAI.

Japkowicz, N. (2001) Concept-learning in the presence of between-class and within-class imbalances. In *Proceedings of the 14th conference of the canadian society for computational studies of intelligence* (pp. 67–77).

Jo, T., & Japkowicz, N. (2004). Class imbalances versus small disjuncts. *SIGKDD Explorations, 6*(1), 40–49.

Lee, T. S., & Chen, I. F. (2005). A two-stage hybrid credit scoring model using artificial neural networks and multivariate adaptive regression splines. *Expert Systems with Applications, 28*(4), 743–752.

Li, X., Ying, W., Tuo, J., Li, B., & Liu, W. (2004). Applications of classification trees to consumer credit scoring methods in commercial banks. *IEEE International Conference on Systems, Man and Cybernetics, 5*, 4112–4117.

Maloof, M. (2003). Learning when data sets are imbalanced and when costs are unequal and unknown. In *Proceedings of the ICML'03 workshop on learning from imbalanced data sets*.

Manevitz, L., & Yousef, M. M. (2001). One-class SVMs for document classification. *Journal of Machine Learning Research, 2*, 139–154.

Sondak, N. E., & Sondak, V. K. (1989). Neural networks and artificial intelligence. In *Proceedings of the 20th SIGCSE technical symposium on Computer science education*.

Turney, P. (2000). Types of cost in inductive concept learning. In *Proceedings of the ICML'2000 workshop on cost-sensitive learning* (pp. 15–21).

Zhang, J., & Mani, I. (2003). kNN approach to unbalanced data distributions: A case study involving information extraction. In *Proceedings of the ICML'2003 workshop on learning from imbalanced datasets*.