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

Article *in* Computational Economics · February 2000

DOI: 10.1023/A:1008699112516 · Source: RePEc

2 authors:

Jorge Galindo
Instituto Tecnológico Autónomo de México (ITAM)
2 PUBLICATIONS **285** CITATIONS
SEE PROFILE

Pablo Tamayo
University of California, San Diego
454 PUBLICATIONS **138,249** CITATIONS
SEE PROFILE

*** D R A F T ***

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

J. Galindo

Department of Economics, Harvard University, Cambridge MA 02138

and

P. Tamayo

Thinking Machines Corp., 16 New England Executive Park, Burlington MA 01803

December 19, 1997

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 dataset 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.

Contents:

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 away 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. An important ingredient to make accurate and realistic risk models is to have accurate predictors of individual risk and a systematic methodology to generate them. This will be the main subject of this paper. Obviously this type of risk models are also of corporate interest for the financial intermediaries themselves.

In this context we make a comparative analysis of different methods of classification on a mortgage loan dataset from a large commercial bank. The motivation is to understand the limitations and potential of different methods and in particular the ones based on machine learning techniques (Michie *et al* [1994]; Mitchell [1997]). This is done by a systematic study and comparison with traditional statistical classification techniques. A multi-strategy approach is used where several algorithms are applied to the same data and their results compared to find the best model. This is justified by the fact that it is very hard to select an optimal model a priory without knowing the actual complexity of a particular problem or dataset. We also introduced a specific methodology for model analysis based on the study of error curves to estimate the noise/bias and complexity of the model and dataset. This methodology provides important insights into the nature of the problem and allows us to address some fundamental questions such as: How noisy is the dataset? How complex is the classification problem? How much data is needed for optimal prediction results? and, What is the best technique for this problem? Past studies comparing different approaches to the classification problem have been sometimes rightly criticized for using only one technique, for not being done in a systematic way or for consisting of mainly anecdotal results. We try to overcome this problem by systematically analyzing a variety of methods including: statistical regression (probit), decision-trees (CART), neural networks and *k*-nearest-neighbors on the same dataset. There are several other advantages in comparing different methods in the same study: the pre-processing of the data is more homogeneous and the results can be compared in a more direct manner. Using state-of-the-art modeling techniques we built more than 9,000 models for one dataset as part of the study.

Many of these new algorithms and methods were originally used by statisticians, computer or physical scientists but nowadays their use have spread successfully to many business applications

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

(Adrians and Zatinge [1996]; Bigus [1996], Bourgoin [1994], Bourgoin and Smith [1995])[2]. Studies of this type within economics have been mostly concerned with neural networks. For example, Hutchinson, Lo, and Poggio [1994] found that neural networks can recover the Black-Scholes formula from a two-year simulated dataset of daily option prices. Kuan and White [1994] tested the approximation abilities of a single hidden layer neural network with that of linear regression in three deterministic chaos examples. The neural network outperforms the linear specification by an ample margin. We believe more work is needed to validate the wide variety of new algorithms and methods available to the modeler.

One serious problem that one faces in global financial modeling is that of scale: in order to make a good global model one may need to produce hundreds of individual models; for example, one for each financial intermediary. This means that the model building process has to be systematized and the models have to be built almost automatically because it is impossible to build them one by one. This is both a computational and a conceptual challenge. One would like to develop a methodology for large scale modeling based on general induction principles so that each individual model selected is close to optimal. The computational resources and technology allow us today, perhaps for the first time, to tackle these problems. A general framework for large-scale economic modeling using machine learning methods will undoubtedly be of great utility. One can envision global models that by incorporating thousands of individual predictive models for risk could provide invaluable information and knowledge for regulatory authorities and macro-economists. The existence of such high level informational infrastructure will take advantage of the ever increasing amounts of data being accumulated in government and corporate databases (Adrians and Zatinge [1996]; Bigus [1996]; Landy [1996]).

Another issue of particular importance for financial decision making we briefly address is the *transparency* or degree of *interpretability* of models. Transparent models are those that can be conceptually understood by the decision-maker. An example of a transparent model is a decision tree expressed in term of profiles or rule sets. Other models such as neural networks can act as very accurate black boxes but at the same time are very opaque in the sense of not providing any 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. One way is traditionally called *early warning system*; the other *is risk decomposition and aggregation* of net risk exposure.

[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] and 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”.

Altman [1981] offers a survey on early warning systems studies performed in the 1970's and early 1980's[4]. Early warning systems rely on some 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]). The methodology groups the financial institutions into two or more categories and then performs some type of statistical discrimination using accounting data information. The problem then becomes predicting failure or problem conditions based on the explanatory variables. We can say that this analysis is phenomenological since it only attempts to describe the failure of the whole institution without making any analytical assessment of 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. Sharpe [1964], Lintner [1965], and Mossin [1966] introduced the Capital Asset Pricing Model (CAPM). The CAPM is developed in a one period set up but this limitation is overcome by Merton's [1973] Intertemporal Capital Asset Pricing Model (ICAPM) and by Breeden's [1979] Consumption Capital Asset Pricing Model (CCAPM). The ICAPM showed that, in equilibrium, the return of financial securities is not only proportional to the risk premium on the market but also to other sources of risk. The CCAPM showed the relation between the return of the securities and aggregate consumption for state independent utility functions. Ross' [1976] Arbitrage Pricing Theory (APT) relaxes CAPM's necessity to observe the market portfolio. All these capital asset pricing models state some dependency of asset prices to risk factors. One drawback of multi-factor models is that besides the market risk they provide little clue to what other risk factors should be considered. Black and Scholes [1973] and the Theory of Rational Option Pricing by Merton [1973] showed that under certain conditions the price of derivatives could 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 will attempt to decompose assets and liabilities classes into exposures to some previously defined risk factors and then to aggregate each exposure along every risk factor. This decomposition relies on the proper identification of the factors and accurate estimation of the exposures. This is one 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 much more information and calculations than an early warning system. The accuracy attained by each of the methodologies is a matter of empirical study.

The methodology developed here could provide the inputs for credit portfolio modeling, or with some modifications, calculate exposure to different factors such as interest, exchange rates, equity

[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.

and commodity prices, and price volatilities[6]. These in turn may serve to compute the "*value at risk*" of different portfolios. Other applications for descriptive and predictive models of risk are rather diverse. For instance, one can estimate the amount of capital provisions, design corporate policy, or perform 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 the financial or business data. Conceptually this is not a new subject and in some ways it is the logical extension and generalization of the methods that have been used by statisticians for decades. For complex real-world data, where noise, non-linearity and idiosyncrasies are the rule, the best strategy is to take an interdisciplinary approach that combines statistics and machine learning algorithms. This type of interdisciplinary, data-driven computational approach, sometimes referred as *Knowledge Discovery in Databases* (Fayyad *et al* [1996], Simoudis *et al* [1996], Bigus [1996], Adrians and Santinge [1996]), is specially relevant today due to the convergence of three factors: I) *Corporate and government financial databases*, where all and every financial transaction can be stored, have growth in size, number and availability. 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 are highly affordable. 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 priory because one might not know the nature and characteristics of the dataset, e.g. its intrinsic noise, complexity or the type of relationships it contains. Algorithms vary enormously

[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].

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]).
- 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]).
- 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.

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.

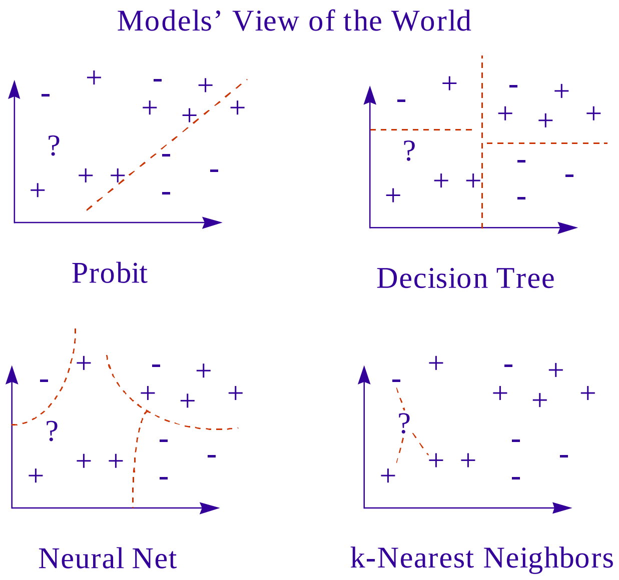


Figure 1. Different models' view of the world. Each algorithm builds a model that represents correlations or regularities according to a particular structure or representation. A new record "?" will be classified according to the prescription of each model's structure (e.g. the particular decision domains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

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 (Fisher [1950], Hand [1981], Lachenbruch and Mickey [1975])
- Bayesian Inference and the Maximum Entropy Principle (Jeffreys [1931], Jaynes [1983]).
- 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 won't review them here but we want to make the reader aware of their existence. Historically many of these were developed independently but recently there has been some progress in terms of understanding their relationships and equivalence in some cases (Li and Vitanyi [1997], Rissanen [1989], Vapnik [1995] and 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 functional relationship in the data (potentially subject to additional constraints such as the ones imposed by the model structure itself).

Perhaps the oldest, and certainly most accommodating induction principle, is the one advocated by Epicurus (Amis [1984]) which basically states: *keep all models or theories consistent with data*. At the other side 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]) 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 therefore its generalization ability and finite sample behavior explicitly (Vapnik [1995]). Other class of principles, the modern versions of the celebrated Occam's razor *(choose the most parsimonious model that fits the data*), are the Minimum Description Length (MDL, Rissanen [1989]), or the Kolmogorov complexity (Li and Vitanyi [1997]), which choose the best model based on finding the shortest or more succinct computational representation or description. These inductive principles have much more in common that what appears at first sight. Particular instances of them are familiar in the form of function approximation and parameter or density estimation, neural net training methods, data compression algorithms, etc. A general protocol for learning from a computational perspective, the Probably Approximately Correct (PAC) model (Kearns and Vazirani [1994]), has been introduce by Valiant [1983] as an attempt to reduce the ambiguity of earlier formulations.

The process or *building* a model and its *application* to new data examples imply a practical computational cost. This has to be taken into account as it may limit the type or models that can be used in a particular situation. Figure 2 shows the relationships between data and models and the deductive, inductive and transductive processes.

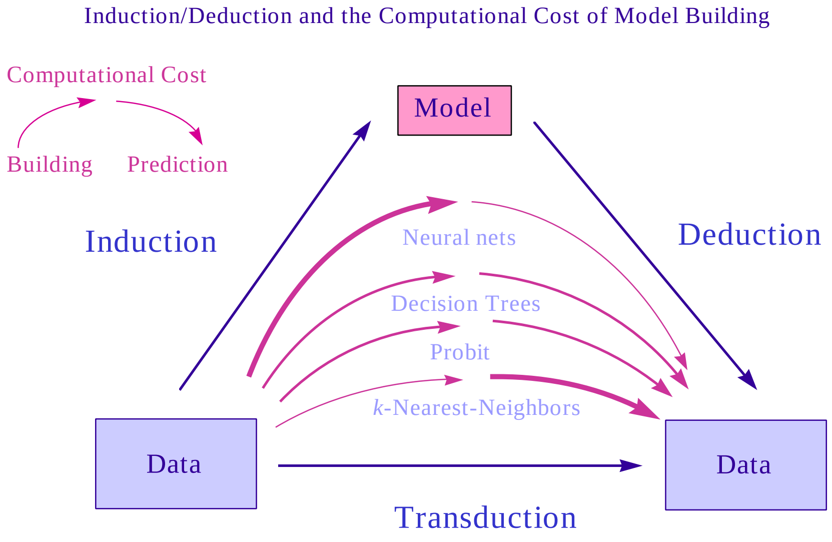


Figure 2. Inductive models: its relationship with data and their computational cost. Models are build with training data and become short representations of the logical or statistical relationships in it. Once a model has been built it can be applied to classify or predict new data in a deductive way. 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).

In Economics and Finance classification or predictive models derived from data are not used in isolation but as part or a larger model or in conjunction with interpretative theories and 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 non-trivial results; iv) *feasible*, in terms of time and resources; and v) *transparent*[7] *and interpretable*[8], in the sense of providing high level representations 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 quite complex, non-normal, non-linear and reflect structural changes such as demographic or market seasonal trends. To some extent one could argue that each dataset 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 original idea of parameter estimation in Statistics. Machine Learning algorithms tend to be much more computational-based and data-driven, and by relying less on assumptions about the data (normality, linearity, etc.), tend to be 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 (i.e. obscure as for example neural nets). The moral is that no single method or algorithm is perfect or guaranteed to work always so one should be aware of the limitations and strengths of each of them. For an interesting discussion about statistical themes and lessons for machine learning methods we refer the reader to Glymor *et al* [1997]. Another difference between new and traditional approaches is that the new algorithms have a more explicit way at taking into account the actual complexity, size or capacity of the model.

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 analysis that we employed in the four algorithms considered in the study. The main elements of the analysis methodology 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

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

[8] See Elder and Pregibon [1996].

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 the ones that are not very relevant to the model. One has to be careful in this filtering process because there are often complicated effects such as the "masking[9]" of variables. In the dataset considered in this paper the number of variables was small enough that we did not have to worry particularly about variable selection, however we analyzed the importance/sensitivity of the variables in the final model.

Train, test/generalization and evaluation error analysis. In classification problems the most direct measure of the performance is the misclassification error: the number of incorrectly classified records divided by the total. For a given binary classification problem this number will vary between the default prediction error (from assigning all records to the majority class) and zero for a perfect model. It is important to measure this error for both, the sample used to build the model and a "test" sample dataset containing records not used in the model construction. This allow us to select the best model in terms of 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 dataset. The format we use for these matrices is shown in Table I.

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

	Actual vs Predicted (Performance Matrix)			
	Predicted (by model)			
	0	1	Total	Total Error
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)

This matrix is useful because it allow us to distinguish asymmetries in the predictions (e.g. false/positives). Once a reasonable model for a given class has been selected a final estimate of error is done with an independent sample (the evaluation dataset). For example for this part of the analysis we divided our 4,000 records of data in the following subsets: 2,000 for training, 1,000 for testing and 1,000 for evaluation. This is a relatively small amount of data but it was all we had available for the study. As we will see in the final results a dataset of approximately 22,000 records will be needed to obtain optimal results for this problem.

Analysis of learning curves and complexity. This, more exploratory approach, is done by computing the average values of train and test (generalization) errors for given values of training

[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.

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 for the complexity and noise in the dataset. 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 relevant to plan larger modeling efforts done in production rather than exploratory datasets. This analysis also allows us to view the problem from the perspective of structural risk minimization (Vapnik [1995]) and bias/variance decomposition (Breiman [1996], Friedman [1997]).

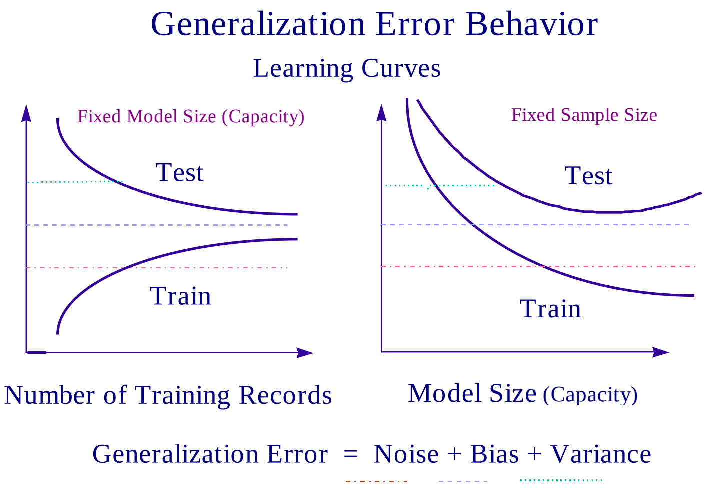

Generalization Error Behavior
Learning Curves
Fixed Model Size (Capacity)
Test
Train
Number of Training Records
Fixed Sample Size
Test
Train
Model Size (Capacity)
Generalization Error = Noise + Bias + Variance

Figure 3. Generalization error behavior and error curves.

Figure 3 describes the basic phenomenology of learning curves. For fixed model size, as the training dataset increases, the train and test errors converge to an 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 as more data is available the model, having a pre-determined fixed size, finds harder and harder to fit and "memorize" it. For very small sample sizes the train error could be zero i.e. the model performs a “lossless” compression of the training data. For a given training sample size there is an optimal model size where the model neither underfits nor overfits the data.

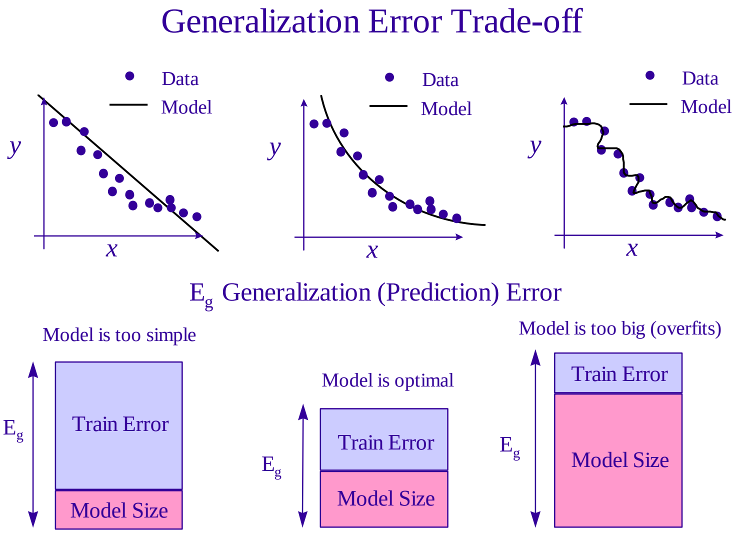


Figure 4. Generalization error trade-off in terms of model size.

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 also be limited by missing important trends. If the model is too large then it overfits the data and loses generalization power by incorporating too many accidents in the training data not shared by other datasets. 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 better the data. However at some point the model size starts to be too large and overfitting sets in resulting in the curve moving upwards. This fundamental behavior is shared by finite-sample inductive models in general [Kearns and Vazirani [1994], Vapnik [1995]) and agrees well with the empirical behavior we observed in all of our models.

The methodology we used for the analysis of mortgage-loan learning curves is as follows: for each dataset size, and keeping the model size fixed, we built 30 models with different random samples from the same original dataset and then averaged the on- (train) and off-sample (test) error rates. These averaged errors were then fitted to an inverse power law: $E_{test} = a + b / m^d$, where a estimates the noise/bias, b and $d$estimate the complexity and m is the sample size. Based on our previous experience and work reported in the literature (e.g. Cortes [1994a-b]) this model appears to work well 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 5-13. 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 d 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 worse case, suggest a value of $d \sim 1$ (Opper and Haussler [1995], Amari [1993]). There is also empirical support for this value from earlier work (Cortes [1994a-b]). We find that for our mortgage-loan dataset $d= 1$ provides a reasonable fit for the error curves and therefore we assumed $d = 1$ when fitting the data[10] Table II shows the basic format we will use to report the curve analysis results.

Table II. Format for the results of learning curve analysis fitting the model: $E_{test} = a + b / m$

Model	Test Error at maximum training sample (standard dev. in parenthesis)	Noise/Bias *a*	Complexity *b*	Optimum training sample size[recs]

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 datasets with good results. Similar methodologies have been reported in the literature (Cortes [1994a-b]) but their widespread use have been limited by the high computational cost of the method. The scaling analysis can be improved in many ways and particularly by extending the model to account for model size to describe the entire learning manifold. This will be the subject of 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 was 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 US) 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 3.7 which contains a summary of results.

3.1 Data Analysis, Preparation and Pre-processing.

The data was already being used for a regression model by the CNBV and therefore it required little pre-processing or manipulation prior to model building. It consists of a single dataset of 4,000 records, each of them corresponding to a customer account, and 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* in such way that the account is considered as "defaulted" only if no payments were made in the last two months. *Credit_Amount* is the value of the credit, *Unpaid_Bal* is the unpaid

[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).

balance, *Overdue_Bal* is the overdue balance, and *Debt* is the total debt 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* take the value 1 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 0 standard deviation for the dataset considered: i) *Adquisition,* takes the value of 1 if the loan was for acquiring an already existing house, and 0 otherwise; ii) *Construction,* takes the value of 1 if the loan was for construction of a new house, and 0 otherwise; iii) *Liquidity,* takes the value of 1 if the loan was to provide liquidity for things such as house remodeling, and 0 otherwise; iv) *Adq_or_Const*, takes the value of 1 if the loan was for buying or constructing the house, and 0 otherwise. These variables remain constant for all record in the dataset and provided no information to explain the dependent variable. Ten variables, *Month1-Month10* contains information of the payment history from June 1995 to March 1996. For each month a 1 entry means that there was no payment in that period, and 0 otherwise.

In addition to the 24 variables, a new variable *Default_Index* was created to condensed information about the payment history and probability of payment in a single attribute. A similar combined variable was useful in the regression model built by CNBV and we decided to 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 means that a payment is made and 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,

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

With the available information the following one-step transition matrix P_{ij}^1 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 experience 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_{i1}^{11-k} if the account is in state *i* in the *kth* month.

03.2 Probit results.

[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.

Traditionally binary classification problems had used 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 in turn produces small differences in the model, however there are no theoretical grounds to favor one technique over the other[13]. The following probit model was developed by us following the guidelines of a similar model used at CNBV. We will use it as our benchmark to compared other algorithms (i.e. the other three methods).

$$P\{Default = 1\} = \Phi(bx_i)$$

where the index bx_i is defined as,

$$bx_i = b_0 - b_1 Dgurantee1_i + b_2 Default\_index_i - b_3 Soc\_interest_i + b_4 Construction_i + b_5 Dguarantee1_i\, Default\_index_i$$

and $F(x)$ is the cumulative normal distribution. Alternative specifications were 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 models remain in the same error range than the one from CNBV. Therefore we decided to use the same specification as CNBV. To assign each predicted probability to the default or non-default group we had to choose a threshold value. Different values for this parameter were used we decided to 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 was computed. Every point in the curve is the average error from 30 bootstrap samples for a given training set size. This means that the 10 points that appear in every curve are the result of fitting 300 models. The first three points, corresponding to sample sizes of 64, 128, and 256 records, were not used for fitting the curve because the inverse power functional form does not fit well for small sizes. The same criterion was applied to all the techniques for consistency.

[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.

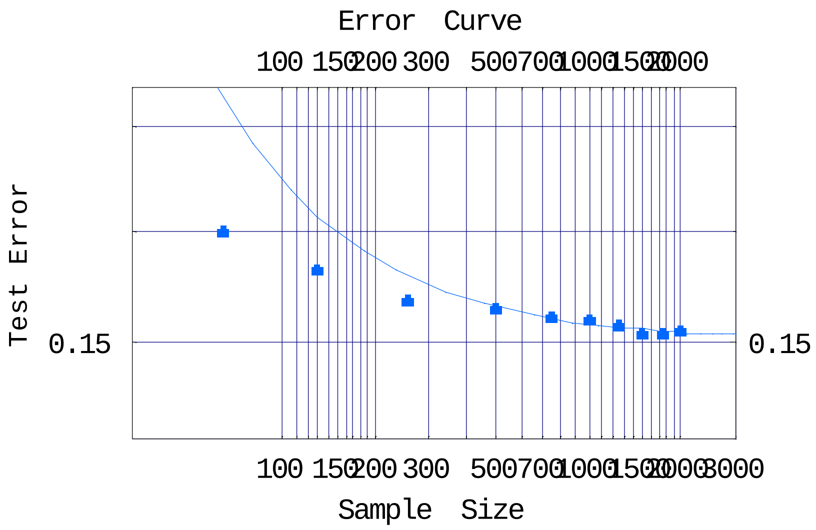


Figure 5. Generalization error curve as a function of sample size for Probit

As can be seen in Figure 5 the average error rate for the probit models starts around 16%, gradually declines to 15% and for large sample sizes it almost converges to its asymptotic value. This is confirmed from the results of fitting the inverse power law model (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 around 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 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 has reached its predictive capacity and the use of additional training records will have a small 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 error bar of the test error rate is in parenthesis.

Model	Test Error at m=2,000	Noise/Bias *a*	Complexity *b*	Optimum training sample size[recs]
Probit	15.13% (0.0047)	0.15025	1.80	1,804

Our interpretation of the relatively small value of the complexity parameter b = 1.8 is that the probit model has limited capacity to "see" all the complexity in the data. This explains why it doesn't take too many records, as in the case of 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 in the error rate for 0's (non-default group) and the 1's (the default group). The model identifies better the non-defaulting than the defaulting group and as a consequence the error rate for 0's is only 6.10% while for 1's is 25.20%.

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		

3.3 Decision-Tree CART model.

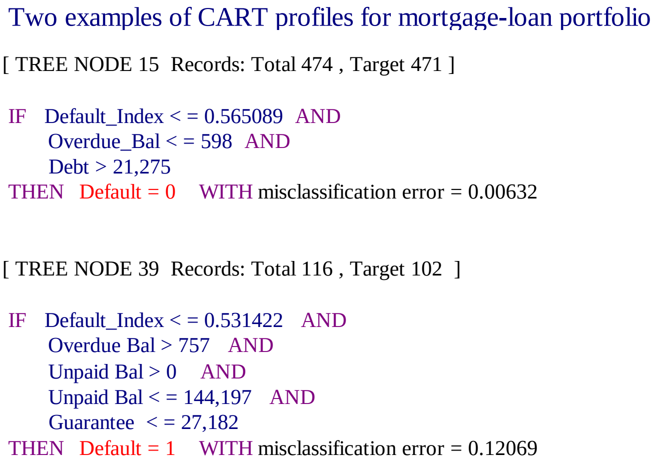
CART (*Classification And Regression Trees*) (Breiman *et al* [1984]) are powerful non-parametric models that produce accurate predictions and easily-interpretable rules to characterize them. They are good representatives of the decision-tree rule-based class of algorithms. Other members of these family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

We made a preliminary study of the effect of changing different parameters (Table V and Table VI). We controlled the size of CART trees by changing the "*density*" parameter (a feature supported by the toolset). This parameter represents the minimum percentage of records of any class that is required to continue the splitting at any 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 was 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 dataset. This optimal subtree is obtaining by a tree *pruning* process where 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 decision tree this pruning process is an example of a practical method for model size or capacity control (Vapnik [1995]).

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



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.

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 that 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 our intuition, the rule is not trivial. A person with a more or less reasonable payment history and with a particularly low overdue balance, and with still some debt to cover, is likely to pay. The second rule identifies a group that defaults and as in the previous rule, the default index is rather low but the overdue balance threshold is higher (757), the unpaid balance is positive but could be considerably high (144,197) and the guarantee value is small (27,182) relatively 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 that recently stopped paying but, more importantly, somebody who has a low incentive to pay due to the low value of the guarantee. As one can see the individual 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 used 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 nodes. For each size 10 different training set sample sizes were used: 64, 128, 256, 500, 750, 1000, 1250, 1500, 1750, and 2,000 records. For each sample size 30 bootstrap averages were made. In total 2,400 tree models were used for the analysis.

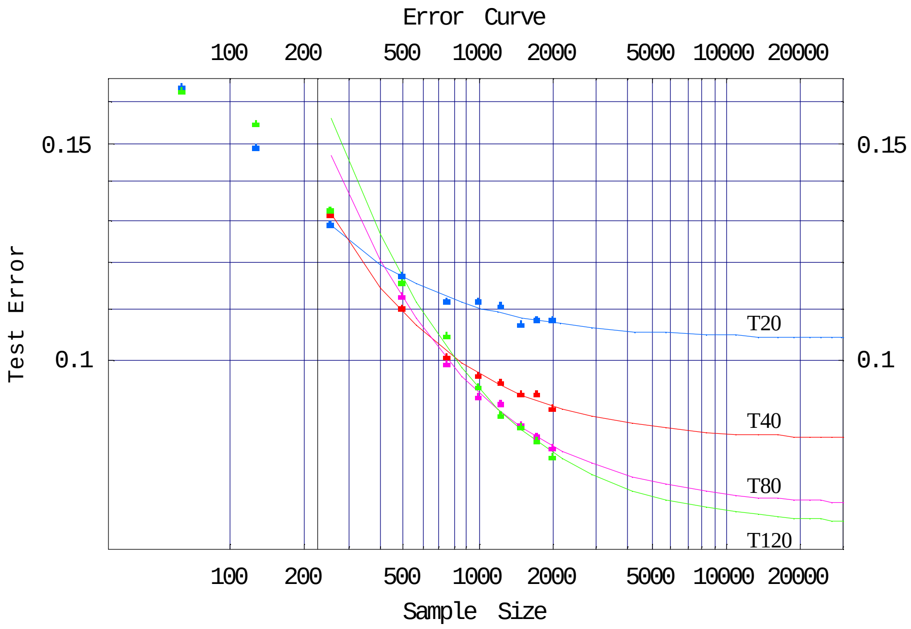


Figure 7. Learning curves for CART trees with 20,40,80, and 120 nodes.

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 ($E_{test} = a + b / m$). As can be seen, the fitted curve does not fit the small sample sizes and we decided to leave 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 that has the highest values for the complexity estimate. The larger the complexity the more records will be needed to attain convergence to the asymptotic value. The best model (120 nodes) attains and 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%.

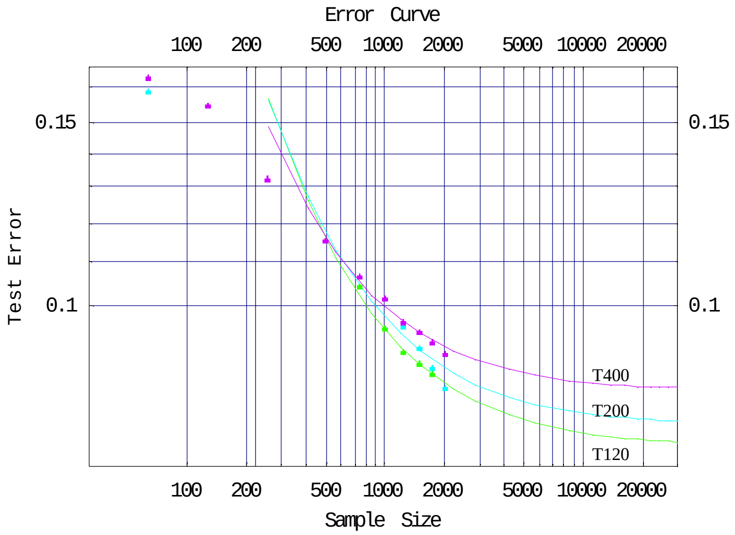


Figure 8. Generalization error curves for trees with 120, 200, and 400 nodes. Here we observe that the generalization error increases for excess model size (capacity).

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 α + 0.001.

Size # of nodes	Test Error at m=2,000	Noise/Bias *a*	Complexity *b*	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

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 identification of the defaulting group. It achieves an error rate of 6.29% compared to an error rate of 11.99% on 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 dataset of 1,000 which remains the same for all the modeling methods.

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

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 very 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 when the number of records increases. 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 will take place for an optimal training dataset size of about 21,675 records.

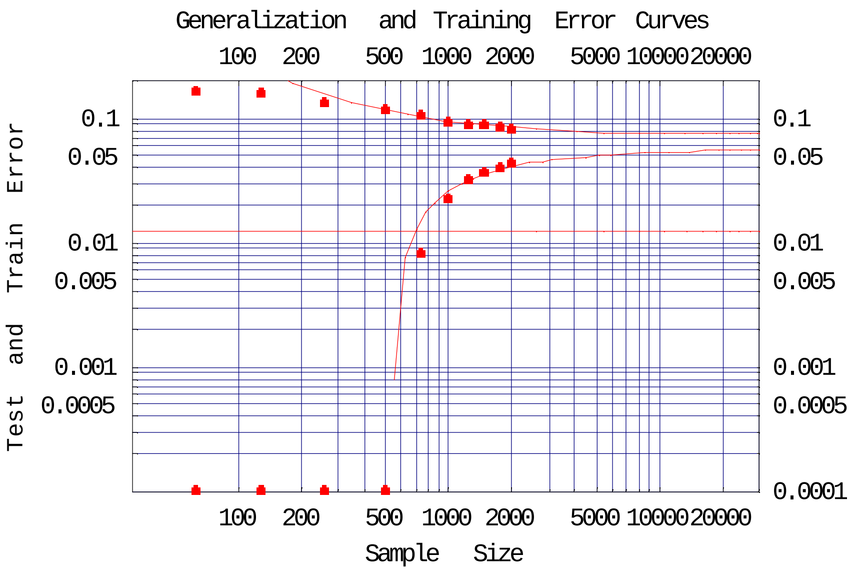


Figure 9. Generalization and training learning curves for the best CART tree model .

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

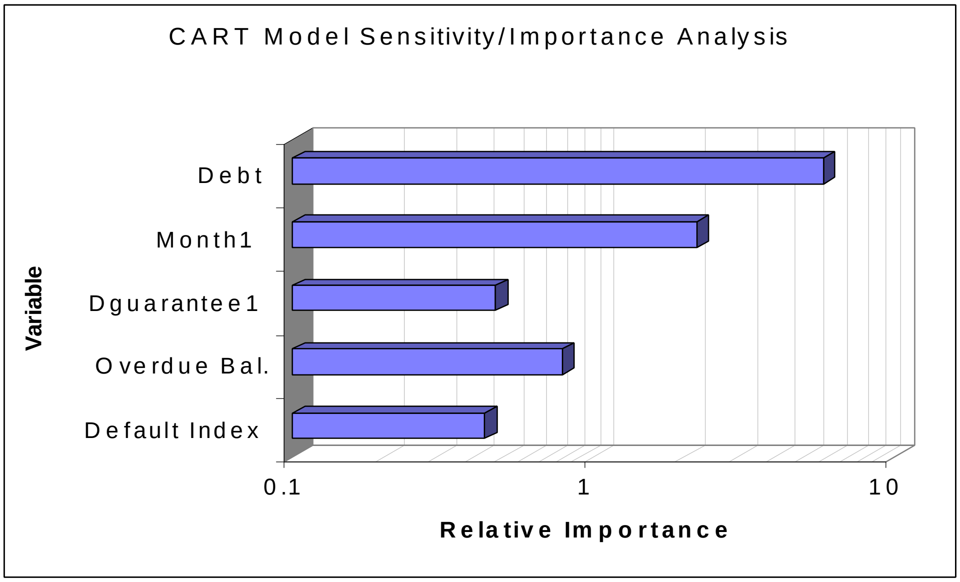


Figure 10 Relative sensitivity/importance for CART.

The graph shows the results of the variables sensitivity/importance analysis[14] for our best CART model. The 5 variables shown are the ones that made the greatest contribution to the model predictions. Perhaps not surprisingly these variables appeared predominantly in the actual CART rules.

3.4 Neural Networks.

We choose to use the standard feedforward neural network architecture (see for example Hassoun [1995]; White [1992]) supported by the Darwin toolset (see Appendix A) and experimented with several training algorithms: *backpropagation, steepest descent, conjugate gradient, modified Newton, and genetic algorithm.* Second order methods such as conjugate gradient allows for much faster training than the standard back-propagation. We also investigated 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 error surface which might be hard for gradient based methods. In addition to manual training we used the *train and test* mode provided by the toolset which is a useful feature to prevent 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.

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%

[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.

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%

After this preliminary network analysis we decided to take the best performer combination, sigmoid for the activation function in the hidden layer and conjugate gradient for the training algorithm, for the rest of the analysis. The relatively poor performance with genetic algorithms is probably due to the fact that we ran them only for a relatively small number of iterations. The analysis of learning curves was done in a similar way to the one described in the previous section for the decision tree models. A total of 4,200 neural network models were used for this part of the analysis. The results are shown in Table IX, Table X, and Table XI. Two approaches were used for the network selection, first we explored different architectures (number of nodes) while holding the number of iterations constant. The number of nodes in the hidden layer was changed from 2 to 16. The best results were 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) remained 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 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 *a*	Complexity *b*	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 explored the effect of changing the number of iterations while keeping the architecture constant. Table X and Figure 11 show the results for the same architecture (8 nodes) but different number of training iterations (10, 40, 80, and 100). Changing the number of iterations had the effect of changing the effective capacity of the neural network (Wang [1994]).

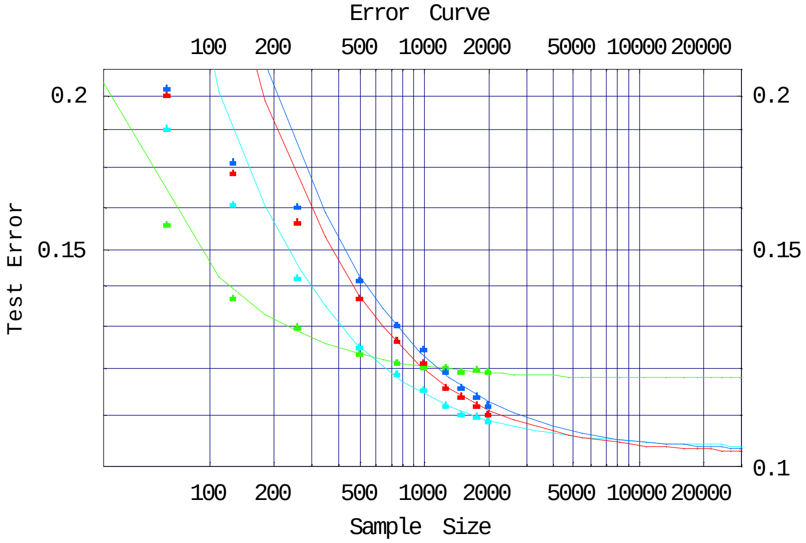


Figure 11. Generalization learning curves for 8-node neural nets with 10, 40, 80, and 100 iterations.

The curve with 10 iterations had the highest average error rate (11.92%) at 2,000 records and it also achieved the highest asymptotic error value. The curve with 40 iterations had the second highest convergence point in the graph (10.03%). If we only looked at the error rate achieved at the largest sample size of 2,000 records, this neural net would appear to be the one with 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 could have been tempted to choose 40 iterations as the optimal size. In actuality 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 seen by the second lowest asymptotic error value. We can also see a decreasing asymptotic noise/bias estimated parameter up to the neural net with 80 nodes. For larger nets this parameter increases.

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

Iterations	Test Error at m=2,000	Noise/Bias *a*	Complexity *b*	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

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

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

Iterations	Test Error at m=2,000	Noise/Bias *a*	Complexity *b*	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

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 and the table below shows the performance matrix for this net. It is interesting to notice that it shows the same type of asymmetry than the probit model: a lower error rate to identify the non-default group (11.18%) than for identifying the default group (19.49%).

Table XII. Neural net with 8 nodes and 80 iterations.

	Actual vs predicted Matrix (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		

3.5 *K*-Nearest Neighbors.

K-nearest neighbors (*k*-NN) is an algorithm somewhat different from the others in the sense that the data itself provides the "model." To predict a new record it finds the nearest neighbors by computing the Euclidean distance and then performing a weighted average or majority vote to obtain the final prediction. It works well for cases of relative low dimensionality with complicated decision boundaries. The toolset we used (see Appendix A) also supports the capability to "train" global attribute weights in such way that they have optimal values in terms of maximizing the prediction accuracy of the algorithm. To train the weights one uses a small additional dataset of a few hundred records (250). This modification tends to improve the results compared with the standard *k*-NN but the algorithm still retains its main characteristics. In practice *k*-NN works somewhat better than expected and this may be due to the not too adverse effect of its high-bias as has been suggested by Friedman [1997].

We performed a learning curve analysis similar to the one described for the other algorithms. The results are shown in Table XIII and Figure 12. In this case the train error is not reported because it is always zero as the "model" is the data itself. One practical problem of applying *k*-NN to our mortgage-loan dataset is the fact that the amount of records is quite small. In high dimensionality datasets one may require significant amounts of data to overcome the "curse of dimensionality" (Friedman [1997]). The fact that as the amount of data increases the model (dataset) increases too makes impossible to fit fixed capacity learning curve models as is done for the other models. It is possible to take into account the change in model size in the model fitting but we did not attempt to do it for the dataset used in this study. A simple extrapolation by inspection indicates that much more than 20,000 records will be needed to make this algorithm produce error rates comparable to CART or the neural net.

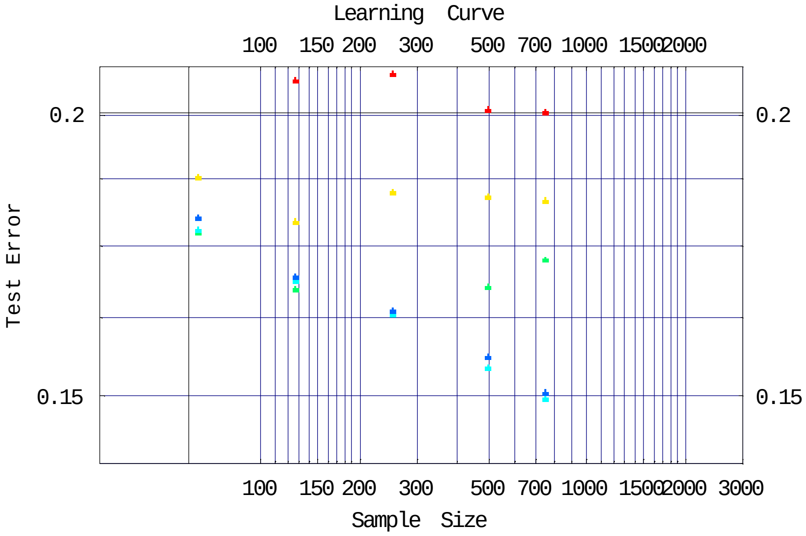


Figure 12. Error behavior for *k*-NN models.

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)

As can be seen in the Figure and Table XIII the optimal number of neighbors *k* appears to be around 24. The model attains a test error rate of 14.95% which is significantly higher than the neural networks or CART rates but comparable to the probit results. We believe this is produced by the relatively small size of the model dataset. The performance matrix for *k* =24 shows the same pattern than the probit and the neural network model. It has the same type of asymmetry since it has a lower error rate to identify the non-default group (12.40%) than for the default group (22.83%).

Table XIV. Performance matrix for *k* =24

	Actual vs predicted Matrix				
	Predicted				
	0	1	Total	Total Error	17.70%
0	431	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 shows a summary of the best models' performance (error rates, complexity and optimal sample sizes). The best model overall is a decision tree of 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) which means that even if larger amounts of data were available this is the limit of prediction accuracy that could be attained with this type of model. The fact that this value is the lowest for all the algorithms also suggests that the intrinsic noise in the dataset might be close to this value. This will be the *limit on accuracy imposed by data quality* as described by Cortes *et al* [1994a]. In addition of having the smallest noise/bias parameter, the complexity of this model is significantly higher confirming the hypothesis that the best model exploits the data in a better way and converges more slowly to its asymptotic value. Based on the fitted model we anticipate that it will take at least 22,000 records to achieve optimal results with CART decision trees. This is the number of records that one will consider in order to build a production-quality predictive risk model for this particular financial institution.

In second place we find 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 by the model is 10.2% (noise/bias = 0.102) gives the limit of prediction accuracy that can be attained with this type of model. We speculate that the difference of about 3% with the best tree results is probably due to the bias in the network model and the fact that the optimal net training point (global minimum) was perhaps not attained in our training. The complexity parameter of 18.17 is less but not too far from the CART model. We conclude that a sample of at least 18,165 records will be needed to attain optimal results with this model.

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

Model	Test Error (2,000 recs.)	Noise/Bias *a*	Complexity *b*	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

The best *k*-NN using 24 neighbors attains 14.95% test error (average). The reason for this higher error compared with the other algorithms is very likely produced by the small size of the "model" dataset. The dimensionality of the dataset is relatively high and this means that large amounts of records might be needed to obtain better results. A simple extrapolation by inspection indicates that much more than 20,000 records will be needed to make this algorithm produce error rates comparable to CART or the neural net. The conclusion is that this algorithm is a viable alternative if one could obtain enough data records. Finally the probit model attained an average test error of 15.13%. Even though this method was the worse in terms of 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 we can see in Figure , for up to 128 records it outperforms the other methods and competes well with the decision-tree.

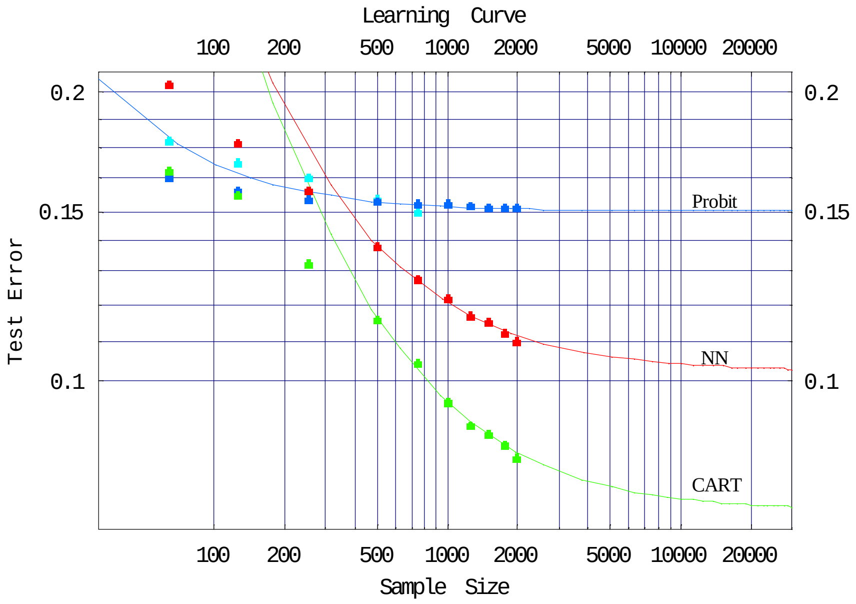


Figure 13. Comparison of results for 4 algorithms (Probit, CART, Neural Nets and *k*-NN).

We find that the use of learning curves and noise/bias and complexity parameters offers an interesting perspective to understand the nature and characteristics of different algorithms or data fitting methods. Unfortunately we don't have available at present other datasets of similar financial institutions to make a comparative study. In such a study one will compare the parameters of the models, and in the case of CART the profiles themselves, to be able to assess degrees of similarity.

As a complementary note we would like to mention that this type of analysis applied to a U.S. 1994 Census dataset (UCI repository[15]), where the problem considered is the prediction of high and low income, produced values of 0.141 and 49.0 for the noise/bias and complexity respectively. This suggests that our home mortgage dataset/problem is less noisy but also less complex than that of the Census dataset. It is interesting to notice the similar structure of the performance matrices for the Probit, Neural Network and *k*-NN models where the errors are higher for discriminating the default group. The one for the CART model is different and this might be one of the reasons this algorithm outperforms the others. Perhaps this asymmetry can be exploited by combining different algorithms and in this way improve the predictions. In Table XVI, we show the results of an exploratory combination of models' prediction by the use of logical operators (i.e. AND and OR). This simple combination method already shows some potential to improve the individual models' results. For exaple the best combination to predict the non-defaulting gruop is given by combining (AND) CART and Probit (3.25%). We especulate that this effect may come from model bias reduction and the nature of the confusion matrix assymetries. The best prediction for the defauling group is attained by combining (OR) CART and Neural Net (4.72). The overall absolute error do not decrease below the CART error.

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

The next step in this investigation of model combination will 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 the global errorr by effective reduction of the variance and bias of the combined model.

[15] http://www.ics.uci.edu/~mlearn/MLRepository.html.

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 comment on possible uses of the models' prediction outputs in this regard.

4.1 Aggregation of risk for one institution.

Credit risk.- Early warning systems introduced in the 1970's and 1980's were mostly phenomenological in the sense of attempting to describe the phenomenon (failure/non-failure) by making a coarse-grained model consisting of one single modeling stage without explicit decomposition of risk. Here we are interested in applying a more fine grained analysis based on the previous calculations of default risk for individual borrowers and then aggregating for the entire portfolio.

One simple way to aggregate the default risk of the credit portfolio is to multiply the probability of default times the amount of *capital at risk* for each loan, and then summing up for 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. This in turn could 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 the case of our portfolio we would need: first, to generate scenarios of default or non-default for the individuals in the portfolio accordingly to the predicted probabilities that result from the predictive models; second, decide the recovery rate in the state of default (this step is important since there is a lot of uncertainty about the recovery rate in the state of default); third, aggregate the individual scenario to come up with one instance of the future value of the portfolio; fourth, repeat many times to generate the distribution of the portfolio.

Other types of risk.- This study focused particularly on credit risk for mortgage-loans but the same methodology could be applied to other credit portfolios (e.g. credit cards, personal and commercial loans) or to analyze other risk factors such as prepayment risk. As a result of the analysis one can identify subsets of the portfolio that may be subject to unbundled or packaged into new financial instruments with particular type of risks. These in turn could be sell to investors most willing to buy take these risks.

Consider trading portfolios where 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 want to be analyzed (e.g. interest and exchange rates, stock and commodity prices,

[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

option volatilities, GDP growth, price indexes, etc.) An important difference is that the classification techniques must be substituted by regression methods of the algorithms. To illustrate this imagine we want to measure the value at risk of a given portfolio[17]. As mention before, the first step is to decide on the *N* systematic risk factors $X_1, X_2, \ldots, X_N$ we want to consider. Then one decomposes 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 with linear regression analysis by estimation of the following model:

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

If we are interested in a non-parametric representation of this specification we estimate the following form,

$$R_j = f\,(E\,(R_j), X_1, X_2, \ldots, X_N) + u_{j,} \qquad (2)$$

and then perform sensitivity analysis to obtain the relative impact of movements on each of the factors while keeping the rest constant. 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,

$$AE_i = \sum_{j=1}^{M} V_j b_{ij} \qquad j=1,\ldots,N. \qquad (3)$$

To get the value at risk one expresses the return per dollar of the entire portfolio (R_P) in the following form,

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

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

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

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

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

Other applications of model's output.- Another application for corporate policy involves the use of profiles (rules) as provided by the decision-tree (see Figure 6). For instance, institutions may instrument a policy where the riskiest group is suject to a special process that reinforces the collection of the loan. An alternative policy might give some benefits to borrowers in a way that motivates them to pay. These policies must be designed to always incentive borrowers to pay their obligations. It is counter beneficial to implement policies that give the wrong incentives, this only

[17] This methodology is similar to R. Merton notes for the MFI course in the HBS.

aggravates the default problem. In general incentive compatibility penalizes bad performance and rewards good performance.

Other possibility for using classification or predictive models is in the area of fine-grained segmentation for customer groups. This is typically done in the context of direct marketing (see for example Bigus [1996], Bourgoin [1994], Bourgoin and Smith [1995]). These segmentation is done not only along risk parameters but taking into account payment modalities, revenue/ROI (Bourgoin and Smith [1995]), or customer equity group information (Blattberg and Deighton [1996]). This analysis is especially relevant for corporate profitability or targeted marketing applications.

4.2 Aggregation of risk in global financial system models.

Regulatory authorities might require to gather information from the entire financial system in order to develop a global model for the system. At first, the analysis could be done separately for some representative institutions to look for differences and similarities between the models (e.g. error rates, noise/bias, and complexity.) The rules from decision-tree models and the sensitivity/importance of the variables can also be subject to comparison. If it turns out that the different models have common properties then some generalizations for the system (the universe in question) could be established.

We can imagine the risk consolidation of the system as a whole, this may require a lot of information and calculation (the problem of scale.) This approach requires to built models of risk for each institution and then agglutinate them according to equations 1-6 and then summing up for 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 the three largest banks holding more than 58% of the mortgage loan market (as of June 1996). On the other hand if we consider a very diluted market, as is the case of the United States, the number of institutions will be in the order of thousands.

Another less demanding computational and informational approach is to take a representative sample of the assets and liabilities of the system and then calculate the global risk from this sample. Then one calculates the exposure of this sample to estimate the global risk of the system. This approach requires less information and it might loose resolution but it is easy to manipulate and compute.

The regulatory authority can also use the model output similarly as the institutions might use it for corporate strategy. For example, after the Mexican crisis of 1994-95 the government implemented financial aid programs targeted to the borrowers of different loan types such as private, business, mortgage, and credit cards. These programs tried to lessen the burden of interest accumulation while at the same time keeping the incentives of the borrowers aligned to fulfill their payments. This type of policies can benefit from more accurate classification of the groups. In this way the overall impact of the policies can be measured more precisely.

5. Conclusions

We found that a combination of different strategies and the application of a systematic model building and selection methodology offer an interesting perspective to understand 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 analyzed the performance of four algorithms for a mortgage loan dataset and determined that decision trees produced the most accurate models. We analyze the different ways in which these institutional models can be combined to provide global models of risk. The next step in this line of research is to extend the analysis for other risk factors and other institutions to make a comparative study.

6. Acknowledgments

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 datasets 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.)

7. References

Adrians, 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* Vol. 6, pp. 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.

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. Vol. 81 (May/June). Pp. 637-654.

Blattberg, R. C. and Deighton, J. 1996. Manage Marketing by the Customer Equity Test, Harvard Business Review, July-Augulst 1996

Breeden, D. T. 1979 “An Intertempora Asset Pricing Model With Stochastic Consumption and Investment Opportunities, “ Journal of Financial Economics, 7 (September). pp265-96. Reprinted in Bhattacharya and Constantinides, eds (1989).

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

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 Artificial Intelligence in the Capital Markets, edited by Freedman, Klein, and Lederman, Probus Publishing.

Cortes, C., Jackel, L. D., Chiang, 1994a. W-P Limits on Learning Machine Accuracy Imposed by Data Quality, Advances in Neural Networks Processing Systems, G. Tesauro, D. S. Touretzky and T. K. Leen Eds. MIT Press. Vol. 7, p239.

Cortes, C., Jackel, L. D., Solla, S. A., Vapnik, V., 1994b. Learning Curves: Asymptotic Values and Rate of Convergence, Advances in Neural Networks Processing Systems, G. Tesauro, D. S. Touretzky and T. K. Leen Eds. MIT Press. Vol. 6, p327.

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

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 1996

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 1996

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

Fisher, R. 1950 A. Statistical Methods for Research Workers, 11 ed.

Friedman, J.H., Bentley, J.L. and Finkel, R.A. 1977. An algorithm for finding best matches in logarithmic expected time, ACM Trans. math. Software 3, 09-226.

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. Europena Conference, EuroCOLT'95,*p23-27. 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, Chichester, John Wiley.

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

Horst, R. and Pardalos, P.M., Eds. 1995. Handbook of Global Optimization, Kluwer.

Hume, D., 1739. An Enquiry Concerning Human Understanding, Prometheus Books, Pub. 1988.

Hutchinson, J. M., Lo A. W., and Poggio, T 1994 A non-parametric Approach to Pricing and HedgingDerivative Securities Via Learning Networks *The Journal of Finance* Vol. XLIX, No. 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., Vazirani U. V. 1994. An Introduction to Computational Learning Theory, Cambridge, Mass. MIT Press.

Keuzenkamp, H. A, and McAleer, M. 1995. Simplicity, Scientific Inference and Econometric Modelling,*The Economic Journal*, 105, p1-21.

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. New York, Hafner press.

Landy, A., 1996. An Scalable Approach to Data Mining, Informix Tech Notes, vol. 6, issue 3, p51.

Li, M and Vitanyi, P. 1997. An Introduction to Kolmogorov Complexity and Its Applications, 2nd. Ed. 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. New York, John Wiley.

Meyer, P. A. and Pifer, H. W. 1970, " Prediction of Bank Failures," Journal of Finance 25, No. 4, 853-868.

Merton, R. C., 1973. "Theory of Rational Option Pricing," Bell Journal of Economics and Management Science, Vol. 4 (Spring), pp. 141-183

Merton, R. C., 1973. "An Intertemporal Capital Asset Pricing Model," Econometrica, 41 (September). pp 867-87. Reprinted in Continuous Time Finance. 1990. Cambridge, MA. Basil Blackwell as Chapter 15.

Merton, R. C., 1997 Class notes for the Management of Financial Intermediaries course at Harvard Business School.

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, *Phys. Rev. Lett.* 75, 3772.

Piatetsky-Shapiro, G. and Frawley, 1991. W. J. Eds. Knowledge Discovery in Databases. MIT Press.

Pindyck, R. S. and Rubinfield, D. L. 1981 Econometric Models and Economic Forecasts. Mc. Graw Hill. 2nd Edition.

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, Vol. 9 (January), pp 277-293.

Sinkey, J. F., Jr. 1975 “ A Multivariate Statistical Analysis on the Characteristics of Problem Banks,”Journal of Finance 30, No.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/~kdd/

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 (1986).

Seung, H. S., Sompolinsky, H. and Tishby N. 1993. Statistical Mechanics of Learning from Examples. *Physical Review A*, vol. 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. Advances in Neural Networks Processing Systems, G. Tesauro, D. S. Touretzky and T. K. Leen Eds. MIT Press. Vol. 7, p239.

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. San Mateo CA, Morgan Kaufmann.

White, H. 1992 Artificial Neural Networks, Blackwell, Cambridge, MA.

Valiant, L. G. A Theory of the Learnable, *Comm. of the ACM* 27, 1134.

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

08. Appendix A: brief summary of software and tool sets 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 a integrated environment for numerical computations, algebraic computations, mathematical functions, graphics, and optimization algorithms. For more information see www.wolfram.com