An Evolutionary Programming Based SVM Ensemble Model for Corporate Failure Prediction

Lean Yu[1,2], Kin Keung Lai[2,3], and Shouyang Wang[1,2]

[1] Institute of Systems Science, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100080, China
{yulean,sywang}@amss.ac.cn
[2] College of Business Administration, Hunan University, Changsha 410082, China
[3] Department of Management Sciences, City University of Hong Kong, Tat Chee Avenue, Kowloon, Hong Kong
{msyulean,mskklai}@cityu.edu.hk

Abstract. In this study, a multistage evolutionary programming (EP) based support vector machine (SVM) ensemble model is proposed for designing a corporate bankruptcy prediction system to discriminate healthful firms from bad ones. In the proposed model, a bagging sampling technique is first used to generate different training sets. Based on the different training sets, some different SVM models with different parameters are then trained to formulate different classifiers. Finally, these different SVM classifiers are aggregated into an ensemble output using an EP approach. For illustration, the proposed SVM ensemble model is applied to a real-world corporate failure prediction problem.

1 Introduction

Ensemble learning has been turned out to be an efficient way to achieve high prediction/classification performance, especially in fields where the development of a powerful single classifier system requires considerable efforts [1]. According to Olmeda and Fernandez [2], an optimal system may not be an individual model but the combination of several of them from a decision support system (DSS) perspective. Usually, ensemble model outperforms the individual models, whose performance is limited by the imperfection of feature extraction, learning/classification algorithms, and the inadequacy of training data. Another reason supporting this argument is that different individual models have their inherent drawbacks and thus aggregating them may lead to a good classifier with high generalization capability. From the above descriptions, we can conclude that there are two essential requirements to the ensemble members and the ensemble strategy. The first is that the ensemble members must be diverse or complementary, i.e., classifiers must show different classification properties. Another condition is that an optimal ensemble strategy is also required to fuse a set of complementary classifiers [1].

To achieve high performance, this study utilizes a new machine learning tool — support vector machine (SVM) first proposed by Vapnik [3] — as a generic model for ensemble learning. The main reasons of selecting SVM as ensemble learning tool reflect the following aspects. First of all, SVM requires less prior assumptions about the input data, such as normal distribution and continuousness, different from statistical

B. Beliczynski et al. (Eds.): ICANNGA 2007, Part II, LNCS 4432, pp. 262 – 270, 2007.

models. Second, they can perform a nonlinear mapping from an original input space into a high dimensional feature space, in which it constructs a linear discriminant function to replace the nonlinear function in the original low dimension input space. This character also solves the dimension disaster problem because its computational complexity is not dependent on the sample dimension. Third, they attempt to learn the separating hyperplane to maximize the margin, therefore implementing structural risk minimization and realizing good generalization ability. This pattern can directly help SVM escape local minima and avoid overfitting problem, which are often shown in the training of artificial neural networks (ANN) [3]. These important characteristics will also make SVM popular in many practical applications.

The basic procedure of using the SVM to construct an ensemble classifier consists of three stages. In the first stage, an initial dataset is transformed into some different training sets by certain sampling algorithms. In this study, a bagging sampling approach [4] is used to generate different training datasets. In the second stage, the SVM models are trained by various training datasets from the previous stage to formulate some generic classifiers with different classification properties. Because different training datasets have different information, the generic classifiers produced by these different datasets should be diverse in terms of some previous empirical analysis [1, 5-6]. In the final stage, these different SVM classifiers are aggregated into an ensemble output using an integration approach. In this study, we use classification accuracy maximization principle to construct an optimal ensemble classifier. Particularly, an evolutionary programming (EP) algorithm [7] is used to solve the maximization problem. For testing purpose, a real-world corporate bankruptcy dataset are used to verify the effectiveness of the proposed SVM ensemble model.

The main motivation of this study is to design a high-performance classifier for corporate failure prediction and compare its performance with other existing approaches. The rest of the study is organized as follows. The next section presents a formulation process of a multistage SVM ensemble model in detail. For illustration and verification purposes, a practical experiment is performed and corresponding results are reported in Section 3. And Section 4 concludes the study.

2 Methodology Formulation Process

In this section, a triple-stage SVM ensemble model is proposed for classification. The basic idea of SVM ensemble originated from using all the valuable information hidden in all individual SVM classifiers, where each can contribute to the improvement of generalization. In our proposed SVM ensemble model, a bagging sampling approach is first used to generate different training sets for guaranteeing enough training data. Using these different training datasets, multiple individual SVM classifiers can be then formulated as ensemble members or components. Finally, all ensemble members are aggregated into an ensemble output.

2.1 Stage I: Data Sampling

Data sampling is one of the most important steps in designing an ensemble model. This step is necessary and crucial for many reasons, most importantly to determine if

the training set at hand is functionally or structurally divisible into some distinct training sets. In this study, the bootstrap aggregating (**bagging**) proposed by Breiman [4] is utilized as data sampling tool.

Bagging is a widely used data sampling method in the machine learning. Given that the size of the original data set *DS* is *P*, the size of new training data is *N*, and the number of new training data items is *m*, the bagging algorithm of generate new training subsets can be shown in Fig. 1.

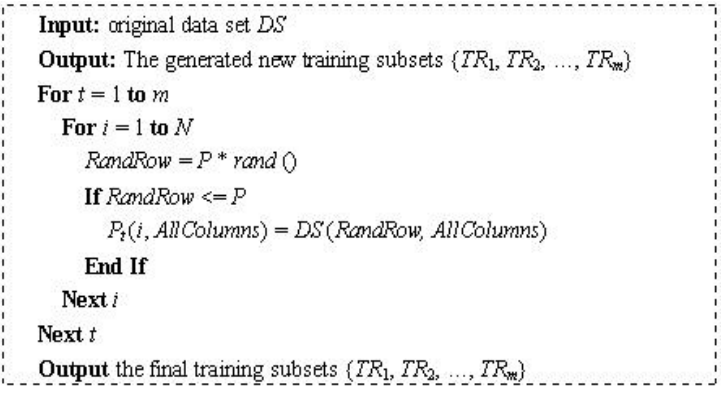

Fig. 1. The bagging algorithm

The bagging algorithm is very efficient in constructing a reasonable size of training set when the size of the original data set is small due to the feature of its random sampling with replacement. Therefore, bagging is a useful data sampling method for machine learning. In this study, we use the bagging to generate different training sets.

2.2 Stage II: Individual SVM Classifiers Creation

According to the definition of effective ensemble classifiers by Hansen and Salamon [8], 'a necessary and sufficient condition for an ensemble of classifiers to be more accurate than any of its individual members is if the classifiers are accurate and diverse.' That is, an effective ensemble classifier consisting of diverse models with much disagreement is more likely to have a good generalization performance. Therefore, how to generate the diverse model is a crucial factor. For the SVM model, several methods can be used to create different ensemble members. Such methods basically rely on varying the parameters related to the design and to the training of SVM models. In particular, some main ways include the following aspects:

(i) Using different kernel functions in SVM model. For example, polynomial function and Gaussian function are typical kernel functions.

(ii) Changing the SVM model parameters. Typically, margin parameter *C* and kernel parameter σ^2 are usually considered.

(iii) Varying training data sets. Because different datasets contains different information, different datasets can generate diverse model with different model parameters.

In our study, the third way is selected because the previous phase creates many different training datasets. With these different training datasets, diverse SVM classifiers with disagreement as ensemble members can be generated. Interested readers can be referred to Vapnik [3] for more details about SVM classification.

2.3 Stage III: Ensemble Members Aggregation

When individual SVM classifiers are generated, each classifier can output its own results in terms of testing set. Before integrating these ensemble members, strategies of selecting ensemble members must be noted. Generally, these strategies can be divided into two categories: (i) generating an exact number of ensemble members; and (ii) overproducing ensemble members and then selected a subset of these [9].

For the first strategy, several common ensemble approaches, e.g., boosting [10], can be employed to generate the exact number of diverse ensemble members for integration purpose. Therefore, no selection process will be used and all generated ensemble members will be combined into an aggregated output. For the second strategy, its main aim is to create a large set of ensemble candidates and then choose some most diverse members for integration. The selection criterion is some error diversity measures, which is introduced in detail by Partridge and Yates [11]. Because the first strategy is based upon the idea of creating diverse neural networks at the early stage of design, it is better than the second one, especially for some situations where access to powerful computing resources is restricted. The main reason is that the second strategy cannot avoid occupying much computing time and storage while creating a large number of ensemble candidates, some of which are to be later discarded.

Actually, a simple way to take into account different opinions is to take the vote of the majority of the population of classifiers. In the existing literature, majority voting is the most widely used ensemble strategy for classification problems due to its easy implementation. Ensemble members' voting determines the final decision. Usually, it takes over half the ensemble to agree a result for it to be accepted as the final output of the ensemble regardless of the diversity and accuracy of each model's generalization. However, majority voting has several important shortcomings. First of all, it ignores the fact some classifiers that lie in a minority sometimes do produce the correct results. Second, if too many inefficient and uncorrelated classifiers are considered, the vote of the majority would lead to worse prediction than the ones obtained by using a single classifier. Third, it does not consider for their different expected performance when they are employed in particular circumstances, such as plausibility of outliers. At the stage of integration, it ignores the existence of diversity that is the motivation for ensembles. Finally, this method can not be used when the classes are continuous [2, 9]. For these reasons, an additive method that permits a continuous aggregation of predictions should be preferred. In this study, we propose an evolutionary programming (EP) [7] based approach to realize the classification/prediction accuracy maximization. The main reason of selecting EP rather than genetic algorithm (GA) is its ability to work with continuous parameters rather than binary coded independent variables. This makes the implementation of method easier and more accurate. Moreover, because of the self-adaptation mechanism in EP, global convergence is achieved faster compared to GA [12].

Suppose that we create p classifiers and let c_{ij} be the classification results that classifier j, j =1, 2, …, p makes of sample i, i = 1, 2, …, N. Without loss of generality, we assume there are only two classes (failed and non-failed firms) in the data samples, i.e., $c_{ij} \in \{0,1\}$ for all i, j. Let $C_i^w = Sign(\sum_{j=1}^{p} w_j c_{ij} - \theta)$ be the ensemble prediction of the data sample i, where w_j is the weight assigned to classifier j, θ is a confidence threshold and *sign*(.) is a sign function. For corporate failure prediction problem, an analyst can adjust the confidence threshold θ to change the final classification results. Only when the ensemble output is larger than the cutoff, the firm can be classified as good or healthful firm. Let $A_i(w)$ be the associated accuracy of classification:

$$A_i(w) = \begin{cases} a_1 & \text{if } C_i^w = 0 \text{ and } C_i^s = 0, \\ a_2 & \text{if } C_i^w = 1 \text{ and } C_i^s = 1, \\ 0 & \text{otherwise}. \end{cases} \tag{1}$$

where C_i^w is the classification result of the ensemble classifier, C_i^s is the actual observed class of data sample itself, a_1 and a_2 are the Type I and Type II accuracy, respectively, whose definitions can be referred to Lai et al. [1, 5-6].

The current problem is how to formulate an optimal combination of classifiers for ensemble prediction. A natural idea is to find the optimal combination of weights $w^* = (w_1^*, w_2^*, \cdots, w_p^*)$ by maximizing total classification accuracy including Type I and II accuracy. Usually, the classification accuracy can be estimated through k-fold cross-validation (CV) technique. With the principle of total classification accuracy maximization, the above problem can be summarized as an optimization problem:

$$(P) \begin{cases} \max\limits_{w} A(w) = \sum_{i=1}^{M} A_i(w) \\ \text{s.t. } C_i^w = sign\left(\sum_{j=1}^{p} w_j c_{ij} - \theta\right), i = 1,2,\cdots M \\ A_i(w) = \begin{cases} a_1 & \text{if } C_i^w = 0 \text{ and } C_i^s = 0, \\ a_2 & \text{if } C_i^w = 1 \text{ and } C_i^s = 1, \\ 0 & \text{otherwise}. \end{cases} \end{cases} \tag{2}$$

where M is the size of cross-validation set and other symbols are similar to the above notations.

Since the constraint C_i^w is a nonlinear threshold function and the $A_i(w)$ is a step function, the optimization methods assuming differentiability of the objective function may have some problems. Therefore the above problem cannot be solved with classical optimization methods. For this reason, an EP algorithm [7] is proposed to solve the optimization problem indicated in (2) because EP is a useful method of optimization when other techniques such as gradient descent or direct analytical method are impossible. For the above problem, the EP is described as follows:

(i) Create an initial set of L solution vectors $w_r = (w_{r1}, w_{r2}, \cdots, w_{rp}), r = 1,2,\cdots,L$ for above optimization problems by randomly sampling the interval [x, y], $x, y \in R$. Each population or individual w_r can be seen as a trial solution.

(ii) Evaluate the objective function of each of the vectors $A(w_r)$. Here $A(w_r)$ is called as the fitness of w_r.

(iii) Add a multivariate Gaussian vector $\Delta_r = N(0, G(A(w_r)))$ to the vector w_r to obtain $w_r' = w_r + \Delta_r$, where G is an appropriate monotone function. Re-evaluate $A(w_r')$. Here $G(A(w_r))$ is called as mutation rate and w_r' is called as an offspring of individual w_r.

(iv) Define $\overline{w_i} = w_i, \overline{w_{i+L}} = w_i', i = 1,2,\cdots,L, \overline{C} = \overline{w_i}, i = 1,2,\cdots,2L$. For every $\overline{w_j}$, $j = 1,2,\cdots,2L$, choose q vectors $\overline{w_*}$ from $\overline{C}$ at random. If $A(\overline{w_j}) > A(\overline{w_*})$, assign $\overline{w_j}$ as a "winner".

(v) Choose the L individuals with more number of "winners" $w_i^*, i = 1,2,\cdots,L$. If the stop criteria are not fulfilled, let $w_r = w_i^*, i = 1,2,\cdots,L$, generation = generation +1 and go to step 2.

Using this EP algorithm, an optimal combination, w^*, of classifiers that maximizes the total classification accuracy is formulated. To verify the effectiveness of the proposed optimal ensemble classifier, a real-world corporate failure dataset is used.

3 Experiment Analysis

The data used in this study is about UK firms from the Financial Analysis Made Easy (FAME) database which can be found in the Appendix of [13]. It contains 30 failed and 30 non-failed firms. 12 variables are used as the firms' characteristics description:

(1) Sales;
(2) ROCE: profit before tax/capital employed (%);
(3) FFTL: funds flow (earnings before interest, tax &depreciation)/total liabilities;
(4) GEAR: (current liabilities + long-term debt)/total assets;
(5) CLTA: current liabilities/total assets;
(6) CACL: current assets/current liabilities;
(7) QACL: (current assets)/current liabilities;
(8) WCTA: (current assets – current liabilities)/total assets;
(9) LAG: number of days between account year end and the date the annual report and accounts were failed at company registry;
(10) AGE: number of years the firm has been operating since incorporation date;
(11) CHAUD: coded 1 if changed auditor in previous three years, 0 otherwise;
(12) BIG6: coded 1 if company auditor is a Big6 auditor, 0 otherwise.

This study is to identify the two classes of corporate bankruptcy problem: failed and non-failed. They are categorized as "0" or "1" in the research data. "0" means failed firm and "1" represents non-failed one. In this empirical test, 40 firms are randomly drawn as the training sample. Due to the scarcity of data, we make the number of good firms equal to the number of bad firms in both the training and testing samples, so as to avoid the embarrassing situations that just two or three good (or bad, equally likely) firms in the testing sample. Thus the training sample includes 20 data of each class. Its aim is to minimize the effect of such factors as industry or size that in some cases can be very important. For the training sample, we do a fifteen-fold

cross validation (i.e, k=15) experiments to determine the best single model. Except from the above training sample, the testing sample was collected using a similar approach. The testing sample consists of 10 failed and 10 non-failed firms. The testing data is used to test results with the data that is not utilized to develop the model. The prediction performance is evaluated by the Type I accuracy, Type II accuracy and total accuracy [1, 5-6].

For constructing an EP-based SVM ensemble model, 20 training sets are generated by bagging algorithm. For each ensemble member, the kernel function is Gaussian function. Related parameters of Gaussian function are obtained by trail and error. In the process of integration, the initial solution vector is between zero and one. For training, the individual size is set to 100 and number of runs is 500. The mutation rate is determined by the Gaussian function, as shown in the previous section. Meantime, the study compares the prediction performance with several commonly used models, such as linear discriminant analysis (LDA), logit regression analysis (LogR), artificial neural network (ANN) and single SVM model. For the ANN models, a three-layer back-propagation neural network (BPNN) with 25 TANSIG neurons in the hidden layer and one PURELIN neuron in the output layer is used. The network training function is the TRAINLM. Besides, the learning rate and momentum rate is set to 0.15 and 0.25. The accepted average squared error is 0.005 and the training epochs are 2000. In the single SVM, the kernel function is Gaussian function with regularization parameter C = 40 and σ^2=10. The above parameters of ANN and SVM are obtained by trial and error using cross-validation techniques.

The all classification results are reported in Table 1. Note that the results reported in Table 1 are the average of fifteen-fold cross-validation experiments and the values in bracket are standard deviations of fifteen-fold cross-validation experiments

Table 1. The results of SVM ensemble and its comparisons with other classifiers

Method	Type I (%)	Type II (%)	Overall (%)
LDA	67.67 [9.23]	71.33 [7.67]	69.50 [8.55]
LogR	72.67 [6.78]	75.00 [7.56]	73.83 [7.15]
ANN	70.67 [8.16]	74.33 [7.26]	72.67 [7.73]
SVM	77.00 [4.14]	82.67 [6.51]	79.83 [6.09]
SVM ensemble	81.33 [4.42]	88.33 [5.56]	84.83 [6.09]

As can be seen from Table 1, we can find the following several conclusions:

(1) The SVM ensemble model is the best of all the listed models in terms of Type I accuracy and Type II accuracy as well as total accuracy, indicating that the proposed evolutionary programming based SVM ensemble model is a promising technique for corporate failure prediction.

(2) For three evaluation criteria, the EP-based SVM ensemble model performs the best, followed by the single SVM, logistics regression, ANN model, and linear discriminant analysis model. Interestedly, the performance of logistic regression model

is better than that of ANN model. The possible reason leading to this conclusion may be data scarcity or other unknown reasons.

(3) Although the performance of the ANN model is worse than that of the logit regression for Type II accuracy, the robustness of logit regression is slightly worse than that of ANN model. The reasons are worth exploring further in the near future.

(4) Using two tailed *t*-test, we find that the differences among the former three methods are insignificant at 5% significance level, and there are significant differences between the former three methods and the latter two methods at 1% significance level. Furthermore, there is a significant difference between the single SVM method and the SVM ensemble model at 10% significance level. From the general view, the EP-based SVM ensemble dominates the other four classifiers, revealing the proposed EP-based SVM ensemble is an effective tool for corporate failure prediction.

4 Conclusions

In this study, a novel evolutionary programming (EP) based support vector machine ensemble classification method is proposed for corporate failure prediction. Through the practical data experiment, we have obtained good classification results and meantime demonstrated that the SVM ensemble model outperforms all the benchmark models listed in this study. These advantages imply that the novel SVM ensemble technique can provide a feasible solution to corporate bankruptcy prediction problem.

Acknowledgements. This work is supported by the grants from the National Natural Science Foundation of China (NSFC No. 70601029), the Chinese Academy of Sciences (CAS No. 3547600), the Academy of Mathematics and Systems Sciences (AMSS No. 3543500) of CAS, and the Strategic Research Grant of City University of Hong Kong (SRG No. 7001806).

References

1. Lai, K.K., Yu, L., Wang, S.Y., Zhou, L.G.: Credit Risk Analysis Using a Reliability-based Neural Network Ensemble Model. Lecture Notes in Computer Science 4132 (2006) 682-690
2. Olmeda, I., Fernandez, E.: Hybrid Classifiers for Financial Multicriteria Decision Making: The Case of Bankruptcy Prediction. Computational Economics 10 (1997) 317-335
3. Vapnik, V.: The Nature of Statistical Learning Theory. Springer, New York (1995)
4. Breiman, L.: Bagging Predictors. Machine Learning 26 (1996) 123-140
5. Lai, K.K., Yu, L., Wang, S.Y., Zhou, L.G.: Neural Network Meta-learning for Credit Scoring. Lecture Notes in Computer Science 4113 (2006) 403-408
6. Lai, K.K., Yu, L., Huang, W., Wang, S.Y.: A Novel Support Vector Machine Metamodel for Business Risk Identification. Lecture Notes in Artificial Intelligence 4099 (2006) 480-484
7. Fogel, D.B.: System Identification through Simulated Evolution: A Machine Learning Approach to Modeling. Ginn Press, Needham, MA (1991)
8. Hansen, L.K., Salamon, P.: Neural Network Ensembles. IEEE Transactions on Pattern Analysis and Machine Intelligence 12 (1990) 993-1001

9. Yang, S., Browne, A.: Neural Network Ensembles: Combining Multiple Models for Enhanced Performance Using a Multistage Approach. Expert Systems 21 (2004) 279-288
10. Schapire, R.E.: The Strength of Weak Learnability. Machine Learning 5 (1990) 197-227
11. Partridge, D., Yates, W.B.: Engineering Multiversion Neural-Net Systems. Neural Computation 8 (1996) 869-893
12. Damavandi, N., Safavi-Naeini, S.: A Robust Model Parameter Extraction Technique Based on Meta-Evolutionary Programming for High Speed/High Frequency Package Interconnects. 2001 Canadian Conference on Electrical and Computer Engineering - IEEE-CCECE, Toronto, Ontario, Canada (2001) 1151-1155
13. Beynon, M.J., Peel, M.J.: Variable Precision Rough Set Theory and Data Discretisation: An Application to Corporate Failure Prediction. Omega 29 (2001) 561-576