Computational Economics **15:** 107–143, 2000.

Credit Risk Assessment Using Statistical and Machine Learning: Basic Methodology and Risk Modeling Applications

J. GALINDO
Department of Economics, Harvard University, Cambridge, MA 02138, U.S.A.

P. TAMAYO
Thinking Machines Corp., 16 New England Executive Park, Burlington, MA 01803, U.S.A.

Abstract. Risk assessment of financial intermediaries is an area of renewed interest due to the financial crises of the 1980's and 90's. An accurate estimation of risk, and its use in corporate or global financial risk models, could be translated into a more efficient use of resources. One important ingredient to accomplish this goal is to find accurate predictors of individual risk in the credit portfolios of institutions. In this context we make a comparative analysis of different statistical and machine learning modeling methods of classification on a mortgage loan data set with the motivation to understand their limitations and potential. We introduced a specific modeling methodology based on the study of error curves. Using state-of-the-art modeling techniques we built more than 9,000 models as part of the study. The results show that CART decision-tree models provide the best estimation for default with an average 8.31% error rate for a training sample of 2,000 records. As a result of the error curve analysis for this model we conclude that if more data were available, approximately 22,000 records, a potential 7.32% error rate could be achieved. Neural Networks provided the second best results with an average error of 11.00%. The *K*-Nearest Neighbor algorithm had an average error rate of 14.95%. These results outperformed the standard Probit algorithm which attained an average error rate of 15.13%. Finally we discuss the possibilities to use this type of accurate predictive model as ingredients of institutional and global risk models.

1. Introduction

In this section we describe the basic motivation for this work and briefly review the traditional approaches to risk assessment and modeling.

1.1. MOTIVATION

Risk assessment of financial intermediaries is an area of renewed interest for academics, regulatory authorities, and financial intermediaries themselves. This interest is justified by the recent financial crises in the 1980's and 90's. There are many examples: the U.S. S&L's crisis with an estimated cost in the hundreds of billions of dollars; the intervention from 1989 to 1992 where Nordic countries injected around $16 billion to their financial system to keep them from bankruptcy; Japan's bad loans were estimated to be in the range of $160 to $240 billion in October of 1993;[1] in recent years the Mexican government spent at least $30 billion to prevent the financial system from collapsing. Besides these highly publicized cases there are many others of smaller magnitude where a more accurate estimation of risk could be translated into a more efficient use of resources.

Important ingredients in making accurate and realistic risk models are accurate predictors of individual risk and a systematic methodology to generate them. These will be the main subject of this paper. Obviously such risk models are also of interest to the financial intermediaries themselves. In this context we compare different methods of classification of a mortgage loan data set from a large commercial bank. We do this to understand both the limitations and potentials of different methods and, in particular, those based on machine learning techniques (Michie et al., 1994; Mitchell, 1997). We accomplish this systematic comparison using traditional statistical classification techniques. A multi-strategy approach is used, where the results of several algorithms applied to the same data are compared to find the best model. This is justified by the difficulty of selecting an optimal model a priori without knowing the actual complexity of a particular problem or data set. We also introduce a specific methodology for model analysis based on the study of error curves to estimate the noise/bias and complexity of the model and data set. This methodology provides insight of the problem and allows us to address such questions as: how noisy is the data set? how complex is the classification problem? how many data are needed for optimal prediction results? and, what is the best technique for this problem? Some studies comparing different approaches to the classification problem have been properly criticized for using only one technique, for being nonsystematic or for consisting of anecdotal results. To overcome this problem, we systematically analyze a variety of methods including: statistical regression (probit), decision-trees (CART), neural networks, and k-nearest-neighbors on the same data set. This produces other advantages: the pre-processing of the data is more homogeneous and the results are more directly compared. Using state-of-the-art modeling techniques we build more than 9,000 models for one data set as part of the study.

Originally, many of these algorithms and methods were used by statisticians, computer or physical scientists. But their use has now spread successfully to many business applications (Adriaans and Zatinge, 1996; Bigus, 1996; Bourgoin, 1994; Bourgoin and Smith, 1995).[2] Within economics, such studies have mostly been concerned with neural networks. For example, Hutchinson, Lo, and Poggio (1994) finds that neural networks recover the Black-Scholes formula from a two-year simulated data set of daily option prices. Kuan and White (1994) compare the approximation capabilities of a single hidden layer neural network with that of a linear regression model in three deterministic chaos examples. The neural network amply outperforms the linear specification. More work is needed to validate the variety of new algorithms and methods available to the modeler.

One serious problem faced in global financial modeling is scale: to make a good global model, one may need hundreds of individual models, say, one for each financial intermediary. So the model building process must be systematized and virtually automated to avoid the impossibility of building them one by one. This is both a computational and conceptual challenge. One would like a methodology for large scale modeling based on general induction principles allowing individual models to be selected close to optimal. Today's computational resources allow us to tackle these problems. A general framework for large-scale economic modeling using machine learning methods will be of great utility. One can envision global models, incorporating thousands of individual predictive models for risk, that provide invaluable information for regulatory authorities and macro-economists.

Another issue important for financial decision making is the *transparency* or degree of *interpretability* of models. Transparent models are those that are conceptually understood by the decision maker, such as a decision tree expressed in term of profiles or rule sets. By contrast, while neural networks can act as accurate black boxes, they are opaque and not able to provide simple clues about the basis for their classifications or predictions.[3]

1.2. REVIEW OF TRADITIONAL APPROACHES

From the perspective of a regulatory authority, there are at least two ways to measure the risk exposure of a financial institution. The first is traditionally called *early warning system*; the second is *risk decomposition and aggregation* of net risk exposure.

Altman (1981) offers a survey of early-warning-systems studies conducted in the 1970's and early 1980's.[4] Early warning systems rely on a failure-non-failure or problem-non-problem definition for the financial institution. For example, the legal declaration of insolvency (Meyer and Pifer, 1970) or the *problem bank* definition from the Federal Depository Insurance Commission (FDIC) (Sinkey, 1978). Here the financial institutions are grouped into two or more categories and then a statistical discrimination is performed using accounting data information. The goal is then to predict failure or problemful conditions given the explanatory variables. This analysis is phenomenological since it attempts only to describe the failure of the whole institution without assessing the factors that produce the failure.[5]

Risk-decomposition-and-aggregation has its roots in the arrival of capital-asset pricing models and the development of Contingent Claim Analysis. The Capital Asset Pricing Model (CAPM) (Sharpe, 1964; Lintner, 1965; Mossin, 1966) is developed in a one-period set up but this limitation is overcome by the Merton (1973) Intertemporal Capital Asset Pricing Model (ICAPM) and by the Breeden (1979) Consumption Capital Asset Pricing Model (CCAPM). The ICAPM shows that, in equilibrium, the return of financial securities is proportional to the risk premium in the market as well as other sources of risk. The CCAPM shows the relation between the return of the securities and aggregate consumption for state independent utility functions. The arbitrage Pricing Theory (APT) of Ross (1976) relaxes CAPM's necessity to observe the market portfolio. All these capital-asset pricing models state some dependency of asset prices to risk factors. A drawback of multi-factor models is that they provide little clue as to what risk factors beyond market risk should be considered. Black and Scholes (1973) and the Theory of Rational Option Pricing by Merton (1973) show that, under certain conditions, the price of derivatives can be expressed as a non-linear combination of different factors and that is possible to construct portfolios that replicate the payoff structure of the derivatives. These hedging portfolios can be used to hedge unwanted risk.

Risk-decomposition-and-aggregation is an ambitious approach. In essence it attempts to decompose assets and liabilities classes into exposures to some previously defined risk factors and then to aggregate each exposure accross every risk factor. This decomposition relies on the proper identification of the factors and an accurate estimation of the exposures, and, as such, gives reason to look for more accurate and sophisticated risk estimation methods and algorithms. Risk decomposition makes risk management easier since it provides the magnitude and the source of the risk; however, it requires more information and calculation than an early warning system. The accuracy attained by each of the methodologies is a matter of empirical study.

The methodology developed here can provide input to a credit portfolio model, or with modification, can be used to calculate exposure to different factors such as interest, exchange rates, equity and commodity prices, and price volatilities.[6]. These in turn may help to compute the *'value at risk'* of different portfolios. Other applications for descriptive and predictive models of risk are as diverse as estimating the amount of capital provisions, designing corporate policy, or performing credit scoring for commercial, personal, or credit cards portfolios.

2. Strategy and Methodology

In this section we briefly review some of the algorithms, inductive principles, and empirical problems associated with model construction. We also introduce a particular methodology for model building, selection and evaluation that we will follow in the rest of the paper.

2.1. A MULTI-STRATEGY STATISTICAL INFERENCE APPROACH TO MODELING

The general problem one encounters is that of finding effective methodologies and algorithms to produce mathematical or statistical descriptions (models) to represent the patterns, regularities or trends in financial or business data. This is not a new subject and basically extends the methods used for decades by statisticians. For complex real-world data, where noise, non-linearity and idiosyncrasies are the rule, a good strategy is to take an interdisciplinary approach that combines statistics and machine learning algorithms. This interdisciplinary, data-driven, computational approach, sometimes referred as *Knowledge Discovery in Databases* (Fayyad et al., 1996; Simoudis et al., 1996; Bigus, 1996; Adriaans and Zantinge, 1996), is specially relevant today due to the convergence of three factors: (I) *Corporate and government financial databases*, where every financial transaction can be stored and be made available. The wide use of data warehouses and specialized databases has opened the possibility for financial modeling at an unprecedented scale (Landy, 1996; Small and Edelstein, 1996). (II) *Mature statistical and machine learning technologies.* There is a plethora of mature and proven algorithms. Recent results on statistics, generalization theory, machine learning, and complexity have provided new guidelines and deep insights into the general characteristics and nature of the model building/learning/fitting process (Michie et al., 1994; Vapnik, 1995; Mitchell, 1997). (III) *Affordable computing resources* including high performance multi-processor servers, powerful desktops, and large storage and networking capabilities. The standardization of operating systems and environments (Unix, Windows NT/95 and Java) has facilitated the integration and interconnection of data sources, repositories and applications.

There are many algorithms available for model construction, so one of the main problems in practice is that of algorithm selection or combination. Unfortunately it is hard to choose an algorithm a priori because one might not know the nature and characteristics of the data set, e.g. its intrinsic noise, complexity, or the type of relationships it contains. Algorithms vary enormously in their basic structure, parameters and optimization landscapes but they can roughly be classified in a few groups (Michie et al., 1994; Weiss and Kulikowski, 1991; Mitchell, 1997).

- Traditional statistics: linear, quadratic and logistic discriminants, regression analysis, MANOVA, etc. (Hand, 1981; Lachenbruch and Mickey, 1975; Eaton, 1983).
- Modern statistics: *k*-Nearest-Neighbors, projection pursuit, ACE, SMART, MARS, etc. (Michie et al., 1994; McLachan, 1992; Weiss and Kulikowski, 1991).
- Decision trees and rule-based induction methods: CART, C5.0, decision trees, expert systems (Michie et al., 1994; Mitchell, 1997).
- Neural networks and related machines: feedforward ANN, self-organized maps, radial base functions, support vector machines, etc. (Michie et al., 1994; Mitchell, 1997; Hassoun, 1995; White, 1992).
- Bayesian Inference and Networks (Fayyad, 1996; Berger, 1985; Carlin and Louis, 1996).
- Model combination methods: boosting and bagging (Freund and Shapire, 1995; Breiman, 1996).
- Genetic algorithms and intelligent-agents (Goldberg, 1989).
- Fuzzy logic, fractal sampling and hybrid approaches (Zadeh, 1994).

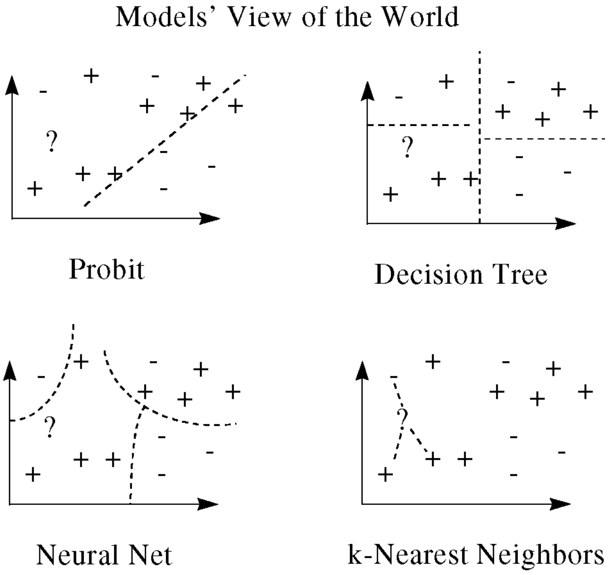


Figure 1. Different models' view of the world. Each algorithm builds a model that represents correlations or regularities according to a particular structure or representation. A new record '?' will be classified according to the prescription of each model's structure (e.g. the particular decision domains and boundaries).

Each algorithm employs a different method to fit the data and approximate the regularities or correlations according to a particular structure or representation. In this study we choose four different algorithms that represent four important classes of predictors: CART decision-trees, feedforward neural networks, *k*-nearest-neighbors, and linear regression (probit). A cartoon representation of each of these is shown in Figure 1.

The recent introduction of model combination methods promises to provide more accurate predictions and reduce the burden of model selection by combining existing algorithms using appropriate re-sampling and combination methods (Freund and Shapire, 1995; Breiman, 1996). The algorithms mentioned in the previous section have been introduced in the context of different disciplines where the problem of data fitting or model building is approached from a particular perspective. These approaches can be roughly be classified as follows:

- Traditional and Modern Statistics and Data Analysis (Eaton, 1983; Fisher, 1950; Hand, 1981; Lachenbruch and Mickey, 1975).
- Bayesian Inference and the Maximum Entropy Principle (Jeffreys, 1931; Jaynes, 1983; Berger, 1985; Carlin and Louis, 1996;.
- Pattern Recognition and Artificial Intelligence (McLachan, 1992; Fukunaga, 1990; Weiss and Kulikowski, 1991).

- Connectionist and Neural Network Models (McClelland and Rumelhart, 1986; Hassoun, 1995; White, 1992).
- Computational Learning Theory and Probably Approximately Correct (PAC) Model (Valiant, 1983; Keans and Vazirani, 1994; Mitchell, 1997).
- Statistical Learning Theory (Vapnik, 1995).
- Information Theory (Cover and Thomas, 1991; Li and Vitanyi, 1997).
- Algorithmic and Kolmogorov Complexity (Rissanen, 1989; Li and Vitanyi, 1997).
- Statistical Mechanics (Seung et al., 1993; Opper and Haussler, 1995).

We do not review them here, but make the reader aware of their existence. Historically many were developed independently but recent studies help to understand their relationships and equivalences in some cases (Li and Vitanyi, 1997; Rissanen, 1989; Vapnik, 1995; Keuzenkamp and McAleer, 1995). The process of choosing and fitting (or training) a model is usually done according to formal or empirical versions of *inductive principles.* These principles have been developed in different contexts, but all share the same conceptual goal of finding the ‘best’, the ‘optimal’, or the most parsimonious model or description that captures the structural or functional relationship in the data (potentially subject to additional constraints such as those imposed by the model structure itself).

Perhaps the oldest, and certainly most accommodating, induction principle is the one advocated by Epicurus (Amis, 1984) that basically states: *keep all models or theories consistent with data*. At the other end of the spectrum, skeptical philosophers have questioned the validity of induction as a valid logical method (see for example Hume (1739) or Popper (1958)). In practice, induction principles are useful beacuse they stand at the core of most data fitting and model building methods. Traditional model fitting and parameter estimation in statistics have usually employed Fisher’s Maximum Likelihood principle (Hand, 1981; Lachenbruch and Mickey, 1975). Another approach to induction is provided by Bayesian inference (Jeffreys, 1931; Jaynes, 1983; Berger, 1985; Carlin and Louis, 1996), where the model is chosen by maximizing the posterior probabilities. Another important principle is based on the minimization of empirical risk (Vapnik, 1995). The structural minimization principle takes into account the model size or ‘capacity’, and thereby its ability to generalize and its finite sample behavior (Vapnik, 1995). Another class of principles – the modern versions of the celebrated Occam’s razor (*choose the most parsimonious model that fits the data*) –, are represented by the Minimum Description Length (MDL) principle, (Rissanen, 1989), or Kolmogorov complexity (Li and Vitanyi, 1997), which chooses the model having the shortest or most succinct computational representation or description. These inductive principles have more in common that first appears. Particular instances of them are familiar in the form of function approximation and parameter or density estimation, neural net training methods, and data compression algorithms. A general protocol for learning from a computational perspective, the Probably Approximately Correct

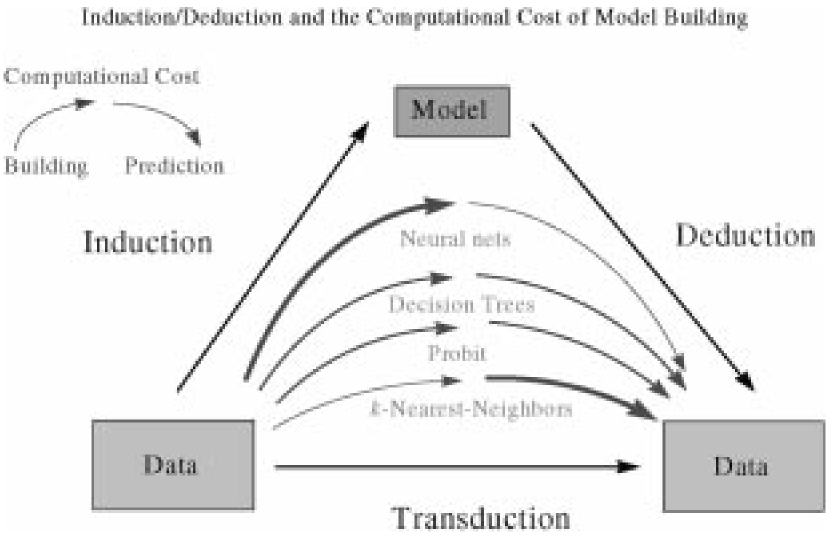


Figure 2. Inductive models: its relationship with data and their computational cost. Models are build with training data and become short representations of the logical or statistical relationships in it. Once a model has been built it can be applied to classify or predict new data in a deductive way. Transduction, as defined by Vapnik (1995), is the process of extrapolation directly from data to data with little or no model construction (for example *k*-NN).

(PAC) model (Kearns and Vazirani, 1994), has been introduced by Valiant (1983) as an attempt to reduce the ambiguity of earlier formulations.

The process of *building* a model and *applying* it to new data implies computational costs. These may limit the type of model that can be used in a particular situation. Figure 2 shows the relationships among data, model, and the deductive, inductive and transductive processes.

In economics and finance, classification or predictive models derived from data are not used in isolation but as part of a larger model or in conjunction with interpretative theories, often in the context of policy setting. Therefore it is desirable that they be (i) *accurate*, in the sense of having low generalization error rates; (ii) *parsimonious*, in the sense of representing and generalizing the relationships in a succinct way; (iii) *non-trivial*, in the sense of producing interesting results; (iv) *feasible*, in terms of time and resources and (v) *transparent*[7] *and interpretable*,[8] in the sense of providing high level representations of and insight into the data relationships, regularities, or trends.

In practice the process of model building is always hampered by the availability and quality of data. The collection process is never perfect or completely accurate, and the data often contain inconsistencies or missing values. The data relationships can be complex, non-normal, non-linear, and reflect structural changes such as demographic or market seasonal trends. To some extent one could argue that each data set is idiosyncratic and unique in space and time. Finally, the dynamic aspects of financial data make model building a continuous process.

Conceptually, statistical and machine learning models are not all that different (Michie et al., 1994; Weiss and Kulikowski, 1991). Many of the new computational and machine learning methods generalize the idea of parameter estimation in statistics. Machine learning algorithms tend to be more computational-based and data-driven, and, by relying less on assumptions about the data (normality, linearity, etc.), are more robust and distribution-free. These algorithms not only *fit* the *parameters* of a particular model but often change the *structure* of the model itself, and in many instances they are better at generalizing complex non-linear data relationships. On the other hand machine learning algorithms provide models that can be relatively large, idiosyncratic, and difficult to interpret. The moral is that no single method or algorithm is perfect or guaranteed to work, so one should be aware of the limitations and strengths of each. For an interesting discussion about statistical themes and lessons for machine learning methods we refer the reader to Glymor et al. (1997). The new algorithms also have more explicit means for accounting for the actual complexity, size, or capacity of a model than do the traditional approaches.

2.2. MODEL BUILDING AND ANALYSIS OF ERRORS AND LEARNING CURVES

In this section we describe the basic elements of the model building methodology and the analysis that we employ in the four algorithms considered here. The main elements of the analysis are

- Basic model parameter exploration.
- Analysis of importance/sensitivity of variables.
- Train, test/generalization and evaluation error analysis including performance matrices.
- Analysis of learning curves and estimates of noise and complexity parameters.
- Model selection and combination of results.

Basic model parameter exploration. This is done at the very beginning by building a few preliminary models to get a sense for the appropriate range of parameter values.

Analysis of importance/sensitivity. The relative importance, in terms of the relative contribution of each variable to the model, is important because it provides the basis for *variable selection* or *filtering*. One starts with as many variables as possible and then eliminates those that are not relevant to the model. One must be careful in this filtering process because there are often complications such as the ‘masking’[9] of variables. In the data set considered in this paper, the number of variables is small enough that we need not worry about variable selection; however, we analyzed the importance/sensitivity of the variables in the final model.

Table I. Format of the performance matrix for a binary classification problem.

	Actual vs. predicted (performance matrix)			Total error
	Predicted (by model)			
	0	1	Total	
Actual 0	$x1$	y	$x1 + y$	Error for $0 = y/(x1 + y)$
Actual 1	z	$x2$	$z + x2$	Error for $1 = z/(z + x2)$
Total	$x1 + z$	$y + x2$	$x1 + x2 + y + z$	Global error $(z + y)/(x1 + x2 + y + z)$

Train, test/generalization and evaluation error analysis. In classification problems the most direct measure of performance is misclassification error: the number of incorrectly classified records divided by the total. For a given binary classification problem, this number varies between the default prediction error (from assigning all records to the majority class) and zero (for a perfect model). It is important that this error be measured both for the sample used to build the model and a ‘test’ sample data set containing records not used in the model construction. This allows selecting the model having the best generalization instead of best fit to the training data. The *performance* or *confusion matrix* provides a convenient way to compare the actual versus predicted frequencies for the test data set. The format we use for these matrices is shown in Table I.

This matrix allows us to distinguish asymmetries in the predictions (e.g. false/ positives vs false/negatives). Once a reasonable model for a given class has been selected, a final estimate of error is done with an independent sample (the evaluation data set). Thus, for this part of the analysis we divide our 4,000 data records into subsets of size 2,000 for training, 1,000 for testing and 1,000 for evaluation. This relatively small amount of data was all that was available for the study. We shall see that a data set of roughly 22,000 records is needed to obtain optimal results for this problem.

Analysis of learning curves and complexity. This exploratory approach computes average values of train and test (generalization) errors for given values of training sample and model size. By fitting simple algebraic scaling models to these curves, one can model the behavior of the learning process and obtain rough estimates of the complexity of and noise in the data set. The results help to understand the intrinsic complexity of the problem, the quality of data, and provide insight into the relationship between error rates, model capacity, and optimal training set sizes. This information is also useful in planing larger modeling efforts relevant to production rather than exploratory data sets and this analysis allows us to view the problem from the perspective of structural risk minimization (Vapnik, 1995) and bias/variance decomposition (Breiman, 1996; Friedman, 1997).

Figure 3 describes the basic phenomenology of learning curves. For fixed model size, as the training data set increases, the train and test errors converge to an

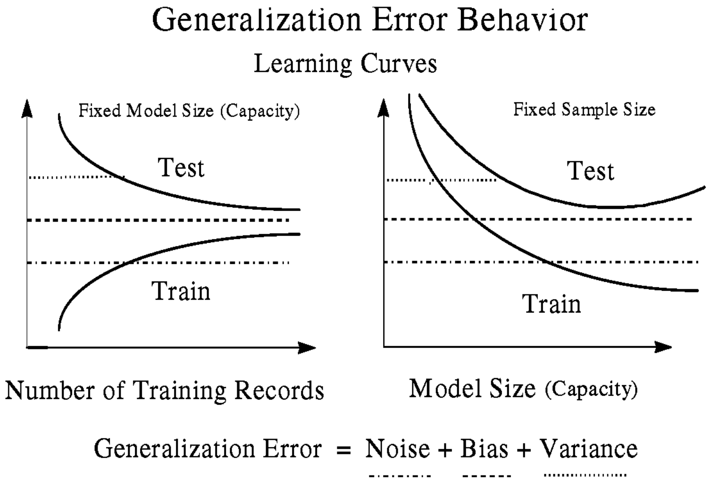


Figure 3. Generalization error behavior and error curves.

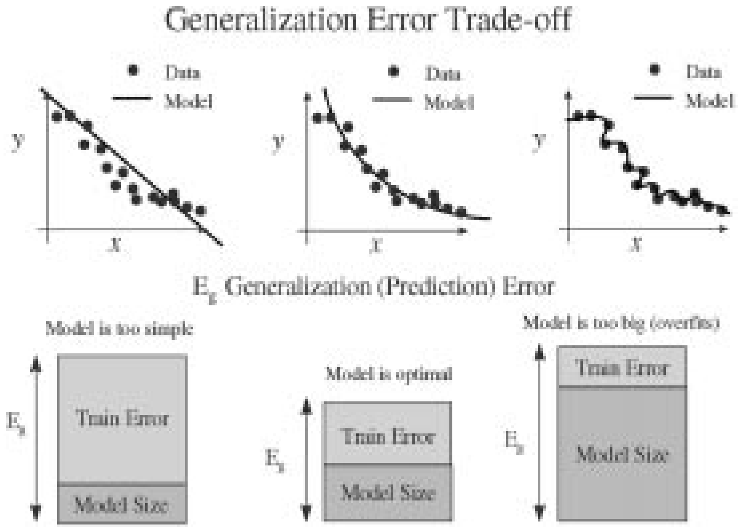


Figure 4. Generalization error trade-off in terms of model size.

asymptotic value determined by the bias of the model and the intrinsic noise in the data. The test error decreases because the model finds more support (data instances) to characterize regularities and therefore generalizes better. The train error increases because with more data, the model with its pre-determined fixed size finds greater difficulty fitting and memorizing. For very small sample sizes the train error could be zero – the model performs a ‘lossless’ compression of the training data. For a given training sample size, there is an optimal model size for which the model neither under- nor overfits the data.

Another way to view these trade-offs is shown in Figure 4. If the model is too small, it will not fit the data very well and its generalization power will be limited

Table II. Format for the results of learning curve analysis fitting the model: $E_{\text{test}} = \alpha + \beta/m$.

Model	Test error at maximum training sample (standard dev. in parenthesis)	Noise/bias α	Complexity β	Optimum training sample size (recs)

by missing important trends. If the model is too large, it will overfit the data and lose generalization power by incorporating too many accidents in the training data not shared by other data sets. This behavior is also shown in the second graph of Figure 3. As the model size is increased the generalization error decreases because a larger model has less bias and fits the data better. However, at some point the model becomes too large, producing overfitting which results in the curve moving upwards. This fundamental behavior is shared generally by finite-sample inductive models (Kearns and Vazirani, 1994; Vapnik, 1995) and agrees well with the empirical behavior we observe in all of our models.

The methodology we use for the analysis of mortgage-loan learning curves is as follows: for each data set size, and fixed model size, we build 30 models with different random samples from the original data set. We then average the on- (train) and off-sample (test) error rates. These averaged errors are then fitted to an inverse power law: $E_{\text{test}} = \alpha + \beta/m^{\delta}$, where α estimates the noise/bias, β and δ estimate the complexity, and m is the sample size. Based on our experience and other work in the literature (e.g. Cortes, 1994a,b), this model works well in describing the empirical learning curve behavior for fixed model size. A typical empirical learning curve as a function of the sample size is shown in Figure 9. Other empirical learning curves for the mortgage-loan models can be seen in Figure 5 and 13. This analysis is not entirely phenomenological because the functional forms are motivated by theoretical models (Vapnik, 1995; Amari, 1993; Seung, 1993; Opper and Haussler, 1995). The inverse power law functional form of our approach is similar to the one used by Cortes and co-workers (Cortes, 1994a,b), but we fit directly to averaged test error curves alone rather than combining them with training curves. The computation of exact functional forms is a very difficult combinatorial problem for most non-trivial models, but functional dependencies (e.g. inverse power laws) and worse-case upper bounds have been calculated (Kearns and Vazinari, 1994; Vapnik, 1995). These theoretical models suggest that the value for the exponent δ will be no worse than 1/2 (Vaknik, 1995). Other formulations, in the context of computational learning theory and statistical mechanics using average rather than worst case, suggest a value of $\delta \approx 1$ (Opper and Haussler, 1995; Amari, 1993). There is also empirical support for this value from earlier work (Cortes, 1994a,b). For our mortgage-loan data set, we find that $\delta = 1$ provides a reasonable fit for the error curves, and therefore we assume $\delta = 1$ when fitting the data.[10] Table II shows the basic format we use to report the curve analysis results.

The learning curve analysis methodology describe here is still under investigation, so we recommend it with caution. It has been used by the authors to study several data sets with good results. Similar methodologies have been reported in the literature (Cortes, 1994a,b) but their widespread use has been limited by high computational cost. The scaling analysis can be improved in many ways, particularly by extending it to account for model size to describe the entire learning manifold. This is a subject for future work.

3. Application of the Analysis to a Financial Institution

Here we apply our methodology to the prediction of default in home mortgage loans. The data were provided to us by Mexico's security exchange and banking commission: Comision Nacional Bancaria y de Valores (CNBV). The Universe of mortgage loans in Mexico is approximately 900,000. From this universe, a sample of 4,000 mortgage loans records from a single financial institution was given to us. The average mortgage loan amount is 266,827 Mexican pesos (around $33,300 U.S.) as of June 1996. This institution's mortgage loan portfolio represents 14.3% of the market. The reader not interested in the details can go directly to Section 3.6 which contains a summary of results.

3.1. DATA ANALYSIS, PREPARATION, AND PRE-PROCESSING

The data were already being used for a regression model by the CNBV, and therefore they required little pre-processing or manipulation prior to model building. The data set of 4,000 records, each corresponding to a customer account, contains a total of 24 attributes. CNBV collected information in this format for several institutions as part of a project to analyze the probability of default. Following CNBV, we define the binary target variable *Default* so that the account is considered as 'defaulted' only if no payments are made during the last two months. *Credit_ Amount* is the value of the credit, *Unpaid_ Bal* is the unpaid balance, *Overdue_ Bal* is the overdue balance, and *Debt* is the total debt, which is equal to the sum of unpaid and overdue balances. There are three variables related to the guarantee of the loan: *Guarantee* is the value of the guarantee, *Dguaratee1* and *Dguarantee2* are unity if the guarantee covers at least 100% or 200% of the total debt, respectively. Two of the variables, *Soc_ Interest* and *Residential*, give information about the type of credit.[11] *Residential* indicates if the credit is a regular loan. Four attributes are related to the use of the loan and had no variation over the data set: (i) *Adquisition*, takes the value of 1 if the loan is for acquiring an already existing house, and 0 otherwise; (ii) *Construction*, takes the value of 1 if the loan is for construction of a new house, and 0 otherwise; (iii) *Liquidity*, takes the value of 1 if the loan is to provide liquidity for operations like house remodeling, and 0 otherwise; (iv) *Adq_ or_ Const*, takes the value of 1 if the loan is for buying or constructing the house, and 0 otherwise. Ten variables, *Month1–Month10* contain information on the payment history from June 1995 to March 1996. For each month, a 1 indicates no payment for that period, and 0 otherwise. In addition to the 24 variables, a new variable *Default_ Index* is created to condense information about the payment history and probability of payment into a single attribute. A similarly combined variable proved useful in the regression model built by CNBV and so we include it in our analysis too. A matrix with 0's and 1's is constructed from the payment information for the first 10 months of each account. State 0 indicates a payment is made, 1 otherwise. Then P_{ij} is defined as the one-step probability that the account in any given period changes from being in state i to state j, namely,

$$P_{ij} = P(state_t = j | state_{t-1} = i).$$

With the available information, the following one-step transition matrix P^1_{ij} is calculated based on the frequency of each transition,

$$P^1 = \begin{bmatrix} P^1_{00} & P^1_{01} \\ P^1_{10} & P^1_{11} \end{bmatrix}.$$

This matrix is raised to power n (from 2 to 10) so that for every string of payment experiences the following variable is created,

$$\text{Default\_Index} = \frac{\sum_{k=1}^{10} P_{i^k 1}}{10},$$

where $P_{i^k 1}$ takes the value of P^{11-k}_{i1} if the account is in state i in the kth month.

3.2. PROBIT RESULTS

Traditionally, binary classification problems use linear, logit, or probit models. The linear model has several limitations.[12] The logit and probit models are similar, but they use the cumulative logistic and normal distributions, respectively. One difference in these distributions is that the logistic distribution has fatter tails, and this produces small differences in the model. However, there are no theoretical grounds to favor one technique over the other.[13] We develop the following probit model similar to one used at CNBV. We use it as a benchmark for comparing the other three algorithms.

$$P\{Default = 1\} = \Phi(\beta x_i),$$

where the index βx_i is defined as,

$$\begin{aligned} \beta x_i &= \beta_0 + \beta_1 Dguarantee1_i + \beta_2 Default\_index_i + \beta_3 Soc\_interest_i \\ &\quad + \beta_4 Construction_i + \beta_5 Dguarantee1_i Default\_index_i \end{aligned}$$

and $\Phi(x)$ is the cumulative normal distribution. Alternative specifications are also used: stepwise probit with all the variables, including and excluding the interaction variable (the last term in the model above). In all cases the predictive power of the

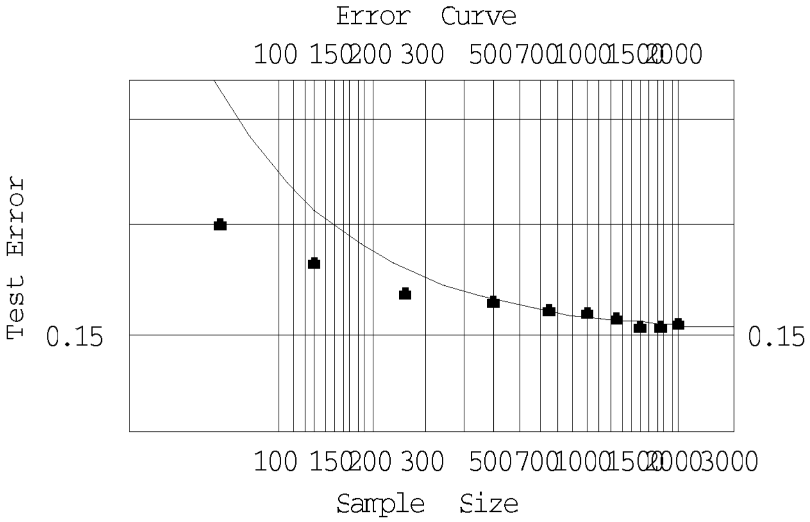


Figure 5. Generalization error curve as a function of sample size for Probit.

models retains the same error range as the one from CNBV. Thus we decided to use the same specification as CNBV. To assign each predicted probability to the default or non-default group we choose a threshold value. While different values for this parameter were examined, we use a value of 0.7 in the final model because it gave the lowest error rate.

For each modeling technique, the generalization learning curve is computed. Each point is the average error from 30 bootstrap samples for a given training set size. Thus, the 10 points appearing in every curve result from fitting 300 models. The first three points, corresponding to sample sizes of 64, 128, and 256 records, are not used for fitting the curve because the inverse power functional form does not fit well for small sizes. For consistency the same criterion is applied to all the techniques.

As can be seen from Figure 5, the average error rate for the probit models begins about 16%, gradually declines to 15%, and for large sample sizes it almost converges to its asymptotic value. This is confirmed by the results of fitting the inverse power law model given in Table III. The estimated noise/bias parameter α – the constant in the model – gives the estimated minimum asymptotic value of the error rate. This means that the asymptotic intrinsic noise in the data plus the model bias is about 15%, and this value could be achieved (within 0.1%) with 1,804 or more records. The number of records is calculated from the functional form of the learning curve fit by solving for the sample size (m) and allowing for an error equal to the convergence value plus the arbitrary value 0.001 (we assume convergence at 0.1% of the asymptotic value). In these circumstances the probit model reaches its predictive capacity and using additional training records has little effect on the generalization error and therefore the accuracy of the model.

Table III. Learning curve results for probit. The asymptotic value for the error rate is 15.02%. The standard deviation of the test error rate is in parenthesis.

Model	Test error at $m = 2,000$	Noise/bias α	Complexity β	Optimum training sample size (recs)
Probit	15.13% (0.0047)	0.15025	1.80	1,804

Table IV. Actual vs. predicted results for Probit.

	Actual vs. predicted (performance matrix)				
	Predicted				
	0	1	Total	Total error	15.80%
Actual 0	462	30	492	Error for 0	6.10%
Actual 1	128	380	508	Error for 1	25.20%
Total	590	410	1,000		

We interpret the relatively small value of the complexity parameter $\beta = 1.8$ to mean that the probit model has limited capacity to 'see' all the complexity in the data. This explains why it requires few records relatively to the other algorithms to attain the asymptotic value. The performance matrix shown in Table IV gives us more information about the source of the predictive power of the probit model. There is an asymmetry between the error rate for 0's (non-default group) and 1's (the default group). The model identifies the non-defaulting group better than the defaulting group, and as a consequence the error rate for 0's is only 6.10% while for 1's it is 25.20%.

3.3. DECISION-TREE CART MODEL

CART (*Classification And Regression Trees*) (Breiman et al., 1984) are powerful non-parametric models that produce accurate predictions and easily-interpretable rules that characterize them. They are good representatives of the decision-tree, rule-based class of algorithms. Other members of this family are C5.0, CHAID, NewID, Cal5 etc. (Michie et al., 1994). A nice feature of this type of model is *transparency*; they can be represented as a set of rules in almost plain English. This makes them ideal for economic and financial applications.

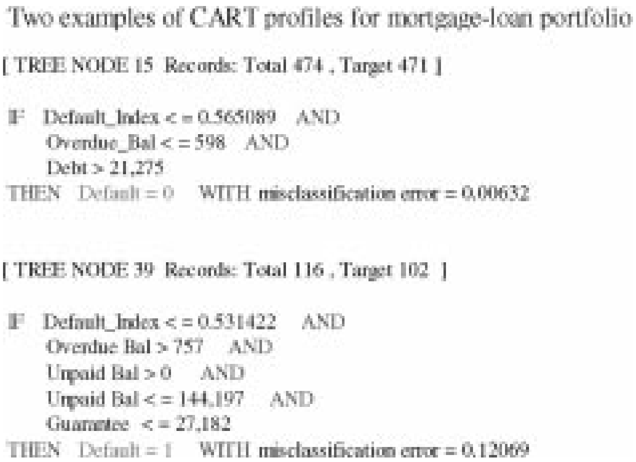
Tables V and VI show the results of a preliminary study of the effect of changing different parameters. We control the size of CART trees by changing the '*density*' parameter (a feature supported by the toolset). This parameter represents the minimum percentage of records in any class that is required to continue splitting at any

Table V. Accuracy vs. time trade-off for CART models.

Density	Tree size (best tree)	Tree size (largest tree)	Test error (%)	Time (secs)
0.2	25	25	10.5	13
0.15	25	25	10.5	13
0.1	35	41	9.8	13
0.05	39	45	7.5	14
0.025	81	89	7	17
0.01	77	121	6.9	20
0.005	109	189	6.5	24
0	161	299	6.7	27

tree node. By changing the value of the *density*, we were able to study the trade-off between accuracy, model size, and time to build a tree model. As the *density* is decreased, the model building time increases and the accuracy of the model improves (Table V). Typically one starts with a relatively high value for the *density* in order to build a preliminary rough model and then decreases its value to make the model more and more accurate. A preliminary exploratory CART model is built with density 0.05 to assess the execution time and size of the tree. The impurity function used in the tree growth process is the Gini index. *The best tree* listed in the second column of the table, corresponds to the subtree of the full CART decision-tree with the smallest error in the test data set. This optimal subtree is obtaining by a *pruning* process in which a set of subtrees is generated by eliminating groups of branches. The branch elimination is done by considering the complexity/error trade off of the original CART algorithm (Breiman et al., 1984). For a decision tree, this pruning process exemplifies a practical means for model size or capacity control (Vapnik, 1995).

In Figure 6, two examples of tree profiles are shown. The interpretation of the rules is straightforward: the first rule identifies a non-defaulting group of customers with not too high default index, an overdue balance less than 598, and a debt greater than 21, 275. People in this profile are predicted to pay with a very low misclassification error of 0.6%. Despite agreeing with intuition, the rule is not trivial. A person with a reasonable payment history, a particularly low overdue balance, and still some debt to cover is likely to pay. The second rule identifies a group that defaults. As in the previous rule, the default index is low, but the overdue balance threshold is higher (757), the unpaid balance is positive but could be considerably higher (144,197), and the guarantee value is small (27,182) relative to the overdue balance – and maybe to the unpaid balance. In this case the group is predicted to default with a misclassification error of 12.06%. This rule may describe the profile of someone who recently stopped paying but, more importantly, has a low incentive to pay because of the low value of the guarantee. As one can see, the individual



Two examples of CART profiles for mortgage-loan portfolio

```
[ TREE NODE 15  Records: Total 474 , Target 471 ]

IF   Default_Index < = 0.565089   AND
     Overdue_Bal < = 598   AND
     Debt > 21,275
THEN   Default = 0    WITH misclassification error = 0.00632


[ TREE NODE 39  Records: Total 116 , Target 102  ]

IF   Default_Index < = 0.531422    AND
     Overdue Bal > 757    AND
     Unpaid Bal > 0    AND
     Unpaid Bal < = 144,197    AND
     Guarantee  < = 27,182
THEN   Default = 1    WITH misclassification error = 0.12069
```

Figure 6. Two examples of tree rules or profiles. In the rules shown, the first number is the number of the tree node that defines that rule, then the number of records that fall into the rule (e.g. 474) and the number of records that actually had the predicted target value (e.g. 471). After these numbers the actual body of the rule is shown.

error of each rule or profile could be smaller or greater than the overall average error. These rules are typical of CART models. After a careful interpretation and validation they can be used as elements of procedures, models, or policies.

Now let us turn to the learning curve analysis. In this analysis we use 8 different tree model sizes. Each size is specified by the maximum number of nodes allowed for the CART tree: 20, 40, 80, 100, 120, 200, 300, and 400. For each size, 10 different training set sample sizes are used: 64, 128, 256, 500, 750, 1000, 1250, 1500, 1750, and 2,000 records. For each sample size, 30 bootstrap averages are made. In total 2,400 tree models are used for the analysis.

In Figure 7 we show the generalization learning curves for trees of different sizes. As expected the generalization errors decrease as the training sample is increased, and the asymptotic value for each of the curves decreases with increasing model size. The lines correspond to the fit of the inverse power law model described in Section 2.2 ($E_{\text{test}} = \alpha + \beta/m$). As can be seen, the fitted curve does not fit the small sample sizes, so we left out sample sizes 64, 128, and 256 from all the curve fittings.

In Figure 8, the graph shows the generalization learning curves for tree of maximum size set to 120, 200, and 400 nodes. We can see the over all behavior illustrated by Figure 3 and 4. A summary of learning curve behavior for several tree models is shown in Table VI. Based on these graphs and Table VI, we can conclude that the optimal size (capacity) for the tree model is 120 nodes. Trees with 80 nodes or less are short on capacity and trees with 200 nodes or more have excess capacity. Notice from Figure 7 and Figure 8 how different size tree models attain different asymptotic error values. The minimum noise/bias is achieved by the 120-node tree, which has the highest values for the complexity estimate. The larger

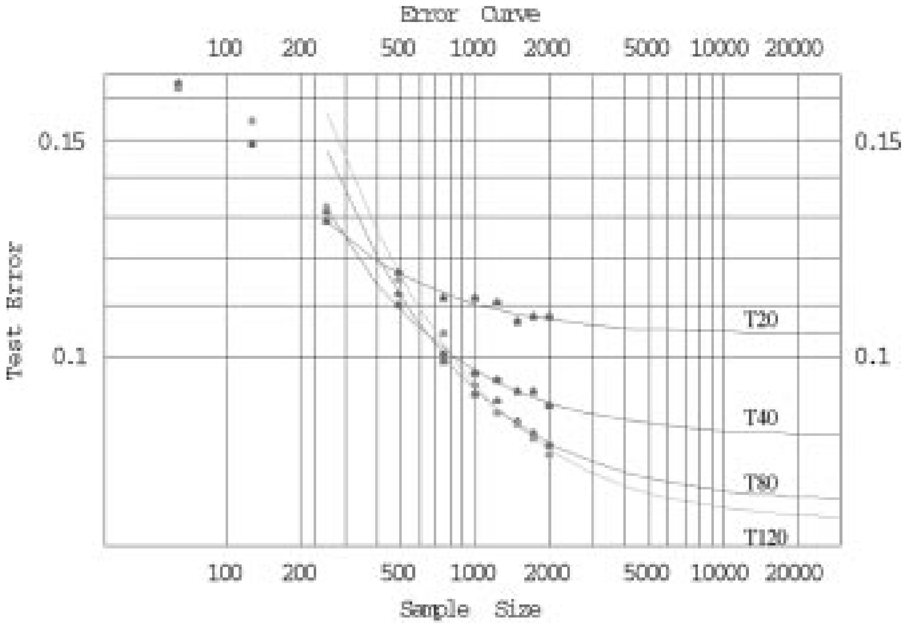


Figure 7. Learning curves for CART trees with 20,40,80, and 120 nodes.

the complexity, the more records are needed to attain convergence to the asymptotic value. The best model (120 nodes) attains an average error rate of 8.31% for 2,000 records. From the inverse power-law fit, we obtain an asymptotic error value of 7.31%.

The performance matrix for the tree with 120 nodes (Table VII) shows that most of the gain in the predictive power of the tree comes from better identifying the defaulting group. It achieves an error rate of 6.29% as compared to an error rate of 11.99% for the non-defaulting group. As described in Section 2.2, the results shown in the performance matrices correspond to the errors calculated on a evaluation data set of 1,000, which remains the same for all the modeling methods.

Finally in Figure 9, we show both generalization and training learning curves for the best CART tree model (120 nodes). We can also see that for small samples (64, 128, 256, and 500) the tree memorizes the training data perfectly with an error training rate close to zero. This is the expected full memorization (lossless compression) effect. As the sample size increases, the training error starts to increase; the model has more and more difficulties ‘memorizing’ the training sample. The final results of this analysis suggest that the intrinsic noise in the data plus the model bias is about 7.31% and that the convergence of train and test errors takes place for an optimal training data set size of about 21,675 records.

We close this section by including the results of the sensitivity/importance analysis of variables. Here we concentrate on the interpretation of sensitivity/importance for our final best model (120 nodes). Figure 10 shows a graph of relative sensitivity/importance for this model.

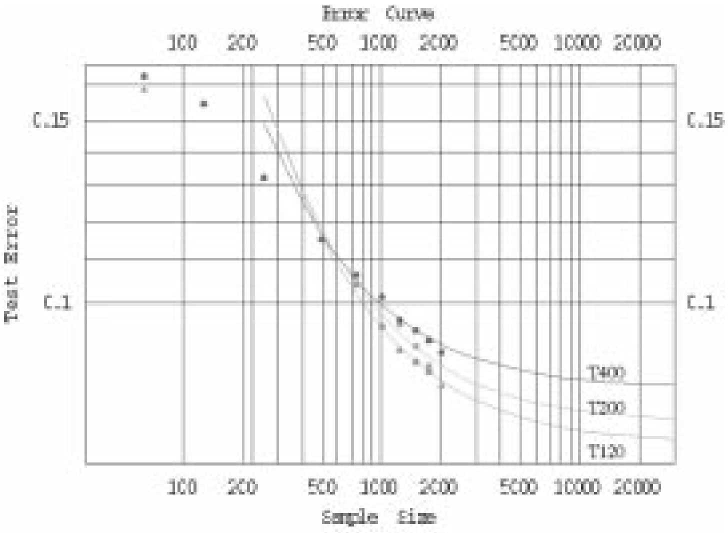


Figure 8. Generalization error curves for trees with 120, 200, and 400 nodes. Here we observe that the generalization error increases for excess model size (capacity).

Table VI. Learning curve results for different model sizes. The results are obtained from fitting the inverse power law model to each of the different capacities (model sizes). The first column shows the number of nodes for the tree, the second column presents the generalization error rate at 2,000 sample size (standard deviation inside the parenthesis). The third and fourth columns show the estimated parameters α and β. Finally the last column shows the number of records needed to obtain an error rate of $\alpha + 0.001$.

Size # of nodes	Test error at $m = 2,000$		Noise/bias α	Complexity β	Optimum training sample size (recs)
20	10.74%	(0.0055)	0.10400	6.36	6,357
40	9.13%	(0.0066)	0.08592	11.65	11,646
80	8.45%	(0.0060)	0.07591	18.13	18,127
100	8.41%	(0.0051)	0.07413	20.19	20,186
120	8.31%	(0.0058)	0.07312	21.68	21,675
200	8.31%	(0.0065)	0.07668	20.69	20,689
300	8.87%	(0.0075)	0.08230	17.16	17,160
400	8.97%	(0.0075)	0.08272	16.88	16,876

Table VII. Performance matrix for the tree with 120 nodes. Notice the asymmetry in the predictions: the model identifies the default group (6.29% error) better than the non-default group (11.99% error)

	Actual vs. predicted (performance matrix)				
	Predicted				
	0	1	Total	Total error	9.10%
Actual 0	433	59	492	Error for 0	11.99%
Actual 1	32	476	508	Error for 1	6.29%
Total	465	535	1,000		

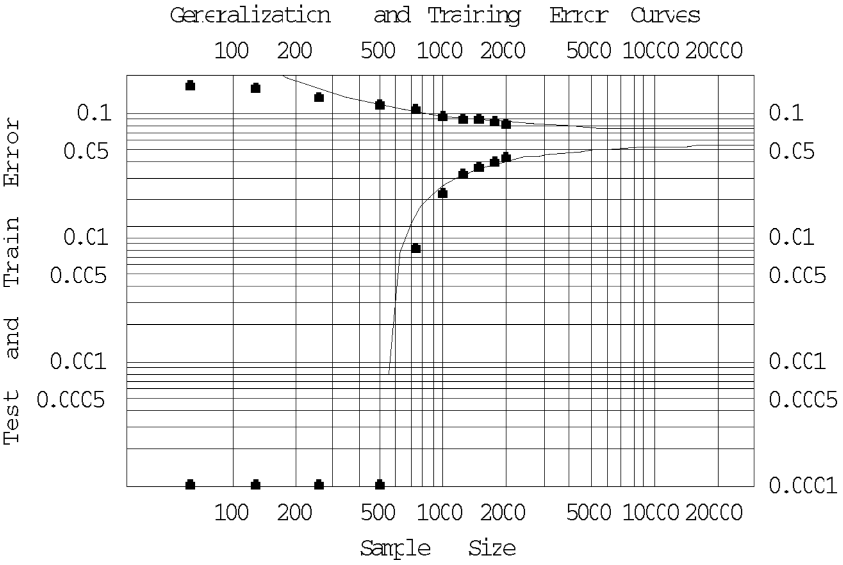


Figure 9. Generalization and training learning curves for the best CART tree model.

The graph shows the results for the variables of the sensitivity/importance analysis[14] for our best CART model. The five variables shown are those that make the greatest contribution to the model predictions. Perhaps not surprisingly these variables appear prominently in the actual CART rules.

3.4. NEURAL NETWORKS

We choose to use the standard feedforward neural network architecture (see for example Hassoun (1995) and White (1992)) supported by the Darwin toolset (see Appendix A), and applied to several training algorithms: *backpropagation, steepest*

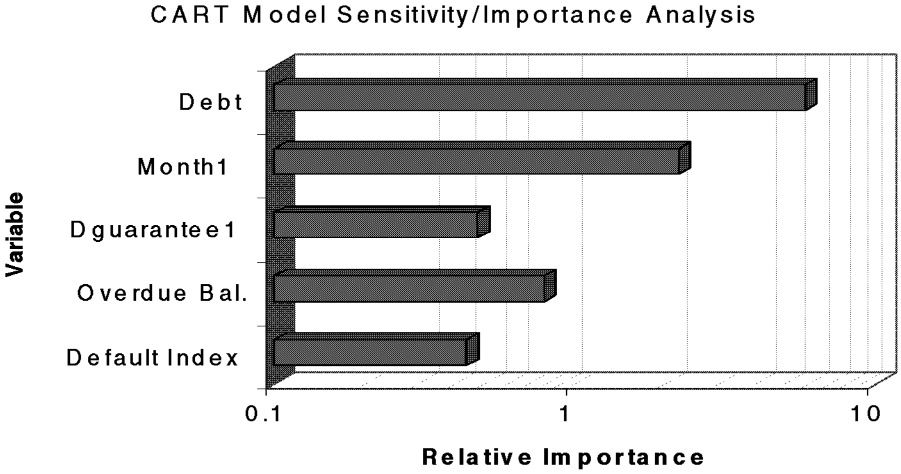


Figure 10. Relative sensitivity/importance for CART.

descent, conjugate gradient, modified Newton, and genetic algorithm. Second-order methods, such as conjugate gradient, allow for faster training than standard back-propagation. We also investigate the effect of changing the activation functions for the hidden layer: sigmoid, linear, and hypertangent. The genetic algorithm allows weight optimization in the region of the error surface, which is possibly difficult for gradient based methods. In addition to manual training we used the *train and test* mode available in the toolset, which is useful for preventing overfitting (it implements a smoothing method for automatic termination of training when the test error starts to increase). The results are summarized in Table VIII.

From the result of this preliminary network analysis, we concentrate on the best performing combination for the rest of the analysis. These are the sigmoid for the activation function in the hidden layer and the conjugate gradient method for the training algorithm. The relatively poor performance of the genetic algorithms is likely due the small number of iterations used. The analysis of learning curves is done similarly to that described earlier for the decision tree models. A total of 4,200 neural network models are used for this part of the analysis. The results are shown in Table IX, Table X, and Table XI. Two approaches are used for the network selection. First we explore different architectures (number of nodes) while holding the number of iterations constant. The number of nodes in the hidden layer is varied from 2 to 16. The best results are obtained for the neural network with 2 hidden nodes, as can be seen in Table IX – where the number of input variables (25), the number of output nodes (1), and the number of iterations (25) remains constant. The error rate increases with the number of nodes in the hidden layer, presumably due to excess model capacity. All things considered, the error changes little and, for this relatively small number of batch iterations (25), the architecture of the net is not that important.

Table VIII. Preliminary exploration for neural networks.

Number of nodes	Activation function	Training algorithm	Number of iterations	Train error	Test error
8	Sigmoid	Back propagation	900	18,97%	19,11%
8	Sigmoid	Steepest descent	96	11,72%	10,84%
8	Sigmoid	Conjugate gradient	46	10,39%	9,88%
8	Sigmoid	Modified newton	36	11,26%	10,44%
8	Sigmoid	Genetic algorithm	9	13,14%	12,15%
8	Linear	Back propagation	900	13,23%	12,68%
8	Linear	Steepest descent	27	12,05%	11,10%
8	Linear	Conjugate gradient	20	12,00%	11,11%
8	Linear	Modified newton	21	11,99%	11,12%
8	Linear	Genetic algorithm	9	15,48%	13,82%
8	Hypertangent	Back propagation	900	13,71%	13,38%
8	Hypertangent	Steepest descent	156	10,86%	10,36%
8	Hypertangent	Conjugate gradient	35	10,28%	10,07%
8	Hypertangent	Modified newton	41	10,43%	9,92%
8	Hypertangent	Genetic algorithm	9	13,84%	13,06%

Table IX. Results for neural nets of different sizes trained for a fixed number of batch iterations (25).

Size	Test error at $m = 2,000$		Noise/bias α	Complexity β	Optimum training sample size (recs)
2	11.00%	(0.0032)	0.10723	5.69	5,689
4	11.04%	(0.0033)	0.10776	5.23	5,233
6	11.09%	(0.0035)	0.10749	6.36	6,357
8	11.09%	(0.0033)	0.10768	6.92	6,916
16	11.15%	(0.0032)	0.10847	6.20	2,877

The second approach explores the effect of changing the number of iterations while keeping the architecture constant. Table X and Figure 11 show the results for fixed architecture (8 nodes) but differing numbers of training iterations (10, 40, 80, and 100). Changing the number of iterations has the effect of changing the effective capacity of the neural network (Wang, 1994).

The curve with 10 iterations has the highest average error rate (11.92%) at 2,000 records, and it also achieves the highest asymptotic error value. The curve with

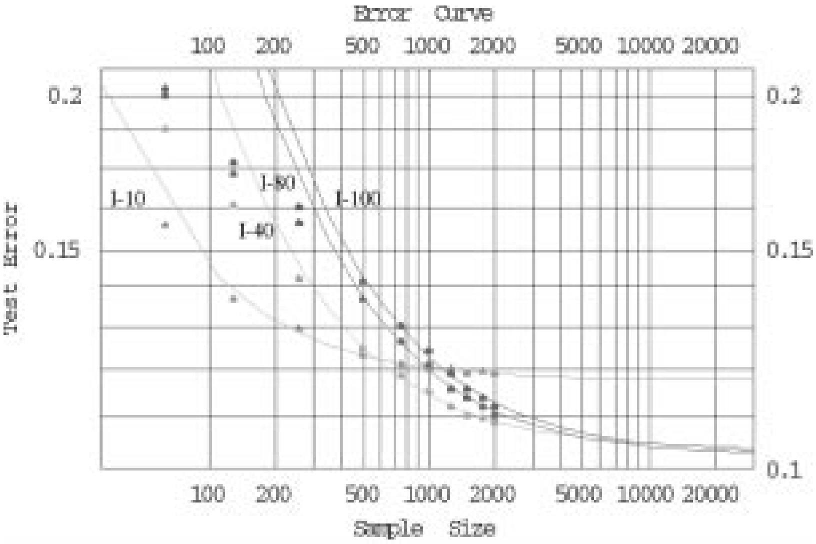


Figure 11. Generalization learning curves for 8-node neural nets with 10, 40, 80, and 100 iterations.

Table X. Neural nets with 8 nodes in hidden layer.

Iterations	Test error at $m = 2,000$		Noise/bias α	Complexity β	Optimum training sample size (recs)
10	11.92%	(0.0032)	0.11785	2.77	2,773
25	11.09%	(0.0033)	0.10768	6.92	6,916
40	10.89%	(0.0033)	0.10365	10.86	10,864
80	11.05%	(0.0034)	0.10240	17.57	17,567
100	11.19%	(0.0038)	0.10284	20.02	20,025

40 iterations has the second highest convergence point (10.03%). If we only look at the error rate achieved at the largest sample size of 2,000 records, this neural net would appear to have the lowest error rate; however, its speed of convergence (given by the absolute value of the slope of the curve) is slower than the one with 80 iterations. As a consequence, we might be tempted to chose 40 iterations as the optimal size. But the neural net with 80 iterations has the lowest asymptotic error and is therefore the optimal one. The neural net with 100 iterations has excess capacity, as evidenced by its having the second lowest asymptotic error value. We can also see decreasing asymptotic noise/bias for neural nets up to 80 nodes. For larger nets this parameter increases.

A similar effect occurs when we fix the number of nodes in the hidden layer at 16 while allowing the number of iterations to change. In this case, the network with 80 iterations is optimal, achieving the lowest asymptotic error rate of 10.22%,

Table XI. Neural network learning curve results with 16 nodes in hidden layer.

Iterations	Test error at $m = 2,000$		Noise/bias α	Complexity β	Optimum training sample size (recs)
10	12.08%	(0.0033)	0.11925	2.88	2,877
25	11.15%	(0.0032)	0.10847	6.20	6,202
40	10.94%	(0.0037)	0.10532	8.55	8,555
60	10.92%	(0.0038)	0.10272	14.09	14,087
80	11.00%	(0.0043)	0.10225	18.17	18,165
100	11.30%	(0.0043)	0.10352	20.71	20,713

Table XII. Neural net with 16 nodes and 80 iterations.

	Actual vs. predicted (performance matrix)				
	Predicted				
	0	1	Total	Total error	15.40%
Actual 0	437	55	492	Error for 0	11.18%
Actual 1	99	409	508	Error for 1	19.49%
	536	464	1,000		

while the one with 60 iterations achieves the lowest error rate at 2,000 records (10.92%). As before, the neural net with 100 iterations has excess capacity. The results are summarized in Table XI.

The differences between the optimal networks (80 iterations) with 8 and 16 nodes are relatively small. We choose the 16 node network as our ‘best’ net, whose performance matrix is shown in Table XII. It is interesting to note that we see the same type of asymmetry here as in the probit model: the error rate in identifying the non-default group (11.18%) is smaller than that in identifying the default group (19.49%).

3.5. K-NEAREST NEIGHBORS

K-nearest neighbors (k-NN) is an algorithm that differs from the others in that the data themselves provide the ‘model’. To predict a new record, it finds the neighbors nearest in Euclidean distance and then performs a weighted average or majority vote to obtain the final prediction. It works well for cases with relatively low dimensionality and complicated decision boundaries. The toolset we use (Appendix A) also supports the capability to ‘train’ global attribute weights so they are optimal in maximizing the prediction accuracy of the algorithm. For this purpose, one uses

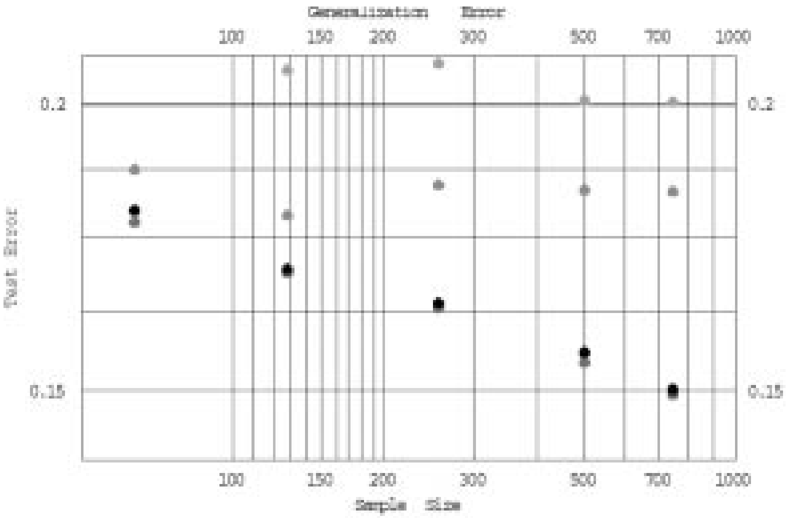


Figure 12. Error behavior for k-NN models.

a small training set of a few hundred additional records (250). This modification improves the results relative to the standard k-NN, but the algorithm retains its main characteristics. In practice, k-NN works better than expected, which may be due to the benign effect of its high-bias as has been suggested by Friedman (1997).

We perform a learning curve analysis similar to those described above. The results are shown in Table XIII and Figure 12. In this case, the train error is not reported; it is always zero since the ‘model’ is the data. A practical problem in applying k-NN to our mortgage-loan data set arises since the number of records is small. In high-dimensionality data sets, significant numbers of data may be required to overcome the ‘curse of dimensionality’ (Friedman, 1997). Since the model increases with the size of the data set, it is impossible to fit fixed-capacity learning-curve models as was done for the other cases. It is, however, possible to account for the change in model size in the fitting, but we did not attempt to do so here. A simple visual extrapolation shows that more than 20,000 records will be required to reduce the error rates for this model comparable to those of CART or the neural net.

As can be seen in Figure 12 and Table XIII, the optimal number of neighbors k appears to be about 24. The model attains a test error rate of 14.95%, which is significantly higher than that of the neural networks or CART but comparable to probit. We believe this results from the relatively small size of the model data set. The performance matrix for $k = 24$ shows the same pattern as the probit and the neural network models: the error rate in identifying the non-default group (12.40%) is lower than in identifying the default group (22.83%).

Table XIII. Error rates for k-NN.

k	Test error for 750 records	
2	20.05%	(0.0077)
4	18.32%	(0.0062)
8	17.25%	(0.0053)
16	15.53%	(0.0098)
20	15.05%	(0.0059)
24	14.95%	(0.0049)
28	15.03%	(0.0050)
32	15.05%	(0.0050)

Table XIV. Performance matrix for $k = 24$.

	Actual vs. predicted matrix				
	Predicted				
	0	1	Total	Total error	17.70%
Actual 0	437	61	492	Error for 0	12.40%
Actual 1	116	392	508	Error for 1	22.83%
	547	453	1,000		

3.6. SUMMARY AND COMPARISON OF RESULTS

Table XV summarizes the performances of the best models (error rates, complexity and optimal sample sizes). The best overall model is a decision tree with 120 nodes, which attains a test error (average) of 8.3% on the largest sample of 2,000 records. The asymptotic test error for this model is 7.3% (noise/bias = 0.073), indicating that, even with larger data sets, no further prediction accuracy could be attained with this type of model. The fact that this value is lowest among all algorithms further suggests that the intrinsic noise in the data set might be close to this value. This, then, is the *limit on accuracy imposed by data quality* as described by Cortes et al. (1994a). This model also has the smallest noise/bias parameter and, the highest complexity, confirming the hypothesis that the good models exploit the data more fully and converge more slowly to their asymptotic value. Based on the fitted model, it appears it will take at least 22,000 records to achieve optimal results with CART decision trees. This is the number of records needed to build a production-quality predictive risk model for this particular financial institution.

Occupying second position is a neural network with 16 hidden nodes trained for 80 iterations. This net attains a test error (average) of 11.0% on 2,000 records. The asymptotic test error estimated for the model is 10.2% (noise/bias = 0.102), which

Table XV. Summary of best models' performance, complexity and optimal sample sizes.

Model	Test error (2,000 recs.)	Noise/bias α	Complexity β	Optimum training sample size (recs.)
CART (120 nodes)	8.3%	0.073	21.7	21,675
Neural net (16,80)	11.0%	0.102	18.1	18,165
k-NN	14.95% (1,000 recs.)	–	–	–
Probit	15.13%	0.150	1.80	1,804

gives the limit of predictive obtainable with this type of model. We speculate that the difference of 3% in comparison with the best tree probably results from the bias in the network model and the likelihood that the optimal net training point (global minimum) was not attained in our training. The complexity parameter of 18.17 is not much less than CART is. It appears that a sample of at least 18,165 records will be needed to attain optimal results with this model.

The best k-NN, using 24 neighbors, attains 14.95% test error (average). The reason for this higher error is likely the small size of the 'model' data set. The dimensionality of the data set is relatively high indicating large numbers of records might be needed to obtain better results. A simple visual extrapolation indicates that more than 20,000 records are needed to produce error rates comparable to CART or the neural net. This algorithm is then, likely to be a viable alternative if one can obtain enough data records. Finally the probit model attains an average test error of 15.13%. Even though this method has the worse asymptotic test error and lowest complexity parameter – presumably due to the limitations of linear discriminants –, it is still competitive for small sample sizes. For example, as seen in Figure 13, it outperforms the other methods up to 128 records and competes well with the decision-tree.

We find that the use of learning curves and noise/bias and complexity parameters offers an interesting perspective in understanding the nature and character of different algorithms or data-fitting methods. Unfortunately, we don't currently have data sets from similar financial institutions to make a comparative study. With such data, one can compare the parameters of the models, and, in the case of CART, the profiles themselves to be able to assess degrees of similarity.[15]

It is interesting to note the similar structures of the performance matrices for the Probit, Neural Network and k-NN models, where the errors are higher in discriminating for the default group. The matrix for the CART model is different, possibly indicating one of the reasons this algorithm outperforms the others. This difference may be exploitable by combining different algorithms to improve predictions. Table XVI shows the results of an exploratory analysis of combining model predictions by the use of logical operators (AND and OR). This simple combination method shows potential. For example the best combination for predicting the non-

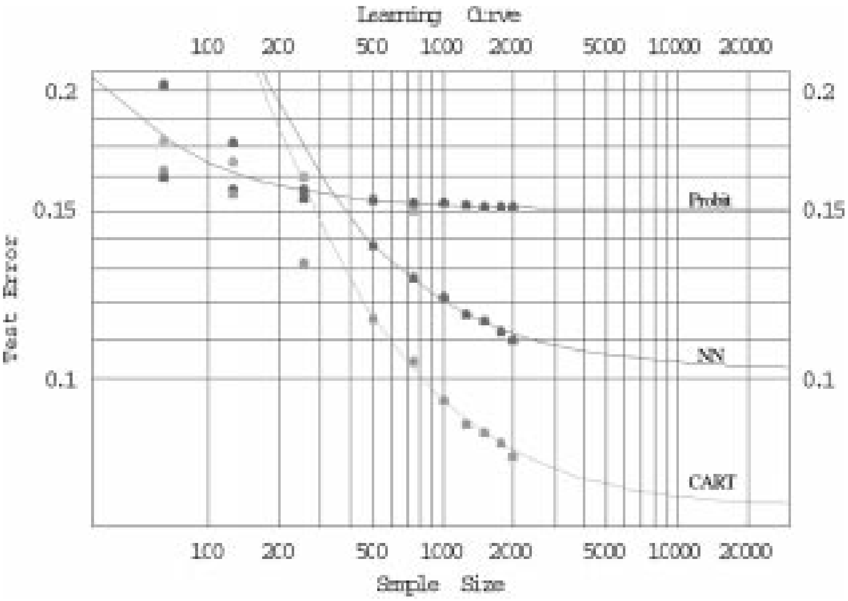


Figure 13. Comparison of results for 4 algorithms (Probit, CART, Neural Nets and k-NN.

defaulting group is (AND) CART and Probit (3.25%). We speculate this may come from model bias reduction and the nature of the performance matrix asymmetries. The best prediction for the defaulting group is obtained by combining (OR) CART and Neural Net (4.72). The overall absolute error is not below the CART error.

The next step is to use more sophisticated model combination methods based on adaptive re-sampling, such as boosting (Freund and Shapiro, 1995) and ARCing (Breiman, 1996). These methods have the potential to reduce global error by effectively reducing the variance and bias of the combined model.

4. Aggregation and Interpretation of Global Risk Models

In this section, we describe different ways to aggregate risk for one institution and for the entire financial system, and we comment on possible uses of the models in this regard.

4.1. AGGREGATION OF RISK FOR ONE INSTITUTION

Credit risk. Early warning systems introduced in the 1970’s and 1980’s are mostly phenomenological; i.e. they attempt to describe a phenomenon (failure/non-failure) with a coarse-grained model having a single modeling stage and no explicit decomposition of risk. Here we apply a more fine grained analysis based on previous calculations of default risk for individual borrowers, which are then aggregated across the entire portfolio.

Table XVI.

Model (s)	Absolute (%)	Error for 0 (%)	Error for 1 (%)
CART	9.10	11.99	6.30
k-NN	17.70	12.40	22.83
NeuralNet	15.40	11.18	19.49
Probit	15.80	6.10	25.20
CART AND k-NN	14.10	3.66	24.21
CART AND Neural Net	12.60	3.86	21.06
CART AND Probit	14.80	3.25	25.98
k-NN AND Neural Net	17.30	7.11	27.17
k-NN AND Probit	16.50	3.86	28.74
Neural Net AND Probit	15.80	4.88	26.38
CART OR k-NN	12.70	20.73	4.92
CART OR Neural Net	11.90	19.31	4.72
CART OR Probit	10.10	14.84	5.51
k-NN OR Neural Net	15.80	16.46	15.16
k-NN OR Probit	17.00	14.63	19.29
Neural Net OR Probit	15.40	12.40	18.31
Mayority rule (CART, NN, k-NN)	13.20	9.35	16.93

A simple way to aggregate the default risk of the credit portfolio is to multiply the probability of default by the amount of *capital at risk* for each loan and then sum over all loans. One may use a simple definition of *capital at risk*, such as the total debt minus the value of the guarantee. This single measure contains some information about the aggregate default risk in the portfolio, which can, in turn, also be used to estimate the amount of provisional reserves required for the portfolio. Another way to aggregate the credit exposure of the portfolio is by using Monte Carlo methods to generate the predicted future distribution for the value of the credit portfolio.[16] Briefly, in our case we would need to (1) generate default or non-default scenarios for the individuals in the portfolio according to the probabilities predicted by the models, (2) decide the recovery rate in the state of default (this is important because of the uncertainty about the recovery rate in the state of default), (3) aggregate the individual scenario to produce one instance of the future value of the portfolio, and (4) iterate to generate the distribution of the portfolio.

Other types of risk. This study focuses on credit risk for mortgage-loans, but the same methodology can be applied to other credit portfolios (e.g., credit cards, personal, and commercial loans) or used to analyze other risk factors such as pre-payment risk. From such an analysis, one can identify subsets of the portfolio that may be unbundled or packaged into new financial instruments with particular type of risks. These in turn could be sold to investors most willing to take these risks.

Consider trading portfolios for which we want to measure the *value at risk*. As is customary, the analysis should include all in- and off-balanced sheet positions of the portfolio and specify the risk factors that need to be analyzed (e.g., interest and exchange rates, stock and commodity prices, option volatilities, GDP growth, price indexes, etc.) However, here the classification techniques must be replaced by regression methods. To illustrate this, suppose we want to measure the value at risk of a given portfolio. As mentioned we must first decide on the N systematic risk factors $X_1, X_2, \ldots, X_N$ to consider. Then we decompose each security's return per dollar (R_j, $j = 1, \ldots M$) into its expected return, its factor exposures, and its 'idiosyncratic' risk (u_j). Traditionally this is done using the linear regression model

$$R_j = E(R_j) + b_{j1}X_1 + b_{j2}X_2 + \cdots + b_{jN}X_N + u_j\,. \tag{1}$$

If we wish a non-parametric representation, we estimate the form

$$R_j = f(E(R_j), X_1, X_2, \ldots, X_N) + u_j\,. \tag{2}$$

and then perform a sensitivity analysis to assess *ceteris paribus* movements of each of the factors. This gives the *factor exposure* (b_{ij}) of each security to every factor. The aggregate exposure (*AE*) of the portfolio along each factor X_i is then computed by,

$$AE_i = \sum_{j=1}^{M} V_j\, b_{ij} \qquad i = 1, \ldots, N\,, \tag{3}$$

where V_j is the nominal value of the position on security j. To get the value at risk, one expresses the return per dollar of the entire portfolio (R_P) as

$$R_P = \sum_{j=1}^{M} w_j E(R_j) + \sum_{i=1}^{N} B_i X_i + \sum_{j=1}^{M} w_j u_j\,, \tag{4}$$

where w_j is the proportion of asset j to the total value of the portfolio and,

$$B_i = \sum_{j=1}^{M} w_j b_{ji}, \qquad i = 1, \ldots, N\,. \tag{5}$$

Finally the *value at risk* is calculated in the standard way,

$$\textit{Value at risk} = \text{Value of portfolio } [E(R_P) - 2.33\text{Variance}(R_P)]\,. \tag{6}$$

Other applications of model's output. Another use of these techniques for corporate policy arises from the profiles (rules) given by the decision-tree (see Figure 6); Institutions, for example, may institute a policy for which the riskiest group is subject to a special process to reinforce the collection of the loan. An alternative policy might give benefits to borrowers to motivate payment. These policies must always be designed to motivate borrowers to pay their obligations. It is counter productive to implement policies with wrong incentives; this only aggravates the default problem. In general, incentive compatibility penalizes bad performance and rewards good performance.

Another possibility for using classification or predictive models is fine-grained segmentation for customer groups, as typically occurs in direct marketing (see for example Bigus, 1996; Bourgoin, 1994; Bourgoin and Smith, 1995). Such segmentation is accomplish both with respect to risk parameters and account payment modalities, revenue/ROI (Bourgoin and Smith, 1995), or customer equity group information (Blattberg and Deighton, 1996). This analysis is specially relevant for studies of corporate profitability or targeted marketing.

4.2. AGGREGATION OF RISK IN GLOBAL FINANCIAL SYSTEM MODELS

Regulatory authorities might need to gather information from the entire financial system in order to develop a global model of risk. Initially, the analysis could be done for representative institutions to assess differences and similarities among the models (e.g. error rates, noise/bias, and complexity.) The rules from decision-tree models and the sensitivity/importance of the factors can also be subject to comparison. If different models are seen to have common properties, then generalizations for the system (the universe in question) could be established.

This approach requires risk models to be built for each institution, processed according to equations 1-6, and then summed across all the institutions. This may work well if the country in question has a very concentrated industry since the calculations involve a relatively small number of institutions. For example, Mexico's banking system has its three largest banks holding more than 58% of the mortgage loan market (as of June 1996). By contrast if the market is diluted, as is the case of the United States, the number of institutions will be in the order of thousands.

A less demanding computational and informational approach takes a representative sample of assets and liabilities of the system and calculates the global risk from this sample. The exposure of this sample is determine to estimate the global risk of the system. This approach requires less information, and, while it might lack resolution, it is easy to manipulate and compute.

The regulatory authority can also use the model output in a manner similar to the way institutions might use it for corporate strategy. For example, after the Mexican crisis of 1994–1995, the government implemented financial aid programs that targeted borrowers of different types of loans such as private, business, mortgage, and credit cards. These programs tried to lessen the burden of interest accumulation while retaining borrower's incentives to fulfill their payments. This type of policy benefits from more accurate classification of the groups so that their overall impact can be measured more precisely.

5. Conclusions

We find that a combination of different strategies and the application of a systematic model building and selection methodology offers an interesting perspective for understanding the characteristics and utility of different algorithms or data-fitting methods. The use of state-of-the-art, high-performance modeling tools allows us to make a systematic study of the behavior of error curves by building thousands of models. We analyze the performance of four algorithms for a mortgage loan data set and find that decision trees produce the most accurate models. We analyze the different ways in which these institutional models can be combined to provide models of global risk. Further research will extend this analysis to include other risk factors and institutions to allow a comparative study.

Acknowledgements

Jorge Galindo-Flores thanks Professors R. Freeman, J. Campbell, and A. Metric for the support and encouragement given to this project; and to CONACYT (Mexico) for financial support (fellowship # 84537). We wish to acknowledge CNBV for contributing the data sets used in the study and to Thinking Machines Corp. for providing computer time and access to the Darwin toolset. Some aspects of the model building and error curve analysis methodology presented in this paper have been developed at Thinking Machines Corp. by J. Berlin, N. Dayanand, G. Drescher, D.R. Mani, C. Wang and one of the authors (P. Tamayo).

Appendix A. Brief Summary of Software and Toolsets Used in the Study

Stata (probit models). Stata is a general purpose statistical package with capabilities for data management, statistical functions, graphs and displays, and programming features. For more information see www.stata.com:80

Darwin (CART, Neural Networks and k-NN). Darwin is a high-performance scalable multi-strategy toolset for large scale Data Mining and Knowledge Discovery. More detailed information can be found at www.think.com.

Mathematica (model fitting). Mathematica is an integrated environment for numerical computations, algebraic computations, mathematical functions, graphics, and optimization algorithms. For more information see www.wolfram.com

Notes

[1] See Introduction in Dewatripont and Tirole (1994).

[2] An interdisciplinary *Knowledge Discovery* approach to find the patterns and regularities in data has taken form over the last five years (see for example Piatestky-Shapiro and Frawley, 1991; Fayyad et al., 1996; Simoudis et al., 1996).

[3] Elder and Pregibon (1996) argue that if accuracy is acceptable a more interpretable model is more useful than a ‘black box’.

[4] These studies were mainly sponsored by regulatory institutions like the Federal Reserve and the Federal Depository Insurance Commission (FDIC).

[5] The most commonly used statistical methods for this approach are linear, quadratic, logit, and probit discriminant analysis.

[6] The April 1995 proposal of the Basle Committee on Banking Supervision allows banks to use *in-house* models for measuring market risk to calculate *'value at risk'*. Market risk is defined as the risk of losses in- and off- balance sheet positions arising from movement in market prices. For more on this see the Basle Committee on Banking Supervision (1997).

[7] The importance of transparency has been advocated by Ralphe Wiggins in the context of business data mining.

[8] See Elder and Pregibon (1996).

[9] Masking takes place, for example, when one of the attributes is highly correlated with another one and then the model ignores it by choosing only the first attribute.

[10] However as expected the inverse power law model does not describe well the learning curve behavior for small samples so we excluded small training samples from the fit (this is done consistently for all the algorithms).

[11] For example, *Soc_ Interest* takes the value of 1 if the loan belongs to a special program that charges a soft interest for households with low economic resources.

[12] The error term is heterocedastic and this produces a loss of efficiency in the estimation; the distribution of the error is not normal and this precludes the use of the usual statistical tests; the predictions of the model may be outside the unit interval and therefore loose their meaning under a probabilistic interpretation. See, for example, Pindyck (1981) for more on comparing these binary choice models.

[13] Greene (1993) p. 638.

[14] We use the sensitivity/importance analysis provided by the toolset that computes these numbers by integrating out each of the variables in the model to measure the relative effect on the prediction results.

[15] As a complementary note, we would like to mention that this type of analysis has been applied to a U.S. 1994 Census data set (UCI repository (http://www.ics.uci.edu/ mlearn/MLRepository.html.)), where one considers the prediction of high and low income. This produced values of 0.141 and 49.0 for the noise/bias and complexity, respectively, suggesting that our home mortgage data set is less noisy and less complex than the Census data set.

[16] For example, Credit Metrics from J.P. Morgan uses Monte Carlo simulation to obtain the future distribution of the portfolio. For more on this see http://jpmorgan.com/RiskManagement/CreditMetrics/CreditMetrics.htm

References

Adriaans, P. and Zantinge, D. (1996). Knowledge discovery and data mining.

Altman, E., Avery, R.B., Eisenbeis, R.A. and Sinkey, J.F., Jr. (1981). *Application of Classification Techniques in Business, Banking and Finance.* Jai Press Inc.

Amari, S. (1993). A universal theorem on learning curves. *Neural Networks*, **6**, 161–166.

Amis, E. (1984). *Epicurus Scientific Method*. Cornell University Press.

Basle Committee on Banking Supervision (1997). Compendium of documents (April) Vol. 2 Advanced supervisory methods, Chapter ll, pp. 82–181.

Berger, J.O. (1985). *Statistical Decision Theory and Bayesian Analysis*. Springer series in Statistics.

Bigus, J.P. (1996). *Data Mining with Neural Networks: Solving Business Problems from Application Development to Decision Support.*

Black, F. and Scholes, M.S. (1973). The pricing of options and corporate liabilities. *Journal of Political Economy*, **81** (May/June), 637–654.

Blattberg, R.C. and Deighton, J. (1996). Manage marketing by the customer equity test. *Harvard Business Review* (July/August).

Breeden, D.T. (1979). An intertemporal asset pricing model with stochastic consumption and investment opportunities. *Journal of Financial Economics*, **7** (September), 265–296. Reprinted in Bhattacharya and Constantinides, eds. (1989).

Breiman, L., Friedman, J.H., Olshen, R.A. and Stone, C.J. (1984). *Classification and Regression Trees*. Wadsworth Inc., Pacific Glove.

Breiman, L. (1996). Bias, variance, and arcing classifiers. Tech. Rep. 460, Statistics Dept. U. of California, Berkeley (April 1996).

Bourgoin, M. (1994). *Applying Machine-Learning Techniques to a Real-World Problem on a Connection Machine CM-5*.

Bourgoin, M. and Smith, S. (1995). Leveraging your hidden data to improve ROI: A case study in the credit card business. In Freedman, Klein and Lederman (eds.), *Artificial Intelligence in the Capital Markets*. Probus Publishing.

Carlin, B.P. and Louis, T.A. (1996). *Bayes and Empirical Bayes Methods for Data Analysis*. Chapman and Hall.

Cortes, C., Jackel, L.D. and Chiang (1994a). W-P limits on learning machine accuracy imposed by data quality. In G. Tesauro, D.S. Touretzky and T.K. Leen (eds.), *Advances in Neural Networks Processing Systems*, Vol. 7, p. 239, MIT Press.

Cortes, C., Jackel, L.D., Solla, S.A. and Vapnik, V. (1994b). Learning curves: Asymptotic values and rate of convergence. In G. Tesauro, D.S. Touretzky and T.K. Leen (eds.), *Advances in Neural Networks Processing Systems*, Vol. 6, p. 327, MIT Press.

Dewatripont, M. and Tirole, J. (1994). *The Prudential Regulation of Banks*. MIT Press.

Eaton, M.L. (1983). *Multivariate Statistics*. Wiley, New York.

Elder and Pregibon (1996). A statistical perspective on knowledge discovery in databases. In *Advances in Knowledge Discovery and Data Mining*. AAAI Press/The MIT Press.

Fayyad, U.M., Piatetsky-Shapiro, G., Smyth, P. and Uthurusamy. R. (eds.) (1996). *Advances in Knowledge Discovery and Data Mining*. AAAI Press/The MIT Press.

Fletcher, R. (1981). *Practical Methods of Optimization*. Wiley-Interscience, John Wiley and Sons.

Fisher, R. (1950). *Statistical Methods for Research Workers*. 11th Edition.

Friedman, J.H., Bentley, J.L. and Finkel, R.A. (1977). An algorithm for finding best matches in logarithmic expected time. *ACM Transactions on Mathematical Software*, **3**, 9–226.

Friedman, J.H. (1997). On bias, variance, 0/1 – loss, and the curse of dimensionality. *Data Mining and Knowledge Discovery*, **1**, 55–77.

Freund, Y. and Shapire, R.E. (1995). A decision theoretic generalization on on-line learning and an application to bosting. *Computational Learning Theory*. 2nd European Conference, EuroCOLT'95, pp. 23–27. http://www.research.att.com/orgs/ssr/people/yoav

Fukunaga, K. (1990). *Introduction to Statistical Pattern Recognition*.

Glymor, C., Madigan, D., Pregibon, D., and Smyth, P. (1997). Statistical themes and lessons for data mining. *Data Mining and Knowledge Discovery*, **1**, 11–28.

Greene, W.H. (1993). *Econometric Analysis*. Macmillan, 2nd Edition.

Goldberg, D. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley.

Hand, D.J. (1981). *Discrimination and Classification*. John Wiley, Chichester.

Hassoun, M.H. (1995). *Fundamentals of Artificial Neural Networks*. MIT Press, Cambridge, Mass.

Horst, R. and Pardalos, P.M. (eds.) (1995). *Handbook of Global Optimization*. Kluwer.

Hume, D. (1739). *An Inquiry Concerning Human Understanding*. Prometheus Books, Pub. 1988.

Hutchinson, J.M., Lo, A.W. and Poggio, T. (1994). A non-parametric approach to pricing and hedging derivative securities via learning networks. *The Journal of Finance*, **XLIX** (3).

Jaynes, E. (1983). *Papers on Probability, Statistics and Statistical Physics*. R.D. Rosenkrantz (ed.), D. Reidel Pub. Co.

Jeffreys, H. (1931). Scientific inference. Cambridge Univ. Press.

Kearns, M.J. and Vazirani, U.V. (1994). *An Introduction to Computational Learning Theory*. MIT Press, Cambridge, Mass.

Keuzenkamp, H.A. and McAleer, M. (1995). Simplicity, scientific inference and econometric modeling. *The Economic Journal*, **105**, 1–21.

Kuan, C.-M. and White, H. (1994). Artificial neural networks: An economic perspective. *Econometric Reviews*, **13** (1).

Lachenbruch, P.A. and Mickey, M.R. (1968). *Discriminant Analysis*. Hafner Press, New York.

Landy, A., 1996. A scalable approach to data mining. *Informix Tech Notes*, **6** (3), 51.

Li, M. and Vitanyi, P. (1997). *An Introduction to Kolmogorov Complexity and Its Applications*. 2nd Edition, Springer-Verlag, New York.

McClelland, J.L. and Rumelhart, D.E. (eds.) (1986). *Parallel Distributed Processing.* MIT Press.

McLachan, G.L. (1992). *Discriminant Analysis and Statistical Pattern Recognition.* John Wiley, New York.

Meyer, P.A. and Pifer, H.W. (1970). Prediction of bank failures. *Journal of Finance*, **25** (4), 853–868.

Merton, R.C. (1973). Theory of rational option pricing. *Bell Journal of Economics and Management Science*, **4** (Spring), 141–183.

Merton, R.C. (1973). An intertemporal capital asset pricing model. *Econometrica*, **41** (September), 867–887. Reprinted in *Continuous Time Finance* (1990). Basil Blackwell as Chapter 15, Cambridge, Mass.

Michie, D., Spiegelhalter, D.J. and Taylor, C.C. (eds.) (1994). Machine learning, neural and statistical classification. Ellis Horwood series in Artificial Intelligence.

Mitchell, T. (1997). *Machine Learning*. McGraw Hill, http://www.cs.cmu.edu/ tom/mlbook.html

Opper, M. and Haussler, D. (1995). Bounds for predictive errors in the statistical mechanics of supervised learning. *Physical Review Letters*, **75**, 3772.

Piatetsky-Shapiro, G. and Frawley, W.J. (eds.) (1991). *Knowledge Discovery in Databases.* MIT Press.

Pindyck, R.S. and Rubinfield, D.L. (1981). *Econometric Models and Economic Forecasts.* 2nd Edition, McGraw Hill.

Popper, K. (1958). *The Logic of Scientific Discovery.* Hutchinson & Co, London.

Rissanen, J.J. (1989). *Stochastic Complexity and Statistical Inquiry.* World Scientific.

Ross, S.A. (1976). Arbitrage theory of capital asset pricing. *Journal of Economic Theory*, December.

Sharpe, W.F. (1963). A simplified model for portfolio analysis. *Management Science*, **9** (January), 277–293.

Sinkey, J.F., Jr. (1975). A multivariate statistical analysis on the characteristics of problem banks. *Journal of Finance*, **30** (1), 21–36.

Simoudis, E., Han, J. and Fayyad U. (eds.) (1996). Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD'96). AAAI Press. See also KDD Nuggets: http://info.gte.com/k̃dd/

Small, R.D. and Edelstein, H. (1997). *Scalable Data Mining in Building, Using and Managing the Data Warehouse*. Prentice Hall PTR.

Stanfill C. and Waltz, D. (1986). Toward memory-based reasoning. *CACM*, **29**, 12l.

Seung, H.S., Sompolinsky, H. and Tishby, N. (1993). Statistical mechanics of learning from examples. *Physical Review A*, **45**, 6056.

Tamayo, P., Berlin, J., Dayanand, N., Drescher, G., Mani, D.R. and Wang. C. (1997). *Darwin: An Scalable Integrated System for Data Mining.* Thinking Machines white paper.

Wang, C., Venkatesh, S.S. and Judd, J.S. (1994). Optimal stopping and effective machine complexity in learning. In G. Tesauro, D.S. Touretzky and T.K. Leen (eds.), *Advances in Neural Networks Processing Systems*, **7**, 239. MIT Press.

Weiss, S.M. and Kulikowski, C.A. (1991). *Computer Systems that Learn: Classification and Prediction Methods from Statistics, Neural Networks, Machine Learning and Expert Systems.* Morgan Kaufmann, San Mateo, Calif.

White, H. (1992). *Artificial Neural Networks*. Blackwell, Cambridge, Mass.

Valiant, L.G. A theory of the learnable. *Communications of the ACM*, **27**, 1134.

Vapnik, V. (1995). *The Nature of Statistical Learning Theory.* Springer-Verlag.

Zadeh, L.A. (1994). Fuzzy logic, neural networks and soft computing. *Communications of the ACM*, **3**, 77.