J. Med. Biol. Eng. (2015) 35:824–832
DOI 10.1007/s40846-015-0093-9

ORIGINAL ARTICLE

Neural-Network-Based Resampling Method for Detecting Diabetes Mellitus

Long-Sheng Chen[1] · **Sheng-Jhe Cai**[1]

Received: 8 April 2015 / Accepted: 16 June 2015 / Published online: 13 November 2015

Abstract Diabetes has been the fifth leading cause of death in Taiwan since 1987, and the complications of this disease are a burden to patients, their families, and society. Recent studies have tried to build a classifier that can easily identify diabetes mellitus by employing data mining approaches. However, these studies have encountered a class imbalance problem caused by skewed data, in which almost all of the instances are labeled as one class (healthy) while only a few instances are labeled as the other class (diabetic). When learning from this type of data, machine learning algorithms tend to produce predictive results with a high level of accuracy for the majority class, but poor predictive accuracy for the minority class. This study proposes the neural-network-based resampling method, which dramatically improves the detection of diabetes. Real diabetes data from a regional hospital in Taiwan and several biological data sets are used to demonstrate the effectiveness of the proposed method.

Keywords Diabetes diagnosis · Resampling · Class imbalance problem · Classification · Support vector machines

1 Introduction

Diabetes mellitus, which can result in a variety of complications, including heart disease, kidney disease, eye disease, erectile dysfunction, and nerve damage, has become a serious problem in society [1]. Diabetes is the most common endocrine disease across all population and age groups. This disease has become one of the leading causes of death in developed countries [2]. According to a report of the World Health Organization (WHO) in 2014, the estimated global prevalence of diabetes was 9 % among adults aged 18 years old and older. About 1.5 million deaths were directly caused by this disease in 2012. More than 80 % of diabetes deaths occur in low- and middle-income countries. By 2030, diabetes will be the 7th leading cause of death in the world [3]. Diabetes, recently called an epidemic by the WHO, is having a huge economic impact in African countries, India, and China. Diabetes is a bigger killer than AIDS, and the cost of supporting a person who has lost a foot due to diabetes may drain three-quarters of the income of a poor family [1, 3].

Researchers have used artificial intelligence and data mining methods to build diagnostic classifiers [4] in order to identify diseases quickly and economically, helping medical experts diagnose patients in developing countries that lack sufficient medical resources. For example, Su et al. [5] utilized a data mining method to diagnose type 2 diabetes using three-dimensional body surface anthropometrical scanning data. Yildirim et al. [6] presented a data mining model that includes an adaptive-network-based fuzzy inference system and rough set methods to predict suitable dosage planning for diabetes patients. Meng et al. [7] compared three methods, namely those based on logistic regression, artificial neural networks, and decision trees, to predict diabetes or pre-diabetes. Aljumah et al. [2] employed an Oracle data miner to predict the modes of treating diabetes. Kang et al. [8] proposed an ensemble of support vector machines (SVMs) to predict anti-diabetic drug failure.

✉ Long-Sheng Chen
lschen@cyut.edu.tw

[1] Department of Information Management, Chaoyang University of Technology, 168 Jifong E. Rd, Wufong District, Taichung 41349, Taiwan

Data mining methods acquire knowledge from examples of existing diagnosis examples and then apply the extracted knowledge to diagnose an illness. However, the data obtained from examples of diagnoses are often imbalanced or skewed, with almost all the instances being labeled as one class (normal), while few instances are labeled as the other class, usually the important class (illness). When building a classifier from such imbalanced/skewed diagnosis data, traditional data mining methods tend to produce high accuracy for the majority class (healthy patients), but poor predictive accuracy for the minority class (diabetic patients) [9–11]. This situation, called the class imbalance problem, poses challenges for typical classifiers that are designed to optimize overall accuracy without taking into account the relative distribution of each class [12, 13]. Many real-world applications involve learning from imbalanced data, such as fraud detection [14], text classification telecommunications management [15], oil spill detection [14, 15], medical diagnosis/monitoring [5, 15–17], financial analysis of loan policy or bankruptcy [18], and protein data [19].

To cope with imbalanced data sets, studies have proposed resampling methods [11, 12, 14, 16, 20, 21], feature selection [22, 23], adjusting the cost matrices [17], and moving the decision thresholds [4, 15, 24]. Resampling methods reduce the data imbalance by undersampling (removing) instances from the majority class or oversampling (duplicating) the examples from the minority class, or both. Feature selection removes irrelevant attributes to build a good classification model when the class distribution is too skewed [22]. Adjusting the cost matrices (adjusting cost) improves the prediction accuracy by adjusting the cost (weight) for each class or by changing the strength of the rules [17]. Approaches that move the decision thresholds try to adapt the decision thresholds by imposing a bias on the minority class. However, each method has both advantages and disadvantages. Taking computational cost into consideration, resampling methods are the most popular and easiest to use. However, they lack a rigorous and systematic treatment of the imbalanced data [24].

The present study proposes the neural-network-based resampling (NNR) method that uses the back-propagation neural network (BPNN) to filter samples and balance class distribution. Then, SVMs are employed to build a model to predict diabetes mellitus. Real diabetes data from a regional hospital in Taiwan and several biological data sets are used to demonstrate the effectiveness of the proposed method. In addition, the proposed NNR method is compared to traditional methods, including those based on oversampling, undersampling, and cost adjustment. The results indicate that the proposed NNR method dramatically improves the detection of diabetes.

2 Class Imbalance Problems

Many solutions have been proposed for class imbalance problems. Some researchers focus on feature selection. For example, Laradji et al. [23] integrated feature selection into ensemble learning methods for improving the performance of defect classification. Yang et al. [25] proposed the comprehensive measure feature selection method for class imbalance problems, and compared it with other feature selection methods. Su and Hsiao [26] employed the Mahalanobis-Taguchi system to improve the performance of classifying imbalanced data.

In practice, when applying these solutions for classifying imbalanced data, computational cost and complexity should be considered. The most important concern is ease of use. Therefore, this study focuses on resampling methods. There are three types of resampling method, namely oversampling, undersampling, and hybrid approaches [27]. Although they are easy to use, resampling methods lack a rigorous and systematic treatment of the imbalanced data [24]. Therefore, lots of works propose different strategies to improve resampling methods.

Oversampling aims to improve imbalance by duplicating the minority examples, but it might introduce some noise. Therefore, Sáez et al. [13] proposed the minority oversampling technique iterative partitioning filter, which overcomes the problems produced by noisy and borderline examples in imbalanced datasets. Li et al. [28] proposed the random walk oversampling approach to deal with imbalanced data. Gao et al. [29] proposed the probability-density-function-estimation-based oversampling approach for two-class imbalanced classification problems.

Undersampling aims to remove the majority examples in training sets to balance the skewed class distribution. Many works have been presented. For instance, Wang et al. [21] used the boundary region cutting (BRC) algorithm to clarify the disorder boundary and proposed a method for reducing the majority class samples in the dense boundary region. In their work, they used SVM to classify text sentiment data. Tahir et al. [30] presented the inverse random undersampling method, which severely undersamples the majority class, thus creating a large number of distinct training sets. Galar et al. [31] presented an ensemble construction algorithm that combines random undersampling with the Boosting algorithm. Yu et al. [32] developed a method based on ant colony optimization (ACO) to handle imbalanced DNA microarray data. In their method, a modified ACO algorithm is employed to filter less informative majority samples.

Hybrid approaches combine oversampling and undersampling, or use a performance index to solve class

imbalance problems. For example, Liu et al. [33] used SVM and presented a sampling approach that combines undersampling and oversampling. Their results showed that their sampling model can effectively improve the classification performance of SVM. Qian et al. [9] presented a resampling ensemble algorithm, in which the minority class examples are oversampled and the majority class examples are undersampled. García et al. [34] compared the performances of several sampling methods such as those based on performance indicators and resampling. Then, they proposed an evaluation index called the index of balanced accuracy. Their experimental results showed that this indicator can effectively deal with class imbalance problems. Zhao et al. [35] proposed a weighted maximum margin criterion to optimize the data set, which made SVM accurately determine the minority class. These resampling techniques do not consider how the data are scattered in the space. Thanathamathee and Lursinsap [27] proposed a technique based on the fact that the location of a separating function in between any two sub-clusters in different classes is defined only by the boundary data of each sub-cluster. Despite lots of works having attempted to determine the appropriate resampling proportion in each class by using a trial-and-error method to build a classier with imbalanced data, the optimal strategy for each class may be infeasible when using such a method. Therefore, Tong et al. [36] presented an analytical procedure for determining the optimal resampling strategy based on design of experiments and response surface methodologies. Chen et al. [37] presented a Mahalanobis distance-support vector machines (MD-SVM) learning scheme. In MD-SVM, MD is used to filter the majority examples, and then SVM is employed to classify imbalanced data. However, Błaszczyński and Stefanowski [11] indicated that integrating bagging with undersampling is more powerful than doing so with oversampling. Therefore, the proposed NNR follows undersampling strategies.

SVM is a popular classifier for dealing with class imbalance problems. Moraes et al. [38] showed that SVM can better handle imbalanced data compared to neural networks considering the computational cost. Sun et al. [39] found that the SVM classifier is the best method for dealing with the imbalanced data from their experiments. Yu et al. [32] used SVM to classify skewed DNA microarray data. In addition, because SVM has a complete theory of modules and is easy to use, it is suitable for high-dimensional and nonlinear classification problems. Therefore, the present study uses SVM as the basic classifier. In addition, this study employs three methods, namely undersampling, oversampling, cost adjustment, as benchmarks.

3 Methods

This section describes the proposed NNR approach. The six major steps are shown in Fig. 1. The procedure is described below.

Step 1 Data collection

We collected biological data from normal and abnormal (diabetic patients/illness) examples. The experimental data sets are from the health examination data of a regional hospital in northern Taiwan and the knowledge extraction based on evolutionary learning (KEEL) website.

Step 2 Data preparation

For the collected data, we deal with missing data and noisy data. Since the data size is large, noisy data and examples that contain missing values are removed. Then, based on the diagnosis results of medical experts, the collected data are labeled.

Step 3 NNR method implementation

The NNR method has two phases for balancing the data distribution using resampling. In the first phase, a BPNN is built. In the second phage, the constructed BPNN is used to undersample data. The details are given bellow.

Phase 1: Back-propagation neural network

The back-propagation learning algorithm [40] is the best known training algorithm for neural networks. This iterative gradient algorithm contains a forward pass and a backward pass. The purpose of the forward pass is to obtain the activation value and the backward pass is used to adjust weights and biases according to the difference between the desired and actual network outputs. These two passes are iterated until the network converges. The feed-forward network training by the back-propagation algorithm can be summarized as follows.

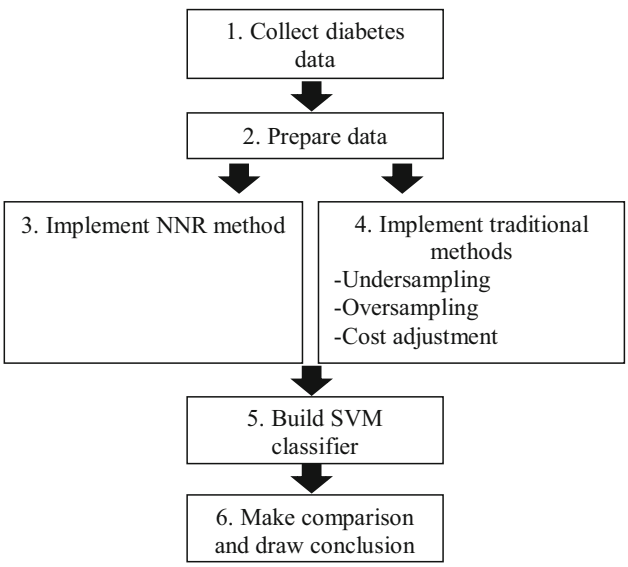

Fig. 1 Implemention procedure of this study

Step 3.1 Determine the architecture.
Step 3.2 Randomly initialize weights.
Step 3.3 Train neural networks.

While the error is too large.
For each training pattern (presented in random order).

Step 3.3.1 Select training pattern and feed it forward to find the actual network output.

Step A Apply the inputs to the network.
Step B Calculate the output for every neuron from the input layer, through the hidden layer(s), to the output layer.
The output from neuron j for pattern p is O_{pj}, where:

$$O_{pj}(net_j) = \frac{1}{1 + e^{-net_j}} \quad (1)$$

and

$$net_j = bias + \sum_k O_{pk} W_{jk} \quad (2)$$

where k ranges over the input indices and W_{jk} is the weight on the connection from input k to neuron j.

Step 3.3.2 Calculate errors and back-propagate error signals.

Step A Calculate the error at the outputs.
The output neuron error signal δ_{pj} is given by:

$$\delta_{\mathrm{pj}} = (\mathrm{T_{pj}} - \mathrm{O_{pj}}) \times \mathrm{O_{pj}} \times (1 - \mathrm{O_{pj}}) \quad (3)$$

where T_{pj} is the target value of output neuron j for pattern p and O_{pj} is the actual output value of output neuron j for pattern p.
Step B Use the output error to compute error signals for pre-output layers.
The hidden neuron error signal δ_{pj} is given by:

$$\delta_{pj} = O_{pj}(1 - O_{pj}) \sum_k \delta_{pk} W_{kj} \quad (4)$$

where δ_{pk} is the error signal of a post-synaptic neuron k and W_{kj} is the weight of the connection from hidden neuron j to the post-synaptic neuron k.

Step 3.3.3 Adjust weights.

Step A Use the error signals to compute weight adjustments.
Compute weight adjustments ΔW_{ji} at time t using:

$$\Delta \mathrm{W_{ji}}(\mathrm{t}) = \eta \times \delta_{pj} \times \mathrm{O_{pi}} + \alpha \times \Delta \mathrm{W_{ji}}(\mathrm{t - 1}) \quad (5)$$

where η is the learning rate and α is the momentum coefficient ($\alpha \in [0, 1]$).
Step B Apply the weight adjustments.
Apply weight adjustments according to:

$$\mathrm{W}_{ji}(\mathrm{t} + 1) = \mathrm{W}_{ji}(\mathrm{t}) + \Delta W_{ji}(\mathrm{t}) \quad (6)$$

.

Step 3.4 Evaluate performance using the test data set.

Phase II: Resampling

Step 3.5 Separate normal and abnormal examples.

In this step, we separate all training examples into normal and abnormal (illness) groups. The minority diabetes examples are kept intact and the majority (healthy) examples are undersampled.

Step 3.6 Rank collected healthy examples.

In this step, we rank all majority (healthy) examples using O_{pj}, which is the actual output value of output neuron j for normal example p.

Step 3.7 Undersample majority examples.

We attempt to sample "different" or "discriminate" majority examples from minority examples. According to the rank list obtained from Step 3.6, we implement the following two undersampling strategies.

Strategy #1: we select examples with small O_{pj} values (remove examples with large O_{pj} values) until the number of minority (diabetes) examples is equal to the number of majority examples. This is also known as the max − min strategy. In this strategy, we remove majority examples that have the highest possibility of belonging to healthy patients.

Strategy #2: we select examples with large O_{pj} values (remove examples with small O_{pj} values) until the number of minority (diabetes) examples is equal to the number of majority examples. This means that majority examples that have the highest possibility of belonging to healthy patients are kept.

Step 4: Undersampling, oversampling, and cost adjustment method implementations

Step 4.1 Implement undersampling.

The majority (healthy) examples are randomly removed until the number of minority (diabetes) examples is equal to the number of majority examples.

Step 4.2 Implement oversampling.

The minority (diabetes) examples are duplicated until the number of minority (diabetes) examples is equal to the number of majority (healthy) examples.

Step 4.3 Implement cost adjustment.

This method improves classification performance by increasing the misclassification cost for minority class. Traditional performance indices consider the misclassification costs of majority and minority instances to be equal. Under the assumption of maximizing the overall classification accuracy, the minority examples are neglected. If we give a penalty (cost) to the minority class, the class imbalance problem will be improved. In this method, different misclassification costs can be incorporated into classes, which avoids direct artificial manipulation of the training set.

We adjust the misclassification cost until the classification performance is improved. For example, if the cost of misclassifying the majority examples (healthy patients) into minority examples (diabetic patients) is equal to 1, we can set the cost of misclassifying the minority examples (diabetic patients) into majority examples (healthy patients) to be larger than 1 until the classification performance is improved. This forces the classifier to tend to increase the ability of identifying diabetic patients.

Step 5: SVM classifier construction

Step 5.1 Construct training and test sets.
The resampled training sets are joined to the test set for learning.

Step 5.2 Select a kernel function and find optimal settings of parameters. In this work, we use the radial basis function kernel function:

$$K(x, x') = \exp\left(\gamma||\mathrm{x} - x'||^2\right) \tag{7}$$

where x and x' represent samples in the input vector, γ is equal to $-1/2\sigma^2$ (where σ is a free parameter), and $\|x-x'\|$ is the Euclidean distance.

Step 5.3 Train SVM.

Step 6: Comparison and conclusions

In this work, we used the geometric mean (GM) of positive accuracy (the ability to identify normal patients) and negative accuracy (the ability to detect the minority diabetic patients) to evaluate the classification performance. We also make comparisons between the proposed NNR method and traditional methods, namely those based on undersampling, oversampling, and cost adjustment. A discussion is then given and conclusions are drawn based on the experimental results.

4 Results and Discussion

4.1 Performance Indices

This section introduces the employed performance measurements. Generally speaking, the easiest way to evaluate the performance of classifiers is based on the confusion matrix, as shown in Table 1.

Traditionally, the performance of a classifier is evaluated by considering the overall accuracy against test cases. However, when learning from imbalanced data sets, this measure is often not sufficient. For example, it is straightforward to create a classifier with an accuracy of 98 % in a domain where the majority class proportion corresponds to 98 % of the examples by simply forecasting every new example as belonging to the majority class. Another fact is that the metric considers different classification errors to be equally important. However, a highly imbalanced class problem has nonequal error costs that favor the minority class, which is often the class of primary interest. Therefore, following other studies [16, 19, 20, 41, 42], we use overall accuracy (OA), GM, and F1 score to evaluate the performance of the models. GM is defined as:

$$\sqrt{\text{Positive accuracy} \times \text{ Negative accuracy}} \tag{8}$$

where *Positive accuracy* (*PA*) and *Negative accuracy* (*NA*) are calculated as $TP/(FN + TP)$ and $TN/(TN + FP)$, respectively (where *TP* true positive, *TN* true negative, *FP* false positive, *FN* false negative. This measure is used to maximize the accuracy for each of the two classes while keeping these accuracies balanced. Another performance index is F1 score, which is defined as:

Table 1 Confusion matrix

	Predicted positive (normal)	Predicted negative (diabetic)
Actual positives (normal)	*TP* (number of true positives)	*FN* (number of false negatives)
Actual negatives (diabetic)	*FP* (number of false positives)	*TN* (number of true negatives)

$$(2 \times \text{Recall} \times \text{Precision})/(\text{Recall} + \text{Precision}) \quad (9)$$

where *Precision* and *Recall* are calculated as $TP/(TP + FP)$ and $TP/(TP + FN)$, respectively.

F1 incorporates the recall and precision into a single number. Therefore, F1 is high when both the recall and precision are high. F1 thus measures the "goodness" of a learning algorithm in the current class of interest.

4.2 Data Collection

Real diabetes data were used. The employed diabetes data are from the health examination database of a regional hospital in northern Taiwan. We obtained 2000 raw data. After 63 examples that contained missing values and noisy data were removed, 1937 objects remained for further analysis. Among them, there were 1729 positive instances (healthy patients) and 208 negative instances (diabetic patients). These examples were divided into training and test objects. A five-fold cross validation experiment was employed. The data sizes of the training and test sets are given in Table 2.

Table 3 shows 23 attributes of these data. They are biochemical or physical test items and their values are continuous except for the first one (i.e., "Gender"). Although there are different types of diabetes (type 1, type 2, and gestational diabetes), they are combined and considered as diabetes. Therefore, we have 2 classes, namely positive (healthy patients) and negative (diabetic patients).

4.3 Experimental Results

Results for this diabetes data set, as shown in Table 4, were averaged over five-fold cross validation experiments, in which the data set was partitioned into five equal-sized sets. Each set was then used in turn as the test set. In this table, PA and NA represent the abilities of detecting healthy and diabetic patients, respectively. G-mean and F1 are integrated indices that balance PA and NA. From this table, the oversampling and cost adjustment (cost = 2) techniques have no significant improvement in detecting diabetic patients, since their NAs are equal to 0 %.

Table 2 Data sizes of training and test sets

Experiment	Training (Pos:Neg)	Test (Pos:Neg)
Fold #1	1383:166	346:42
Fold #2	1383:166	346:42
Fold #3	1383:166	346:42
Fold #4	1383:167	346:41
Fold #5	1384:167	345:41

Pos positive examples (healthy patients), *Neg* negative examples (diabetic patients)

The undersampling method increases the ability of identifying diabetic patients (NA = 100 %), but the ability of detecting healthy patients decreases to 5.66 %, which is unacceptable. For the proposed method, strategy #1 is significantly better than strategy #2 in terms of GM, OA, and F1. However, strategy #2 has the highest ability of detecting minority examples (NA: 91.84 %) among all methods, even strategy #2 loses classification ability of identifying majority examples (PA: 73.56 %). Therefore, NNR with strategy #1 is a better method than strategy #2. Compared to conventional methods, NNR with strategy #1 has the best performances in terms of OA, GM, and F1. Moreover, the proposed method (NNR with strategy #1) has the lowest standard deviation, indicating stable classification.

Figure 2 shows comparisons between the proposed method and traditional methods. The oversampling and cost adjustment techniques outperform the undersampling method. Generally speaking, among these techniques, NNR with strategy #1 significantly improves the detection of diabetic patients and has stable performance.

4.4 Validation Using Other Biological Data Sets

In order to validate the effectiveness of the proposed methods, we utilized three biological data sets from KEEL. They can be accessed at http://sci2s.ugr.es/keel/studies.php?cat=imb. These imbalanced data are related to "thyroid" and "yeast". Table 5 shows their basic information.

Table 6 summarizes the results of these imbalanced data sets. The proposed NNR method with strategy #1 outperforms NNR with strategy #2, undersampling, oversampling, and cost adjustment in "new-thyroid1" and "yeast-2_vs_4" in terms of GM and F1. However, for the "yeast3" data set, the NNR method with strategy #1 is ranked second and third in terms of GM and OA, respectively. To sum up, the proposed NNR method with strategy #1 has the best performance for two of the three biological imbalanced data sets. The proposed method is thus effective for data over than diabetic data.

Table 3 Attributes employed for detecting diabetes

#1	Gender	#5	FEV1 (forced expiratory volume in one second)	#9	SGOT (serum glutamic oxaloacetic transaminase)	#13	BUN (Blood urea nitrogen)	#17	Thyroxine	#21	HDL (high-density lipoprotein)
#2	Age	#6	PFR (peak filpng rate)	#10	SGPT (serum glutamic-pyruvic transaminase)	#14	Creatinine	#18	Uric acid	#22	ELDL (elevated low density lipid cholesterol)
#3	Vital capacity	#7	Albumin	#11	ALP (alkaline phosphatase)	#15	Glucose AC (ante cibum)	#19	Cholesterol	#23	LDL (low-density lipoprotein)
#4	Predicted VC (vital capacity)	#8	Total protein	#12	Total bilirubin	#16	Glucose PC (post cibum)	#20	Triglyceride		

Table 4 Summary of experimental results (DM)

Method Index	NNR strategy #1 (%)		NNR strategy #2 (%)		Oversampling (%)		Undersampling (%)		Cost adjustment (cost = 2) (%)	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD
PA	98.38	1.38	73.56	12.27	100	0.00	5.66	10.92	100	0.00
NA	76.45	5.14	91.84	3.96	0.00	0.00	100	0.00	0.00	0.00
GM	86.68	3.09	81.86	5.34	0.00	0.00	16.30	19.39	0.00	0.00
OA	96.03	1.44	75.52	10.62	89.26	0.12	15.80	9.69	89.26	0.12
F1	97.78	0.82	83.84	7.69	94.33	0.07	9.29	17.36	94.33	0.07

SD standard deviation

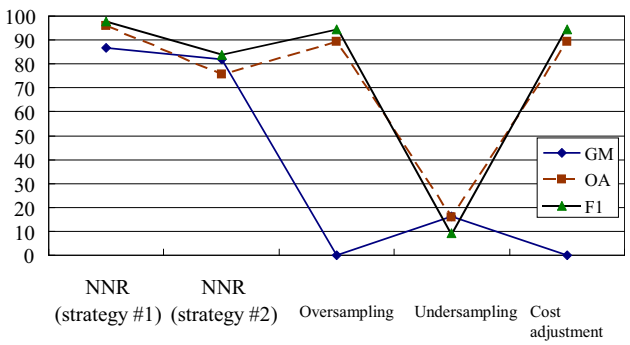

Fig. 2 Comparisons between proposed method and traditional methods

5 Conclusion

This study proposed a neural-network-based resampling method to improve the ability of SVM classifiers to detect diabetic patients. The proposed NNR has two phases. In the first phase, a BPNN filters the majority examples by implementing two resampling strategies. The results indicate that an effective strategy is to keep examples that have low probabilities of belonging to the majority class, and to remove examples that have high probabilities of belonging to the majority class. In the second phase, the resampled training set is used to build SVM classifiers. The max–min concept is applied in the proposed method. Real-world data and three biological data sets from the KEEL database were employed to evaluate the effectiveness of the proposed method and three traditional methods, namely oversampling, undersampling, and cost adjustment. The experimental results show that the proposed method is superior in terms of identifying diabetic patients.

The proposed NNR method was shown to be superior to traditional solutions for classifying imbalanced medical/biological data. It is useful for detecting some rare diseases such as Middle East Respiratory Syndrome and Severe

Table 5 Employed biological data sets

Data set name	Data size	No. of attributes	IR	Data source
New-thyroid 1	215	5	5.14	http://sci2 s.ugr.es/keel/imbalanced.php#subA
Yeast 3	1484	8	8.1	
Yeast-2_vs_4	514	8	9.08	

Table 6 Results for biological data sets

Method Data set	NNR strategy #1 (%)	NNR strategy #2 (%)	Undersampling (%)	Oversampling (%)	Cost adjustment (%)
New-thyroid 1					
GM	91.29	84.90	84.90	81.65	65.47
OA	86.05	95.35	97.67	93.35	90.76
F1	98.67	97.37	91.43	97.37	94.74
Yeast 3					
GM	91.08	83.94	73.85	82.82	92.22
OA	84.85	77.10	59.60	91.92	90.91
F1	90.85	85.34	70.59	95.40	96.93
Yeast-2_vs_4					
GM	98.91	90.02	86.66	74.39	94.83
OA	98.06	95.15	93.20	97.09	98.06
F1	98.91	89.94	90.91	98.40	98.90

Acute Respiratory Syndrome. In the beginning of the infectious period of these rare diseases, the number of positive examples will be much fewer than the number of normal patients.

In the future, we hope to build an automatic diagnosis system that can identify diabetic patients. Such a system will be helpful in developing countries that lack sufficient medical resources. Moreover, in this study, we use 20 biological data which still needs complex equipment to get experiment data, future works can utilize other kind of input variables that can be got easily. Feature selection methods can also be introduced to select the important input variables. This might shorten the computational time required for building predictive models and reduce the cost of collecting data. Moreover, the ability to predict pre-diabetes will give medical experts more time to cure diabetes.

Acknowledgments This work was supported in part by the Ministry of Science and Technology, Taiwan (Grant NSC 101-2628-E-324-004-MY3).

References

1. Sumi, S., Yanai, G., Qi, M., Sakata, N., Qi, Z., Yang, K., et al. (2014). Review: Macro-encapsulation of islets in polyvinyl alcohol hydrogel. *Journal of Medical and Biological Engineering, 34*, 204–210.
2. Aljumah, A. A., Ahamad, M. G., & Siddiqui, M. K. (2013). Application of data mining: Diabetes health care in young and old patients. *Journal of King Saud University-Computer and Information Sciences, 25*, 127–136.
3. WHO. Facts and figures about diabetes. Accessed March 1, 2015. http://www.who.int/diabetes/facts/en/.
4. Srikanth, T., Napper, S. A., Calloway, J. & Reddy, M. R. S. (1997) An expert system to identify different classes of diabetic cardiac autonomic neuropathy (DCAN). *IEEE proceedings of sixteenth southern biomedical engineering conference*, (pp. 458–461).
5. Su, C.-T., Yang, C.-H., Hsu, K.-H., & Chiu, W.-K. (2006). Data mining for the diagnosis of type 2 diabetes from three-dimensional body surface anthropometrical scanning data. *Computers & Mathematics with Applications, 51*, 1075–1092.
6. Yildirim, E. G., Karahoca, A., & Uçar, T. (2011). Dosage planning for diabetes patients using data mining methods. *Procedia Computer Science, 3*, 1374–1380.
7. Meng, X.-H., Huang, Y.-X., Rao, D.-P., Zhang, Q., & Liu, Q. (2013). Comparison of three data mining models for predicting diabetes or prediabetes by risk factors. *Kaohsiung Journal of Medical Sciences, 29*, 93–99.
8. Kang, S., Kang, P., Ko, T., Cho, S., Rhee, S., & Yu, K. (2015). An efficient and effective ensemble of support vector machines for anti-diabetic drug failure prediction. *Expert Systems with Applications,*. doi:10.1016/j.eswa.2015.01.042.
9. Qian, Y., Liang, Y., Li, M., Feng, G., & Shi, X. (2014). A resampling ensemble algorithm for classification of imbalance problems. *Neurocomputing, 143*, 57–67.
10. Ibarguren, I., Pérez, J. M., Muguerza, J., Gurrutxaga, I., & Ibarguren, O. A. I. (2015). Coverage based resampling: Building robust consolidated decision trees. *Knowledge-Based Systems,*. doi:10.1016/j.knosys.2014.12.023.
11. Błaszczyński, J., & Stefanowski, J. (2015). Neighbourhood sampling in bagging for imbalanced data. *Neurocomputing, 150*, 529–542.
12. Zhang, H., & Li, M. (2014). RWO-Sampling: A random walk over-sampling approach to imbalanced data classification. *Information Fusion, 20*, 99–116.
13. Sáez, J. A., Luengo, J., Stefanowski, J., & Herrera, F. (2015). SMOTE-IPF: Addressing the noisy and borderline examples problem in imbalanced classification by a re-sampling method with filtering. *Information Sciences, 291*, 184–203.
14. Chawla, N. V., Bowyer, K., Hall, L., & Kegelmeyer, W. (2002). SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research, 16*, 231–357.
15. Chawla, N. V., Japkowicz, N., & Kolcz, A. (2004). Editorial: Special issue on learning from imbalanced data sets. *SIGKDD Explorations, 6*, 1–6.
16. Batista, G., Prati, R. C., & Monard, M. C. (2004). A study of the behavior of several methods for balancing machine learning training data. *SIGKDD Explorations, 6*, 20–29.

17. Grzymala-Busse, J. W., Stefanowski, J., & Wilk, S. (2004). A comparison of two approaches to data mining from imbalanced data. *Lecture Notes in Computer Science, 3213*, 757–763.
18. Jo, T., & Japkowicz, N. (2004). Class imbalances versus small disjuncts. *SIGKDD Explorations, 6*, 40–49.
19. Provost, F., & Fawcett, T. (2001). Robust classification for imprecise environments. *Machine Learning, 42*, 203–231.
20. Guo, H., & Viktor, H. L. (2004). Learning from imbalanced data sets with boosting and data generation: The DataBoost-IM approach. *SIGKDD Explorations, 6*, 30–39.
21. Wang, S., Li, D., Zhao, L., & Zhang, J. (2013). Sample cutting method for imbalanced text sentiment classification based on BRC. *Knowledge-Based Systems, 37*, 451–461.
22. Maldonado, S., Weber, R., & Famili, F. (2014). Feature selection for high-dimensional class-imbalanced data sets using support vector machines. *Information Sciences, 286*, 228–246.
23. Laradji, I. H., Alshayeb, M., & Ghouti, L. (2015). Software defect prediction using ensemble learning on selected features. *Information and Software Technology, 58*, 388–402.
24. Huang, K., Yang, H., King, I., & Lyu, M. (2004). Learning classifiers from imbalanced data based on biased minimax probability machine. *Proceedings of the 04' IEEE computer society conference on computer vision and pattern recognition (CVPR'04)*, (pp. 558–563).
25. Yang, J., Liu, Y., Zhu, X., Liu, Z., & Zhang, X. (2012). A new feature selection based on comprehensive measurement both in inter-category and intra-category for text categorization. *Information Processing & Management, 48*, 741–754.
26. Su, C.-T., & Hsiao, Y.-H. (2007). An evaluation of the robustness of MTS for imbalanced data. *IEEE Transactions on Knowledge and Data Engineering, 19*, 1321–1332.
27. Thanathamathee, P., & Lursinsap, C. (2013). Handling imbalanced data sets with synthetic boundary data generation using bootstrap re-sampling and AdaBoost techniques. *Pattern Recognition Letters, 34*, 1339–1347.
28. Li, S., Zhou, G., Wang, Z., Lee, S. Y. M., & Wang, R. (2011) Imbalanced sentiment classification. *Proceedings of the 20th ACM international conference on information and knowledge management*, (pp. 2469–2472).
29. Gao, M., Hong, X., Chen, S., Harris, C. J., & Khalaf, E. (2014). PDFOS: PDF estimation based over-sampling for imbalanced two-class problems. *Neurocomputing, 138*, 248–259.
30. Tahir, M. A., Kittler, J., & Yan, F. (2012). Inverse random undersampling for class imbalance problem and its application to multi-label classification. *Pattern Recognition, 45*, 3738–3750.
31. Galar, M., Fernández, A., Barrenechea, E., & Herrera, F. (2013). EUSBoost: Enhancing ensembles for highly imbalanced data-sets by evolutionary undersampling. *Pattern Recognition, 46*, 3460–3471.
32. Yu, H., Ni, J., & Zhao, J. (2013). ACOSampling: An ant colony optimization-based undersampling method for classifying imbalanced DNA microarray data. *Neurocomputing, 101*, 309–318.
33. Liu, Y., Yu, X., Huang, J. X., & An, A. (2011). Combining integrated sampling with SVM ensembles for learning from imbalanced datasets. *Information Processing & Management, 47*, 617–631.
34. García, V., Sánchez, J. S., & Mollineda, R. A. (2011). On the effectiveness of preprocessing methods when dealing with different levels of class imbalance. *Knowledge-Based Systems, 25*, 13–21.
35. Zhao, Z., Zhong, P., & Zhao, Y. (2011). Learning SVM with weighted maximum margin criterion for classification of imbalanced data. *Mathematical and Computer Modelling, 54*, 1093–1099.
36. Tong, L.-I., Chang, Y.-C., & Lin, S.-H. (2011). Determining the optimal re-sampling strategy for a classification model with imbalanced data using design of experiments and response surface methodologies. *Expert Systems with Applications, 38*, 4222–4227.
37. Chen, L.-S., Hsu, C.-C., & Chang, Y.-S. (2010). Developing a novel two-phase learning scheme for the class imbalance problem. *International Journal of Innovative Computing, Information and Control, 6*, 4979–4994.
38. Moraes, R., Valiati, J. F., Wilson, P., & Neto, G. (2013). Document-level sentiment classification: An empirical comparison between SVM and ANN. *Expert Systems with Applications, 40*, 621–633.
39. Sun, A., Lim, E. P., & Liu, Y. (2009). On strategies for imbalanced text classification using SVM: A comparative study. *Decision Support Systems, 48*, 191–201.
40. Rumelhart, D., & McClelland, J. (1986). *Parallel distributed processing*. Cambridge, MA: MIT Press.
41. Radivojac, P., Chawla, N. C., Dunker, A. K., & Obradovic, Z. (2004). Classification and knowledge discovery in protein databases. *Journal of Biomedical Informatics, 37*, 224–239.
42. Estabrooks, A., Jo, T., & Japkowicz, N. (2004). A multiple resampling methods for learning from imbalanced data sets. *Computational Intelligence, 20*, 18–36.