An Investigation Of The Effect Of Variable Reduction On Classification Accuracy Rates Of Consumer Loans

Jozef Zurada, University of Louisville, USA

ABSTRACT

The profitability of loan granting institutions depends largely on the institutions' ability to accurately evaluate credit risk. Their goal is to maximize income by issuing as many good loans to consumers as possible while minimizing losses associated with bad loans. Financial institutions have been using various computational intelligence methods and statistical techniques to improve credit risk prediction accuracy. This paper examines historical data from consumer loans issued by a German bank to individuals. The data consists of the financial attributes of each customer and includes a mixture of loans that the customers paid off and defaulted upon.

This paper examines and compares the classification effectiveness of four computational intelligence techniques: 1) logistic regression (LR), 2) neural networks (NNs), 3) support vector machines (SVM), and 4) k-nearest neighbor (kNN) on three data sets to predict whether a consumer defaulted or paid off a loan. The first data set contains a full set of 20 input variables. The second and third data sets contain a reduced set of ten and six variables, respectively. The results from computer simulation show a limited effect of variable reduction on improvement in the classification performance.

Keywords: Classification; Loan-Granting Decisions; Variable Reduction; Logistic Regression; Neural Networks; Support Vector Machines; Decision Trees

INTRODUCTION

Many financial services institutions are developing credit scoring models to support their credit decisions. The ultimate objective of these models is to increase accuracy in loan-granting decisions so that more creditworthy applicants are granted credit, thereby increasing profits, and non-creditworthy applicants are denied credit, thus decreasing losses. Even a slight improvement in accuracy rates may translate into significant future savings measured in millions of US dollars. Determining whether a particular consumer should receive a loan is an inherently complex and, to a large extent, unstructured process. A financial institution must examine many independent financial attributes of each loan candidate in an accurate, prompt, and cost effective manner. The financial institution approximates the risk of default by the candidate and weighs that risk against the benefit of potential earnings on the loan. Any improvement in making a reliable distinction between those who are likely to repay the loan and those who are not would allow the bank to reject the riskiest loans and to adjust the terms of the granted loans according to the risk of default. The volume and complexity of raw data inherent in credit-risk assessment can be tackled by several traditional statistical techniques and newer computational intelligence methods.

This paper examines and compares the classification effectiveness of four computational intelligence techniques (LR, NNs, SVM, and *k*NN) on three data sets to predict whether a consumer defaulted or paid off a loan. The first data set contains a full set of 20 input variables. The second and third data sets contain a reduced set of ten and six variables, respectively. This paper contains sections on literature review; an explanation of the fundamentals of logistic regression (LR), neural networks (NN), support vector machines (SVM), and the *k*-nearest neighbor (*k*NN) method; a description of the features of the data; experiments and simulation results; and conclusion and recommendations for future work.

LITERATURE REVIEW

The research on credit-scoring and loan-granting decisions is abundant. For example, in one of the early papers, McLeod *et al*. (1993) discussed general features of NNs and their suitability for the credit-granting process. Glorfeld and Hardgrave (1996) presented a comprehensive and systematic approach to developing an optimal architecture of a NN model for evaluating the creditworthiness of commercial loan applications. The NN developed using their architecture was capable of correctly classifying 75% of loan applicants and was superior to NNs developed using simple heuristics. Desai *et al*. (1996) analyzed the usefulness of NNs and traditional techniques, such as linear discriminant analysis (LDA) and LR, in building credit scoring models for credit unions. Desai *et al*. studied data samples containing 18 variables collected from three credit unions and showed that NNs were particularly useful in detecting bad loans, whereas LR outperformed NNs in the overall (bad and good loans) classification accuracy. Tessmer (1997) examined credits granted to small Belgian businesses using a decision tree (DT)-based learning approach. The author focused on the impact of Type I credit errors (classifying good loans as bad loans) and Type II credit errors (classifying bad loans as good loans) on the accuracy, stability and conceptual validity of the learning process.

Subsequent authors built on the existing research by comparing the performance of various data mining techniques in various credit risk assessment contexts. Jagielska *et al*. (1999) investigated credit risk classification abilities of NNs, fuzzy logic, genetic algorithms, rule induction software, and rough sets and concluded that the genetic/fuzzy approach compared more favorably with the neuro-fuzzy (NF) and rough set approaches. Piramuthu (1999) analyzed the beneficial aspects of using both NNs and NF systems as well as variable reduction for credit-risk evaluation decisions. NNs performed significantly better than NF systems, in terms of classification accuracy, on both training as well as testing data.

In more recent series of papers, Khashman (2009) uses NNs on an Australian data set and finds that single-hidden layer NN outperforms double-hidden layer NN and that a training to validation ratio of 43.5:56.5 percent is the best training scheme on the data. Bellotti and Crook (2009) use SVM, LR, LDA and *k*NN on a very large data set (25,000 records) from a financial institution and find that SVM is comparatively successful in classifying credit card debtors who do default; but unlike the other compared models, a large number of support vectors are required to achieve the best performance. Two comparative studies (Zurada, 2007, 2010) use LR, NN, DT, memory-based reasoning (MBR) and an ensemble model using German and SAS-1 data sets. Both found that for some cut-off points and conditions, DTs perform well with respect to classification accuracy and that DTs are attractive tools for decision makers because they can generate easy to interpret if-then rules. Finally, very few papers (Piramuthu, 1999) tested the effect of variable reduction on general classification performance of the methods in the credit scoring context.

METHODOLOGY

As the methods used in this study are pretty well known, we only provide their short summary. The purpose of the LR model is to obtain a regression equation that could predict in which of two groups an object could be placed (e.g. a *good loan* category or a *bad loan* category). The LR regression model also attempts to predict the probability that a binary target will acquire the event of interest (e.g. *loan payoff* or *loan default*) as a function of one or more independent variables (i.e., amount of loan, customer job category, reason of loan, number of credit lines open, etc.).

NNs are mathematical models that mimic the way the human brain functions and processes information. They are nonlinear systems built of highly interconnected neurons. The most attractive features of these networks are their ability to adapt, generalize, and learn from training patterns. NN models are characterized by their three properties - computational, network architecture learning properties. A typical neuron contains a summation node and a nonlinear activation function. A neuron accepts vectors on input called training patterns/examples. Neurons are organized in layers and are connected by weights represented by small numerical values. In this study, we used the most common type of the NN architecture - a two-layer feed-forward NN with error back-propagation. Most commonly, the network has two layers - a hidden layer and an output layer. The neurons at the hidden layer receive the values of input vectors and propagate them concurrently to the output layer.

SVM, originally developed by Vapnik (1998), is a method that represents a blend of linear modeling and instance-based learning to implement nonlinear class boundaries. This method chooses several critical boundary patterns, called support vectors, for each class (*bad loan* and *good loan* of the output variable) and creates a linear discriminant function that separates them as widely as possible by applying linear, quadratic, cubic or higher-order polynomial term decision boundaries. A hyperplane that gives the greatest separation between the classes is called the maximum margin hyperplane. SVMs are slow but often yield accurate classifiers because they create subtle and complex decision boundaries.

In solving a new case, the *k*-NN approach retrieves the cases it deems sufficiently similar and uses these cases as a basis for solving the new case (Mitchell, 1997). The *k*-NN algorithm takes a data set of existing cases and a new case, to be classified, where each existing case in the data set is composed of a set of variables and the new case has one value for each variable. The normalized Euclidean distance or Hamming distance between each existing case and the new case (to be classified) is computed. The *k* existing cases that have the smallest distances to the new case are the *k*-nearest neighbors to that case. Based on the target values of the *k*-nearest neighbors, each of the *k*-nearest neighbors votes on the target value for the new case. The votes are the posterior probabilities for the class dependent variable.

DATA SET USED IN THE STUDY

We used the German data set which has already been used in a number of studies. The data set, which we later call a full data set, contains 20 input variables. The name of the attribute is listed first, followed by its description and the number of levels (unique values) the attribute takes in case of nominal/ordinal attributes. The variables on the interval scale are: 1) Age - Age of applicant [years], 2) Amount - Amount of credit requested [$], 3) Depends - Number of dependents, 4) Duration - Length of loan [months], 5) ExistCr - Number of existing accounts at this bank, 6) DebtPer - Debt as a percent of disposable income [%], and 7) Resident - Stay at current address [Years]. The binary variables are Foreign – 8) Foreign worker [Yes/No] and 9) Telephone –Telephone registered under customer’s name [Yes/No]. The nominal/ordinal variables are: 10) Balance - Balance in existing checking account [4 levels], 11) Debtors - Other debtors or guarantors [3 levels], 12) TimeEmp - Time at present employment [5 levels], 13) Credit - Credit history [5 levels], 14) Housing - Rent/Own a house [3 levels], 15) Employed - Employment status (4 levels), 16) Marital - Marital status and gender (5 levels), 17) Other - Other installment loans [3 levels], 18) CoApp - Collateral property for loan [4 levels], 19) Purpose - Reason for loan request [11 levels], and 20) Savings - Savings account balance [5 levels]. There is also one output binary variable Credit Rating Status which takes two outcomes [Good/Bad].

The nominal/ordinal variables created significant problems as they have to be converted to dummy variables. Simply, in 1-to-*n* coding, each level of a nominal variable represents one dummy variable. This resulted in many additional dummy variables which have to be added on input to the models. Some of the variables on the ordinal scale could be coded as numeric. We have not, however, used this approach, though it would limit the number of dummy variables in the models.

COMPUTER SIMULATION AND DISCUSSION OF THE RESULTS

Computer simulation was performed using data mining software Weka (http://www.cs.waikato.ac.nz/ml/weka/). To obtain reliable classification rates, we used 10-fold cross-validation and repeated it 10 times. Tables 1-3 present the correct classification accuracy rates for a standard 0.5 cut-off, whereas Table 4 shows the areas under ROC curves. The rates and the areas are averaged over 100 runs. The cut-off should be interpreted as follows. The event is set to detect bad loans. Thus, if the model generates probability ≥ 0.5, the loan is classified as a bad loan; otherwise it is a good loan. The LR method and the full data set with 20 input variables are the baselines. Across the rows, we compare the performance of each of the three methods (NN, SVM or *k*NN) to LR; whereas down the columns we compare the performance of each of the two data sets with the reduced number of attributes to the full data set with 20 input attributes. We applied t-test to find out if the rates between the models and data sets are statistically significant. The superscripts to the right of the rates indicate that the method performs significantly better/worse (^{bb,ww}) at $\alpha = 0.01$ or (^{b,w}) at $\alpha = 0.05$ than the LR model. The subscripts to the left of the rates indicate that the data set with reduced number of variables performs significantly better/worse (${}_{bb,ww}$) at $\alpha = 0.01$or (${}_{b,w}$) at $\alpha = 0.05$ than the full data set with 20 variables.

We used several variable reduction techniques provided by Weka. These included the methods based on entropy reduction, R^2 and χ^2, to name a few. We used 10-fold cross validation in identifying the most relevant attributes. All methods were consistent in identifying pretty much the same attributes regardless of the fold/run. Consequently, we created two smaller data sets with 10 and 6 input attributes each. The ten attributes were Amount, Checking, Duration, Employed, History, Housing, Other, Property, Purpose, and Savings. The six attributes were Amount, Checking, Duration, Employed, History, and Savings.

Depending on the models or data sets, the overall rates vary between a low of 73.2% and a high of 76.0% (Table 1). The LR and SVM perform better than the two remaining models for two of the three data sets. The *k*NN model appears to be significantly worse than LR for the first two data sets. However, *k*NN appears to significantly outperform LR for the third data set with the least number of variables. In general, the variable reduction does not improve the accuracy rates of the models, except *k*NN for the third data set. As Table 2 presents, the classification accuracy rates for bad loans differ very significantly across the four models and the three data sets; they are within the range [33.0%, 49.3%]. LR and NN appear to work the best. Variable reduction does not cause an expected improvement in the rates. The performance of *k*NN (43.1%) improves significantly for the third data set, but it is still much worse than the performance (49.3%) of the NN model for the full data set. Table 3 depicts the rates for good loans. They are between 86.8% and 90.6%. The differences between the rates across the models and data sets are not so dramatically different from those shown in Table 2. For all three data sets, SVM and *k*NN models appear to be significantly better than LR, whereas NN seems to be significantly worse. Applying attribute reduction improves the classification rates for three models; i.e., LR, NN, and SVM.

Table 1: The Overall Correct Classification Accuracy Rates [%]

Method	**LR**	**NN**	**SVM**	***k*NN**
Data Set				
Full data set with 20 variables	75.5	74.9^{w}	75.4	73.7^{ww}
Data set with 10 variables	75.5	75.1	75.2	73.6^{ww}
Data set with 6 variables	${}_{w}74.7$	74.8	${}_{ww}73.2^{ww}$	${}_{bb}76.0^{bb}$

Table 2: The Correct Classification Accuracy Rates of Bad Loans [%]

Method	**LR**	**NN**	**SVM**	***k*NN**
Data Set				
Full data set with 20 variables	49.1	49.3	47.5^{ww}	35.8^{ww}
Data set with 10 variables	${}_{ww}46.1$	${}_{ww}47.1$	${}_{ww}43.3^{ww}$	36.0^{ww}
Data set with 6 variables	${}_{ww}41.4$	${}_{ww}45.8^{bb}$	${}_{ww}33.0^{ww}$	${}_{bb}43.1^{b}$

Table 3: The Correct Classification Accuracy Rates of Good Loans [%]

Method	**LR**	**NN**	**SVM**	***k*NN**
Data Set				
Full data set with 20 variables	86.8	85.9^{ww}	87.3^{bb}	89.9^{bb}
Data set with 10 variables	${}_{bb}88.1$	${}_{bb}87.1^{ww}$	${}_{bb}89.0^{bb}$	89.7^{bb}
Data set with 6 variables	${}_{bb}89.0$	${}_{bb}87.2^{ww}$	${}_{bb}90.6^{bb}$	90.1^{bb}

ROC curves reveal the global classification performance of the models and data sets for a continuum of cut-offs from within the range [0%, 100%]. Table 4 shows the areas under ROC curves which may vary between 50% and 100%. The smallest area is 73.8% (*k*NN for the data set with six variables) and the highest area amounts to 78.2% (LR for the full data set). Variable reduction makes the rates significantly worse, which is rather a surprise finding.

Table 4: The Areas Under ROC Curves [%]

Method	LR	NN	SVM	*k*NN
Data Set				
Full data set with 20 variables	78.2	77.6^{ww}	78.1	75.2^{ww}
Data set with 10 variables	$_{ww}77.6$	77.6	77.8	$_{w}74.4^{ww}$
Data set with 6 variables	$_{ww}76.8$	$_{ww}76.6^{w}$	$_{ww}76.3^{ww}$	$_{ww}73.8^{ww}$

CONCLUSION

This paper presents the effect of attribute reduction on the correct classification accuracy rates for the German data set. It compares the rates and areas under ROC curves across the methods and data sets. LR and the full data set are the baselines to which we compare the remaining three methods and two data sets, respectively. Though variable reduction causes significant improvement in classifying good loans, it has a negative influence on overall classification accuracy rates, rates for bad loans, and areas under ROC curves. To be able to generalize the results obtained in this preliminary study, future computer simulation should include a larger number of data sets drawn from the customer credit scoring context.

AUTHOR INFORMATION

Jozef Zurada earned his Ph.D. in Computer Science and Engineering from the University of Louisville in 1995. He is a Professor of Computer Information Systems in the College of Business at the University of Louisville. Dr. Zurada's research interests include applications of advanced computational intelligence methods for aiding in decision-making in business and manufacturing systems. He published two books and numerous articles in refereed journals and conference proceedings. Dr. Zurada teaches Data Mining and Knowledge Discovery, Infrastructure Technologies, and Database Design. He is a member of IEEE and ACM. E-mail: jmzura01@louisville.edu

REFERENCES

1. Bellotti, T., & Crook, J. (2009). Credit Scoring with Macroeconomic Variables Using Survival Analysis. *Journal of the Operational Research Society, 60*(12), 1699-1707. doi: 10.1057/jors.2008.130.
2. Desai, V.S., Crook, J.N., and Overstreet, G.A. (1996). A Comparison of Neural Networks and Linear Scoring Models in the Credit Union Environment. *European Journal of Operation Research*, *Vol. 95*, 24-37.
3. Glorfeld, L.W., and Hardgrave, B.C. (1996). An Improved Method for Developing Neural Networks: The Case of Evaluating Commercial Loan Credit Worthiness. *Computer & Operations Research*, *Vol. 23*, No. 10, 933-944.
4. Jagielska, I., Matthews, C., and Whitfort, T. (1999). An Investigation into the Application of Neural Networks, Fuzzy Logic, Genetic Algorithms, and Rough Sets to Automated Knowledge Acquisition for Classification Problems, *Neurocomputing*, *Vol. 24*, 37-54.
5. Khashman, A. (2009). A Neural Network Model for Credit Risk Evaluation. *International Journal of Neural Systems, 19*(4), 285-294.
6. Mitchell, T.M. (1997). *Machine Learning*, WCB/McGraw-Hill, Boston, Massachusetts.
7. McLeod, R.W., Malhotra, D.K., and Malhotra, R. (1993). Predicting Credit Risk: A Neural Network Approach. *Journal of Retail Banking*, *XV*(3), 37-40.
8. Piramuthu, S. (1999). Feature Selection for Financial Credit-Risk Evaluation Decisions, *INFORMS Journal on Computing*, *11*(3), 258-266.
9. Tessmer, A.C. (1997). What to Learn from Near Misses: An Inductive learning Approach to Credit Risk Assessment. *Decision Sciences*, *Vol. 28*, No. 1, pp. 105-120.
10. Vapnik, V. N. (1998). *Statistical Learning Theory*. New York: Wiley.
11. Zurada, J. (2007). Rule Induction Methods for Credit Scoring. *Review of Business Information Systems, 11*(2), 11-22.
12. Zurada, J. (2010). Could Decision Trees Improve the Classification Accuracy and Interpretability of Loan Granting Decisions? Paper presented at the 43rd Hawaii International Conference on System Sciences (HICSS'2010), Hawaii.

NOTES