Risk Prediction and Risk Factors Identification from Imbalanced Data with RPMBGA+

Topon K Paul*, Ken Ueno
Corporate Research & Development Center
Toshiba Corporation
1, Komukai-Toshiba-cho, Saiwai-ku
Kawasaki-shi, Kanagawa 212-8582, Japan
toponkumar.paul@toshiba.co.jp
ken.ueno@toshiba.co.jp

Koichiro Iwata, Toshio Hayashi,
and Nobuyoshi Honda
Toshiba Corporation
1-1, Shibaura 1-chome, Minato-ku
Tokyo 105-8001, Japan
koichiro.iwata@toshiba.co.jp
toshio7.hayashi@toshiba.co.jp
nobuyoshi.honda@toshiba.co.jp

ABSTRACT

In this paper, we propose a new method to predict the risk of an event very accurately from imbalanced data in which the number of instances of the majority class is very larger than that of the minority class and to identify the features that are relevant for the target risk factor. To solve the trade-off between the prediction rates of the majority and the minority classes, three input parameters are used, which supply the costs of misclassification of an instance from the majority and the minority classes or the sensitivity threshold of the minority class. To get relevant features and to utilize the prior information about the relationship of a feature with the target risk factor, a probabilistic model building genetic algorithm called RPMBGA$^+$ is employed. By applying the proposed technique to the health checkup and lifestyle data of Toshiba Corporation, we have found that the proposed method improves the sensitivity of the minority class and selects a very small number of informative features.

Categories and Subject Descriptors

I.2.8 [**ARTIFICIAL INTELLIGENCE**]: Problem Solving, Control Methods, and Search; I.2.6 [**ARTIFICIAL INTELLIGENCE**]: Learning—*Knowledge acquisition, Parameter learning*; I.5 [**PATTERN RECOGNITION**]: Applications

General Terms

Algorithms

Keywords

Risk prediction, feature selection, classification, rare event, imbalanced data, genetic algorithm, fitness evaluation

*Corresponding author

GECCO'08, July 12–16, 2008, Atlanta, Georgia, USA.
ACM 978-1-60558-131-6/08/07.

1. INTRODUCTION

Today various organizations and companies are interested in designing a system that can predict the risk associated with an event of an entity, such as a person, a device and a system, very accurately. Among the different types of events, rare events, such as the abnormal behavior of a person at a public place, a loan defaulter in a consumer finance company, and the heart-attack of a person, are of special interest because the prediction of a rare event is very difficult. In this context, at first, an event is defined in terms of different features, and the target risk factor, such as behavior, loan status, and blood pressure, is decided. Then data about the features of the events are collected from different entities where the labels of the target risk factor are known. The value of the target risk factor in an event indicates whether the event is normal or rare; the interpretation of the normal event and the rare event depends on the problem. For example, in a consumer finance company, a loan defaulter is a rare event and a person who has repaid the loan is a normal event whereas in a video surveillance system, the abnormal behavior of a person is a rare event and the normal behavior of a person is a normal event. The collected data usually contains a very high number of normal events but a very small number of rare events; that is, the data are imbalanced, and the normal events form the majority class while the rare events form the minority class. Hereafter, the data of an event is referred to as an instance, and the value of the target risk factor in an instance is referred to as a class label.

In designing a risk prediction system, the first step is to design a model that is trained using a collection of labeled instances from the majority and the minority classes. The main constituent of the model is a classifier, such as IB1 [1], k-nearest neighbor (kNN) classifier [3], naive-Bayes classifier, decision tree (C4.5) [15], neural network [16], and support vector machine (SVM) [18]. Learning of a model means the learning of the values of different parameters of the constituents, such as the number of the nearest neighbors (k) in the kNN classifier, and the values of the weights of different instances (support vectors) in SVM. Sometimes, the learning of a model involves the selection of a subset of features (hereafter, referred to as a feature subset) that are highly related with the target risk factor. In that case, the goodness of a candidate feature subset is evaluated using a classifier and a scoring method [10, 11, 12, 13].

Next step is to evaluate the model. After a model is learned, its performance is evaluated by using validation data that are not used during the learning of the model, and this performance gives an estimate of how accurately the model will predict the label of the target risk factor. Sometimes, the accuracy measured using a cross-validation technique during the learning of the model is used as a measure of the performance of the model on unseen data.

1.1 Challenges of the task

Very accurate prediction of the class label of an instance using a model that is learned on imbalanced data is very difficult because during the learning of the model, the majority class biases the model toward it, and the learned model very often fails to accurately predict the class label of an instance from the minority class. In some cases, it has been found that all the instances from the majority class are correctly classified by the model but none of the instances from the minority is correctly classified. If a model is designed by focusing on the accurate prediction of the instances from the minority class, the model will misclassify a very large number of instances from the majority class. That is, most traditional methods face a trade-off between the accurate prediction of an instance from the majority and the minority classes—when the prediction rate of one class increases, the prediction rate of the other class decreases.

In addition, all the features that characterize an instance are not always relevant for the distinction of the instances of the majority class from those of the minority class. The irrelevant features sometimes affect negatively the performance of a model learned using other relevant features. Moreover, acquiring the values of these irrelevant features sometimes may cost money and time. That is why, sometimes selection of a relevant feature subset is done during model selection. Given n features, there are $2^n - 1$ candidate feature subsets. When the number of features and/or the number of instances in a data set is (are) very large, the exhaustive search for the optimal feature subset is not possible because the search space becomes huge, and the computation time becomes high. Instead, a global heuristic search algorithm, such as genetic algorithm (GA) [6, 7] may be used to generate candidate feature subsets. However, not all the heuristic methods return an optimal feature subset with a very small number of features selected in it. When the number of features is very large, approximately half of the features are selected in each candidate feature subset of a genetic algorithm. A very small number of selected features may sometimes provide the insights into the problem at hand.

Sometimes, some features of a data set are known to be associated with the target risk factor. However, the relationships of other features with the target factor are unknown. In that case, we need to find a feature subset with the known and the unknown features that in combination with a classifier will predict the label of a test instance very accurately. Most traditional methods of feature subset selection do not take into account this aspect of the data.

1.2 Some related works

For classification of imbalanced data, various techniques have been used during the learning of a classifier. These techniques include resizing training sets [9, 8], adjusting misclassification costs [4], learning rules for skewed data (RLSD) [20], and cost sensitive boosting [17]. In resizing training sets, either the minority class is over-sampled or the majority class is under-sampled. In both cases, the misclassification rate of the samples of minority class will decrease but the misclassification rate of majority class will increase. In adjusting misclassification costs, higher cost is set to the misclassification of a sample from the minority class. In cost sensitive boosting, genetic algorithm is used to search the optimum cost setup of each class. However, these methods have not focused on identifying a subset of important features from the data.

1.3 Summary of the proposed method

To solve the trade-off between the prediction rates of the majority class and the minority class by a model on imbalanced data, a three-parameter input unit is proposed, which supplies either the costs of misclassification of an instance from the majority and the minority classes or the sensitivity threshold of the minority class. Based on the values of these three input parameters, an appropriate scoring method is applied. If the costs of misclassification are supplied, an aggregated cost of misclassification of instances by the model is provided; otherwise, a score is returned by combining the sensitivity and the specificity information. When the sensitivity returned by the model is lower than the threshold, a normal score is returned that balances the sensitivity and specificity to some extent; otherwise, a scaled up score is returned. In this paper, we present some functions that can be employed to calculate the appropriate fitness score.

To get a very relevant feature subset and to utilize the prior information about the relationships of a feature with the target risk factor, a feature subset generation method based on probabilistic model building genetic algorithm (PMBGA) [14] is proposed. The method generates candidate feature subsets by sampling a probability vector in which each value specifies the probability of a feature being selected in a candidate feature subset. The prior information about the feature is used during the initialization and the update of the probability vector. A candidate feature subset is evaluated using the method described previously. The feature subset generation method starts with randomly generated feature subsets in each of which approximately half of the features of the data are selected but successively modifies the number of selected features in the candidate feature subsets and finally terminates with a highly relevant feature subset.

1.4 Effectiveness of the proposed method

The effectiveness of the proposed method is demonstrated by performing some experiments on health checkup and lifestyle data of Toshiba Corporation. It has been found that the proposed system improves the area under ROC curve (AUC) as well as the G-score and selects a very small number of informative features.

2. METHOD

The important features are identified from the imbalanced data by using a supervised learning method that utilizes a classifier, a score calculation method, and three evaluation parameters (w_1, w_2, θ). Among the evaluation parameters, w_1 and w_2 are the misclassification costs of an instance from the majority and the minority classes, and $\theta \in [0, 1]$ is the threshold of sensitivity of the minority class. That is, the evaluation parameters provides a vector of values of (w_1, w_2, θ). If the misclassification costs are known, the

vector of values becomes $(w_1, w_2, 0)$; if the misclassification costs are unknown, the vector becomes $(0, 0, \theta)$.

The classifier classifies the data corresponding to the selected features in a feature subset using a cross-validation method and returns the classification statistics: the numbers of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN), which are then utilized by the score calculation method. For classification of imbalanced data, the weights of the majority class and the minority class play a vital role. If no weights are provided, the classification statistics will be biased by the majority class. Even it may happen that all the instances from the majority class are correctly classified but none from the minority class are correctly classified. In this case, different feature subsets will get the same fitness score. If the weights of misclassification are not provided as the evaluation parameters (when w_1 and w_2 are both zero), these weights should be determined from the distribution of the classes.

The score calculation method utilizes the classification statistics and the three evaluation parameters and returns the goodness score of a feature subset, i.e., how good the selected features are for the classification of the instances.

2.1 Generation of candidate feature subsets

The proposed feature subsets generation method is based on random probabilistic model building genetic algorithm (RPMBGA) [11, 12], a variant of genetic algorithm. RPMBGA is a global search heuristic like genetic algorithm but it maintains a vector of probabilities of the features in addition to a population of a set of candidate feature subsets and generates new solutions (feature subsets) by sampling the probability vector instead of using crossover and mutation operations of genetic algorithm. Each candidate feature subset in the population is a vector of 0s and 1s. If a value in the vector is 1, the corresponding feature is selected; otherwise, the feature is not selected. For example, if a data set has 10 features, (1,0,1,0,1,0,1,0,0,1) is a candidate feature subset in which the first, the third, the fifth, the seventh, and the tenth feature are selected. A value $P(X_i, t)$ in the vector of probabilities indicates the probability of the feature X_i being selected in a candidate feature subset in generation t. In our proposed method, the prior information about the features is utilized during the initialization of the probability vector and the update of the probability vector. Since the proposed method is an extension of RPMBGA, we call this proposed method RPMBGA$^+$. The pseudocode of RPMBGA$^+$ is as follows:

Procedure RPMBGA$^+$
BEGIN
  Initialize different controlling parameters;
  Initialize probability vector;
  Generate N feature subsets by sampling the probability vector;
  Evaluate each feature subset in N;
  WHILE termination_criteria NOT satisfied
  BEGIN
    Select S top ranked feature subsets from the population of previous generation;
    Update probability vector;
    Generate O new feature subsets by sampling the updated probability vector;
    Evaluate each feature subset in O;
    Generate new population by combining N and O;
  END
  Get the top ranked feature subset;
END

First, the values of various controlling parameters, such as the population size (N), offspring size (O), selection size (S), and maximum number of generations, are set. Next the probability vector is initialized in the following ways:

$$P(X_i, 0) = \begin{cases} p_i & \text{if } p_i > 0; \\ 0.5 & \text{otherwise} \end{cases} \quad (1)$$

where p_i is the prior information about the relationship of the feature with the target risk factor. When no information is known about the relationship of the feature with the target factor, the probability is set to 0.5, which means that the feature may or may not be selected as an important feature.

Given a probability vector, a candidate feature subset is generated in the following way:

Procedure GenerateFeatureSubset(P(X,t))
  /*P(X,t) is the vector of probabilities at generation t.*/
  BEGIN
    FOR i=1 to n /*n=number of features*/
      BEGIN
        Generate a random number $r \in [0, 1]$;
        IF $(P(X_i, t) > r)$ **THEN**
          X[i]=1;
        ELSE
          X[i]=0;
      END
    Get the candidate feature subset X;
  END

Then the candidate feature subsets are evaluated using a classifier and a scoring method that utilizes the three evaluation parameters. Detailed description about the evaluation of a feature subset is given in next subsection. If the termination criterion, such as the maximum number of generations has passed or the best feature subset in the population has reached the optimum fitness, is not satisfied, the S top ranked feature subsets are selected, based on the goodness scores of the candidate feature subsets in the population, for the update of the probability vector. Using the selected candidate feature subsets, the probability vector is updated in the following way:

$$P(X_i, t+1) = \begin{cases} p_i & \text{if } p_i > 0; \\ \psi(P(X_i, t), M(X_i, t)) & \text{otherwise} \end{cases} \quad (2)$$

where p_i is the prior information about the relationship of the feature X_i with the target risk factor, $M(X_i, t)$ is the probability distribution of the feature X_i in the selected candidate feature subsets, and $\psi(P(X_i, t), M(X_i, t))$ is a function that returns a value between 0 and 1. Various methods have been proposed in literature for the update function $\psi(P(X_i, t), M(X_i, t))$; for example, in PBIL [2], $\psi(P(X_i, t), M(X_i, t))$ is defined in the following way:

$$\psi(P(X_i, t), M(X_i, t)) = \alpha P(X_i, t) + (1 - \alpha) M(X_i, t) \quad (3)$$

where $\alpha \in [0, 1]$ is called learning rate and fixed through each iteration. In RPMBGA [12], that function is defined as follows:

$$\psi(P(X_i, t), M(X_i, t)) = \alpha\beta P(X_i, t) + (1 - \alpha)(1 - \beta) M(X_i, t) \quad (4)$$

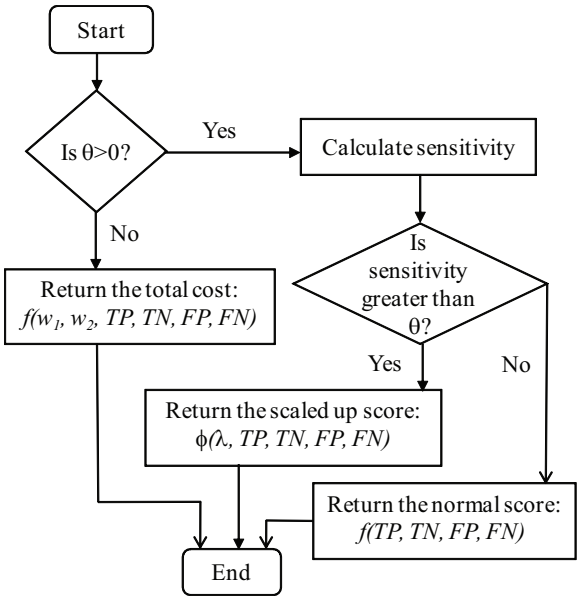

Figure 1: Steps for calculation of score of a feature subset

where $\alpha \in [0, 1]$ is called learning rate and fixed through each iteration, and $\beta \in [0, 1]$ is a random number and changes at each iteration. Due to the inclusion of an extra random parameter less than 1.0, $\psi(P(X_i, t), M(X_i, t))_{RPMBGA} < \psi(P(X_i, t), M(X_i, t))_{PBIL}$. Therefore, when the number of features in a data set is huge, RPMBGA will return a smaller size feature subset than PBIL.

In the next step of RPMBGA$^+$, the newly generated feature subsets are evaluated. Afterward, the old population and the newly generated feature subsets are combined to generate new population. There are different strategies, such as elitism, and CHC [5] to generate the new population. In elitism, $(N > O)$, and the top $(N - O)$ feature subsets of the old population are retained, and the remaining feature subsets are replaced with the newly generated ones. In CHC, $O = N$, and the new population is generated by selecting the best N feature subsets from the combination of $(N + O)$ feature subsets.

After the termination of the algorithm, the selected features in the top ranked feature subset are taken as the important features of the target risk factor.

2.2 Evaluation of a feature subset

During the classification of imbalanced data, a classifier faces the trade-off between sensitivity and specificity. For some applications high sensitivity is required; for some applications, high specificity is desired. In some cases, the misclassification costs of an instance from the majority and the minority class are known and a classification model is desirable that minimizes the total misclassification cost. In most cases, the misclassification costs are unknown but a reasonable sensitivity is desired. Taking into account these situations, we propose a method that returns an appropriate score depending on the situation and the classification statistics. The flowchart of our proposed method for calculation of the score of a feature subset is presented in Fig. 1. Depending on the value of sensitivity threshold (θ), either a total cost of the misclassified instances or a score using sensitivity and specificity information is returned. Here some examples of calculation of score are provided to show how the system works. However, the proposed system is not limited to the following examples; other score calculation methods may be used. Two examples of total cost calculation are given below:

$$f(w_1, w_2, TP, TN, FP, FN) = w_1 * FN + w_2 * FP; \quad (5)$$

and

$$f(w_1, w_2, TP, TN, FP, FN) = \sqrt{w_1 * (FN)^2 + w_2 * (FP)^2}. \quad (6)$$

Two examples of normal score $f(TP, TN, FP, FN)$ are as follows:

$$\text{G-score} = \sqrt{(sensitivity * specificity)}; \quad (7)$$

and

$$\text{AUC} = \frac{1}{2}(sensitivity + specificity) \quad (8)$$

where

$$sensitivity = \frac{TP}{(TP + FN)}; \text{ and}$$
$$specificity = \frac{TN}{(TN + FP)}.$$

Three examples of scaled up score are given below:

$$\phi(\lambda, TP, TN, FP, FN) = \sqrt[\lambda]{f(TP, TN, FP, FN)}; \quad (9)$$

$$\phi(\lambda, TP, TN, FP, FN) = \lambda + f(TP, TN, FP, FN); \quad (10)$$

and

$$\phi(\lambda, TP, TN, FP, FN) = \lambda * f(TP, TN, FP, FN) \quad (11)$$

where $f(TP, TN, FP, FN) \in [0, 1]$ is the normal score and $\lambda > 1$ is a scaled up parameter, say $\lambda = 2$.

3. EXPERIMENTS AND RESULTS

3.1 Data set

In co-operation with occupational health physicians in Toshiba Corporation, Japan, we analyze the anonymized health checkup and life style data of the employees in Toshiba Corporation by employing the proposed system. We are working on this analysis as a part of the health promotion project started in 2007. From the health checkup and lifestyle database, we extracted the data of those employees who had health checkup and lifestyle data of the last seven years and did not have any missing information. The query resulted in 6475 records (instances). Among those employees under investigation, 1347 employees had high blood pressure (labeled as either red or yellow), and the remaining employees had blood pressure within the normal range. Total number of features in the data set is 320.

3.2 Setup of the experiments

Various experiments are performed on the data set using genetic algorithm, RPMBGA and RPMBGA$^+$ with C4.5, SVM or the kNN classifier. For C4.5 and SVM, we use the WEKA [19] implementation with the default settings, and

for kNN, we use our own implementation with Euclidean distance and $k = 5$. Since the number of instances is very large compared to the number of features, we use 10-fold cross-validation technique to evaluate the goodness of a feature subset. As a normal evaluation score, we use the geometric mean of the sensitivity and the specificity (7), and as a scaled-up score, we use (9) with $\lambda = 2$. We assume that the costs of misclassification of a high blood pressure person as normal person and a normal person as a high blood pressure person are unknown and set the sensitivity threshold θ to 0.5. The settings of other parameters are as follows: population size=100, elite size=2, selection size=50%, α (RPMBGA and RPMBGA$^+$)=0.9, crossover and mutation probabilities for GA are 0.8 and 0.1, and maximum number of generations per run=20. For GA, RPMBGA, and RPMBGA$^+$, 10 independent runs are performed.

For genetic algorithm, the initial population is generated randomly but we restrict the number of selected features in each candidate feature subset to 50. Without this restriction, GA terminates with approximately half of the features being selected. We use one-point crossover and bit mutation in GA to generate new feature subsets. Due to lack of prior information about the relationships of the features with blood pressure, each value in the probability vector of both RPMBGA and RPMBGA$^+$ is initialized with 0.5; therefore, the number of selected features in each feature subset in these two algorithms will be higher than that in GA. For all the three methods, elitism is used to generate new population; the best two feature subsets of the previous generation survive for the next generation, and the remaining 98 feature subsets are replaced with the newly generated ones. In each run, the algorithm terminates when both the sensitivity and the specificity are 1.0 (=100%) or the maximum number of generations has passed.

3.3 Results

3.3.1 Classification statistics

The results obtained by applying various classifiers and feature selection methods are shown in Table 1. The column 'Features' indicates the feature selection method used in the experiments; 'All features' means all the features are used while 'Single feature' means the best feature in the dataset that results in the highest G-score. A feature selection and/or a classifier is (are) evaluated in terms of sensitivity, specificity, accuracy, AUC, and geometric mean of sensitivity and specificity (G-score).

First the classifiers are employed without any feature selection. As it can be seen that the accuracy is totally influenced by the majority class; specificity is very high but the sensitivity is very low. SVM obtains the lower sensitivity as well as the G-score. C4.5 and kNN relatively improve the sensitivity of the minority class but this sensitivity is still lower than the specificity. Next the kNN classifier is applied to the data of single attributes to determine whether there exists any single attribute that has higher classification ability. Interestingly, it has been found that there exists a single attribute that has better data separation capability than all other attributes.

In order to obtain a better feature subset, genetic algorithm is applied with C4.5, SVM or the kNN classifier. In the experiments, we have found that the use of SVM with GA is meaningless because for every feature subset, the SVM perfectly classifies all the instances from the majority class but none from the minority class. Comparatively better results in terms of AUC and G-score are obtained with C4.5.

Table 2: Number of selected features

Method	Classifier	Min	Max	Avg	BestG
GA	C4.5	21	45	29	28
GA	kNN	16	34	24	29
RPMBGA	kNN	3	29	14	25
RPMBGA$^+$	kNN	3	33	14	4

Finally, RPMBGA and the proposed RPMBGA$^+$ are applied to the data set with the kNN classifier. In terms of sensitivity, AUC and G-score, these two methods are much better than the previous methods, and the RPMBGA$^+$ obtains the best results. In comparison with the results of all features using C4.5 classifier, RPMBGA$^+$ improves the sensitivity by 28% and G-score by 26%.

3.3.2 Selected features

In Table 2, the number of selected features by GA, RPMBGA and RPMBGA$^+$ are presented. In the table, 'Avg' means an average value, and a value in the column 'BestG' indicates the number of selected features in the feature subset that has the highest G-score. Though GA and RPMBGA are very much competitive in terms of classification statistics, RPMBGA and RPMBGA$^+$ on the average always come up with a feature subset that has a very small number of features selected. Since the feature subset having the highest G-score is selected as the most important feature subset, RPMBGA$^+$ is better than other two methods. Starting from the initial population where each feature subset has about 50% of the features selected, RPMBGA or RPMBGA$^+$ successively reduces the size and finally terminates with a feature subset in which several features selected.

By analyzing the selected features, it has been found that the blood pressures of previous years have a great influence on the blood pressure in the following year. The four features that result in higher G-score are the diastolic (low blood) pressures in the sixth and the third most recent year, and the systolic (high blood) pressure in the fourth and the first most recent year. In addition to blood pressure, it has been found that the weight in the third most recent year also influences the blood pressure in the most recent year. To determine the influence of other factors on the blood pressure, the proposed system should be applied after removing the data of the blood pressures in the previous years.

4. CONCLUSION

In this paper, we have focused on identification of important features from the imbalanced data using a population based global search heuristics. To maintain the balance between the sensitivity and the specificity, we have proposed a new evaluation technique that results in higher G-score as well as higher AUC. In addition to these, we have presented frameworks about how to incorporate the prior information about the features during selection of the important features from data. Using RPMBGA$^+$ with the new evaluation technique, we have found that the acquired AUC or the G-score as well as the sensitivity of the minority class is much better than GA-based method, and the number of selected features in the best feature subset is very small. These experiments

Table 1: Classification statistics

Features	Classifier	Sensitivity	Specificity	Accuracy	AUC	G-score
All features	C4.5	0.36	0.85	0.75	0.61	0.56
All features	SVM (RBF)	0.0	1.0	0.79	0.5	0.0
All features	SVM(Linear)	0.29	0.95	0.81	0.62	0.53
All features	kNN	0.34	0.69	0.61	0.51	0.48
Single feature	kNN	0.47	0.72	0.67	0.59	0.58
GA	C4.5	0.37±0.02	0.88±0.01	0.77±0.01	0.63±0	0.57±0.01
GA	SVM (RBF)	0.0±0.0	1.0±0.0	0.79±0.0	0.5±0.0	0.0±0.0
GA	kNN	0.31±0.01	0.90±0.01	0.78±0.01	0.61±0.01	0.53±0.01
RPMBGA	kNN	0.56±0.01	0.76±0.01	0.72±0.01	0.66±0.01	0.65±0.01
RPMBGA$^+$	kNN	**0.64**±0.02	0.70±0.01	0.69±0.01	**0.67**±0.01	**0.82**±0.01

strongly suggest that RPMBGA$^+$ is very much effective in selecting important features from the data containing huge number of features.

Due to very long execution time of the algorithm, we have performed a limited number of experiments on the health checkup and lifestyle data. In our future works, we want to perform more experiments on the data. In addition, it has been found that the sensitivity of the minority class can be improved by reducing the size of the majority class by applying a suitable instance subset selection method. In our future works, we want to apply the proposed method to the data that has been downsized by reducing the instances from the majority class. We also want to perform some other comparative experiments employing some deterministic search algorithms.

5. REFERENCES

[1] D. W. Aha, D. Kibler, and M. Albert. Instance-based learning algorithms. *Machine Learning*, 6:37–66, 1991.

[2] S. Baluja. Population-based incremental learning: A method for integrating genetic search based function optimization and competitive learning. Technical Report CMU-CS-94-163, Carnegie Mellon University, Pittsburgh, Pennsylvania, 1994.

[3] B. Dasarathy. *Nearest Neighbor(NN) Norms: NN Pattern Classification Techniques*. IEEE Computer Society Press, 1991.

[4] P. Domingos. Metacost: A general method for making classifiers cost-sensitive. In *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 155–164, San Diego, CA, USA, 1999.

[5] L. Eshelman. The CHC adaptive search algorithm. In *Foundations of Genetic Algorithms I*, pages 265–283. Morgan Kauffman, San Mateo CA, 1991.

[6] D. E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, Reading, MA, 1989.

[7] J. Holland. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor, Michigan, 1975.

[8] M. Kubat and S. Matwin. Addressing the curse of imbalanced data sets: One-sided sampling. In *Proceedings of the Fourteenth International Conference on Machine Learning*, pages 179–186, 1997.

[9] C. Ling and C. Li. Data mining for direct marketing: Problems and solutions. In *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 73–79, 1998.

[10] T. K. Paul and H. Iba. Identification of informative genes for molecular classification using probabilistic model building genetic algorithm. In *Proceedings of Genetic and Evolutionary Computation Conference 2004*, pages 414–425. 2004.

[11] T. K. Paul and H. Iba. Extraction of informative genes from microarray data. In *Proceedings of the 2005 Conference on Genetic and Evolutionary Computation*, pages 453–460, 2005.

[12] T. K. Paul and H. Iba. Gene selection for classification of cancers using probabilistic model building genetic algorithm. *BioSystems*, 82(3):208–225, 2005.

[13] T. K. Paul and H. Iba. Prediction of cancer class with majority voting genetic programming classifier using gene expression data. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 23 August 2007. Preprint on IEEE Computer Society Digital Library. IEEE Computer Society, 11 April 2008.

[14] M. Pelikan, D. Goldberg, and F. Lobo. A survey of optimizations by building and using probabilistic models. Technical Report, Illigal Report 99018, Illinois Genetic Algorithms Laboratory, University of Illinois at Urbana-Champaign, USA, 1999.

[15] J. Quinlan. *C4.5:Programs for Machine Learning*. Morgan Kaufman Publishers, 1993.

[16] B. D. Ripley. *Pattern Recognition and Neural Networks*. Cambridge University Press, Cambridge, UK, 1996.

[17] Y. Sun, M. S. Kamel, and Y. Wang. Boosting for learning multiple classes with imbalanced class distribution. In *Proceedings of the Sixth International Conference on Data Mining*, pages 592–602, 2006.

[18] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, New York,USA, 1995.

[19] I. H. Witten and E. Frank. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, San Francisco, 2nd edition, 2005.

[20] J. Zhang, E. Bloedorn, L. Rosen, and D. Venese. Learning rules from highly unbalanced data sets. In *Proceedings of the Fourth IEEE International Conference on Data Mining (ICDM'04)*, 2004.