Available online at www.sciencedirect.com
ScienceDirect
Expert Systems with Applications 36 (2009) 2633–2644
Expert Systems with Applications
www.elsevier.com/locate/eswa

Effects of feature construction on classification performance: An empirical study in bank failure prediction

Huimin Zhao *, Atish P. Sinha [1], Wei Ge

Sheldon B. Lubar School of Business, University of Wisconsin-Milwaukee, P.O. Box 742, Milwaukee, WI 53201-0742, USA

Abstract

While extensive research in data mining has been devoted to developing better classification algorithms, relatively little research has been conducted to examine the effects of feature construction, guided by domain knowledge, on classification performance. However, in many application domains, domain knowledge can be used to construct higher-level features to potentially improve performance. For example, past research and regulatory practice in early warning of bank failures has resulted in various explanatory variables, in the form of financial ratios, that are constructed based on bank accounting variables and are believed to be more effective than the original variables in identifying potential problem banks. In this study, we empirically compare the performance of two sets of classifiers for bank failure prediction, one built using raw accounting variables and the other built using constructed financial ratios. Four popular data mining methods are used to learn the classifiers: logistic regression, decision tree, neural network, and *k*-nearest neighbor. We evaluate the classifiers on the basis of expected misclassification cost under a wide range of possible settings. The results of the study strongly indicate that feature construction, guided by domain knowledge, significantly improves classifier performance and that the degree of improvement varies significantly across the methods.

Keywords: Data mining; Classification; Feature construction; Bank failure prediction; Financial ratios

1. Introduction

Data mining techniques are employed by organizations to mine operational data for interesting patterns and decision models that could be used for decision support. Classification is an important type of predictive data mining problem where the dependent variable that needs to be predicted is categorical. Various classification techniques have been developed in fields, such as multivariate statistical analysis, machine learning, and artificial neural networks (Weiss & Kulikowski, 1991) and applied to solve classification problems in a variety of business domains, such as bankruptcy prediction (Kim & McLeod, 1999; Sung, Chang, & Lee, 1999) and credit evaluation (Sinha & May, 2005). These techniques automatically induce prediction models, called *classifiers*, based on historical data about previously solved problem cases. The classifiers can then be applied to recommend solutions to new problem cases.

While extensive research in data mining has been devoted to developing better classification algorithms, relatively little research has been conducted to examine the effects of feature construction guided by domain knowledge on classification performance (Dybowski, Laskey, Myers, & Parsons, 2003). However, in many application domains, substantial domain knowledge exists along with historical data. In such domains, that knowledge may be used to construct higher-level features based on the original features, potentially improving the performance of learned classification models. As an example, extensive past research and regulatory practice in bank failure prediction has identified various explanatory variables, in the form of financial ratios, that are constructed based on bank

* Corresponding author. Tel.: +1 414 229 6524; fax: +1 414 229 5999.
E-mail addresses: hzhao@uwm.edu (H. Zhao), sinha@uwm.edu (A.P. Sinha).
[1] Tel.: +1 414 229 3301; fax: +1 414 229 5999.

0957-4174/$ - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2008.01.053

accounting variables and are believed to be more effective than the original variables in predicting bank failures. On the other hand, blind application of data mining techniques—that is, treating data mining as a "black-box" process—prevents the incorporation of such domain-specific high-level concepts, which might ultimately lead to further performance improvement.

The objective of this study is to examine the effects of feature construction, guided by domain knowledge, on the performance of automatically learned classifiers. Specifically, we conduct an empirical study in the domain of bank failure prediction, comparing the performance of classifiers built using raw accounting variables with that of classifiers built using a set of financial ratios constructed from those raw variables. Our results show that the use of the financial ratios, instead of raw accounting variables, significantly improves the performance (measured in terms of expected misclassification cost) of classifiers learned using several widely used classification methods, including logistic regression, C4.5 decision tree, back-propagation neural network, and *k*-nearest neighbor. Our results also show that the effects of feature construction on classification performance vary across classification methods. The use of financial ratios improves the performance of logistic regression and back-propagation neural networks more than that of C4.5 decision tree and *k*-nearest neighbor.

Our study makes an important contribution to existing research in data mining by empirically demonstrating that feature construction, guided by domain knowledge, improves performance of classifiers built using different methods. This implies that the difficulty for various data mining methods to learn higher-level concepts is inherent in their induction biases and limitations. Also, instead of using classification accuracy or error rate as the performance measure—as has been the norm—we evaluate the classifiers with respect to expected misclassification cost under a wide range of settings of the prior probability of bank failure and the cost ratio between the two types of misclassification errors. Expected misclassification cost is a more appropriate measure of performance than overall accuracy or error rate in cost-sensitive classification problems (Elkan, 2001; Hand, Mannila, & Smyth, 2001; Provost, Fawcett, & Kohavi, 1998; Zhao, 2008), such as bank failure prediction, where misclassifying a problem bank as being healthy is considered by banking regulators much more costly than misclassifying a healthy bank as being unsound. The results are consistent across all the settings, indicating that the findings are relatively robust. We also find that the degree of performance improvement through feature construction depends on the data mining method. Therefore, a method that performs below par in a pure data mining situation should not be excluded from consideration when high-level features can be constructed based on domain knowledge.

The paper is organized as follows. We review the existing research related to feature construction and bank failure prediction in the next section. Next, in Section 3, we describe our research design and analyze the results. Section 4 concludes the paper and identifies a set of future research directions.

2. Background

In this section, we briefly review past research in feature construction and bank failure prediction.

2.1. Feature construction

The availability of numerous commercial data mining software products and research prototypes now enables someone who has little expertise in data mining and the business domains to quickly build classification models by simply providing training data and specifying model structures. However, applying data mining methods in real decision support practices demands much more than simply supplying data and manipulating a few parameters. Finding good solutions to practical classification problems requires not only the selection and specification of techniques and model structures, but also the selection and representation of input features. Data representation is often the most arduous task in data mining processes and is usually application dependent. The ultimate success of classification applications relies as much (if not more) on data representation as on the choice of learning techniques and model structures (Cherkassky & Lari-Najafi, 1992; Piramuthu, Ragavan, & Shaw, 1998; Radcliffe & Surry, 1995). In practice, the selection and representation of input features that best describe a given problem is often a tedious trial-and-error process.

Real-world datasets often contain a large number of features, some of which are either redundant or irrelevant to the given tasks (Fayyad, Piatetsky-Shapiro, & Smyth, 1996). This happens when it is unknown which features are relevant to a target concept and especially when domain knowledge is unavailable or incomplete. Many features are then introduced to represent an unknown domain. The presence of irrelevant and redundant features may mask or obscure the distribution of truly relevant features for a target concept and hence harm the performance of classification models (John, Kohavi, & Pfleger, 1994; Koller & Sahami, 1996; Ragavan, Rendell, Shaw, & Tessmer, 1993). In addition, increasing the dimensionality of the feature space will generally result in increased complexity of interactions among the features and increased degree of noise, making it difficult for classification methods to learn high-level concepts effectively and efficiently (Piramuthu et al., 1998). For a classification method like Naïve Bayes, selecting a subset of features may partially remedy the problems resulting from violations of its underlying assumption of conditional independence among features (Langley & Sage, 1994).

Besides feature irrelevancy and redundancy, the presence of feature interaction may be another source for classification performance degradation. Feature interaction

refers to a situation where some features are not individually related to the target concept, but are so when they are combined with other features in some manner (Markovitch & Rosenstein, 2002). For example, while a company's financial condition is often described in raw accounting terms, business experts usually use higher-level characterizations, such as leverage, liquidity, profitability, and growth rate, to evaluate the company's financial risk. When the low-level measurements cannot describe the target concept concisely, transforming the original feature space into a more appropriate representation by constructing higher-level features may enhance the learning process of classification methods (Zheng, 2000).

To address the problems of feature irrelevance, redundancy, and interaction, various approaches have been used for preprocessing the original features (Dash & Liu, 1997; Langley & Sage, 1994; Matheus & Rendell, 1989; Pakath & Zaveri, 1995; Piramuthu et al., 1998; Ragavan et al., 1993; Salcedo-Sanz, DePrado-Cumplido, Segovia-Vargas, Pérez-Cruz, & Bousoño-Calzón, 2004; Tahai, Walczak, & Rigsby, 1998). These approaches can be categorized into three types: feature selection, feature extraction, and feature construction (Liu & Motoda, 1998). Feature selection involves selecting a "good" subset of features that retain the most useful information for a given task. Feature extraction and feature construction involve finding a set of "composite" features, which are functions of the original features. Feature extraction projects a high-dimension feature space to a low-dimension space via linear/non-linear transformations such that most of the information in the original features is retained. Feature construction addresses the problem of feature interaction by discovering good combinations of the original features.

Many past studies have adopted a data-driven approach and focused on automating the process of searching for the best representation of input features (e.g., Langley & Sage 1994; Matheus & Rendell, 1989; Pakath & Zaveri 1995; Piramuthu et al. 1998; Ragavan et al. 1993; Tahai et al. 1998). A number of studies have combined feature-preprocessing algorithms with classification methods to improve their performance. For example, West (1985) extended a logit model by preprocessing features with factor analysis. The combined factor-logistic approach was shown to be more effective in predicting bank bankruptcy. Piramuthu et al. (1998) used feature construction algorithms to construct new feature sets and improved the performance of neural network for predicting bankruptcy and default loan payment.

Unfortunately, this process, especially feature construction, usually cannot be fully automated because of several reasons. First, the natural (original) representations of most business applications are not suitable for the learning techniques' inductive direction/bias. Second, an exhaustive search of the entire feature space for a give problem is practically infeasible. The number of possible feature subsets increases exponentially with respect to the number of original features (2^p possible subsets for p features). Further, the potential feature space is almost infinite if new features can be constructed based on the original features in many ways. While various feature-preprocessing algorithms have adopted greedy heuristics to construct and traverse feature spaces, such as stepwise selection and evolutionary strategy (e.g., genetic algorithm), it has been shown that no non-exhaustive heuristic can dominate others and guarantee to produce optimal solutions (Cover & Van Campenhout, 1977; Mucciardi & Gose, 1971; Murthy, 1998). Thus, effective and efficient feature construction typically requires domain or even application-specific knowledge.

Recognizing the importance of domain knowledge, a few studies have suggested that domain knowledge may play an important role in effective and efficient construction of input features (Feelders, Daniels, & Holsheimer, 2000; Tahai et al., 1998). However, prior research has not adequately examined the influence of higher-level domain-specific features vis-à-vis lower-level raw variables on the performance of data mining classifiers on cost-sensitive business problems.

2.2. Bank failure prediction

Maintaining a "safe and sound" banking system is a major responsibility of bank regulatory bodies of the United States, including the Office of Comptroller of the Currency (OCC), the Federal Reserve (Fed), and the Federal Deposit Insurance Corporation (FDIC). Because on-site examinations conducted by regulatory agents tend to be expensive and time-consuming, they are complemented by off-site monitoring of bank conditions. Classification models (called "early warning systems" by banking agencies) built to predict likely bank failures sufficiently in advance help regulators allocate scarce resources for on-site bank examinations and take appropriate actions.

Several bank failure prediction models have been developed since the mid 1970s. Most of the earlier models were built using classical statistical techniques, such as multivariate discriminant analysis (Looney, Wansley, & Lane, 1989; Pettway & Sinkey, 1980; Sinkey, 1975, 1977, 1978, 1979; Stuhr & Van Wicklen, 1974), logit regression (Gajewski, 1989; Martin, 1977; Thomson, 1991; West, 1985; Whalen & Thomson, 1988), factor analysis (West, 1985), simultaneous-equations model (Demirguc-Kent, 1989), and Cox proportional hazards model (Lane, Looney, & Wansley, 1986; Looney et al., 1989; Whalen, 1991). Later studies have also used neural networks (Swicegood & Clark, 2001; Tam & Kiang, 1992), split-population survival-time model (Cole & Gunther, 1995), Bayesian belief networks (Sarkar & Sriram, 2001), and isotonic separation (Ryu & Yue, 2005). Some of the models have been routinely applied in the regulatory practices of banking agencies.

Most of these models (e.g., Barr & Siems, 1994; Cole & Gunther, 1995; Gajewski, 1989; Korobow, Stuhr, & Martin, 1977; Lane et al., 1986; Looney et al., 1989; Martin, 1977; Pettway & Sinkey, 1980; Santomero & Vinso, 1977; Sarkar & Sriram, 2001; Sinkey, 1975, 1977, 1978, 1979;

Stuhr & Van Wicklen, 1974; Tam & Kiang, 1992; Thomson, 1991; West, 1985; Whalen, 1991; Whalen & Thomson, 1988) predict likely bank failures based on a set of high-level constructs called *financial ratios*, instead of low-level accounting variables. These financial ratios are usually constructed based on publicly available balance and income data (in the so-called call reports) that commercial banks are required to report to regulatory authorities on a regular basis. They are designed to reflect the soundness of a commercial bank in several aspects. Given the importance of the subject, extensive research has been devoted to the design and identification of such financial ratios in the last three decades. As a result, a large set of financial ratios has been identified and applied in regulatory practices. These financial ratios are believed to be more effective explanatory variables than the raw accounting data in the call reports in predicting and explaining bank failures.

Most of the financial ratios used in existing research can be classified into the categories of the CAMEL rating framework used by FDIC. CAMEL is an acronym for the five major characteristics of a bank's financial and operational conditions: *Capital adequacy*, *Asset quality*, *Management quality*, *Earnings ability*, and *Liquidity position*. FDIC developed the CAMEL rating system in the early 1970s to assist in their scheduling of on-site bank examinations. A bank's capital base is critical since it is the last line of defense against losses to uninsured depositors and general creditors. Capital adequacy is a measure of the level and quality of a bank's capital base. Asset quality measures the level of risk of a bank's assets. This is related to the quality and diversity of loan borrowers and their ability to repay the loans. Management quality is a measure of the quality of a bank's officers and the efficiency of its management structure. Earnings ability is a measure of the performance of a bank and the stability of its earnings stream. Liquidity measures a bank's ability to meet unforeseen deposit outflow in a short time. Each of these general characteristics in theory could have an impact on a bank's failure. While a bank's losses on assets are a direct cause of its failure, the other characteristics provide measures of the ability of the bank to remain operational in spite of these losses.

Problems closely related to bank failure prediction include bankruptcy prediction for general corporations (Altman, Marco, & Varetto, 1994; Anandarajan, Lee, & Anandarajan, 2001; Bryant, 1997; Mckee, 2000; Nanda & Pendharkar, 2001; OLeary, 1998; Pompe & Bilderbeek, 2005) or particular industries, such as insurance (Salcedo-Sanz et al., 2004). Studies in these areas have adopted similar approaches, although the particular financial ratios used are dependent on the domains.

3. Research design

The objective of this study is not to build yet another early warning system for bank failures or to identify another set of promising explanatory variables, but to investigate the effects of feature construction (in the form of constructed financial ratios) on the performance of classifiers learned using various classification methods. In this section, we describe the data sets and the experiment design in detail. We also briefly review the classification methods used to build classifiers. We then report on the results and discuss the findings.

3.1. Datasets

Because bank failure is a very rare event, most previous studies on bank failure prediction have adopted choice-based, relatively balanced samples, by using most available failed banks during a particular period and sampling a comparable number of surviving banks in the same period (e.g., Gajewski, 1989; Lane et al., 1986; Looney et al., 1989; Pettway & Sinkey, 1980; Sinkey, 1975, 1978, 1979; Tam & Kiang, 1992; Thomson, 1991; Whalen, 1991). As an experimental control, many studies have also attempted to match each failed bank with non-failed banks in several characteristics, such as geographic location, bank size, and charter type (e.g., Lane et al., 1986; Pettway & Sinkey, 1980; Sinkey, 1975; Tam & Kiang, 1992).

We followed these conventions in preparing the samples. The failed banks included all banks that filed bankruptcy at FDIC in 1991 and 1992. The 121 banks that failed in 1991 and the 119 others that failed in 1992 were identified from the FDIC website (www.fdic.gov). As an experimental control, a failed bank was matched with a non-failed bank based on three characteristics: (1) geographic location (i.e., state), (2) size of assets, and (3) charter type (federal chartered or state chartered). As a result, a balanced sample of 480 banks (240 failed and 240 non-failed) was generated for prediction one year prior to failure. The 468 banks (234 failed and 234 non-failed) among the 480 banks that had survived for at least two years were used in another sample for prediction two years prior to failure.

The sample for prediction two years prior to failure was used to test the robustness of the models since the earlier the prediction, the more uncertain it is. One-year-ahead and two-year-ahead models have been frequently used in previous studies (e.g., Barr & Siems, 1994; Lane et al., 1986; Looney et al., 1989; Tam & Kiang, 1992). While some studies have also built models with even longer prediction periods, such models are often not accurate enough to be practically useful and will not be examined in this study.

The sample data we used consisted of bank data one year or two years prior to failure (depending on the prediction period) and were obtained from the Federal Reserve database. As the primary goal of this study was to examine the effects of feature construction on classifier performance, two separate data sets were prepared for each sample using different sets of features. In the data set for the control group, each bank was described by 93 raw accounting variables directly available from the internal call reports in the Federal Reserve database, representing the scenario with

no feature construction. Six other variables in the call reports were excluded because they were intuitively irrelevant to the classification problem (e.g., address, phone number, and ID) or had too many missing values. In the data set for the experiment group, each bank was described by a set of financial ratios we identified from previous studies, representing the scenario with feature construction guided by domain knowledge.

A comprehensive review of the bank failure prediction literature has revealed that over a hundred financial ratios constructed based on raw accounting variables have been used to measure the five CAMEL components (e.g., Barr & Siems, 1994; Cole & Gunther, 1995; Gajewski, 1989; Korobow et al., 1977; Lane et al., 1986; Looney et al., 1989; Martin, 1977; Pettway & Sinkey, 1980; Santomero & Vinso, 1977; Sarkar & Sriram, 2001; Sinkey, 1975, 1977, 1978, 1979; Stuhr & Van Wicklen, 1974; Tam & Kiang, 1992; Thomson, 1991; West, 1985; Whalen, 1991; Whalen & Thomson, 1988). While most studies have followed the CAMEL framework in general, the specific measures they have used are often substantially different. One major difference between the measures used in different studies is in their complexity. For example, in measuring asset quality, *portfolio diversification level* can be measured by the ratios of the loans in individual industries to the total loans. Some studies, however, have developed much more complex measures, such as *loan portfolio Herfindahl index*, which is a sum of the squares of these ratios.

While these different forms of measures all represent certain domain knowledge, in this study, we have chosen to use those measures in the simplest ratio forms—in which both the numerator and the denominator are single raw accounting variables directly available from the call reports in the Federal Reserve database—with only one exception for liquidity, for which most previous studies have used complex measures. We deliberately chose the simple financial ratios in order to demonstrate that even feature construction as simple as these ratios may significantly improve the performance of learned classifiers. As a result, 26 financial ratios (presented in Table 1) were selected as input features for bank failure prediction, representing a scenario with feature construction. This set of financial ratios can be interpreted as a full set of measures of the five CAMEL components.

A closer examination of these constructed features reveals some interesting patterns of how domain knowledge is represented through different combinations of raw accounting variables. For example, among the 26 constructed features, nine of them are constructed by dividing raw accounting variables by total assets and eight of them are constructed by dividing the variables of loan-specific assets by gross loans. These constructed features in some sense reflect preliminary domain knowledge of *normalization*. The goal of normalization is to eliminate the effects of some irrelevant factors in describing a company's financial condition. For example, net income divided by total assets represents a company's earnings ability better than net income alone, because the effect of the company's size is eliminated. Similarly, late loan divided by gross loan is a better indicator of a bank's loan quality than late loan alone.

Table 1
Selected financial ratios for bank failure prediction

No	Dimension	Attribute	Definition
1	Capital adequacy	C_eqas	Total_Equity/Total_Assets
2	Asset quality	A_loas	Gross_Loans/Total_Assets
3		A_colo	Comm_Loans/Gross_Loans
4		A_inlo	Indi_Loans/Gross_Loans
5		A_relo	Real_Loans/Gross_Loans
6		A_lalo	Loan_Late90/Gross_Loans
7		A_aclo	Loan_notAccruing/Gross_Loans
8		A_lolo	Loan_LossProvision/Gross_Loans
9		A_chlo	Charge_off_Loan/Gross_Loans
10		A_allo	Loan_allowance/Gross_Loans
11		A_loeq	Total_Loans_NetofUnearned/ Total_Equity
12	Management	M_inas	NetIncome/Operating_Income
13		M_exas	NetIncome_before_Extra / Total_Assets
14	Earnings ability	E_exas	Operating_exp/Total_Assets
15		E_inas	Operating_income/Total_Assets
16		E_inex	Interest_Income/Interest_Exp
17		E_opinopex	Operating_Income/Operating_Exp
18		E_inas	NetIncome/Total_Assets
19		E_ineq	NetIncome/Total_Equity
20		E_exde	Interest_Exp_Dep/Total_Deposits
21		E_inlo	Interest_Income_Loans/ Gross_Loans
22		E_itin	Interest_Income/ Operating_Income
23	Liquidity position	L_caas	Cash/Total_Assets
24		L_seas	(Cash+Securities_M)/Total_Assets
25		L_feas	(Cash+ FedFunds_Sold+ USTreasury+USGoverOblig)/ Total_Assets
26		L_lode	Gross_Loans /Total_Deposits

While the 26 financial ratios are all in relatively simple forms, they constitute important domain knowledge, which is not explicitly captured in, and cannot be automatically learned from, the raw accounting data. Without some knowledge of the financial domain, even a data mining specialist would not know how to combine different raw accounting variables in meaningful ways to construct such intermediate concepts.

In summary, four data sets were used, two for prediction one year prior to failure and the other two for prediction two years prior to failure. There were 480 banks (240 failed and 240 non-failed) in the data sets for one-year-ahead prediction and 468 banks (234 failed and 234 non-failed) in the data sets for two-years-ahead prediction. For each prediction period (one year or two years), a data set with 93 raw accounting variables and a data set with 26 constructed financial ratios were used for comparison. In each data set, a bank is described by a dependent variable $\mathbf{y}$ ($\mathbf{y} = 1$ if the bank failed and 0 otherwise) and a sequence of independent variables $x = \langle x_1, x_2, \ldots, x_m \rangle$.

3.2. Classification methods

In general, a binary classification problem can be described by a vector of independent variables, $x = \langle x_1, x_2, ..., x_m \rangle$, called *features* or *attributes*, and a binary dependent variable, **y**, called *class*. A classification algorithm is given a training data set, consisting of solved problem *cases* (also referred to as *instances* or *examples*) whose class memberships are already known, and generalizes the training data into a general predictive model, $\hat{y} = f(x)$, called a *classifier*. This learned classifier can then be used to predict the class memberships of other problem cases not used in training the classifier. While numerous classification techniques have been developed in such fields as multivariate statistical analysis, machine learning, and artificial neural networks, we selected four widely used techniques—*logistic regression*, *C4.5 decision tree*, *back-propagation neural network*, and *k-nearest neighbor*—available in the Weka data mining software package (Witten & Frank, 2005) to build classifiers in this study.

Logistic regression (LR) is a widely used statistical method for classification. It assumes that the logarithm of the *odds ratio* (called *logit*) is linear with regard to the attributes (Hosmer & Lemeshow, 2000), i.e.,

$$g(x) = \ln \frac{P(y=1|x=x)}{P(y=0|x=x)} = \sum_{j=1}^{m} \beta_j x_j + \beta_0.$$

The decision boundary that separates the classes in the attribute space is linear. The coefficients, $\beta_j (j = 0, 1, 2, \ldots, m)$, can be found using an *iterative weighted least squares procedure*, such that the two classes of training examples are maximally separated.

Decision tree techniques follow a "divide and conquer", recursive partitioning heuristic search strategy and generate tree-like sequential decision models. Most decision tree inducers assume that the overall prediction decision can be made via a sequence of small tests (or decisions), each of which usually involves a single attribute x_i. Different decision tree inducers mainly differ in the goodness measure used to select the splitting attribute at each intermediate tree node. For example, a popular decision tree inducer named ID3 selects the attribute that results in the maximum *information gain* (or *entropy reduction*) (Quinlan, 1986). C4.5, a successor of ID3, replaces information gain with *gain ratio* to compensate ID3's bias of favoring highly-branching attributes (Quinlan, 1993). The tree building procedure is usually followed by a *pruning* phase, in which some of the sub-trees are replaced by leaves or raised to substitute their parents, to reduce the chance of *overfitting*, a situation where a learned classifier classifies the training data accurately but does not generalize well to test data (Murthy, 1998). Pruning reduces the performance of the learned decision tree on training data but may increase the true performance of the decision tree during prediction.

Back-propagation is one of the most widely used neural network learning techniques for classification (Chauvin & Rumelhart, 1995; Rumelhart, Hinton, & Williams, 1986). Neural networks are highly interconnected networks, which learn by adjusting the weights of the connections between nodes on different layers. A back-propagation neural network has an *input layer* (corresponding to the attribute vector **x**), an *output layer* (corresponding to the class **y**), and possibly one or more *hidden layers* between the input layer and the output layer. The input layer generates a set of linear terms, each a weighted sum of the input variables. The linear terms are then non-linearly transformed (e.g., using the Sigmoid function) to form the nodes on the first hidden layer. This process of nonlinear transformation of weighted sums of the inputs is repeated through multiple layers up to the output layer, where the final classification decision is made. Such multiple layers of nonlinear transformations allow very flexible decision boundaries to be formed such that the training data can be accurately discriminated. However, this high flexibility also implies a high risk of overfitting. An appropriate degree of model flexibility (mainly defined by the number of hidden layers and the number of nodes on each hidden layer) needs to be chosen such that the learned neural network can not only discriminate the training data well but also predict the outcomes of test cases relatively accurately. The default setting used in Weka is based on a widely used rule of thumb, that is, there is one hidden layer and the number of hidden nodes is the average of the number of input nodes and the number of output nodes.

Unlike the methods described above, *k*-nearest neighbor is an instance-based method (Aha & Kibler, 1991) and does not learn a model in the true sense. It assigns a new case to the majority class among the *k*-closest cases in the training set (Hand et al., 2001). The objective is to identify the cases that are similar to the new case and then classify the new case based on the outcome of the majority of its neighbors. There are three major components of a nearest-neighbor solution: the set of stored cases; the distance metric used to compute the distance between cases; and the value of *k* (Weiss & Indurkhya, 1998). The default distance metric used in Weka is the Euclidean distance metric. If $x' =< x'_1, x'_2, \ldots, x'_m >$ is the input vector corresponding to the new case, and $x =< x_1, x_2, \ldots, x_m >$ is the input vector for a stored case, then the squared Euclidean distance between them is $\sum_{i=1}^{m} (x_i - x'_i)^2$.

For more detailed information about these classification techniques, the reader is referred to (Aha & Kibler, 1991; Chauvin & Rumelhart, 1995; Hosmer & Lemeshow, 2000; Quinlan, 1993; Rumelhart et al., 1986). For more detailed information about the Weka data mining package, the reader is referred to (Witten & Frank, 2005). We used the default parameter settings of Weka for all the chosen classification methods. All the results about *k*-nearest neighbor reported later are results of $k = 3$. Because the results are similar across several values of *k*, including 1, 3, and 5, we only report on the results of $k = 3$.

3.3. Experimental design

We used a factorial design with two factors. The first factor, feature construction, had two levels corresponding with whether feature construction was performed (i.e., the 93 raw accounting variables were used) or not (i.e., the 26 constructed financial ratios were used). The second factor, classification method, had four levels corresponding with the four chosen classification techniques.

The performance of each learned classifier was measured in terms of (normalized) expected misclassification cost. Forecast of likely bank failures is subject to two types of errors. A *false negative* error is made when a classifier misclassifies an actual failure as a survivor. A *false positive* error is made when a classifier misclassifies an actual survivor as a failure. The expected misclassification cost of a classifier f is defined as

$$\mathrm{Cost}(f) = C_{10} \cdot p \cdot p_{10} + C_{01} \cdot (1-p) \cdot p_{01},$$

where p is the prior probability for failure, p_{10} and p_{01} are false negative and false positive error rates, respectively, and C_{10} and C_{01} are the unit costs of false negative and false positive errors, respectively. Since the assignment of C_{10} and C_{01} is often subjective, the ratio between the two costs is more meaningful than their absolute values. Let $r = \frac{C_{10}}{C_{01}}$ denote the cost ratio, a normalized measure for expected misclassification cost, which falls into the range of [0, 1], can be defined as

$$\mathrm{Cost}'(f) = \frac{\mathrm{Cost}(f)}{C_{10} \cdot p + C_{01} \cdot (1-p)} = \frac{r \cdot p \cdot p_{10} + (1-p) \cdot p_{01}}{r \cdot p + (1-p)}.$$

Note that classification accuracy (or inversely, error rate) that has been frequently used in previous empirical studies can be considered as a special case of expected misclassification cost when the prior probabilities of the two classes (failed and non-failed) as well as the misclassification costs of the two types of errors are equal (i.e., $p = 0.5$, $r = 1$). However, this assumption does not hold at all in the case of bank failure prediction. First, bank failures are usually very rare. Second, misclassifying an actual failure as a survivor (false negative) is considered by banking regulators to be far more costly than misclassifying an actual survivor as a failure (false positive). While a false negative error would allow a potential problem bank to pass by unnoticed, with severe consequences if the bank eventually fails, a false positive error would only result in a healthy bank being placed in the on-site examination queue. As long as only a few healthy banks are scheduled for early examinations, the cost of false positive errors would be much smaller than the benefits derived from early warning of potential failures (Looney et al., 1989; Pettway & Sinkey, 1980). Gajewski (1989) estimated that the average failure probability in 1986 was close to 0.98%. Sinkey (1979) estimated that the prior probability of a commercial bank becoming a problem institution was about 2% on average and that the cost ratio of the two types of errors was approximately 58-to-1 (false negative to false positive). Frydman, Altman, and Kao (1985) used a prior probability for failure of 0.02 and eight cost ratios, 1, 10, 20, 30, 40, 50, 60, and 70, in their study. Tam and Kiang (1992) used two prior probabilities for failed banks, 0.01 and 0.02, and eight cost ratios, 1, 5, 25, 40, 50, 60, 75, and 100, in another study.

Following these conventions, we used two prior probabilities for failed banks, 0.01 and 0.02, and ten cost ratios, 10, 20, 30, ..., 100. We object to the use of the cost ratio of 1 because it is obviously unrealistic. Indeed, under such a setting, virtually no classification method can learn a classifier that performs significantly better than a naïve classifier that simply classifies every bank as a survivor. With a given setting of prior probability of failure and cost ratio, the classification methods in Weka can re-weight the two types (positive and negative) of cases accordingly and attempt to minimize the expected misclassification cost instead of overall error rate (Witten & Frank, 2005). Note that two settings with equal weight ratio $\frac{pr}{(1-p)}$ are equivalent in terms of the normalized expected misclassification cost used in this study. We believe that our choices cover a sufficiently wide range of possible settings.

Because the performance of a classifier on training data—known as *apparent performance* and *re-substitution performance*—is usually overly optimistic and not reliable, we used stratified tenfold cross-validation, a highly recommended performance estimation technique (Kohavi, 1995), to estimate the performance of learned classifiers. This technique randomly splits (with stratification) the full data set into ten subsets called folds. One of the ten folds is used for estimating the performance of a classifier learned using the remaining nine folds. This process is repeated ten times, each time with one fold as the hold-out test set. The average over the ten runs is used as an overall performance estimate.

We used a 2×4 factorial ANOVA for each prediction period under each setting of prior probability of failure and cost ratio. We expected to detect a relatively large effect size (Cohen, 1988) on the main factor of feature construction and the interaction between feature construction and classification method. Setting power = 0.8, significance level = 0.05, Cohen's $f = 0.4$, we needed a sample size of 58 and 76 for detecting the effects of feature construction and the interaction term, respectively, according to Cohen (1988). We therefore collected 10 data points for each of the 2×4 cells of the ANOVA, resulting in a sample size of 80. We estimated the expected misclassification cost of learned classifiers using 10 independent stratified tenfold cross validations for each of the 8 cells, each time with a different random seed.

3.4. Empirical results

Table 2 summarizes the means of normalized expected misclassification cost of the classifiers learned using different classification methods with or without feature construction under various settings of prediction period, prior

Table 2
Means of expected misclassification cost of the learned classifiers

Period(year)			Without feature construction				With feature construction			
	p	r	LR	Tree	Neural	k-NN	LR	Tree	Neural	k-NN
1	0.01	10	0.087	0.076	0.091	0.071	0.053	0.063	0.057	0.057
		20	0.109	0.104	0.150	0.098	0.065	0.079	0.077	0.083
		30	0.133	0.122	0.196	0.121	0.071	0.095	0.092	0.104
		40	0.153	0.133	0.227	0.140	0.073	0.102	0.097	0.123
		50	0.164	0.138	0.245	0.165	0.077	0.116	0.109	0.138
		60	0.173	0.141	0.251	0.166	0.077	0.121	0.112	0.140
		70	0.175	0.146	0.260	0.167	0.078	0.127	0.114	0.142
		80	0.180	0.145	0.257	0.168	0.080	0.134	0.113	0.143
		90	0.185	0.147	0.250	0.168	0.080	0.131	0.114	0.145
		100	0.185	0.148	0.241	0.169	0.081	0.131	0.115	0.146
	0.02	10	0.115	0.103	0.151	0.098	0.065	0.079	0.077	0.083
		20	0.152	0.132	0.228	0.141	0.074	0.103	0.100	0.124
		30	0.170	0.141	0.252	0.166	0.078	0.121	0.109	0.140
		40	0.179	0.145	0.258	0.168	0.080	0.135	0.114	0.143
		50	0.186	0.148	0.241	0.169	0.080	0.132	0.114	0.146
		60	0.188	0.149	0.229	0.169	0.080	0.129	0.112	0.149
		70	0.185	0.143	0.229	0.170	0.083	0.124	0.113	0.151
		80	0.182	0.143	0.226	0.171	0.084	0.122	0.110	0.152
		90	0.180	0.142	0.227	0.171	0.084	0.121	0.107	0.154
		100	0.178	0.146	0.225	0.185	0.086	0.115	0.108	0.175
2	0.01	10	0.138	0.109	0.099	0.110	0.075	0.096	0.091	0.110
		20	0.187	0.183	0.168	0.150	0.112	0.149	0.129	0.144
		30	0.220	0.225	0.225	0.183	0.135	0.191	0.164	0.173
		40	0.247	0.254	0.274	0.212	0.161	0.215	0.176	0.198
		50	0.265	0.256	0.325	0.293	0.178	0.234	0.189	0.233
		60	0.276	0.271	0.356	0.293	0.183	0.243	0.204	0.236
		70	0.284	0.280	0.373	0.294	0.184	0.248	0.213	0.239
		80	0.285	0.283	0.401	0.294	0.184	0.256	0.222	0.241
		90	0.286	0.286	0.414	0.295	0.186	0.256	0.227	0.243
		100	0.282	0.293	0.410	0.295	0.189	0.256	0.231	0.245
	0.02	10	0.187	0.183	0.170	0.150	0.113	0.150	0.131	0.145
		20	0.249	0.254	0.279	0.213	0.162	0.216	0.176	0.199
		30	0.277	0.272	0.356	0.293	0.183	0.244	0.204	0.236
		40	0.287	0.284	0.402	0.294	0.184	0.257	0.222	0.241
		50	0.283	0.293	0.417	0.295	0.189	0.255	0.233	0.245
		60	0.278	0.295	0.407	0.295	0.192	0.250	0.239	0.249
		70	0.270	0.294	0.376	0.296	0.197	0.243	0.235	0.251
		80	0.264	0.298	0.356	0.296	0.199	0.243	0.240	0.254
		90	0.255	0.293	0.345	0.297	0.197	0.232	0.235	0.256
		100	0.253	0.289	0.330	0.293	0.198	0.230	0.236	0.272

probability of failure, and cost ratio. Under every setting, for every classification method, the classifier learned with feature construction has a lower expected misclassification cost than the corresponding classifier learned without feature construction.

The ANOVA results show that the use of the financial ratios, instead of raw accounting variables, significantly improves the performance ($p < 0.01$), with respect to expected misclassification cost, of classifiers learned using several widely used classification methods, including logistic regression, C4.5 decision tree, back-propagation neural network, and k-nearest neighbor. The results hold across all settings of prediction period, prior probability of failure, and cost ratio. The results imply that feature construction, guided by domain knowledge, significantly improves classification performance. In a different context, Berry and Trigueiros (1993) argued that it may be preferable to allow a neural network to form its own internal abstractions on the hidden layer based on raw accounting variables. However, in our experiment, neural network also benefited from the pre-constructed financial ratios.

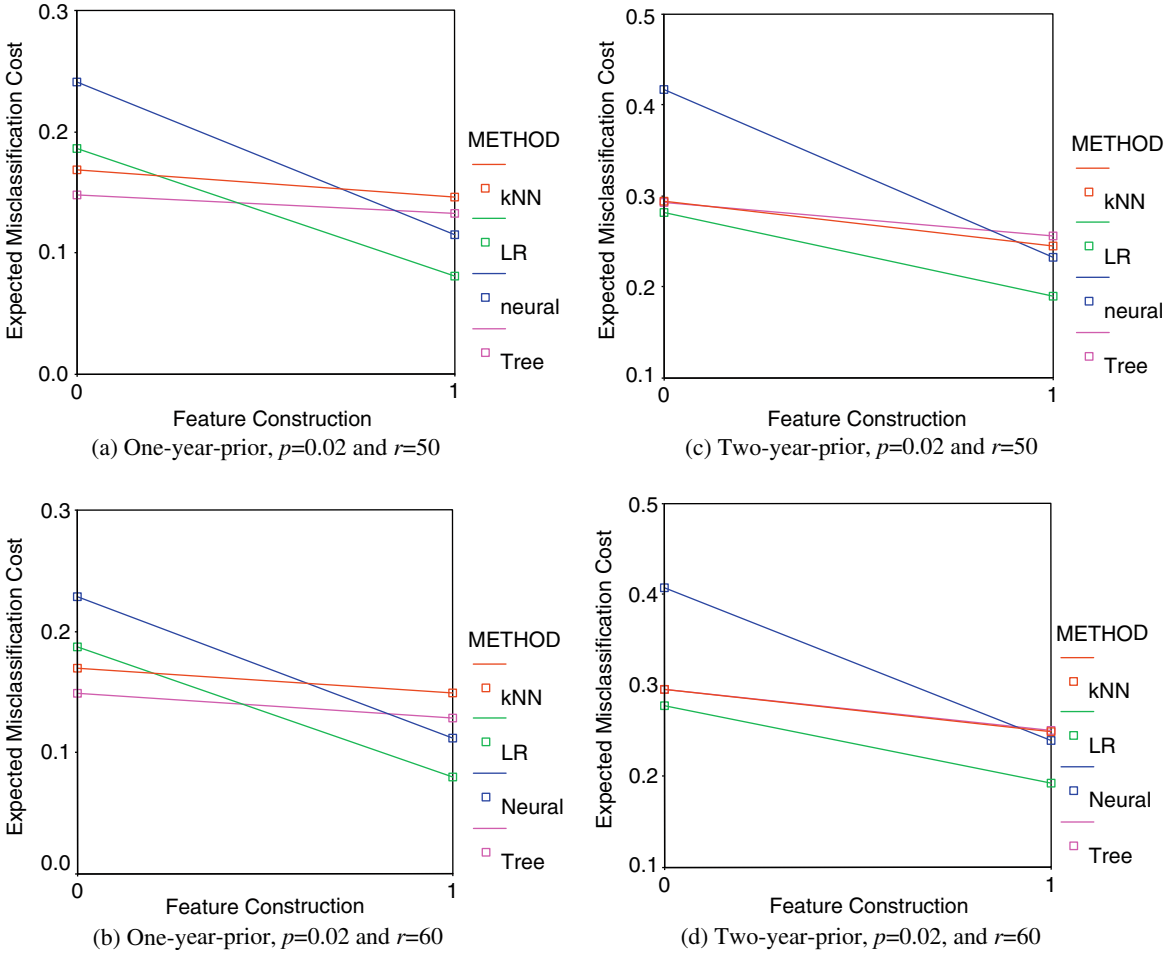
The results also show that the performance across different classification methods is significantly different ($p < 0.001$). The results hold across all settings of prediction period, prior probability of failure, and cost ratio. Previous empirical studies (e.g., Tam & Kiang, 1992; Weiss & Kapouleas, 1989) in various application domains have also found differences in the performance of different classification methods. However, the results are not consistent across domains. It has been suggested (Weiss & Kulikowski, 1991) that the results are dependent on the problem context. The "no-free-lunch theorems" (Wolpert, 1994) also suggest that a universally superior classification method may not exist.

More importantly, the results also show that the effect of feature construction (i.e., financial ratios) on classifica-

tion performance varies significantly across classification methods ($p < 0.001$). The use of financial ratios improves logistic regression and back-propagation neural networks more than C4.5 decision tree and *k*-nearest neighbor. The results hold across all settings of prediction period, prior probability of failure, and cost ratio.

Since we have performed 40 independent ANOVA tests, one may be concerned with the inflation of the significance level. However, even with a conservative adjustment of the significance level based on the first-order Bonferroni inequality (i.e., multiplying each observed significance level by the number of independent tests) (Hand et al., 2001), the effects of the two factors and the interaction between the two factors are still statistically significant at the 0.05 level, under every setting of prediction period, prior probability of failure, and cost ratio. In addition, the results across all settings are similar, indicating that the findings of the study are relatively robust. While the expected misclassification costs of the two-years-ahead classifiers are higher than that of the corresponding one-year-ahead classifiers, the effects of feature construction and the interaction between feature construction and classification method are similar for both prediction periods.

Fig. 1 illustrates the results under two particular settings that are the closest to Sinkey's estimation (1979), i.e., prior probability of failure is 0.02 and cost ratio is 50 or 60. In both settings and both prediction periods (one year and two year), the following results hold: (1) Feature construction reduces expected misclassification cost for every classification method. (2) The degree of cost reduction depends on the classification method. The cost reduction on logistic regression and back-propagation neural network is larger than that on *k*-nearest neighbor and C4.5 decision tree. Without feature construction, the performance of back-propagation neural network is worse than that of *k*-nearest neighbor and C4.5 decision tree. With feature construction, the performance of back-propagation neural network becomes better than that of *k*-nearest neighbor and C4.5 decision tree. Without feature construction, the performance of logistic regression is worse than (in one-year-prior prediction) or close to (in two-year-prior prediction) that of *k*-nearest neighbor and C4.5 decision tree. With feature construction, the performance of logistic regression becomes much better than that of *k*-nearest neighbor and C4.5 decision tree. With feature construction, the top two performers are logistic regression and back-propagation neural network.



(a) One-year-prior, p=0.02 and r=50

(c) Two-year-prior, p=0.02 and r=50

(b) One-year-prior, p=0.02 and r=60

(d) Two-year-prior, p=0.02, and r=60

Fig. 1. Performance of the classifiers under several settings.

While it has been suggested that domain knowledge may play an important role in effective and efficient construction of high-level features (Feelders et al., 2000; Tahai et al., 1998), the effects of feature construction, guided by domain knowledge, on classification performance of data mining methods in business applications have not been adequately examined in previous research. On the other hand, most of the previous studies on bank failure prediction have used financial ratios, which are constructed high-level concepts known to domain experts, but these studies have not evaluated the performance improvement due to the use of such constructed features rather than the original features. Our study provides empirical evidence that feature construction guided by domain knowledge can lead to significant performance improvement of data mining classifiers within the context of a cost-sensitive business problem.

4. Conclusion and future research directions

We have empirically examined the effects of feature construction guided by domain knowledge on classifier performance in the domain of bank failure prediction. Our study makes an important contribution to existing research in data mining by empirically demonstrating that constructed high-level features, such as financial ratios, can significantly improve the performance of classifiers built using different methods. It is therefore important that future research in data mining not only study feature construction, but also address the broader issue of the fusion of data mining and domain knowledge. We also found that the degree of performance improvement through feature construction is not equal among the classifiers, but depends on the data mining method used. Therefore, a method that performs below par in a pure data mining situation should not be excluded from consideration when higher-level features can be constructed based on domain knowledge.

Our study opens up several avenues for future research. First, the generalizability of our findings can be validated in other business domains, where both domain knowledge and historical data are available for building prediction models. Second, we focused on a binary classification task in this study, but future studies could examine if the results extend to multiple-class prediction tasks and regression tasks, such as sales forecasting. Third, more studies are needed in different domains to understand what other types of domain knowledge could influence performance. In the domain of bank failure prediction, the form of domain knowledge (i.e., financial ratios) is relatively simple. In other domains, the form of domain knowledge can be as complex as sophisticated expert systems. Different ways for incorporating domain knowledge need to be explored depending on particular applications. Since we found that even simple domain knowledge tends to significantly improve classification performance, more complex knowledge available in other domains could be even more promising in improving performance.

References

Aha, D., & Kibler, D. (1991). Instance-based learning algorithms. *Machine Learning, 6*(1), 37–66.

Altman, E. I., Marco, G., & Varetto, F. (1994). Corporate distress diagnosis: Comparisons using linear discriminant analysis and neural networks (the Italian experience). *Journal of Banking and Finance, 18*, 505–529.

Anandarajan, M., Lee, P., & Anandarajan, A. (2001). Bankruptcy prediction of financially stressed firms: An examination of the predictive accuracy of artificial neural networks. *International Journal of Intelligent Systems in Accounting, Finance & Management, 10*(2), 69–81.

Barr, R. S., & Siems, T. E. (1994). Predicting bank failure using DEA to quantify management quality. Financial Industry Studies Working Paper, Federal Reserve Bank of Dallas (pp. 1–31).

Berry, R. H., & Trigueiros, D. (1993). Applying neural networks to the extraction of knowledge from accounting reports: A classification study. In Trippi, R. R. & Turban, E. (Eds.) Neural networks in finance and investing (pp. 103–123). Chicago, IL: Probus Publishing.

Bryant, S. M. (1997). A case-based reasoning approach to bankruptcy prediction modeling. *International Journal of Intelligent Systems in Accounting Finance & Management, 6*(3), 195–214.

Chauvin, Y., & Rumelhart, D. E. (1995). *Backpropagation: Theory, Architectures, and Applications*. Hillsdale NJ: Lawrence Erlbaum Association.

Cherkassky, V., & Lari-Najafi, H. (1992). Data representation for diagnostic neural networks. *IEEE Expert, 7*(5), 43–53.

Cohen, J. (1988). *Statistical power analysis for the behavioral sciences* (2nd ed.). Hillsdale, NJ: Erlbaum.

Cole, R., & Gunther, J. (1995). Separating the likelihood and timing of bank failure. *Journal of Banking and Finance, 19*(6), 1073–1089.

Cover, T. M., & Van Campenhout, J. M. (1977). On the possible orderings in the measurement selection problem. *IEEE Transactions on System, Man and Cybernetics, 7*(9), 657–661.

Dash, M., & Liu, H. (1997). Feature selection for classification. *Intelligent Data Analysis, 1*, 131–156.

Demirgue-Kent, A. (1989). Modeling large commercial-bank failures: A simultaneous equation analysis. Working paper 8905, Federal Reserve Bank of Cleveland: Cleveland, OH.

Dybowski, R., Laskey, K. B., Myers, J. W., & Parsons, S. (2003). Introduction to the special issue on the fusion of domain knowledge with data for decision support. *Journal of Machine Learning Research, 4*, 293–294.

Elkan, C. (2001). The foundations of cost-sensitive learning. In *proceedings of the 17th international joint conference on artificial intelligence* (pp. 973–978), Seattle, WA.

Fayyad, U., Piatetsky-Shapiro, G., & Smyth, P. (1996). The KDD process for extracting useful knowledge from volumes of data. *Communications of ACM, 39*(11), 27–34.

Feelders, A., Daniels, H., & Holsheimer, M. (2000). Methodological and practical aspects of data mining. *Information and Management, 37*, 271–281.

Frydman, H., Altman, E., & Kao, D. (1985). Introducing recursive partitioning for financial classification: the case of financial distress. *Journal of Finance, 40*(1), 269–291.

Gajewski, G.R. (1989). Assessing the risk of bank failure. In *Proceedings from a conference on bank structure and competition* (pp. 432–456), Federal Reserve Bank of Chicago, Chicago, IL.

Hand, D., Mannila, H., & Smyth, P. (2001). *Principles of data mining*. Cambridge, MA: MIT Press.

Hosmer, D. W., & Lemeshow, S. (2000). *Applied logistic regression* (2nd ed.). John Wiley & Sons Inc.

John, G.H., Kohavi, R., & Pfleger, K. (1994). Irrelevant features and subset selection problem. In *proceedings for the 11th international conference on machine learning*.

Kim, C. N., & McLeod, R. (1999). Expert, linear models, and nonlinear models of expert decision making in bankruptcy prediction: A lens

model analysis. *Journal of Management Information Systems, 16*(1), 189–206.

Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In C. S. Mellish (Ed.), *Proceedings of the 14th international joint conference on artificial intelligence* (pp. 1137–1143). San Mateo, CA: Morgan Kaufmann.

Koller, D., & Sahami, M. (1996). Toward optimal feature selection. In *Proceedings for the 13th international conference on machine learning*.

Korobow, L., Stuhr, D.P., & Martin, D. (1977). A nationwide test of early warning research in banking. Federal Reserve Bank of New York, Quarterly Review, Autumn, (pp. 37–52).

Lane, W. R., Looney, S. W., & Wansley, J. W. (1986). An application of the cox proportional hazards model to bank failure. *Journal of Banking and Finance, 10*, 511–531.

Langley, P., & Sage, S. (1994). Induction of selective Bayesian classifiers. In *proceedings of the 10th conference on uncertainty in artificial intelligence*.

Liu, H., & Motoda, H. (1998). *Feature extraction, construction, and selection: A data ming perspective*. Kluwer.

Looney, S. W., Wansley, J. W., & Lane, W. R. (1989). An examination of misclassification with bank failure prediction models. *Journal of Economics and Business, 41*, 327–336.

Markovitch, S., & Rosenstein, D. (2002). Feature generation using general constructor functions. *Marching Learning, 49*(1), 59–98.

Martin, D. (1977). Early warning of bank failure: Logit regression approach. *Journal of Banking and Finance, 1*, 249–276.

Matheus, C.J., & Rendell, L. (1989). Constructive induction in decision trees. In *Proceedings of 11th IJCAI*.

Mckee, T. E. (2000). Developing a bankruptcy prediction model via rough sets theory. *International Journal of Intelligent Systems in Accounting Finance & Management, 9*(3), 159–173.

Mucciardi, A. N., & Gose, E. E. (1971). A comparison of seven techniques for choosing subsets of pattern recognition properties. *IEEE Transactions on Computer, C-20*, 1023.

Murthy, S. K. (1998). Automatic construction of decision trees from data: A multi-disciplinary survey. *Data Mining and Knowledge Discovery, 2*, 345–389.

Nanda, S., & Pendharkar, P. (2001). Linear models for minimizing misclassification costs in bankruptcy prediction. *International Journal of Intelligent Systems in Accounting Finance & Management, 10*(3), 155–168.

OLeary, D. E. (1998). Using neural networks to predict corporate failure. *International Journal of Intelligent Systems in Accounting, Finance & Management, 7*(3), 187–197.

Pakath, R., & Zaveri, J. S. (1995). Specifying critical inputs in a genetic algorithm-driven decision support system: an automated facility. *Decision Sciences, 26*(6), 749–779.

Pettway, R. H., & Sinkey, J. Jr., (1980). Establishing on-site bank examination priorities: An early warning system using accounting and marketing information. *Journal of Finance, 35*(1), 137–150.

Piramuthu, S., Ragavan, H., & Shaw, M. J. (1998). Using feature construction to improve performance of neural networks. *Management Science, 44*(3), 416–430.

Pompe, P. P. M., & Bilderbeek, J. (2005). Bankruptcy prediction: the influence of the year prior to failure selected for model building and the effects in a period of economic decline. *International Journal of Intelligent Systems in Accounting, Finance and Management, 13*(2), 95–112.

Provost, F., Fawcett, T., & Kohavi, R. (1998). The case against accuracy estimation for comparing induction algorithms. In *proceedings of the 15th international conference on machine learning* (pp. 445–453), Madison, WI.

Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning, 1*, 81–106.

Quinlan, J.R. (1993). C4.5: Programs for Machine Learning. Morgan Kaufmann.

Radcliffe, N. J., & Surry, P. D. (1995). Fundamental limitations on search algorithms: Evolutionary computing in perspective. In Jan Van Leeuwen (Ed.), *Computer Science Today: Recent Trends and Developments: Lecture Notes in Computer Science*. London: Springer-Verlag.

Ragavan, H., Rendell, M., Shaw, M., & Tessmer, A. (1993). Complex concept acquisition through directed search and feature caching, and practical results in a financial domain. In *Proceedings of 13th IJCAI*.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning internal representations by error propagation. In D. E. Rumelhart & J. L. McClelland (Eds.), *Parallel distributed processing* (pp. 318–362). Cambridge, MA: MIT Press.

Ryu, Y. U., & Yue, W. T. (2005). Firm bankruptcy prediction: Experimental comparison of isotonic separation and other classification approaches. *IEEE Transactions on Systems, Man, and Cybernetics: Part A, 35*(5), 727–737.

Salcedo-Sanz, S., DePrado-Cumplido, M., Segovia-Vargas, M. J., Pérez-Cruz, F., & Bousoño-Calzón, C. (2004). Feature selection methods involving support vector machines for prediction of insolvency in non-life insurance companies. *International Journal of Intelligent Systems in Accounting, Finance and Management, 12*(4), 261–281.

Santomero, A. M., & Vinso, J. D. (1977). Estimating the probability of failure by commercial banks. *Journal of Banking and Finance, 1*, 185–205.

Sarkar, S., & Sriram, R. S. (2001). Bayesian models for early warning of bank failures. *Management Science, 47*(11), 1457–1475.

Sinha, A. P., & May, J. H. (2005). Evaluating and tuning predictive data mining models using receiver operating characteristic curves. *Journal of Management Information Systems, 21*(3), 249–280.

Sinkey, J. (1975). A multivariate statistical analysis of the characteristics of problem banks. *Journal of Finance, 31*(1), 21–38.

Sinkey, J. (1977). Identifying large problem/failed banks: The case of the Franklin National Bank of New York. *Journal of Financial and Quantitative Analysis, 12*, 779–800.

Sinkey, J. (1978). Identifying problem banks: how do the banking authorities measure a bank's risk exposure. *Journal of Money, Credit and Banking, 10*, 184–193.

Sinkey, J. (1979). Problem and failed institutions in the commercial banking industry. Contemporary studies in economic and financial analysis, Vol. 4. JAI Press, Greenwich, Conn.

Stuhr, D.P., & Van Wicklen, R. (1974). Rating and financial condition of banks: a statistical approach to aid bank supervision. Monthly Review, Federal Reserve Bank of New York, September (pp. 233–238).

Sung, T. K., Chang, N., & Lee, G. (1999). Dynamics of modeling in data mining: Interpretive approach to bankruptcy prediction. *Journal of Management Information Systems, 16*(1), 63–85.

Swicegood, P., & Clark, J. A. (2001). Off-site monitoring systems for predicting bank underperformance: A comparison of neural networks, discriminant analysis, and professional human judgment. *International Journal of Intelligent Systems in Accounting, Finance & Management, 10*(3), 169–186.

Tahai, A., Walczak, S., & Rigsby, J. T. (1998). Improving artificial neural network performance through input variable selection. In P. Siegel, K. Omer, A. deKorvin, & A. Zebda (Eds.), *Applications of fuzzy sets and the theory of evidence to accounting II* (pp. 277–292). Stamford, CT: JAI Press.

Tam, K. Y., & Kiang, M. Y. (1992). Managerial applications of neural networks: The case of bank failure predictions. *Management Science, 38*(7), 926–947.

Thomson, J.B. (1991). Predicting bank failures in the 1980s. Economic Review (Q I), (9–20).

Weiss, S. M., & Indurkhya, N. (1998). *Predictive data mining: A practical guide*. San Francisco, CA: Morgan Kaufmann.

Weiss, S.M., & Kapouleas, I. (1989). An empirical comparison of pattern recognition, neural nets, and machine learning classification methods. In Proceedings of the eleventh international joint conference on artificial intelligence (pp. 781–787). San Francisco: Springer.

Weiss, S. M., & Kulikowski, C. A. (1991). *Computer systems that learn: classification and prediction methods from statistics, neural nets, machine learning, and expert systems*. San Mateo, CA: Morgan Kaufmann.

West, R. G. (1985). A factor-analytic approach to bank condition. *Journal of Banking and Finance, 9*, 254–266.

Whalen, G. (1991). A proportional hazards model of bank failure: An examination of its usefulness as an early warning tool. *Economic Review, Federal Reserve Bank of Cleveland, 27*, 21–31.

Whalen, G., & Thomson, J.B. (1988). Using financial data to identify changes in bank condition. Economic Review, (Q II), (pp. 17–26).

Witten, I. H., & Frank, E. (2005). *Data mining: practical machine learning tools and techniques* (2nd ed.). San Francisco, CA: Morgan Kaufman.

Wolpert, D. H. (1994). The relationship between PAC, the statistical physics framework, the Bayesian framework, and the VC framework. In D. H. Wolpert (Ed.), *The mathematics of generalization – Proceedings of the SFI/CNLS workshop on formal approaches to supervised learning* (pp. 117–214). Addison-Wesley.

Zhao, H. (2008). Instance weighting versus threshold adjusting for cost-sensitive classification. Knowledge and information systems. doi:10.1007/s10115-007-0079-1.

Zheng, Z. (2000). Constructing X-of-N attributes for decision tree learning. *Machine Learning, 40*(1), 35–75.