

Data Mining in Finance:
Report From the Post-NNCM-96 Workshop on
Teaching Computer Intensive Methods for
Financial Modeling and Data Analysis

Andreas S. Weigend
Leonard N. Stern School of Business
New York University
aweigend@stern.nyu.edu

October 1997

Working Paper Series
Stern #IS-97-19

Published in: *Decision Technologies for Financial Engineering: Proceedings of the Fourth International Conference on Neural Networks in the Capital Markets (NNCM-96)*. Edited by A.S.Weigend, Y.S.Abu-Mostafa, and A.-P.N.Refenes. Singapore: World Scientific, 1997.
<http://www.stern.nyu.edu/~aweigend/Research/Papers/Weigend.NNKM96.ps>

**DATA MINING IN FINANCE:
REPORT FROM THE POST-NNCM-96 WORKSHOP ON
TEACHING COMPUTER INTENSIVE METHODS FOR
FINANCIAL MODELING AND DATA ANALYSIS**

ANDREAS S. WEIGEND

Department of Information Systems

Leonard N. Stern School of Business, New York University

44 West Fourth Street, MEC 9-74, New York, NY 10012

aweigend@stern.nyu.edu

www.stern.nyu.edu/~aweigend

The conference *Neural Networks in the Capital Markets* was followed by a workshop at Caltech on November 23, 1996. The purpose of the workshop was to bring people together to share their experiences and ideas on designing and teaching new courses in financial engineering, bridging the gap between computationally advanced methods for learning from data, and real world problems in finance. This report highlights some of these ideas, and describes in detail one specific course taught at the Leonard N. Stern School of Business at New York University. That course, *Data Mining in Finance*, focuses on computer intensive data analysis methods that are relevant to financial problems. The methods originate in fields such as statistics and probability, artificial intelligence and machine learning, computational science, applied mathematics, and engineering. They include neural nets, genetic algorithms, bootstrap methods, Bayes nets, Hidden Markov models, and clustering techniques. These methods are applied to problems that include building and evaluating technical trading strategies, and computing and managing market and credit risk. Beside the theory and the lectures, that course has two hands-on components. For each method, a computer assignment on a real world data set is placed between two lectures. The first lecture introduces the problem, discusses the essentials of the method to solve this problem, and demonstrates a software implementation. After students have gained some experience with the method and software in the lab, the second lecture summarizes strengths and weaknesses, and discusses further applications in the capital markets. The second hands-on component is a group project. It explores one of the approaches presented during the semester on a problem and data set selected by each individual group, usually in collaboration with a Wall Street firm. The course concludes with several Wall Street traders giving their perspectives on the practice and promise on data mining in finance.

In recent years the application of data mining—statistical artificial intelligence, machine learning, information extraction, knowledge discovery—to technical trading and finance has seen exciting results, as witnessed by several conferences, including NNCM (Neural Networks in the Capital Markets) in 1993 and 1995 (Refenes *et al.*, 1996) in London, in 1994 and 1996 (Weigend *et al.*,

1997) in Pasadena, and CIFer (Computational Intelligence in Financial Engineering) in New York City since 1995 (CIFer, 1997). In addition to ideas originating in academia, Wall Street firms have also contributed methods that are emerging as industry benchmarks for risk transparency and management (RiskMetrics, 1997; CreditMetrics, 1997)

The successful application of these new methods in the financial services industries require analytical maturity, mathematical skills, and statistical sophistication. The organizers of NNCM-96 felt that the course offerings in areas of financial data mining, computational finance, and financial engineering were lagging behind academic research and Wall Street practice. To provide a forum for ideas on how the teaching in this area can be improved, I organized a post-NNCM-96 workshop at Caltech on November 23, 1996. The participants, most of them teaching at universities, came from Europe and the United States.

It is impossible to list all the rich and complex ideas, experiences and questions that were brought up. To give some flavor to what is possible in a business school environment, this report describes one specific realization of such a course that I have developed at New York University's business school.

The course focuses on data mining techniques that are relevant for finance and business. It introduces state-of-the-art approaches and results from the fields of financial engineering and computational finance, and provides the relevant background from other disciplines such as statistics, machine learning, and traditional finance.

The **objectives of the course** include:

- relate and integrate often disparately experienced pieces of knowledge that the students have been exposed to in earlier courses;
- show the strengths and weaknesses of each of the methods on financial data;
- provide a hands-on exploration for every method, and demonstrate industrial-strength software packages when available;
- deepen the knowledge and understanding of one of the techniques through a group project in conjunction with one major financial firm,^a

^aIdeally, these group projects form around a problem and data set one of the members of the group, typically a part-time MBA's, brings from his or her workplace. Group projects also emerge from consulting and interactions with affiliate firms to the program. In all cases, a tight coupling is important between the group of students and the firm providing problem definition and data.

- compare the promise of the new methods with the practice on Wall Street through some lectures and in-class discussions with several traders from key financial institutions.

The **teaching strategy** is based on the fact that hands-on experiences greatly facilitate the mastering of theoretical concepts and also help the student retain significantly more than from lectures alone. Therefore, the course goes every week through the following lecture–lab–lecture sequence:

- the instructor introduces the problem to be solved, explains the key aspects of the new method in sufficient detail, and briefly demonstrates the implementation in the classroom;
- the students apply the method to a real world problem in the computer lab;
- the lab experiences are reviewed and the general principles extracted, and further examples of successes and failures of this method on financial problems are presented, again in the classroom.

The **target audience** consists of several student populations, reflecting the diversity and interest in this emerging field:

- *Master of Science in Statistics with Specialization in Financial Engineering* from the Stern School (NYU's business school),
- *Financial Engineering Track for MBAs*, also from Stern,
- *Masters of Science in Mathematics in Finance* from the Courant Institute (NYU's mathematics and computer science departments).

These programs are two-year programs taken by full time students. Both MS programs require a total of 12 courses. The course described here is usually taken by students in their last semester when they have a solid foundation not only in finance, but also in applied probability and statistics, including courses on time series and on multivariate regression. It is an elective for the other two programs. It can also be taken by any Ph.D. student at Stern.

Apart from selecting from the full spectrum of more traditional finance courses at Stern,^b there are several other courses that also include some hands-on experience and complement this data mining course. They include *Financial Information Systems* (taught by Bruce Weber, Information Systems Department) that uses derivatives analysis software, and *Statistical Computing with*

^bIn its latest survey of MBA programs (October 21, 1996), *Business Week* ranked Stern's Finance Department fourth in the nation.

Application to Finance (developed by Sean Chen, Statistics and Operations Research Department) that focuses on sampling methods for finance and includes a module on iterative sampling methods (Markov Chain Monte Carlos), and a module on Bayesian inference and decision making. Furthermore, the course *Knowledge Systems in Organizations* (developed by Vasant Dhar, Information Systems Department) gives an overview of knowledge discovery and data mining techniques as they are important for organizations, focusing on their impact on business and issues regarding their deployment.

The **high-level outline** of this course on data mining in finance is divided into four parts:

1. Introduction (3 weeks);
2. Building and Evaluating Trading Models (5 weeks);
3. Managing Risk (3 weeks);
4. Projects, Practice, and Promise (3 weeks).

<i>Wk</i>	<i>Topic</i>
1	Introduction: <i>i</i>) Computer environment, <i>ii</i>) Statistics background
2	Explorative data analysis (Visualization, Sonification, Wavelets)
3	Model evaluation and testing (Bootstrap, Simulation)
4	Time series prediction using nonlinear neural networks
5	Trading strategy development using neuro-fuzzy modeling
6	Trading rule discovery using genetic algorithms
7	Nonlinear regime switching models (Hidden Markov models)
8	GARCH and stochastic volatility (State space models)
9	Market risk (Value-at-Risk, <i>RiskMetrics</i>)
10	Credit risk (Nonlinear logistic regression, <i>CreditMetrics</i>)
11	Transaction risk (Bayes nets and graphical models)
12	Student projects (groups of five in collaboration with firms), and
13	Presentations by Wall Street traders and discussion, and
14	Special topics (pairs trading, execution, ICA, new products, ...)

Table 1: Week-by-week schedule for *Data Mining in Finance*. The schedule for the group projects is given in Part 4 the main text.

The table lists the contents on a week-by-week basis, and each week is described in detail throughout the remainder of this document.

1 Part 1: Introduction—Understanding through Visualization and Simulation

1.1 Tutorials: Managing Backgrounds and Expectations

The first class (of tutorial nature) takes place in the lab and goes through the mechanics of computer use, ranging from the specific setup in the computer labs and the access of the data sets for the assignments, to simple examples of the two main packages used in this course, S-Plus and Matlab.^c In the first week, the students fill out a questionnaire about their expectations, technical backgrounds, and interests. Answers will help the instructor identify problems early and optimally adjust the level of the course. For the group project, students are asked whether they have any specific problems in mind that they would be interested working on. Furthermore, part-time MBA students, usually working in related fields, often have access to specific data sets from their workplace. The learning experience of the group project draws on these real questions and real data from some of their members. Integrating these real world projects and presenting the final results both at Stern and in the firms is an important ingredient in the last part of this course.

The second class (also of tutorial nature) reviews some concepts from probability and statistics that the students are expected to be familiar with.^d This class also makes the expected background knowledge explicit and points to the pertinent literature that can help refresh some potentially rusty knowledge. At the end of the second class, groups of five students are formed. Care is taken to have in each group at least one student strong in theory, and one student strong in practical computer issues. Furthermore, students who already have a specific problem they want to model are distributed evenly across the groups.

The next two weeks continue with introductory character. Each week is dedicated to one of the two main software packages used in the course. The general approaches and specific commands that will be useful throughout the course are explained in conjunction with some interesting real world data.

^cAn argument can be made that the education should be streamlined into one single software package that the students have encountered in prior classes, such as Minitab. However, in a course covering such a wide terrain such as this one, it is important to draw on the advantages of different packages. Furthermore, familiarity with several modeling languages is an important asset.

^dThese concepts include random variables, distributions, the correlation coefficient and its meaning, linear regression and the error on parameters, the maximum likelihood framework, conditional probabilities, conditional expectations, conditional variances, the Expectation Maximization algorithm.

1.2 Exploring Data Through Visualization and Sonification

The second week uses a cross-sectional data set of equity returns, firm size, and book-to-market value.^e Fama & French (1992) argued that size and book-to-market play a dominant role in explaining cross-sectional differences in expected returns. Re-analyzing their data with Trellis graphics (a method of conditional histograms) uncovers some striking features, including the complete disappearance of the risk premium on size (Knez & Ready, 1997). The background reading is an introduction to S-Plus, and the lab consists of reproducing the Trellis plots and further exploring the data that could lead to an understanding of the economic forces underlying the size effect. Besides gaining some initial familiarity with S-Plus, the students learn to appreciate the power and importance of exploratory data analysis and visualization. If time remains, there will also be a demonstration of various ways of revealing structure and information through rendering time series data in the auditory domain (sonification), as well as examples of multi-scale analysis with wavelets, two examples of modern exploratory data analysis.

1.3 Understanding Results Through Simulation (Monte Carlo and Bootstrap)

The third week introduces bootstrapping, a computer intensive method to generate distribution of variables of interest, such as the profit of a certain trading strategy. The lab uses both computer generated examples and real world data to gain an appreciation of the inherent uncertainty of results in noisy environments. The synthetic data is generated to show just how easy it is to fool oneself about the profitability of a trading strategy. The reading includes chapters on model evaluation and testing procedures of a forthcoming book by Blake LeBaron (Economics Department, University of Wisconsin, Madison). The students obtain a deeper understanding of the stochastic nature of financial markets and experience the central importance of absolutely clean methodologies for model testing, in particular the use of true out-of-sample data in any performance evaluation. This assignment is carried out in Matlab.

2 Part 2: Building and Evaluating Prediction and Trading Models

Making predictions and building trading models are central goals for financial institutions. They were the earliest areas of the application of modern machine

^eThe data include all NYSE, AMEX and Nasdaq non-financial firms in the monthly CRSP tapes for the period July 1963 through December 1990.

learning techniques to real world problems. In this course, we now turn to estimating the parameters of a model. In contrast, the methods in the first part assumed that a model already existed, and that the sole task was their evaluation.

Consisting of five weeks, this second part of the course is the longest individual part. The readings are taken from several sources, including *Neural Networks in Financial Engineering* (Refenes *et al.*, 1996), *Decision Technologies for Financial Engineering* (Weigend *et al.*, 1997), and the Proceedings of *Computational Intelligence for Financial Engineering* (CIFEr, 1997).

One important ingredient throughout this part is the methodology of turning predictions (e.g., the expected value of a future return and its variance) into actions (e.g., taking a position of a certain size). For students not familiar with statistical decision theory and the utility function framework, the concepts needed for this part of the course will be presented in an extra lecture of tutorial character at the beginning of this part.

Most of the hands-on components use either Matlab or S-Plus. When appropriate, comparisons to Excel are presented since most students are familiar with spreadsheets. Presenting two conceptually different approaches to one problem clarifies the advantages and disadvantages of the different computational paradigms.

The five weeks of this part range from neural networks and neuro-fuzzy models, over trading rule discovery using genetic algorithms, to nonlinear time series models with hidden states. The details of week 4 through week 8 are given below.

2.1 Nonlinear time series prediction

This week begins reviewing—from the perspective of data mining—the concepts the students had been exposed to in the prerequisite time series course, as well as presenting some implementations of linear time series models in S-Plus and Matlab. The central part of this week introduces “classical” neural networks for regression, corresponding to a nonlinear autoregressive model if the inputs consist of past values of the time series (tapped delay line). This framework is easily extended to incorporate inputs that are of various quantities which are derived from the time series itself (e.g., exponential moving averages, curvatures, volatility estimates), as well as additional exogenous time series (e.g., interest rates, indices, other markets, news). In the lab, the students use both linear methods and nonlinear neural networks on the same data sets, in order to understand the strengths and problems of the more flexible neural network approach. One example shows that nonlinear methods can indeed outperform

linear methods on financial forecasts, and also that the problem of overfitting can be very serious.

2.2 Neuro-fuzzy trading model

The second week on neural networks extends the framework to incorporate rules and relationships some traders believe in. The learning algorithm moves their parameters and adjust their weights. In addition to this neuro-fuzzy framework, the class presents various methods to combat overfitting, including pruning (removing parameters that model the noise rather than the signal), clearing (the combination of cleaning the data and learning the model), and adding adaptive noise to the inputs (to prevent the weights from learning more than there is). During this week, as an exception, the lab is replaced by a one-day workshop where a practitioner shows the entire process of building a successful trading model, using SENN (Simulation Environment for Neural Networks). The data set of daily data compiled from multiple sources is provided by Georg Zimmermann (Siemens AG, München). The students are to gain an appreciation of the different phases and iterative nature of the modeling process; however, they are not expected to reproduce the model in detail. Students who cannot attend the full-day workshop (held on a Saturday), complete an assignment on predicting conditional variances and conditional percentiles in addition to the usual conditional mean (in Matlab), serving as a preparation for risk management.

2.3 Trading rule discovery using genetic algorithms

This week introduces genetic algorithms as a search technique and shows how it can be used to solve problems such as volatility prediction, portfolio optimization, and stock picking. Often, in very complicated search spaces, the genetic algorithm can find interesting niches or relationships among variables. We show how genetic algorithms can be used to create rule-like models for prediction and classification. While these resulting models are less flexible than general neural networks, they are more amenable to interpretation. More importantly, genetic algorithms can produce many alternate models of the phenomenon which are similar in performance. The derived rules can be discussed with experts, and modified or discarded when they seem not to make any sense. The assignment uses futures data from an equity index to build models that forecast volatilities. The subsequent discussion shows how techniques influence the formulation of a problem and the representations chosen for the data and the model.

2.4 Time series with discrete hidden states

The time series and trading problems so far have all been framed as regression problems. While this has made the computation relatively easy, time specific information is ignored: reshuffling all the input-output pairs of training would not have changed the resulting model. In contrast, this and the following week exploit time series specific information by introducing hidden states that are non-local in time. This week's framework of hidden Markov models uses several discrete states with nonlinear neural networks as sub-models (also called experts or agents). We assume that we know the number of sub-models and their structure; the algorithm estimates the parameters of the sub-models, the probabilities of transition between them, and, for each time step, the probabilities to be in each state. The assignment first analyzes a toy problem where the hidden state is known from the way the data was generated. However, it is not used in building the model, but afterwards, in order to verify that the model indeed discovered the hidden states. On the real world data, the students discover that the regimes form according to market volatility, rather than according to other possible dynamic criteria they might have expected (such as trending vs. mean reverting). The assignment is based on an implementation in Matlab by Shanming Shi (J.P.Morgan, New York).

2.5 Time series with a continuous hidden state

State space models, the focus of this week, keep the assumption that the dynamics is hidden, but—in contrast to having several discrete hidden states with individual dynamics—they assume that the dynamics is captured by a continuous hidden state. State space models apply to time series processes with observational noise, i.e., noise that is not fed back into the systems but added during the observation.

The assignment uses the state space approach to model volatility,^f showing the striking advantage of this model class in the presence of observational noise over a standard autoregressive approach. The state space model is compared to stochastic volatility models and to generalized autoregressive heteroskedastic models, as implemented as S+GARCH. The data consists of eight years of

^fThe source for “observational noise” for volatility time series is not a mistake in recording the observation, as the term might imply, but the following: The model describes, as always, the evolution of the expected value. We, however, do not observe the expected value, but a realization drawn from a distribution with that expected value. For example, if the returns were Gaussian, their squares would be χ^2 -distributed with one degree of freedom. Such realizations are very noisy; their standard deviation is of the same size as the value itself! This is the source for the “observational noise” in volatilities.

high-frequency foreign exchange data (in ϑ -time), kindly provided by Michel Dacorogna (Olsen and Associates, Zürich).

3 Part 3: Statistical Modeling in Risk Management

Risk is the degree of uncertainty of future returns. Risk management has become a central concern for financial institutions, since derivatives have become a major component of the markets. The approaches to risk management are becoming increasingly sophisticated in their computational and statistical methodology.

Risk stems from a variety of sources and exposures. The first of the three weeks addresses *market risk*, the uncertainty of future earnings resulting from changes in market conditions. The second week addresses *credit risk*, the possible inability of a counterparty to meet its obligations. Both of these weeks draw heavily on methodology developed by J.P.Morgan, released in 1994 and 1997, respectively. The third week discusses *transaction risk*, using the example of identifying credit card fraud, one of the success stories of data mining techniques for financial problems.

During these three weeks, the students are also carrying out exploratory studies in preparation for their final group projects. This implies that the hands-on homework assignments require slightly less individual work than those in the first half of the semester.

3.1 Market risk

Market risk is the uncertainty of future earnings due to changes in market conditions, such as prices of the assets, exchange rates, interest rates, volatility, and market liquidity. The students are introduced to FourFifteen, a state-of-the-art tool, named after J.P.Morgan's internal market risk report, which was originally produced daily at 4:15 p.m. FourFifteen is a risk engine and a set of reporting tools. It is based on J.P.Morgan's RiskMetrics framework (RiskMetrics, 1997). Version 2 of FourFifteen is the result of J.P.Morgan teaming up with a software and interface company (The MathWorks Inc., the company that produces Matlab), and with a data company (Reuters).⁹ FourFifteen is an excellent didactic tool, allowing the students to compare the original RiskMetrics variance-covariance approach with historical value-at-risk and Monte Carlo value-at-risk computations. They learn to break down the

⁹Up-to-date data will be obtained from www.riskmetrics.reuters.com.

different contributions to market risk, and gain a deeper understanding of the implications of the choices that have to be made in modeling market risk.

3.2 Credit risk

The booming global economy in recent years has created a business environment which enticed many institutions to take more credit risk. The proliferation of complex financial instruments has created uncertain and market-sensitive exposures that are more difficult to manage than traditional instruments. Active credit risk management, that includes the generation of consistent risk-based credit limits and rational risk-based capital allocations, is currently among the hardest challenges in the risk area.

The course uses J.P.Morgan's CreditMetrics, a portfolio approach to credit risk analysis that treats all elements of the portfolio on an equal basis (CreditMetrics, 1997). This approach also considers the correlations of the changes in credit quality. It gives not only expected losses, but also their uncertainty (i.e., variance), expressed in the same value-at-risk framework as market risk.

In the assignment, students use CreditManager, J.P.Morgan's software implementation of CreditMetrics, to evaluate investment decisions, credit extension and risk-mitigating actions. They learn to identify concentrations of risk within a single portfolio, and to explore various scenarios by investigating marginal risk statistics.^h They also compare different risk measures (value-at-risk, standard deviation, percentiles), to gain an appreciation of the decisions that need to be made in modeling credit risk, and how these decisions affect the outcome. The data set includes historic probabilities of default and migration (upgrade and downgrade), correlations, recovery rates, and credit spreads.

3.3 Transaction risk

With the widespread use of credit cards and rise of more complex transactions in electronic commerce, detecting fraudulent transactions automatically has become a major area for the use of data mining techniques. While credit risk is essentially a static problem (or made static by incorporating migration), transaction risk is fundamentally dynamic: whether a transaction is likely to be fraudulent depends on both the usage pattern in the past and recent history. Bayes nets naturally incorporate prior knowledge in the structure by exploiting conditional independence and modeling as much as possible locally. They are a powerful method for discovering patterns and hidden causes in problems

^hMarginal risk refers to the difference between the total portfolio risk before the marginal transaction in comparison to it after that transaction.

with categorical variables (e.g., merchant type). After introducing Bayes nets, we apply them to predict the probability that a transaction is fraudulent, and we show how to select one of the possible actions in this setting of an asymmetric payoff matrix. In the lab, the students can choose between three hands-on approaches to the problem of detecting fraudulent transactions on credit cards; their choice tends to be influenced by the method they decided to use in their group project. The first approach uses a Bayes net to reinforce the concepts they had just encountered in class. The second approach uses a genetic algorithm to find relationships likely to indicate fraudulent behavior (see week 6, Section 2.3 above). The third approach uses nonlinear logistic regression expressed as a neural network (see week 4, Section 2.1 above). All approaches will be compared to ordinary logistic regression.

This part concludes with the discussion of a business case featuring First Union National Bank (Charlotte, NC). This bank, the nation's sixth largest banking company, uses a neural network in their non-linear transaction processing. This case study, written by Vasant Dhar at Stern (Dhar, 1997), serves as the transition from risk management to the final part of the course that focuses on the role data mining methods are playing (and are going to be playing) in business and finance.

4 Part 4: Projects and Contact with Wall Street Traders

The physical and intellectual proximity between Stern and Wall Street gives us access to world-class traders who are willing to share some of their insights on applying the methods discussed in the course. One of the elements of the last three weeks are *presentations by guest speakers* to the class with sufficient time for discussion. One of the talks is complemented by a case study from a firm where the students are asked to critique the process and the results and suggest and justify alternative methods. Besides allowing students to obtain further information, these interactions allow the trader to get to know some of the students. Such informal interactions often yield to valuable contacts including internships and jobs for the students after obtaining their degrees. Most of these presentations are scheduled in the last three weeks of the semester.ⁱ One or two classes are reserved to present special topics, ranging from execution issues (that the students already encountered in *Financial Information Systems*) to modern techniques such as independent component analysis (ICA)

ⁱA viable alternative is to lighten the load in the second part by eliminating the topic of either week 5 or week 7. This makes space for one external presentation on trading models at the end of the second part.

applied to pairs trading.

In the last three weeks, besides attending the lectures and discussions, the students are expected to primarily focus on their group projects.^j These *group projects* are an important part of the course: they deepen the understanding of the specific method for each group, and they are true real world introductions to some of the unforeseen and unforeseeable problems that are an integral part of working with new data.

The time line of the project is the following. Based on students' interests, groups of five students were formed at the end of the second class. In the third week, one specific project from the past was discussed in detail, highlighting the ingredients that contributed to its success. It is important to give a good example of the scope of the project and to clarify its role in the learning process. By week 4, each group will submit two one-page proposals that includes a description of the data sets. Several iterations with feedback from the instructor and the teaching assistant, as well as learning about new techniques during the course, tends to change the proposal significantly. By week 8 (before spring break), each group is required to converge on one of their two proposals. The selected proposal is finalized in week 9 to a well-focused and sufficiently narrow contract between the group members and the instructor. It is important that the expectations and evaluation criteria are made clear. Weeks 10 and 11 explore the data and obtain simple benchmarks for the subsequent evaluation of the proposed method. In the remaining three weeks, the entire lab time is focused on obtaining results and writing up the project. During the week of final exams, half a day is set aside for the presentation of the course projects. These case presentations give students the gratification to show off what they have learned and applied to a problem they were genuinely interested in. This event is also open to first-year students as a preview of what they could learn in their next year, and to the sponsoring companies as a glance of this exciting field of data mining and financial engineering at Stern.

This report describes one "realization" of a course that grew out of the post-NNCM-96 workshop. I would like to thank the participants for this open and fruitful discussion. This document by no means pretends to do any justice to the full complex exchange at the workshop, but I hope it may serve as a specific example for what can be done in a single-semester course at a business school. I thank Sean Chen, Vasant Dhar, Steve Figlewski, Joel Hasbrouck, Ed Melnick, Sridhar Seshadri, Bruce Weber and Norm White for their valuable feedback

^jNo readings, prepared lab sessions, or individual home works are assigned in the last three weeks.

and help with coordination with their courses at Stern. I thank Fei Chen and Balaji Padmanabhan for sharing the students' perspective, and last but certainly not least Caroline Kim for all her help with proofreading, logistics, and everything else. I thank Sean Curry of The MathWorks Inc. for his truly enthusiastic support with MATLAB, and Doug Martin of the University of Washington (Seattle) and Mathsoft, Inc. for data, code, and support with S-Plus. I genuinely welcome any comments, related experiences and suggestions.

References

- CIFER. 1997. *Proceedings of the IEEE/IAFE 1997 Conference on Computational Intelligence for Financial Engineering*. IEEE Service Center, Piscataway, NJ.
- CreditMetrics. 1997. *CreditMetrics-Technical Document*. J.P.Morgan, New York, first edition.
- Dhar, V. 1997. Managing credit risk through embedded intelligence in on-line transactions processing: First Union National Bank, Charlotte, NC. Technical report, Department of Information Systems, Leonard N. Stern School of Business, New York University.
- Fama, E. F. and K. R. French. 1992. The cross-section of expected stock returns. *Journal of Finance* 47, 427-465.
- Knez, P. J. and M. J. Ready. 1997. On the robustness of size and book-to-market in cross-sectional regressions. *Journal of Finance* 52, 1355-1382.
- Matlab. 1997. *Getting Started with MATLAB; Using MATLAB (Version 5.1)*. The MathWorks Inc., Data Analysis Products Division, Natick, MA.
- Refenes, A.-P. N., Y. Abu-Mostafa, J. Moody and A. Weigend (eds). 1996. *Neural Networks in Financial Engineering (Proceedings of NNCM-95, London)*. World Scientific, Singapore.
- RiskMetrics. 1997. *RiskMetrics-Technical Document*. J.P.Morgan / Reuters, New York, fifth edition.
- S-Plus. 1997. *User's Guide (Version 4.0)*. Mathsoft Inc., Data Analysis Products Division, Seattle, WA.
- Weigend, A. S., Y. S. Abu-Mostafa and A.-P. N. Refenes (eds). 1997. *Decision Technologies for Financial Engineering (Proceedings of NNCM-96, Pasadena)*. World Scientific, Singapore.