Credit Rating Using a Hybrid Voting Ensemble

Elias Kamos[1], Foteini Matthaiou[1], and Sotiris Kotsiantis[2]

[1] Hellenic Open University, Greece
kamos.h@nbg.gr,fmatthaiou@yahoo.gr
[2] Department of Mathematics, University of Patras, Greece
sotos@math.upatras.gr

Abstract. Credit risk analysis is an essential topic in the financial risk management. Credit risk analysis has been the main focus of financial and banking industry. A number of experiments have been conducted using representative supervised learning algorithms, which were trained using two public available credit datasets. The decision of which specific method to choose is a complex problem. Another option instead of choosing only one method is to create a hybrid ensemble of classifiers.

1 Introduction

One of the important decisions financial institutions have to make as part of their operations is to make a decision whether or not to give a loan to an applicant. With the appearance of large data storing services, huge amounts of data have been stored regarding the repayment performance of past applicants. It is the aim of credit scoring to examine this data and build models that differentiate consistent from bad payers using features such as amount on savings account, purpose of loan, marital status, etc.

Many machine learning techniques have been used to build credit-scoring models [2], [7]. The decision of which specific method to choose is a difficult problem for credit risk analysis [1]. A high-quality alternative to choosing only a method is to create an ensemble of classifiers [18], [11]. In this study, we have implemented a hybrid decision support system that combines representative algorithms using a voting methodology and achieves better accuracy than any simple method.

The following section attempts a brief literature review for credit risk analysis. Section 3 provides a brief description of the used datasets. Section 4 presents the presented method and the experimental results for the representative compared combining techniques. Finally, section 6 discusses the conclusions and some future research directions.

2 Literature Review

Because of credit risk analysis importance, there is a growing research interest about credit risk analysis. A recent survey on credit scoring and credit modeling is [16]. Many different approaches including individual models, such as kernel classifiers [7], classification tree [19], artificial neural networks (ANN) [9], [10], [6] support vector

I. Maglogiannis, V. Plagianakos, and I. Vlahavas (Eds.): SETN 2012, LNAI 7297, pp. 165–173, 2012.

machine (SVM) [20], [8] and some hybrid models, such as neuro-fuzzy system [22] and immune classifiers [3] were widely applied to credit risk analysis tasks. In the above individual models, it is difficult to say that the accuracy of one model is consistently better than that of another model in all circumstances.

In most situations, the performance of these individual models is problem-dependent. In the hybrid models [21], [19], some researchers have revealed that these hybrid classifiers which hybridize two or more classification methods can provide higher classification accuracy than that of individual models. Motivated by this finding, we integrate multiple classifiers into an aggregated output to achieve the further performance improvement.

3 Data Description

We used two publicly credit datasets: Credit-a dataset and Credit-g dataset.

Table 1. Credit-a Dataset – List of Attributes

Attribute	Type
Sex	Nominal
Age	Continuous
Mean time at addresses	Continuous
Home status	Nominal
Current occupation	Nominal
Current job status	Nominal
Mean time with employers	Continuous
Other investments	Nominal
Bank account	Nominal
Time with bank	Continuous
Liability reference	Nominal
Account reference	Nominal
Monthly housing expense	Continuous
Savings account balance	Continuous
Class (Reject / Accept)	Nominal

In Credit-a dataset, each case out of 690 represents an application for credit card facilities described by eight discrete and six continuous attributes, with two decision classes (Accept / Reject). The database attributes are shown in Table 1. The German Credit dataset (Credit-g) contains observations on 20 variables for 1000 past applicants for credit. Each applicant was rated as “good credit” (700 cases) or “bad credit” (300 cases). The database attributes are shown in Table 2.

Table 2. Credit-g Dataset – List of Attributes

Attribute	Type
Checking account status	Nominal
Duration of credit in months	Continuous
Credit history	Nominal
Purpose of credit	Nominal
Credit amount	Continuous
Average balance in savings account	Nominal
Present employment	Nominal
Installment rate as % of disposable income	Continuous
Personal status	Nominal
Other parties	Nominal
Present resident since - years	Continuous
Property magnitude	Nominal
Age in years	Continuous
Other payment plans	Nominal
Housing	Nominal
Number of existing credits at this bank	Continuous
Nature of job	Nominal
Number of people for whom liable to provide maintenance	Continuous
Applicant has phone in his or her name	Nominal
Foreign worker	Nominal
Class (Reject / Accept)	Nominal

4 Experimental Results and Proposed Technique

For the purpose of this study, a representative algorithm for each supervised learning technique was used. The most commonly used C4.5 algorithm [13] was the representative of the decision trees in our study. The K2 algorithm [24] was the representative of the Bayesian networks in our study. BP algorithm [24] - was the representative of the ANNs. Ripper [4] was the representative of the rule-learners. The 3-NN algorithm that combines robustness to noise and less time for classification than using a larger k for kNN was also used [24]. Finally, the Sequential Minimal Optimization (or SMO) algorithm was the representative of the SVMs as one of the fastest methods to train SVMs [12].

All accuracy estimates were calculated by averaging the results from stratified 10-fold cross-validation in the datasets. It must be mentioned that we make use of the

free available source code for our experiments by the book [24]. The results for the credit-a as well as the credit-g datasets are presented in Table 3. Three evaluation criteria were used to measure the classification results:

- Total accuracy = (number of correct classification) / (the number of evaluation sample)
- Type I accuracy = (number of both observed bad and classified as bad) / (number of observed bad)
- Type II accuracy = (number of both observed good and classified as good) / (number of observed good)

Table 3. Accuracy of simple models in credit datasets

Dataset		K2	C4.5	3NN	BP	RIPPER	SMO	LogReg
credit-a	Total	86.23%	86.08%	84.63%	86.52%	85.79%	84.92%	85.21%
	Type-I	79.8%	83.7%	82.1%	86%	86%	92.2%	86.3 %
	Type-II	91.4%	88%	86.7%	86.9 %	85.6 %	79.1%	84.3 %
credit-g	Total	75.5%	70.5%	73.3%	72.5%	71.7%	75.1%	75.2%
	Type-I	85.9%	84%	86.1%	77.4%	87.3%	87.1%	86.4%
	Type-II	51.3%	39%	43.3%	61%	35.3%	47%	49%

Lately in the area of machine learning and data mining the concept of combining classifiers is suggested as a new direction for the improvement of the accuracy of individual classifiers. Witten & Frank [24] provides an accessible and informal reasoning, from statistical, computational and representational viewpoints, of why ensembles can improve classification results. The most typical method is to use a mixture of learning algorithms on all of the training data and combine their predictions according to a voting scheme. This technique attempts to achieve diversity [23] in the classification errors of the classifiers by using different learning algorithms, which vary in their method of search and representation. The intuition is that the classifiers generated using different learning biases are expected to make errors in different manner [26].

Using a voting methodology as an aggregation rule with the classifiers in the proposed algorithm, we wait for producing good results based on the idea that the majority of classifiers are more probable to be right in their decision when they agree in their estimation. According to the proposed algorithm, during the classification of a test example the ensemble model calculate the votes of each class and if the votes of the base-classifiers of the most possible class is at least two times the votes of the next possible class then the decision is that of the most possible class. But, if the global voting ensemble is not so sure e.g. the votes of the most possible class is less than two times the votes of the next possible class; the model finds the k nearest neighbors using the selected distance metric and train the local voting ensemble using these k instances. Finally, in this case the model uses simple voting of the global voting classifiers with local voting classifiers for the classification of the testing instance.

The proposed ensemble is described by pseudo-code in Fig 1.

```
Training:
Build Global Classifiers using the training set
Classification:
 1. Obtain the test instance
 2. Calculate the decisions of the base-classifiers.
 3. If the votes of the most possible class is at least two
    times the votes of the next possible class then the de-
    cision is the most possible class else
       a. Find the k(=50) nearest neighbors using the se-
          lected distance metric (Manhattan in our imple-
          mentation)
       b. Using as training instances the k instances
          train the local classifiers
       c. Aggregate the decisions of global classifiers
          with local classifiers by simple voting for the
          classification of the testing instance.
```

Fig. 1. Integrating Global and Local Voting

The proposed algorithm requires choosing the value of *k*. There are quite a few methods to do this. Firstly, simple solution is to fix *k* a priori before the beginning of the learning process. However, the best *k* for a specific dataset is clearly not the best for another dataset. A second, more time-consuming solution is to determine this best *k* automatically through the minimization of a cost criterion. The idea is to apply a model selection process upon which the different hypothesis that may be built. One technique to do that is to evaluate the error on a test set and thus keep as *k* the value for which the error is the least. In the current implementation we decided to use a fixed value for *k* (=50) in order to a) keep the training time low and b) since about this size of instances is appropriate for a simple algorithm, to construct a relatively precise model.

Subsequently, we compare in Table 4 the proposed methodology (GlLocVot) for the credit-a as well as the credit-g datasets with:

- The methodology of selecting the best classifier according to 3-cross validation (BestCV) [24].
- Grading methodology using the instance based classifier IBk with ten nearest neighbors as the meta level classifier [14]. In grading, the meta-level classifier predicts whether the base-level classifier is to be trusted. The base-level attributes are used also as meta-level attributes, while the meta-level class values are correct and incorrect. Only the base-level classifiers that are predicted to be correct are taken and their predictions combined by summing up the probability distributions predicted.
- Simple Voting methodology using the same base classifiers [25].
- Stacking that replaces this with a trainable classifier [17]. This is possible, since for the training set, we have both the predictions of the base learners and the true class. The matrix containing the predictions of the base learners as

predictors and the true class for each training case will be called the meta-data set. The classifier trained on this matrix is called the meta-classifier or the classifier at the meta-level. Stacking methodology that constructs the meta-data set by adding the entire predicted class probability distribution instead of only the most likely class using MLR as meta-level classifier was used in our experiments [17].

In Table 4, we represent with "v" that the proposed method looses from the specific algorithm. That is, the specific algorithm performed statistically better than the proposed method according to t-test with $p<0.05$ [24]. Furthermore, in Table 4, "*" indicates that proposed method performed statistically better than the specific algorithm according to t-test with $p<0.05$. In all the other cases, there is no significant statistical difference between the results.

Table 4. Accuracy of ensembles in credit datasets

Dataset		GlLocVot	Voting	BestCV	Grading	Stacking
Credit-a	Total	88.6%	87.39%	84.63% (*)	85.21% (*)	87.39%
	Type-I	83.9%	88.3% (v)	82.4%	85%	86.3%
	Type-II	92.4%	86.7% (*)	86.4% (*)	85.4% (*)	88.3% (*)
Credit-g	Total	78.7%	76% (*)	75.6% (*)	75.9% (*)	76.1% (*)
	Type-I	88.2%	88%	87%	88.1%	89.6%
	Type-II	56.7%	48% (*)	49% (*)	47.3% (*)	44.7% (*)

As a conclusion, our approach performs better than selecting the best classifier from the ensemble by cross validation and other tested combining methods in the credits dataset. Because of the encouraging results obtained from these experiments, we can expect that the proposed technique can be effectively applied to the classification task in real world cases, and perform more accurately than traditional data mining approaches.

5 Conclusion

Deciding whether an applicant is a 'good' or a 'bad' risk is known as credit scoring. In general, credit scoring includes any technique for classifying risks into a set of predefined categories [5]. Traditional credit scoring methods award points for certain features that the creditor considers important such as the amount of the applicant's income, whether he or she owns a home, and how many years he has worked in his last job [15].

The aim of this study was to investigate the usefulness and compare the performance of supervised machine learning techniques in credit scoring. In terms of classification accuracy, the proposed new voting methodology achieves better accuracy than any examined simple and ensemble method. The weakness of the proposed method is the decreased comprehensibility. With involvement of multiple classifiers in decision-making, it is more difficult for non-expert users to perceive the underlying reasoning procedure leading to a decision.

Tracking progress is a time-consuming job that can be handled automatically by the implemented tool (see Figure 2). The tool expects the training set as an Attribute-Relation File Format (ARFF). There is not any restriction in attributes' order. However, the class attribute must be in the last column. After the training of the model (this takes some time to complete, from few seconds to few minutes), the user is able to predict the class of the new single instance. While the experts will still have the vital role in evaluating process, the tool can use the data required for reasonable and efficient monitoring.

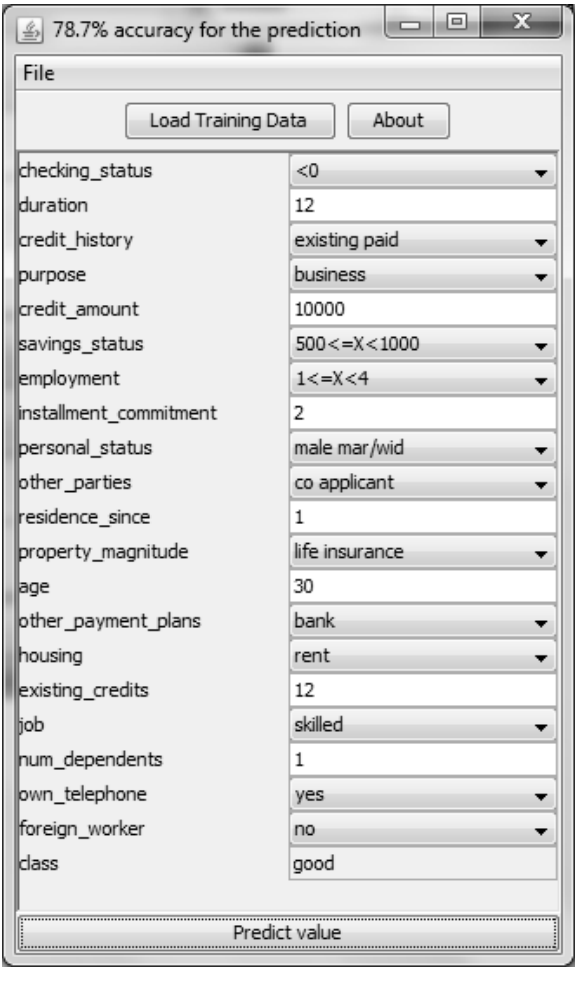

Fig. 2. A screenshot of the implemented decision support tool

References

1. Baesens, B., Van Gestel, T., Viaene, S., Stepanova, M., Suykens, J., Vanthienen, J.: Benchmarking state-of-the-art classification algorithms for credit scoring. Journal of the Operational Research Society 54(6), 627–635 (2003)

2. Yap, B.W., Ong, S.H., Husain, N.H.M.: Using data mining to improve assessment of credit worthiness via credit scoring models. Expert Systems with Applications 38(10), 13274–13283 (2011)
3. Chang, S.-Y., Yeh, T.-Y.: An artificial immune classifier for credit scoring analysis. Applied Soft Computing (November 12, 2011), 10.1016/j.asoc.2011.11.002
4. Cohen, W.: Fast Effective Rule Induction. In: International Conference on ML, pp. 115–123 (1995)
5. Hand, D.J.: Good practice in retail credit scorecard assessment. Journal of the Operational Research Society 56, 1109–1117 (2002)
6. Hájek, P.: Municipal credit rating modelling by neural networks. Decision Support Systems 51(1), 108–118 (2011)
7. Huang, S.-C.: Using Gaussian process based kernel classifiers for credit rating forecasting. Expert Systems with Applications 38(7), 8607–8611 (2011)
8. Huang, Z., Chen, H.C., Hsu, C.J., Chen, W.H., Wu, S.S.: Credit Rating Analysis with Support Vector Machines and Neural Networks: A Market Comparative Study. Decision Support Systems 37, 543–558 (2004)
9. Khashman, A.: Credit risk evaluation using neural networks: Emotional versus conventional models. Applied Soft Computing 11(8), 5477–5484 (2011)
10. Khashman, A.: Neural networks for credit risk evaluation: Investigation of different neural models and learning schemes. Expert Systems with Applications 37(9), 6233–6239 (2010)
11. Yu, L., Yue, W., Wang, S., Lai, K.K.: Support vector machine based multiagent ensemble learning for credit risk evaluation. Expert Systems with Applications 37, 1351–1360 (2010)
12. Platt, J.: Using sparseness and analytic QP to speed training of support vector machines. In: Kearns, M.S., Solla, S.A., Cohn, D.A. (eds.) Advances in Neural Information Processing Systems 11. MIT Press, MA (1999)
13. Quinlan, J.R.: C4.5: Programs for machine learning. Morgan Kaufmann, San Francisco (1993)
14. Seewald, A.K., Fürnkranz, J.: An Evaluation of Grading Classifiers. In: Hoffmann, F., Adams, N., Fisher, D., Guimarães, G., Hand, D.J. (eds.) IDA 2001. LNCS, vol. 2189, pp. 115–124. Springer, Heidelberg (2001)
15. Li, S., Tsang, I., Chaudhari, N.: Relevance vector machine based infinite decision agent ensemble learning for credit risk analysis. Expert Systems with Applications 39, 4947–4953 (2012)
16. Thomas, L.C., Oliver, R.W., Hand, D.J.: A Survey of the Issues in Consumer Credit Modelling Research. Journal of the Operational Research Society 56, 1006–1015 (2005)
17. Ting, K., Witten, I.: Issues in Stacked Generalization. Artificial Intelligence Research 10, 271–289 (1999)
18. Tsai, C.-F., Chen, M.-L.: Credit rating by hybrid machine learning techniques. Applied Soft Computing 10(2), 374–380 (2010)
19. Wang, G., Hao, J., Ma, J., Jiang, H.: A comparative assessment of ensemble learning for credit scoring. Expert Systems with Applications 38(1), 223–230 (2011)
20. Wang, G., Ma, J.: A hybrid ensemble approach for enterprise credit risk assessment based on Support Vector Machine. Expert Systems with Applications 39(5), 5325–5331 (2012)
21. Wang, G., Ma, J.: Study of corporate credit risk prediction based on integrating boosting and random subspace. Expert Systems with Applications 38(11), 13871–13878 (2011)
22. Wang, Y.Q., Wang, S.Y., Lai, K.K.: A New Fuzzy Support Vector Machine to Evaluate Credit Risk. IEEE Transactions on Fuzzy Systems 13, 820–831 (2005)

23. Windeatt, T.: Diversity measures for multiple classifier system analysis and design. Information Fusion 6, 21–36 (2005)
24. Witten, I., Frank, E., Hall, M.: Data Mining: Practical Machine Learning Tools and Techniques, 3rd edn. Morgan Kaufmann (2011) ISBN 978-0-12-374856-0
25. Xiaoyan, M., Watta, P., Hassoun, M.H.: Analysis of a Plurality Voting-based Combination of Classifiers. Neural Process Lett. 29, 89–107 (2009)
26. Zouari, H., Heutte, L., Lecourtier, Y.: Controlling the diversity in classifier ensembles through a measure of agreement. Pattern Recognition 38, 2195–2199 (2005)