Mining Multi-Dimensional Data for Decision Support

June M. Donato,[1] Jack C. Schryver,[1] Gregory C. Hinkel,[1] Richard L. Schmoyer Jr.,[1] Nancy W. Grady,[1,2] Michael R. Leuze[1,2]

[1] Oak Ridge National Laboratory, Oak Ridge TN 37831, USA
[2] Joint Institute for Computational Science, Knoxville TN 37996, USA

1 Introduction

While it is widely recognized that data can be a valuable resource for any organization, extracting information contained within the data is often a difficult problem. Attempts to obtain information from data may be limited by legacy data storage formats, lack of expert knowledge about the data, difficulty in viewing the data, or the volume of data needing to be processed.

The rapidly developing field of Data Mining or Knowledge Data Discovery is a blending of Artificial Intelligence, Statistics, and Human-Computer Interaction. Sophisticated data navigation tools to obtain the information needed for decision support do not yet exist. Each data mining task requires a custom solution that depends upon the character and quantity of the data.

A problem of international concern is personal bankruptcy, a rapidly increasing yet little understood phenomenon. Attempts to understand personal bankruptcy have involved the mining of credit card data. Credit card data is multi-dimensional in that it contains monthly account records, which are categorical or list based, and daily transaction records, which are time-series data. This problem presents unique requirements for data mining practitioners, due to the rapid response required for the approval of credit card transactions.

This paper presents a two-stage approach for handling the prediction of personal bankruptcy using credit card account data, combining decision tree and artificial neural network technologies. Topics to be discussed include the pre-processing of data, including data cleansing, the filtering of data for pertinent records, and the reduction of data for attributes contributing to the prediction of bankruptcy, and the two steps in the mining process itself.

2 Overview of the Two-stage Approach

The goal of this data mining project was to discover patterns in credit card transactions that indicate the onset of bankruptcy. Credit card account and transaction data were provided through a partnership with a major credit card issuer. Two sets of "sanitized" (i.e., card holder identification encrypted) data files were obtained: (1) "extract" files containing general account information for December 1995 and June 1996 and (2) individual "transaction" records for those accounts from January 1995 to June 1996.

In the first stage of the data mining approach, extract data attributes were selected for their high correlation with bankruptcy behavior and were used as input to a Decision Tree Inducer. The Tree Inducer generated a tree structure that classifies the behavior of accounts as OK, Delinquent, or Bankrupt. The resulting decision tree classification and transaction information were then passed to the second stage. This stage utilizes Neural Networks, specifically Recurrent Neural Networks, to model nonlinear relationships among variables.

3 Data Management and Computational Resources

We are using a dual processor Sparcstation 20 as our primary computer for processing the sample credit card data. This machine currently has 64 Megabytes of memory and 10 Gigabytes of disk storage. The extract data files for December 1995 and June 1996 consist of over one million credit card accounts or 1.1 Gigabytes of data. The transaction files for the corresponding accounts from January 1995 through June 1996 consist of 800 Megabytes per month. The data have been read, converted, and stored in a local tape silo.

4 Data Cleansing

One area of concern in data mining is the caliber of the data being studied. Frequently, even in financial applications, some preprocessing of the data is required to insure consistency of data fields.

For example, what does a null entry in a data base record mean? For a field of dollar amounts, a null may represent a value of $0.00, or it may mean that the value is not available, i.e., missing. In some situations it may be necessary to delete or ignore records with missing or inconsistent entries. However, in some situations this approach is neither possible nor desirable, and intelligent values must be substituted. This process requires expert knowledge of the applications field. In some cases, a new "dummy" value may be introduced to represent the missing value.

Also, when data is created and transfered from one location to another via tape archive or by any other method, errors in the data encoding or decoding may occur. These errors may result in corrupted data, values that do not adhere to the record definitions or values that do not agree with the declared type of the field in the database. Such records must be found and removed, and if possible replaced with the correct records.

Then there is also the issue of checking calculations based on the data. For example, if a running account balance is calculated from the transaction data, it should match the final balance listed in the monthly summary files. If this is not the case, it is important to determine why. It may not be due to errors in the data, but, as we have discovered in this application, differences in the definition of reporting periods for the different data base tables. Summary data for each person is produced on the person's cycle date, but this date is different from the date used in accumulating transaction balances.

5 Data Selection

We have taken stratified random samplings of accounts from the June 1996 extract file. These samplings are described in the general categories listed below.

Bankrupt. Accounts bankrupt in June 1996. Sample size approximately 4000 records.

Charged off. Accounts closed with their balanced "charged-off" for a variety of reasons, but that were not classed as bankrupt. Sample size approximately 2000 records.

Delinquent n Months. Accounts which are neither bankrupt nor charged-off but are delinquent by n months of payments. Sample size approximately 500 records for each value of $n = 1, 2, 3$.

OK accounts. Accounts deemed to be in good standing; not bankrupt, not charged-off, and without delinquencies. Sample size approximately 2000 records.

Random. A completely random selection (less any duplicates that appear in any of the above groups). Sample size approximately 2000 records.

These samples represent a total of 12942 accounts selected for analysis. The records for these accounts were taken from the December 1995 extract file along with the transactions from July 1995 through June 1996. This data was loaded into an Oracle database.

Further, we selected a subset of these accounts which were were considered "OK" or "Delinquent 3 months or less" according to the December 1995 extract data. This selection results in 9521 accounts used during the decision tree and neural net stages.

6 Data Reduction

The next step in the Knowledge Data Discovery process is to restrict the number of fields to be used in the mining. Irrelevant fields will at best reduce the performance of the mining algorithm, and at worst reduce its accuracy. Statistical techniques provide a means to determine which variables might be correlated with the desired classification results.

We began investigating the relationship between bankruptcy for charge-off by June 1996 with December 1995 extract data and transaction data prior to June 1996. Accounts designated as bankrupt or charged-off in December 1995 were not considered. For each possible predictor variable from the December 1995 extract data we performed a statistical analysis to investigate the association of the predictor with the June 1996 classification. A different analysis was performed depending on whether the predictor was itself a class variable or an (essentially) continuous variable. For discrete predictor variables we use a frequency-table analysis and compute the Cramer's V [1] statistic. For continuous predictor variables we use analysis of variance and compute the R-square (squared correlation) coefficient. Both of these statistics measure correlation, with a value of one denoting a perfect association.

The dataset contained 233 variables, 53 character and 180 continuous. While no variables seemed outstanding, numerous variables showed potential for predicting the June 1996 classification. Seventeen variables were selected for further analysis and were used as input to the decision tree stage.

7 Decision Tree Stage

In this stage, the input to the Tree Inducer consisted of those attribute values that were found to be most correlated to bankruptcy behavior. Typically, ten percent of the sample accounts were used to generate a decision tree (a set of decision rules). The resulting tree was then tested against the entire subset of 9521 accounts.

Graphically, the nodes of a decision tree represent tests on specific extract data attributes; each leaf represents a class (e.g., Bankrupt) or a probability distribution among classes to which the instances of the accounts are assigned. These leaf classifications provide preliminary clustering of the accounts into distinct behavioral patterns. In the next stage, for each of several interesting leaf nodes, a more accurate model using a neural network was developed.

An example tree, using one of many data sets in the University of California, Irvine repository ([5]), is shown in Fig. 1. This display was generated using the *MLC++* ([4]) software library.

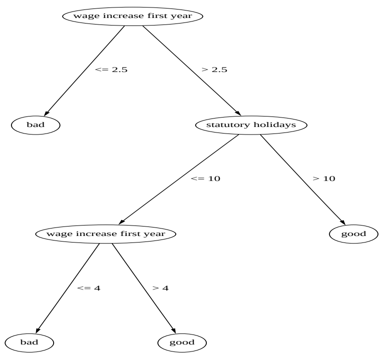

Fig. 1. Example Decision Tree

The example in Fig. 1 shows a pruned decision tree of size 7 with 4 leaves.

The current best pruned tree for the credit card extract data has size 244 (leaves and nodes) with a classification error rate of 24.4% using a confidence rate of 95%. The confusion matrix for the sample of 9521 accounts is given in Table 1.

Table 1. Example Confusion Matrix

Tree Classification			
BKR	DEL	OK	Actual Class
1988	668	465	BKR
810	1224	539	DEL
290	339	3174	OK

Of particular interest are those accounts with OK status that were classified as Bankrupt by the decision tree. Future analyses will examine these (290) accounts for behaviors that may foreshadow future bankruptcy.

8 Artificial Neural Network Stage

Partially recurrent neural networks (PRNNs) were chosen for the second stage because of their ability to model nonlinear relationships.

A significant feature of the bankruptcy prediction problem is that transaction data are ordered in temporal sequences. A specific PRNN architecture known as an Elman network ([2], [6]) has been proven to be a powerful tool ([3]) for classification of time series data. The Elman network is particularly suitable for learning transaction sequences.

The entire transaction sequence is not learned as a single pattern in an Elman PRNN. Rather, transactions are posed to Elman networks one at a time in sequence, like a pushdown stack. Since the network accepts only a single transaction, Elman PRNNs are quite attractive for use in a real-time computing environment, e.g., where a quick credit card transaction authorization decision is needed. Only a small list of numbers representing the current state of the PRNN for each account must be stored on-line. This may be cheaper and faster than requiring on-line access to a whole credit card transaction history.

Since the transactions are arranged in a pushdown stack, the fixed input size and alignment problems are elegantly handled by the Elman PRNN. Only the number of transaction attributes is fixed; the number of transactions is free to vary. Training is tractable because networks are more compact due to smaller input vector size, and a greater number of training patterns—one pattern per transaction.

A single transaction is presented to an Elman PRNN at the input layer. Example continuous input variables include amount of transaction, current balance, and time interval between transactions. Several discrete inputs are encoded as bit-vectors. Static inputs, such as credit line, average daily balance, and the amount in arrears, were extracted from the Oracle database. The decision tree classification and class probability estimates were also used as input variables.

The input layer is fully interconnected with the hidden layer, which encodes the emergent features of the input. A single output unit indicates the probability of bankruptcy and is connected to all units in the hidden layer. Elman PRNNs store sequence information in a state vector.

After each transaction is processed, activations in the hidden layer are copied into the context layer. The context layer is a memory storage that preserves the current network activation state and allows the network to retain sequence information. The context layer is fed back into the hidden layer during presentation of the next transaction at the input layer. A special reset unit is normally set to zero, but when set to unity, the network resets activations in the context layer to default values, thereby "erasing" the contextual memory.

A stratified sample was drawn from the appropriate decision tree leaf and split into two datasets, along with the corresponding transaction information. These pattern sets were then input into Elman PRNNs using the Stuttgart Neural Network Simulator (SNNS) [7]. A cross-validation procedure was used to train and validate the pair of datasets.

Since bankruptcy prediction is a supervised learning problem, it was necessary to provide a teaching output for each transaction in every account.

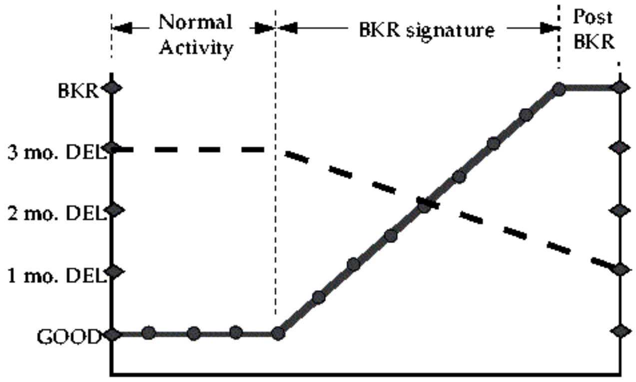

Fig. 2. Linear Teacher Model

The simple piecewise linear model shown in Fig. 2 was constructed to meet the teacher assumptions. Delinquent accounts were arbitrarily assigned positive scores dependent on the duration of delinquency. The bankruptcy prediction remains flat until reaching the window of the bankruptcy signature. The length of this window is a free model parameter. Progress to the end state is linear within this window.

For a proof-of-principle demonstration, 198 accounts with five or more transactions were selected as a stratified sample from a decision tree root and were randomly divided into two equal-size pattern sets. Transaction data were selected from a period spanning January 1996 through June 1996. Static input variables were drawn from the December 1995 extract file, and the teacher output was based on the December 1995 and June 1996 extract files. The trained networks represented the lowest Mean-Squared-Error (MSE) in 5000 supervised learning trials. Each set was trained on 10 different Elman PRNN architectures.

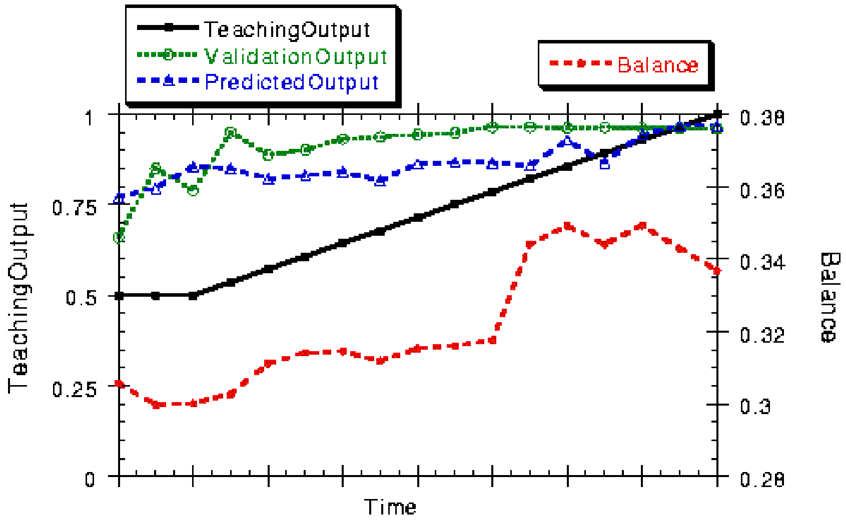

Fig. 3. Example Network Output for a Single Account

Figure 3 shows network output for a single account. Test and validation outputs are quite similar, although the test output follows the teaching output a little more closely. Network outputs are highly autocorrelated, indicating that global trends are quite stable and are not strongly affected by attributes of individual transactions. Network outputs do not track current balance very closely, for example. Both test and validation outputs begin with large error but converge on the teacher output.

9 Conclusions

Experience with initial training sets suggests that PRNNs can be trained to perform quite well on transaction data. There is a loss of accuracy when the PRNNs are generalized to new accounts, but a substantial amount of predictive ability is retained. However, the percent correct classification is only a crude measure of performance in the present context. A more appropriate measure would capture the "bottom line," or the potential financial savings to the credit card issuer.

Acknowledgments

This research was sponsored by the Laboratory Directed Research and Development Program of Oak Ridge National Laboratory, managed by Lockheed Martin Energy Research Corp. for the U.S. Department of Energy under Contract No. DE-AC05-96OR22464. Support for travel to HPCN'98 was provided by the Joint Institute for Computational Science of the University of Tennessee and Oak Ridge National Laboratory.

References

1. Bishop, Y. M. M., Feinberg, S. E., Holland, P. W.: *Discrete Multivariate Analysis*. MIT Press, Cambridge, MA (1975)
2. Elman, J. L.: Finding structure in time. Cognitive Science **14**(2) (April-June 1990) 179–212
3. Kremer, S. C.: On the computational power of Elman-style recurrent networks. IEEE Transactions on Neural Networks **6**(4) (July 1995) 1000–1004
4. Kohavi, R., Sommerfield, D., Doughterty, J.: Data mining using *MLC++*: A machine learning library in C++. In *Tools With AI* (1996) 234–245 http://www.sgi.com/Technology/mlc/docs.html
5. The Machine Learning Database Repository, University of California, Irvine http://www.ics.uci.edu/AI/ML/Machine-Learning.html
6. Tsoi, A. C., Back, A. D.: Locally recurrent globally feedforward networks: A critical review of architectures. IEEE Transactions on Neural Networks **5**(2) (March 1994) 229–239
7. Zell, A., Mamier, G., Vogt, M., et al.: SNNS Stuttgart Neural Network Simulator User Manual. Version 4.1, Report No. 6/95, University of Stuttgart (1995)