An Adaptive Diversity-Based Ensemble Method for Binary Classification

Xing Fan, Chung-Horng Lung, Samuel A. Ajila
Department of Systems and Computer Engineering
Carleton University
1125 Colonel By Drive, Ottawa K1S 5B6, Ontario Canada
XingFan@cmail.carleton.ca; {chlung; ajila}@sce.carleton.ca

***Abstract*— This paper proposes a novel ensemble method to improve the performance of binary classification. The proposed method is a non-linear combination of base models and an application of adaptive selection of the most suitable model for each data instance. Ensemble methods, an important type of machine learning technique, have drawn a lot of attention in both academic research and practical applications, and they use multiple single models to construct a hybrid model. A hybrid model generally performs better compared to a single individual model. The proposed approach in this paper based on a hybrid model has been validated on Repeat Buyers Prediction dataset, and the experiment results show up to 18.5% improvement on F_1 score, compared to the best individual model. In addition, the proposed method outperforms two other commonly used ensemble methods (Averaging and Stacking) in terms of improved F_1 score.**

Keywords—Adaptive method; Ensemble method; Binary classification; ROC curve; F_1 score; AdaBoost; Xgboost; Logistic regression;

I. INTRODUCTION

In supervised learning techniques, ensemble learning method is the technique that uses multiple single models to construct a hybrid model in order to achieve better performance compared to that of using a single model [1]. A workflow for solving classification problems by applying ensemble methods consists of the following. First, raw data usually need to be pre-processed for initializing a training dataset, during which feature extraction and normalization are applied. Second, training sets for each individual single model are derived from the initialized dataset. Third, single models are trained from different training datasets or by different algorithms. Finally, all the single models are combined to construct the ensemble and then the final [hybrid] model is constructed based on the results of individual models and the hybrid model is further validated and tested.

There ensemble learning techniques could be categorized into two types: *Type I* techniques focus on deriving new training sets from the initial training set to train multiple different single models. *Type II* techniques focus on finding ways to blend the individual models [1]. In this paper, we are interested in *Type II* techniques as method for improving predictive performance in binary classification problems [2]. The most popular *Type II* ensemble techniques assign weights to all trained single models and then linearly combine them [3], [4]. Thus, for all unknown instances, the contribution of each individual model to the final prediction is fixed, which may limit performance. Therefore, dynamic adjustment of each single model's contribution for different instances is helpful and it requires methods based on non-linear blending. Moreover, the existing *Type II* ensemble methods take all the outputs of the combined single models into account, which means that the weaknesses of the single models are also kept in the hybrid model. If an ensemble method can adaptively select the most suitable single model to predict each instance, then the weaknesses of the single models can be avoided. In order to maximize the complementation effect, only those single models with the highest pairwise diversities should be selected to construct the ensemble model.

The objectives of this research are therefore to find an ensemble model that can: (i) select base models that are based on their pairwise diversities; (ii) recognize the best suitable base model for each data instance; and (iii) predict each unknown instance using a suitable base model. The contributions of this paper are: (i) a novel ensemble method for solving binary classification problem; the pairwise diversities of single classifiers are measured using two different methods and the results of these measurements are used to select and combine the base classifiers; (ii) multiple machine learning algorithms are used to train different base models; and (iii) the proposed ensemble method is validated using the Repeat Buyer Prediction Competition [5].

The remainder of this paper is organized as follows: Section II discusses the background and related work. Section III describes the proposed ensemble method. Section IV presents an analysis of the experimental results and finally, conclusion and future directions are discussed in Section V.

II. BACKGROUND AND RELATED WORK

Machine learning is divided into two main types: supervise and unsupervised learning. Other types such as semi-supervise learning and reinforcement learning can be derived from the two main types [6], [7]. Supervised learning problems fall into two categories – classification and regression. The outputs of classification are discrete while that of regression is continuous. In this paper, the proposed ensemble method is for solving classification problems.

0730-3157/17 $31.00 © 2017 IEEE
DOI 10.1109/COMPSAC.2017.49

Classification problems can be categorized into two types: binary classification and multiclass classification [8]. The former identifies instances as one of the two pre-defined classes, whereas the latter classifies instances into one of the more than two classes. The research work in this paper focuses on binary classification.

A. Binary Classifiers and Confusion Matrix

Formally, a classifier is a derived function that maps instances to targets, of which all parameters are determined. A machine learning algorithm is a process that estimates the parameters of the function by learning the train data, so that the classifier could fit the train data. A binary classifier is a function that maps instances to positive or negative class label [9]. According to the types of outcomes, classifiers could be recognized as a discrete classifier or probabilistic classifier. A discrete classifier directly generates a predicted class label whereas a probabilistic classifier assigns each instance a score to indicate its degree of confidence of belonging to one class [10]. A relative threshold is assigned to probabilistic classifiers to determine the predicted classes of instances. The instances with a score higher than the threshold is considered as positive, while those with a score lower than the threshold is classified as negative.

Confusion Matrix is generally used for presenting and interpreting the performance of a classifier on a dataset. It is an n×n matrix (n represents the number of classes) constructed from the Cartesian product of actual classes and predicted classes [11]. Each entry of the matrix indicates the number of instances in each ordered pair of the Cartesian product. For binary classification, n equals to 2, therefore represents the two classes of Positive and Negative. The Precision (PPV) and Recall (TPR) [11] are two universal metrics describing the performance of a classifier on a dataset. However, it is worth mentioning that the accuracy may not be an adequate measure of performance because it is very sensitive to class distribution of the given dataset [12]. Therefore, in this paper, Area Under the Receiver Operating Characteristic Curve (AUC) and F1 score are selected as the metrics to evaluate the performance of the proposed ensemble method. A ROC curve, is a plot that visualizes the performance of a binary classifier. It is drawn in a 1× 1 square area called ROC space, of which x-axis and y-axis are defined as False Positive Rate (FPR) and True Positive Rate (TPR), respectively [12]. Therefore, the plots in ROC space describe the trade-off between the benefit (TPR) and the cost (FPR) of a classifier. For each discrete classifier on a given dataset, there is only one corresponding confusion matrix, which is plotted as a single point in ROC space [12]. On the other hand, probabilistic classifiers could yield different confusion matrixes as threshold varies, so that multiple points are plotted in ROC space [13].

F_1 score [14] is a commonly used metric to evaluate the performance of a classifier. F_1 score is defined as the harmonic mean of precision and recall:

$$F_1 = 2 \times [(\text{precision} * \text{recall}) / (\text{precision} + \text{recall})]$$

$$= 2 \times [(\text{PPV} * \text{TPR}) / (\text{PPV} + \text{TPR})]$$

Precision denotes the degree of confidence for a classifier to predict an instance as positive, while recall indicates the ability of a classifier to identify the positive instances. The best value of F_1 score is 1 and the worst is 0. F_1 score conveys the balance between precision and recall.

B. Classification Learning Algorithms

In this work, the following factors are taken into account when selecting a machine learning algorithm: (i) The algorithm should have the ability to learn a probabilistic classifier because of AUC and F_1 score. (ii) The algorithm should be capable to learn sparse data since the given dataset is under high sparsity. (iii) The selected algorithm is simple so that the parameter tuning for the ensemble models could be easier. (iv)The algorithm should achieve a proper balance between the performance of the single model and its training time as well as its resource requirement. So, with this in mind we have selected Logistic Regression under the general regression model (GRM), and for the ensemble, we have selected Boosting, Bagging, and Stacked Generalization.

Logistic regression, a generalization of linear regression can learn a model and it is a function mapping an instance to a predicted class [7]. The core concept of Bagging is to train multiple base classifiers from different new training sets that are randomly sampled from the original train data with replacement. In this paper, Random Forests provided by Scikit-learn is used. Boosting is a family of ensemble learning algorithms that combines a set of weak learners to construct a relatively strong one. Adaptive Boosting (AdaBoost) and Gradient Boosting Machine (GBM) are the two most popular Boosting algorithms [18], [19]. In this paper, the implementations of AdaBoost and an GBM called Extreme Gradient Boosting (Xgboost) provided by Scikit-learn are used. Stacked Generalization, which is also known as Stacking (proposed in [21]), a high level ensemble method that introduces a meta-level model to combine a group of base models, which are usually of diverse types.

C. Related Works

Ensemble methods such as Bagging, Boosting and Stacking which combine the decisions of multiple hypotheses are some of the strongest existing ensemble methods and therefore provide a reliable foundation for the proposed ensemble method. The work in [22] applied Random Forests to classification of hyperspectral data on the basis of a binary hierarchical multi-classifier system. The work in [23] presented an algorithm for musical style and artist prediction from an audio waveform by utilizing AdaBoost for the selection and aggregation of audio features. Gradient Boosting Machine (GBM) was applied in [24] for incorporating diverse measurements of bone density and geometry so as to improve the accuracy of bone fracture prediction compared to standard measurement. The work in [25] examined the generalization behavior by comparing single level learning models to multiple level learning models (stacked generalization method) on a multilayer neural networks. Result shows that the stacked generalization scheme could improve

classification performance and accuracy compared to the single level model.

There are several schemes proposed for the customized selection of classifiers. For example, the work in [26] proposed the method of Dynamic Integration, in which weighted voting were applied by giving higher weight to a classifier if its training data were in the same region as the testing example. The work in [27] introduced an approach to group classifiers by their similarities and to retain one representative classifier per cluster. The work in [28] proposed a linear ensemble algorithm that takes into account both the accuracy of individual classifiers and the diversity among classifiers.

The dataset used in this work is from the Repeat Buyers Prediction Competition (RBP) [5]. In this two-stage competition with different amount of data provided, the work by Liu [29] won the first place in Stage 1 by using both feature engineering and model training. The work by He [30] won the first place in Stage 2, in which a four-step solution was introduced (i) characteristics analysis and strategy design, (ii) feature extraction and selection, (iii) data training, and (iv) a hybrid ensemble on both models and features. Both of the award winners suggested that feature engineering is the key element to their works and that the ensemble method applied in these two approaches perform better than any of the individual classifier. However, only linear ensemble methods were utilized in their works. The goal of our approach is to use non-linear method.

III. The Proposed Ensemble System

To solve classification problems, a classifier is learnt by an algorithm, so that a hypothesis can be proposed based on the classifier that best fits the given training set. For a dataset, a single classifier may have its own "blind area", i.e., some points cannot be clearly identified, especially when the dataset is high-dimensional or non-linear.

In the proposed ensemble method, a meta-level model is learnt, which recognizes the capable base classifier for each data instance. In addition to the meta-level model from all trained single classifiers, only those with high pairwise diversities are selected as base models. Finally, the constructed ensemble model is able to support adaptive selection of the proper base model for each data instance. In order to accomplish this, two key problems demand prompt solutions: (i) the pairwise diversity needs to be defined for measurement; (ii) a training set needs to be designed for learning the meta-level model.

A. System Design

The system consists of seven modules: Pre-processor, Relabeler, Base Model Selector, Single Model Learner, Meta Model Learner, Meta Model and Base Models, which are organized as shown in Fig. 1. Each module is briefly described as follows.

- The Pre-processor is responsible for data transformation, data normalization and feature engineering, as well as splitting the known data into a training set and a testing set.
- The Single Model Learner learns single classifiers from the training set with the original labels for data instances.
- The Base Model Selector takes both the testing set and the outputs of the Single Model Learner as inputs, to determine the Base Models for constructing the ensemble.
- The Relabeler receives the training set with original labels for data instances, and then relabels the training set with the outputs of the Base Model Selector. The relabeling of each instance is performed according to the behaviors of Base Models on the original training set.
- The Meta Model Learner uses the relabeled training set to train the meta model.
- The Meta Model takes the feature vector of each unknown instance to predict its most suitable base model, and then passes the feature vector to the selected Base Model for the final prediction, which is the output of the system

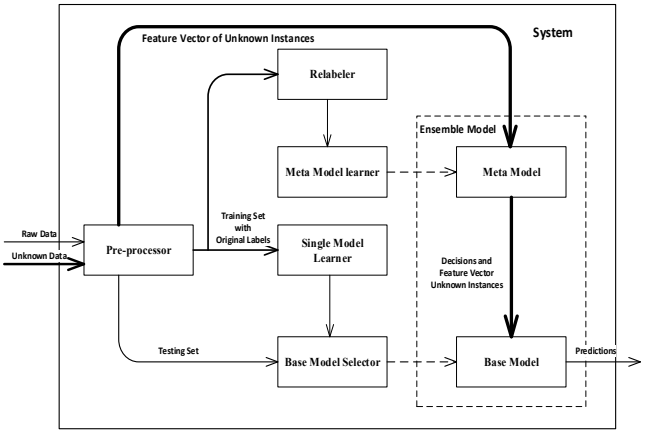

Fig. 1. Proposed system's architecture

B. Base model Selector and Re-labeler

The Base Model Selector is the module that determines the base models for constructing the final ensemble model. The pairwise diversities are calculated from the ranks, and the algorithm for determining the single models based on diversities is described as follows:

Algorithm 1: Base Model Selection

1. Given a testing set $S = \{s_i | i = 1, 2 \dots n\}$ and a group of trained single classifiers $C = \{c_j | j = 1, 2 \dots m\}$, where n and m denote the number of samples in testing set and the number of trained single classifiers, respectively;
2. **For** each classifier c_j in C:
   a. Use c_j to make predictions for each s_i in S to obtain a set of probabilities $P_j = \{p_{ij} | i = 1, 2 \dots n\}$, where p_{ij} represents the probability of s_i being positive generated by c_j;
   b. Sort all s_i according to their p_{ij} in ascending order, and then obtain a set of ranks $R_j = \{r_{ij} | i = 1, 2 \dots n\}$, where r_{ij} denotes the rank of s_i in all ordered test samples given by c_j;
3. **End For**

4. Calculate the root-mean-square deviation (RMSD) of each set of two R_j as the metric of pairwise diversity $D_{pq} = \sqrt{\frac{\sum_{i=1}^{n}(r_{ip}-r_{iq})^2}{n}}$;
5. Choose the k single classifiers c_j with highest RMSDs as the base models for constructing the ensemble.

The function of Re-labeler is to design a new training set for training the meta-level model. The aim is to identify the proper base model for a given instance. A training set consists of two parts: feature vectors and labels. The former describes the characteristics of data instances, while the latter indicates the class of instances in a certain class space. Accordingly, the algorithm of relabeling is designed as follows:

Algorithm 2: Relabeling Training Set for Meta Model

1. Given a training set $S = \{s_i | i = 1, 2 \dots n\}$ obtained from pre-processor and a group of base classifiers $B = \{b_j | j = 1, 2 \dots m\}$ determined by base model selector;
2. **For** each base classifier b_j in B:
   a. Use b_j to make predictions for each s_i in S to obtain a set of probabilities $P_j = \{p_{ij} | i = 1, 2 \dots n\}$, where p_{ij} represents the probability of s_i being positive generated by b_j;
   b. Sort all s_i according to their p_{ij} in ascent order, and then obtain a set of ranks $R_j = \{r_{ij} | i = 1, 2 \dots n\}$, where r_{ij} denotes the rank of s_i in all ordered samples given by b_j;
3. **End For**
4. **For** each training sample s_i in S:
5. **If** s_i is positive:
   a. relabel s_i as b_j which generates the largest r_{ij};
6. **End If**
7. **If** s_i is negative:
   a. relabel s_i as b_j which generates the smallest r_{ij};
8. **End If**
9. **End For**
10. **Return** the relabeled training set

The training set derived from Algorithm 2 represents the proper base model for each instance, and then it is further used for training meta-level model, so that the meta-level model can be used to predict the appropriate base models for unknown instances.

IV. Experiment And Results

The dataset is the Repeat Buyer Prediction (RBP) dataset, obtained from a machine learning competition, which was held by International Joint Conference on Artificial Intelligence (IJCAI) and Alibaba Group in 2015 [5]. The repeat buyer dataset consists of the behavior log of anonymized users accumulated during the 6 months before and on the "Double 11" day, with labels indicating whether a customer is a repeat buyer of a merchant or not. The dataset is 1.92 GB in total and stored as three comma-separated values (CSV) files: user behavior logs, user profile information, and training and testing. All the developments and experiments are conducted on a customized Google Cloud Platform compute using Ubuntu 16.04 LTS with 6 cores (2.3GHz) virtualized from Haswell processors, 32 GB memory and 128 GB SSD.

A. The Procedure

The dataset is preprocessed using Pandas. Then feature engineering is applied. After single models are trained using Logistic Regression, Gradient Boosting Machine, AdaBoost, Random Forests and Factorization Machine, then, they are tested on the testing set for evaluations. The final ensemble model is constructed based on the trained single models summarized as follows: determine base models; relabel the training set and learn the meta-level model; and use meta-level model to combine the selected base models. After, the training set that was used for training single models was relabeled, so that a meta-level classifier can be trained on this relabeled training set. Similar to calculating the pairwise diversities, the new labels are also determined based on ranking and base models are applied on the training set rather than the testing set. For simplicity, it is assumed that the number of base models is two, and the two base models are logistic regression and GBM (lr and xgb). Therefore, the new class space consists of two class labels (lr and xgb), and the trained meta-level model is a binary classifier.

The last step of the ensemble model construction is utilizing the trained meta-level model to combine the base models.

B. Experiment Design

The experiments are designed to evaluate the performance (using AUC and F1 score) and the overhead that is introduced to the system in terms of time consumption. In addition, both single models and ensemble models are taken into account for evaluation. For each single model, the changing of performance is traced as its parameters vary. Multiple parameters are selected for each single model, and only one of them is tuned each time with other parameters unchanged. The combination of a set of parameters leading to a relatively best performance for each single model is considered as the best parameters of the single model. Then single models with best parameters are compared horizontally.

Four machine learning algorithms (Logistic Regression, AdaBoost, Xgboost and Random Forests) are selected to train single models, and the list of tuned parameters for these various single models is as follows: Inverse of regularization strength, the maximum number of iterations for Logistic Regression; the number of constructed decision trees (AdaBoost, Xgboost, Random Forests); a rate indicating the learning rate of constructed decision trees, the learning rate of loss function solver; and the maximum tree depth of the decision tree.

C. Analysis and Results

Three methods of base model selection are evaluated and compared, and they are: RMSD (Root-mean-square derivation) diversity-based; ZOL (Zero-one loss) diversity-based; and Performance-based (AUC and F1 score).

In addition, three other ensemble methods are used in the paper for comparison with the proposed ensemble method: Averaging, Stacking with Logistic Regression (lr) and Stacking with Xgboost (xgb).

Fig. 2 shows the result of the AdaBoost performance for the RBP dataset, from which no direct connection was found between the value of AUC and F_1 score. For this dataset, as the value of *n_estimator* grows, both AUC and F_1 score increased at first and then decreased, but with the different inflection point.

Fig. 3 shows the performance of Logistic Regression on RBP dataset. A relatively smaller C (stronger regularization) indicates a very strict penalty on the error instance when learning the classifier. It is observed that a smaller C has benefits for both AUC and F1 score. On the other hand, as *max_iter* increases, a growth in performance is observed to certain extent.

Fig. 4 shows the performance of Xgboost on the dataset where a relatively lower learning rate is preferred. Both *n_estimator* and *max_depth* represent the complexity of a model, in which *max_depth* indicates the maximum allowable depth of each tree. Normally, a model with relatively lower complexity could benefit AUC. On the other hand, a model with relatively higher complexity could increase F_1 score.

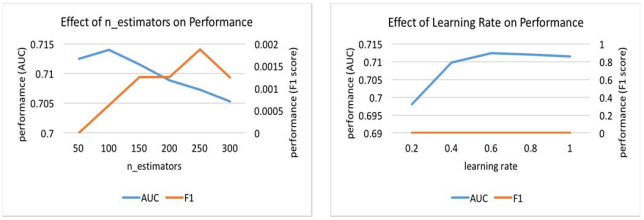

Fig. 2. Result of AdaBoost performance in RBP

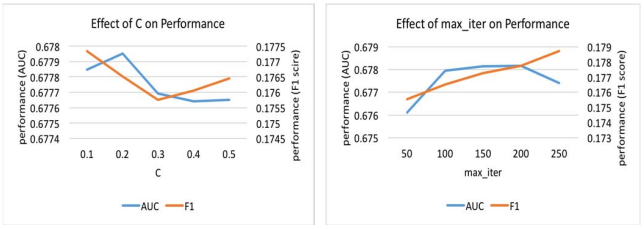

Fig. 3. Result of Logistic Regression performance in RBP

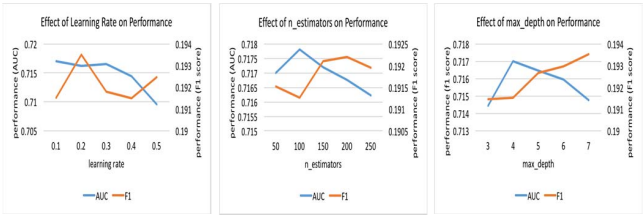

Fig. 4. Result of Xgboost performance in RBP

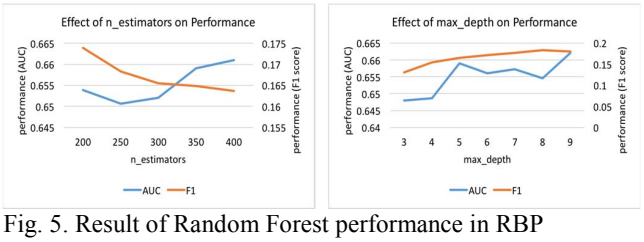

Fig. 5. Result of Random Forest performance in RBP

Fig. 5 presents the performance of Random Forest. It is observed that the increase of *n_estimator* helps improve AUC. On the contrary, higher F_1 scores were mostly acquired among low *n_estimator* values. Increasing *max_depth* has positive impact on both AUC and F_1 score for the dataset. In summary, both the AUC and F_1 score results obtained from Xgboost are better than that of logistic regression, and AdaBoost does not perform well on F_1 score.

The best performance results (AUC and F_1 score) for each single model are selected for a comparison. Fig. 6 depicts the best results for those single models. Xgboost performs the best for both AUC and F_1 score on RBP dataset. Fig. 7 shows the performance comparison among different ensemble methods. The averaging of base models generates a little higher AUC performance result than the best base model (Logistic Regression). The model from the proposed method based on RMSD diversity also achieves a slightly higher performance than that of the best base model (but slightly lower than averaging). However, Stacking (lr) and Stacking (xgb) failed to further enhance AUC. In addition, all these four ensemble methods with other two base models (ZOL and best performance) did not contribute to increase AUC.

Fig. 6. Comparison of single model's best performance on RBP

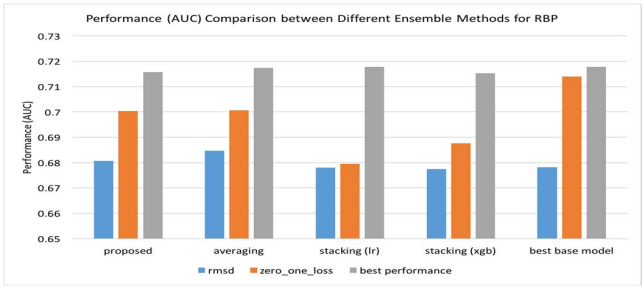

Fig. 7. AUC comparison of different ensemble methods

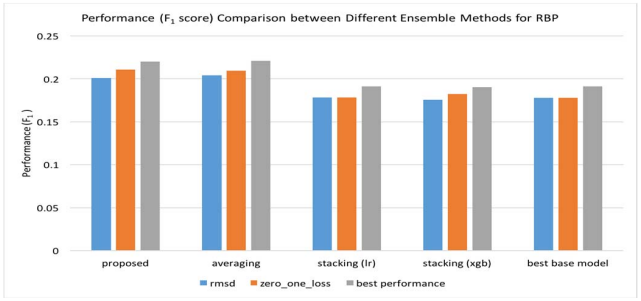

Fig. 8. F1 score comparison of different ensemble methods

As shown in Fig. 8 the proposed method and averaging increase the F_1 score with all three base models compared to best base model. The combination of two single models with highest ZOL diversity leads to the highest improvement on F_1 score for the proposed method (18.5%).

D. Overhead

Table 1 shows the computational overhead of the ensemble methods for the dataset. It can be observed that the proposed ensemble method introduced higher overhead in both relabel and meta-level model training, but the time consumption is still within an acceptable range.

Table 1. Ensemble Overhead in RBP

Base Model	Proposed Method		Stacking (lr)	Stacking (xgb)
	Relabel	Meta Model Training		
lr & rf	6min 51s	3min 8s	444ms	1.39s
lr & ab	5min 41s	3min 8s	418ms	1.43s
ab & xgb	6min 28s	3min 10s	488ms	1.29s

V. Conclusions And Future Work

In this paper, a novel ensemble method is proposed, which is capable of performing adaptive selection of the best base model for each unknown instance. The proposed method is validated on RBP dataset, and the performance of AUC and F_1 score was compared with those of other two existing ensemble techniques (Averaging and Stacking).

The proposed ensemble method has an overall performance improvement in terms of F_1 score, which is considered a more valuable metric in practice. There is a significant performance improvement of 18.5% using the proposed ensemble method compared with the best base model. This improvement indicates that the proposed ensemble method has an exceptional ability of dealing with imbalanced datasets. Moreover, a higher pairwise diversity of combined base models can lead to a further improvement toward the performance of ensemble model.

The limitations in this work include the following. The meta-level model is trained as a binary classifier, thus, only two base models can be combined. Furthermore, the combination of base models is determined on the basis of predicted probabilities from the meta-level model.

References

[1] Opitz, David, and Richard Maclin. "Popular ensemble methods: An empirical study." Journal of Artificial Intelligence Research 11 (1999): 169-198.

[2] Kuncheva, Ludmila I. "Diversity in multiple classifier systems." Information Fusion 6.1 (2005): 3-4.

[3] Canuto, Anne MP, et al. "Performance and diversity evaluation in hybrid and non-hybrid structures of ensembles." Fifth International Conference on Hybrid Intelligent Systems. IEEE, (2005).

[4] Canuto, Anne MP, et al. "Using weighted dynamic classifier selection methods in ensembles with different levels of diversity." International Journal of Hybrid Intelligent Systems 3.3 (2006): 147-158.

[5] Repeat Buyers Prediction Competition. [online] Ijcai-15.org. Available at: http://ijcai-15.org/index.php/repeat-buyers-prediction-competition [Accessed 10 Sep. 2016].

[6] Archive.ics.uci.edu. (2016). UCI Machine Learning Repository. [online] Available at: http://archive.ics.uci.edu/ml/ [Accessed 16 Sep. 2016].

[7] Barber, David. "Bayesian reasoning and machine learning." Cambridge University Press, (2012).

[8] Har-Peled, Sariel, Dan Roth, and Dav Zimak. "Constraint classification: A new approach to multiclass classification." International Conference on Algorithmic Learning Theory. Springer Berlin Heidelberg, (2002).

[9] Alpaydin, Ethem. "Introduction to Machine Learning." (2010): 249-256.

[10] Friedman, Jerome H. "Greedy function approximation: a gradient boosting machine." Annals of Statistics (2001): 1189-1232.

[11] Kohavi, Ron, and Foster Provost. "Glossary of terms." Machine Learning 30.2-3 (1998): 271-274.

[12] Fawcett, Tom. "An introduction to ROC analysis." Pattern Recognition letters 27.8 (2006): 861-874.

[13] Wikipedia. (2016). Receiver operating characteristic. [online] Available at: https://en.wikipedia.org/wiki/Receiver_operating_characteristic [Accessed 16 Sep. 2016].

[14] Cohen, William, Pradeep Ravikumar, and Stephen Fienberg. "A comparison of string metrics for matching names and records." Kdd Workshop on Data Cleaning and Object Consolidation. Vol. 3. (2003).

[15] Davis, Jesse, and Mark Goadrich. "The relationship between Precision-Recall and ROC curves." Proceedings of the 23rd International Conference on Machine Learning. ACM, (2006).

[16] Breiman, Leo. "Bagging predictors." Machine Learning 24.2 (1996): 123-140.

[17] Freund, Yoav, et al. "Using and combining predictors that specialize." Proceedings of the Twenty-ninth Annual ACM Symposium on Theory of Computing. ACM, (1997).

[18] [10] Friedman, Nir, Dan Geiger, and Moises Goldszmidt. "Bayesian network classifiers." Machine Learning 29.2-3 (1997): 131-163.

[19] Chen, Tianqi, and Carlos Guestrin. "Xgboost: A scalable tree boosting system." (2016).

[20] Wolpert, David H. "Stacked generalization." Neural Networks 5.2 (1992): 241-259.

[21] Ham, Jisoo, et al. "Investigation of the random forest framework for classification of hyperspectral data." IEEE Transactions on Geoscience and Remote Sensing 43.3 (2005): 492-501.

[22] Bergstra, James, et al. "Aggregate features and AdaBoost for music classification." Machine Learning 65.2-3 (2006): 473-484.

[23] Atkinson, Elizabeth J., et al. "Assessing fracture risk using gradient boosting machine (GBM) models." Journal of Bone and Mineral Research 27.6 (2012): 1397-1404.

[24] Ghorbani, Ali A., and Kiarash Owrangh. "Stacked generalization in neural networks: generalization on statistically neutral problems." Proceedings of the International Joint Conference Neural Networks. IEEE, (2001).

[25] Wang, Shuang-Quan, Jie Yang, and Kuo-Chen Chou. "Using stacked generalization to predict membrane protein types based on pseudo-amino acid composition." Journal of Theoretical Biology 242.4 (2006): 941-946.

[26] Tsymbal, Alexey, et al. "Dynamic integration of classifiers for handling concept drift." Information Fusion 9.1 (2008): 56-68.

[27] Bhatnagar, Vasudha, et al. "Accuracy–diversity based pruning of classifier ensembles." Progress in Artificial Intelligence 2.2-3 (2014): 97-111.

[28] Giacinto, Giorgio, and Fabio Roli. "Dynamic classifier selection based on multiple classifier behaviour." Pattern Recognition 34.9 (2001): 1879-1881.

[29] Liu, G, et al. "Report for Repeated Buyer Prediction Competition by Team 9*TAR." In Proceedings of the 1st International Workshop on Social Influence Analysis Soclnf 2015.

[30] He, B, et al. "Repeat Buyers Prediction after Sales Promotion for Tmall Platform." In Proceedings of the 1st International Workshop on Social Influence Analysis Soclnf 2015.