UNIVERSIDAD COMPLUTENSE DE MADRID

FACULTAD DE CIENCIAS ECONÓMICAS Y EMPRESARIALES
Departamento de Fundamentos del Análisis Económico II
(Economía Cuantitativa)

AUTOMATIC PREDICTION AND MODEL SELECTION

MEMORIA PARA OPTAR AL GRADO DE DOCTOR

PRESENTADA POR

Maximiliano Marinucci

Bajo la dirección de los doctores
Teodosio Pérez Amaral y Halbert White

Madrid, 2008

- **ISBN: 978-84-669-3174-8**

AUTOMATIC PREDICTION AND MODEL SELECTION

By
Massimiliano Marinucci

SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY
AT
UNIVERSIDAD COMPLUTENSE DE MADRID
28223 SOMOSAGUAS, MADRID, SPAIN
JANUARY 2008

.

UNIVERSIDAD COMPLUTENSE DE MADRID
Departamento de Fundamentos del Análisis Económico II,
Facultad de CC.EE., Campus de Somosaguas, 28223 Madrid.

AUTOMATIC PREDICTION AND MODEL SELECTION

by Massimiliano Marinucci

Supervisors:

- Prof. Teodosio Pérez Amaral. Universidad Complutense de Madrid. Spain.
- Prof. Halbert White. University of California at San Diego (UCSD). USA.

Examining Committee:

- Prof. Alfonso Novales Cinca. Universidad Complutense de Madrid. Spain.
- Prof. Michael McAleer. University of Western Australia, Perth. Australia.
- Prof. Manuel Domínguez Toribio. Universidad Complutense de Madrid. Spain.
- Prof. Maria Isabel Ayuda. Universidad de Zaragoza. Spain.
- Prof. Cesar Molinas. Universidad de Barcelona. Spain.

This version printed January 14, 2008
.

.

To my parents: Maria and Faustino

Table of Contents

Abstract

This dissertation is about Automatic Model building and Prediction procedures that are useful to approximate and forecast the expected conditional mean of a stationary target variable. We review the theoretical foundations of model selection and compare the out-of-sample predictive ability of different automatic selection procedures, focusing especially on the the RETINA method proposed by Pérez-Amaral, Gallo & White (2003).

A new software implementation of RETINA called RETINA Winpack is proposed. This software piece is designed for immediate use by non-specialist applied researchers. As an important advantage over the original RETINA implementation, it handles extreme observations and allows for distinctive treatment of categorical inputs. Using RETINA Winpack, we present an empirical application to Telecommunications demand using firm-level data. RETINA Winpack is proven to be useful for model specification search among hundred of candidate inputs and for finding suitable approximations that behave well out-of-sample in comparison with alternative linear baseline models.

With the aim of increasing the flexibility of the RETINA method in order to deal with non-linearities in the target variable, a new method called RETINET is presented. It generalizes RETINA by expanding the functional approximating capabilities in a way which is similar to Artificial Neural Networks (ANN), by avoiding some of the difficulties related to their practical implementation. As an advantage over traditional ANN, RETINET's specifications retain, to some extent, analytical interpretability. Based on two different simulation examples the method provides favorable evidence with respect to the out-of-sample forecasting ability provided by both simpler and/or more complex modeling alternatives. RETINET balances between a) Flexibility b) Parsimony c) Reverse engineering ability, and d) Computational speed. The proposed method is inspired by a Specific to General philosophy, going from the *simple* to the *sophisticatedly simple*, avoiding unnecessary complexity.

Acknowledgements

I am indebted with Prof. Teodosio Pèrez-Amaral (Universidad Complutense de Madrid) for his constant support in my research and Prof. Halbert White (University of California at San Diego) for his insights in extending the RETINA procedure by using Artificial Neural Networks. I wish to thank the University of California at San Diego (USA) and the Universidad Complutense of Madrid (Spain) for granting a scholarship that gave me the opportunity to spend the 2005-2006 academic year at San Diego, where under the supervision of Prof. White I developed the RETINET algorithm and wrote part of this dissertation. The Spanish Ministry of Science and Technology (project SEJ2004-06948) is gratefully acknowledged for supporting part of this research.

I am also indebted with Homa Karimabadi[1] who gave me the opportunity to show the potential usefulness of the RETINET algorithm beyond the domain of Econometrics, and particularly, in the field of the Geo-physic sciences.

For their valuable comments I also would like gratefully acknowledge all the participants that assisted at the:

- International Telecommunication Society, 13rd Biennal Conference, Madrid, Spain, Sep. 8-10, 2002.
- International Telecommunication Society, 15th Biennal Conference, Berlin, Germany, Sep. 4-7, 2004.
- EC^2, 15th Conference, The Econometrics of Industrial Organization, Marseille, France, Dec. 16-18, 2004.
- Villa Mondragone 5^{th} Workshop in Economic Theory and Econometrics, Frascati (Rome), Italy, 3-6 July 2005.

In addition I also wish to thank all the professors of the doctorate courses at the Universidad Complutense of Madrid and especially professors Arthur Treadway, Alfonso

[1] CEO of Sciberquest Inc., Solana Beach, California, USA.

Novales and Alberto Mauricio as well as professors Franco Peracchi and Giuseppe Arbía who initiated me to Econometric discipline during my under-graduate courses at the faculty of Economics of the Università D'Annunzio of Pescara, Italy. Thanks are due to Prof. Lester Taylor (University of Arizona, USA) who recently retired, and Prof. Julia Campos (Universidad de Salamanca, Spain) for their useful comments and suggestions on my empirical research using the RETINA algorithm to forecast Telecommunications demand. I am also very grateful to Professors Michael Mc Aleer (University of Perth, Australia) and Manuel Domínguez Toribio for their final comments on this dissertation.

Thanks go also to Mr. Luis Castro[2], who never denied me the necessary work permits to attend the Ph.D. courses while I was employed at his company during the years 1998-2001.

A special mention goes to all my Ph.D. colleagues at the Universidad Complutense with whom I shared many moments of my years spent in Madrid. In particular I wish to mention David Martín-Barroso, Leonel Cerno and Juan Angel Jímenez Martín for the strong friendship I established with them. Among others I acknowledge Francisco Pascual (UCSD), Christian Brownlees (Università di Firenze, Italy) and Livio Fenga (Università di Tor Vergata, Rome, Italy) for the nice time enjoyed together in San Diego and many conversations we had on common research topics.

I am grateful to my brother Marco and my sister in law Daniela for their encouragement in taking this way.
Last but not least, I am grateful to my mother Maria and my father Faustino, for their patience and *love*, and especially for always trusting me. Without them this work would never have come into existence (literally).

Madrid, Spain — Massimiliano Marinucci
Summer, 2007

[2] Former CEO of Inner Strategic Market Research, nowadays Synovate Research, Madrid, Spain

.

Chapter 1

Introduction

All models are wrong, some of them are useful.
George Box (1979)

1.1 Problem Statement

Throughout this dissertation we consider the problem of *approximating* the expected value of a real valued target variable Y_t given a $p \times 1$ random vector of predictors, or inputs, X_t, where $t \in N$ represents a generic indexing of the available observations. Broadly speaking the ultimate goal of this problem is to translate a potentially complex phenomena in a compact mathematical representation which is useful and has certain desirable properties for *prediction*, using the information included in the data at hand. In what follows we assume the convention that at time t we observe X_t prior to the realization of Y_t for which the prediction is to be made. Formally we are interested to obtain a satisfactory approximation of

$$\mu(X_t) \equiv \mathbb{E}[(Y_t|X_t)]$$

By restricting our attention on predictions based on the conditional mean we rule out other type of approximations (e.g. those based on the conditional median or conditional mode). To what extent this mathematical representation will be acceptable depends on many aspects being studied and especially on its intended final use. Here we are concerned mainly with the *prediction* problem, and will consider *interpretational* issues only after a "certain type of representation of the data" has been chosen[1]. We may be tempted to think in models in terms of *causal relationships*

[1] We intentionally use the expression "certain type of representation of the data" to remark that

but in economics as well as other social sciences, our phenomena of interest Y_t is almost always generated from a *natural* experiment[2]. The main consequence is that we do not know the true form that μ_t assumes. In addition it may be well the case that some inputs may be not available, or if they are, they represent error-laden measurements of variables that may or not be causally related to the target variable. Because of this, it makes sense to assume that any representation $m \in \mathcal{M}$ we use for μ, is typically miss-specified. From a different point of view, we could say that a true μ may not exist or is of infinite dimension[3]. Related to the idea of *true* representation we have the following definition of *correct* and *incorrect* specification:

Definition 1.1.1. A model for $\mathbb{E}(Y_t|X_t)$ is *correctly specified* if $\mu(X_t)$ belongs to the collection $\mathcal{M}$ of the considered representations. Conversely when the collection $\mathcal{M}$ is restricted such that it doesn't include $\mu(X_t)$ then we say the model is *misspecified.*

As (White 2006) (p.462) argues, when the objective of the modeling is *prediction*, model misspecification is by no means a problematic aspect, provided that we choose suitably a representation m from a collection of possible $\mathcal{M}$ for μ. The availability of a representation m derived from the data that provides a good approximation to the underlying relationship, although may not have the exact functional form as the true solution μ, is still very useful. If in addition, we are able to find an analytical expression for m this would allow easy comparison to other models, could be used to extract the relative importance of various predictors, and (depending on the problem) could be used to decipher the underlying phenomena.

In any case, we must acknowledge that by using an element m of the collection $\mathcal{M}$ our predictions may be not accurate as if we had the information about μ, the true Data Generating Process (DGP)[4].

there is not a unique way but a vast array' of different possibilities which may be all equally valid and useful to represent the data, including the averaging (or combination) of different models.

[2] A natural experiment is a naturally occurring event or situation, which a researcher exploits to help answer a research question. Natural experiments are quasi-experiments in the sense that the researcher has little or no control over the situation that is being observed. Natural experiments rely solely on observations of the variables of the system under study, rather than manipulation of just one or a few variables as occurs in controlled experiments. The main consequence is that since we can not reproduce the same experiment in a laboratory, it is impossible to unequivocally determine causation in any representation built from natural experiment data.

[3] For example think about an autoregressive stationary process of infinite order, which is equivalent to a MA(1) process. If $\mathcal{M}$ is the restricted class of all possible autoregressive models, then μ is not in $\mathcal{M}$. Hence, any autoregressive model of finite order q in $\mathcal{M}$, represents just an approximation of the true DGP $\mu \equiv \text{MA}(1)$.

[4] We want to stress once again that the true DGP μ isn't necessarily related to the data X in a causal sense. In practical situations, the sample available doesn't necessarily contain all the relevant information, and some predictors are not necessarily relevant for prediction.

The challenge then becomes how to choose a representation m from $\mathcal{M}$ such that our predictions achieve a satisfactory level of accuracy in some sense that will be clarified later.

1.2 Model Complexity, Parsimony principle

In order to choose an element from the model $\mathcal{M}$, we may refer to two very broad and general principles that can be adopted as scientific modeling philosophy. Since the objective of our empirical analysis is to find a *best approximating model*, not a *true model*, the goal is to find a possibly accurate approximation $\hat{m}$ of the information contained in the data at hand. Indeed, there is a long-standing tradition in science that simple theories are preferable to complex ones. This is known as *Occam's razor* or the *principle of parsimony* that provides the following criterion for deciding among scientific theories or explanations: one should always choose the simplest explanation of a phenomenon, the one that requires the fewest leaps of logic. Or said differently, one should not increase, beyond what is necessary, the number of entities required to explain anything. This is almost opposite to *Epicurus principle* of multiple explanations that "if more than one theory is consistent with the data, keep them all". A well known example of multiple theories derived from the same data is Keplers laws of planetary motion and Copernicus's refinement of the Ptolemaic theory of epicycles. Ironically, at the time, Kepler's laws did not account for the known data quite as well as Copernicus's refinement. Kepler's laws were ultimately chosen over those by virtue of their simplicity.

1.3 The number of possible parameterizations

How many possible representations are there in $\mathcal{M}$? Each representation m maps from $\mathbb{R}^k \to \mathbb{R}$ where k is the number of parameters to be estimated. In linear regression given k predictors, there are 2^{k-1} representations in $\mathcal{M}$. This number rapidly grows as the dimension of the coefficient vector does, and in high dimensional problems (i.e. $k > 40$) this is a serious problem, both for *model selection* and *estimation*, because we should find the solution of the underlying discrete combinatorial optimization problem. For more complicated models (e.g. Artificial Neural Networks) the estimation of a single parameterization may be so time-consuming that it is practically impossible to find the "best" combination of predictors.
Since it is usually infeasible to evaluate all possible representations, heuristic methods are applied to find a suitable subset of $\{1, \ldots, K\}$ in the space of all non-empty

subsets. In linear regression this problem is called *Subset regression* (Miller 2002, Tibshirani 1996*b*, Breiman 1995). In principle, all possible combinations of independent variables should be tried for selecting a suitable representation. Since nowadays databases easily reach many thousand of observations, consist of hundred of variables and often must be processed in real time, this could be a formidable task, even if high performance computers are available.

Efficient search techniques like *branch and bounds*[5] or *leap and bounds* (Brusco & Stahl 2005) methods may be applied for exhaustive search and solve the model selection problem if the number of possible coefficients to be estimated is not too high. In linear regression, computational efficiency can also be achieved by using the *sweeping* technique which will be discussed in chapter 4.

Besides the practicability of these approaches, there are also several theoretical considerations:

- The contribution of a single variable to the prediction of Y may not easily be assessed if only a small number of observations is available. (Efficiency)

- A simple criterion, like the goodness of fit, R^2, may lead to wrong conclusions if the number of selected variables is large relative to the number of observations (over-fitting).

- The selection of combinations is guided by the available data; thus the resulting final selection reflects the “best” model for the given data set, and not the “best” subset for the population. This leads to “selection” biases as discussed by Breiman (1996*b*).

- Some of the selection methods are specifically tailored to linear (regression) models; they are unusable with non-linear methods such as neural networks.

- Computational efficiency is an important aspect to be taken into account, especially if the number of predictors is very high (> 100).

Technically, any proposed model building and selection procedure should be able to overcome most of the practical difficulties in applied work, and automate whenever possible routine operations on the data. In building these procedures we find useful to mention mining projects which usually consist of six phases, as adapted from the

[5] From Wikipedia: “Branch and bound (BB) is a general algorithm for finding optimal solutions of various optimization problems, especially in discrete and combinatorial optimization. It consists of a systematic enumeration of all candidate solutions, where large subsets of fruitless candidates are discarded en masse, by using upper and lower estimated bounds of the quantity being optimized.”

industry standard, CRISP-DM (Cross-Industry Standard Process for Data Mining, `www.crisp-dm.org`):

1. **Research understanding phase** - Translate research objectives into the formulation of a data mining problem definition. - Prepare a preliminary strategy for achieving these objectives.

2. **Data understanding phase** - Collect the data. - Use exploratory data analysis to familiarize yourself with the data and discover initial insights. - Evaluate the quality of the data.

3. **Data preparation phase**

    (a) *Clean the raw data.* For example, there may exist gaps in the data and/or different data sampling rates may have been used in the collection of various variables. This phase is very labor intensive. - Select the cases and variables to analyze. - Cast the data, including the variables of interest, in a form suitable for the modeling tools.

    (b) Perform transformations on certain data variables.

4. **Modeling phase** Select and apply appropriate modeling techniques. - Calibrate model settings to optimize results. - Often, several different techniques may be used for the same data mining problem. - Sometimes it may be necessary to loop back to the data preparation phase to bring the form of the data into line with the specific requirements of a particular estimation technique.

5. **Evaluation phase**

    Evaluate the one or more models delivered in the modeling phase for quality and effectiveness before deploying them for use in the field. - Determine whether the model in fact achieves the objectives set for it in the first phase. - Establish whether some important facet of the research problem has not been accounted for sufficiently. - Come to a decision regarding use of the results.

6. **Deployment phase**

    Make use of the models created: Model creation does not signify the completion of a project. Periodic revisions and monitoring of its out-of-sample predictive ability remains important.

Of the six steps above, it is phases 3b, 4, and 5 that lend themselves to an algorithmic treatment and are the subject of Automatic Modeling techniques. These phases of a modeling project pose the most serious challenge to non-experts. In standard data-mining approaches, the user is left to decide on and to integrate particular strategies for each of these three phases. This is a complex task even for an experienced user. Throughout this dissertation we propose an approach that integrates and automates these steps based on a set of *optimally* motivated strategies. The approach offers automated model building and model evaluation. The automated methodologies are exposed through high-level interfaces which hide the statistical concepts from the users, thus helping to bridge the conceptual gap usually associated with automatic modeling. Note, however, that the automation does not remove the need for human direction of data mining. Since a sufficient exhaustive search almost leads to some apparent pattern in the data, there is always a risk to mistake the spurious for the substantive (White 2000). Also since the data is often used twice, both for model selection and inference, the term “Automatic Modeling” has also acquired a negative connotation because is easily leads to “Data Snooping”, or cruel activities like “torture the data until it confesses” (Miller 2002). The approach adopted here, tries to overcome these negative aspects of data dredging. Rather, here we consider that Automatic Modeling techniques should be incorporated into a human process of problem solving as useful tools. The human direction is particularly essential in the research and the data understanding phases, as well as in the deployment phase of the whole modeling process.

1.3.1 Automatic Modeling Procedures

The expression “Automated Modeling” is usually used as a synonymous of “Machine Learning” or model building without human intervention. Recently much research has been initiated in the use of automated model selection procedures, taking advantage of our access to computer power. Here our main concern is to implement *automatic data-driven strategies* to pick the most convenient representation (subset in $\mathcal{M}$) defined as the best approximation to $\mathbb{E}(Y_t|X_t)$. From this point of view, *approximation* and *model selection* problems are strictly related and interact constantly in any modeling procedure. For applications in Econometrics, readers may refer to the $20th$ anniversary issue of *Econometric Theory* (vol.21 2005), a monograph exclusively dedicated to Automated Inference and the future of Econometrics. This volume contains contributions on various aspects of the theme of automation, introducing the notion of *automatic discovery*, analyzing the *validity* *of selection procedures*, and discussing the *methodological implications for inference* that arise in the use of automated procedures. We will review some of these aspects in the first chapter. Here we define *Automated Modeling* as an *algorithmic approach to data exploration and knowledge discovery.* Automated Modeling is directly related to “Data-Mining”. Data mining is an umbrella term and is being routinely used in a variety of data-centric fields. In the recent years these techniques have acquired a positive connotation as a means of automatically extracting valuable information and relationship from massive databases. In real world applications, Data mining algorithms are used to monitor daily transactions on nearly one billion active credit and debit cards and have significantly reduced fraud rates. Use of data mining techniques in the analysis of biological data has given rise to a rapidly growing field of bioinformatics (see special issue on Bioinformatics in Pattern Recognition, 39, 2006). NASA’s Mars Exploration Rover Missions provided successful and well publicized deployment of data mining techniques to auto-navigation and path planning for rovers. These techniques are being further developed for rover exploration in future planetary missions. Recently, data mining has begun to find more use in analysis of spacecraft data and some interesting studies have been attempted (Lundstedt 1992, Dmitriev & Suvorova 2000, Jankovicová, Dolinskỳ, Valach & Vörös 2002, O’Brien & McPherron 2003).

1.3.2 Reverse engineering

Less known and much less emphasized, but of significant interest to sciences, are automatic procedures that provide their solution in terms of analytical functions. We refer to the automatic derivation of analytical solutions from the data as *reverse engineering* the data. An example would be to derive Newton’s law of gravity from the planetary data. The analytical form of the model is essential. It allows one to examine the role of various terms and decipher the underlying relation and allow sharing and easy computation by others. Automatic modeling procedures that have reverse engineering properties are interesting because they are useful for inductive reasoning. An excellent example of inductive reasoning in the field of Economics is given by Cobb & Douglas (1928) who proposed the well known Cobb-Douglas production function which is still widely used in many applied economic studies:

$$T = AK^{\alpha}L^{(1-\alpha)} \quad \alpha \in [0,1] \tag{1.3.1}$$

where K and L are the quantities of capital and labor respectively, and $\alpha/(1-\alpha)$ represents the substitution elasticity among the inputs K and L. It is remarkable that this function was discovered not from an *a priori* reasoning but through a process of induction from the empirical data. Cobb and Douglas observed that labor share of the US income had been approximatively constant over time during the years 1899 to 1922 and independent of the relative prices of capital and labor. They deduced, under the assumption of constant returns to scale, perfect competition in the input and output markets and profit maximizing firms, that the production function had to be of the form given in 1.3.1. The Cobb-Douglas function forms the foundation for Solow's growth theory (Solow 1956) and research into productivity growth factors, such as "technological progress" and "human capital development". The same inductive reasoning was applied by Arrow, Chenery, Minhas & Solow (1961) to discover the Constant-Elasticity-of-Substitution function (C.E.S.), which represents a more general production function that the Cobb-Douglas. These examples are by no means exhaustive of the "inductive" process of discovery which has a long tradition in science. While the prediction of observations is a forward problem, the use of actual observations to infer the properties of a model is an inverse problem. Inverse problems are difficult because they may not have a unique solution, and it is for this reason that automatic modeling strategies with *reverse engineering* capabilities have inevitably a heuristic nature, but yet are still very useful for building useful representations of real world phenomena. The key advantage of automatic procedures with reverse engineering capabilities is that the solution is always in an analytical form rather than a "black box" as in most standard data mining modeling technique like Artificial Neural Networks (ANN). It is rather surprising that the reverse engineering capability of mining algorithms has not been in the forefront of scientific data mining. Nevertheless, deriving analytical equations from data does not in general always produce the same exact functional form. However, the resulting analytical equation will be a very close proxy for the actual equation. Ideally one would like to be able to derive the underlying laws describing a system from the data. In trying to craft such a capability in algorithm form, however, one is faced with a number of issues. First, data samples are usually limited and have embedded noise. Second, the necessary functional forms (e.g., ratios, derivatives, etc.) appropriate for a system that one is modeling may not be covered in the collection of bases and/or the transformations of a given algorithm. Third, observe that definition of the *optimality* is relative to the goal (prediction or structural interpretation and estimation) and the specific class of candidate representations in $\mathcal{M}$.

1.4 Outline

This dissertation is organized in 6 chapters including this introduction. In chapters 2 and 3 we review the theoretical foundations for the developments included in chapters 4,5 and 6. In particular, chapter 2 includes a review on methods and the work done so far in the field of Prediction, Model Assessment and Selection. We will review approximation techniques using linear modeling techniques in chapter 3. In chapter 4 we review the fundamentals of Automatic Model building and selection and introduce the reader to different modeling strategies, focusing especially on the features of the RETINA method and providing Monte Carlo evidence that this method is as good as many others and has the advantage of incorporating approximation capabilities that other automatic methods do not have. Chapter 5 includes an empirical application of RETINA to model the demand of US firm level telecommunications data, which shows the utility of RETINA as an approximation and an automatic modeling tool. Chapter 6 presents the development of a new automatic model building tool called RETINET. The method takes advantage of libraries of highly non-linear transformations. Conclusions follow by providing a brief summary of this work and suggesting some future research directions.

Chapter 2

Prediction, Model Assessment and Selection

2.1 Optimal prediction

In many instances in economics and finance we need to obtain a point forecast of a target variable Y_t given a vector of predictors X_t. One common way to proceed, under quadratic loss, is to construct point forecasts as approximations to the conditional expectation of Y_t given X_t, $\mu(X_t) = \mathbb{E}[Y_t|X_t]$ which yields the best possible prediction of Y_t given X_t under Prediction Mean Squared Error (PMSE), provided Y_t has finite variance. That is, μ solves the problem

$$\min_{m \in \mathcal{M}} \mathbb{E}\left[(Y_t - m(X_t))^2\right]$$

where $\mathcal{M}$ is the set of functions m of X_t having finite variance and the expectation is taken with respect to the joint distribution of Y_t and X_t. Given X_t the PMSE may be decomposed as:

$$\begin{aligned}
\mathbb{E}[Y_t - m(X_t)]^2 &= \mathbb{E}[(Y_t - \mu(X_t) + \mu(X_t) - m(X_t))^2] \\
&= \mathbb{E}[(Y_t - \mu(X_t))^2] + \mathbb{E}[(\mu(X_t) - m(X_t))^2] \\
&\quad +2\mathbb{E}[(Y_t - \mu(X_t))(\mu(X_t) - m(X_t))] \\
&= \underbrace{\mathbb{E}[(Y_t - \mu(X_t))^2]}_{\text{Pure Error}} + \underbrace{\mathbb{E}[(\mu(X_t) - m(X_t))^2]}_{\text{Approximation Error}} \qquad (2.1.1)
\end{aligned}$$

The equality follows by applying the law of iterated expectations since:

$$\begin{aligned}
\mathbb{E}[(Y_t-\mu(X_t))(\mu(X_t)-m(X_t))] &= \mathbb{E}[\mathbb{E}[(Y_t-\mu(X_t))(\mu(X_t)-m(X_t))]|X_t] \\
&= \mathbb{E}[\mathbb{E}[(Y_t-\mu(X_t))|X_t](\mu(X_t)-m(X_t))] \\
&= 0\cdot[\mu(X_t)-m(X_t)] \\
&= 0
\end{aligned}$$

The PMSE is decomposed in two parts: the *pure Prediction Mean Square Error* σ^2 and the *Approximation Mean Square prediction Error* (AMSE), $\mathbb{E}[(\mu(X_t)-m(X_t))^2]$. It follows that in order to minimize the PMSE it is sufficient to minimize the AMSE, which is always non-negative. Notice that the AMSE is zero if and only if we choose $m=\mu$, that is *only if the model $\mathcal{M}$ is correctly specified* in the sense that the representation m is the right one. In practice the parameters of our model $m(X_t)$ are unknown and we have to estimate them. One possibility is to use the OLS estimator, $\hat{m}(X_t)$. We next define the loss or *generalization error* as

$$L=\mathbb{E}[(Y_t-\hat{m}(X_t))^2] \tag{2.1.2}$$

where the expectation above averages over everything that is random. In many occasions we might want to use a linear model to approximate the unknown conditional expectation of our target variable given our set of predictors. In this case our model is given by $\mathcal{L}=\{l(X_t):l(X_t)=X_t'\beta,\beta\in\mathbb{R}^k\}$ and we might estimate β using the OLS method, which is well known to be consistent for β^* such that $\beta^*=\text{argmin}_\beta\mathbb{E}[(\mu(X_t)-X_t\beta)^2]$ so when we estimate the parameters and compute the approximation error (AMSE), it can be decomposed as follows

$$\begin{aligned}
\mathbb{E}[\big(\mu(X_t)-\hat{m}(X_t)\big)^2] &= \mathbb{E}[\big((\mu(X_t)-\mathbb{E}[\hat{m}(X_t)]+\mathbb{E}[\hat{m}(X_t)]-\hat{m}(X_t)\big)^2] \\
&= \mathbb{E}[((\mu(X_t)-\mathbb{E}[\hat{m}(X_t)])^2]+\mathbb{E}[\big(\mathbb{E}[\hat{m}(X_t)]-\hat{m}(X_t)\big)^2] \\
&\quad +2\mathbb{E}[\big(\mu(X_t)-\mathbb{E}[\hat{m}(X_t)]\big)\big(\mathbb{E}[\hat{m}(X_t)]-\hat{m}(X_t)\big) \\
&= \underbrace{\mathbb{E}[((\mu(X_t)-\mathbb{E}[\hat{m}(X_t)])^2]}_{\text{Bias}^2}+\underbrace{\mathbb{E}[\big(\mathbb{E}[\hat{m}(X_t)]-\hat{m}(X_t)\big)^2]}_{\text{Variance}}
\end{aligned}$$

as $\mathbb{E}[\big(\mu(X_t)-\mathbb{E}[\hat{m}(X_t)]\big)\big(\mathbb{E}[\hat{m}(X_t)]-\hat{m}(X_t)\big)$ is 0.

The first component of the above decomposition is squared bias; that is, the amount by which the average of the estimate differs from the true mean. The second term is the variance, the expected squared deviation of $\hat{m}(X_t)$ around its mean. Importantly what this expression tells us is that there is a trade-off between bias and variance. For linear models fitted by ordinary OLS under usual assumptions and correct specification, the expected estimation bias is zero. However an unbiased representation may have a large mean-squared error if it has a large variance. This will be the case if $m(X_t)$ is highly sensitive to the peculiarities (such as noise, collinearity and the choice of the sample points) of each particular estimation set and it is this sensitivity which causes regression problems to be ill-posed (Tikhonov & Arsenin 1977). Introducing *bias* is equivalent to restricting the range of parameterizations m for which a model $\mathcal{M}$ can account. Typically this is achieved by reducing the number parameters. In the linear regression terminology, this would consist in doing *subset selection* by choosing only the most useful predictors. This can be carried out directly by the researcher but there are many methods to automatically perform this operation, some of which will be discussed in chapter 4.

2.1.1 Regularization

Another possibility of introducing bias consists by imposing a penalty on the magnitude of the parameters such they are shrunk towards the origin. This operation is called "regularization". *Ridge regression* Hoerl & Kennard (1970) is a specific form of regularization with quadratic penalty:

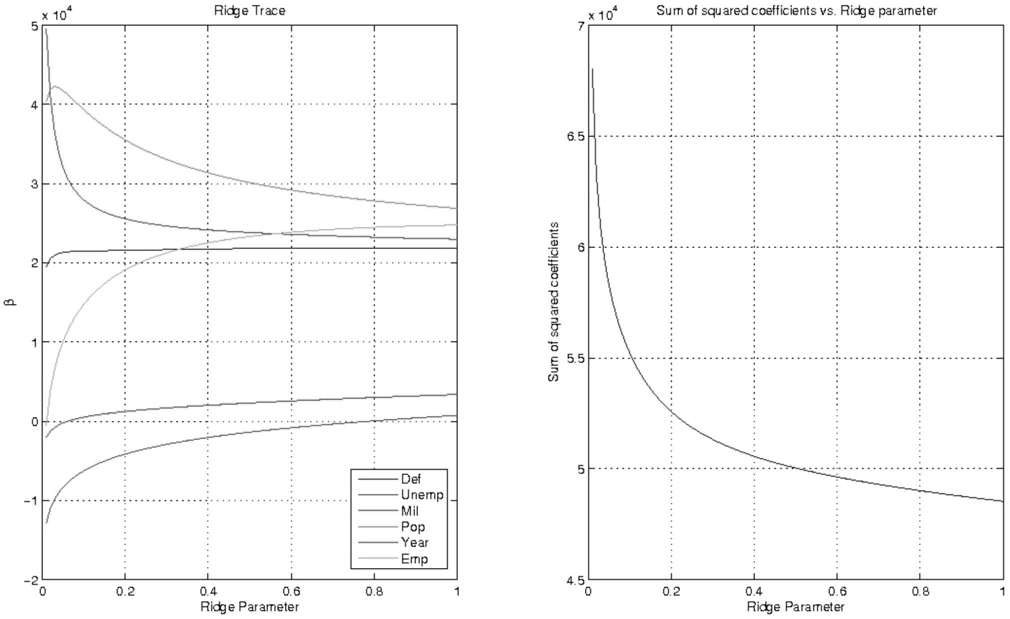
$$\hat{\beta}_{\text{RIDGE}} = \text{argmin}_{\beta}\{(Y - X\beta)^2 + \lambda\beta^2\}$$

Where λ is called penalty or "ridge" parameter. An equivalent way to write the ridge problem is:

$$\hat{\beta}_{\text{RIDGE}} = \text{argmin}_{\beta}\{(Y - X\beta)^2\} \qquad s.t. \qquad \|\beta\|_2 \leq s$$

where s is the size of the constraint imposed on the coefficients. This method can not only reduce the variance but also the bias which model selection introduces (Miller 2002). Ridge regression reduces the effective number of parameters. In other words the resulting loss of flexibility makes the chosen representation less sensitive. Compared with subset selection which is a discrete process (a predictor is included or not in the approximating function), regularization methods are continuous processes. This is because shrinking parameters towards the origin is equivalent to subset selection in the limit, when the coefficient of a given predictor is set to zero. There are many forms of regularization which depend on the functional form of the penalties used. An example is the LASSO (Tibshirani 1996*b*) which uses a L_1 penalty function (see chapter 4). Out of sample prediction performance is usually enhanced by these methods, especially when the design matrix is severely ill-conditioned or when the complexity of the parameterization needs to be limited. Thus *we may improve accuracy of prediction either by expanding the set of candidate models $\mathcal{M}$ and/or*

Figure 2.1: Ridge Example with Longley data



by regularization, that is, shrinking some parameters towards zero. An improved accuracy in prediction may be achieved in some equilibrium point where there is an acceptable balance between *model complexity* and *generalization ability.* As an example of ridge regression we use a well known data set which is often employed as an example of ill-conditioning of the design matrix. This is the Longley data set (Longley 1967). The ridge trace is depicted in figure 2.1 (left), which shows how the coefficients vary as a function of the ridge parameter λ. On the right we show how the sum of the squared coefficients decays as a function of the penalty parameter. The ridge trace was introduced first by Hoerl & Kennard (1970) in order to choose the ridge parameter for which the coefficients are not rapidly changing and have “sensible” signs. In practice this method has been criticized for being highly subjective and other methods are usually employed such as cross-validation (also GCV is a popular choice, see Golub, Heath & Wahba (1979)).

2.2 Model Selection

There is no agreement on a general definition of *model selection*, since it goes far beyond subset selection in regression models. Broadly speaking the need for model selection procedures arises when researchers have to decide among models classes based on data. In general, we should apply any selection procedure with some care, examining the structure of several good- fitting parameterizations rather than restricting our attention to a single "best". In practice it is common to estimate different models before choosing the one which is used for practical purposes: *prediction*, *interpretation* and *hypothesis testing*. In our context, the goal of model selection is to assess the performance of different alternative parameterizations m in order to choose the best one in terms of predictive ability. In linear regression when the objective is *interpretation* and/or *hypothesis testing*, the problem of model selection consists in finding a suitable subset of predictors. The model selection approach is different from the more traditional hypothesis testing approach and is appealing because Model selection allows one to focus on the issue at hand: out of sample forecasting performance and doesn't require the specification of a correct model for its valid application, as does the traditional hypothesis testing approach. In fact the distributional properties of single coefficients and statistics used in the selection process, depend upon every modeling decision. See for example Leeb & Pötscher (2005) for a discussion of these issues[1]. Here we remark that a good subset of predictors for prediction may be inappropriate for hypothesis testing and/or interpretation of each single coefficient.

When considering the properties of a model selection procedure an obvious question that one may ask is whether it is *consistent*. A selection procedure is *consistent* in the sense that it selects the true DGP with probability approaching one as the sample size increases.

$$P\big\{\hat{m}_n = \mu\big\} \xrightarrow{P} 1$$

The subindex n makes explicit the dependency of any quantity on the sample size n. Obviously such property is useful to the extent that the true representation is included in the set of considered specifications $\mathcal{M}$. If we do not make this assumption, *consistency* becomes a less useful criterium to evaluate the asymptotic behavior of any model selection procedure. As we will see further the asymptotic behavior of many model selection procedures depends on the presence or absence of the true

[1] These authors show that data-driven model selection have an important impact on any post-selection estimator, and ignoring these effects leads to invalid inference. Since our concern is about approximation of unknown functional forms we skip this discussion here.

DGP in $\mathcal{M}$. Since we do not make any assumption regarding the true representation of μ we consider an optimality criterion for a model selection procedure which is better suited in the context of approximation and prediction. Considering our prediction problem, a selection procedure is optimal if it tends to select a representation m which provides the lowest *generalization* error (or smallest expected loss) as the estimation sample size increases. In linear regression this means that a subset of predictors that contains all relevant inputs will be called a "good" subset (thus yielding a correctly specified model), while the subset that contains all relevant inputs but no others will be called the "best" subset (which yields the correctly specified model with the smallest dimension). Here we consider "good" and "best" subsets in an asymptotic sense (as the number of observations available goes to infinity). With a small estimation sample (training set), it is possible that a subset that is smaller than the "best" subset may provide better generalization error.
Shao (1997) provides a more general definition of consistency that includes the case where all models are misspecified (as it typically occurs in economics and finance as well as other disciplines). For Shao (1997), a selection procedure is consistent if

$$P\{\hat{m}_n = m_n^*\} \xrightarrow{P} 1 \tag{2.2.1}$$

Here m_n^* is the representation that minimizes mean squared error loss among the representations considered in $\mathcal{M}$. In some cases a selection procedure is not consistent but $(\hat{m}_n)$ is still "close" to m_n^* in the sense that:

$$\frac{L_n(\hat{m}_n)}{L_n(m_n^*)} \xrightarrow{P} 1 \tag{2.2.2}$$

A selection procedure satisfying this condition is said by Li (1987) among others to be *asymptotic loss-efficient*. Here the subscript n refers to the fact that both losses explicitly depend on the sample size used for their estimation. Notice that the best parameterization does not imply the consideration of any truth μ, we simply refer to it as the best approximating representation m^*. Observe that the property of *Asymptotic Loss Efficiency* is a weaker requirement than *consistency* in the sense that, a consistent procedure is asymptotic loss efficient but the converse is not necessarily true. This brings us back to the problem of estimating the *generalization error*. The *plug-in* principle suggests to estimate this quantity using the available information, that is computing the sample average loss given by:

$$\hat{L}(\hat{m}) = \frac{1}{N}\sum_{t=1}^{N}(Y_t - \hat{m}(X_t))^2$$

However it is well known that this estimator doesn't represent a reliable estimate of the generalization error, because the same data is being used to fit the model and assess its error. The training error consistently decreases with model complexity, and in the limit, when $k \to N$ for a given N, it drops to zero. A model with zero training error would overfit the data and will typically generalize poorly. In fact as the model becomes more and more complex it is able to adapt to more complicated underlying structures (decrease in bias), but the estimation error increases with model complexity (increase in variance). In between there is an optimal model complexity that gives minimum *generalization* error. The consequence is that the training error will be an overly optimistic estimate of the generalization error. The expected difference between the generalization error and the training error is also called the *optimism* effect. In order to have a reliable selection procedure we need to solve this problem and we may adopt two different strategies, either estimating the *optimism* effect included in the training error, or alternatively estimating directly the generalization error using cross-validation or bootstrapping methods based on "re-sampling" (Weiss & Kulikowski 1991, Efron & Tibshirani 1993, Hjorth 1994, Plutowski & White 1993, Shao & Tu 1995).

2.3 Selection procedures

It can be shown that the optimism effect included in the in-sample error is positively related with the covariance between the Y_t and the predicted values $\hat{Y}_t$. In the linear regression model the expression of the optimism effect in a linear model is very simple (Hastie, Tibshirani & Friedman 2001):

$$\frac{2p}{N} \cdot \sigma^2 \tag{2.3.1}$$

This shows that the optimism increases linearly with the number of predictors p, but decreases as the training sample increases. Similar versions of the optimism formula also hold for other error models such as binary data and entropy loss. In linear models, statistical theory provides several simple estimators of the generalization error under various sampling assumptions (Efron & Tibshirani 1993, Miller 2002). These estimators adjust the training error for the number of parameters being estimated, and in some cases for the noise variance if that is known. Mallows' C_p (Mallows 1973) is an example. This model selection procedure is obtained using 2.3.1 :

$$C_p = N^{-1} \sum_t^N [Y_t - \hat{\mu}(X_t)]^2 + \frac{2p}{N}\sigma^2$$

One of the disadvantages of Mallows C is that the variance of the noise σ^2 must be known. In practice this is estimated by a consistent estimator. Generalization error estimators that do not require the noise variance to be known in advance, are:

AIC The *Akaike Information Criterion* (Akaike 1973), which is similar to the Mallows C_p procedure but applicable whenever a likelihood loss function is adopted. The AIC measure the relative Kullback-Leibler discrepancy. When errors are assumed to be gaussian then the AIC statistic is equivalent to Mallows C_p, thus in the case of linear regression model they are equivalent if the error term is assumed to be normally distributed. For models linear in the parameters and gaussian distributed errors we have:

$$\mathrm{AIC} = N \ln(\hat{\sigma}^2) + 2k$$

where $k = p+2$ represents the total number of estimated parameters, that is, p predictors, the constant term and the residual variance[2] It has been shown that the Akaike Criterion may perform poorly when there are too many parameters in relation to the available observations. Sugiura (1978) proposed a second order derivation for the AIC called *corrected AIC* (AICC):

$$\mathrm{AICC} = \mathrm{AIC} + \frac{2k(k+1)}{N-k-1}$$

It is usually advocated to always adopt AICC instead of AIC, since when N is large with respect to k the second order correction is negligible and since AICC $\approx$ AIC. Burnham & Anderson (2002) (pp.66) suggest its use when the ratio $N/k < 40$.

BIC The Schwarz's Bayesian Criterion, known as BIC (Schwarz 1978, Raftery 1995). The BIC procedure follows from a bayesian approach to model selection is generically:

$$\mathrm{BIC} = N \ln(\hat{\sigma}^2) + \ln(N)k$$

Notice that BIC is very similar with the factor of 2 replaced by $\ln(N)$ to AIC, but its motivation is quite different. It arises from a Bayesian approach

[2] The penalization term k is an asymptotic estimator of $tr(J(\theta) \cdot I(\theta)^{-1})$ where $J(\theta)$ and $I(\theta)$ are respectively the first and the second partial derivatives of the likelihood function. This bias adjustment term defines the Takeuchi Information Criterion (TIC) (Takeuchi 1976) . For a gaussian linear model : TIC= $N \ln(\hat{\sigma}^2) + tr(J(\theta) \cdot I(\theta)^{-1})$. Although TIC is more general than the AIC (see Burnham & Anderson (2002, p. 353-372) for further details on AIC derivation and TIC generalization), it is almost ignored in applied research because, unless the sample size is big, the elements of $J(\theta)$ and $I(\theta)$ will be poorly determined.

to model selection. In this case choosing the model with the lowest BIC is equivalent to choosing the representation with the largest posterior probability, that is, the probability of selecting m_i given the data Y, X:

$$\Pr[m_i|(Y,X)] = \frac{\Pr[(Y,X)m_i] \cdot \Pr(m_i)}{\int \Pr[(Y,X)m_i] \cdot \Pr(m_i)dm}$$

This approach takes explicitly into account that there is an uncertainty in estimating the parameterization m_i given the data set (Y, X). Assuming a gaussian prior, the implicit process prior for $\mu(X)$ is also Gaussian and it can be shown that (Burnham & Anderson 2002, p. 303)

$$\Pr[m_i|(Y,X)] = \frac{\exp(-\frac{1}{2}\Delta\mathrm{BIC}_i)\Pr(m_i)}{\sum_i^{\mathcal{M}} \exp(-\frac{1}{2}\Delta\mathrm{BIC}_i)\Pr(m_i)}$$

where $\Delta\mathrm{BIC}_i = \mathrm{BIC}_i - \mathrm{BIC}_{min}$ and $\Pr(m_i)$ is the prior probability placed on parameterization m_i. BIC_{min} is the value minimum BIC among all m's in the set $\mathcal{M}$.

MDL Rissanen's Minimum Description Length principle MDL (Rissanen 1978) has been developed in the field of information theory. In this framework, provided that each parametrization m gives a description of the observed data, we discriminate between competing m's based on the fit and the complexity of each description. The fundamental idea behind the MDL Principle is that any regularity in a given set of data can be used to compress the data, i.e. to describe it using fewer symbols than needed. For the linear model the model selection procedure is:

$$\mathrm{MDL} = \frac{N}{2}\ln(\mathrm{RSS}) + \frac{1}{2}\ln\det(X'_m X_m)$$

where X_m refers to the $n \times p$ matrix of the predictors included in the specification m, and RSS is the residual sum of squares.

All these measures[3] are quantifying the relative distance between candidate models both in terms of the goodness of fit and in terms of model complexity. Roughly speaking, they measure the quantity of *information* corresponding to each parameterization m estimated from the sample. Theoretical foundations of these measures have deep roots in information theory (Shannon & Weaver 1963). The idea is that

[3] For classification problems, the formulas are not as simple as for regression with normal noise. For example, see Efron (1986) regarding logistic regression.

information in the sample is something that can be quantified and that the quantity of information is closely related to its probability. For example the MDL criterion is motivated by the following problem: “how much data could one pack into a sequence of number of bits, or conversely, how could one store a certain amount of data using the least number of bits? If a parameterization may be represented as a sequence of 0 and 1, then any representation $m \in \mathcal{M}$ is packing a different amount of information contained in the sample. Define the probabilistic space $\{\mathcal{M}, \mathcal{F}, \mathcal{P}\}$ where m lives. The amount of information carried by each model m is strictly related to its *selection probability.* In other words the idea is that the amount of data one can pack into a certain number of bits is related to the redundancy or information content of the data. Intuitively this means that the more redundancy, the more parsimonious should be our representation m of μ.

The use of all the above procedures for model selection is quite simple. One picks up the model with lowest index over the set $\mathcal{M}$ of models considered. Notice that we can estimate not only the optimal model given $\mathcal{M}$, but also assess the relative merits of the representations considered. Other indexes as the generalized AIC such the GIC method (Nishii 1984), belong to the same families of indexes that try to estimate the optimism effect from the training sample.

2.4 Re-sampling methods

2.4.1 Split-sample or hold-out validation.

The most commonly used method for estimating generalization error is to reserve part of the data as a “test” set, which must not be used in any way during the estimation stage. The test set must be a representative sample of the cases that one wants to generalize to. After training, one predicts the values of Y_t on the test set, and the error on the test set provides an unbiased estimate of the generalization error, provided that the test set was chosen randomly. The disadvantage of split-sample validation is that it reduces the amount of data available for both training and validation (Weiss & Kulikowski 1991). If one uses this method to choose which of several different candidates m to use for prediction purposes, the estimate of the generalization error of the best model will be optimistic. To clarify this point, if we estimate several parameterizations using one data set, and use a second (validation set) data set to decide which representation is best, we will need use a third (test set) data set to obtain an unbiased estimate of the generalization error of the chosen model (Miller 2002). As we will see further the RETINA procedure splits the sample in three parts in order to overcome this problem. Hjorth (1994) explains how this principle extends to cross-validation and bootstrapping.

2.4.2 Cross-validation methods

Cross-validation is an improvement on split-sample validation that allows one to use all of the data for estimation. The disadvantage of cross-validation is that one has to re-estimate all the candidate representations m many times. In *K-fold* cross-validation, one divides the data into K subsets of (approximately) equal size. Then one estimates the model K times, each time leaving out one of the subsets from the estimation set, but using only the omitted subset to compute whatever error criterion is of interest. If K equals the sample size, this is called *leave-one-out* cross-validation. *Leave-D-out* is a more elaborate and expensive version of cross-validation that involves leaving out all possible subsets of D cases. Observe that cross-validation is quite different from the *split-sample* or *hold-out* method that is common in subset regression and Neural Networks training. In the *split-sample* method, only a single subset (the validation set) is used to estimate the generalization error, instead of K different subsets; i.e., there is no "crossing". The distinction between *cross-validation* and *split-sample validation* is relevant since the former delivers more reliable results for small data sets; this fact is shown by Goutte (1997) discussing the results of Zhu & Rohwer (1996). For an insightful discussion of the limitations of cross-validatory choice among several learning methods, see Stone (1977). A variant of *leave-one-out* cross-validation is *generalized* cross-validation(GCV) which was introduced by Craven & Wahba (1979). GCV chooses the model that minimizes

$$GCV_n \equiv \frac{N^{-1}\sum_{t=1}^{N}(Y_t - \hat{m}_n(X_t))^2}{\left(1 - N^{-1}trM_n\right)^2}$$

where trM_n denotes the trace of M_n, the projection matrix of our predictors (also usually called the *hat* matrix).

2.4.3 Bootstrapping

Bootstrapping is an improvement on cross-validation that often provides better estimates of generalization error at the cost of even more computing time. Bootstrapping seems to work better than cross-validation in many cases (Efron 1983). In the simplest form of bootstrapping, instead of repeatedly analyzing subsets of the data, one repeatedly analyzes re-samples of the data. Each re-sample is a random sample with replacement from the full sample. For the bootstrap it is common to use between 50 to 2000 re-samples. There are many more sophisticated bootstrap methods that can be used not only for estimating generalization error but also for estimating confidence bounds for Artificial Neural Networks (Efron & Tibshirani 1993). For estimating generalization error in classification problems, the *.632+ bootstrap* (an improvement on the popular *.632 bootstrap*) is one of the currently favored methods that has the advantage of performing well even when there is severe overfitting. Use of bootstrapping for Artificial Neural networks is described in Baxt & White (1995), Tibshirani (1996*a*), and Masters (1995). However, the results obtained so far are not very thorough, and it is known that bootstrapping does not work well for some other methodologies such as empirical decision trees (Breiman, Friedman, Olshen & Stone 1984, Kohavi 1995, Ripley 1996), for which it can be excessively optimistic.

2.5 Model Selection procedures for dependent data

When we consider time series data and therefore dependent observations, the model selection procedures outlined above retain their asymptotic properties as long as our model errors are martingale differences and thus are uncorrelated. Nevertheless, if they exhibit correlation the model selection criteria become biased and require generalizations. Generalized Mallow's C_L circumvents this problem by explicitly incorporating the error variance-covariance matrix, which of course has to be consistently estimated. The family of cross-validation statistics also has to be modified. One way of doing that is by considering H-block cross-validation procedures. H-block cross-validation (Burman, Chow & Nolan 1994) is appropriate under stationarity. This approach removes the t^{th} observation and h observations preceding and following the t^{th} observation, then computes the average of the square differences between the t^{th} value of the dependent variable and the predicted value when the $2H$ observations around observation t have been omitted from the data set for $t = 1, 2, \ldots, T$. Ordinary cross-validation is a special case of H-block cross-validation for which $H = 0$. This procedure can also be generalized and used for Leave-D-out cross-validation where H observations are removed preceding and following each of the D observations in the validation set and is called hv-block cross-validation (Racine 2000). A computationally efficient way to implement this algorithm is described in Racine (1997).

2.6 Asymptotic properties of model selection procedures

Shao (1997) provides a very general framework for studying the asymptotic loss-efficiency of linear model selection procedures when models are estimated by OLS. He extends previous results by Li (1987) and analyzes under which conditions criteria such as AIC, Mallow's C_p, BIC, cross-validation and generalized cross-validation are asymptotic loss-efficient. He distinguishes the cases where there is only one correctly specified representation (just the best model), more than one (many good models[4]) or all models are misspecified. When all models are misspecified or at most only one is correctly specified, AIC, Mallow's C_p, leave-1-out cross validation and generalized cross-validation are asymptotically loss-efficient as defined above. Notice that when models are misspecified, series function approximations (such as series basis expansions or neural network models) are good choices and in such a case regularity conditions for the optimality of the former statistics require the use of approximations that don't converge "too fast" to the true conditional expectation as well as sufficient error moment bounds (where the errors are defined as the difference between our target variable and the true conditional expectation). Nevertheless, criteria such as BIC and leave-D-out cross-validation with D/N tending to one are not asymptotic loss-efficient. They only have this property when more than one model is correctly specified. Moreover, in this case they are in addition consistent as shown by Shao. Regarding leave-D-out cross-validation, Shao's result seems inconsistent with the analysis by Kearns (1997) of split-sample validation, which shows that the best generalization is obtained with D/N strictly between 0 and 1, with little sensitivity to the precise value of D/N for large data sets. But the apparent conflict is due to the fundamentally different properties of cross-validation and split-sample validation. It is also important to keep in mind that the family of leave-D-out statistics with D/N tending to 0 have the same asymptotic behavior as leave-1-out cross-validation. With respect to hold-out cross-validation, it is important to mention that it would be asymptotic loss-efficient for a loss based on a number of observations equal to those in the training sample. Nevertheless it is not for a loss based on the whole sample.

[4] See section 2.2 for the definition of "best" and "good" representation in the case of linear regression.

2.7 Asymptotic loss efficiency in finite samples

The purpose of this section is to show the finite sample behavior of different model selection procedures when we approximate the conditional mean of a given data generating process by using a misspecified model. The main motivation of this exercise is of practical order: since we are interested in "'good approximations" of an unknown data generating process, using an asymptotic efficient selection procedure also warrants that we are picking a specification that should work well for forecasting purposes, given a set of competing misspecified parameterizations. As we discussed so far, if we assume that our models are misspecified the *consistency* property defined in 2.2.1 is no longer useful simply because the true data generating process cannot be found. Asymptotic loss efficiency criteria is more appropriate here, because in this circumstance, the probability under which we pick up a "good" approximating specification among those available, approaches one as the sample size grows. In order to show how these theoretical results work in practice, we conduct a small Monte Carlo experiment in which we measure asymptotic loss efficiency of AIC, its corrected version for small samples (AICC), BIC, Mallows C_l, Leave-one Out cross-validation (LOO CV) and Generalized cross-validation (GCV). On average, except BIC which under misspecification is not asymptotic loss efficient, we expect the ratio 2.7.1 to converge to one as we increase the sample size.

2.7.1 Design of Monte Carlo experiments

In our experiments we consider the following two non-linear data generating processes:

DGP1

$$Y_t = 1 + \ln X_{1t}^2 + \ln X_{2t}^2 + \varepsilon_t$$

DGP2

$$Y_t = 10 \sin(\pi X_{1t}) + 20(X_{2t} - .5)^2 + \varepsilon_t$$

In both cases the $N \times 2$ predictors matrix are independent and uniformly distributed on the interval $[0, 1]$. The error term ε is normal and centered at the origin with unitary standard deviation. In order to ensure our miss-specification hypothesis we approximate each of these functions with a single layer feed-forward neural network (SLFFNN). SLFFNN are non-parametric methods (see chapter 3) which are *universal approximators* (Hornik, Stinchcombe & White 1989) in the sense that they can fit any arbitrary function under some mild regularity conditions. The idea is quite simple as in many other non-parametric methods: a target variable is approximated by a superposition of simpler functions of the data typically called *basis functions.* Thus the selection problem here is to select an appropriate number of basis functions that forecast well out-of-sample. If too many basis functions are selected, there is a risk of over-fitting and an increase in variance forecasts. Conversely picking too few basis functions will produce under-fitting and an increase in forecasts bias. In both cases the approximation will perform poorly out-of-sample, hence we expect loss efficient selection procedures to pick an optimal number of basis with increasing probability as the sample size grows.

We consider sample sizes of $N = 100, 250, 500, 1000$ in our experiments. Since the smallest is $N = 100$ we limited the series expansion to $Q = 33$ basis functions:

$$\mathbb{E}(Y_t|X_t) \approx \sum_{q=1}^{33} \psi_q(X_{1t}, X_{2t})\hat{\beta}$$

where $\psi(\mathbf{X}_t) = \gamma_1^{-1/2} f[(\mathbf{X}_t'\gamma_2 - \gamma_0)/\gamma_1]$ with parameters $\Gamma = \{\gamma_0, \gamma_1, \gamma_2\}$ where γ_0 represents a centering vector, γ_1 is a scaling vector and γ_2 is a direction vector on the unit sphere in $\mathbb{R}^2$. The parameter set Γ is fixed a priori conveniently and only the β's are estimated by OLS[5]. The basis function ψ corresponds to a *Ridgelet* (see section 3.1.3) but we could have used alternative basis functions as well[6]. Given that the basis functions are added in a stepwise fashion we obtain up to $Q = 33$ candidate parameterizations for each DGP and each sample size. Model selection statistics are computed at each step and the specification with the lowest selection statistic is retained. Afterwards we compare the sum of the squared errors of the selected model against the sum of the squared errors of the truly best out-of-sample approximating model among all $Q = 33$ candidates. For this purpose we compute the following loss ratio:

$$\frac{L_n(\hat{m}_n)}{L_n(m_n^*)} \tag{2.7.1}$$

where $L_n(\hat{m}_n)$ and $L_n(m_n^*)$ are defined as in 2.2.2. The average value and the standard deviation of 2.7.1 is computed across 100 Monte Carlo simulations for each considered sample size. At each iteration we generate a test sample having a size of 25% of the estimation sample.

[5] We will give later a justification to proceed in this way, rather than estimating Γ by some non-linear optimization procedure.

[6] The logistic function is a popular choice as well . The main reason for which we chose ridgelets is due to their better approximating properties. See chapter 3 for deeper insights.

2.7.2 Results

Main results of the experiment for DGP1 and DGP2 are reported respectively in tables 2.1 and 2.2. From table 2.1 observe that all mean values of 2.7.1 except the one corresponding to BIC converge fast to one as the sample size grows. At the same time the number of selected basis increases and a better out-of sample performance is obtained, while this is not true for BIC which picks severely under-fitted specifications. Table 2.2 shows that from $N = 500$ to $N = 1000$ BIC diverges while other statistics converge to one. Again BIC which selects severely under-fitted specifications. Its standard deviation increases, meaning that uncertainty in picking the right amount basis functions for optimal forecasting increases as the sample grows instead of decreasing as do the others.

2.8 Conclusions

The literature on prediction and model selection is highly interdisciplinary and interest in this field is fast growing. The review we presented in this chapter in by no means exhaustive.

In this chapter we make the basic assumption that models can be regarded as convenient approximations of an unknown data generation process. Because of this assumption, any model is inherently misspecified. In this context we consider model selection procedures that behave well under the miss-specification hypothesis. If prediction is the goal, and miss-specification is assumed, *asymptotic loss efficiency*, rather than *consistency*, is a desirable property of any model selection procedure. Among others, AIC, AICC, Mallows C and GCV benefit of this property, while BIC does not, since it typically selects under-parameterized that have a high bias and poor out-of-sample performance assessed by the RMSE.

Table 2.1: Mean values of 2.7.1. Standard deviation in parenthesis. R=100 Monte Carlo replications of DGP2. Size of test sample $N_\tau = 250$. Notice all mean values except the one corresponding to BIC converge fast to one as the sample size grows. At the same time the number of selected basis increases and a better out-of sample performance is obtained, while this is not true for BIC which picks severely under-fitted specifications.

DGP1: $Y_t = 1 + \ln X_{1t}^2 + \ln X_{2t}^2 + \varepsilon_t$

Loss ratio (eq.2.7.1)						
Sample	LOO CV	AIC	AICC	BIC	Mallows C	GCV
100	1.2413 (.1597)	1.1618 (.1339)	1.1747 (.1380)	1.2990 (.1708)	1.2327 (.1519)	1.2173 (.1522)
250	1.1212 (.0649)	1.0843 (.0611)	1.0918 (.0595)	1.1745 (.0672)	1.0932 (.0616)	1.1021 (.0627)
500	1.0330 (.0388)	1.0241 (.0323)	1.0268 (.0341)	1.1418 (.0420)	1.0265 (.0329)	1.0287 (.0340)
1000	1.0049 (.0100)	1.0040 (.0083)	1.0040 (.0083)	1.1219 (.0289)	1.0040 (.0083)	1.0041 (.0083)
Number of Selected basis						
100	9.18 (4.1386)	14.23 (9.2530)	12.91 (8.68)	6.60 (1.4283)	9.17 (4.1184)	10.19 (6.2349)
250	11.86 (6.4172)	15.87 (9.0451)	14.81 (8.4979)	6.70 (1.0630)	15.13 (8.1298)	13.62 (7.9143)
500	23.91 (9.2337)	25.73 (8.4532)	25.17 (8.6788)	7.11 (1.5485)	25.23 (8.3831)	24.72 (8.6407)
1000	31.11 (3.8442)	31.16 (3.9591)	31.16 (3.9591)	7.70 (2.1190)	31.16 (3.9591)	31.14 (3.9548)

Table 2.2: Mean values of 2.7.1. Standard deviation of the mean in parenthesis. R=100 Monte Carlo replications of DGP2. Notice how, from $N = 500$ to $N = 1000$ BIC diverges while other statistics converge to one. This is because BIC which picks severely underfitted specifications (high bias). Also notice that its standard deviation diverges, meaning that uncertainty in picking the right amount basis functions for optimal forecasting increases as the sample size increases instead of decreasing as do the others.

DGP2: $Y_t = 10\sin(\pi X_{1t}) + 20(X_{2t} - .5)^2 + \varepsilon_t$

			Loss ratio (eq.2.7.1)			
Sample	LOO CV	AIC	AICC	BIC	Mallows C	GCV
100	1.1185	1.1938	1.1655	1.1492	1.1373	1.1311
	(.1496)	(.2376)	(.2187)	(.1438)	(.1703)	(.1727)
250	1.0719	1.0851	1.0829	1.1201	1.0835	1.0731
	(.0601)	(.0812)	(.0768)	(.0993)	(.0766)	(.0652)
500	1.0391	1.0364	1.0351	1.1153	1.0331	1.0327
	(.0429)	(.0413)	(.0401)	(.0489)	(.0384)	(.0392)
1000	1.0145	1.0166	1.0167	1.1256	1.0167	1.0166
	(.0137)	(.0162)	(.0163)	(.0632)	(.0163)	(.0164)
			Number of Selected basis			
100	8.10	11.12	9.72	5.41	8.74	8.16
	(4.3139)	(7.1879)	(6.1904)	(1.1670)	(3.9840)	(3.6762)
250	9.61	11.43	11.25	5.92	11.30	9.81
	(4.9130)	(6.7131)	(6.4177)	(.6274)	(6.3616)	(4.8098)
500	13.88	14.36	14.19	6.25	14.62	13.85
	(5.1755)	(5.4065)	(5.1414)	(.8986)	(5.2073)	(4.7315)
1000	17.62	17.82	17.50	7.66	18.32	17.41
	(4.6493)	(4.7987)	(4.6680)	(2.8713)	(4.9313)	(4.6478)

Chapter 3

Approximation Methods

In this chapter we will discuss approximation methods that will be incorporated in automatic prediction and model selection methods discussed later. Namely we will follow the approach proposed by White (2006). We will see that despite its simplicity, a linear parametric model can be easily adapted to provide a quite general framework useful for function approximation. This can be achieved by simple transformations of the original predictors, keeping the approximating equation linear in the parameters. These parameterizations may outperform more sophisticate non-linear models in situation where the estimation sample is small, the data is sparse or there is a low signal to noise ratio of the estimated parameters. Linear models are also appealing because they are easily interpretable and thus provide reverse engineering capabilities and mathematically represent first order Taylor approximations to $\mu(X)$.

In order to enhance the flexibility in linear parameterizations one can use simple transformations of the predictors that may include polynomial terms, or more involved non-linear combinations of the predictors as in Artificial Neural Networks (ANN). Here, in order to avoid computational problems which are typical in estimation of ANN's we will consider a class of function called Generically Comprehensive Revealing (Stinchcombe & White 2000) and we will point out the main differences of the approach adopted here with respect to the one typically taken in the ANN literature.

We restrict our attention to *parametric* regression methods. In fact, even if *non-parametric* regression methods rule out the possibility of model miss-specification, they pose special challenges in high dimensional problems. This fact is known in the literature as the *curse of dimensionality* (Bellman 1961). The main consequence is that if we wish to be able to estimate with the same accuracy as in low dimensions, we need the sample size to grow exponentially as the number of inputs increases.

As we will see, *parametric* methods are not totally exempt of this problem, because when the number of parameters to be estimated is high with respect to the available observations, the precision in estimation is affected. We consider approximation methods flexible enough to capture non-linearities in the data, which are computationally feasible and avoid numerical difficulties due to non-linear optimization. It is for this reason that models which are linear in the parameters play a central role in this chapter.

In what follows we assume that the data generation process is stationary in mean and variance and therefore includes a limited time dependence. For cross-section data, it means independent and identically distributed (i.i.d.) observations. In time series applications, stationarity is compatible with considerable time dependence. Here we assume as much dependence as is compatible with the availability of suitable asymptotic distribution theory (White 1984). Our discussion thus applies straightforwardly to unit root time-series processes after first differencing or other suitable transformations, such as those relevant for cointegrated processes. In order to simplify our discussion, we leave explicit treatment of these cases aside here. Relaxing the stationarity assumption in order to accommodate heterogeneity is not difficult, but the notation necessary to handle this relaxation is more cumbersome than is justified here.

3.1 Approximating parameterizations

Consider a parametrization of the form $m(X_t, \theta)$ to approximate μ_t as a function of the data X_t and some finite dimensional parameter vector $\theta \in \Theta$, such that m belongs to a collection of functions $\mathcal{M}$ having finite variance. The input vector may be chosen in advance, or may be measurements of random variables or both. In what follows we do not distinguish the two situations. The point prediction based on the representation m using the estimated θ for an *out of sample* predictor vector X_{t+1} is:

$$\hat{Y}_{t+1} = m(X_{t+1}, \hat{\theta})$$

We emphasize the *out of sample* nature of the predictor vector X_{t+1} since in practical applications forecasts are not based on the estimation sample, because the associated target variable Y_{t+1} is not available until X_{t+1} is observed. The objective of the prediction exercise is to reduce the uncertainty about the as yet unavailable Y_{t+1}, but in order to do this we have to approximate the observed Y_t given X_t. Our problem is to estimate θ so as to minimize the expected squared forecast error:

$$\min_{\theta} \mathbb{E}[(Y_t - m(X_t, \theta))^2] \tag{3.1.1}$$

Notice that by narrowing the solutions space of the parameterizations in $\mathcal{M}$, still allows for a very wide range of possible solutions. Representations which are linear in the parameters and linear in the predictors correspond to the well known linear regression model. Given p predictors and putting $\theta = \beta$ the model is:

$$\mathcal{L} \equiv \{m : \mathbb{R}^P \to \mathbb{R} |\ m(X_t) = l(X_t, \beta) \equiv X_t'\beta, \beta \in \mathbb{R}^P\}$$

Solving for β yields:

$$\beta^* = \arg \min_{\beta \in \mathbb{R}^{p+1}} \mathbb{E}[(Y_t - X_t'\beta)^2]$$

and provided that $\mathbb{E}(X_t X_t')$ exists and is invertible, it can be easily shown that the solution of this problem is:

$$\beta^* = \mathbb{E}[(X_t X_t')]^{-1} \mathbb{E}(X_t Y_t)$$

Now, $X_t'\beta^*$ is called the population linear projection of Y_t on X_t. Assuming the data has been generated by a stationary and ergodic process we can estimate β by OLS, since:

$$\hat{\beta} \xrightarrow{P} \beta^*$$

Applying the *plug-in principle* using the available sample information we substitute the unknown expectations with the corresponding sample means to estimate β:

$$\hat{\beta} = (X'X)^{-1}X'Y$$

where X is the $N \times p$ matrix with rows X_t' and Y_t is a $N \times 1$ vector of the response. The optimal point prediction forecast is then simply:

$$\hat{Y}_{t+1} = X_{t+1}'\hat{\beta}$$

Notice that the linear model makes a strong assumption about the dependence of $\mathbb{E}(Y_t)$ on X_t, namely that the dependence is linear in each predictor. Once we have fitted the model we can examine the predictor effects separately, in the absence of interactions.

Even if the linear model has several appealing features it may be not adequate to model non-linearities in the data. In general nonlinear models are employed because they allow greater flexibility than linear specifications and thus greater forecast accuracy is expected from them. A way to ensure this is to build representations

m which nest linear models in non-linear models. As alluded above, the challenge posed by attempting to use non-linear models is that their computation that may or may not behave well, in that the algorithm may or may not converge, and, even with considerable effort, the algorithm may well converge to a local optimum instead of to the desired global optimum. As the advantage of flexibility arises entirely from nonlinearity in the predictors and computational challenges arise entirely from nonlinearity in the parameters, it makes sense to restrict attention to parameterizations that preserve linearity in the parameters in order to ensure a closed form solution of the estimation optimization problem. For this purpose let's define the non-linear representation n as:

$$\mathcal{N} \equiv \{n : \mathbb{R}^K \to \mathbb{R} |\ n(X_t) = \psi(X_t)'\beta, \beta \in \mathbb{R}^K\}$$

where $n(X_t) = \psi(X_t)'$ is some non-linear function of the predictors. Solving this minimization problem yields:

$$\beta^* = \arg \min_{\beta \in \mathbb{R}^K} \mathbb{E}[Y_t - \psi(X_t)'\beta]^2$$

Under usual assumptions, the solution is simply the OLS estimator using $\psi(X_t)$ instead of the original predictors X_t:

$$\beta^* = \mathbb{E}[\psi(X_t)'\psi(X_t)]^{-1}\mathbb{E}[\psi(X_t)'Y_t]$$

Again we may use the *plug-in* principle to find an estimate $\hat{\beta}$ from the available sample. With the problem framed in this way, an important next question is:

> "What kind of functions ψ of the predictors should we consider?"

There is a vast range of possible choices of such functions; in the following we mention some of the leading possibilities. Choosing among these depends not only on the properties of the transformation functions, but also on ones prior knowledge about μ, and ones empirical knowledge about μ, that is, the data.

3.1.1 Approximation by simple transformations of the inputs

We now concentrate our attention on linear representations in the parameters that can improve the approximation ability of m using transformations $\psi(X_t)$ of the predictors X_t. From a practical point of view, a possibility is represented by using transformations of the form:

$$\psi(X_t) = X_{it}^{\alpha_1} X_{jt}^{\alpha_2} \quad \text{with: } \{i, j = 1, 2, \ldots, p\} \quad \text{and } \alpha_1, \alpha_2 = -1, 0, 1 \tag{3.1.2}$$

We will refer further to these special type of transformations as *level one transformations*, following the nomenclature adopted by Pérez-Amaral et al. (2003) who proposed them in order to gain flexibility for their automatic model selection algorithm called RETINA.

Notice that 3.1.2 delivers the identity transformation of the $i-th$ predictor when $\alpha_1 = 1$ and $\alpha_2 = 0$. Viceversa we get the identity transformation of the $j-th$ predictor when $\alpha_1 = 0$ and $\alpha_2 = 1$. Similarly, when $i = j$ and $\alpha_1 = \alpha_2 = 1$ we obtain interactions and when $i \neq j$ with $\alpha_1 = \alpha_2 = -1$ we obtain inverses of the interactions between predictors i and j. Notice that $\psi(X_t)$ also include a constant term for $\alpha_1 = \alpha_2 = 0$. As an example, if the original input matrix X includes just two regressors, X_1 and X_2, then we would obtain the following collection of predictors (we omit the observation index t without loss of generality):

$$\left\{1,\ X_1,\ X_2,\ X_1X_2,\ X_1^2,\ X_2^2,\ \frac{X_1}{X_2},\ \frac{X_2}{X_1},\ \frac{1}{X_1},\ \frac{1}{X_2},\ \frac{1}{X_1^2},\ \frac{1}{X_2^2},\ \frac{1}{X_1X_2}\right\}$$

In this special case ψ transformations include the original predictors, implying that the resulting parametrization for m would mix-up linear and non-linear terms. Also, given p predictors, m would have exactly $1 + 2p + 2p^2$ associated parameters to estimate (including the constant). Another possible extension is to consider higher order *level one* transformations of the form:

$$\psi(\psi_t) = \psi_{it}^{\alpha_1}\psi_{jt}^{\alpha_2} \qquad i, j = 1, 2, \ldots, p \qquad \alpha_1, \alpha_2 = -1, 0, 1$$

we obtain:

$$\psi(\psi_t) = (X_{it}^{\alpha_1}X_{jt}^{\alpha_2})^{\alpha_1}(X_{it}^{\alpha_1}X_{jt}^{\alpha_2})^{\alpha_2}$$

which involves higher order polynomial terms up to the fourth order and interactions between cubic and squared terms. As an example if we construct $1+2p+2p^2$ level one transforms we will have $1+2(1+2p+2p^2)+2(1+2p+2p^2)^2 = 5+12p+20p^2+16p^3+8p^4$ terms. As the reader may notice from table 3.1 the number of candidate predictors increases very quickly as the number of inputs increases, and a method to select the predictors or reduce the effective number of parameters is needed.

Some remarks on *level one* transformations. Observe that *Level one* transformations include pairwise interactions of original variables, and second these transformations rule out the appearance of further unknown parameters inside ψ because that may result in non-concavity of the loss function. This ensures the possibility to use Ordinary Least Squares estimations, and avoid more involved estimations procedures.

Table 3.1: Number of level 1 transforms as a function of the original predictors p

p	ψ_t	$\psi(\psi_t)$
1	5	61
2	13	365
3	25	1301
4	41	3445
5	61	7565
6	85	14621
7	113	25765
8	145	42341
9	181	65885
10	221	98125

Besides these two very important advantages that increase flexibility, there are also some drawbacks. For example some of the *level one* transformations may not be defined for certain values in the domain of the predictors as is the case of inverse transformations which are not defined when the predictor assumes a zero value or is a binary variables which is coded with 0 and 1 values. In this special case it is convenient to allow binary variables to enter the regression equation without any prior transformations (in which case their associated coefficients would represent group-specific constants) or just let them interact with other variables (in which case their associated coefficients would deliver group-specific slopes). It is also important to remark that some transformations can generate substantial collinearity between the linear and nonlinear functions. As an example, consider a simple case of a polynomial parametrization $m(X_t) = c + X_t + X_t^2$ where X_t consist just of one predictor. Squared terms are common in economics, for example, age and the square of age often enter in labor force participation models. It can be shown that the collinearity between X_t and X_t^2 is 0 when $\mathbb{E}(X_t) = 0$ but dramatically increases to high collinearity as $\mathbb{E}(X_t)$ increases. One possible solution in order to reduce such effects is by standardizing the predictors prior and after their transformations. Standardization is necessary when performing ridge estimation, since the resulting coefficients are not scale invariant. For example if the scales used to express the individual predictors and their transformations are changed, then the ridge coefficients do not change inversely proportional to the changes in the variable scales. Observe that in this context ridge estimation may result beneficial for reducing the effect of ill-posed problems due to collinearity, and although these techniques deliver biased estimations of the parameters the out of sample predictive ability can be substantially improved.

3.1.2 Approximation by using Basis Expansions

Approximation using superposition of functions has existed since the early 1800's, when Joseph Fourier discovered that he could superpose sines and cosines to represent more complicated periodic functions, but there are many other possibilities. The idea is quite simple and consist in approximating a regression surface by a superposition of functions which are called *basis functions*, that is:

$$n(X_t) = \sum_{q}^{Q} \psi(X_t)\, \beta_q$$

where ψ are univariate mappings $\mathbb{R} \to \mathbb{R}$ (thus excluding pairwise operations on the predictors). A simple example is a polynomial series expansion. However, in multivariate settings polynomial series expansion are more difficult to apply since, we have to define a multivariate algebraic polynomial for all elements p of the predictor vector X_t with degree dependent on Q. Similar to algebraic polynomials are Bernstein, Chebychev, or Hermite polynomials. Other important and powerful extensions of the algebraic polynomials are the classes of piecewise polynomials and splines (Wahba & Wold 1975, Wahba 1990). Well-known types of splines are linear splines, cubic splines, and B-splines.

3.1.3 Approximation using Artificial Neural Networks

Another popular class of approximating functions is represented by Artificial Neural Networks (ANN). Historically ANN's were models inspired by the structure and behavior of biological neurons and the nervous system, but after this point of inspirations all resemblance to biological system ceases. The use of ANN's is quite popular not only for regression, but also for classification and discrimination. There is a huge variety of different architectures of ANN's today. See Kuan & White (1994) for a discussion on ANN's from an econometric perspective, Trippi & Turban (1992) for application in Finance and Cheng & Titterington (1994) about their use in statistics. The fashion for neural networks, which started in the mid 80's has given rise to new names for concepts already familiar to statisticians. Some examples are reported in Table 3.2. Some of such terms already appeared in the previous chapters. In the following they will be used interchangeably, although we will generally use a statistical jargon. Feed-forward Neural Networks are perhaps

Table 3.2: Equivalent terms in Neural Networks and Statistics

Statistics	**Neural Networks / Engineering**
Specification	Network Architecture
Estimation	Learning
Regression	Supervised learning
Interpolation	Generalization
Estimation Sample	Training Set
Predictors	Inputs
Dependent variables	Outputs
Parameter shrinkage/ ridge regression	Weight Decay
Non-linear transformation	Activation/Squashing/Ridge function
Derived features	Nodes

the most popular type of ANN's in use today. Their mathematical structure is quite simple and is similar to an additive basis expansion considered above:

$$n(X_t, \theta) = \sum_{q=1}^{Q} \psi(X_t, \Gamma_q)\beta_q \tag{3.1.3}$$

where the Q terms of the sum are usually called neurons or *nodes* and ψ are usually multivariate mappings $\mathbb{R}^k \to \mathbb{R}$, where k is the number of the inputs. In general there is no need to use all available k inputs for a particular ψ. Instead we could use just a subset of them which in turn is a *selection problem* we might avoid as we shall see in chapter 6. Here Γ is a set of parameters which determines shifts and variations in directions of ψ. In the ANN literature ψ is called *activation*, *squashing function* or *ridge function* and is usually differentiable, bounded and monotone. Any cumulative distribution function could be used as squashing function but a prevailing choice is the logistic function $\Lambda(z) = 1/(1 + e^{-z})$. Differentiability is used to solve the problem of non-linear optimization to estimate Γ while the properties of ψ as a squashing function enables the ANN to be a *universal approximator* of any Borel measurable function from one finite dimensional space to another, regardless the input dimension, the norm metric considered and provided that Q is sufficiently large, as demonstrated by Hornik et al. (1989). This universal approximation result justifies the use of ANN approximation and explains its success. A schematic of a single hidden layer, feed-forward neural network is presented in figure 3.1.

The logistic function is probably one of the most popular basis functions in the ANN literature. Nonetheless there is a wide range of choices available for the definition of the basis functions ψ. Given that our primary objective is to obtain as good an

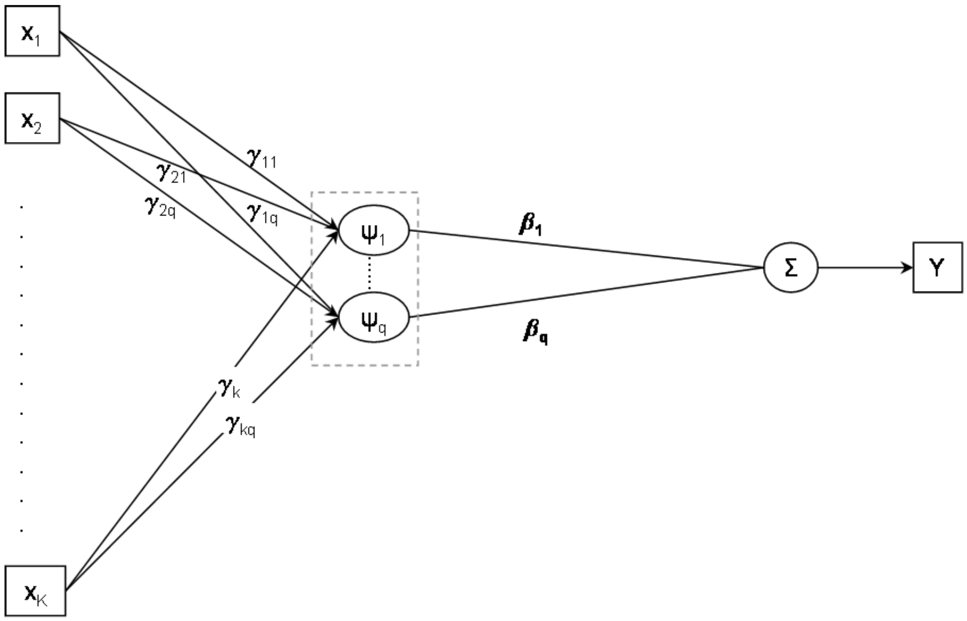


Figure 3.1: Single hidden layer, feed-forward neural network

approximation to $\mathbb{E}(Y_t|X_t)$ as possible, besides the more traditional logistic function we also consider other two powerful approximation methods which are Radial Basis Functions (Powell 1987, Lendasse, Lee, de Bodt, Wertz & Verleysen 2002) and Ridgelets (Candès 1998), which we will discuss briefly below.

Radial Basis Functions Radial Basis Functions (RBF) are a special class of functions used in the ANN literature. Their characteristic feature is that their response decreases (or increases) monotonically with distance from a central point defined in the input space. The center, the distance scale, and the precise shape of the radial function are parameters of the model, all fixed if it is linear. The radial basis functional form arises by taking:

$$\psi(X_t, \Gamma) = \exp[p_2(X_t, \Gamma)]$$

where p_2 is a polynomial of (at most) degree 2 in X with coefficients $\Gamma = \{\gamma_1, \gamma_2\}$. Here γ_1 represents a centering vector and γ_2 is a $p \times p$ symmetric positive semi-definite matrix which scales departures of X_t from γ_1. The $p_2(X_t, \Gamma)$ is restricted to have the form $p_2(X_t, \Gamma) = -.5(X_t - \gamma_1')'\gamma_2^{-1}(X_t - \gamma_1')$. Notice that ψ results proportional to a multivariate normal density with mean vector γ_0 and γ_1 a suitable generalized inverse of a given covariance matrix. Thus RBF are linear combinations of multivariate densities, accommodating a mixture of densities as a special case.

Ridgelets Ridgelets are a class of powerful functions that represent an extension of wavelets to the multivariate case. Ridgelets arise by taking

$$\psi(X_t, \Gamma) = \frac{1}{\sqrt{\gamma_1}} f\left(\frac{X_t'\gamma_2 - \gamma_0}{\gamma_1}\right)$$

with set of parameters $\Gamma = \{\gamma_0, \gamma_1, \gamma_2\}$ where $\gamma_0, \gamma_1 \in \mathbb{R}$ and γ_2 is a direction vector on the unit sphere in $\mathbb{R}^k$. The function f must be chosen such that ψ satisfies the following admissibility condition of vanishing moments, that is:

$$\int \psi(X)X^j dX = 0, \quad j = 1, \ldots, p \tag{3.1.4}$$

As an example of a function satisfying such condition we may take the $j-$th derivative of the standard normal density $\phi(z) = \frac{1}{\sqrt{2\pi}} \exp\left(-z^2/2\right)$ which is admissible for any $\psi(z) = D^h\phi$ where $h = p/2$ and $D = d/dz$ (White 2006). Thus, a ridgelet with $p = 4$ inputs, arises by taking $h = p/2$, that is, the second derivative of a standard normal density function. Figure 3.2 reports the plot of the first three derivatives of the standard normal. Notice that from a practical point of view, the admissibility condition implies that ψ oscillates, has zero average value, zero average slope, etc. We are motivated to use Ridgelets because, as Candès (1999) shows, they turn out to be optimal for representing otherwise smooth multivariate functions that may exhibit linear singularities[1]. This is in sharp contrast to Fourier series, which can be badly behaved in the presence of singularities. In the univariate case we could overcome this problem using wavelets, but their ability to deal with linear singularities in higher dimensions doesn't hold. Candès (2003) provides an extensive discussion of the properties of ridgelet regression estimators, and, in particular proposes regularization methods by thresholding the coefficients from a ridgelet regression[2]. In particular, Candès (2003) discusses the superiority in multivariate contexts of ridgelet methods to kernel smoothing and wavelet thresholding methods.

[1] When a smooth function $f(x)$ has a linear singularity at $x = c$ the first derivative at $x = c$ is not defined. As an example consider a mutilated gaussian distribution defined on the interval $[0, \infty)$ which is singular at the origin but is differentiable elsewhere.

[2] Thresholding refers to setting to zero some estimated coefficients whose magnitude does not exceed some pre-specified value. A typical choice is $\hat{\sigma}\sqrt{2\log N}$ where $\hat{\sigma}$ is an estimate of the standard deviation of the noise. See Hastie et al. (2001, pp.154) for an intuitive justification of this.

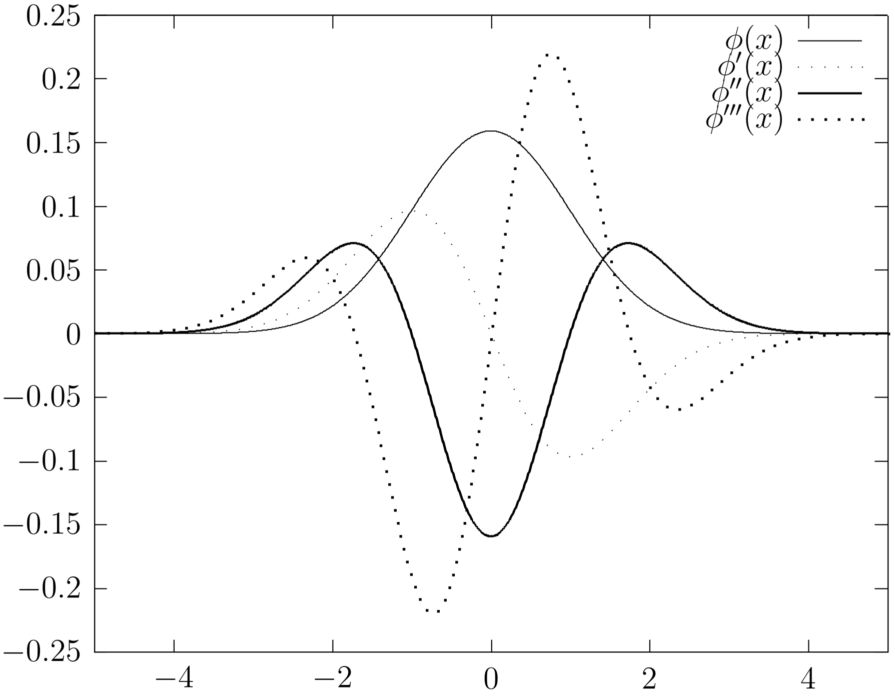


Figure 3.2: An example of three different ridgelet transforms which arise by taking the $j - th$, $j = 1, 2, 3$ derivative of a standard normal density. These functions ensure that admissibility condition (eq.3.1.4) is satisfied which traduces in a oscillatory behavior with zero mean, zero average slope, etc.

3.1.4 Generically Comprehensive Revealing functions

The main differences of ANN's with respect to series expansions and level one transforms, is that the derived feature vector ψ results from applying a nonlinear transformation to a linear combination of the original predictors. This involves the estimation of an unknown parameter set $\theta = \{\beta, \Gamma\}$, where β and Γ can be estimated simultaneously by using some non-linear optimization method. Barron (1993) shows that ANN are efficient functional forms for approximating multidimensional functions, but this property doesn't always compensate for the operational difficulties of fitting them. Jones (1997), Vu (1998) show that it is impossible to design algorithms running in polynomial time that would produce accurate estimates of the unknown parameters in ANN's. It turns out, however, that by suitably choosing ψ, it is possible to retain the flexibility of ANN's without requiring the Γ's to be free parameters. In this circumstance, estimation problem reduces to the estimation of the linear regression parameters (β's) by optimizing a well defined convex objective function, with a unique minima resulting from the minimization of the residual sum of the squares.

A special class of ψ functions that satisfy such condition are Generically Comprehensive Revealing Functions. This family of functions have been proposed by Stinchcombe & White (2000) extending the results of Bierens (1990) on consistent specification tests. In this context we refer to consistency as the property of having power against any arbitrary model misspecification. Put succinctly the results of Bierens imply that given a random variable ε_t and a random vector X_t, under some general conditions $\mathbb{E}(\varepsilon_t|X_t) \neq 0$ with non zero probability implies $\mathbb{E}[\exp(X_t'\gamma)\ \varepsilon_t] \neq 0$ for almost every $\gamma \in \Gamma$ a non empty compact set. Here since we define $\varepsilon_t = Y_t - m(X_t', \theta)$:

$$\mathbb{E}(\varepsilon_t|X_t) = \mathbb{E}[(Y_t - m(X_t', \theta))|X_t)] = \mu(X_t) - m(X_t, \theta) \neq 0$$

then for almost every $\gamma \in \Gamma$ we have:

$$\mathbb{E}[\exp(X_t'\gamma)(Y_t - m(X_t, \theta^*))] \neq 0$$

In other words if a parameterization m is misspecified, the residuals will be correlated with $\exp(X_t'\gamma)$ for any γ. Stinchcombe & White (2000) show that this result holds more in general for a class of functions ψ which they call *Generically Comprehensive Revealing* (GCR) functions. The *revealing* property stems from the fact that they can reveal a model misspecification of any form, while the *generic* property derives from the fact that virtually any γ will reveal the misspecification. An important class of such functions which is GCR is the class of non-polynomial real analytic functions.

These functions are infinitely differentiable such that the Taylor series at any point X_0 in its domain is convergent for X close enough to X_0 and its value equals $m(X_t)$. It follows that we can choose any arbitrary parameter value for $\gamma \in \Gamma$, and as we will see this powerful result can be used to construct ANN architectures without estimating γ as free parameters, but fixing them a priori in some way. For model building and approximation purposes, we could generate a collection of γ parameters at random and choose among them just those that deliver transformations of the inputs which are most correlated with the target variable. A remarkable fact is that among the functions belonging to the class of GCR functions there is the exponential family. Thus, any exponential function (like a normal density) may be used to build GCR functions with the properties defined above. It follows that the logistic, the radial basis and the ridgelet function are all GCR. Following White (2006), we can take advantage of these results to obtain flexible parameterizations which are nonlinear in predictors but preserve linearity in parameters. For this purpose let's define the parametrization of a single hidden layer ANN as:

$$m(X_t, \theta) = \sum_{q=1}^{Q} \psi(X_t' \gamma_q) \beta_q$$

where ψ is GCR. Now define the residual term:

$$\hat{\varepsilon}_t = Y_t - m(X_t, \hat{\theta})$$

It follows that if with non-zero probability $\mu(X_t) - m(X_t, \theta) \neq 0$ then for almost every $\gamma \in \Gamma$ we have $\mathbb{E}[\psi(X_t'\gamma)\ \varepsilon_t] \neq 0$. Given that Γ is compact we can pick γ_{q+1} such that:

$$|\text{corr}(\psi(X_t'\gamma_{q+1}), \varepsilon_t)| \geq |\ \text{corr}(\psi(X_t'\gamma_q), \varepsilon_t)| \qquad \forall \gamma \in \Gamma$$

where corr(·,·) denotes the correlation of the indicated variables. This suggests a process of adding nodes in a stepwise manner, stopping when at the $i-$th iteration $|\text{corr}(\psi(X_t'\gamma_{q+i}), \varepsilon_t)| < \delta$, where δ is any arbitrary small number, or when any other stopping rule has been reached (eg. minimum AIC).

3.1.5 Final remarks

As we anticipated, the choice of the basis functions depends on their approximation properties but also on ones prior knowledge about μ, that is, the data. Unfortunately there are no fast prescriptions about how to choose the ψ's, especially in the circumstances commonly faced by economists, where one may have little prior information about the form of the conditional mean function and its smoothness.

Knowing the smoothness could help in choosing the most appropriate approximating function, which should be the one for which a minimum number of Q basis function is needed to approximate well our response. As a practical matter, then, it may make sense to consider a collection of different bases, and let the data drive us to the best choice. Such a collection of bases is often called a *library*. Given a base function ψ we expect better approximations to μ as the number of terms of the expansion increases, but at the same time given the limited amount of available data, we would like to use a small number of terms as possible, in order to achieve a parsimonious representation. As White (2006) points out, this suggests not to force the inclusion of the terms in a strict order (e.g. zero order polynomial first, followed by first order polynomials and son on). Instead we should just consider those terms useful in approximating μ. Parameterizations of this form are denominated *highly non-linear approximations*, as not only there is a nonlinearity associated with choosing Q basis functions, but there is the further choice of the basis itself or of the elements of the library (eg. mixing *level one* transforms and ANN's with logistic squashing functions simultaneously). The choice of the inputs is also another non-trivial aspect to consider in practical ANN applications. Inputs should be selected carefully prior to the construction of the network. In theory one could choose the inputs based on prior knowledge, but this information is seldom available especially in empirical Economic studies. Another aspect to consider is the choice of the number Q of basis functions. In practical applications this means that the architecture of the network is not known in advance. Indeed any asymptotic loss efficient method can be used when choosing among competing architectures (specifications), and the strategies to generate can be automated easily as we will see later.

3.2 Conclusions

In this chapter we reviewed different methods which may prove to be useful for approximating an unknown data generating process. The main conclusion is that there is no need for parameterizations which are non-linear in the parameters. Since they suffer that the likelihood function may have several local minima, we prefer to include non-linearities using simple transformations of the inputs or taking advantage of Generically Comprehensive Revealing functions as will be evident in chapter 6. Libraries of different approximating functions may be useful where, as in economic data occurs, the degree of smoothness of the target variable is usually unknown. Selection of the inputs and the choice of the number Q of basis functions remains an important practical issues to solve in order to implement automatic modeling and approximation procedures. We shall see in the next chapters how this can be implemented in practice.

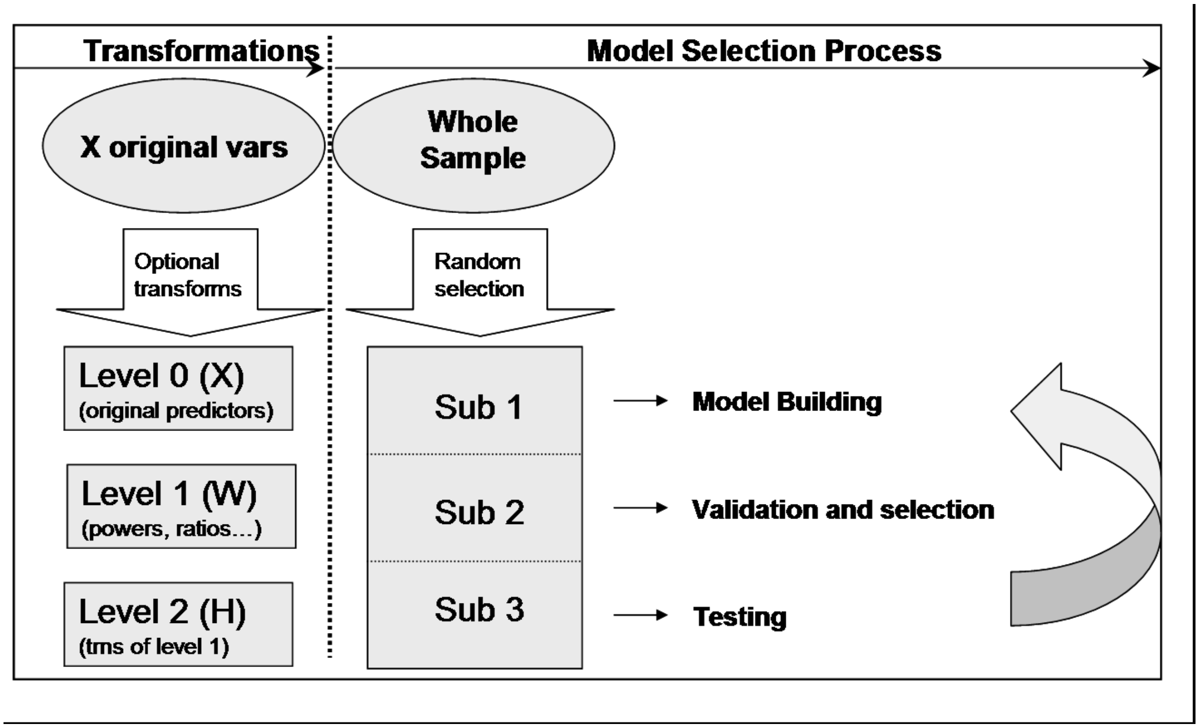
Chapter 4

Automatic Model Selection

In previous chapters we reviewed what desirable properties a model selection criteria should have and what kind of parameterizations approach we can use in order to approximate a given target variable. Libraries or dictionaries of basis functions and in particular ANN's may prove useful for approximation. *Generically Comprehensive Revealing* functions provide a convenient shortcut to avoid the computational burden of non-linear optimization. Model selection criteria provide stopping rules and aim to avoid over-fitting and select a parameterization with good "generalization abilities". Since in our context there is a potentially unlimited number of transformations and basis function among which to choose as candidate predictors, Automatic Model Selection Algorithms become essential in order to choose a parametric formulation for forecasting purposes. In this context, an heuristic approach is adopted, where the data is the final arbiter of how well a particular approach works, unless external information or some theory provides some guidance about the most convenient parameterization.

Despite the controversy surrounding many model selection strategies, in the last decade different approaches with "'good" properties have been developed. The automatization of selection procedure has gained more attention by the scientific community and many methods, some based on regularization techniques (the Non-negative Garrote (Breiman 1995), the LASSO (Tibshirani 1996*b*)) other based on subset selection, (Gets (Hendry & Krolzig 2001), RETINA (Pérez-Amaral et al. 2003)) have gained increasing popularity since some of them are readily available in many commercial statistical packages. In the following we consider a method proposed recently by Pérez-Amaral et al. (2003) (PAGW from now on) called RETINA. (RElevant Transformations of the Input Network Approach), which is based on earlier work by White (1998). We will review in this chapter its philosophy and its properties and compare it with other methods as an automatic model selection tool.

Figure 4.1: The RETINA algorithm



.

In addition we present a software implementation called RETINA Winpack, which has been designed for applied researchers. The RETINA method is of special interest here since it was designed to identify a parsimonious set of predictors useful for forecasting purposes and approximate the conditional mean $\mu(X)$ of an unknown functional form.

It is worth to clarify that this chapter is concerned with the mechanics of producing candidate specifications, not with their statistical properties which can be derived based on other considerations. The discussion that follows applies to the case when the number of predictors P is less than the number of observations N, although ridge regression or variable grouping may be considered when the number of candidate predictors exceeds N. Our problem here is consider automatic methods which starting from a set of predictors find a subset $k < P$ which has a satisfactory out-of sample predictive ability. We shall now discuss in more detail how RETINA deals with these important aspects in empirical modeling.

4.1 The RETINA algorithm

We next describe the RETINA algorithm from a *high level* perspective reported in figure 4.1, while a much more detailed description can be found in table 4.1 at page 49. RETINA is a procedure which has been developed considering simultaneously two problems discussed so far:

- the *approximation problem*, because it provides some flexibility to accommodate possible non-linearities of the response.
- the *selection problem* which allows to automatically select a parsimonious subset of predictors among a potentially very large set of candidates.

A third aspect to consider is relative to automatization and practical implementation of the rules given by the procedure. We shall now discuss in more detail how RETINA deals with these aspects.

Approximation. RETINA deals with the approximation problem by using simple non-linear transformations of the predictors as those discussed earlier in section 3.1.1 called *level one* transformations (eq.3.1.2). Previous to any specification search the algorithm generates these transforms, allowing certain degree of flexibility by expanding the predictor set embodying both interactions and non-linearities. Interactions allow to capture local curvature and subset behavior as well as non-linearities of the form of squares, inverses of the original inputs which may substitute more complicated non-linear functions. Thus specification is linear in the parameters, but non-linear in the inputs. This avoids non-linear optimization of the objective loss function where different local minima may be present. Summarizing the specifications that handles the original RETINA algorithm, as proposed by PAGW, is of the form:

$$\mathbb{E}(Y_t|X_t) = \sum_k^K \psi_k(X_t)\beta_k \quad \psi(X_t) = X_{it}^{\alpha_1} X_{jt}^{\alpha_2} \quad i,j = 1,2,\dots,p \quad \alpha_1,\alpha_2 = -1,0,1 \tag{4.1.1}$$

Recall that this functional form includes as a special case the specification with only original predictors. A possible extension is to consider higher order *level one* transformations of the form already discussed in section 3.1.1, but there are many other possibilities as we shall see later. As we already pointed out undertaking such non-linear transformations of the inputs, may generate substantial collinearity between the untransformed and the transformed ones. Collinearity artificially generated by level one transformations may be limited by *prior* and/or *posterior* variable standardization. Nonetheless as we will see later in this section RETINA is able to prevent such problems based on a selective search that explicitly controls for collinearity among the predictors.

Selection. RETINA deals with the selection problem considering the following: 1) a method for specifications search using a collinearity index, 2) a method for out of sample performance evaluation, and 3) a method for further reduction and final specification selection. To implement these methods, the procedure splits the sample into three disjoint subsamples:

1. The building sub-sample: used for the specification search.

2. The validation sub-sample: used for out-of sample validation and re-estimation of candidate parameterizations obtained as at point 1.

3. The testing subsample: used for out-of-sample testing and, eventually, further specification reduction.

The main motivation for using three sub-samples relies on the fact that we want to avoid "too optimistical predictions" by using the same data set to 1) build a set of candidate models, 2) estimate their parameters and finally 3) evaluate their predictive ability and possibly adopt an even more parsimonious representation than the current specification. Ideally we would like to have a new fresh data set available for each of these steps in order to avoid selection biases (Miller 2002) due to the fact that we use the same data set to build and choose a specification.

As we anticipated, RETINA starts by generating transformations $\psi(X_t)$ of the candidate predictors and splitting the whole data set into three sub-samples of approximate equal size[1], say $Sub1$, $Sub2$ and $Sub3$.

On subsample $Sub1$, a set of specifications is obtained by ordering each transformation on the basis of the univariate absolute correlation with the response. This serves as the basis to build candidate specifications. The first specification considered always includes just a constant. Then the predictors are included in the specification in a stepwise manner following their rank order, only if the R^2 of its regression

[1] In the case of time series data, each subsample is chosen such that all the observations are contiguous within the whole data set. In the case of cross-section data, we can pick a subsample by selecting randomly in the whole sample.

Table 4.1: The prototype Retina algorithm based on PAGW

Stage 0 – Preliminary

1. Data building and sorting
   (a) Generate the set of transformed variables $\psi(X_t) = \{W_{1t}, \ldots, W_{kt}\}$.
   (b) Divide the sample into three sub-samples.

Stage I – Isolating a *candidate* model

2. Using data on the first sub-sample:
   (a) order the variables in $\psi_\lambda(X)$ according to their (absolute) sample correlation with the dependent variable in the first sub-sample alone. Let $W_{(1)}$ be the variable with the largest absolute correlation with $Y, W_{(2)}$ be the second most correlated, and so on.
   (b) Consider various sets of regressors all of which include a constant and $W_{(1)}$: each set of regressors $\psi_\lambda(X)$ is indexed by a *collinearity threshold* $\lambda \in [0, 1]$ and is built by including $W_{(j)}$ $(j = 2, \ldots, k)$ in $\psi_\lambda(X)$ if the R^2 of the regression of $W_{(j)}$ on the variables already included in the model is $\leq \lambda$.
   (c) The number of sets of regressors is controlled by the number of values of λ between 0 and 1 chosen, say, m.
3. Using Data both on the first and second Sub-sample:
   (a) Estimate each model by regressing Y on each set of regressors $\psi_\lambda(X)$ using the data on the first sub-sample only and compute an out-of-sample prediction criterion (the cross-validated mean squared prediction error) using the data on the second sub-sample only. This involves the estimation of m models.
   (b) Select a "candidate" model as the one corresponding to the best out-of-sample performance $\psi^*_\lambda(X)$.

Stage II – Search Strategy

4. Using data from both the second and third Sub-sample:
   (a) Search for a more parsimonious model: estimate all models including a constant and all the regressors in $\psi^*_\lambda(X)$ one at a time in the order they were originally produced by procedure sub 2.a, this time on the basis of the absolute correlations of the second sub-sample or of the correlations of the first and the second sub-sample together.
   (b) Perform an evaluation of the models out of sample (using the data on the third sub-sample) calculating a performance measure (the cross-validated mean squared prediction error, possibly augmented by a penalty term for the number of parameters in the model)

Stage III – Model Selection

5. Repeat Stage I and stage II changing the order of the sub-samples. Produce a candidate model for each sub-sample ordering.
6. Select the model which has the best performance over the whole sample using AIC or any other Asymptotic Loss efficient selection procedure.

against all the remaining inputs is below a given value of $\lambda \in [0, 1]$. The role of λ is crucial in the specification building stage since it represents the thresholds parameter that controls the amount of collinearity allowed among the predictors. Choosing m thresholds values varying between 0 and 1 RETINA will deliver a sequence of m specifications.

Next, we step into the validation stage: all m specifications obtained so far, are compared in terms of their out-of-sample predictive ability on the second subsample $Sub2$, the *validation sub-sample*. Only the specification with the best accuracy in predicting on the *validation sub-sample* is retained. The winning specification is indexed by λ^*.

Finally the testing stage is performed. Given the winning specification indexed by λ^*, we re-estimate it along with all its nested specifications based on their absolute correlation ranking with the response. We choose the one which has the best out-of sample performance on the test sub sample $Sub3$. We call this a *final* specification. At the end of the whole process, in order to gain efficiency, the final specification is re-estimated using the whole sample and the model selection process ends.

Iteration Observe that the final specification so far, is conditional on the order by which the sub-samples are fed into the modeling process. In the above description we used $Sub1$ as the building sample, $Sub2$ as the validation sample and $Sub3$ as the testing sample. However we may consider any possible sub-sample ordering and, as an example, use $Sub2$ for the building stage, $Sub1$ for the validation stage and $Sub3$ for the testing stage. In total, given three sub-samples, there are six possible orderings (see table 4.2), which may deliver six specifications that differ or not from each other. The final choice may then be decided by using any Asymptotic Loss Efficient model selection criteria, like the AIC, AICC and Mallows C_p. From another point of view, RETINA allows us to examine the structure of several “good” fitting specifications rather than restricting the attention on a single best. Producing a number of candidates is also useful to evaluate selection uncertainty, which is conditional on several aspects of the data such as:

1. Small sample size.

2. Possible presence of outliers.

3. Clusters of heterogeneity.

Table 4.2: Subsample rotations in the RETINA procedure.

Ordering	Building	Validation	Testing
1	$Sub1$	$Sub2$	$Sub3$
2	$Sub1$	$Sub3$	$Sub2$
3	$Sub2$	$Sub3$	$Sub1$
4	$Sub2$	$Sub1$	$Sub3$
5	$Sub3$	$Sub1$	$Sub2$
6	$Sub3$	$Sub2$	$Sub1$

Having a small sample size is not under researcher's control, but as we will see in section 4.2, a new software implementation called RETINA Winpack is able to deal with outliers and/or identified cluster of heterogeneity which, in empirical applications, enhances the performance of the prototype RETINA algorithm described here.

4.1.1 Some simple examples

Example 1: Exponential target Here, we demonstrate RETINA's performance using a data set generated with an exponential function, and show that (i) it can recover the original function when exponential functions are included in the library of transformations, and (ii) that if we exclude exponentials from its list of transformations, it recovers the first few terms of a Taylor expansion of an exponential. We generated $T = 100$ of three uniformly distributed variables in the interval $[-2, 2]$ and the response was generated as:

$$Y = 3 + 5X_1 + e^{X_3} + \varepsilon \tag{4.1.2}$$

where the noise is Gaussian distributed $\sigma = 1$. Note that Y has no dependence on X_2 and we were interested in checking whether the variable selection would discard X_2 from the model. The Taylor expansion of an exponential function is particularly simple:

$$\exp(X) \approx 1 + X + \frac{1}{2}X^2 + \dots$$

Using just level one transformations the suggested specification is:

$$Y = 3.95 + 4.98X_1 + 1.44X_3 + 0.66X_3^2 \tag{4.1.3}$$

This is quite close to the original equation 4.1.2, with the exponential replaced be the first terms of the Taylor expansion. Notice that the coefficients are not exactly what would be calculated from the expansion; this is due the fact that RETINA provides a good fit using all the available input data, whereas a Taylor expansion is constructed so that it is most accurate around $X = 0$. Also note that the spurious variable X_2 is absent from the parameterization. Adding exponential transforms to the list of transformations we get the following specification:

$$Y = 3.03 + 4.99X_1 + 0.99e^{X_3} \tag{4.1.4}$$

Comparing equations 4.1.3 and 4.1.4, we observe that the selection method implemented in RETINA is able to derive the original equation to a very high accuracy from the data.

Example 2: Orthogonal predictors When predictors are orthogonal to each other the cost of search for a new specification can be easily assessed. Suppose the ranking in terms of univariate predictive ability of the j-th variable corresponds to the index j. That is, X_1 is the predictor most correlated with the response, X_2 is the second most correlated, and so on. Since all predictors are uncorrelated, none of them will fail the collinearity check, regardless the value assumed by the collinearity threshold λ. Once the predictors have been ranked based on their correlation with the response, p increasing in the number of parameters specifications $m_0, m_1, m_2, \dots, m_p$, are obtained for any value of $\lambda_1 < \lambda_2 < \cdots < \lambda_l$. The search process will always end with the full model. This is illustrated in table 4.3.

Example 3: Correlated predictors Here we consider a case in which there are three predictors, two of them being jointly highly predictive but at the same time strongly correlated. This example is taken from Miller (2002) and is often proposed to illustrate the behavior of *forward search* algorithms. Here we use it to illustrate the behavior of the RETINA specification search algorithm and to understand more in detail how it works. Let's generate some data from the process $Y_t = X_{1t} - X_{2t}$ and correlation matrix:

$$R = \begin{array}{r|rrrr} X_1 & 1.0000 & & & \\ X_2 & .9999 & 1.0000 & & \\ X_3 & .0000 & -0.0007 & 1.0000 & \\ Y & .0000 & -0.0016 & 0.4472 & 1.0000 \end{array}$$

There are three predictors and $Y = X_1 - X_2$, but it is X_3 the predictor most correlated with the response. This may be well the case in which one considers the log of a predictor expressed in *per-capita* terms or in a time series setting the log taken with respect to the ratio between a predictor and its own lagged value, which represents the relative variation of that predictor over time. From the correlation

Table 4.3: RETINA steps in the building sample for three orthogonal predictors X_1, X_2, X_3 sorted in terms of predictive ability with the response and $\lambda_1 = 0.1 < \lambda_2 = 0.2 < \cdots < \lambda_{10} = 1.0$

λvalue	Search Iteration	Current Spec.	Candidate	CI	Included	Intermediate Spec.	Candidate Spec.
0.1	1	1	X_1	.0000	Yes	$1, X_1$	
	2	$1, X_1$	X_2	.0000	Yes	$1, X_1, X_2$	
	3	$1, X_1, X_2$	X_3	.0000	Yes	$1, X_1, X_2, X_3$	$1, X_1, X_2, X_3$
0.2	1	1	X_1	.0000	Yes	$1, X_1$	
	2	$1, X_1$	X_2	.0000	Yes	$1, X_1, X_2$	
	3	$1, X_1, X_2$	X_3	.0000	Yes	$1, X_1, X_2, X_3$	$1, X_1, X_2, X_3$
...	...	...	...	...	...	...	
	...	...	...	...	...	...	
	...	...	...	...	...	...	...
1.0	1	1	X_1	.0000	Yes	$1, X_1,$	
	2	$1, X_1$	X_2	.0000	Yes	$1, X_1, X_2$	
	3	$1, X_1, X_2$	X_3	.0000	Yes	$1, X_1, X_2, X_3$	$1, X_1, X_2, X_3$

Notes: The specification search of RETINA provides always the same specification $1, X_1, X_2, X_3$ for any value of λ.

matrix R above X_1 and X_2 are almost perfectly correlated. Thus X_3 is included first in the regression equation. Then for any value of the collinearity threshold λ, X_2 is included second and X_1 is included last. It is instructive to see that the best model which includes only X_1 and X_2 is not detected by the RETINA search procedure, however a "good" model given by X_3, X_2 and X_1 is still considered. All specification search steps are reported in table 4.4.

4.2 Software implementations of RETINA: a comparison

RETINA is available in different versions as stand-alone application and as Matlab and Gauss codes. The stand-alone application, RETINA Winpack for Windows, is available upon request from the author. Another Matlab implementation is due to Brownlees (2005). The Winpack version of RETINA is especially designed to be used in an initial and exploratory stage of the specification search when considering real data sets. RETINA Winpack uses the prototype PAGW's RETINA as the basis, but makes it applicable to real world problems by including specific customization features that are necessary in these circumstances. In table 4.5, we

Table 4.4: RETINA steps in the building sample for the case in which two highly correlated predictors together have high predictive value

		Current				Intermediate	Candidate
λvalue	Iteration	Spec.	Candidate	CI	Included	Spec.	Spec.
	1	1	X_3	.0000	Yes	$1, X_3$	
0.1	2	$1, X_3$	X_2	.0007	Yes	$1, X_3, X_2$	
	3	$1, X_3, X_2$	X_1	.9999	No	$1, X_3, X_2$	$1, X_3, X_2$
	1	1	X_3	.0000	Yes	$1, X_3$	
0.2	2	$1, X_3$	X_2	.0007	Yes	$1, X_3, X_2$	
	3	$1, X_3, X_2$	X_1	.9999	No	$1, X_3, X_2$	$1, X_3, X_2$
	...	...	...	...	...	...	
...	...	...	...	...	...	...	
	...	...	...	...	...	...	...
	1	1	X_3	.0000	Yes	$1, X_3,$	
1.0	2	$1, X_3$	X_2	.0007	Yes	$1, X_3, X_2$	
	3	$1, X_3, X_2$	X_1	.9999	Yes	$1, X_3, X_2, X_1$	$1, X_3, X_2, X_1$

Notes: The specification search of RETINA provides two possible specifications $1, X_3, X_2$ and $1, X_1, X_2, X_3$ given the default λ-grid used by PAGW. The algorithm is yet able to find a good parameterization given by the full specification.

point out which are the main contributions of RETINA Winpack with respect to the earliest version of PAGW's RETINA as well as RETINA for Matlab (from now on RETINA MATLAB).

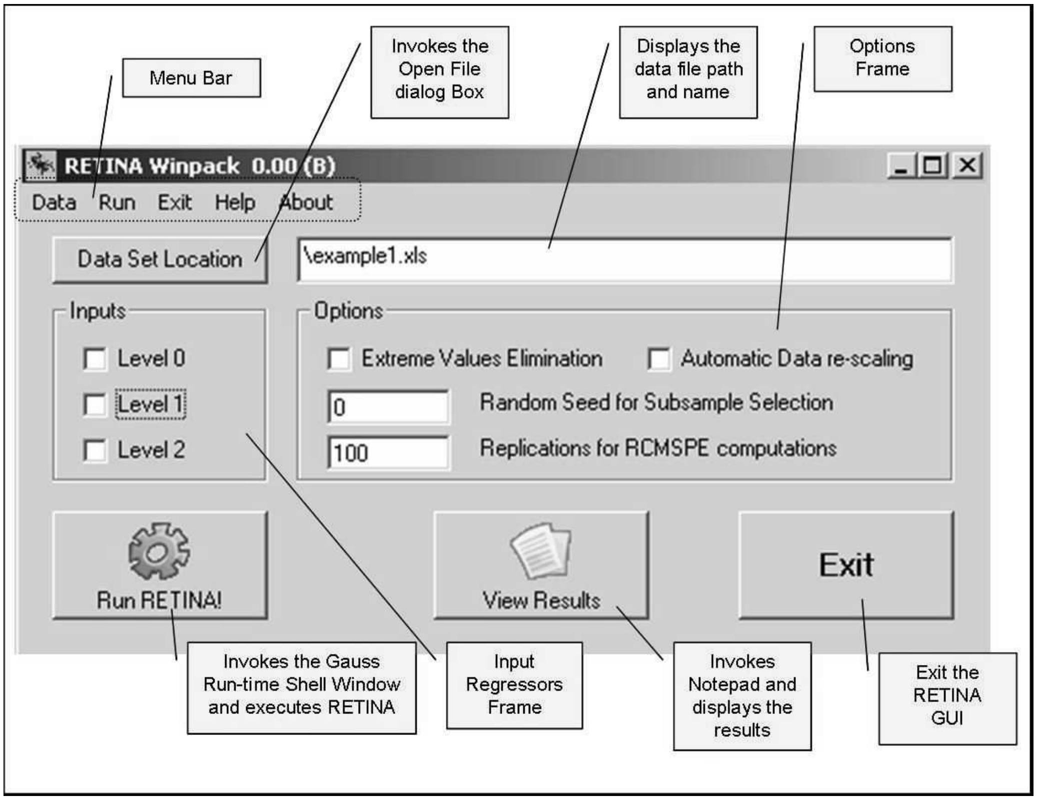
Purpose. First of all, a basic distinction concerns the purpose of each implementation: the PAGW's RETINA (from now on RETINA PAGW) was a prototype intended to be used primarily for Monte Carlo simulations. The Winpack is primarily intended to be used for exploratory analysis on real data sets. The RETINA MATLAB implementation can be adapted to both situations. Nonetheless the main advantage for less experienced users in using RETINA Winpack is the Graphical User Interface (GUI) illustrated in figure 4.2 that simplifies all operations involved in the application of the procedure. Also, RETINA Winpack is the only version which has been carefully documented so far. An online user manual is available from inside the GUI: it includes installation instructions and is aimed to support the user in setting up the data, interpreting the result and reproduce some easy examples.

Input Data RETINA Winpack accepts the input data in Excel format, which is widely used in practice and doesn't require special knowledge from the user.

Table 4.5: Comparative features of PAGW's RETINA vs. RETINA Winpack and RETINA for Matlab.

	RETINA PAGW	**RETINA WINPACK**	**RETINA MATLAB**
Purpose	Prototype	Exploratory Specification Analysis of Real data	Exploratory Specification Analysis of Real and Artificial data
Dependency	Gauss 3 or higher	Gauss Runtime 6 or higher (free for non-commercial use)	Matlab 6 or higher
Delivery format	Source code	Executable (EXE)	Dynamic library (DLL)
OS platform	Depends on Gauss	Windows	Windows, Linux
Programming language	Gauss 3	Gauss 6 and Visual Basic	C
Graphical User Interface (GUI)	No	Yes	No
User Documentation	No	Yes	No
Data input	No, only internally generated data	Yes, from Microsoft Excel	Yes, importing as Matlab data set
Transformations	For continuous predictors	For continuous and categorical predictors.	For continuous predictors
User control over input transformations	No	Yes	Yes
Informative Output	Success rate over Monte Carlo experiments	Selected predictors, Summary statistics	Selected predictors, (no summary statistics)
Automatic Data scaling	No	Yes	No
Automatic Outliers detection	No	Yes	No
Union Model	No	Yes	No
Computational efficiency in repeated operations	No	Yes (sweeping)	Unknown.

Figure 4.2: The RETINA Winpack user interface


Menu Bar
Invokes the Open File dialog Box
Displays the data file path and name
Options Frame
RETINA Winpack 0.00 (B)
Data
Run
Exit
Help
About
Data Set Location
Inputs
Options
Run RETINA!
View Results
Exit
Invokes the Gauss Run-time Shell Window and executes RETINA
Input Regressors Frame
Invokes Notepad and displays the results
Exit the RETINA GUI

\example1.xls

□ Level 0

□ Level 1

□ Level 2

□ Extreme Values Elimination

□ Automatic Data re-scaling

0 Random Seed for Subsample Selection

100 Replications for RCMSPE computations

This feature is not present under the RETINA PAGW nor RETINA MATLAB although the latter may do so, it needs some little programming skills (that unexperienced users may not have) to import real data sets.

Pre-processing routines. Another relevant feature which is desirable in practical applications is the availability of data preprocessing routines which are able to detect outliers and automatically re-scale the data. As we have seen so far, re-scaling is important prior to data transformation as well as outlier detection. The procedure to detect outliers is the one proposed by Peña & Yohai (1999), which is especially adequate for regression problems in large data sets. These are unique features of the Winpack version which are documented and are not included in the other versions.

Transformations. Also, in applied research users may want to control the types of transformations of the inputs to be used, and additionally may want to distinguish binary/categorical from continuous inputs. This situation poses special problems since categorical data cannot be used to generate simple *level* *one* transformations. Categorical inputs usually reflect a specific group membership, such as gender, age, or the size (big, small) of a firm, its activity sector and so on. For these predictors, RETINA Winpack considers just interactions with continuous predictors which are built internally by the software. In particular, RETINA Winpack allows for automatic building of interactions between binary predictors and continuous predictors. This allows the user to easily extend the parameterization 4.1.1 to a more general one where the vector of coefficients β includes group specific constants and group specific slopes. Observe that in this case the resulting formulation is akin to an analysis of covariance and can be further extended to the case in which we also consider interactions between categorical inputs and genuine *level one* transformations. An empirical application of this feature is reported in the next chapter where the problem consists in predicting the telecommunications services usage by a sample of US firms. As we shall see the inclusion of categorical indicators is essential to obtain accurate forecasts, since they account for an intrinsic heterogeneity which is present among different types of firms. RETINA Winpack allows the user to control explicitly these features through "‘checkboxes" on its GUI. RETINA MATLAB allows some of these options through simple commands, but doesn't distinguish between categorical and continuous inputs.

The union specification Among others, an additional feature implemented in the Winpack version of RETINA is the *union model*. The *union model* summarizes the six final specifications and finds a subset specification. In practice this is obtained taking all the predictors selected by the previous six steps and creating a new (union) subset of predictors which are now considered as the starting inputs for a new selection round. The main difference with the previous steps is that here the inputs are only those already suggested. The specification search, the estimation and the validation are performed over the whole sample, and there is no sub-sample rotations involved. Another important distinction is that in this step, no controls for collinearity are performed. Instead we consider just the ordering of each predictor with the response in order to build a specifications sequence where the AIC index is tracked. The resulting specification is the one with the lowest AIC and is usually compared with the previous six ones.

Informative Output: RETINA Winpack always ends its selection process suggesting at most seven final specifications, but it may be the case that some of these parameterizations are coincident resulting in a lower number of alternative specifications. The hope is that the procedure delivers a low number of alternatives meaning that, regardless the starting subsample adopted, the specification delivered doesn't vary that much and the procedure is rather insensitive to the available sample features. Given multiple alternatives among the final models the user is faced with the problem of selecting one of these. In RETINA Winpack the user is provided with some useful statistics in order to take a decision on which specification to adopt. These are based on cross-validated PMSE and the AIC statistic. Alternatively, one may choose to consider all models simultaneously adopting a model averaging strategy. This allows to estimate unconditional prediction errors of all parameters (Burnham & Anderson 2002). Any model selection procedure discussed in section 2.3 could be adopted, but AIC seems to be justified in view of the fact that we do not assume the existence of any true DGP (μ) and because of its Asymptotic Loss Efficiency property discussed in section 2.6.

4.2.1 Computational considerations

In RETINA, as for other automated subset regression procedures, computational efficiency is an important issue. The execution time depends essentially on the chosen cardinality v of λ thresholds (which is under the user's control) and the number of predictors p which depends on the selected transformations and original inputs X_t. The specification search involves $v \times p$ OLS estimations, where p is the total number of predictors (including transformations). These estimations include those necessary to obtain the collinearity index at each λ-step. Efficient algorithms have the advantage of being faster, and fortunately there are computational shortcuts that prove to be useful in this context.

Let's briefly consider the magnitude of operations involved in a typical RETINA execution with six sub-sample rotation. For each subsample configuration let' say $sub1$,$sub2$ and $sub3$, we perform a specification search in $sub1$, then validate in $sub2$, Re-estimate in $sub1 + sub2$ the resulting specifications and test in $sub3$. Let's summarize the number of OLS computations as follows:

1. Each specification search depends on the cardinality v of the λ thresholds grid. At each λ_i one has to compute the necessary collinearity indexes in order to check the inclusion or exclusion condition of a candidate predictor. The number of OLS computations for the specification search is of order[2]

[2] It is of order $O(v \times (p-1))$ and not $O(v \times p)$ since we start the specification search always

$O(v \times (p-1))$ and depends on the correlation structure of the data and the given threshold value.

2. Given that the specification search ends providing at most v candidate specifications there are at most v OLS estimations to perform in order to test their predictive ability in sub-sample *sub*2.

3. For the test sample *sub*3: we consider the winner specification from the previous step and test it in sub-sample *sub*3 along with all its nested sub-models. This involves a number of OLS estimations which depend on the dimension of the selected model. In any case the order of OLS operations is at most of dimension p if all predictors are included in the specification obtained from previous steps.

The following rules are adopted in RETINA Winpack in order to avoid unnecessary OLS computations and minimize computational time:

Specification search: Compute all collinearity indexes just once and store their values in memory. At future iterations of the specification search, the collinearity indexes may be retrieved from the memory if necessary, and don't need to be computed again. As an example consider the case with orthogonal predictors. There are $p = 3$ predictors and $v = 10$ thresholds. In total we should perform $v \times (p-1) = 20$ OLS estimations in order to compute all collinearity indexes, but in this case, only 2 are necessary.

Sub-samples rotations: Even if there are six possible rotations of the sub-samples involved in the procedure, there is no need to perform the specification search six times. In fact results of the specification search will be the same for rotation $sub1 - sub2 - sub3$ as for $sub1 - sub3 - sub2$, where the first sub-sample, in this case *sub*1 is used for the specification search. This consideration may save a significant amount of computational time since the specification search is the most computing intensive stage of RETINA.

Validation and testing: As we have seen, once the specification search ends with a list of proposed specifications, we need to assess them in terms of their forecasting ability in the *validation* subsample and further in the *testing* subsample. These steps imply the estimation of many specifications, which may be time very consuming, especially when the $X'X$ moment matrix needs to

including the most correlated predictor with the response.

be inverted in presence of a large number of predictors. Fortunately there are many computational shortcuts that avoid inverting from scratch moment matrix $X'X$ when a new (old) predictor is included (removed) from the regression equation. Miller (2002) provides a very complete and exhaustive overview of these methods. They deliver a very efficient way to compute the statistics used in multiple regression[3] since only the moment matrix is needed to compute all statistics, there is no need to keep the whole raw data set in computer's memory. Second, the usefulness of such methods is evident when we want to estimate a large number of different regressions involving the same response variable but having different sets of predictors, as is our case. In virtue of its programming simplicity, RETINA Winpack uses the "sweeping" method (Dempster 1969) for the validation and testing stage. For a regression model:

$$Y_t = \beta_0 + \beta_1 X_{1t} + \beta_2 X_{2t} + \cdots + \beta_p X_{pt} + \varepsilon_t$$

the sweeping procedure starts with a moment matrix M. Let $\tilde{M}$ be the new matrix produced by sweeping on the $k-$th row and column of M. The elements of $\tilde{M}$ will be:

$$\begin{aligned}
\tilde{m}_{k,k} &= 1/m_{k,k} & \\
\tilde{m}_{i,k} &= m_{i,k}/m_{k,k} & \text{for } i \neq k \\
\tilde{m}_{k,j} &= m_{k,j}/m_{k,k} & \text{for } j \neq k \\
\tilde{m}_{i,j} &= (m_{i,j}m_{k,k} - m_{i,k}m_{k,j})/m_{k,k} & \text{for } i,j \neq k
\end{aligned}$$

If we define M as a partitioned matrix:

$$M = \begin{bmatrix} M_{XX} & M_{XY} \\ M_{YX} & M_{YY} \end{bmatrix}$$

where M_{XX} is the moment matrix of the predictors and M_{XY} is the vector of the moments between the predictor X's and the response Y while M_{YY} is the scalar associated with the moments of the response itself. After sweeping all

[3] Multiple correlation, residual variance, regression slopes, and standard errors of slopes, plus some other values.

the rows and columns of M_{XX} we have:

$$\tilde{M} = \begin{bmatrix} M_{XX}^{-1} & -M_{XX}^{-1}M_{XY} \\ \\ M_{YX}M_{XX}^{-1} & M_{YY} - M_{YX}M_{XX}^{-1}M_{XY} \end{bmatrix}$$

The term $(M_{YX}M_{XX}^{-1})$ include the coefficients β of the regression and $M_{YY} - M_{YX}M_{XX}^{-1}M_{XY}$ represents the residual error of the variance. An example of sweeping can be found in table 4.6. Notice that sweeping is a reversible operation, thus sweeping can be used to include/exclude a variable from the estimated equation.

The main drawback of sweeping is that rounding error can accumulate over many estimations. One way to check on this is to recompute backwards all the steps involved, since sweeping is reversible – to get back to a moment matrix that will differ from the original one only due to rounding error, and then see how much rounding error has accumulated. In summary using *sweeping* or any other moment matrix up/down-dating method, we avoid computations of M^{-1} from scratch and in virtue of the fact that memory storage requirements are less demanding by just operating on the moment matrix instead of the raw data matrix, this results in a significant saving of computational execution time.

Algorithm 1: The Sweeping algorithm

```
Data: Moment matrix $M(i, j)$, predictor index= $k$
begin
    - Initialize empty matrix $S$ having same dimension of $M$
    - Initialize pivot:
    $pv = 1/m(k, k)$
    - Sweep:
    $S = M - M(., k)M(k, .)pv$
    $S(k, .) = M(k, .)pv$
    $S(., k) = -M(., k)pv$
    $S(k, k) = pv$
end
Result: Return $S$, the swept matrix
```

Table 4.6: An example of the *sweeping* procedure

We have a moment matrix M with 4 predictors in the first four rows/columns. The response is in the fifth column.

$$M = \begin{array}{cccc|c} X_1 & X_2 & X_3 & X_4 & Y \\ 1.0000 & 0.2000 & 0.3000 & 0.4000 & 0.5000 \\ 0.2000 & 1.0000 & 0.2000 & 0.3000 & 0.4000 \\ 0.3000 & 0.2000 & 1.0000 & 0.2000 & 0.3000 \\ 0.4000 & 0.3000 & 0.2000 & 1.0000 & 0.2000 \\ \hline 0.5000 & 0.4000 & 0.3000 & 0.2000 & 1.0000 \end{array}$$

After sweeping of predictors $k = 1, 2, 3, 4$ we have:

$$\tilde{M} = \begin{array}{rrrr|r} X_1 & X_2 & X_3 & X_4 & Y \\ 1.2706 & -0.0684 & -0.2812 & -0.4315 & -0.4373 \\ -0.0684 & 1.1278 & -0.1488 & -0.2812 & -0.3160 \\ -0.2812 & -0.1488 & 1.1278 & -0.0684 & -0.1245 \\ -0.4315 & -0.2812 & -0.0684 & 1.2706 & 0.0946 \\ \hline 0.4373 & 0.3160 & 0.1245 & 0.0946 & 0.6365 \end{array}$$

where:

1. An upper-left 4×4 sub-matrix, which equals the inverse of the corresponding sub-matrix of the original M.
2. A lower-right scalar, which is the residual variance over the total variance. From the example we may easily compute the R^2 as $1 - 0.6365 = 0.3635$
3. An 1×4 lower left vector contains the regression coefficients for predicting the un-swept variables (in this case the response) from the swept ones. In this case $0.4373, 0.3160, 0.1245, -0.0946$ are the coefficients for predicting Y.

Observe that we omitted to report all intermediate steps, relative to sweeping the predictor 1, first, the predictor 2 second, and so on. In this case, each of these intermediate steps would provide information of the regressions of Y, respectively, over predictor 1, over predictors 1 and 2, over predictors 1,2 and 3. This speeds up the estimation of these subsets model. Also, since sweeping is reversible we may easily exclude a predictor sweeping it again. As an example if we would like to exclude predictor 2 then we should sweep it again and we would obtain the coefficient of the regression of Y on 1,3,4.

4.3 A comparison between different approaches

In this section we review briefly some existing automatic model building procedures and compare them to RETINA conducting three Monte Carlo experiments. This comparison is not intended to be exhaustive, but still wants to assess the usefulness of the RETINA method against other selection methods in terms of forecasting ability. Systematic horse race studies on comparing RETINA with alternative automatic selection methods are scarce, and limited to the econometric literature (Pérez-Amaral et al. 2003, Castle 2005, Pérez-Amaral, Gallo & White 2005). In their original contribution PAGW compare RETINA on a Monte Carlo basis against stepwise regression and the Breiman's Non-Negative Garrote method (Breiman 1995). Pérez-Amaral et al. (2005) compare RETINA and PcGets[4] on a conceptual basis and report a comparison based on real world telecommunications data. Castle (2005) compares RETINA against PcGets based on the same telecommunications data set of Pérez-Amaral et al. (2005) besides artificial data sets already considered in Lovell (1983) and Hoover & Perez (1999). All these studies focus especially on the ability of recovering the true underlying DGP. In particular PAGW show the superiority of RETINA against stepwise regression and the NN-Garrote in recovering the true DGP for different settings of the underlying DGP. Castle (2005) concludes that both RETINA and PcGets methods are useful for their intended use of modeling and forecasting, with no clear winner, although RETINA shows a tendency of selecting more parsimonious models than PcGets. Nonetheless none of the above studies compared RETINA in terms of forecasting ability in Monte Carlo settings. Given our main interests in forecasting, we propose a new study considering three Monte Carlo studies where the main concern is exclusively the out-of-sample PMSE ability. We provide evidence of the finite sample properties of RETINA and assess its validity and accuracy in forecasting compared against other methods which are popular also in fields other than econometrics:

- Stepwise regression method (Draper & Smith 1966) which is perhaps one of the most popular selection method used in regression.
- Ridge regression (Hoerl & Kennard 1970), which has been discussed briefly in section 2.1.1 at page 13.
- Non-Negative Garrote (Breiman 1995).
- Least Absolute Shrinkage and Selection Operator (LASSO) (Tibshirani 1996*b*).

[4] This is a particular software implementation of the Gets methodology.

Table 4.7: A comparison between automatic model selection algorithms

Method	Search Strategy	Modeling Approach	Control for collinearity	Embedded Flexibility	User settings
1.RETINA	Forward	Building, Estimating Validating 3 subsamples	Yes	Yes	Collinearity threshold
2.Ridge	-	Regularization, shrinkage	Yes	no	Ridge parameter
3.NN-Garrote	-	Constrained estimation	Yes	no	Garrote parameter
4.LASSO	Forward and Backward	Constrained estimation	Yes	no	-
5.LARS	Forward	Building	Yes	no	-
6.Stepwise	Forward and Backward	Hypotheses testing	No	no	Nominal test sizes
7.Gets	Backward	Building validating 2 sub-samples	No	no	Nominal test sizes

- Least Angle Regression (LARS), (Efron, Hastie, Johnstone & Tibshirani 2004).

- General to specific (Gets) methodology (Campos, Ericsson & Hendry 2005).

A brief description for these methods (for ridge estimation see section 2.1.1), can be found in the next section, although interested readers should refer to the original contributions for further insights. Table 4.7 provides a quick reference relative to these differences. A main distinction is between methods that perform *subset selection* like Stepwise, Gets and RETINA and methods based on some *regularized estimation* strategy as the ridge, NN-Garrote, the LASSO and the LARS. Another basic distinction concerns the use of validation strategies as is for RETINA and Gets. These validation strategies are not embedded in other methods which need the user to define some stopping rule to select the final parameterization. Finally, RETINA is the only method that automatically embeds non-linear transformations of the inputs.

4.3.1 A short review of automated selection methods

Stepwise selection Among automated model selection algorithms Stepwise regression (Draper & Smith 1966) is probably the best known data driven method in the linear regression literature. Stepwise selection is a method that allows moves in either direction, dropping or adding variables at the various steps. Backward stepwise selection involves starting off in a backward approach and then potentially adding back variables if they later appear to be significant. The process is one of alternation between choosing the least significant variable to drop and then re-considering all dropped variables (except the most recently dropped) for re-introduction into the model. This means that two separate significance levels must be chosen for deletion from the model and for adding to the model. The second significance must be more stringent than the first. Forward stepwise selection is also a possibility, though not as common. In the forward approach, variables once entered may be dropped if they are no longer significant as other variables are added.

The non-negative garrote, the LASSO and the LARS. These three techniques share all the characteristic of imposing a constraint on the size of the regression coefficients. The Non-negative garrote (NN-Garrote) was introduced by Breiman (1995) and consists in imposing a constraint on the absolute values of the regression coefficients:

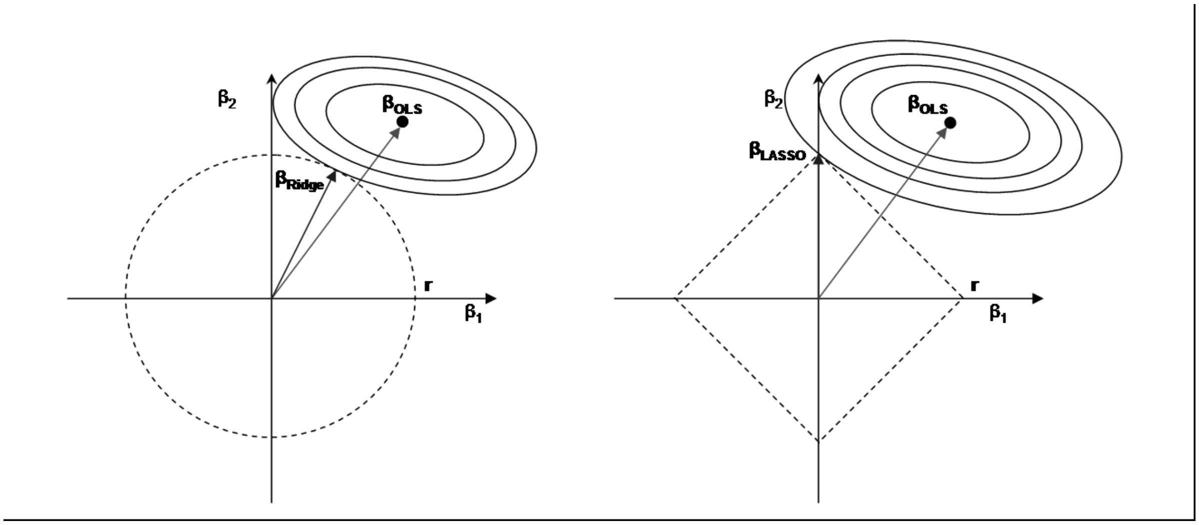
$$\hat{\beta}_{\text{NN-G}} = \text{argmin}_\beta \sum_t \left(Y_t - \sum_j c_j \beta_j X_{jt} \right)^2 \quad \text{s.t.} \quad c_j \geqslant 0 \quad \text{and} \quad \sum_j c_j \leqslant s$$

Observe that with respect to the ridge technique, here a different shrinkage factor is applied to each predictor. As the NN-Garrote and ridge, the LASSO (Tibshirani 1996*b*) is also a constrained Least Squares problem. LASSO stands for *Least Absolute Selection and Shrinkage Operator*. It minimizes the usual sum of squared errors, with a bound on the sum of the absolute values of the coefficients, that is:

$$\hat{\beta}_{\text{LASSO}} = \text{argmin}_\beta \sum_t \left(Y_t - \sum_j c_j \beta_j X_{jt} \right)^2 \quad \text{s.t.} \quad \sum_j |\beta| \leqslant s$$

This is the solution to a quadratic programming problem, and recently a simple modification of another method, the LARS (Least Angle Regression)(Efron et al. 2004), has been showed to provide also the LASSO solutions. The LARS procedure works roughly as follows. As with classic Forward Selection, we start with all coefficients equal to zero, and find the predictor most correlated with the response, say $X_{(1)}$. We take the largest step possible in the direction of this predictor until some other predictor, say $X_{(2)}$, has as much correlation with the current residual. At this point LARS parts company with Forward

Figure 4.3: Ridge and Lasso



.

Selection. Instead of continuing along $X_{(1)}$, LARS proceeds in a direction equiangular between the two predictors until a third variable $X_{(3)}$ earns its way into the "most correlated" set. LARS then proceeds equiangular between $X_{(1)}$, $X_{(2)}$ and $X_{(3)}$, that is, along the "least angle direction",until a fourth variable enters, and so on. The LARS algorithm with LASSO modification is a forward stepwise algorithm that produces all the solutions of the LASSO algorithm in a computing time proportional to the number of predictors.

General to specific (Gets) Approach. This approach is ascribed to the LSE school of econometrics and is described in Campos et al. (2005), Gilbert (1989), Pagan (1987), Hendry & Krolzig (2005) and Mizon (1995). The starting point of the methodology is to consider a sufficiently general model (the General Unrestricted Model or GUM), which includes all potentially relevant factors to describe the complexity of the real world which one wants to model. In order for the GUM to be *admissible* it has to accomplish with certain conditions such as *congruency* with the data and has to be *consistent* with some economic theory. The approach tests downwards the GUM to find a valid restriction, that is a more parsimonious model which conveys all the information contained in the more general model. The procedure is outlined in table 4.24 at page 80. The method is based on the theory of *encompassing* which implies that one specification encompasses another if it conveys all of the information included by the initial specification. The Gets method has been refined and improved as an automatic model selection tool in the commercial PcGets package, but a less sophisticate implementation coded in Matlab language by HP which can be used to reproduce the results of their paper, is available from: `http://www.feweb.vu.nl/econometriclinks/journal/volume2/HooverKD PerezSJ`.

4.3.2 Monte Carlo evidence

We will now investigate the properties of different automatic model selection algorithms in practice. We will look at the performance with respect the sample size, the nature of the data (cross-section or time series) and different features of the DGP. All simulations are conducted considering 1000 replications for each instance and different experimental setting most of which have been already used in other Monte Carlo studies (Breiman 1992, Hoover & Perez 1999, Lovell 1983, McQuarrie & Tsai 1998). We are concerned in evaluating different algorithms in terms of out-of-sample one-step-ahead forecasting errors, given that using the training set to derive a parameterization we would get too optimistic predictions and the errors would be biased downwards. A test for predictive ability is required on hold out data. We use different error measures, the Root Mean Square Error (RMSE) the Mean Absolute Error (MAE) and the Mean Relative Error (MRE), to compare performance across models on the hold-out sample:

$$RMSE = \sqrt{\frac{1}{H}\sum_{t=T+1}^{N+H}(Y_t - \hat{Y}_t)^2} \qquad (4.3.1)$$

$$MAE = \frac{1}{H}\sum_{t=T+1}^{T+H}|Y_t - \hat{Y}_t| \qquad (4.3.2)$$

$$MRE = \frac{1}{H}\sum_{t=T+1}^{T+H}\frac{|\hat{Y}_t - Y_t|}{Y_t} \qquad (4.3.3)$$

Here H is the number of observations in the hold-out sample. RMSE statistic delivers the average forecasting accuracy over squared errors, while MAE does the same over absolute deviations which are less sensitive to large forecast errors than their squared counterpart. MRE puts a different weight on errors depending on the magnitude of the value to be predicted. Small absolute deviations from the true response value may be big in comparison to the value to be predicted and vice versa. To assess whether forecasts are statistically different between different selection algorithms we use a modified version of the Morgan-Granger-Newbold[5]

[5] This is a test of out-of-sample Mean Square Error equality, assuming that the forecasting errors are unbiased, are normally distributed and serially uncorrelated.

test of comparative predictive accuracy, in Harvey, Leybourne & Newbold (1997) (HLN) which corrects for the effect of non-normality of the forecast errors. Defining as two competing forecasts at $T+h$ as $\hat{e}_{T+h}$ and $\tilde{e}_{T+h}$ the test considers the following orthogonalizing transforms:

$$u_{1,T+h} = \hat{e}_{T+h} - \tilde{e}_{T+h} \tag{4.3.4}$$

$$u_{2,T+h} = \hat{e}_{T+h} + \tilde{e}_{T+h} \tag{4.3.5}$$

A test for difference between the forecasts $\hat{e}_{T+h}$ and $\tilde{e}_{T+h}$ is equivalent to a test of null correlation between $u_{1,T+h}$ and $u_{2,T+h}$ which corresponds to testing the null hypothesis $H_0 : \beta = 0$ in the regression:

$$u_{2,T+h} = \beta u_{1,T+h} + \epsilon_{T+h} \qquad h = 1, \ldots, H$$

The test statistic is given by:

$$\text{HLN} = \hat{\beta} \left[\frac{\sum_{i=t+1}^{T+H} u_{1,t}^2 \hat{\epsilon}_i^2}{\left(\sum_{t=T+1}^{T+H} u_{1,t}^2 \right)^2} \right]^{-\frac{1}{2}}$$

which is asymptotically $t-$distributed with $H-1$ degrees of freedom under the null hypothesis.

4.3.3 Design of Experiments

We now describe the data sets used in Monte Carlo experiments.

Breiman data. The data for this experiment has been generated using the following procedure proposed by Breiman (1992). The predictors matrix X was generated independently from a multivariate normal distribution centered at the origin and with covariance satisfying $\mathbb{E}[X_i, X_j] = \rho^{|i-j|}$. Different settings were considered: the uncorrelated case with $\rho = 0$ and the correlated case with $\rho = 0.9$. Three sample sizes were considered: $N = 50$, $N = 100$ and $N = 500$. In all settings we considered 15 predictors with only three non-zero β_j coefficients centered at every 4th variable. The specification was calibrated such that $R^2 \approx 0.75$. A $\mathcal{N}(0,1)$ disturbance was added. For each ρ correlation setting, simulations were repeated 1000 times independently.

Moving Average MA(1) Misspecified as Autoregressive Models. This experiment is inspired by McQuarrie & Tsai (1998) who considered an MA(1) data generation process. The candidate parameterization are restricted to be autoregressive models, thus the true model does not belong to the set of candidate models $\mathcal{M}$. The MA(1) data generation process is:

$$Y_t = \theta u_{t-1} + u_t \quad u_t \sim \text{i.i.d. } N(0, \sigma^2)$$

We consider two settings, $\theta = .5$ and $\theta = .9$. Both are stationary and can be written in terms of an infinite order AR model:

$$Y_t = \sum_{j=0}^{\infty} \theta^j Y_{t-j}$$

Notice that the autocorrelations of the MA(1) process decay much more quickly for $\theta = .5$ than for $\theta = .9$ and thus better approximation may exist in small samples in the former case. The case where $\theta = .9$ has AR parameters that decay quite slowly and in small samples no good approximation may exist. A maximum of 15 predictors obtained as lagged valued of the response Y_t were considered, thus allowing candidate AR models of order 1 through 15 to be fitted to the data.

Lovell's (1983) and Hoover and Pèrez (1997) Time series data. To assess the performance on time series data we consider the simulation framework used by Lovell (1983) and Hoover & Perez (1999) (HP from now on). The data represent macroeconomic variables of the US economy from 1960.3 to 1995.1. A description of the data set is reported in table 4.9. There are a total of 40 predictors which include current and first lag of independent variables and four lags of the consumption variable. The data set includes two sets of quite closely related time series, the fiscal variables $(3, 4, 5)$ and monetary variables $(10, 11, 12, 13)$. As in Lovell's and HP's work, each replication of the Monte Carlo experiment consisted in generating a pseudo-real consumption dependent variable accordingly to an explicit specified stochastic process and one draw from a random number generator. For the sake of simplicity we consider here only three of the nine specifications considered by the HP study. These are reported in Table 4.8 and reflect the original numbering used in the HP paper. Briefly model 3 takes the log of simulated consumption as the dependent variable and is an AR(2) time-series model. Model 7 is a *monetary* dynamic model and model 9 is defined by Lovell as *eclectic* because here consumption is related to both the M1 monetary aggregate and government purchases. For testing purposes observations from 1991.4 to 1995.1 (a 10% of the sample) are excluded from the training sample used for the specifications search.

Table 4.8: HP DGP's used to generate artificial consumption target variables.

Model 3:	$\ln(Y3)_t = .395\ln(Y3)_{t-1} + .3995\ln(Y3)_{t-2} + .00172u_t$
Model 7:	$Y7_t = 1.33X11_t + 9.73u_t^*$
Model 9:	$Y9_t = .67X11_t - .023X3_t + 4.92u_t^*$
	$u_t \sim N(0,1) \qquad u_t^* = u_{t-1}^* + u_t\sqrt{7/4}$

4.3.4 Methodology

In all experiments, all automatic selection procedures competed in forecasting by using always the same set of predictors. In other words RETINA was not given any advantage in terms of approximating ability over its competitors, by using *level one* transformations[6]. A number of $R = 1000$ replications were carried out for each DGP and forecasting statistics were recorded for each method and each repetition. Other settings, specific to each method were as follows. The nominal size of the Gets procedure that governs the critical values used in all of the tests employed in the search were 1%,5% and 10%. For the Ridge method, the shrinkage parameter was found using GCV. For the stepwise procedure a nominal size of 5% was used. For LARS and the LASSO we used the LARS package written by Vanden Berghen (2005) and adopted as stopping rule a 5-fold cross-validation. Finally the constraint on coefficients for the NN-garrote is obtained by GCV.

4.3.5 Results

Results of predictive forecast measures for Breiman's, HP data and MA(1) are presented respectively in tables 4.10, 4.17 and 4.21. Recall these are always referred to out-of-sample predictive ability. Forecasting accuracy is evaluated on the basis of RMSE, MAE and MRE statistics as well as the values of the HLN test statistics of equivalent forecast errors averaged over all Monte Carlo replications (tables 4.11 to 4.16 for Breiman data, tables 4.18 to 4.20 for Lovell's data and 4.22 to 4.23 for the MA(1) process).

A striking fact is that all methods perform equally well in terms of forecasting performance. The forecasts from all methods are similar and it is difficult to draw

[6] This "fair" comparing approach is also used in Castle (2005) and Pérez-Amaral et al. (2005).

Table 4.9: Hoover and Perez (1999) data set.

Identifier	Variable	Times differenced for stationarity†
1. *DCOINC*	Index of four coincident indicators	1
2. *GD*	GNP price deflator	2
3. *GGEQ*	Government purchases of goods and services	2
4. *GGFEQ*	Federal purchases of goods and services	1
5. *GGFR*	Federal government receipts	2
6. *GNPQ*	GNP	1
7. *GYDQ*	Disposable personal income	1
8. *GPIQ*	Gross private domestic investment	1
9. *FMRRA*	Total member bank reserves	2
10.*FMBASE*	Monetary base	2
11.*FM1DQ*	M1	1
12.*FM2DQ*	M2	1
13.*FSDJ*	Dow Jones Stock Price	1
14.*FYAAC*	Moody's AAA corporate bond yield	1
15.*LHC*	Labor force (civilian,aged > 16)	1
16.*LHUR*	Unemployment rate	1
17.*MU*	Unfilled orders (manufacturing, all industries)	1
18.*MO*	New orders (manufacturing, all industries)	2
19.*GCQ*	Personal consumption expenditure (Response)	1

Note†: Indicates the number of times the series had to be differenced before a Phillips-Perron test could reject the null hypothesis of stationarity at a 5% significance level. Candidate variables for specification search were the original non-stationary predictors, their stationary transforms, and the lagged values of the simulated consumption response.

substantive results. The only exception is relative to Ridge estimation, which delivers a significant lower predictive accuracy, measured by the RMSE, especially for Lovell's data experiment. Test of forecast error equivalence confirms this in Tables 4.18 to 4.20 in which the HLN statistic is recorded in the lower diagonal. Negative entries refer to lower predictive performance of the method in column with respect to the method in row. Indeed Lovell's data set doesn't represent a favorable setting for ridge estimation since the proportion of relevant variables is quite low compared to the available candidates (between one and three out of 40) and where the collinearity among predictors is high. This may explain the result. The same tendency is found also for the Breiman and the MA1 experiment, although these represent somewhat more favorable settings because the total number of candidate variables in both cases is 15. A better suited shrinkage method in this context is the NN-Garrote, which tends to perform somewhat better in terms of RMSE than the remaining methods, although the difference is not statistically significant for all experiments considered (Tables 4.11 to 4.16, 4.18 to 4.20 and 4.22 to 4.23). Stepwise selection performs quite satisfactory especially for small samples sizes on the Breiman data set, but seems to be somewhat less accurate when the sample size grows. The same phenomena happens for the Gets strategy whose accuracy isn't that good as other methods when the sample size increases. On the other side, RETINA seems to improve as the sample size increases suggesting that it is better suited where the number of observations is large. The LASSO and the LARS method seem to be particularly accurate for smaller sample sizes.

4.4 Conclusions

In this chapter we presented an automatic modeling tool useful for forecasting purposes called RETINA (Pérez-Amaral et al. 2003). The method implements an automated strategy for specification search and out-of-sample model validation and testing. We reviewed in detail its main characteristics and presented a specific implementation for real data sets called RETINA Winpack, which adds specific features such as the *union model* and more importantly it allows to distinguish for continuous and categorical inputs and specify a very wide class of parameterizations which are similar to those used in an analysis of covariance setting. The Winpack has a user-friendly graphical interface which allows the less experienced to easily explore possible useful specifications that include transformations of the original data set.
The chapter also tries to fill a gap present in the literature on comparing RETINA with other methods (Pérez-Amaral et al. 2003, Castle 2005, Pérez-Amaral et al.

2005) since we explicitly assess its validity as an automatic modeling tool focusing exclusively on the out-of-sample forecasting ability against a variety of methods (Stepwise regression, Non-negative Garrote, LARS, LASSO, Ridge and the General to Specific methodology). A striking fact that emerges from the experiments is that there is no clear winner in terms of forecasting ability. RETINA seems to behave better in large sample problems, while other methods are better suited for smaller sized problems. These phenomena may be explained by the fact that the procedure always splits the sample into three sub-samples, which may reduce the efficiency especially in the specification and estimation stage. Tests for forecast equality do not show evidence of better performance of RETINA but it doesn't do worse at all considering different settings in which sample sizes (even small, eg. 50 observations), the number of candidate predictors, and the nature of the data (time series or cross-section) vary systematically across experiments.

Table 4.10: One step ahead predictive ability measures for different correlation patterns and sample sizes. Breiman data.

ρ	N	RETINA	Ridge	NN-Garrote	LASSO	LARS	Stepwise	Gets
				Hold-Out RMSE				
$\rho = 0.0$	50	1.0756 (0.0086)	1.1944 (0.0076)	1.0545 (0.0069)	1.0741 (0.0069)	1.0741 (0.0069)	1.0564 (0.0070)	1.0585 (0.0072)
	100	1.0099 (0.0046)	1.0792 (0.0050)	1.0193 (0.0046)	1.0348 (0.0048)	1.0348 (0.0048)	1.0248 (0.0047)	1.0215 (0.0047)
	500	1.0015 (0.0020)	1.0135 (0.0020)	1.0023 (0.0020)	1.0063 (0.0020)	1.0063 (0.0020)	1.0044 (0.0020)	1.0036 (0.0020)
$\rho = 0.9$	50	1.1549 (0.0077)	1.1759 (0.0079)	1.0888 (0.0074)	1.0671 (0.0071)	1.0660 (0.0071)	1.1046 (0.0074)	1.1322 (0.0076)
	100	1.0861 (0.0053)	1.0875 (0.0050)	1.0508 (0.0049)	1.0496 (0.0050)	1.0486 (0.0050)	1.0522 (0.0050)	1.0534 (0.0051)
	500	1.0028 (0.0020)	1.0119 (0.0020)	1.0039 (0.0020)	1.0083 (0.0022)	1.0082 (0.0022)	1.0030 (0.0020)	1.0021 (0.0020)
				Hold-Out MAE				
$\rho = 0.0$	50	0.8761 (0.0074)	0.9703 (0.0065)	0.8561 (0.0059)	0.8727 (0.0058)	0.8727 (0.0058)	0.8582 (0.0059)	0.8605 (0.0061)
	100	0.8119 (0.0039)	0.8674 (0.0042)	0.8198 (0.0039)	0.8324 (0.0040)	0.8324 (0.0040)	0.8243 (0.0040)	0.8216 (0.0040)
	500	0.8005 (0.0017)	0.8099 (0.0017)	0.8009 (0.0017)	0.8041 (0.0017)	0.8041 (0.0017)	0.8027 (0.0017)	0.8021 (0.0017)
$\rho = 0.9$	50	0.9408 (0.0067)	0.9521 (0.0066)	0.8856 (0.0063)	0.8657 (0.0059)	0.8646 (0.0059)	0.8957 (0.0062)	0.9158 (0.0064)
	100	0.8737 (0.0045)	0.8742 (0.0042)	0.8455 (0.0042)	0.8434 (0.0041)	0.8428 (0.0041)	0.8459 (0.0042)	0.8464 (0.0042)
	500	0.8016 (0.0017)	0.8088 (0.0017)	0.8024 (0.0017)	0.8060 (0.0018)	0.8059 (0.0018)	0.8016 (0.0017)	0.8009 (0.0017)
				Hold-Out MRE				
$\rho = 0.0$	50	0.2001 (0.2195)	0.0442 (0.2243)	0.2756 (0.1959)	0.2621 (0.1859)	0.2621 (0.1859)	0.1258 (0.1924)	0.1278 (0.1920)
	100	0.6597 (0.5582)	0.5591 (0.6584)	0.6425 (0.5444)	0.6246 (0.5766)	0.6246 (0.5766)	0.5494 (0.4847)	0.5651 (0.4860)
	500	-0.2964 (0.3429)	-0.0141 (0.3207)	-0.2469 (0.3431)	-0.2643 (0.3545)	-0.2643 (0.3545)	-0.2897 (0.3568)	-0.2680 (0.3448)
$\rho = 0.9$	50	6.4115 (5.5288)	-0.9416 (1.6434)	2.3541 (1.5911)	5.2591 (4.4879)	5.1705 (4.3948)	4.7737 (3.8725)	4.8823 (3.9388)
	100	1.1350 (0.6152)	1.7350 (0.8910)	1.5288 (0.5871)	1.3096 (0.5000)	1.2811 (0.5000)	1.3794 (0.5635)	1.3641 (0.5628)
	500	5.2163 (3.4828)	5.1258 (3.3898)	4.9455 (3.2546)	5.0486 (3.3747)	5.0461 (3.3747)	4.5945 (3.0466)	5.2144 (3.4929)

Breiman data. Average HLN test statistics of equivalent forecast errors. Negative entries refer to lower predictive performance of the method in column with respect to the method in row. On average, none of the absolute values of the HLN statistic is larger than two.

Table 4.11: $n = 100$ and $\rho = 0$.

	1	2	3	4	5	6
1. RETINA	–					
2. Ridge	1.0699					
3. NN-Garrote	0.0413	−1.3639				
4. LASSO	0.2345	−1.1237	0.3107			
5. LARS	0.2345	−1.1237	0.3107	0.0000		
6. Stepwise	0.0195	−1.2233	−0.0113	−0.2688	−0.2688	
7.Gets	−0.0009	−1.2274	−0.0227	−0.2956	−0.2956	−0.0130

Table 4.12: $n = 100$ and $\rho = 0$.

	1	2	3	4	5	6
1.RETINA						
2.Ridge	1.0541					
3.NN-Garrote	0.3181	−1.0478				
4.LASSO	0.5167	−0.8365	0.3411			
5.LARS	0.5167	−0.8365	0.3411	0.0000		
6.Stepwise	0.3109	−0.9143	0.1070	−0.2199	−0.2199	
7.Gets	0.2361	−0.9540	−0.0066	−0.2800	−0.2800	−0.0767

Table 4.13: $n = 500$ and $\rho = 0$.

	1	2	3	4	5	6
1.RETINA						
2.Ridge	0.8825					
3.NN-Garrote	0.1852	−0.9130				
4.LASSO	0.4219	−0.6692	0.4270			
5.LARS	0.4219	−0.6692	0.4270	0.0000		
6.Stepwise	0.2815	−0.7510	0.2029	−0.2056	−0.2056	
7.Gets	0.2000	−0.7895	0.0865	−0.2752	−0.2752	−0.0938

Breiman data. Average HLN test statistics of equivalent forecast errors. Negative entries refer to lower predictive performance of the method in column with respect to the method in row. On average, none of the absolute values of the HLN statistic is larger than two.

Table 4.14: $n = 50$ and $\rho = 0.9$

	1	2	3	4	5	6
1.RETINA						
2.Ridge	0.1187					
3.NN-Garrote	−0.6754	−0.9430				
4.LASSO	−0.9055	−0.8938	−0.2621			
5.LARS	−0.9221	−0.9086	−0.2596	−0.0220		
6.Stepwise	−0.5068	−0.5894	0.2829	0.4778	0.4925	
7.Gets	−0.2527	−0.4253	0.6356	0.6609	0.6675	0.2631

Table 4.15: $n = 100$ and $\rho = 0.9$.

	1	2	3	4	5	6
1.RETINA						
2.Ridge	0.1047					
3.NN-Garrote	−0.5074	−0.8051				
4.LASSO	−0.6035	−0.6467	−0.0804			
5.LARS	−0.6177	−0.6691	−0.0918	−0.0102		
6.Stepwise	−0.4780	−0.6021	0.0757	0.1475	0.1509	
7.Gets	−0.4668	−0.5905	0.0719	0.0976	0.0988	0.0012

Table 4.16: $n = 500$ and $\rho = 0.9$.

	1	2	3	4	5	6
1. RETINA						
2. Ridge	0.7648					
3. NN-Garrote	0.1132	−0.8220				
4. LASSO	0.1513	−0.5334	0.0770			
5. LARS	0.1534	−0.5316	0.0814	0.0062		
6. Stepwise	−0.0497	−0.7832	−0.0947	−0.1935	−0.1956	
7. Gets	−0.1494	−0.8197	−0.2280	−0.2720	−0.2737	−0.1127

Table 4.17: One-step ahead predictive ability measures for Hoover and Pérez models $3, 7$ and 9. Observe that Ridge estimation delivers a significant lower RMSE predictive accuracy (see also tables 4.18, 4.19 and 4.20). This may due to the fact that there is a high number of irrelevant predictors. Recall that ridge regression shrinks all regression coefficients towards zero, but retains all of them.

model	RETINA	Ridge	NN-Garrote	LASSO	LARS	Stepwise	Gets
Hold-Out RMSE							
Model 3	0.0017 (0.0000)	0.0032 (0.0000)	0.0019 (0.0000)	0.0019 (0.0000)	0.0019 (0.0000)	0.0020 (0.0000)	0.0020 (0.0000)
Model 7	9.4482 (0.0626)	15.0431 (0.1077)	9.3808 (0.0593)	10.3504 (0.0720)	10.3500 (0.0720)	9.9321 (0.0770)	10.0810 (0.0818)
Model 9	4.7179 (0.0322)	7.4996 (0.0529)	4.7056 (0.0304)	5.2125 (0.0371)	5.2123 (0.0370)	4.9862 (0.0400)	4.9896 (0.0424)
Hold-Out MAE							
Model 3	0.0014 (0.0000)	0.0026 (0.0000)	0.0016 (0.0000)	0.0015 (0.0000)	0.0015 (0.0000)	0.0016 (0.0000)	0.0016 (0.0000)
Model 7	7.6493 (0.0535)	12.0867 (0.0877)	7.5875 (0.0495)	8.3605 (0.0602)	8.3604 (0.0602)	8.0385 (0.0640)	8.1515 (0.0687)
Model 9	3.8302 (0.0276)	6.0299 (0.0430)	3.8202 (0.0258)	4.2119 (0.0307)	4.2119 (0.0307)	4.0354 (0.0331)	4.0233 (0.0344)
Hold-Out MRE							
Model 3	1.6035 (1.0554)	3.1764 (2.1307)	1.4719 (0.7428)	1.2296 (0.6669)	1.2296 (0.6669)	1.2887 (1.0234)	2.0510 (1.1216)
Model 7	0.1176 (0.4514)	1.5315 (1.1552)	0.3072 (0.3819)	0.6012 (0.4953)	0.6015 (0.4953)	0.9671 (0.7748)	0.9831 (0.7961)
Model 9	-0.2466 (1.0353)	-1.9479 (2.6976)	0.5520 (0.7358)	-0.8876 (1.0085)	-0.8875 (1.0085)	-2.0880 (1.8160)	-0.7355 (0.9224)

Table 4.18: Average HLN test statistics of equivalent forecast errors for Hoover and Pérez Model 3. Negative entries refer to lower RMSE predictive performance of the method in column with respect to the method in row. Notice that Ridge regression (column 2) provides significantly lower accuracy on out-of-sample forecasts than its competitors.

	1	2	3	4	5	6
1.RETINA	−					
2.Ridge	3.9247	−				
3.NN-Garrote	1.1538	−3.1572	−			
4.LASSO	0.6227	−3.6072	−0.6624	−		
5.LARS	0.6227	−3.6072	−0.6624		−	
6.Stepwise	1.1336	−3.3231	−0.0353	0.5713	0.5713	−
7.Gets	1.0839	−3.3515	−0.0187	0.5495	0.5495	−0.0199

Table 4.19: Average HLN test statistics of equivalent forecast errors for Hoover and Pérez Model 7

	1	2	3	4	5	6
1.RETINA	−					
2.Ridge	2.8647	−				
3.NN-Garrote	−0.1439	−2.8625	−			
4.LASSO	0.6717	−2.4494	0.7509	−		
5.LARS	0.6713	−2.4494	0.7503	−0.0061	−	
6.Stepwise	0.1529	−2.8897	0.1861	−0.5451	−0.5441	−
7.Gets	0.2782	−2.8043	0.2737	−0.3785	−0.3776	0.1519

Table 4.20: Average HLN test statistics of equivalent forecast errors for Hoover and Pérez Model 9.

	1	2	3	4	5	6
1.RETINA	−					
2.Ridge	2.9588					
3.NN-Garrote	−0.0989	−2.8911	−			
4.LASSO	0.7740	−2.3835	0.7479	−		
5.LARS	0.7739	−2.3817	0.7471	0.0009	−	
6.Stepwise	0.2480	−2.8359	0.1998	−0.5436	−0.5429	−
7.Gets	0.1947	−2.8458	0.1425	−0.5462	−0.5492	0.0106

Table 4.21: MA(1) Artificial data. One-step ahead predictive ability measures with $n = 1000$ replications

MA(1) Parameter	RETINA	Ridge	NN-Garrote	LASSO	LARS	Stepwise	Gets
Hold-Out RMSE							
$\theta = .5$	1.1253 (0.0086)	1.3140 (0.0118)	1.0930 (0.0077)	1.1359 (0.0084)	1.1357 (0.0083)	1.1428 (0.0085)	1.1441 (0.0089)
$\theta = .9$	1.2398 (0.0100)	1.3157 (0.0127)	1.1788 (0.0091)	1.2521 (0.0097)	1.2523 (0.0097)	1.2317 (0.0099)	1.2267 (0.0104)
Hold-Out MAE							
$\theta = .5$	0.9208 (0.0074)	1.0777 (0.0107)	0.8931 (0.0067)	0.9298 (0.0073)	0.9297 (0.0073)	0.9340 (0.0073)	0.9366 (0.0077)
$\theta = .9$	1.0107 (0.0086)	1.0802 (0.0112)	0.9597 (0.0078)	1.0184 (0.0082)	1.0184 (0.0082)	1.0045 (0.0084)	1.0021 (0.0088)
Hold-Out MRE							
$\theta = .5$	0.9658 (0.6282)	2.7798 (1.3150)	1.0675 (0.5147)	1.9786 (0.6885)	1.9777 (0.6885)	1.1809 (0.5966)	1.5722 (0.7785)
$\theta = .9$	2.9447 (1.4116)	2.0783 (1.1438)	1.9999 (0.8930)	2.2672 (1.0921)	2.2489 (1.0921)	3.0085 (1.3954)	2.8316 (1.3624)

Average HLN test statistics of equivalent one-step-ahead forecast errors for MA(1) DGP. Negative entries refer to lower predictive performance of the method in column with respect to the method in row.

Table 4.22: $\theta = .5$.

	1	2	3	4	5	6
1.RETINA	–					
2.Ridge	1.1248	–				
3.NN-Garrote	−0.5507	−1.5371	–			
4.LASSO	−0.0038	−1.2877	0.6689	–		
5.LARS	−0.0059	−1.2883	0.6685	−0.0037	–	
6.Stepwise	0.1514	−1.0720	0.7305	0.2423	0.2430	–
7.Gets	0.1262	−1.0911	0.6966	0.1895	0.1910	−0.0309

Table 4.23: $\theta = .9$

	1	2	3	4	5	6
1.RETINA	–					
2.Ridge	0.2560	–				
3.NN-Garrote	−0.7701	−0.7347	–			
4.LASSO	−0.0003	−0.2164	0.9350	–		
5.LARS	0.0047	−0.2194	0.9388	−0.0004	–	
6.Stepwise	−0.1236	−0.3700	0.5770	−0.1069	−0.1103	–
7.Gets	−0.1580	−0.4613	0.4036	−0.2309	−0.2335	−0.0734

Table 4.24: The Hoover and Perez (1999) Gets algorithm

Stage I – Preliminary: Formulate a General Unrestricted Model (GUM)

1. Split the sample into two parts: the training sample and the test sample.
2. Formulate a General Unrestricted Model and Check for Admissibility
   - (a) Check consistency with theory
   - (b) Check congruency with data
   - (c) Run battery of tests:
     - i. Normality of residuals (Jarque Bera, 1980)
     - ii. Autocorrelation of residuals up to second order (Godfrey, 1978, Breusch and Pagan 1980)
     - iii. Autocorrelated conditional heteroskedasticity (ARCH) up to second order (Engle, 1982)
     - iv. Chow test for in-sample and out-of sample coefficient stability (Chow, 1960)
     - v. If specifications passes all test at the nominal test size go ahead, otherwise re-formulate the GUM or adopt a looser test size.

Stage II – Specification search and reduction

3. Using the training sample: Rank the regressors based on the t-statistics. Initialize the number of search paths, (HP use 10 paths). For each search path:
   - (a) Eliminate least significant variable in the subset of non-significant variables (according to the nominal test size) and re-estimate.
   - (b) Run again the battery of tests as above and run an F-test of the hypothesis that the current specification is a valid restriction of the general specification
   - (c) If the current specification passes all of the tests the variable with the next lowest $t-$statistic is removed.
   - (d) Re-run the battery of tests as in I.2.c). If the current specification fails any of these tests, the last eliminated predictor is restored and the current specification is re-estimated eliminating the variable with the next lowest $t-$ statistic.
   - (e) The process of variable elimination ends when a current specification passes the battery of tests an either has all variables significant or cannot eliminate any remaining non-significant variables without failing one of the tests.
4. Using the whole sample, re-estimate the model
   - (a) If all variables are significant the current specification is the terminal specification.
   - (b) If any variables are non-significant they are removed as a block and the battery of test is performed again on the current specification.
     - i. If the new model passes and all variables are significant the new model is the terminal model and try a new search path.
     - ii. If the model does not pass, restore the block and try a new search path.
     - iii. If the new model passes and some variables are insignificant, return to II.4.b).
   - (c) Iterate: after a terminal specification has been reached, store it in memory and the next search path is tried until all paths have been searched.

Stage III – Model Selection.

Once all paths have ended in a terminal specification, the final specification for the replication is the terminal specification with the lowest standard error of the regression.

Chapter 5

An application of RETINA to Telecommunications data

In this chapter we present an application of the RETINA procedure to predict the business telecommunications demand for short, medium and large distance telephone services. We analyze firm level data of US companies from the Bill Harvesting data base[1] of 1997. Parts and samples of this data set were also used by Pérez-Amaral & Marinucci (2002), Pérez-Amaral et al. (2005) and Castle (2005). The data base is a cross-section of 13766 firms observed in 1997 and includes expenditures for local calls and time spent for medium and large distance calls. Many predictors are available, providing detailed information about the characteristics of the firm such as the number of employees or its physical extension. However the data presents many problems such as anomalous observations and missing information including the prices charged for the services. Additional difficulties arise because of unobserved predictors. Taylor (1996) suggests that consumption patterns among firms may depend on determinants other than prices charged, such as the size, the activity sector and the localization of the business. Thus we expect that incorporating this information in the prediction function may help in explaining and predicting telecommunications demand. Unobserved sources of variation are found by running mixture regressions. Using this method we find additional predictors which we use as inputs of RETINA (i.e. the dummy variables which represent the group membership of each firm for each type of demand). Since the number of candidate predictors (including transformations) is potentially very large, we exploit the fact that the automated procedure is able to select "good" parameterizations for predictive purposes. We find quite parsimonious representations for each type of

[1] Bill Harvesting is a proprietary methodology of PNR & Associates (now TNS Telecoms).

demand and obtain estimations of consumption elasticities relative to specific characteristics of the firms. The specifications suggested by the procedure are assessed on the ground of cross-validation measures which are readily available from the output of the RETINA Winpack software (see section 4.2). We compare suggested specifications versus a linear baseline specification. Suggested specifications perform satisfactorily in terms of the Cross-Validated Mean Square Prediction Errors (CMSPE). Interestingly firm's telephone equipment variables show to be relevant predictors. On the other hand, the output of the firm, as well as its physical extension, have second order, yet significant effects on the demand for telecommunication services. Estimated elasticities are different for the three demands but always positive for access form (single-line or private network). Cross-elasticities also show possible "substitution patterns" between different telephone equipments. The rest of the chapter is organized as follows: section 5.1 discusses related work and the empirical approach we adopt, section 5.2 describes the data, section 5.3 discusses the methodology we used, section 5.4 presents the results and section 5.5 contains the conclusions. The appendixes follow.

5.1 Business toll demand forecasting

The literature on econometric modeling of Telecommunications demand is very extensive[2]. The theoretical framework for the modeling is well known and goes back to Artle & Averous (1973), Von Rabenau & Stahl (1974) and Rohlfs (1974) among others. Public empirical studies on business demand are not so abundant. A relevant contribution in this field is the pioneering work of Ben-Akiva & Gershenfeld (1989) which focuses on the demand for different types of access lines[3]. Another example is where they estimated the demand for local telecommunications services by using Bill Harvesting. Within a classical microeconomic framework business telecommunications demand, is considered as a production input. Demand, considered as a function of the price and other production factors, is derived from the firm's cost minimization conditions. In this circumstance we follow Taylor (1996) as a useful

[2] Many residential demand studies use Bill Harvesting or other customer databases. For example Kridel & Taylor (1997) presented a study of carrier choice, usage demand and price elasticities for the residential intra-LATA toll market using Bill Harvesting data. Taylor (1996) estimated competitive cross-price elasticities for the residential intra-LATA toll market with a two stage approach and using Bill Harvesting data. Levy (1998) estimated a semi-parametric generalized additive Tobit model of residential Intra-LATA Telephone demand on a cross-section of residential telephone consumers across 28 states using bills of GTE customers.

[3] They consider a discrete choice framework to estimate price elasticities with respect to the choice of different telephone systems (PBX, Centrex).

starting point of our empirical analysis. He divides firms into four generic types, where each type is referred to as a stage. Firms need telecommunications services not only for external communications but also for internal use, and this need increases nonlinearly with the size, the location and the activity sector of the firm.

Stage I firms are assumed to operate from a single location and are supposed to have mostly external communications needs. Moreover, they are supposed to access the public network with few single-line telephone systems. These are usually small-sized businesses.

Stage II firms have multiple locations in the same locality. As the number of employees increases, the internal use of telecommunications grows. Increased usage can be accommodated by increasing the number of lines until the purchase (or rental) of a small private network is considered. Nonetheless, purchasing toll services in bulk (WATS, 800 service)[4] is frequent as a valid alternative to such a decision and small businesses usually still work well with multiple single-lines.

Stage III firms in general tend to be larger than stage I and II firms. But the main difference is that they have multiple locations in different localities. Stage III firms may switch from multiple single lines to private networks if there is a sufficient volume of communications between fixed points. In this stage access to the public network is still required for external needs, while internal needs are largely satisfied by the private network. Nonetheless, frequently so-called smart switches are used to select the lowest cost for external or internal calls. This is done by routing a call over the private network and then into the corresponding local destination area.

Stage IV firms include multinational corporations located in multiple countries. The main difference with respect to previous stages is their bigger size and the fact that their workers are spread across different states and countries. Thus International Toll services are required for business activity.

[4] WATS: Wide Area Telephone Service is a flat rate or a special rate pay-by-the-minute (measured) billing for a specified calling area. It is usually offered by companies that buy transmission capacity in bulk from other network operators in order to re-offer it to customers at lower prices.

Table 5.1: Summary of an a priori segmentation scheme proposed by Taylor (1994).

	Stage I	**Stage II to IV**
Locations	Single location	Multiple locations, same locality, or in multiple localities
Type of Usage	External	Internal +External
Type of access	Multiple Single Lines (Business Lines, Hunting Lines)	Multiple Single Lines Private Network (PBX, Centrex, WATS, 800 service)
Sociodemographic characteristics	The number of Employees may be low with respect to firms that are not in stage I	Number of employees larger than in stage I

5.2 The Data

Our Bill Harvesting database has complete information on 4391 firms. Details about the data pre-processing can be found in the appendix. The data has been provided by PNR & Associates (Philadelphia, PA) which today forms part of the TNS group. Since the AT&T divestiture (January 1^{st}, 1984) local telecommunication services in this area are provided in a quasi-monopoly regime by Bell South. In fact 78% of the firms were served by this company and the rest by other independent carriers.

Since visual inspection of the histograms and empirical densities of the original variables shows highly skewed distributions, log-transforms have always been considered. Logarithmic transformations tend to normalize the data, stabilize the variances and limit the potential negative effect of the most extreme observations. Variables with zero values, such as the number of lines, have all been augmented by a unit constant prior to transformations. We also consider $log - ratio$ transforms, by using the *log* of the ratio between the original variables *(BUS, HUN, PBX, CTX, SAL, EMT, SQFT)* and the number of workers employed locally *EMH*. Worker per capita transforms, obtained by dividing the variables by the number of employees working locally *EMH*, have been chosen since they are common in the literature and reduce heteroscedasticity. A description of the original variables is reported in Table 5.2, while descriptive statistics of their transformations over the complete data sample are given in table 5.3 and figure 5.1.

Table 5.2: 1997 Bill Harvesting Data: Variable definitions†.

Variable	Description
$LOCAL$	Total expenditures for local calls in dollars
$INTRA$	Total duration of intra-LATA calls in minutes
$INTER$	Total duration of inter-LATA calls in minutes
BUS^a	Number of Business Lines +1
HUN^b	Number of Hunting Lines +1
PBX^c	Number of PBX Trunks +1
CTX^d	Number of Centrex Lines +1
SAL	Sales expressed in dollars
EMT	Total number of employees
EMH	Number of employees working locally
$SQFT$	Square footage of the firm
POP	Population habitat size
$IMILLS$	Inverse of the Mills ratio (see Appendix 5.6.1)
$STAGE\ I$	Binary variable= 1 if Firm is at stage I
$BSOUTH$	Binary variable= 1 if Service is provided by Bell South
AL	Binary variable= 1 if Alabama
GA	Binary variable= 1 if Georgia
KY	Binary variable= 1 if Kentucky
LA	Binary variable= 1 if Lousiana
MS	Binary variable= 1 if Missouri
NC	Binary variable= 1 if North Connecticut
SC	Binary variable= 1 if South Connecticut
TN	Binary variable= 1 if Tennessee
FL	Binary variable= 1 if Florida (omitted to avoid perfect colinearity)

†Source: PNR & Associates, Philadelphia, PA, now TNS.

a. BUS: Business Lines. A service that handles all the routine business telecommunications applications. Data transmissions for fax, email, and Internet access are usually charged at the same price as voice calls.

b. HUN: Hunting Lines. A service that bundles all the telephone lines (2 lines up) in the same location to be easily accessible with a single number (pilot number).

c. PBX: PBX Trunks. Connections between an organization's PBX (Private Branch eXchange) and the outside telephone network. Telephone users within the customer's company share these connections for making and receiving calls outside the company's network.

d. CTX: Centrex Lines. (Central office exchange service) is a service which is functionally equivalent to the PBX and consists of up-to-date phone facilities offered by the telephone company to business users so they do not need to purchase the equipment. The Centrex service effectively partitions part of its own centralized capabilities among its business customers. The customer is spared the expense of having to keep up with fast-moving technology changes and the phone company has a new set of services to sell. In many cases, Centrex has now replaced the private branch exchange. The central office has effectively become a huge branch exchange for all of its local customers. In most cases, the Centrex service provides customers with as much if not more control over the services they have than PBX did.

Notice that Business and Hunting Lines can be considered as single line access forms while PBX and Centrex services are network access forms.

Table 5.3: Univariate statistics of the *log* of each variable per worker.

	Mean	Std. Dev.	Median	Kurtosis	Skewness	n
$\ln(LOCAL/EMH)$	2.556	1.049	2.613	.580	−.135	4391
$\ln(INTRA/EMH)$	1.296	1.767	1.428	.001	−.416	1261
$\ln(INTER/EMH)$	2.538	1.573	2.693	−.108	−.322	1176
$\ln(BUS/EMH)$	−1.614	1.864	−1.061	.147	−.898	4391
$\ln(HUN/EMH)$	−1.919	1.447	−1.609	.820	−.691	4391
$\ln(PBX/EMH)$	−2.490	1.354	−2.398	.207	−.323	4391
$\ln(CTX/EMH)$	−2.259	1.699	−2.197	.294	−.295	4391
$\ln(SAL/EMH)$	1.249	3.499	.182	.280	1.176	4391
$\ln(EMT/EMH)$	.249	.706	.000	21.277	4.217	4391
$\ln(SQFT/EMH)$	5.928	1.235	5.968	1.399	−.273	4391
$\ln(POP)$	1.193	2.273	9.770	−1.461	.210	4391

Figure 5.1: Histograms, by the stage of the firm.

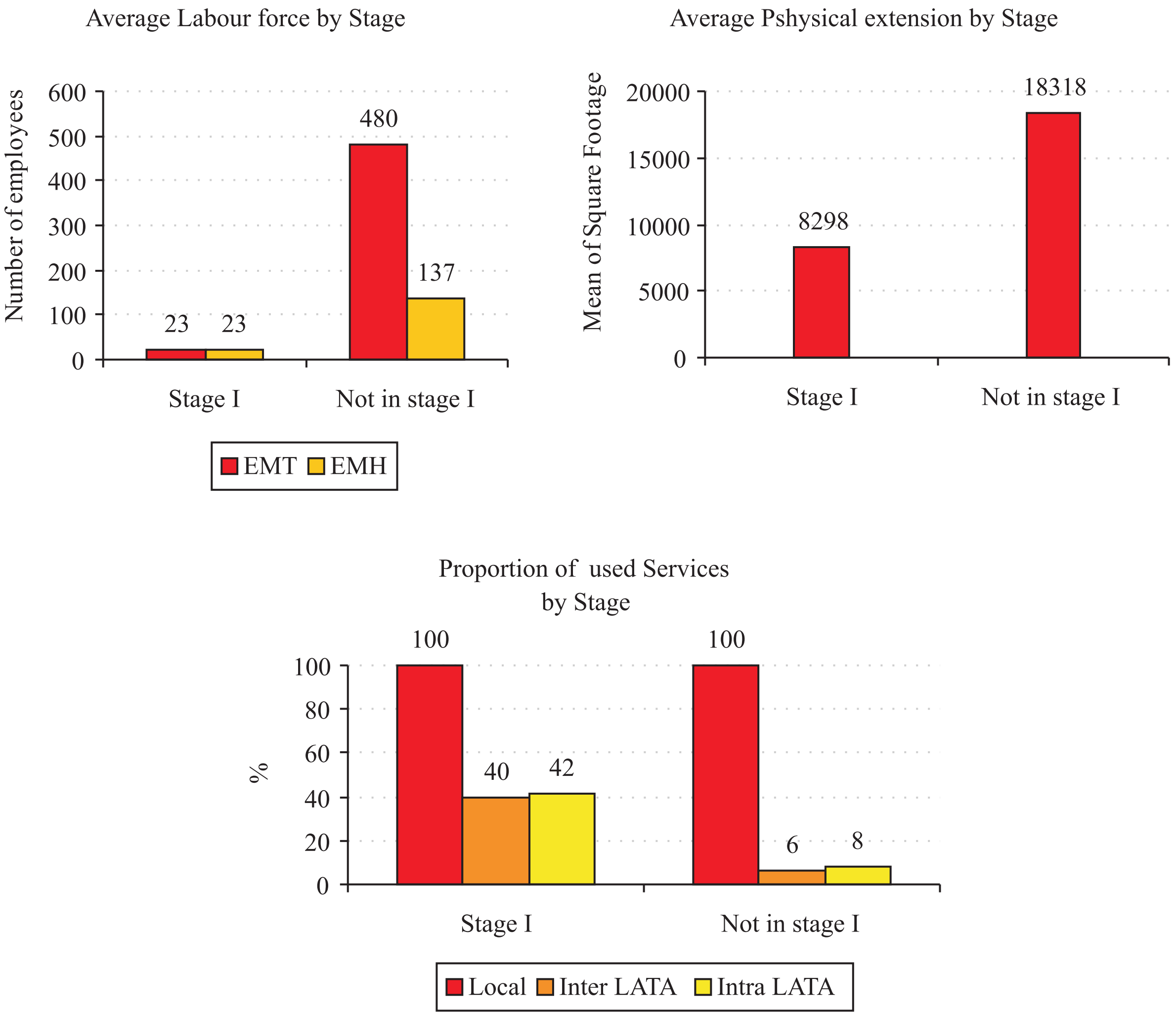


Table 5.4: Demand by type of access: Firms that demand intra-LATA and inter-LATA calls do not own private networks (Vertical %).

Type of access	**Firm demands only Local calls** ($n = 2921$)	**Firm demands intra-LATA or inter-LATA calls** ($n = 1542$)
Firm owns Multi-Single Lines (Business or Hunting lines)	80.2%	99.6%
Firm owns Private Networks (PBX or Centrex)	39.3%	.7%

The data include four types of variables:

Access form variables: There are four different types of lines, which may be grouped into two categories. The first includes single-line access equipment: business lines (BUS) and hunting lines (HUN). The second group represents private network access forms and includes PBX trunks (PBX) and Centrex lines (CTX).

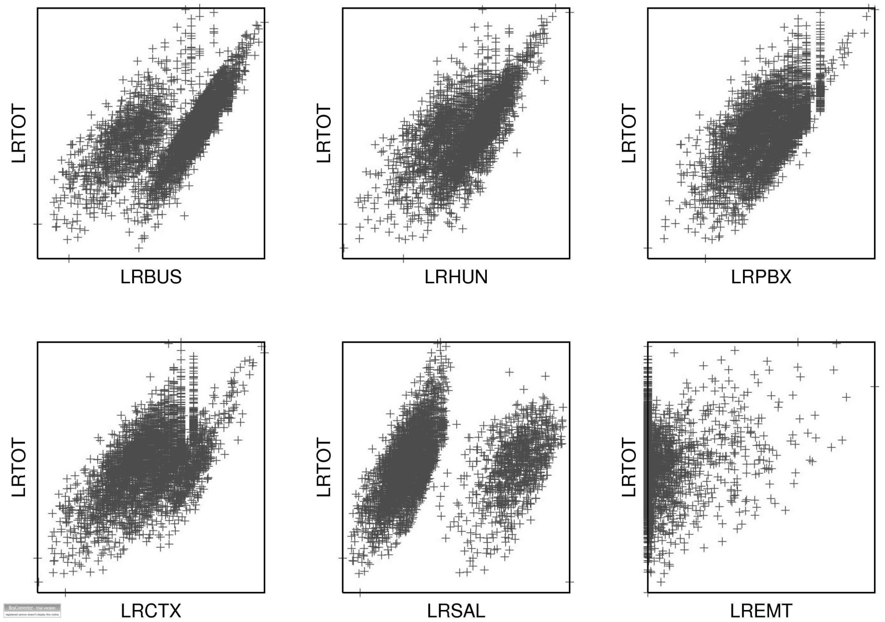
Socio-demographic variables: These are the population habitat size (POP) and the States (*AL, GA, KY, LA, MS, NC, SC, TN*).

Business size and dispersion related variables, such as the number of employees in the whole business (EMT), the number of workers employed locally (EMH) and the physical extension of the firm ($SQFT$).

Output variable: the sales of the firm (SAL).

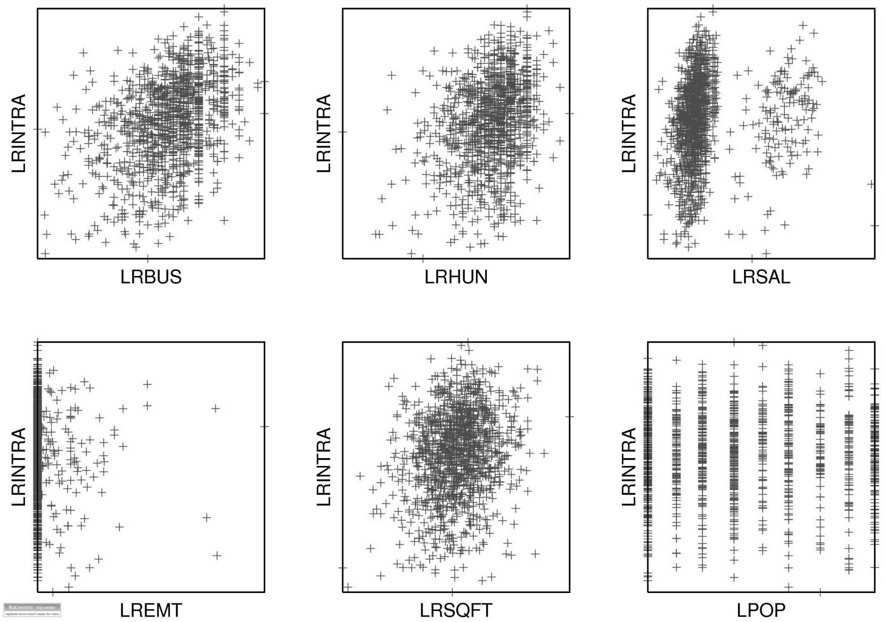
Bivariate plots of the transformed variables, reported in Figures 5.2, 5.3 and 5.4, announce that the modeling problem is difficult especially because of non-linearities and heterogeneity among businesses with respect to telecommunications services. From these plots we find initial evidence of heterogeneity. In some cases as for local demand (Figure 5.2), two moderately separated clusters may be visually identified. Clusters appear upward sloping and as elliptic shaped clouds, suggesting that they may have different mean and covariance structures. Demand for medium and long distance services (Figures 5.3 and 5.4) also accounts for evident heterogeneity especially with respect to the firm output proxied by sales (SAL). Nonetheless for the remaining variables, heterogeneity is visually much less evident and statistical methods are necessary to assess its existence.

Figure 5.2: Bivariate plots of Local demand vs. explanatory variables. (The LR-prefix stands for the *log* transformation of the original variables divided by *EMH*).



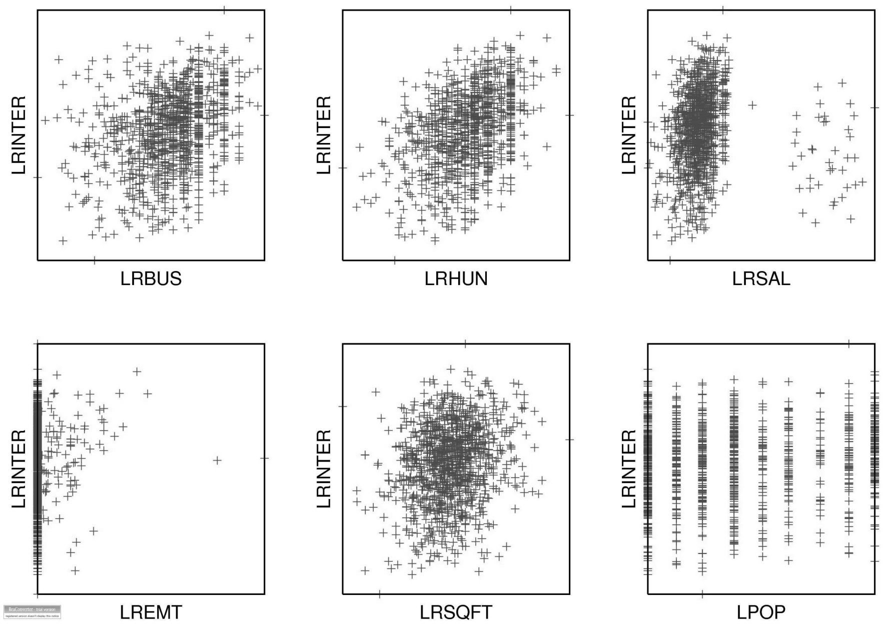
Two or more groups are visible. Heterogeneity patterns with respect to the demand for local services are visible for the number of Business Lines (LRBUS) and sales (LRSAL).

Figure 5.3: Bivariate plots of intra-LATA demand vs. explanatory variables. (The LR-prefix stands for the *log* transformation of the original variables divided by *EMH*).



Intra-LATA services show a possible two-cluster structure especially with respect to sales (LRSAL).

Figure 5.4: Bivariate plots of inter-LATA demand vs. explanatory variables. (The LR-prefix stands for the *log* transformation of the original variables divided by *EMH*).



Heterogeneity in inter-LATA services is not so evident. Non-linearities emerge from LREMT, the log of the ratio between the total number of employees (EMT) and the number of workers employed locally (EMH).

A new variable called Stage I was also added to the analysis. This is a dummy variable which proxies Taylor's definition of Stage I firms, i.e., single location businesses with only a single-line access form used for external communication purposes[5]. This variable is used to show some other interesting facts as reported in Figure 5.1. For example note that stage I firms are on average smaller, in terms of number of employees and their physical extension, than firms at higher stages, although to some extent this also depends on the nature of the markets sold in. Moreover, Figure 5.1 shows that intra-LATA and inter-LATA services are almost exclusively demanded by stage I firms. On the contrary, bigger or multiple-location firms that are not at stage I make a more intensive use of local services. Yet this seems plausible only if such firms use some "smart" switches which route non-internal calls over the private network and then into the appropriate local area through a local call.

Finally, from Table 5.4 we also learn that firms using intra-LATA or inter-LATA services use almost exclusively single-line equipment access forms (99.6%). In other

[5] Location conditions have been inferred from the difference between the number of employees working locally (EMH) and the total number of employees of the company (EMT). If the difference $EMT - EMH = 0$ then the firm is assumed to be single location.

words the private network dimension will not play a relevant role in the explanation of medium and long distance calls and may be dropped without losing relevant information during the modeling process. We conclude this by bearing in mind that telecommunication services demanded by the firms are related with the dimension of the firm and its location.

5.3 Methodology

Our analysis begins with the defintion of a baseline specification that we call Benchmark Linear Model (BLM). We define a BLM for each type of demand. The BLM is the specification that one would consider a priori, without using any particular selection strategy using only the original predictors provided in the data base. This is the natural starting point because we require any alternative specification to have a lower approximation error than the BLM. Our final objective is to obtain a approximating function which we call Useful Representative Model (URM). The URM should have higher predictive ability over the corresponding BLM, keeping the number of parameters as low as possible (parsimony).

We assume that it is possible to approximate telecommunication demand as a function of firms characteristics. Relevant information is available on the number of employees, which can give an idea about the dimension of internal communication needs. Output is proxied by sales although its relevance is unclear a priori, since phone calls are made by people and sales may influence the volume of calls only if the business involves a heavy tele-marketing activity. Socio-demographic variables such as the population habitat size and the geographic region are included in the general specification as well, but their effects are uncertain. The signs of the coefficients are expected to be positive in the case of the number of different types of lines and the relative size of the firm. Response variables are defined as follows:

ln(LOCAL/EMH): $log - ratio$ of the expenditures in local calls in dollars per worker.

ln(INTRA/EMH): $log - ratio$ of the duration of intra-LATA calls in minutes per worker.

ln(INTER/EMH): $log - ratio$ of the duration of inter-LATA calls in minutes per worker.

As starting point, we adopt a double logarithmic specification for the BLM, which expresses telecommunications consumption in per worker terms as a function of the candidate predictors (or any transformation of them):

$$\begin{aligned}\ln\left(\frac{Y_{t,j}}{EMH_t}\right) &= \beta_0+\beta_1\ \ln\left(\frac{BUS_t}{EMH_t}\right)+\beta_2\ln\left(\frac{HUN_t}{EMH_t}\right)+\beta_3\ln\left(\frac{PBX_t}{EMH_t}\right)+\\ &+ \beta_4\ \ln\left(\frac{CTX_t}{EMH_t}\right)+\beta_5\ \ln\left(\frac{SAL_t}{EMH_t}\right)+\beta_6\ \ln\left(\frac{EMT_t}{EMH_t}\right)+\\ &+ \beta_7\ \ln\left(\frac{SQFT_t}{EMH_t}\right)+\beta_8\ \ln\left(POP_t\right)+\delta_1 IMILLS_t+\\ &+ \delta_2\ STAGEI_t+\delta_3\ BSOUTH_t+\sum_{h=4}\delta_h STATE_{t,h}+u_t \qquad (5.3.1)\end{aligned}$$

The term u_t is an *i.i.d.* disturbance, *IMILLS* is the inverse of the Mills ratio (which is explained in section 5.6.1) and $Y_{t,j}$ represents alternatively the total local bill ($j=1$), the intra-LATA minutes ($j=2$) or the inter-LATA minutes ($j=3$). In the following sections we discuss how to obtain possible URM models, from which we can choose a final URM*. Without loss of generality, equation (5.3.1) may be expressed in a more compact form using matrix notation as follows:

$$\ln\left(\frac{Y_j}{EMH}\right)=X'\beta+F'\delta+u \qquad (5.3.2)$$

Where:

X: is a matrix which includes ln(BUS/EMH), ln(HUN/EMH), ln(PBX/EMH), ln(CTX/EMH), ln(SAL/EMH), ln(EMT/EMH), ln($SQFT/EMH$), ln(POP).

F: is a matrix which includes *STATE*, *IMILLS*, *BSOUTH*, and *STAGEI*.

u: is a $T\times 1$ vector of *i.i.d.* disturbance term.

In general, X represents a matrix of predictors susceptible to be transformed by *level one* (see section 3.1.1), while F represents a matrix of predictors that the researcher wants to enter "as it is" in the initial specification of the model. If we allow for *level one* transformations of X we may generalize 5.3.2 as follows:

$$\ln\left(\frac{Y_j}{EMH}\right)=W'\beta+F'\delta+u \qquad (5.3.3)$$

where $W=X_r^{\alpha}X_s^{\beta}$ with $r,s=(1,\ldots,P)$ and $\alpha,\beta=-1,0,1$. Here P is the total number of untransformed continuous inputs. The main difference with respect to the BLM specified in (5.3.1) is that here we allow transformations of the original regressors, while the BLM exclusively considers logs of ratios of variables per worker. We use (5.3.3) because we want RETINA to generate the W transforms and identify which ones may help to predict better than the BLM. We can further generalize equation (5.3.3) by using the dummy variables included in F to model group-specific slopes and allowing interactions between such dummy variables and the continuous regressors. Formally, assume H_g to be a subset matrix of F with $g-1$ columns, which represents some specific grouping which accounts for heterogeneity in the data set. This leads to:

$$\ln\left(\frac{Y_j}{EMH}\right) = W'\beta + [H_g \times W]'\beta_h + F'\delta + u \tag{5.3.4}$$

with: $H_g \subset F$.
This specification is akin to an analysis of covariance formulation where the parameters of W may vary across the categories by using dummy indicators included in F to model group-specific constants, or in H_g to model group-specific slopes. In our case H_g predictors are obtained clustering the data using finite mixture of regression. Details about this step may be found in the appendix.

5.4 Results

In this section we present the main results of this study. For the sake of brevity, details about estimations are reported in the appendix as well as the description of the partitions found using finite mixtures framework. We may summarize the results obtained so far as follows:

- In Table 5.5 we report summary statistics of the Benchmark Linear Models in comparison with the final URM specifications suggested by RETINA. They show that modeling heterogeneity and non-linearities substantially increases the overall fit and predictive ability of the estimated models with respect to the correspondent BLM's. The $\bar{R}^2$ increases for all the proposed models which is a significant improvement in in-sample-fit. Also the RCMSPE drops to between one half and one third of the benchmark model, which is a marked improvement in the out-of-sample forecast ability.

- Suggested specifications include access equipment variables (table 5.4.5). Relevant first order effects for medium distance (intra-LATA) and for long distance calls (inter-LATA) are single lines access[6], whereas local demand additionally includes network access equipment variables[7] in the final specification[8]. As expected, the signs of these effects are positive.

[6] $\ln(BUS/EMH)$ and $\ln(HUN/EMH)$.

[7] $\ln(PBX/EMH)$ and $\ln(CTX/EMH)$.

[8] URM$_6$: Table 5.19.

Table 5.5: Comparison of Benchmark Linear Models (BLM) and Useful Representative Models (URM)†.

	Local (n=4391)		Intra-LATA (n=1261)		Inter-LATA (n=1176)	
	BLM	URM_6	BLM	URM	BLM	URM
Parameters	19	41	18	5	16	10
$\bar{R}^2$	.682	.930	.191	.711	.243	.730
Std.Err. Estimate	.592	.278	1.589	.950	1.369	.818
Robust CMSPE	.595	.286	1.619	.955	1.392	.827
AIC	-4590	-11207	1188	-122	757	-463
BIC	-4462	-10945	1285	-91	843	-412

† Here we use non-weighted models for direct comparison between BLM and URM. The overall fit of the estimated URM models improves with respect to the corresponding BLM's.

- The specification of the three telecommunication demands never includes the physical extension of the firm[9].

- First order effects never include the output of the firm (SAL) in the final specification. However this variable appears in second order terms.

- There are significant pairwise interactions between access equipment variables[10] for local demand and between single access systems[11] for inter-LATA demand. The signs are always negative.

- Heterogeneity parameters estimated via finite mixtures are always included in the demand functions, in the form of specific constants or slopes.

- These heterogeneity parameters also influence elasticity of the demands with respect to the relevant predictors. We observe that access form variables, namely single access lines (Business and Hunting lines) and network accesses (PBX trunks and Centrex), produce larger relative variations in demand than the remaining explanatory variables.

The above results suggest that:

1. Access equipment variables are good predictors of telecommunication demand.

[9] $\ln(SQFT/EMH)$.

[10] $\ln(BUS)$, $\ln(HUN)$, $\ln(PBX)$, $\ln(CTX)$.

[11] $\ln(BUS)\ln(HUN)$.

Table 5.6: Final suggested specifications. (t-statistics in parentheses)

Local demand

$$
\begin{aligned}
\ln\left(\frac{\widehat{LOCAL}}{EMH}\right) = {} & \underset{(102.99)}{2.630} + \underset{(79.07)}{2.722} \ln(EMH) + \underset{(82.60)}{1.098} \ln\left(\frac{BUS}{EMH}\right) + \underset{(52.15)}{.694} \ln\left(\frac{HUN}{EMH}\right) \\
& + \underset{(89.10)}{1.219} \ln\left(\frac{PBX}{EMH}\right) + \underset{(101.88)}{.774} \ln\left(\frac{CTX}{EMH}\right) + \underset{(6.84)}{.184} \ln\left(\frac{EMH}{SAL}\right) \\
& - \underset{(-25.99)}{.194} \ln(BUS)\ln(HUN) - \underset{(-6.34)}{.214} \ln(BUS)\ln(PBX) \\
& - \underset{(-36.98)}{.204} \ln(HUN)\ln(PBX) - \underset{(-35.95)}{.143} \ln(HUN)\ln(CTX) \\
& + \underset{(8.19)}{.012} \ln(EMH)\ln(EMT) + \underset{(12.70)}{.179}\, BSOUTH + \underset{(14.58)}{.262}\, AL + \underset{(6.34)}{.107}\, GA \\
& + \underset{(6.97)}{.155}\, KY + \underset{(15.97)}{.371}\, LA + \underset{(19.78)}{.445}\, MS + \underset{(11.09)}{.262}\, SC + \underset{(1.49)}{.249}\, TN
\end{aligned}
$$

$n = 4391$ $\quad \bar{R}^2 = .891$ $\quad \hat{\sigma} = .346$ $\quad$ RCMSPE (1000) = .349

$\sum \hat{\varepsilon}^2 = 522.485$ $\quad$ AIC = −9305 $\quad$ BIC = −9171

intra-LATA

$$
\ln\left(\frac{\widehat{INTRA}}{EMH}\right) = \underset{(51.90)}{3.015} + \underset{(23.88)}{.662} \ln\left(\frac{BUS}{EMH}\right) - \underset{(-45.75)}{2.637}\, H_1 - \underset{(-3.36)}{.203}\, BSOUTH - \underset{(-14.80)}{1.205}\, LA
$$

Weighted Statistics: $n = 1261$ $\quad \bar{R}^2 = .701$ $\quad \hat{\sigma} = 1.766$

Non Weighted Statistics

$n = 1261$ $\quad \bar{R}^2 = .711$ $\quad \hat{\sigma} = .950$ $\quad$ RCMSPE (1000) = .955

$\sum \hat{\varepsilon}^2 = 1134.094$ $\quad$ AIC = −121.755 $\quad$ BIC = −90.918

inter-LATA

$$
\begin{aligned}
\ln\left(\frac{\widehat{INTER}}{EMH}\right) = {} & \underset{(44.27)}{3.858} + \underset{(7.49)}{.481} \ln\left(\frac{BUS}{EMH}\right) + \underset{(3.85)}{.234} \ln\left(\frac{HUN}{EMH}\right) - \underset{(-24.03)}{2.051}\, H_1 \\
& + \underset{(9.20)}{.626} \ln(HUN)^2 + \underset{(2.19)}{.001} \ln(POP)^2 - \underset{(-8.01)}{.397} \ln(BUS)\ \ln(HUN) \\
& - \underset{(-3.29)}{.787}\, H_1 \frac{\ln(SAL)}{\ln(POP)} + \underset{(5.05)}{.840}\, H_1 \ln\left(\frac{EMT}{EMH}\right)
\end{aligned}
$$

Weighted Statistics: $n = 1176$ $\quad \bar{R}^2 = .733$ $\quad \hat{\sigma} = 1.774$

Non Weighted Statistics: (5.4.1)

$n = 1176$ $\quad \bar{R}^2 = .730$ $\quad \hat{\sigma} = .818$ $\quad$ RCMSPE (1000) = .827

$\sum \hat{\varepsilon}^2 = 780$ $\quad$ AIC = −463 $\quad$ BIC = −412

2. Interactions between different telephone access equipments, are not negligible.

3. The sales account only for a small proportion of explained variance for the proposed models, since their effects are second order.

4. Heterogeneity needs to be taken into account to represent the data and evaluate leading elasticities with respect to the relevant inputs.

We now discuss the details relative on the findings.

5.4.1 BLM Demand Models

In Table 5.12 in the Appendix, we report the Benchmark Linear Models for local, intra-LATA and inter-LATA demand. The estimations show that:

1. Demands appear to be sensitive to equipment variables (*BUS, HUN, PBX, CTX*).

2. Constant terms for intra-LATA and inter-LATA are not significant.

3. The sales (*SAL*) variables have wrong signs. This may be due to heterogeneity (see Figures 5.2, 5.3 and 5.4).

4. The Stage I indicator is negative for local calls, confirming that firms at stages higher than the first make a more intensive use of local services by routing long distance calls over their private network (*PBX*, *CTX*).

5. Dimension of the firm appears to be relevant for local services demand, again reflecting the fact that larger-sized firms demand *ceteris paribus* use more local services than firms at stage I.

The sample fit for local calls is quite satisfactory, ($\bar{R}^2 = .682$), but this is not the case of intra-LATA ($\bar{R}^2 = .191$) and inter-LATA demands ($\bar{R}^2 = .243$). These results suggest that alternative specifications should be taken into account.

5.4.2 Local Demand URM

Summary statistics for a set of alternative specifications of local demand are reported in Table 5.7. The final selected model is URM$_6$ which has been chosen among six possible URM's suggested by RETINA by varying the inputs as detailed in Table 5.17. We start by defining a new specification, say URM$_1$, and adding to the BLM the heterogeneity parameters of the optimal three-cluster solution. In Table 5.7 we see that URM$_1$ slightly improves predictive ability with respect to the BLM and $\bar{R}^2$ increases from .682 (BLM) to .708 (URM$_1$).

However substantial improvement in prediction is achieved with the use of W transformations generated by RETINA. This is the case of URM$_2$, which includes W transforms of worker per capita log-ratios. With 27 parameters URM$_2$ has an $\bar{R}^2 = .883$, thus explaining an increased variance of about 20% with respect to the BLM and about 18% with respect to URM$_1$. Out of sample predictive ability, measured by the Robust Cross Mean Square Prediction Error (RCMSPE)[12] increases substantially (about 60% of the BLM) as do the information statistics (AIC, BIC). Perhaps the most interesting results are obtained for URM$_3$ and URM$_4$ in which we exclude all mixture heterogeneity parameters and just use per capita log-ratios together with W transforms of the logs of the original variables. Both models slightly outperform URM$_2$, in terms of predictive ability without using mixture heterogeneity parameters. URM$_3$ is a very appealing specification suggested by RETINA because it has just 20 parameters, almost as many as the number of parameters of the BLM (19), while URM$_4$ has 27 parameters and shows a modest forecasting improvement with respect to URM$_3$. We can say more about URM$_3$ by looking at its specification in table 5.4.5. Note that RETINA suggests that interaction effects are not negligible for the final specification. Selected W transformations mainly involve interactions between different types of lines: $\ln(BUS)\ln(HUN)$, $\ln(BUS)\ln(PBX)$, $\ln(HUN)\ln(PBX)$ and $\ln(HUN)\ln(CTX)$. All of them have negative signs indicating a negative impact on demand. Ramsey's RESET test Ramsey (1969) was computed for URM$_3$ to test departure from the null hypothesis of correct model specification. With an $F(2, 4369) = 105.24$ the null hypothesis of correct specification is rejected, suggesting that there is room to improve the results. Just as URM$_3$, URM$_4$ is still not well specified, RESET F (2, 4363) = 69.74, and thus we reject the null hypothesis of correct specification. A natural way to re-specify URM$_3$ is to add heterogeneity parameters suggested by finite mixtures. Both URM$_5$ and URM$_6$ incorporate two-cluster and three-cluster mixture parameters, respectively. Estimates of URM$_6$ are shown in Table 5.18. Inclusion of heterogeneity parameters improves prediction ability at the expense of having a larger number of parameters (37 and 41 for URM$_5$ and URM$_6$, respectively). But this gain in prediction ability is larger than the loss in precision of the estimates since AIC and BIC statistics both show evidence in favor of URM$_5$ and URM$_6$ over previous models. URM$_6$ has an $\bar{R}^2$ of .930 and RCMSPE which is about half that of the BLM. Both models include line-equipment

[12] See Marinucci (2005) for details on RCMSPE.

Table 5.7: Local Demand: Comparison of selected statistics of candidate URM models with respect to the BLM†.

Specification	**BLM**	URM$_1$	URM$_2$	URM$_3$	URM$_4$	URM$_5$	URM$_6$
No. of Parameters	19	21	27	20	26	37	41
No. of Clusters	1	3	3	1	1	2	3
RETINA Selection	No	Yes	Yes	Yes	Yes	Yes	Yes
W Transforms	No	No	Yes	Yes	Yes	Yes	Yes
Specific Constants	No	Yes	Yes	No	No	Yes	Yes
Specific Slopes	No	No	No	No	No	Yes	Yes
$\bar{R}^2$	.682	.708	.883	.891	.896	.912	.930
$\hat{\sigma}$	.592	.567	.359	.346	.339	.312	.278
RCMSPE(1000)^a	.595	.571	.365	.349	.343	.317	.286
$\sum \hat{\varepsilon}^2$	1530	1403	562	522	500	423	336
AIC^b	−4590	−4965	−8973	−9305	−9484	−10199	−11207
BIC^c	−4462	−4824	−8794	−9171	−9312	−9956	−10939

†Different URM models have been selected by RETINA using different initial specifications for X,F and H. Details are reported in Table 5.17.

- URM$_1$: Obtained starting with BLM + three specific constants corresponding to the optimal S_1, G_3 three cluster solution.
- URM$_2$: As in URM$_1$ + W transforms.
- URM$_3$: Here heterogeneity mixture parameters are excluded. Auxiliary log-transforms of original variables (*BUS, HUN, PBX, CTX, EMT, EMH, SQFT*) are used to generate W transforms and original log-ratios are included in F, the untransformed inputs.
- URM$_4$: A different specification proposed by RETINA using the same specification as in URM$_3$.
- URM$_5$: As in URM$_3$, but this time allowing heterogeneity parameters corresponding to the S_1, G_2 two-cluster solution.
- URM$_6$: As in URM$_3$ and including heterogeneity parameters of the S_1, G_3 optimal three-cluster solution.

a. Robust Cross Mean Square Prediction Error is an approximation of the out of sample $\hat{\sigma}^2$ using 1000 bootstrap random selection of three disjointed sub-samples. See Marinucci (2005) for details.

b. Here AIC is specified as $n \ln(\hat{\sigma_\varepsilon}^2) + 2k$, where n is the sample size and k is the number of parameters.

c. Here BIC is specified as $n \ln(\hat{\sigma_\varepsilon}^2) + \ln(n)k$, where n is the sample size and k is the number of parameters.

interactions as in URM$_3$ (see Table 5.18), but additional demand variation is modeled by cluster-specific slopes, namely regressors that are selected by RETINA from the $[H_g \times W]$ term of equation (5.3.4). Going back to Table 5.7, the out of sample prediction ability (RCMSPE) is only 48% of the BLM for URM$_6$ (.286/.595) and 53% for URM$_5$ (.317/.595) while it is 59% for URM$_3$ (.349/.595). Almost all of the variables already used in the earlier specification of the BLM are included in the URM. These are: Business Lines (BUS), Hunting Lines (HUN), PBX trunks (PBX) and Centrex lines (CTX). RESET test for URM$_6$ gave $F(2, 4349) = 1.10$ which does not reject the null hypothesis of correct specification. In URM$_6$, the specification suggested by RETINA includes untransformed variables as well as interactions and cross-ratios between them. Equipment variables (such as type and number of lines) have non-linear effects on demand. Non-linearities may arise due to a variety of reasons including the unavailability of other relevant variables such as the nature of the business activity or whether usage is primarily internal or external. In order to capture the above mentioned non-linearities, the proposed URM$_6$ for local services includes a variety of transformations that go beyond the a priori specification of the BLM. Final WLS estimations that incorporate heteroskedasticity correction of URM$_6$ are shown in Table 5.19. $F - tests$ for variable exclusion were also carried out, since some of the initial 41 variables were no longer significant, finally reducing the number of parameters of URM$_6$ from 41 to 37.

5.4.3 Intra-LATA URM

The intra-LATA and inter-LATA demand results are quite different. As seen from Table 5.12, both BLM's show relatively poor fits and high standard errors of the estimation over the whole data set. The estimations suggest that both demands are sensitive to the number of single-line accesses in the business. Moreover we observe that the constant term in both BLM's is not significant. The negative sign of the (EMT/EMH) coefficients is due to the fact that both medium and long distance services are mostly demanded by single-location and small sized firms. Since these results are somewhat unsatisfactory from the prediction point of view, we apply here a selection strategy similar to the one used for local demand. For the sake of brevity, for intra-LATA and inter-LATA demand we report just the final selected URM models. The final selected Useful Representative Model (URM) for intra-LATA minutes is reported in equation 6: RETINA selects a very simple formulation as URM of intra-LATA demand. This model has only 5 parameters and, with the inclusion of just one specific constant for cluster 1 (H_1), we take into account heterogeneity in the data set. Moreover the Bell South effect is negative, reflecting the fact that intra-LATA services tend to be provided by alternative companies. But perhaps the most interesting characteristic of the inter-LATA demand model concerns the $\ln(BUS/EMH)$ ratio, which represents the effect of basic single line access demand. In other words, intra-LATA demand is found to be especially sensitive to the number of business lines, while the effect of the other variables negligible. The model passes the RESET specification test; with F(2,1254) = .958 we do not reject the null hypothesis of correct specification. Then we applied weighted OLS to correct for heteroscedasticity. The $\bar{R}^2$ of the intra-LATA URM increases from .191 to .711 (.701 for weighted estimation), while the RCMSPE is about a fifth of the BLM corresponding value. Also the standard error of estimate is about 60% with respect to the corresponding BLM value. This model shows very appealing features because its specification includes only five variables in modeling the demand of intra-LATA calls. With respect to the corresponding BLM, we gained in terms of predictive ability and also in terms of a more parsimonious representation.

5.4.4 Inter-LATA URM

For inter-LATA demand, we also obtain a quite parsimonious representation with just 9 parameters, after considering a set of potential URM candidates suggested by RETINA. The selected URM for inter-LATA minutes has been estimated by WLS for heteroscedasticity correction. Here, significant effects are provided by the number of lines per capita, namely the number of business (BUS) and hunting (HUN) lines. Also their interaction is relevant, as well as the square of $\ln(HUN)$. Again, these interactions have negative signs. $\bar{R}^2$ is .730 versus .243 of the corresponding BLM, and RCMSPE (.827) is only 59% of the corresponding BLM (1.392) value. The model suggested by RETINA is a very significant improvement over the BLM.

5.4.5 Elasticities

We are interested in evaluating the leading elasticities both for local and inter-LATA final URM models. In the case of the intra-LATA URM, since the demand specification is very simple, we do not need to make further calculations to evaluate the elasticities because the corresponding coefficients may be interpreted directly. Elasticity of intra-LATA demand with respect to the number of business lines is .66 (see eq. 6). On the other hand, evaluation of the local and inter-LATA elasticities is more tedious because the respective URM's often embed nonlinear transformations of the inputs. As a consequence, expressions for the elasticities of the local and the

Table 5.8: Selected Elasticities based on URM estimates†.

	BUS	*HUN*	*PBX*	CTX	SAL	EMT	SQFT	POP
Local ($)	1.02	.21	.87	.74	-.04	.02	.00	.00
intra-LATA (min.)	.66	-	-	-	-	-	-	-
inter-LATA (min.)	.29	.44	-	-	.03	.36	-	.03

† See Tables 5.20 and 5.9 for elasticity expressions of local and inter-LATA demand, respectively.

Table 5.9: Selected Cross-Elasticities from inter-LATA URM weighted model (eq.7).

$$\frac{\partial \ln(INTER)}{\partial \ln(BUS) \partial \ln(HUN)} = -\frac{.397}{BUS \cdot HUN} < 0$$

$$\frac{\partial \ln(LOCAL)}{\partial \ln(BUS) \partial \ln(PBX)} = -\frac{.200}{PBX} < 0$$

$$\frac{\partial \ln(LOCAL)}{\partial \ln(HUN) \partial \ln(CTX)} = -\frac{.540}{CTX \cdot HUN} < 0$$

$$\frac{\partial \ln(LOCAL)}{\partial \ln(HUN) \partial \ln(PBX)} = \frac{-.111 + .100 H_1}{PBX \cdot HUN} \lesseqgtr 0$$

inter-LATA URM also embed heterogeneity parameters and other non linearities represented by further transformations of the inputs, as shown in Table 5.20 and Table 5.21. Note that the reported expressions in most cases depend on the values assumed by other variables. We evaluate the elasticities at the average values of the influencing variables. The results are shown in Table 5.8. For local demand, we found larger positive elasticities for telephone equipment of the firm. Elasticities with respect to the number of basic accesses, namely the number of business lines, is close to one. Elasticities with respect to network access forms, PBX trunks and Centrex lines, are .87 and .74, respectively. Demand elasticities are quite irrelevant for the other explanatory variables, including the number of workers in the firm (*EMH, EMT*), sales (*SAL*), and physical extension (*SQFT*) and population habitat size (*POP*). Single line access forms were also positively related to demand for inter-LATA services. Elasticity is .29 for business lines and .44 for hunting lines. On the other hand, the elasticity with respect to the total number of employees is .36.

5.5 Conclusions

In this chapter we estimate business telecommunications demands for local, intra-LATA and inter-LATA services using US Telecommunications data. Graphical bivariate analysis and Benchmark Linear Model estimation show strong evidence of heterogeneity which must be modeled in order to achieve a useful representation of the data. We achieve this goal by first using finite mixtures of normal heteroscedastic components to partition the data into homogeneous subgroups. For local demand we fit three components, while two components were fitted both for intra-LATA and inter-LATA demand. We then perform an automatic model search using the RETINA algorithm to obtain a flexible model useful for out of sample prediction. RETINA generates an expanded regressor set using the firm group membership as a heterogeneity parameter to estimate specific constants and specific slopes. In addition RETINA includes interactions and nonlinear transformations of the original variables as candidate regressors. We find that telephone equipment variables are almost always selected as relevant first order effects. Moreover, the corresponding coefficients are always positive. Also heterogeneity parameters and negative interactions between different forms of access are significant and play an important role in demand prediction. As a result, the demand elasticities, evaluated for the relevant variables at the average values, show that:

- Local calls demand is most sensitive to a relative variation of the number of business lines (1.02) and network access equipment (PBX Trunks (.87) and Centrex (.74)), while a change in the remaining explanatory variables is not significantly linked to relative variations of demand.

- Intra-LATA demand was sensitive only to single line access equipment represented by the number of business lines (elasticity is .66), while the effect of most of the remaining explanatory variables was negligible.

- Inter-LATA demand elasticity is positive with respect to business lines (.29) and hunting lines (.44) but also shows a positive relationship with respect to the total number of workers of the whole business (.36).

With these results we are tempted to claim that modeling of business telecommunications demand using RETINA for this data set is adequate for its intended primary use of out of sample forecasting.

5.6 Appendices

5.6.1 Data pre-processing

Prior to the model specification a large preprocessing stage was undertaken. Only 4463 observations had complete data and our effective sample size varies along with the type of demand[13]. Local services are used by all the firms, while intra-LATA and inter-LATA services have been used by only 29% and 27% of the businesses, respectively. Descriptive analysis and estimations were carried out twice, first by using complete records, and then by using the total data set of 13743 observations where missing information is imputed with a method suggested by Troyanskaya, Cantor, Sherlock, Brown, Hastie, Tibshirani, D. & Altman (2001). In general, results over the imputed data set differed slightly with respect to the results obtained over the reduced record set, and the results are not reported here. In modeling intra-LATA and inter-LATA demand, PBX trunks (PBX) and Centrex lines (CTX) have been excluded a priori from the respective BLM's due to the lack of variation across the considered sample. Also, for the same areas, the Alabama and South Connecticut dummy indicators have been excluded from the specification of inter-LATA traffic. Furthermore, in order to avoid perfect collinearity between the State dummy indicators, Florida is taken as the reference level and is always removed from the analysis. Finally all the equipment variables such as business lines (BUS), hunting lines (HUN), PBX trunks (PBX) and Centrex lines (CTX) have been augmented by one because these variables present zero values, and therefore log transformations would be undefined. Outliers were detected by using an automated procedure proposed by Peña & Yohai (1999). The procedure is implemented in RETINA Winpack[14] and may be run optionally by the user prior to model selection. This reduced the effective sample size of local demand to 4391 firms, while 1261 were kept for intra-LATA demand and 1176 for inter-LATA demand. Also, prior to estimation, data have been re-scaled to avoid the potential negative effect of different orders of magnitude.

5.6.2 Estimation

Estimation of the BLM is straightforward for local traffic but not for intra-LATA and inter-LATA demand since, as seen in Section 5.2, not all firms use public carriers

[13] The variables which reported missing values were (number of missing values reported in parenthesis): $SQFT$ (9270), EMH (416), EMT (2458), SAL (2735).

[14] See section 4.2.

for this type of service. Direct estimation of the BLM by OLS, using only the sample with nonzero demand, would be inconsistent since the mean of the error estimates could be biased by sample selection effects. In these cases we first use a probit model to explain the probability of the firm having a non-zero demand. Probit analysis provides us with a new variable called the inverse of the Mills ratio (*IMILLS*). After this step we can go ahead with the OLS estimation of the BLM considering only those firms with some toll calling activity. But this time, among the regressors, we include the inverse of the Mills ratio as an explanatory variable because it adjusts the mean of the error term which is not necessarily equal to zero. Probit estimations for intra-LATA and inter-LATA demand are provided in Table 5.11.

5.6.3 Modeling Heterogeneity using Finite Mixtures

Since the presence of heterogeneity can in our case be visually detected from bivariate scatter plots as seen in Section 5.2, the problem of modeling heterogeneity may well be addressed by using available information at hand (geographic indicators, stage of the firm and so on). Nonetheless, this a priori information may not account for all the heterogeneity in the data set. Finite mixtures may then be used to detect or represent any additional group structure, if present, in the data. The only assumption in this case is that the distribution of our dependent variable may be approximated as a weighted sum of normal distributions, each of which has an expected mean expressed as a function of the explanatory variables, without loss of generality if we define:

$$\ln\left(\frac{Y_j}{EMH}\right) = W'\beta_g + F'\delta_g + \sigma_g\, u$$

where $u \sim N(0,1)$ and $\beta_g, \delta_g, \sigma_g$ may assume different G values with probabilities $(\pi_1, \ldots, \pi_G)$: the conditional distribution of the dependent variable with respect to the candidate regressors may then be expressed formally as a mixture of G components as:

$$\ln(Y_j/EMH) \mid W, F \;\sim \sum_{g=1}^{G} \pi_g\, N(\, W'\beta_g + F'\delta_g, \sigma_g^2)$$

Using this formulation[15], the Expectation Maximization (EM) algorithm[16] is then used to estimate the maximum likelihood parameters of the regression equations of each group $\hat{\beta}_g, \hat{\delta}_g, \hat{\sigma}_g$, and the posterior probabilities $\hat{\pi}_g$ for each firm. Cluster

[15] Note that we are assuming normal heteroscedastic components. See Appendix 5.6.1 for more details.

[16] See McLachlan & Peel (2000) for a discussion on finite mixture modeling.

membership (the H_g matrix) is then determined by assigning each observation to the group for which posterior probability is highest. What is relevant here is that this methodology allows us to obtain a consistent inference about H_g better suited to our objectives than any other traditional non-parametric clustering method, eg. K-means (MacQueen (1967)), Ward (Ward Jr. (1963)). Traditional clustering methods are concerned with grouping objects, in our case the firms, by minimizing some distance measure among them. The distance measures are defined on the basis of a specific metric which typically is chosen by the researcher on an a priori basis (Euclidean distance is usually considered). Thus traditional clustering methods do not involve the estimation of any a priori parametric model structure on the variables. With Finite Mixtures, on the contrary, distributional assumptions and conditional heterogeneity among the variables, rather than unconditional heterogeneity, are explicitly taken into account and a parametric (or semi-parametric) inference about a specific partition model is possible. We fitted a number of Gaussian mixture models to capture additional sources of variation for each demand. We specify the dependent variable to be distributed as a mixture of normal distributions with heteroscedastic components allowing different variances for each component. Indeed, there are many different initial specifications that may be used for clustering our data via finite mixtures. Moreover, within each specification, the number of groups of the resulting partition must be assessed after estimations. Interested readers may refer to Table 5.10 for details. When heteroscedastic components are specified, the likelihood function is unbounded for the component covariances, which in turn implies that a global maximizer does not exist, (see McLachlan & Peel (2000)). This means that great care must be taken in order to ensure that the provided estimations do not correspond to a spurious local solution on the edge of the parameter space for σ_g, $g = (1 \ldots G)$, which should be discarded. For this reason, we compared a wide range of solutions by using different strategies to select the starting parameter values. For their definition, we used both $k - means$ clustering (MacQueen 1967) and 100 random initial partitions of the original data set. Using this strategy we fitted up to 5 groups for each initial model specification relative to each demand. As regards the number of groups to be retained for subsequent analysis, since regularity conditions do not hold for the log-likelihood function, usual likelihood ratio tests cannot be applied here. Thus the decision on the number of partitions to retain is based on information criteria (both AIC and BIC in our case) as well as on an a priori hypothesis about a two-cluster structure especially for local and intra-LATA demand. As discussed in the foot note of Table 5.10, only in the case of inter-LATA demand was there a need to assess a two cluster structure using a bootstrap likelihood ratio test. Computations were carried out using the *Flexmix* (Leisch 2003) and the *Mixreg* packages designed for the R software (R Development Core Team 2007). For the model selection step, we used RETINA Winpack which allows to perform model selection and estimate a variety of econometric models including those corresponding to equations (5.3.2), (5.3.3) and (5.3.4).

In summary we used a two-step approach to obtain a set of possible URM models from which we can choose a final URM*. The first step is to model heterogeneity, fitting finite mixtures to each demand. The second is to perform a variable selection on an expanded regressors set which, besides the original variables, also includes transformations of the form $X_r^\alpha \; X_s^\beta$ as well as heterogeneity parameters. More specifically:

1. First fit a mixture of regressions for each demand and for each proposed initial specification by estimating the maximum likelihood mixture parameters via the EM algorithm.

2. Decide the number G of clusters to be retained for subsequent analysis in each case (we use AIC and BIC).

3. Obtain the corresponding H_g matrixes (if any) by assigning each observation to the cluster for which the posterior cluster membership probability is highest.

4. Once a partition has been chosen, consider a general specification as in equation (5.3.4) but this time including the cluster membership matrix H_g:

   - into F in order to model group-specific constants
   - into H_g in order to model group-specific slopes

5. Then use RETINA to automatically select only the most relevant predictors among W, F, and the $H_g \times W$ interactions between predictors and clusters. Obtain a candidate URM*.

This approach works well in practice. One can get different candidate URM's by running the above steps for different specifications of the inputs, namely X, F and H. All of them represent a *candidate model set* on which Multi-Model Inference, MMI (Burnham & Anderson 2002) may be carried out by comparing the models on the basis of AIC and BIC criterion.

5.6.4 Mixtures of Linear Demand Models

After applying the EM algorithm, the number of groups was selected by examining both the AIC and BIC criteria over three different specifications, say S_1, S_2 and S_3 where $S_1 \equiv$ BLM, S_2 the relevant regressors of the BLM selected by RETINA, and S_3 the BLM excluding all dummy variables. The AIC and BIC statistics of the fitted mixture of linear demand models using S_1, S_2 and S_3 as initial specification are reported in Table 5.10[17].

Strong evidence for a two group solution was found for intra-LATA demand using the S_2 specification suggested by RETINA, while for local calls we adopted both a two cluster and a three cluster solution using the S_1 BLM specification. For inter-LATA demand there is apparently weaker evidence of heterogeneity although finally a bootstrap likelihood ratio test was finally used to choose a two group structure using the S_2 specification proposed by RETINA. The estimated models are reported in Tables 5.13, 5.14, 5.15 and 5.16. Interestingly the results show that most differences among clusters can be captured by differences in constants. For example, while in the intra-LATA or inter-LATA BLM's the constant term was not statistically significant (see Table 5.12), homogeneous clusters found by using mixture modeling show significant variations across the constant terms of two groups (see Table 5.15 for intra-LATA and Table 5.16 for inter-LATA demand). Basically, this means that firms belonging to clusters with higher constants may be "heavy users", while components with lower constants may be "light users" of the service. Other differences among groups are associated with component variances and slope parameter estimates. Interestingly we find a close relationship between these results and the descriptive statistics shown in Table 5.1. For example, consider the coefficient of $\ln(EMT/EMH)$ for inter-LATA demand (Table 5.16). This parameter gives an indication of the effect of the relative size of the local subsidiary with respect to the whole business. It gives an approximation of the dimension of the firm's internal communication needs. As we can see from Table 5.16 inter-LATA "heavy users" (cluster 2) are not sensitive to the $\ln(EMT/EMH)$ ratio since the corresponding coefficient is not significantly different from zero. This reflects the fact that "heavy users" of inter-LATA service are mostly stage I firms, which are smaller and single location firms. In fact, since the proportion of single location firms is higher in this cluster, EMT tends to EMH and this causes the $\ln(EMT/EMH)$ ratio to tend towards zero. More evidence of heterogeneity is reported in Table 5.14. Here, local services demand is decomposed

[17] See also Appendix 5.6.1 for more details about the justification of using AIC and BIC as selection criterion for mixture models.

Table 5.10: Model selection of BLMM (Benchmark Linear Mixture Models)†.

AIC Statistic									
Local				*intra − LATA*			*inter − LATA*		
Groups	S_1	S_2	S_3	S_1	S_2	S_3	S_1	S_2	S_3
1	7871	8161	8628	4766	4774	4796	4094	4170	4139
2	4155	4960	5415	4721	4730*	4766	4067	4139*	4116
3	3403	4405	4823	4674	4734	4769	4020	4149	4110
4	3300	4231	4592	4614	4739	4749	3984	4154	4101
5	3209*	4146*	4561*	4564*	4747	4744*	3928*	4160	4092*
BIC Statistic									
Local				*intra − LATA*			*inter − LATA*		
Groups	S_1	S_2	S_3	S_1	S_2	S_3	S_1	S_2	S_3
1	7999	8238	8692	4864*	4795	4842*	4180*	4190	4185*
2	4417	5145	5550	4922	4776*	4864	4244	4184*	4212
3	3799*	4648	5027	4978	4806	4918	4288	4220	4257
4	3831	4556*	4866*	5020	4837	4949	4344	4250	4299
5	3873	4614	4906	5073	4870	4996	4379	4260	4340

†In Table 5.10 we show up to five groups solution for each demand. Solutions were obtained using both $k - means$ starting values and 100 random starting values for each partition. Values marked with an asterisk represent the lowest values of AIC and BIC along specifications S_1, S_2 and S_3. More in detail:

S_1 : An initial specification as in equation (5.3.1). This is adopted as a natural starting point, since it is the BLM specification.

S_2 : An initial specification suggested by performing a variables selection on eq. (5.3.1). Here we choose a more parsimonious specification than the BLM, where selected regressors were suggested by RETINA.

S_3 : An initial specification using a specification as in (5.3.1) but excluding all the dummy variables. This is just an additional specification allowing only continuous regressors.

Solutions proposed by both criteria do not generally coincide. BIC criterion is in general the preferred statistic since AIC has been observed to over-estimate the number of components (McLachlan & Peel 2000). In fact AIC tends to suggest a higher dimensional solution excluding some special cases. The lowest AIC value for local demand models corresponds to a five group solution ($\text{AIC}_{LOCAL,G5,S1} = 3209$) while $\text{BIC}_{LOCAL,G5,S1} = 3799$ suggests a three group solution using specification S_1. Note that these are the lowest values with respect to alternative group solutions, but also with respect to specifications S_2 and S_3. Nonetheless, since a two cluster solution is visually expected we also take into consideration a two group solution for subsequent steps. A similar reasoning is applied to intra-LATA demand. We find evidence for a two groups solution since the lowest BIC statistic across alternative specifications corresponds to S_2 for which [18] $\text{BIC}_{INTRA,G5,S1} = 4776$. The choice of the number of groups is more difficult in the inter-LATA case. Heterogeneity is not strongly supported as in the case of local and intra-LATA demand, since lowest information statistics provide opposite results: we find the lowest $\text{AIC}_{INTER,G5,S1} = 3928$ suggests 5 groups using specification S_1, but $\text{BIC}_{INTER,G1,S1} = 4180$ suggests evidence in favor of absence of heterogeneity in the data proposing a one-cluster solution. Nonetheless we observe that the second best solution is for the two groups S_2 specification, for which $\text{BIC}_{INTER,G2,S2} = 4184$. But the differences $\text{BIC}_{INTER,G2,S2} - \text{BIC}_{INTER,G1,S1} = 4184 - 4180 = 4$ and $\text{BIC}_{INTER,G1,S2} - \text{BIC}_{INTER,G2,S2} = 4190 - 4184 = 6$ indicate only a weak evidence in favor of the S_1 and S_2 *absence of heterogeneity* model. To verify this hypothesis at least on the S_2 specification we run a bootstrapped likelihood ratio test where null hypothesis is the one group solution and the alternative is a two group solution. Departure from the null hypothesis was significant using $n = 100$ replications at $\alpha = .0001$ level thus we finally decided on a two cluster solution.

into three components: cluster 1 with a constant term of 4.112 (virtually equal to the whole sample estimate), cluster 2 with a constant term of 2.497 and cluster 3 with a constant term of 3.125. For the sake of convenience we will call cluster 1 "heavy users" cluster 2 "light users", and cluster 3 "medium users". We observe that estimated demand elasticities of single-line accesses such as business (BUS) and hunting (HUN) lines have positive signs as expected and are significant. Nonetheless, for network systems such as PBX trunks (PBX) and Centrex lines (CTX), the signs of the elasticities vary across clusters: PBX trunk elasticities are negative (-1.259) for "light users", and Centrex line elasticities are also negative (-1.229) for "medium users" - with very high $t - values$.

A final comment is due for Sales, which is the variable that proxies the firm output. Heterogeneity of demand with respect to sales (SAL) is evident from Figure 5.2, where the upward sloping cloud may suggest a positive relationship between local demand and the firm's sales. Nonetheless the estimated parameter has negative signs across clusters (Table 5.14). This suggests that the heterogeneity attributed to sales has no correlation with heterogeneity due to different access equipment in the firm, which in turn is represented by four variables (BUS, HUN, PBX, CTX) and accounts for a greater proportion of explained variance.

Table 5.11: Probit models for intra-LATA and inter-LATA demand ($t-statistics$ are reported in parenthesis)

Dependent Variable	intra-LATA (Yes=1)	inter-LATA (Yes=1)
Observations	4463	4463
log-L	−1806.962	−1343.420
restricted log-L	−2675.425	−2594.076
Chi-sq (dgf)	1736.927 (18)	2501.312 (16)
sig.	.0000	.0000
$Constant$	−.815 (−3.381)	−.954 (−3.565)
$\ln(BUS/EMH)$	.389 (11.439)	.376 (9.095)
$\ln(HUN/EMH)$	−.238 (−6.288)	−.153 (−3.590)
$\ln(PBX/EMH)$	.205 (5.929)	.313 (8.446)
$\ln(CTX/EMH)$	−.015 (−.578)	.065 (2.316)
$\ln(SAL/EMH)$	−.119 (−1.793)	−.255 (−16.439)
$\ln(EMT/EMH)$	−.102 (−1.987)	−.198 (−3.063)
$\ln(SQFT/EMH)$	−.020 (−.806)	−.026 (−.925)
$\ln(POP)$	−.006 (−.459)	.026 (1.870)
$STAGEI$	.774 (9.534)	1.053 (1.785)
$BSOUTH$	−.083 (−1.314)	.010 (.143)
AL	.503 (5.390)	–
GA	−.043 (−.514)	.103 (1.247)
KY	1.599 (15.670)	1.904 (18.568)
LA	.493 (4.597)	1.327 (12.299)
MS	.911 (9.186)	1.442 (14.330)
NC	1.347 (12.342)	2.240 (17.634)
SC	1.575 (15.117)	–
TN	1.004 (1.055)	1.720 (15.638)

Predicted counts for intra-LATA and inter-LATA probit models

intra-LATA				inter-LATA			
	Predicted				*Predicted*		
Actual	0	1	*Total*	*Actual*	0	1	*Total*
0	2890	292	3182	0	3111	156	3267
1	503	778	1281	1	364	832	1196
Total	3393	1070	4463	*Total*	3475	988	4463

Table 5.12: Benchmark Linear Models for Local, intra-LATA and inter-LATA traffic ($t - statistics$ are reported in parenthesis).

Dependent Variable	$\ln(LOCAL/EMH)$	$\ln(INTRA/EMH)$	$\ln(INTER/EMH)$
Observations	4391	1261	1176
$\bar{R}^2$	.682	.191	.243
Std.err.est.	.592	1.589	1.369
Robust CMSPE	.595	1.619	1.392
AIC	−4590	1188	757
BIC	−4462	1285	843
$Constant$	4.112 (45.115)	1.374 (1.297)	.941 (1.349)
$\ln(BUS/EMH)$	.290 (29.320)	.694 (3.515)	.713 (5.330)
$\ln(HUN/EMH)$	.168 (15.181)	.111 (1.228)	.509 (6.632)
$\ln(PBX/EMH)$	.157 (15.371)	–	–
$\ln(CTX/EMH)$	.112 (16.950)	–	–
$\ln(SAL/EMH)$	−.024 (−5.796)	.009 (.165)	−.343 (−5.985)
$\ln(EMT/EMH)$	.138 (9.520)	−.324 (−1.615)	−.188 (−.945)
$\ln(SQFT/EMH)$	.001 (.087)	.017 (.379)	.036 (.878)
$\ln(POP)$	.004 (.804)	.009 (.378)	.110 (5.577)
$IMILLS$	–	.256 (.420)	1.159 (3.591)
$STAGEI$	−.490 (−17.197)	−.064 (−.176)	.125 (.448)
$BSOUTH$	.221 (9.142)	−.355 (−2.956)	.051 (.544)
AL	.092 (2.884)	.859 (2.516)	–
GA	.047 (1.614)	.356 (1.726)	−.374 (−2.172)
KY	−.275 (−7.100)	.881 (1.383)	1.242 (3.171)
LA	.089 (2.199)	−.517 (−1.551)	.712 (2.130)
MS	.004 (.099)	.646 (1.417)	1.119 (3.221)
NC	−.598 (−13.489)	.763 (1.281)	2.156 (4.520)
SC	−.156 (−3.808)	.819 (1.289)	–
TN	−.101 (−2.450)	.435 (.920)	.923 (2.472)

Table 5.13: Local Demand: Two-cluster solution Benchmark Linear Mixture Models (BLMM)($t - statistics$ in parenthesis).

$\ln(LOCAL/EMH)$	Total Sample	cluster 1	cluster 2
Observations	4391	3287	1104
$\bar{R}^2$	.682	.962	.764
Std.err.est.	.592	.191	.566
$Constant$	4.112 (45.115)	2.482 (7.516)	4.742 (26.067)
$\ln(BUS/EMH)$	.290 (29.320)	1.178 (166.347)	.171 (7.734)
$\ln(HUN/EMH)$	.168 (15.181)	.249 (47.777)	.032 (1.796)
$\ln(PBX/EMH)$	.157 (15.371)	−1.165 (−123.813)	.341 (2.932)
$\ln(CTX/EMH)$	.112 (16.950)	.761 (137.977)	.219 (19.808)
$\ln(SAL/EMH)$	−.024 (−5.796)	−.017 (−11.296)	−.052 (−4.125)
$\ln(EMT/EMH)$	.138 (9.520)	.008 (1.171)	.169 (8.813)
$\ln(SQFT/EMH)$	.001 (.087)	.000 (−.044)	−.018 (−1.018)
$\ln(POP)$	.004 (.804)	.005 (2.676)	−.005 (−.598)
$STAGEI$	−.490 (−17.197)	.029 (2.512)	−.272 (−4.372)
$BSOUTH$	.221 (9.142)	.237 (26.234)	.033 (.696)
AL	.092 (2.884)	.406 (32.997)	−.143 (−2.274)
GA	.047 (1.614)	.243 (22.739)	−.345 (−5.098)
KY	−.275 (−7.100)	.347 (23.727)	−.324 (−4.040)
LA	.089 (2.199)	.316 (19.833)	.271 (3.559)
MS	.004 (.099)	.451 (29.652)	−.059 (−.753)
NC	−.598 (−13.489)	.034 (1.817)	−.800 (−1.211)
SC	−.156 (−3.808)	.415 (28.702)	−.623 (−5.004)
TN	−.101 (−2.450)	.355 (24.162)	−.132 (−1.216)

Table 5.14: Local Demand: Three-cluster Benchmark Linear Mixture Models (BLMM) for Local Calls Billing ($t - statistics$ in parenthesis)†.

$\ln(LOCAL/EMH)$	Total Sample	cluster 1	cluster 2	cluster 3
Observations	4391	585	2622	1184
$\bar{R}^2$	.682	.759	.972	.984
Std.err.est.	.592	.586	.163	.134
$Constant$	4.112 (45.115)	4.113 (15.347)	2.497 (73.969)	3.125 (74.760)
$\ln(BUS/EMH)$	.290 (29.320)	.146 (4.684)	1.248 (179.245)	1.046 (139.117)
$\ln(HUN/EMH)$	.168 (15.181)	.072 (3.051)	.203 (41.950)	.104 (2.523)
$\ln(PBX/EMH)$	.157 (15.371)	.247 (1.484)	−1.259 (−138.700)	1.101 (136.011)
$\ln(CTX/EMH)$	.112 (16.950)	.215 (15.284)	.841 (165.416)	−1.229 (−106.006)
$\ln(SAL/EMH)$	−.024 (−5.796)	−.054 (−2.96)	−.026 (−18.348)	−.043 (−22.712)
$\ln(EMT/EMH)$	.138 (9.520)	.256 (9.212)	−.002 (−.368)	−.017 (−2.661)
$\ln(SQFT/EMH)$	.001 (.087)	.013 (.493)	−.001 (−.156)	−.007 (−1.593)
$\ln(POP)$	.004 (.804)	.003 (.283)	.004 (2.204)	−.001 (−.527)
$STAGEI$	−.490 (−17.197)	−.040 (−.465)	.037 (3.387)	−.042 (−3.225)
$BSOUTH$	.221 (9.142)	−.165 (−2.349)	.158 (18.509)	.331 (29.423)
AL	.092 (2.884)	−.046 (−.502)	.425 (37.658)	.211 (13.002)
GA	.047 (1.614)	−.430 (−4.037)	.068 (6.296)	.391 (31.415)
KY	−.275 (−7.100)	−.382 (−2.898)	.218 (14.668)	.330 (21.140)
LA	.089 (2.199)	.481 (4.542)	.311 (2.804)	.220 (11.350)
MS	.004 (.099)	.175 (1.584)	.357 (24.757)	.451 (24.196)
NC	−.598 (−13.489)	−.584 (−5.266)	−.002 (−.116)	.242 (11.329)
SC	−.156 (−3.808)	−.444 (−2.617)	.392 (28.314)	.358 (18.019)
TN	−.101 (−2.450)	−.030 (−.213)	.299 (2.840)	.362 (18.870)

†These parameter estimates correspond to the optimal three-cluster solution of specification S_1. See Table 5.10 where the choice of the mixture regression partition is discussed for Local demand calls.

Table 5.15: Selected Benchmark Linear Mixture Models (BLMM) for intra-LATA minutes ($t - statistics$ in parenthesis).

$\ln(INTRA/EMH)$	Total Sample	cluster 1	cluster 2
Observations	1261	472	788
$\bar{R}^2$	.177	.361	.349
Std.err.est.	1.603	1.030	.902
$Constant$	1.923 (32.597)	.281 (4.451)	2.849 (68.752)
$\ln(BUS/EMH)$	.738 (15.691)	.776 (15.490)	.626 (18.713)
LA	−1.037 (−5.361)	−.973 (−4.507)	−1.244 (−9.348)

Table 5.16: Selected Benchmark Linear Mixture Models (BLMM) for inter-LATA minutes ($t - statistics$ in parenthesis).

$\ln(INTER/EMH)$	Total Sample	cluster 1	cluster 2
Observations	1176	505	665
$\bar{R}^2$	.184	.389	.396
Std.err.est.	1.422	.928	.763
$Constant$	3.404 (48.208)	2.133 (29.648)	4.354 (88.110)
$\ln(HUN/EMH)$	.693 (15.891)	.743 (16.666)	.637 (2.937)
$\ln(EMT/EMH)$	.442 (2.982)	1.153 (6.073)	−.102 (−.110)

Table 5.17: Local Demand: Specification of X,F and H inputs of RETINA †.

	URM_1	URM_2	URM_3	URM_4	URM_5	URM_6
$\ln(BUS/EMH)$	X	W	F	F	F	F
$\ln(HUN/EMH)$	X	W	F	F	F	F
$\ln(PBX/EMH)$	X	W	F	F	F	F
$\ln(CTX/EMH)$	X	W	F	F	F	F
$\ln(SAL/EMH)$	X	W	F	F	F	F
$\ln(EMT/EMH)$	X	W	F	F	F	F
$\ln(SQFT/EMH)$	X	W	F	F	F	F
$\ln(BUS)$	–	–	W	W	W	W
$\ln(HUN)$	–	–	W	W	W	W
$\ln(PBX)$	–	–	W	W	W	W
$\ln(CTX)$	–	–	W	W	W	W
$\ln(SAL)$	–	–	W	W	W	W
$\ln(EMT)$	–	–	W	W	W	W
$\ln(EMH)$	–	–	W	W	W	W
$\ln(SQFT)$	–	–	W	W	W	W
$\ln(POP)$	X	W	W	W	W	W
$STAGEI$	F	F	F	F	F	F
$BSOUTH$	F	F	F	F	F	F
AL	F	F	F	F	F	F
GA	F	F	F	F	F	F
KY	F	F	F	F	F	F
LA	F	F	F	F	F	F
MS	F	F	F	F	F	F
NC	F	F	F	F	F	F
SC	F	F	F	F	F	F
TN	F	F	F	F	F	F
H_1	F/H	F/H	–	–	F/H	F/H
H_2	F/H	F/H	–	–	–	F/H

† Each letter of the table is referred to a specification as in model 5.3.4.

Table 5.18: Local Calls: URM$_6$ OLS parameter estimates.

Observations	4391
$\bar{R}^2$	.930
Std.err.est.	.278
Robust CMSPE	.286
AIC	−11207
BIC	−10939

	Variable	coefficient	t-statistic
	$Constant$	2.644	8.783
A priori transforms	$\ln(EMH)$	2.427	61.851
	$\ln(BUS/EMH)$	1.095	76.492
	$\ln(HUN/EMH)$	.341	18.568
	$\ln(PBX/EMH)$	1.201	77.146
	$\ln(CTX/EMH)$	.767	55.685
Specific constants	$STAGEI$	.028	2.031
	$BSOUTH$	.163	14.159
	AL	.307	2.694
	GA	.099	7.064
	KY	.180	9.861
	LA	.357	18.921
	MS	.389	21.035
	SC	.317	16.586
	TN	.265	13.747
Interaction Terms	$\ln(BUS)\ \ln(HUN)$	−.077	−1.642
	$\ln(BUS)\ \ln(PBX)$	−.145	−5.040
	$\ln(HUN)\ \ln(PBX)$	−.101	−15.432
	$\ln(HUN)\ \ln(CTX)$	−.055	−13.331
	$\ln(EMH)/\ln(SAL)$	.119	4.100
	$[\ln(SAL)\ \ln(SQFT)]^{-1}$	4.511	9.162
Specific slopes of Cluster 1	$H_1\ \ln(EMH)^2$	.058	19.144
	$H_1\ \ln(BUS)\ \ln(EMH)$	−.172	−22.787
	$H_1\ \ln(HUN)\ \ln(EMH)$	−.027	−4.916
	$H_1\ \ln(PBX)\ \ln(EMH)$	−.139	−25.710
	$H_1\ \ln(CTX)\ \ln(EMH)$	−.064	−16.913
	$H_1\ \ln(HUN)\ \ln(PBX)$	.082	8.469
	$H_1\ [\ln(SAL)\ \ln(SQFT)]^{-1}$	−3.185	−4.723
	$H_1\ \ln(BUS)/\ln(POP)$	2.195	6.373
	$H_1\ \ln(HUN)/\ln(POP)$	.450	1.986
	$H_1\ \ln(EMH)/\ln(POP)$	−1.381	−7.654
	$H_1\ \ln(EMT)/\ln(POP)$	1.630	14.357
Specific slopes of Cluster 2	$H_2\ \ln(BUS)^2$	.022	4.174
	$H_2\ \ln(SAL)^{-2}$	−1.151	−8.161
	$H_2\ \ln(SQFT)^{-2}$	−9.551	−8.977
	$H_2\ \ln(CTX)\ \ln(EMH)$	.015	5.185
	$H_2\ \ln(EMH)\ \ln(POP)$	−.003	−4.505
	$H_2\ \ln(BUS)/\ln(SAL)$	.161	2.756
	$H_2\ \ln(HUN)/\ln(SQFT)$	.506	3.472
	$H_2\ \ln(EMH)/\ln(POP)$	−.446	−6.724

Table 5.19: WLS estimation of URM$_6$ with heteroskedasticity correction†.

Observations	4391
$\bar{R}^2$	.973
Std.err.est.	.278

	Variable	coefficient	t-statistic
	$Constant$	2.534	166.664
A priori transforms	$\ln(EMH)$	2.592	11.937
	$\ln(BUS/EMH)$	1.150	155.441
	$\ln(HUN/EMH)$	.370	29.58
	$\ln(PBX/EMH)$	1.239	119.72
	$\ln(CTX/EMH)$	.804	6.735
Specific constants	$STAGEI$	.178	26.892
	AL	.380	38.466
	GA	.119	13.372
	KY	.233	21.59
	LA	.309	23.085
	MS	.405	35.803
	SC	.377	46.607
	TN	.307	33.193
Interaction Terms	$\ln(BUS)\ \ln(HUN)$	−.100	−19.099
	$\ln(BUS)\ \ln(PBX)$	−.200	−6.318
	$\ln(HUN)\ \ln(PBX)$	−.111	−21.988
	$\ln(HUN)\ \ln(CTX)$	−.054	−15.52
	$\ln(EMH)/\ln(SAL)$	.057	3.346
	$[\ln(SAL)\ \ln(SQFT)]^{-1}$	5.336	2.662
Specific slopes of Cluster 1	$H_1\ \ln(EMH)^2$	.058	12.556
	$H_1\ \ln(CTX)\ \ln(EMH)$	−.076	−15.211
	$H_1\ \ln(HUN)\ \ln(EMH)$	−.027	−5.391
	$H_1\ \ln(HUN)\ \ln(PBX)$	.100	7.721
	$H_1\ \ln(PBX)\ \ln(EMH)$	−.151	−19.665
	$H_1\ [\ln(SAL)\ \ln(SQFT)]^{-1}$	−.149	−13.382
	$H_1\ \ln(EMH)/\ln(POP)$	−1.112	−3.587
	$H_1\ \ln(EMT)/\ln(POP)$	1.863	9.001
	$H_1\ \ln(BUS)/\ln(POP)$	1.625	4.180
	$H_1\ \ln(SAL)/\ln(POP)$	−.531	−3.153
Specific slopes of Cluster 2	$H_2\ \ln(SAL)^{-2}$	−1.320	−16.472
	$H_2\ \ln(SQFT)^{-2}$	−9.782	−17.047
	$H_2\ \ln(EMH)\ \ln(CTX)$	.008	2.835
	$H_2\ \ln(EMH)\ \ln(POP)$	−.002	−4.989
	$H_2\ \ln(EMH)/\ln(POP)$	−.389	−1.103
	$H_2\ \ln(BUS)/\ln(SAL)$	.332	9.438
	$H_2\ \ln(HUN)/\ln(SQFT)$	.369	4.034

† Several transformations have been dropped since they were no longer significant after WLS estimations.

Table 5.20: Selected Elasticities from local calls URM$_6$ weighted model (Table 5.19)

$$\begin{aligned}
\frac{\partial \ln(LOCAL)}{\partial \ln(BUS)} &= 1.150 - .149\,H_1\ \ln(EMH) - .100\ \ln(HUN) - .200\ \ln(PBX) + \\
&+\ \frac{1.626\,H_1}{\ln(POP)} + \frac{.332\,H_2}{\ln(SAL)}
\end{aligned}$$

$$\begin{aligned}
\frac{\partial \ln(LOCAL)}{\partial \ln(HUN)} &= .029 - .149\,H_1\ \ln(BUS) - .076\,H_1\ \ln(CTX) + .008\,H_2\ \ln(CTX) + \\
&-\ .027\,H_1\ \ln(HUN) - .151\,H_1\ \ln(PBX) + .116\,H_1\ \ln(EMH) + \\
&-\ .002\,H_2\ \ln(POP) - \frac{1.112H_1}{\ln(POP)} - \frac{.389\,H_2}{\ln(POP)} + \frac{.057}{\ln(SAL)}
\end{aligned}$$

$$\begin{aligned}
\frac{\partial \ln(LOCAL)}{\partial \ln(PBX)} &= 1.239 - .200\ \ln(BUS) - .151\,H_1\ \ln(EMH) - .111\ \ln(HUN) + \\
&+\ .100\,H_1\ \ln(HUN)
\end{aligned}$$

$$\frac{\partial \ln(LOCAL)}{\partial \ln(CTX)} = .804 - .076\,H_1\ \ln(EMH) + .008\,H_2\ \ln(EMH) - .054\ \ln(HUN)$$

$$\begin{aligned}
\frac{\partial \ln(LOCAL)}{\partial \ln(SAL)} &= -\frac{.531\,H_1}{\ln(POP)} + \frac{2.637\,H_2}{\ln(SAL)^3} - \frac{.332\,H_2\ \ln(BUS)}{\ln(SAL)^2} \\
&-\ \frac{.057\ \ln(EMH)}{\ln(SAL)^2} - \frac{5.336}{\ln(SAL)^2\ \ln(SQFT)}
\end{aligned}$$

$$\frac{\partial \ln(LOCAL)}{\partial \ln(EMT)} = \frac{1.863\,H_1}{\ln(POP)}$$

$$\begin{aligned}
\frac{\partial \ln(LOCAL)}{\partial \ln(EMH)} &= .029 - .149\,H_1\ \ln(BUS) - .076\,H_1\ \ln(CTX) + .008\,H_2\ \ln(CTX) + \\
&+\ .116\,H_1\ \ln(EMH) - .027\,H_1\ \ln(HUN) - .151\,H_1\ \ln(PBX) - \frac{1.112H_1}{\ln(POP)} + \\
&-\ \frac{.389\,H_2}{\ln(POP)} - .002\,H_2\ \ln(POP) + \frac{.057}{\ln(SAL)}
\end{aligned}$$

$$\begin{aligned}
\frac{\partial \ln(LOCAL)}{\partial \ln(POP)} &= -.002\,H_2\ \ln(EMH) - \frac{1.625\,H_1\ \ln(BUS)}{\ln(POP)^2} + \frac{1.112\,H_1\ \ln(EMH)}{\ln(POP)^2} + \\
&\quad \frac{.389\,H_2\ \ln(EMH)}{\ln(POP)^2} - \frac{1.863\,H_1\ \ln(EMT)}{\ln(POP)^2} + \frac{.531\,H_1\ \ln(SAL)}{\ln(POP)^2}
\end{aligned}$$

Table 5.21: Selected Elasticities from inter-LATA URM weighted model (eq.7).

$$\begin{aligned}
\frac{\partial \ln(INTER)}{\partial \ln(BUS)} &= \ln(HUN) \\[2em]
\frac{\partial \ln(INTER)}{\partial \ln(HUN)} &= .234 - .397 \ln(BUS) + 1.251 \ln(HUN) \\[2em]
\frac{\partial \ln(INTER)}{\partial \ln(EMT)} &= .840\, H_1 \\[2em]
\frac{\partial \ln(INTER)}{\partial \ln(EMH)} &= .285 - .840\, H_1 \\[2em]
\frac{\partial \ln(INTER)}{\partial \ln(POP)} &= .002 \ \ln(POP) + .787\, H_1 \ \ln(SAL) \ \ln(POP)^{-2} \\[2em]
\frac{\partial \ln(INTER)}{\partial \ln(SAL)} &= \frac{.787\, H_1}{\ln(POP)}
\end{aligned}$$

Chapter 6

RETINET

6.1 Introduction

This chapter reports the results of an on-going project for the development of a new automatic approximation and modeling tool called RETINET. In previous chapters we have seen both approximation and automatic selection procedures that allow to build predictive functions inspired by the principle of parsimony. Among other subset selection methods, RETINA has the unique feature to automatically include level one transformations of type $X_{it}^{\alpha_1} X_{jt}^{\alpha_2}$ with $\alpha_1, \alpha_2 = -1, 0, 1$ that may be able to capture the presence of simple non-linear structures in the data. However, even if this approach may be useful in many contexts, it may well be the case that different non-linear structures are not approximated by level one transformations. As a possible remedy to this, one could extend the class of transformations to include a wider choice of functional forms, such as logarithms, lags or expansions, such as polynomials, splines, or trigonometric series. Another convenient approach is to use artificial neural network functions which in virtue of their approximation properties (Cybenko 1989, Hornik et al. 1989, Barron 1993) can achieve an approximation rate of order $O(q^{-1})$ by using a number of parameters $O(qT)$ that grow linearly in q where q is the number of nodes in the hidden layer and T is the sample size. This is in contrast with traditional polynomial, spline and trigonometric expansions which require exponentially $O(q^T)$ terms to achieve the same approximation rate. Hence ANNs are at least asymptotically more parsimonious than these series expansions in approximating unknown functions. In spite of these results, a less attractive feature of ANNs is relative to the computational effort for parameter estimation. Moreover in empirical applications, it is not always clear how the specification (network architecture) should be defined. Finally, ANNs do not provide reverse engineering capabilities, which to some extent level one transformations still retain. Regardless of these practical issues, ANN become an attractive alternative econometric tool for nonparametric applications and an interesting way to extend RETINA's approximation capabilities.

In this context, it seems natural the development of a more advanced automatic modeling and prediction tool, which incorporates the advantages of both RETINA and ANN modeling, while keeping some of the principles we have been inspired so far: *low computational effort, reverse engineering capabilities* and *parsimony*. In this chapter we propose a new algorithm called REelevant Transformations of the Inputs NETwork (RETINET). RETINET integrates the approximation capabilities and specification search of RETINA and ANN in order to achieve more flexibility in approximating an unknown function. Our approach takes into account different aspects related to the empirical construction of ANNs architectures and their estimation. The typical problem associated with neural network estimation is that the functional form embodied in these models may easily become "too flexible" which may easily result in overfit (Looney 1997), a situation in which the model does not generalize well out of sample. Model overfit in ANN's is fundamentally caused by an over-complexity in the model specification that is directly analogous to the overfitting problems that one encounters with linear models. As we already discussed in chapter 2, the solution consists in simplifying the specification by dropping variables and/or using some form of regularization like ridge regression.

Recently White (2006) described a new family of methods called QuickNet which in part underlies RETINET. These methods aim directly at balancing the competing dangers of underfit and overfit to identify the level of model complexity that guarantees the best out-of-sample prediction performance without ad-hoc modifications to the fitting algorithms themselves. While QuickNet provides a general method, it leaves open several choices in specific implementation of the various modeling steps. RETINET uses QuickNet as the basis, draws from RETINA for its specific implementation schemes, and includes additional customization features. RETINET creates a predictive model architecture that is linear in the parameters but yet non-linear (level one) or highly non-linear in the inputs (ANN-likewise). This allows us to avoid the minimization of complicated non-linear estimation procedures for the parameters of such transformations, preserving the linearity in the parameters which is also a characteristic of RETINA.

The chapter is organized as follows: the next section will briefly overview some of the work done in the econometric literature related to Neural Networks. Section 6.3 specifies the econometric model and gives a statistical interpretation of it. Section 6.4 discusses specific aspects of building flexible functional forms. A model specification strategy, susceptible of algorithmic treatment implemented in RETINET, is presented in section 6.5. Two applications using simulated data sets are presented in the last two sections. Final remarks and conclusions follow.

6.2 Some applications of ANN in econometrics: a brief overview

ANNs have recently gained popularity as an emerging and challenging computational methodology, and they offer a new avenue to explore the dynamics of a variety of applications in economics and finance. Single layer feed-forward networks by far have been the most popular in time series econometrics, and therefore, we restrict attention to this particular form of ANN. Models using single layer networks for forecasting exchange rates have been investigated in a number of studies by Kuan & Liu (1995), Brooks (1997), Zhang, Patuwo & Hu (1998), Zhang, Patuwo & Hu (2001), White & Racine (2001) and Kodogiannis & Lolis (2002) to mention a few. In several applications, Tang & Fishwich (1993), Jhee & Lee (1993) and Hill, O'Connor & Remus (1996) have shown that ANNs perform better than linear ARIMA models in terms of out-of-sample predictive ability, specifically, for irregular series and for multiple-period-ahead forecasting. Forecasting macroeconomic variables such as inflation using ANN have been considered by Swanson & White (1997), Nakamura (2005) and Binner, Elger, Nilsson & Tepper (2006). Another application often encountered in the literature is relative to forecasting of stock returns. White (1988) assesses the *efficient markets* hypothesis[1] using single layer feed-forward ANN. Recently, hybrid ANNs integrating predictions from GARCH models as inputs, have been proposed to overcome the difficulty of ANN in predicting volatility of financial time series (Roh 2007). This is by no means an exhaustive list of the applications of ANN in economics and finance but may give the reader an idea on how much interest there is nowadays in exploring new research directions using ANNs.

[1] In finance, the efficient market hypothesis asserts that financial markets are informationally efficient, or that prices on traded assets, e.g., stocks, bonds, or property, already reflect all known information and therefore are unbiased in the sense that they reflect the collective beliefs of all investors about future prospects.

6.3 Specification of flexible functional forms

If we put all the parameterization possibilities already discussed in chapter 3 in a single specification we obtain a generic functional approximation of the form:

$$\begin{aligned} Y_t &= \alpha'\tilde{X}_t && \text{(a) Pure Linear Component} \\ &+ \textstyle\sum_g^l \zeta_g(X_t)'\delta_g && \text{(b) Level One Transformations} \\ &+ \textstyle\sum_h^q \psi(X_t,\Gamma_h)'\beta_h && \text{(c) ANN Transformations} \\ &+ \varepsilon_t \end{aligned} \tag{6.3.1}$$

The notation makes clear the distinction between the different components of the proposed specification. In particular we have: (a) the vector of inputs $\tilde{X}_t \in \mathbb{R}^{k+1}$ and their associated coefficients vector $\alpha \in \mathbb{R}^{k+1}$. Here $\tilde{X}_t$ defined as $\tilde{X}_t = [1, X_t']'$ where $X_t \in \mathbb{R}^k$ may include lagged values of the response Y_t as well as exogenous predictors X_t and their lagged values. The second term (b) of eq. 6.3.1 includes level one transformations of the inputs, of the type discussed in section 3.1.1 given by: $\zeta(X_t) = X_{it}^{a_1}X_{jt}^{a_2}$ with $i, j = 1, 2, \ldots, k$ and $a_1, a_2 = -1, 1$ with an associated coefficients vector $\delta \in \mathbb{R}^{k+2k^2}$. Finally we have (c) the *ANN transformations* of the inputs X_t given by $\psi(X_t, \Gamma_h)$, often called the activation function. The function ψ is chosen among a *library* Ψ of Generically Comprehensive Revealing (GCR) functions (see section 3.1.4). In particular we consider a library of the following three GCR activation functions:

The Logistic Function:

$$\psi_{lgt}(X_t, \Gamma) = \frac{1}{1 + \exp(-\gamma_1' X_t + \gamma_2)}$$

where $\Gamma = \{\gamma_1, \gamma_2\}$ with direction vector $\gamma_1 \in \mathbb{R}^k$ and centering scalar $\gamma_2 \in \mathbb{R}$.

The Radial Basis Function:

$$\psi_{rbf}(X_t, \Gamma) = \exp[-.5(X_t - \gamma_1')'\gamma_2^{-1}(X_t - \gamma_1')]$$

where $\Gamma = \{\gamma_1, \gamma_2\}$. Here $\gamma_1 \in \mathbb{R}^k$ represents a centering vector and γ_2 is a $k \times k$ symmetric positive semi-definite matrix which scales departures of X_t from γ_1.

The Ridgelet Function:

$$\psi_{rdg}(X_t, \Gamma) = \frac{1}{\sqrt{\gamma_1}} f\left(\frac{X_t'\gamma_2 - \gamma_0}{\gamma_1}\right)$$

with parameters set $\Gamma = \{\gamma_0, \gamma_1, \gamma_2\}$ where $\gamma_0, \gamma_1 \in \mathbb{R}$, and γ_2 is a direction vector on the unit sphere in $\mathbb{R}^k$. The function f has to be admissible (see section 3.1.3). In this particular case we use the $(k/2)^{\text{th}}$ derivative of the standard normal which has been shown to satisfy the admissibility condition (White 2006).

The coefficients β_h associated to each ψ ANN transform configure a vector of parameters $\beta \in \mathbb{R}^q$. Furthermore ε_t is a sequence of independently distributed random variables with zero mean and variance σ. Notice that since we allow for simultaneous presence of the three activation functions in our approximation, 6.3.1 should be written as follows:

$$Y_t = \alpha'\tilde{X}_t + \sum_{g}^{l} \zeta_g(X_t)'\delta_g + \sum_{h=1}^{q_{lgt}} \psi_{lgt}(X_t, \Gamma_h)'\beta_h + \sum_{h=1}^{q_{rbf}} \psi_{rbf}(X_t, \Gamma_h)'\beta_h + \sum_{h=1}^{q_{rdg}} \psi_{rdg}(X_t, \Gamma_h)'\beta_h + \varepsilon_t$$

where $q_{lgt} + q_{rbf} + q_{rdg} = q$ of eq. 6.3.1. Nonetheless in order to keep notation simple we prefer the representation of eq. 6.3.1 with the implicit assumption that ψ refers to a library of activation functions rather than just a single definition.

Notice that, in order to achieve more flexibility, a possible variant of eq. 6.3.1 is:

$$Y_t = \alpha'\tilde{X}_t + \sum_{g}^{l} \zeta_g(X_t)'\delta_g + \sum_{h}^{q} \psi[\zeta(X_t), \Gamma_h]'\beta_h + \varepsilon_t \qquad (6.3.2)$$

which allows for simple transforms $\zeta(X_t)$ to act as inputs of the highly non-linear transformation ψ. The topology of 6.3.1 and 6.3.2 is represented respectively in figures 6.1, and 6.2. In 6.1 a single layer feed-forward network is represented having as special feature the simultaneous presence of highly non-linear transforms (the ψ's) and the level one transforms as well (the ζ's). Direct input-to-output connections, associated with the α coefficients account for linear relationships, allowing the hidden units to concentrate on nonlinearities. Direct input-to-output connections are also called *jump connections* in virtue of the fact that in this case the inputs X_t influence directly the response Y_t. This network has a single layer architecture because input nodes X_t are directed towards the corresponding hidden units, and the resulting transformations ψ are a weighted sum of these. Equation 6.3.2 is represented graphically in fig. 6.2, where the ζ level one transforms are used as derived

Figure 6.1: Single Layer architecture with direct Jump Connections

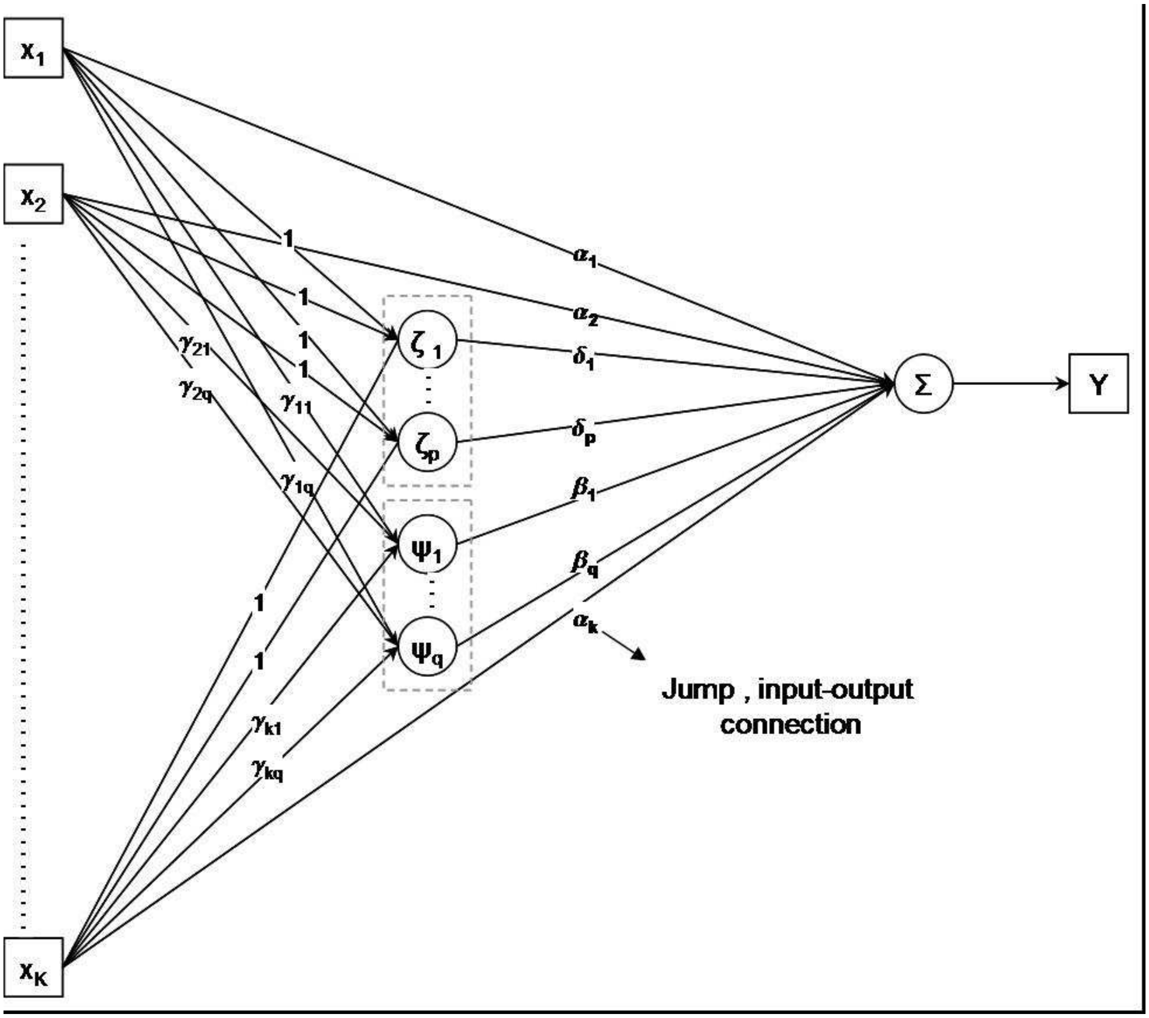
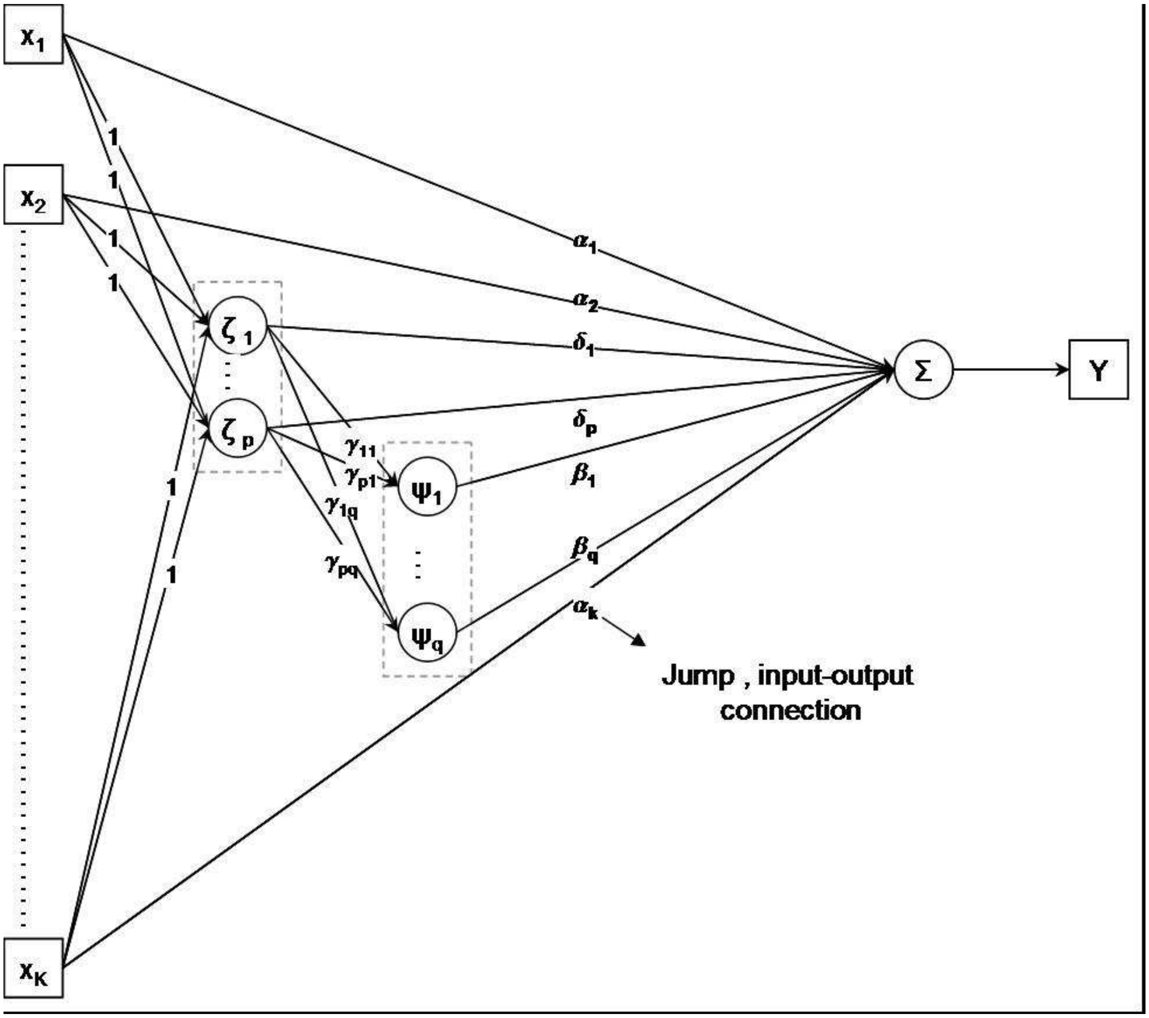


Figure 6.2: Double layer architecture with direct Jump Connections



inputs for the hidden units of type ψ. In this latter case we obtain a double layer network, with the first layer of ζ transforms and the second layer of ψ's transforms. An important feature of the specifications proposed so far is that they nest the simple linear regression model as a special case. Moreover eq. 6.3.1 and 6.3.2 also nest RETINA specifications (see chap.4) and the usual definition of single layer feed-forward networks which is given by:

$$Y_t = \alpha' \tilde{X}_t + \sum_{h}^{q} \psi_{lgt}(X_t, \Gamma_h)' \beta_h + \varepsilon_t \tag{6.3.3}$$

where ψ_{lgt} is the logistic activation function. The main difference with respect to our formulation is that we consider a library of different approximation functions (including simple level one transforms besides more involved ANN transforms) as discussed in chapter 3. This point will be discussed further in section 6.4.3, but for now, it allows to refer to eq.6.3.1 and eq.6.3.2 as network architectures. In virtue of the following we will use the term specification and network architecture interchangeably.

Also notice that certain special cases of eq. 6.3.1 using a logistic activation function for ψ are of interest. In particular Franses & van Dijk (2000) show that when $X_t = Y_{t-r}$ in ψ it may accommodate Multiple logistic smooth transition auto-regressive models (MLSTAR) with $q+1$ regimes in which only the intercept changes according to the regime. Other special cases is the Self-Exciting Threshold autoregressive Model (SETAR) (Terasvirta 1994) also discussed in Franses & van Dijk (2000).

6.4 Building flexible functional forms

In the previous section we defined a very general flexible functional form that considers libraries of linear and non-linear transformations of the input data. The resulting equation 6.3.1 is *highly non-linear* in the inputs and parameters Γ. In order to build such flexible functional forms, we proceed in a *specific to general* direction, from smaller to larger models, letting the data to determine the final model architecture. This is like a “growing” process as an analogy with biological systems. This analogy is useful because it establishes a correspondence between the size and the complexity of the resulting specification. In practice the idea is the following: we always start the specification with linear terms. When then let grow the size of the specification adding more complex non-linear terms chosen from a library of non-linear transformations. This procedure corresponds to increase the degree of non-linearity of the specification as the number of terms in the model grows. As an important advantage of this approach it retains, as much as possible, *reverse engineering capabilities* in our final model.

In an attempt to automatize this modeling procedure we also need to consider the difficulties associated with the *estimation* of the model, as its complexity, in the sense above described, increases. Adding non-linear terms also increases computational costs due to non-linear estimation. As already discussed in chapter 3, taking advantage of the properties of GCR activation functions, it is possible to circumvent this problem by generating a collection Ψ of ψ transformations from which we choose in a way described later some particular ψ^*. In addition, other aspects to consider are relative to the choice of the inputs, the type of activation function, and the number of hidden nodes. All these elements may heavily influence the forecasting performance of the network. In the following sections we shall discuss more in detail these aspects and some common solutions proposed in the literature, and in the approach considered here.

6.4.1 Choice of the inputs

The choice of the inputs is a very important step in building any network architecture. In many empirical works using ANNs, the choice of the inputs is driven by systematic experimentation. In a dynamic context it is natural to include lagged values of the response as inputs in order to capture memory effects (Auto-Regressive Neural Networks, AR-NN). In the univariate case the input vector is:

$$X_t = \{Y_{t-1}, Y_{t-2}, \ldots, Y_{t-r}\}$$

where r is the index of the maximum lag order considered. Another class of networks are the Auto-Regressive Neural Networks with exogenous regressors (ARX-NN) where:

$$X_t = \{Y_{t-1}, Y_{t-2}, \ldots, Y_{t-r}, Z_t, Z_{t-1}, Z_{t-2}, \ldots, Z_{t-s}\}$$

where Z_t is a vector of exogenous predictors[2]. One difficulty in AR-NN and ARX-NN models is represented by the choice of the correct number of lags which is

[2] Auto-Regressive Neural Networks with exogenous predictors are studied by Chen, Racine & Swanson (2001). They establish root mean square convergence rates for ANN estimates of the conditional mean function with stationary $\beta-$mixing data. They consider three classes of ANN: one smooth sigmoid activation functions, the second uses radial basis, and the third uses ridgelets, and provide evidence that all networks outperform linear models based on different forecasting measures. Their results provide the theoretical justification for using neural networks to fit multivariate economic time series.

unknown a priori[3]. A common way to proceed is to choose the order r of the lags of the dependent variable by estimating a sequence of auto-regressive specifications up to order r. Then the specification that delivers the lowest AIC or BIC statistic is chosen and the corresponding lagged values of the response are considered as inputs of the network. This strategy may become problematic for two reasons: first because one may consider exogenous predictors in addition to lagged values of the response as candidate inputs. In this circumstance one needs also to consider lagged values of each exogenous predictor and different combinations of lagged response and exogenous predictors should be considered which increases the difficulty of the selection task. Second, when dealing with unordered or cross-section data, lack of an a priori ordering is always a possibility. But even if ordering of the inputs is possible, a parsimonious alternative, taking fewer unordered terms which avoids an unnecessarily set of inputs, should be considered.

Among other existing non-parametric approaches (Tschernig & Yang 2000, Yao & Tong 1994) which are computationally very demanding, a simpler procedure for input selection has been suggested by Medeiros, Teräsvirta & Rech (2006). They propose to select the inputs by linearizing the ANN model specification by using a polynomial series expansion. After estimating the coefficients associated with each term of the expansion by OLS, one starts an iterative procedure by dropping one term at time and re-estimating the expansion without the removed term. Subsequently, one repeats the procedure by dropping two terms at time, then three terms and so forth until the polynomial is a function of a single regressor and, finally, just a constant. For each estimation one tracks a model selection statistics and finally chooses the inputs included in the specification with the lowest statistic. The procedure amounts to estimating $\sum_{i=1}^{q} \frac{q!}{i!(n-i)!}$ regressions by OLS which is still a formidable task especially when the number is potentially large. In addition even if this strategy has a lower computational cost with respect to non-parametric methods, it still suffers the problem that the network architecture has to be established in advance, which limits the practical implementation of the method. Since a definitive solution doesn't exist, and the solution is inevitably heuristic in its nature, automatic subset selection may prove to be a valid alternative. In particular we consider

[3] As Kuan & Liu (1995) observes, the inclusion of the lagged response in the input set may not be sufficient to characterize the behavior of the output. In order to mitigate this deficiency networks with "memory" have been proposed in the literature. The Elman (1990) recurrent network is such an example. Networks with recurrent architecture have a richer dynamic structure and may better approximate non-linear dynamics if it present in the data. In particular the Elman network mimics an MA process in time series analysis. Nonetheless, in this first implementation of RETINET we shall skip this discussion for now, and just focus on AR-NN type networks.

the RETINA algorithm very useful for this purpose and motivate this choice by the fact that the procedure already embeds simple transformations of the inputs which are akin to polynomial series expansions (see section 4.1.1). RETINA performs automatically the selection task thus providing a great saving in terms of time while keeping the computational effort low, compared with non-parametric methods as well as compared with the procedure proposed by Medeiros et al. (2006).

6.4.2 Choice of the number of nodes

The choice of the number of nodes in a neural network resembles the decision that one has to make when deciding how many terms to retain when approximating by using series expansions. This problem is similar to subset selection for linear models. Again, a possibility is to use systematic experimentation together with selection procedures outlined in chapter 2. One may estimate many networks with an increasing number of nodes and then choose the one that has better out-of-sample predictive performance or better model selection criterion. Alternatively one estimates a large ANN model and subsequently reduces the size of the network by pruning applying an appropriate technique such as cross-validation. See Fine (1999). As an alternative Medeiros et al. (2006) propose an LM-testing procedure to deal with this problem when Maximum Likelihood (ML) estimation is used. Nonetheless, as already pointed out by different authors, ML is prone to computational difficulties as many other non-linear estimation methods. However, when the hidden units are selected from a collection of ANN transformations, a possibility is to use automatic subset selection methods developed for linear models to accomplish this task. Consider the following single layer neural network:

$$\hat{Y}_t = \hat{\alpha}' X_t + \sum_{h}^{q} \hat{\delta}_h \psi(X_t, \hat{\Gamma}_h)$$

Now we rewrite it as:

$$\hat{Y}_t = \hat{c} + \sum_{h}^{q} \hat{\delta}_h \hat{W}_{ht}$$

where $W_t = \psi(X_t, \hat{\Gamma}_h)$ are viewed as derived predictors and the linear terms are collapsed into a constant term $\hat{c} = \hat{\alpha}' X_t$. Re-parameterizing the ANN specification in this way we get back to a linear model, where automatic subset selection for linear models may be used in order to select the number of hidden units. Once the flexible functional form has been obtained we are motivated use RETINA as a subset selection method to obtain a more parsimonious specification. This provides an effective heuristic method to choose the final number of nodes of the network.

6.4.3 Choice of the Activation function

The activation function ψ is an important part of the network architecture. As Fletcher (1996) points out, selecting the correct one has the same effect on training as selecting the correct topology; networks that use the correct activation functions are smaller and simpler than those that do not. As already discussed in chapter 3 the appropriate choice of a class of approximant functionals depends on the smoothness properties of the underlying data generation process. From the theoretical point of view there aren't specific theoretical reasons for which one should employ just one type of activation function in the network specification. From a practical point of view, the use of libraries from which to choose an appropriate "mix" of functions may result beneficial simply because of the different approximation properties of each of them. To our knowledge, currently, no ANN software implementations and/or ANN applications provide libraries of different activation functions that can be used simultaneously in a network specification. This is probably motivated by the fact that heuristics to choose the appropriate "mix" is problematic in many ways. First, because it is not clear how to define and justify an a priori network specification given these circumstances, and most importantly because estimation difficulties may easily arise.

The RETINET procedure considers this as a possibility. As pointed out in section 6.3, RETINET implements and allows to combine three types of activation functions: the logistic function, radial basis function and ridgelets. We are motivated to use logistic function since it is computationally inexpensive and because, historically, it has been by far the most used in the literature. Radial basis functions are useful to accommodate mixtures of multivariate normal functions of the input space. Ridgelets are powerful at detecting singularities and sharp profiles of the data, although it must be noticed that its computations is more involved than other transforms since it depends on the number of inputs considered (see section 3.1.3). Fortunately, when a normal density kernel is employed, it is possible to compute ridgelets by using a well known recursive relationship valid for Hermite polynomials which allows to easily compute any derivative of the normal distribution. This method, implemented in the RETINET algorithm is described in section 6.9.1 of the appendix.

6.4.4 Hidden parameters generation

As anticipated, to build our approximating function, we generate a library Ψ of candidate transformations of the inputs ψ, by, randomly generating the weight parameters Γ. Doing this we take into consideration the fact that the resulting transforms ψ should be not too collinear with the inputs X_t or $\zeta(X_t)$. By the same token, those inputs that are approximatively constant or have reduced range of variation should also be avoided. In order to ensure these desirable properties we proceed as White (2006) suggests.

First we generate the hidden units by scaling adequately and selecting randomly the elements of the weights set Γ such that the magnitudes of the parameters (usually position and scale) are comparable and independent each other. As an example take the logistic squashing function $\psi(\gamma_0 + \gamma_1 X_t)$ with a single predictor X_t having mean zero. Here $\Gamma = \gamma_0, \gamma_1 \in \mathbb{R}$. If γ_0 (the scale parameter) is chosen too large in absolute value compared to $\gamma_1 X_t$ it will happen that $\psi(\gamma_0 + \gamma_1 X_t)$ behaves approximately as $\psi(\gamma_0)$ that is it will be roughly constant. If γ_1 (the scale parameter) is chosen small relative to the standard deviation of X_t then it will happen that $\psi(\gamma_0 + \gamma_1 X_t)$ will vary proportionally to $\gamma_0 + \gamma_1 X_t$ and therefore will be collinear with respect to $\psi(\gamma_0 + \gamma_1 X_t)$. To avoid such problems let $\psi(\gamma_0 + \gamma_1 X_t)$ behave as a nonlinear function of X_t. It is thus recommendable to scale γ_0, γ_1 adequately and second, to choose them independently. Independence is be warranted by choosing γ_0, γ_1 randomly. This also ensures that correlation among predictors is reduced which is also enforced by further standardization of the generated hidden units. Standardization is beneficial also in that it reduces potential numerical problems that may arise during matrix inversions if the magnitudes of the variances of the predictors vary greatly. These considerations are general and hold also in the multivariate case, as well as for different activation functions, although a different tune up has been necessary for Ridgelets and Radial Basis Functions.

6.5 RETINET's modeling strategy

At this point we are ready to combine the above ingredients into a coherent modeling strategy which, in its automated version, we call RETINET.

From a high level RETINET starts selecting the input units adopting the RETINA automated strategy discussed in chapter 4, and builds the linear component which may include some simple transformation of inputs that prove to be useful in order to retain some reverse engineering properties in the final network specification.

Second, it builds an ANN component on top of the linear part trying to explain additional the residual variance which the linear part is not able to capture. This is done by adding in a stepwise manner non-linear transformations of the inputs ψ which are taken from a library of a possibly large randomly generated collection of non-linear transforms.

Finally, the resulting specification is pruned in order to achieve a more parsimonious final representation. This last stage is obtained by means of RETINA which at this point just acts as a subset selection tool.
Next we will describe in more detail the whole procedure:

Step 1: Selection of inputs to build the linear component. The first step is aimed to select the inputs of the network and an approximating specification using simple transformations $\zeta(X_t)$. We accomplish this by selecting a suitable subset of X_t and $\zeta(X_t)$ transforms. We do this using the RETINA strategy described in the previous chapter. At the end of this stage one could stop if no more accuracy in predicting the response is necessary, estimating by OLS a reduced version of eq. 6.3.1 which excludes the ψ's ANN-like components:

$$\hat{Y}_t = \hat{\alpha}'\tilde{X}_t + \sum_{g}^{l} \zeta_g(X_t)'\hat{\delta}_g \tag{6.5.1}$$

then compute the residual term as:

$$\hat{\varepsilon}_t = Y_t - \hat{Y}_t \tag{6.5.2}$$

Step 2: Generate the Ψ library. Here we generate a potentially large number, say v of ψ_j, $j = 1, \ldots, v$ ANN transforms of the inputs X_t, where v has been chosen in advance. This collection of transforms configures a matrix Ψ stored in the computer memory.

Step 3: Select a $\psi^* \in \Psi$. From the collection of the Ψ randomly generated transforms. We choose one $\psi^* \equiv \psi(X_t, \Gamma^*)$ such that:

$$\psi* = \text{argmax}_{\psi \in \Psi} |\rho(\hat{\varepsilon}_t, \Psi)|$$

where ρ is the univariate correlation between $\hat{\varepsilon}_t$ and each $\psi_j \in \Psi$. This is equivalent to search along the space orthogonal to the predictors already included in the specification at step 1.

Step 4: Add the new candidate ψ_j^* and estimate the new specification. Add ψ_j^* to the 6.5.1 specification and estimate α, β, δ by OLS. This delivers:

$$\hat{Y}_t = \hat{\alpha}'\tilde{X}_t + \sum_{g}^{l} \zeta_g(X_t)'\hat{\delta}_g + \psi_j(X_t, \Gamma^*)'\hat{\beta}_1$$

from which we obtain the residuals :

$$\hat{\varepsilon}_t = Y_t - \hat{\alpha}'\tilde{X}_t - \sum_{g}^{l} \zeta_g(X_t)'\hat{\delta}_g - \psi(X_t, \Gamma^*)'\hat{\beta}_1 \tag{6.5.3}$$

Step 5: Iterate steps 2 to 4, Q times. After iterating Q times (where Q has been established in advance) we obtain a specification of the form

$$\hat{Y}_t = \hat{\alpha}'\tilde{X}_t + \sum_{g=1}^{l} \zeta(X_t)'\hat{\delta}_g + \sum_{h=1}^{q_{lgt}} \psi_{lgt}(X_t, \Gamma_h)'\hat{\beta}_{hh} + \sum_{h=1}^{q_{rbf}} \psi_{rbf}(X_t, \Gamma_h)'\hat{\beta}_{hh} + \sum_{h=1}^{q_{rdg}} \psi_{rdg}(X_t, \Gamma_h)'\hat{\beta}_{hh}$$

Step 6: Network reduction and pruning. Use RETINA subset selection search to select the final specification.

Some considerations:

Remark 1. Using RETINA in Step 1 provides $\zeta(X_t)$ transformations as a first order approximation to the unknown underlying DGP, and as such, less ψ terms are usually required for the approximation, thus retaining as much a possible an analytic formulation in the final specification to facilitate reverse engineering.

Remark 2. Steps 2 to 4 are akin to White (2006) Quicknet strategy. However we may optionally also want to consider the following possibilities:

- in step 3 we may allow to choose the first n ψ's most correlated terms with the residual series, instead of selecting just one.
- in step 3 we may want to store the Ψ's generated at each step and cumulate these across iterations, such that a wider set of possible candidate ψ's is available at each iteration.

Remark 3. We could use RETINA to select a set of ψ's at each iteration instead of using the absolute univariate correlation as a criterion. However this increases the computational cost, although it may provide a better specification in a lower number of iterations.

Remark 4. We could use a stopping rule such as the AIC criterion instead of using a final RETINA selection stage as in step 6. Nonetheless the solution provided by RETINA usually offers a more parsimonious solution since it re-shuffles all the terms included in the final specification and simultaneously evaluates the predictive ability of different combinations of them.

Remark 5. White's Quicknet doesn't provide a strategy to select the inputs which is provided in step 1 by RETINA.

The mechanics of the whole procedure is represented in figure 6.3. If the specification containing only the linear transforms of the input variables does as the user expected, or the user wishes to check the model accuracy when the non-linear terms are used, RETINET proceeds to building and then adding the non-linear transformations to the model which may lead to increase in accuracy. Once the non-linear transformations ψ of the input are created, they are merged with the linear transforms selected in the previous steps of the algorithm, and a final subset selection step suggests the final specification.

6.6 An application to simulated Time Series

In this section we propose some simple examples that show the forecasting ability of the specifications suggested by RETINET. First we are interested in evaluating RETINET's suggested specifications versus single layer ANN networks estimated by Maximum Likelihood, (ML-ANN in the following) on the basis of their respective out-of sample forecasting ability. A second, and possibly even more relevant question, is whether the forecasting performance of RETINET's models provide some gain with respect to simpler linear approximations.

In order to clearly distinguish RETINET specifications that include ANN-like transformations of the inputs (eq.6.3.1 (a+b+c)), from those that do not (eq.6.3.1 (a+b)), we shall refer to the former as RNET-ANN and to the latter as RNET-LIN. Since RETINET's suggested models always incorporate a linear component, we expect an out-of-sample forecasting ability which, at least, doesn't perform worse in terms of forecasting ability than less elaborate linear models.

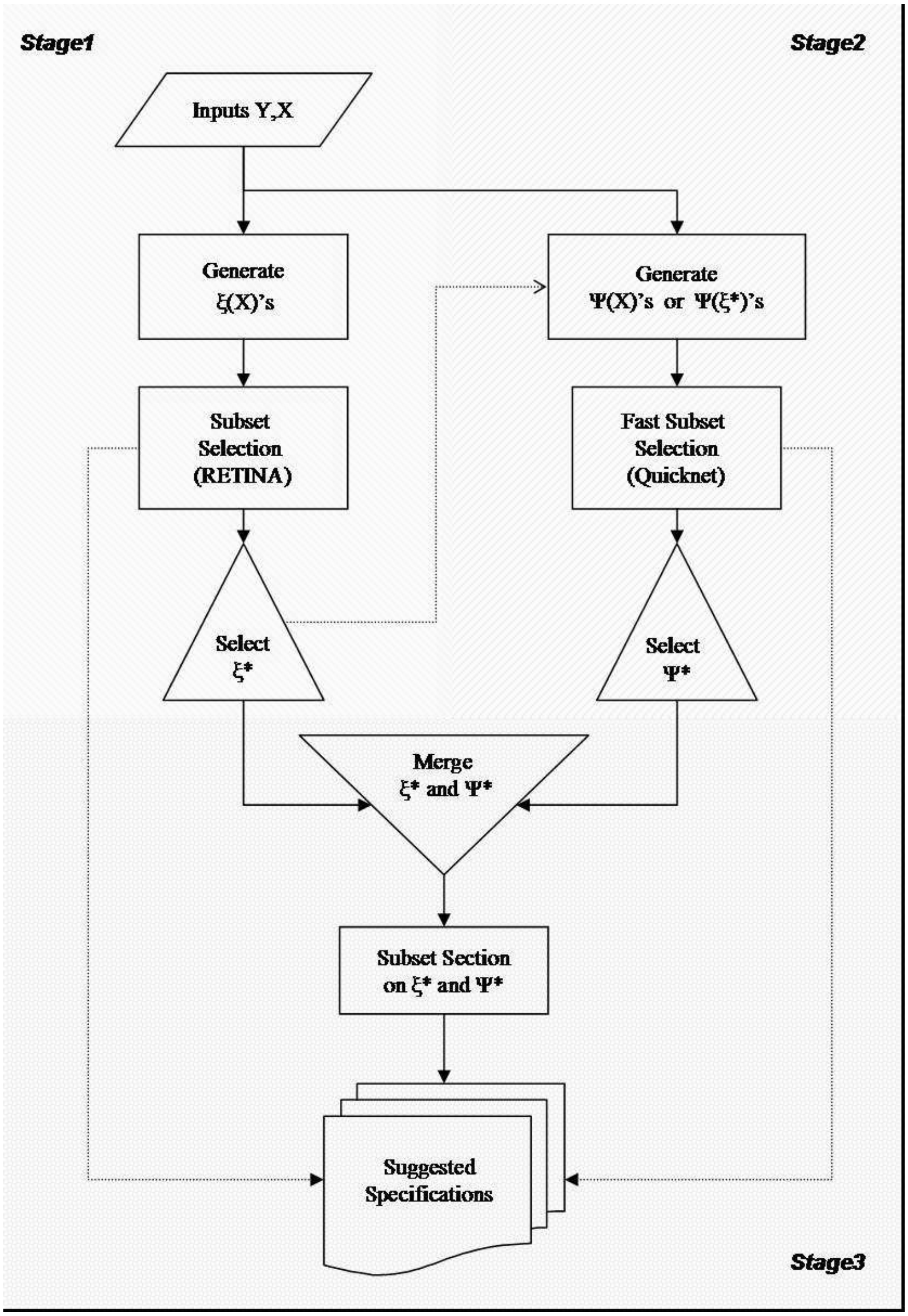

Stage1
Stage2
Inputs Y,X
Generate ξ(X)'s
Generate Ψ(X)'s or Ψ(ξ*)'s
Subset Selection (RETINA)
Fast Subset Selection (Quicknet)
Select ξ*
Select Ψ*
Merge ξ* and Ψ*
Subset Section on ξ* and Ψ*
Suggested Specifications
Stage3

Figure 6.3: The RETINET algorithm

Based on the parsimony principle, RETINET should deliver a specification with just linear components when the highly non-linear transformations of the inputs do not provide any generalization ability (i.e. poor out-of-sample predictive ability).

However two situations should be distinguished with this outcome: the first is when no neglected (non-linear) structure is present in the residuals, which is by no means problematic for the purpose of prediction. The second situation, less satisfactory in terms of forecasting ability, is when non-linear structure is still present in the residual term. In this case, one should consider the presence of non-linearities in conditional moments other than the first, which as an example, may be due to heteroscedasticity. Finally, there exists always a possibility that poor predictive performance may be due to lack of relevant inputs, which are essential to capture completely the features of the target.

In what follows we evaluate the forecasting performance of RETINET's suggested models using a number of different criteria which in part have already been discussed in section 4.3.2. In particular we consider the out-of-sample RMSE and MAE statistics to compare the forecasts of the more elaborate RNET-ANN models with respect to RNET-LIN, as well as with respect to random walk forecasts. Loosely speaking, the comparison between RNET-ANN and RNET-LIN models is of special interest here in order to check whether the procedure behaves as expected by delivering an RNET-ANN specification only if needed. We check the results using the Harvey et al. (1997) - HLN - and the Diebold & Mariano (1995) - DM - test statistics of equivalent forecast errors.

In addition we also evaluate the ability of the various models to correctly forecast the sign of the target variable. This criterion is of special interest especially in financial applications since financial agents are mostly interested in predicting the sign of the returns in order to decide future investment actions. For this purpose we consider the Success Ratio (SR) statistic defined as:

$$SR = \frac{1}{m}\sum_{j=1}^{m} I_j[Y_{t+j} \cdot \hat{Y}_{t+j|t+j-1} > 0]$$

where I is an indicator function. Observe that SR is simply the proportion of m forecasts $\hat{Y}_{t+1|t+j-1}$ that have the same sign as the realizations Y_{t+j}, that is, the number of times the sign of $\hat{Y}_{t+1|t+j-1}$ is correctly predicted. Based on this measure Pesaran & Timmermann (1992) proposed a test of Directional Accuracy (DA) where the null hypothesis is that Y_{t+j} and $\hat{Y}_{t+1|t+j-1}$ are distributed independently.

In order to compare RNET-ANN against ML-ANN models, the same number of lagged realizations of the response were used as inputs. These were selected by

RETINET among the lagged inputs:

$$X_t = \{Y_{t-1}, Y_{t-2}, \ldots, Y_{t-10}\}$$

The number of hidden units were assumed to be same suggested by RETINET. This ensures that the comparison between both models is "honest" since they have similar characteristics.

For each DGP we generated a sequence of 400 realizations and kept 100 observations for out-of-sample predictive assessment. Forecasts were evaluated one-step ahead. Stationarity conditions were checked by means of various test on GLS de-trended series, including Phillips & Perron (1988) test. We consider the following five DGP's which are often used as examples of non-linear time series. Realizations for each DGP are reported graphically for the estimation as well for the test sample in figure 6.4. RETINET settings were as follows: we let the algorithm choose among the logistic, the radial basis function and the ridgelets activation function. The library Ψ included $v = 1000$ of each activation function at each iteration. A maximum of $Q = 20$ hidden nodes was allowed. For the sake of simplicity, no level one transforms were produced thus fitting an equation of the form of:

$$\hat{Y}_t = \hat{\alpha}'\tilde{X}_t + \sum_{h=1}^{q_{lgt}} \psi_{lgt}(X_t, \Gamma_h)'\hat{\beta}_{hh} + \sum_{h=1}^{q_{rbf}} \psi_{rbf}(X_t, \Gamma_h)'\hat{\beta}_{hh} + \sum_{h=1}^{q_{rdg}} \psi_{rdg}(X_t, \Gamma_h)'\hat{\beta}_{hh} \quad (6.6.1)$$

We now describe the DGP's used in this example:

Exponential Autoregressive (EXPAR) process

$$Y_t = [0.5 + 0.9\exp(-Y_{t-1}^2)]Y_{t-1} - [0.8 - 1.8\exp(-Y_{t-1}^2)]Y_{t-2} + \varepsilon_t$$

This is an exponential autoregressive model of order 2. Exponential autoregressive time series models can capture certain types of nonlinear dynamics, accounting for amplitude-dependent frequency, jump phenomena and limit cycles. This class of models were introduced by Haggan & Ozaki (1981) to represent time series that behave as nonlinear random vibrations. A realization of the process is represented in figure 6.4.

Stochastic Chaos (SC) Process:

$$Y_t = Y_{t-1}(1 - Y_{t-1})\varepsilon_t$$

with $Y_0 = .5$ and $\varepsilon_t \sim U[0, 1]$. This process generates only positive values and alternates periods of high volatility followed by flat intervals, hence it may be useful to represent either implied volatility or observed volatility processes in financial markets (McNelis 2005). The dynamic of the process is stable provided that the the starting value is $Y_0 \in [0, 1]$ otherwise it diverges quickly. Observe that a convenient re-parameterization of the process is to take logarithms on both side of the equation which yields:

$$\log(Y_t) = \log(Y_{t-1}) + \log(1 - Y_{t-1}) + \log(\varepsilon_t) \tag{6.6.2}$$

Notice that in this case we are violating the usual hypothesis of normal distributed errors, since the shocks in 6.6.2 are asymmetric and non-positive. A realization of the log-transformed process is shown in figure 6.4.

Self Exciting Threshold Auto-Regressive (SETAR) Process This is an example taken from Franses & van Dijk (2000). Among others, SETAR processes (Tong 1978, Tong & Lim 1980) are non-linear auto-regressive processes proposed to model non-linearities in returns of financial time series. SETAR models are auto-regressive models characterized by two or more regimes determined by the value of the lagged values of the series relative to a threshold value c. Here we consider two regimes determined by the value of the first lag with respect a threshold value of zero.

$$Y_t = \begin{cases} \phi_{0,1} + \phi_{1,1}Y_{t-1} + \phi_{2,1}Y_{t-2} & \text{if } Y_{t-1} > 0 \\ \phi_{0,2} + \phi_{1,2}Y_{t-1} + \phi_{2,2}Y_{t-2} & \text{if } Y_{t-1} \leq 0 \end{cases} \tag{6.6.3}$$

The autoregressive parameters were set equal to $\phi_{0,1} = 0$, $\phi_{1,1} = 0$, $\phi_{2,1} = 0$, $\phi_{1,2} = .5$ and $\phi_{2,2} = .14$, hence this is process is a White Noise in the regime $Y_{t-1} > 0$ while is an AR(2) if $Y_{t-1} \leq 0$. The threshold value is set to zero and $\sigma = .2$ is the same in both regimes. A realization of the process is represented in figure 6.4.

Bilinear (BIL) process

$$Y_t = \beta Y_{t-2}\varepsilon_{t-1} + \varepsilon_t$$

where $\beta = .6$ and $\sigma_\varepsilon = 1$. Granger & Andersen (1978) showed that this model has null auto-correlations at all lags and thus cannot be forecasted by linear models. A realization of the bilinear process is reported in fig.6.4.

GARCH(1,1)

$$Y_t = z_t\sqrt{h_t}$$

Table 6.1: Standard deviations and Median Absolute Deviation of simulated Time series discussed in section 6.6

	EXPAR	**SC**	**SETAR**	**BIL**	**GARCH**
Standard Deviation	1.393	3.810	0.338	1.279	0.576
Median Absolute Deviation	1.578	2.076	0.251	0.985	0.511

with $h_t = \omega + \alpha_1 Y_{t-1}^2 + \beta_1 h_{t-1}$. We set $\alpha_1 = .2$, $\beta_1 = .6$ and $\omega = 1 - \alpha_1 - \beta_1$ while the shocks z_t are distributed as a standard normal. GARCH (Bollerslev 1986) and ARCH models (Engle 1982) are widely used in finance for modeling conditional heteroscedasticity in financial time series. A realization of this process is illustrated in fig.6.4.

We do not expect RETINET and/or ML-ANN models to capture non-linearities in the case of the bilinear and GARCH(1,1) process, since the non-linearities embedded in these processes occur in the (conditional) second moment of Y_t, whereas ANN and RETINET are expected to capture non-linearities in the first (conditional) moment. Nonetheless we are motivated to use RETINET and ML-ANN in these settings because GARCH-type effects often arise in high frequency data and are one of the prominent features of daily and weekly financial data.

Also notice that some of the proposed processes may be well approximated by linear models. This is evident considering the autocorrelation (ACF) and partial autocorrelation (PACF) functions of each process which are reported in fig.6.5. In particular observe that the ACF of EXPAR process suggests an AR(1) process. SC's process is clearly auto-regressive and has the root of the characteristic equation closer to the non-stationary region. SETAR's process has a ACF dying out slowly and a PACF off after the first lag which may suggest an ARMA(1,1) process with both negative AR and MA coefficient. Standard deviations and Median Absolute Deviations of simulated Time series are reported in table 6.1.

6.6.1 Results

The results of this simulation exercise are reported in tables 6.2,6.3 and 6.4. Table 6.4 reports the forecasts error statistics both in-sample and out-of-sample.

Overall specifications obtained for RNET-ANN models included at most three out of ten possible lags of the response that were used as inputs. This is shown in table 6.2. Only in the case of the GARCH specification, RNET-LIN included just a constant. Further, and only for the GARCH DGP, the final proposed specification RNET-ANN included just a constant, meaning that no non-linear terms were retained by

Figure 6.4: Simulated Time Series

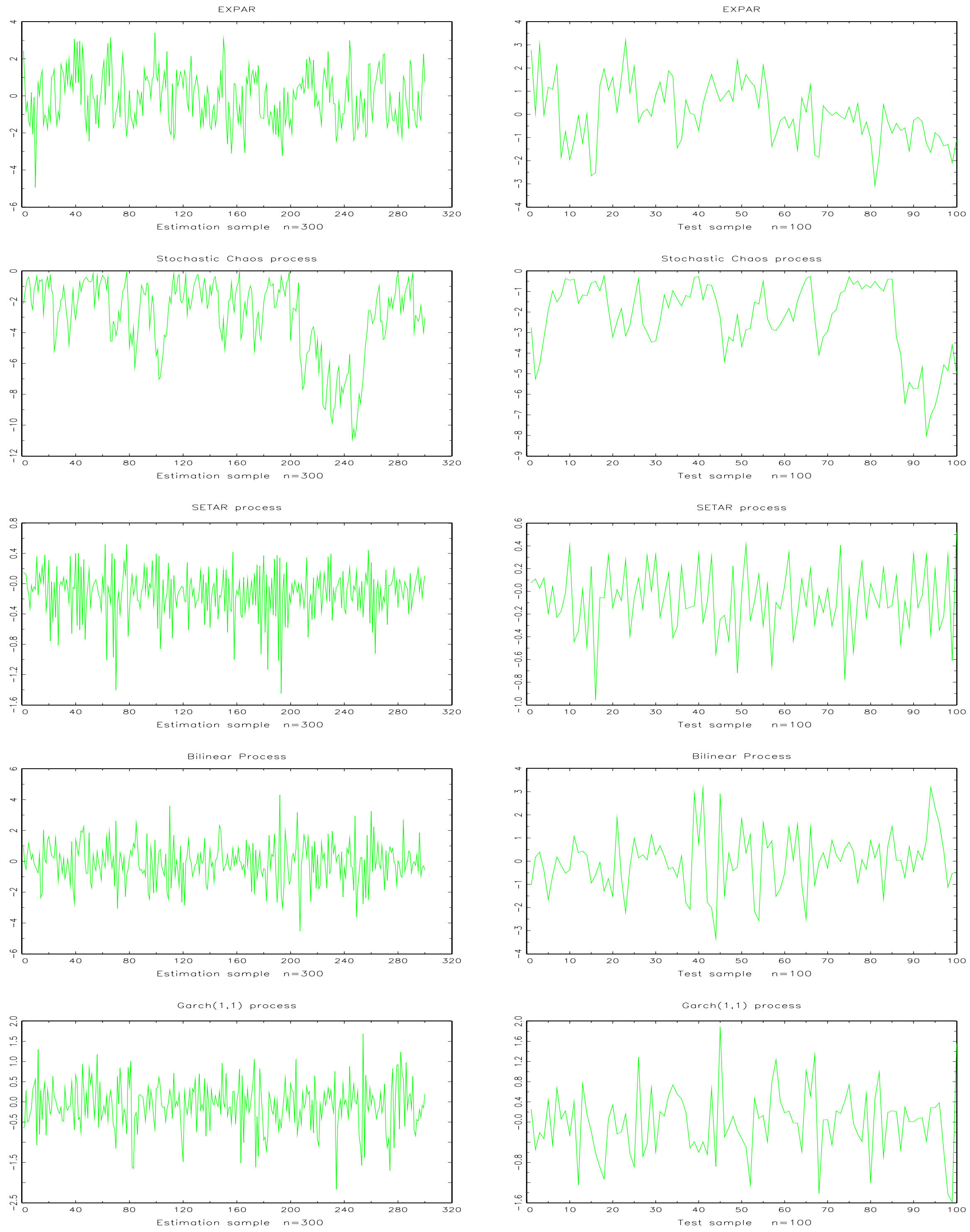
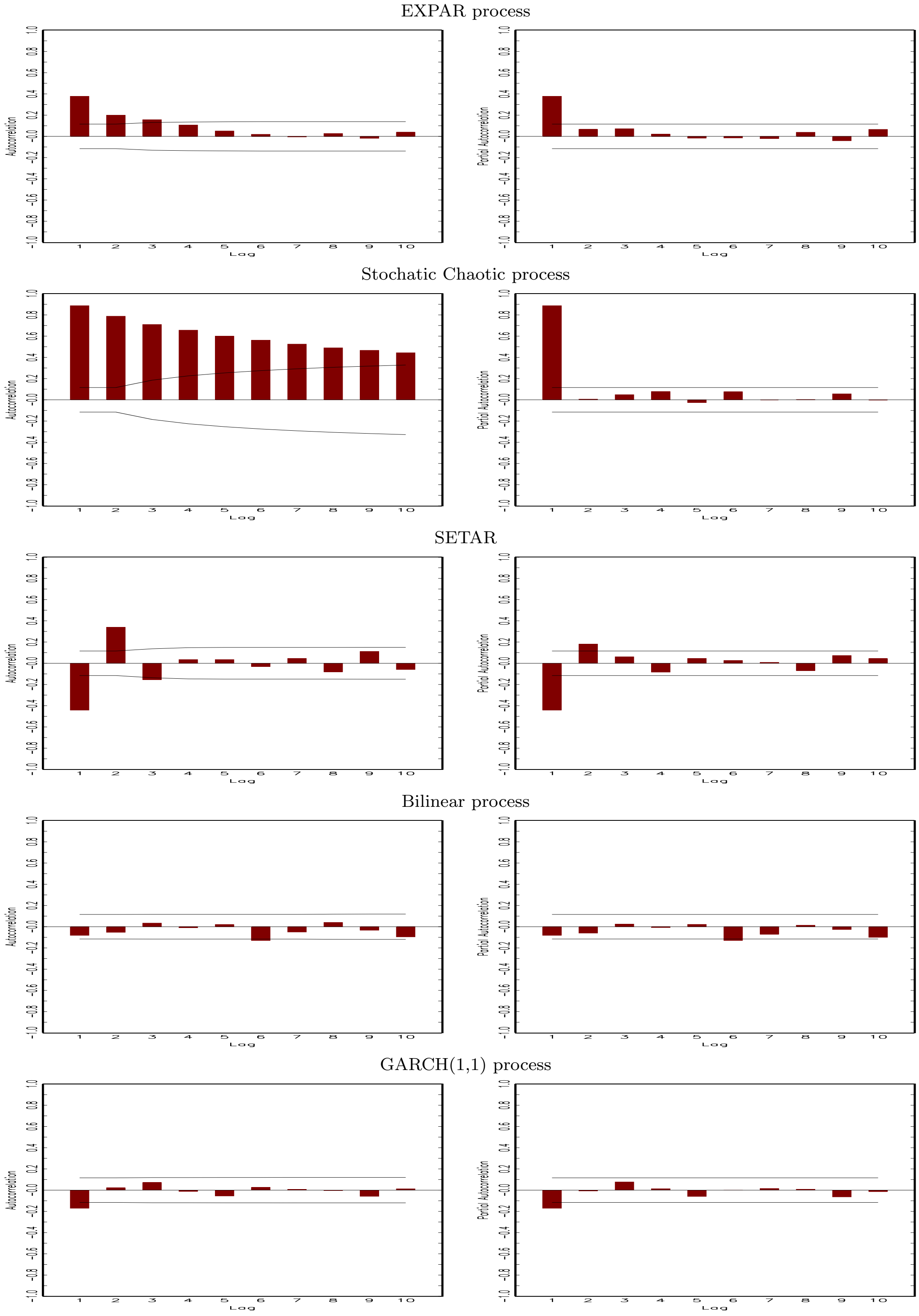


Figure 6.5: Conditional mean ACF (left) and PACF (right) simulated Time Series in fig. 6.4 †. (Horizontal lines indicate confidence interval bounds)



EXPAR process

Stochatic Chaotic process

SETAR

Bilinear process

GARCH(1,1) process

Table 6.2: Details about RNET-LIN, RNET-ANN and ML-ANN models as well as ML-ANN optimization details of simulated DGP's discussed in section 6.6. ML-ANN stands for Maximum Likelihood estimated ANN. For the GARCH(1,1) process, no ML-ANN was estimated since RETINET suggested a specification including just a constant. For all other DGP's, ML-ANN's are characterized by using a logistic activation, the same inputs and the same number of nodes selected by RETINET. Notice that ML optimization always converged although the Hessian matrix failed to invert most of times due to numerical problems.

		EXPAR	**SC**	**SETAR**	**BIL**	**GARCH**
RETINET	Lags selected	Y_{t-1}, Y_{t-3}	Y_{t-1}	Y_{t-1}, Y_{t-2} Y_{t-9}	Y_{t-1}	Constant
	# Nodes	2	7	4	4	-
	# Ridgelets	2	6	3	4	-
	# Rad.Basis	-	1	-	-	-
	# Logistic	-	-	1	-	-
ML-estimated ANN	ML convergence	Ok	Ok	Ok	Ok	-
	Hessian Inversion	Ok	Failed	Failed	Failed	-

the algorithm. Hence, in this case the RNET-ANN solution is equivalent to the linear RNET-LIN solution.

For all the remaining DGPs, the RNET-ANN solutions differed from the RNET-LIN, and a number between 2 (EXPAR) and 7 (SC) non-linear terms were selected. Among these different types of activation functions were used, although ridgelets appeared more often than others, confirming their good approximation properties especially in presence of sharp profiles of the data. Maximum Likelihood estimated ANN optimization routines always converged although inversion of the Hessian matrix failed most of times, which reflects the fact that numerical problems in non-linear optimization routines are always a possibility, even in naive setting like this.

From table 6.3, comparing RETINET final models (RNET-ANN) versus alternative specifications does not generally yield significant differences (measured by DM and HLN statistics), excluding for forecasts produced by random walk models, which in almost all cases have lower predictive ability. Only in the case of the SETAR DGP, RNET-ANN model outperforms at a 10% significance level the RNET-LIN forecasts.

However it must be noticed that numerical results of out-of-sample RMSE, MAE and SR in table 6.4 show a general trend where the RNET-ANN models always outperform the forecasts provided by RNET-LIN models and RW models. This is encouraging since the algorithm provides RNET-ANN solutions which consistently

Table 6.3: Test statistics of equivalent forecast errors of the RNET-ANN estimated model versus the RNET-LIN, the RW and ML-ANN model, considering a quadratic loss function (QLOSS) and an absolute loss function (ALOSS).

Statistic	Model	**EXPAR**	**SC**	**SETAR**	**BIL**	**GARCH**
DM-QLOSS	RNET-LIN	0.750	0.573	1.728*	0.812	-
	RW	1.116	4.408**	12.350**	1.591	9.578
	ML-ANN	-0.578	2.892**	1.505	-0.573	-
DM-ALOSS	RNET-LIN	0.185	1.476	1.769*	0.862	-
	RW	0.665	10.400**	19.150**	1.970**	7.618**
	ML-ANN	-1.147	3.561**	0.398	0.215	-
HLN-QLOSS	RNET-LIN	0.483	-0.192	0.837	-0.051	-
	RW	1.427**	9.756**	6.151**	5.214**	14.260**
	ML-ANN	-0.147	0.372	0.072	-0.152	-

Test are based on the Diebold & Mariano (DM) statistic and the Harvey, Leybourne, Newbold, (HLN) statistic. The latter is valid only under quadratic loss functions, hence tests for equality absolute errors were computed only for the DM statistic.$(**, *)$ indicate respectively $(5\%, 10\%)$ significance levels. Positive entries refer to a higher predictive ability of RNET-ANN models with respect to RNET-LIN, RW and ML estimated ANN (ML-ANN).

result in a lower out-of-sample forecast error respect to simpler linear specifications; see table 6.4. In fact RNET-ANN performs reasonably well compared both to simpler linear specifications such as random walk forecasts and purely linear RNET-LIN specification in almost all cases.

When comparing with respect to ML estimated networks, the out-of-sample performance of RNET-ANN models, in almost all cases, is quite similar and in some cases better. See the out-of-sample RMSE results for the bilinear, BIL, and the stochastic chaotic model, SC. Nonetheless notice that, in terms of directional accuracy, the forecasts provided by RNET-ANN are consistently better than those provided by alternative methods. See the case of SETAR, EXPAR and BIL processes in table 6.4. In the case of the bilinear process, RNET-ANN and RW directional forecast are more accurate than maximum likelihood estimated neural networks.

These results, provide evidence that RETINET models are, in terms of predictive ability, at least not worse than more computationally intensive Maximum Likelihood estimated ANN. Also the algorithm behaves as expected, providing parsimonious solutions in most situations, finding an adequate balance between bias and variance. More tests are needed to complement the evidence reported here. Nonetheless we consider that these results are encouraging and suggest that the algorithm is capable of detecting informative non-linearities in the first conditional moment where these exist, and avoids some numerical inconveniences typical of non-linear models.

Table 6.4: One step ahead forecasting performance of RETINET versus alternative models for DGP's discussed in section 6.6. In comparison RNET-LIN and the RW models, the RNET-ANN model shows consistently lower out-of-sample RMSE and higher direccional accuracy measured by the SR statistic.

			RNET-LIN	RNET-ANN	RW	ML-ANN
EXPAR	In-sample	RMSE	1.280	1.122	1.546	1.176
		MAE	1.398	1.222	1.485	1.215
		SR	69%	72%	67%	68%
	Out-of-sample	RMSE	1.144	1.077	1.359	1.077
		MAE	0.974	1.033	1.152	0.964
		SR	75%	76%	72%	73%
SC	In-sample	RMSE	1.113	0.868	1.145	1.068
		MAE	0.832	0.738	0.938	0.754
		SR	100%	100%	100%	100%
	Out-of-sample	RMSE	0.985	0.959	1.049	0.980
		MAE	0.771	0.735	0.937	0.731
		SR	100%	100%	99%	100%
SETAR	In-sample	RMSE	0.276	0.207	0.533	0.203
		MAE	0.262	0.209	0.446	0.215
		SR	64%	67%	42%	63%
	Out-of-sample	RMSE	0.274	0.243	0.475	0.240
		MAE	0.263	0.273	0.451	0.237
		SR	53%	65%	35%	63%
BIL	In-sample	RMSE	1.271	1.136	1.876	1.268
		MAE	1.021	1.068	1.552	1.01
		SR	54%	63%	51%	55%
	Out-of-sample	RMSE	1.245	1.232	1.717	1.250
		MAE	1.098	1.167	1.439	1.094
		SR	43%	55%	57%	42%
GARCH	In-sample	RMSE	0.570	0.570	0.836	-
		MAE	0.511	0.511	0.624	-
		SR	56%	56%	47%	-
	Out-of-sample	RMSE	0.644	0.644	0.952	-
		MAE	0.597	0.597	0.643	-
		SR	45%	45%	53%	-

Comparisons across column show forecasting performance, in-sample and out-of-sample, relative to: 1) RETINET's linear model (RNET-LIN) which is nested into 2) RETINET's final specification (RNET-ANN) 3) the random walk forecast (RW) and 4) the ANN model estimated by Maximum Likelihood (ML-ANN). See table 6.2 for further details about the characteristics of the RNET-ANN and the ML-ANN models. See section 6.6 for an explanation of forecast measures, RMSE, MAE and SR. Notice that the predictive ability of RNET-ANN models is similar compared to the ML estimated ANN models. In the case of the GARCH process RETINET delivered a specification including just the constant, and as such, RNET-LIN and RNET-ANN are equivalent and no ML-ANN was estimated.

6.7 An application to Geophysics

In this section we show an application of RETINET to solve a problem of geophysical sciences. More details about the results of this work may be found in Karimabadi, Sipes, White, Marinucci, Dmitriev, Chao, Driscoll & Balac (2007). We are motivated to use the RETINET strategy here because of its reverse engineering properties and partial interpretability of its suggested specification. Actually this investigation has been the starting point for developing an integrated datamining open-source software denominated *Minetool* which integrates RETINET among other applications designed especially for scientific data-mining[4]. From now on we will refer to it as the RETINET-Minetool package.

The object of investigation is a simulated *magnetopause* data set. The magnetopause is the thin boundary separating the shocked solar wind plasma from the plasma of the magnetosphere of the earth. The form of this boundary varies depending on some physical magnitudes such as the solar wind intensity and the earth's magnetic field. The magnetopause has a bullet-shaped front, gradually changing into a cylinder. Its cross-section is approximately circular. An illustration of the magnetopause is represented in figure 6.6.

Our starting point is an empirically derived model of magnetopause by Shue, Kokubun, Song, Russell, Steinberg, Chao, Zastenker, Vaisberg, Singer & Detman (1998) which was obtained by using a least squares fit to a pre-defined functional form using spacecraft data:

$$R = R_0 \left(\frac{2}{1 + \cos\theta} \right)^{\alpha} \tag{6.7.1}$$

where:

$$\alpha = (.58 - .01B_z)(1 + .01D_p)$$

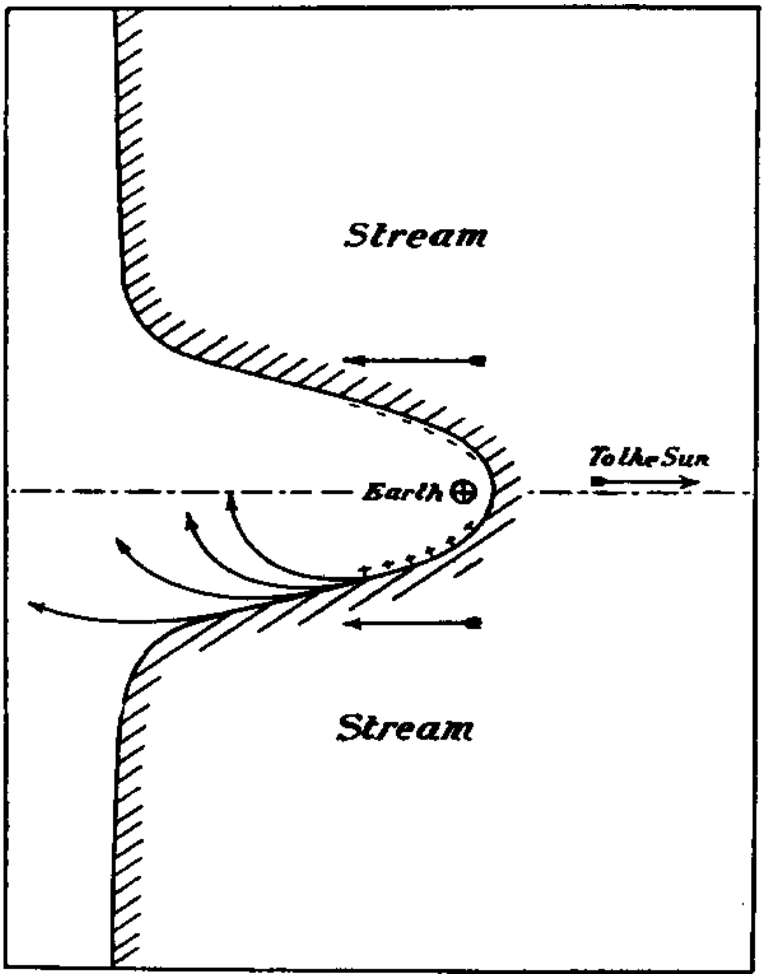
and:

$$R_0 = \begin{cases} (11.4 + .013B_z)D_p^{-1/6.6} & \text{if } B_z \geq 0 \\ \\ (11.4 + .140B_z)D_p^{-1/6.6} & \text{if } B_z \leq 0 \end{cases} \tag{6.7.2}$$

Here R and θ are polar coordinates representing the position of the magnetopause, and B_z and D_p are the $z-$component of the interplanetary magnetic field (IMF) and solar wind dynamic pressure, respectively. This model has a complex dependence on B_z and D_p including a change in the functional form as a function of sign of B_z.

[4] Current efforts are concentrated on time series classification tools. RETINET-MineTool is being implemented in C code by Sciberquest Inc., Solana Beach, CA, USA.

Figure 6.6: The Magnetopause of the Earth. Source: National Aeronautics and Space Administration (NASA)



We find it convenient to work with the following variables: $\ln(R)$, $\cos(\theta)$, $\ln(Dp)$, and B_z. We then generate a data set of $\ln(R)$ for a range of values in $\cos(\theta)$, $\ln(Dp)$, and B_z. The idea is to use the resulting data set and derive a model for:

$$\ln(R) = f(\cos(\theta), \ln(D_p), B_z) \tag{6.7.3}$$

which can then be compared against the original equation 6.7.1. We consider two cases: (i) noiseless case and (ii) noisy case where:

$$\ln(R') = \ln(R) + \sigma\epsilon \tag{6.7.4}$$

with R' representing the magnetopause distance from noise. Here R is from eq. 6.7.1, σ is the deviation of standard gaussian distribution defined on the interval $[4RE, 20RE]$ where RE is the earth's radius[5]. These boundaries establish a realistic representation of the orbital bias in experimental measurements, which are usually restricted by a satellite perigee and apogee.

6.7.1 Results

In this section we compare the results, including the resulting equations, from various algorithms. We use three different error measures, root mean squared error (RMSE), mean absolute error (MAE), and mean relative error (MRE) as defined in section 4.3.2, to compare performance across models. In all cases we choose a Benchmark Linear Model (BLM) based on linear regression of the three normalized[6] predictors $\cos(\theta)_s, \ln_s(D_p)$, and B_{zs}:

$$\widehat{\ln(R)} = 0.986 - 0.063\cos(\theta)_s - 0.084\ln_s(D_p) + 0.025B_{zs} \tag{6.7.5}$$

Noiseless Case

A first RNET-LIN specification was found using 43 double level 1 transformations of the form $\zeta[\zeta(X_t)]$ (see section 3.1.1) including the constant term. The 10 leading terms are listed in Table 6.5. Further, the algorithm provided an improved specification including ANN transformations of the original inputs X_t. The resulting RNET-ANN specification retained 33 terms from the RNET-LIN specification plus 45 ANN terms which included both ridgelets and radial basis transformations.

[5] Distances in the magnetosphere are often measured in Earth radii (RE), with one Earth radius amounting to 6371 km or 3960 miles. In these units, the distance from the Earth's center to the "nose" of the magnetosphere is about 10.5 RE and to the flanks abreast of the Earth about 15 RE, while the radius of the distant tail is 25-30 RE. By way of comparison, the moon's average distance is about 60 RE.

[6] The sub index s refers to normalized variables.

Table 6.5: RETINET-MineTools 10 leading terms of the Best Linear Model. The term ρ stands for the bivariate correlation of each predictor with the response.

Predictor	ρ	Coeff.
$\ln_s(D_p)$	0.757	-9.15E-02
$\cos(\theta)_s$	0.680	7.59E-03
$\cos(\theta)_s \ln_s^2(D_p)$	0.578	-5.67E-02
$[\ln_s(D_p)]^3$	0.564	1.20E-03
$\cos_s(\theta) B_{zs}$	0.531	-1.73E-03
$\cos_s(\theta)[\ln_s(D_p)]^3$	0.463	-7.58E-03
B_{zs}	0.218	3.75E-02
$\cos_s(\theta) \ln_s(D_p) B_{zs}$	0.166	-2.32E-04
$\cos_s(\theta)^2$	0.155	-2.30E-03
$\cos(\theta)[\ln_s(D_p)]^2 B_{zs}$	0.155	3.77E-04

Table 6.6: Performance comparison of the two RETINET models on the hold-out data

	Benchmark Model	**RNET-LIN**	**RNET-ANN**
RMSE	$2.66E-02$	$8.56E-04$	$3.42E-04$
MAE	$2.06E-02$	$5.89E-04$	$2.52E-04$
MRE	$-9.30E-04$	$1.58E-05$	$-1.17E-05$

Table 6.6 compares the performance of the three suggested specifications on the hold-out data. A visual method of gauging the performance of the results is to plot the actual versus the predicted values as shown in Fig. 6.7. In the zero error limit, all data will be lined up along the 45° line. Visual inspection of this figure along with the error measures in table 6.6 reveal a number of interesting points. First, the simple regression model (benchmark) does a reasonable job and with RMSE of about 0.026 would be considered adequate for most space physics applications. Secondly, the RNET-ANN model achieves an amazingly high accuracy. Figure 6.8 shows the distribution of the MRE as a function of the inputs $\ln(R)$, B_z, $\cos(\theta)$, and $\ln(D_p)$. Such a figure can be used to help identify any heteroscedasticity in the model performance as may occur if the quality of data (e.g., coverage, noise, etc.) varies significantly across observations. In the present case the data are more sparse at large values of B_z and dynamic pressure, but the error shows a fairly homoscedastic distribution.

Comparison with other Data Mining Models

Models obtained so far by applying RETINET were compared with a set of standard data mining techniques described in the appendix (section 6.9.2).

Figure 6.7: Plot of actual versus predicted values for a benchmark model based on linear regression and RNET-LIN and RNET-ANN models

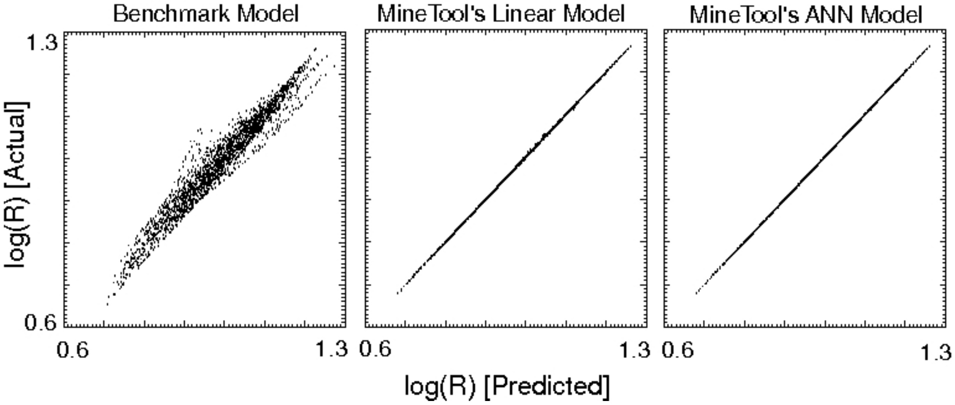


Figure 6.8: Plots of relative error as a function of log(R) and three input variables for each of the benchmark linear model, RNET-LIN and RNET-ANN models.

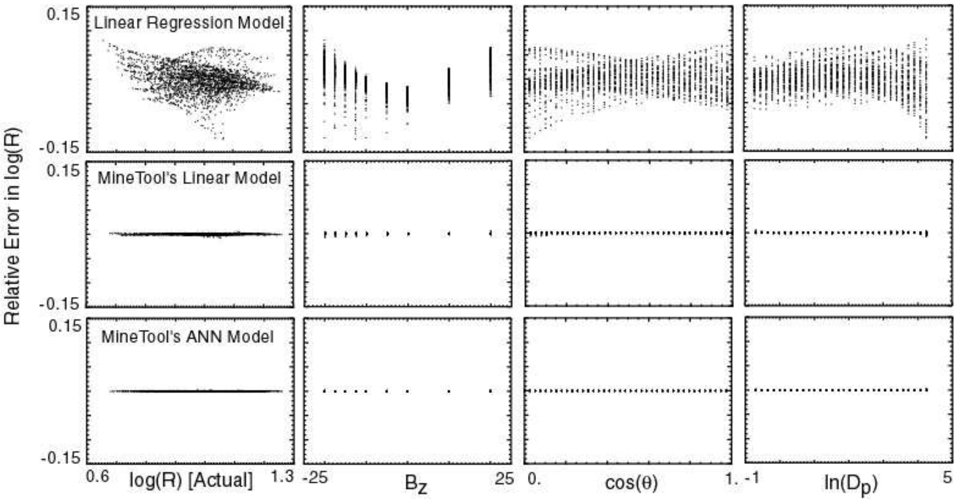


Table 6.7: Performance comparison of alternative data-mining techniques and commercial packages. See section 6.9.2 for further details.

	RMSE	MAE	MRE
Benchmark	2.66E-02	2.06E-02	-9.30E-04
RNET-LIN	8.56E-04	5.89E-04	1.58E-05
RNET-ANN	3.42E-04	2.52E-04	-1.17E-05
Genetic	2.25E-03	1.17E-03	4.50E-03
Neuro	4.90E-03	3.70E-03	-1.60E-05
GMDH	4.50E-03	3.30E-03	-1.10E-05
MT	7.40E-03	4.95E-03	2.70E-04
RT	9.80E-03	8.30E-03	3.52E-05
ANN	1.10E-02	9.00E-03	-8.70E-03
SVM	2.25E-02	1.63E-02	-2.90E-03
RBF	1.38E-02	1.07E-02	-5.90E-04
PR	2.75E-02	2.10E-02	-1.00E-03
Bagging-MT	6.00E-03	4.79E-03	1.97E-04
Bagging-RT	6.00E-03	4.79E-03	1.97E-04
Bagging-ANN	6.80E-03	4.80E-03	-1.60E-04
MS	9.80E-03	8.40E-03	2.26E-05

Nomenclature: GMDH (Group Method of Data Handling), MT (Model Tree), RT (Regression Tree), NN (Neural Networks), SVM (Support Vector Machines), RBF (Radial Basis function Network), PR (Pace Regression), MS (Multi-Scheme)

Table 6.7 show the performance accuracy of each of the models. Overall, the Model Tree (MT), Regression Tree (RT) and Multi-Scheme methods gave the best non-RETINET-MineTool, basic method results, with an RMSE of 0.0074 and 0.0098 respectively. The MT, having given a smaller output evaluated slightly better, most likely due to offering a more general output model that tends to overfit less than the more detailed RT. Bagging (see section 6.9.2) of the top three basic methods (MT, RT and ANN) definitely increased their performance, especially in the case of ANN, RMSE decreased from 0.0110 to 0.0068. Excluding RETINET-Minetool models, the best overall performers are bagged model trees and bagged regression trees, evaluating at the RMSE of 0.0060. These two models are similar in how they choose the attributes to split the search space and best distinguish among the input space, and therefore, when bagged, gave identical composite results. As figure 6.9 illustrates, even the best performers in Weka (Frank, Hall, Trigg, Holmes & Witten 2004) do not achieve the same accuracy as the Neural Net package of Ward Systems or RETINET-MineTool in this example. This is partly due to the fact that Weka includes many different data mining algorithms and as a result some of the algorithms are not as fine tuned as the specialized packages that concentrate on one or two types of algorithms. Although RETINET-MineTool provides the highest accuracy, most models provided accuracy that would be adequate for most space physics applications. None of the other techniques except pace regression provides an analytical form. In short, RETINET-MineTool provides the most accurate model and has the added advantage that the solution is in analytical form.

Effect of Noise in the Data

We now generate a data set based on Eq. 6.7.4 with $\sigma = 0.1$. It is easy to show that this puts a theoretical limit of 0.1 in the accuracy of the forecasts that can be obtained. Using RETINET-MineTool, the modeling steps are the same as for the noiseless case. The benchmark model is similar in form to that for the noiseless case but with different coefficients:

$$\widehat{\ln(R)} = 0.988 + 0.061\cos(\theta)_s - 0.080\ln_s(D_p) + 0.024B_{zs} \qquad (6.7.6)$$

The best candidate model now consists of only 12 terms (including the constant) as compared to 43 terms for the noiseless case. This is because the presence of noise puts a theoretical limit on the level of accuracy that can be achieved, thereby limiting the number of terms required. This testifies to the power of our algorithm which only keeps the minimum number of terms required to achieve the desired accuracy and

Table 6.8: RETINET-MineTools 10 leading terms of the RNET-LIN model in the noisy case. The term ρ stands for the bivariate correlation of each predictor with the response.

Predictor	ρ	Coeff.
$\ln_s(D_p)$	0.556	-8.90E-02
$[\ln_s(D_p)]^3$	0.500	9.70E-03
$\cos(\theta)_s$	0.426	-5.70E-02
$\sin_s(\theta)[\ln_s(D_p)]^2$	0.339	-6.16E-03
B_{zs}	0.160	3.40E-02
B_{zs}^3	0.097	-1.20E-02
$\cos_s(\theta)B_{zs}$	0.083	1.20E-02
$\sin_s(\theta)[\ln_s(D_p)]^3$	0.061	-7.70E-03
B_{zs}^{-2}	0.058	1.00E+00
B_{zs}^{-3}	0.058	3.39E+00

hence avoiding overfit. The top 10 explanatory variables are listed in 6.8. The neural RNET-ANN model (not shown) consists of 8 terms plus 6 hidden layer terms that involve ridgelets, radial basis functions and logistic functions. Table 6.9 compares the relative performance of the RNET-ANN and RNET-LIN models as well as that from ANN routine GMDH from Ward Systems. Note that the benchmark and the best model yield very similar results to the RNET-ANN model which is only slightly less accurate. GMDH yields somewhat more accurate result than RNET-ANN model but all four models in this case are very comparable in performance. Table 6.9 shows the performance comparison of the basic and advanced methods in Weka. The accuracy of predictions is very close to the theoretical value in all models. This is further illustrated in figure 6.10 where we plot the actual versus predicted values of the response. The spread about the 45° slope in all cases is due to the presence of noise in the data which limits the accuracy of prediction. If particular attention is paid to the RT model plot in figure 6.10, one notices vertical stripes in the model output. This is due to the RT model slightly generalizing the output and assigning it a certain value, multiple times inside the tree structure, where actual values of the target were around that predicted value. The slight over-generalization is caused by the presence of noise that was added to the data set. Because the data set was fairly noisy, the RT method simply tried to generalize and extract the "gist" from the noisy data.

Table 6.9: Performance comparison of alternative data-mining techniques and commercial packages. See section 6.9.2 for further details.

	RMSE	MAE	MRE
Benchmark	9.59E-02	7.68E-02	1.12E-02
RNET-LIN	9.66E-02	7.73E-02	-1.10E-02
RNET-ANN	9.64E-02	7.70E-02	-1.00E-02
GMDH	9.90E-02	7.90E-02	1.00E-02
MT	9.70E-02	7.80E-02	1.18E-02
RT	9.76E-02	7.80E-02	1.18E-02
ANN	9.60E-02	7.70E-02	1.12E-02
SVM	1.00E-01	8.00E-02	1.10E-02
RBF	9.80E-02	7.87E-02	1.15E-02
PR	1.00E-01	8.00E-02	1.10E-02
Bagging-MT	9.75E-02	7.80E-02	1.11E-02
Bagging-RT	1.00E-01	8.00E-02	1.00E-02
Bagging-ANN	9.60E-02	7.70E-02	1.11E-02
Multi-Scheme	9.60E-02	7.70E-02	1.00E-02

Nomenclature: GMDH (Group Method of Data Handling), MT (Model Tree), RT (Regression Tree), NN (Neural Networks), SVM (Support Vector Machines), RBF (Radial Basis function Network), PR (Pace Regression)

Figure 6.9: Plots of actual versus predicted values of log(R) for various methods in Weka in hold-out data. NN refers to the artificial neural net algorithm in Weka software whereas genetic and neural refer to artificial neural net algorithms in Ward System package.

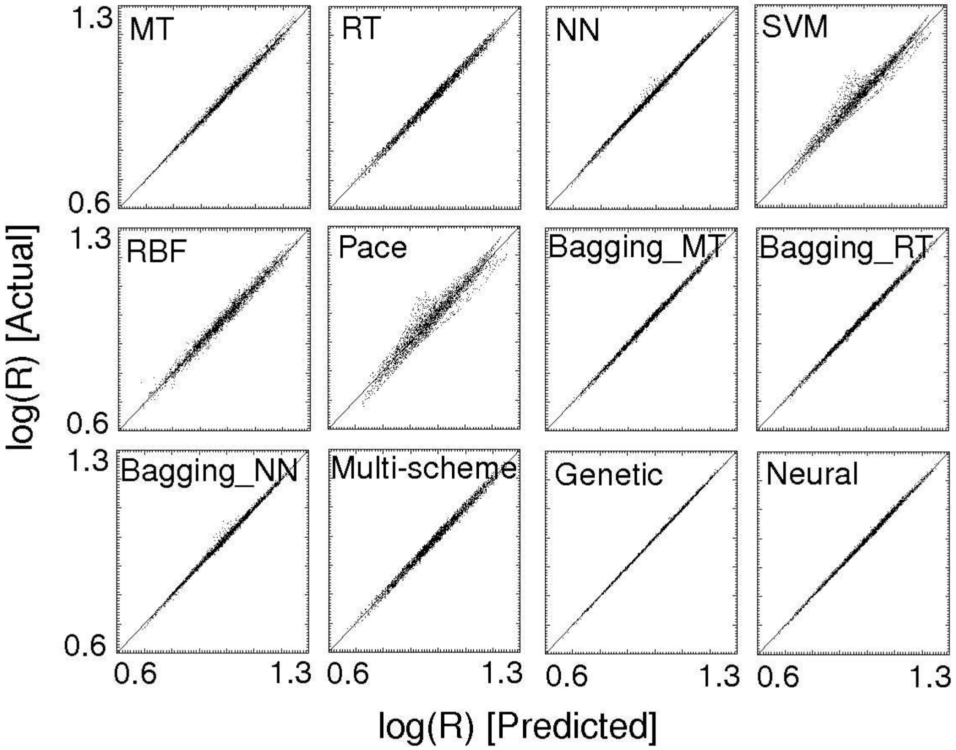
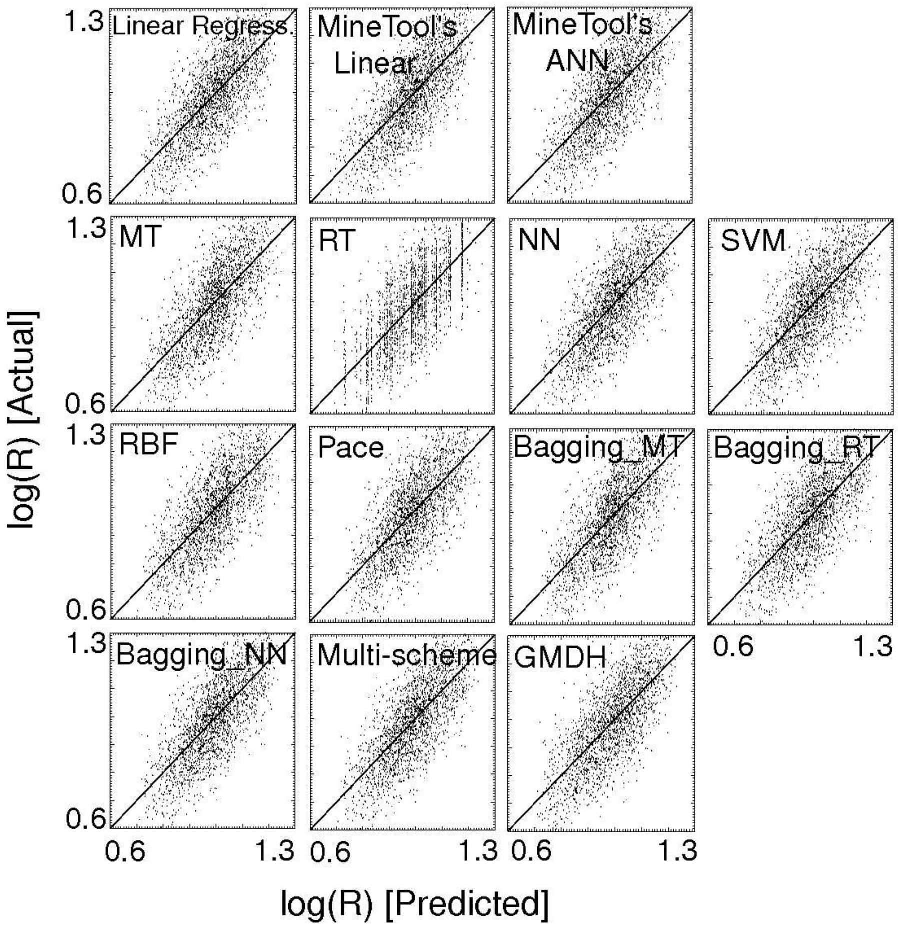


Figure 6.10: Plots of actual versus predicted values of the response from a variety of models in the presence of noise, using the hold-out data.



6.8 Conclusions

In this chapter we presented the RETINET algorithm which avoids technical difficulties related with non-linear estimations procedures to compute the weights of the ANN transformations. It also provides guidance about 1) the decision on which inputs X_t to use for the construction of the ANN transformations and 2) the selection of the most promising ANN candidate transforms to choose. Non-linear estimations procedures are avoided since the weights of the hidden units are generated randomly in virtue of the results discussed in section 3.1.4.

Being the specified equations always linear in the parameters OLS estimation techniques can be used, and computations, even if intensive, can be performed reasonably fast. The algorithm presented here differs from White's (2006) Quicknet in that we use RETINA to control for over-fitting and model evaluation method. Finally the procedure chooses among a mix of three different types of squashing functions, namely Logistic, Radial Basis and Ridgelets ANN, which provides a library of approximants and hence, more parsimonious models may be obtained.

Our choice is obviously not intended to be limited to these three functions but represents an example of how the Quicknet network building strategy has been extended in the present context. Obviously many other powerful approximation methods that are special cases of ANN may be considered, such as those discussed briefly in chapter 3. The fact that we don't restrict the inclusion of just one basis function at a time, nor adopt any specific order for their inclusion in the approximation function, is dictated by the consideration that by letting the algorithm choose suitably which basis function to adopt in any given instance, one may obtain a better final approximation. These are important advantages compared with networks estimated by non-linear methods.

A summary of the differential characteristics between RETINET, Quicknet and non-linear estimated ANN is reported in table 6.10. Another advantage of RETINET is that the final model to some extent retains analytical interpretability. This facilitates easier dissemination of the model as well as exploration of the effects of individual or groups of terms.

Table 6.10: Differential characteristics of Traditional, non-lineraly estimated ANN, RETINET and White's prototype Quicknet

	Traditional ANN†	**Prototype Quick net**	**RETINET**
Linearity/Non linearity	Non linear in the parameters and predictors.	Linear in the parameters, Non-Linear in the predictors.	Linear in the parameters, Non-Linear in the predictors.
Estimation of parameters	Non-Linear LS or Maximum Likelihood	OLS	OLS
Linear component (Jump Connections)	Usually there are no linear components although there are no major limitations to this	No, the user has to specify the linear component to start with.	Yes, using a previous RETINA building and selection stage. In this stage simple non-linear transformations of the inputs may be used (level 1 transforms).
Inputs	The user defines the inputs to build ANN transforms	The user defines the inputs to build ANN transforms	May use all inputs (single layer) or chose among level one transforms previously selected by RETINA (double layer network)
Bases implemented	Usually Logistic	Logistic, Ridgelets	Logistic, Radial Basis, Ridgelets
Network Architecture selection (specification)	Information/cross-validations criterion (AIC, BIC)	Information/cross-validations criterion (AIC, BIC)	RETINA selection, at the end, AIC.
Computational Issues	Non-convexity of the objective function.	Convex objective function.	Convex objective function. Uses updating of moment matrix, no computations from scratch when adding bases.

6.9 Appendices

6.9.1 Fast Computation of Ridgelets transforms

As already seen in section 3.1.3, in order to compute ridgelets transforms we may consider any particular function ψ which has vanishing moment (see eq. 3.1.4). A function satisfying such condition is the $j-$th derivative of the standard normal density where $j = p/2$ given p predictors. From a practical point of view, to build a ridgelet transformation when the inputs p are varying, implies that we need to compute the appropriate derivatives $\psi(X) = D^h\phi$ where $h = p/2$ and $D = d/dX$ of the standard normal function for each $p = 1, \dots, P$ predictor. One would like to have a method that computes quickly any of such derivatives for a different number of inputs. This is essential for automatizing the computation of ridgelets. Fortunately there is a simple way to compute successive derivatives of the standard normal. Let ϕ^i with $i = 1, 2, \dots$ be the $i-$th derivative of $\phi(X)$. Then:

$$\begin{aligned}
\phi'(X) &= -X\phi(X) \\
\phi''(X) &= (X^2 - 1)\phi(X) \\
\phi'''(X) &= -(X^3 - 3X)\phi(X)
\end{aligned}$$

It can be shown that it is possible to compute the derivatives of any order of the normal density distribution by considering a family of polynomials better known as *Hermite* polynomials which satisfy the following recursive relationship:

$$H_i(X) = XH_{i-1} - (r-1)H_{r-2}$$

where it is agreed by convention that $H_0(X) = 1$. The interesting thing about the Hermite polynomials in this context is that it suffices to multiply the normal density by those functions, the result is the $i - th$ derivative of the normal density:

$$\phi^i(X) = (-1)^i H_i(X)\phi(X)$$

6.9.2 Data Mining Algorithms

There is a plethora of data mining algorithms and approaches such as Bayesian techniques, genetic programming and machine learning, artificial neural networks (ANN), evolutionary algorithms, and Support Vector Machines (see Berthold & Hand (2003) for a review). Here we describe several of the more commonly used techniques that we tested and compared with RETINET-MineTool implementation.

Artificial Neural Networks: We used several ANN packages including Matlab, Ward System (Ward Systems Group, Inc., 1997), and Weka (Frank et al. 2004). We found the package from Ward Systems to be the fastest and it led to the very accurate results. In the noiseless case, we used two different paradigms in the NeuroShell Predictor, the Genetic Method and the Neural method.

The Neural method took 15 seconds to train on Pc with 2GHz speed and 2 Gb ram memory, and the genetic method 10 minutes before we stopped it. For comparison, the four-step RETINET-MineTool modeling process typically takes about 7 minutes with the training time taking only 2-3 seconds for this data set. The speed advantage of RETINET-MineTool becomes even more apparent for large data sets.

We also used another ANN technique from an older version of NeuroShell called Method of group account of arguments or Group Method of Data Handling (GMDH) (Farlow 1984). GMDH involves building a sequence of layers with complex links, which represent different parts of a polynomial. The polynomial parts are generated using linear and non-linear regressions. The initial layer is just a simple input layer. The first layer is created as a polynomial of input nodes that is selected as the best one from various polynomial structures (candidates). The best candidate is called a “winner”. A special algorithm (in this case the genetic algorithm) selects the winners.

The second layer is constructed as a polynomial, which uses both the network input nodes and output nodes of the first layer. The third layer uses the input nodes and output nodes of the second layer. The number of layers increases and the process is repeated from layer to layer, until the network cannot achieve more accuracy based on a pre-selected criterion. Finally the network represents a polynomial expression.

The GMDH can use different criteria to stop the training. We used a criterion of “calibration”, which requires a test data set for determination of the “best” model. The genetic algorithm is incorporated into the GMDH as an independent algorithm, which is used for generation of a variety of the polynomial forms and selection of the winners.

Ward Systems also has a genetic programming software that is not yet publicly released. This software yielded less accurate results than the two paradigms in the NeuroShell Predictor and it took much longer to run but it yielded simple analytical forms unlike GMDH or neural nets.

We also found similar results using other genetic programming (GP) software. GP tends to be very time consuming and less accurate but results in analytical form of the solution, which has its advantages over the output of traditional ANN techniques. The rest of the methods we used were all implemented in Weka. We explored several basic methods, such as model trees (MT), regression trees (RT), support vector machines (SVM), radial-basis function networks (RBF) and Pace Regression (PR). We also investigated the performance of advanced, ensemble or meta methods that combine several basic classifiers, such as bagging, boosting and stacking.

Regression Trees (RT): To create regression tree models we used Weka's representations tree method, a fast decision/regression tree learner. The method builds a decision/regression tree by means of information gain/variance and prunes it using reduced-error pruning (with back-fitting). It also only sorts values for numeric attributes once. Missing values are dealt with by splitting the corresponding instances into pieces (as in C4.5 method Quinlan (1993)). The method was one of the fastest in building a model of the data; on average, it took 0.17-0.24 seconds to complete the task. The resulting regression trees were very large, having around 2000-4000 nodes.

Support Vector Machines (SVM): When used for classification, the SVM algorithm creates a hyperplane that separates the data into two classes with the maximum-margin. Given positive and negative training examples, a maximum-margin hyperplane is identified which splits the two categories of training examples, such that the distance between the hyperplane and the closest examples (the margin) is maximized.

For non-linear classifiers the classification is accomplished by applying the "kernel trick" where every dot product is replaced by a non-linear kernel function. This allows the algorithm to fit the maximum-margin hyperplane in the transformed feature space. If the kernel used is a radial basis function, the corresponding feature space is a Hilbert space of infinite dimension.

Maximum margin classifiers are well regularized, so the infinite dimension does not ruin the results. Some common kernels include a version of a SVM used for regression was proposed in late 1990's and is called Support Vector Regression (SVR) (Schölkopf, Smola, Burges & Soentpiet 1999).

The model produced by support vector classification (as described above) only depends on a subset of the training data, since the cost function for building the model does not take into consideration the training points that lie beyond the margin. The specific SVM implementation we used is the sequential minimal optimization algorithm proposed by Smola & Schölkopf (2004) for training a support vector regression model Smola, Scholkopf & Muller (1998). This implementation globally replaces all missing values and transforms nominal attributes into binary ones. It also normalizes all attributes by default (hence the output coefficients are based on the normalized/standardized data, not the original data). This implementation of the SVM allows control over whether feature-space normalization is performed (only available in the case of non-linear polynomial kernels). Moreover, it allows the use of an RBF kernel or a polynomial one. The SVM modeling time was by far the longest, taking anywhere from 310 to 880 seconds on a Pc with 2GHz speed and 2 Gb ram memory. The resulting model is a linear combination of the kernel function values, which is not very easily readable and does not fit our reverse engineering definition.

Model Trees (MT): Model trees produces a tree with linear models in the leaves (instead of a numeric value such as in regression trees). The modeling process was short, taking 35-77 seconds to complete on a Pc with 2GHz speed and 2 Gb ram memory. These models were also smaller in size as compared to regression trees, on average producing a tree with around 200 nodes (i.e. 200 linear models).

Pace Regression (PR): Pace regression (Wang 2000) is a method for fitting linear models in high dimensional spaces. Under regularity conditions, pace regression is provably optimal when the number of coefficients tends to infinity (Wang & Witten 2002). It consists of a group of estimators that are either overall optimal or optimal under certain conditions. Pace only took approximately 0.05-0.07 seconds to complete the modeling task. The model is fairly simple, but produces not as accurate out-of-sample forecasts as other models.

Advanced/Meta Methods: The basic idea of meta learning schemes it to build different "experts" and let them vote. The advantage of such a model is that it often improves predictive performance. The disadvantage is that it produces output that is very hard to analyze. Some of the well-known meta schemes include bagging, boosting, and stacking.

Bagging (Breiman 1996*a*) employs the simplest way of combining predictors: by voting or averaging, which means that each model receives an equal weight. The term “bagging” comes from “bootstrap aggregating”. Bagging is performed by creating N training sets from the original data set with N observations, by sampling with replacement. Then, a classifier is built for each training set, and, finally, the output from all of the classifiers is combined to produce an averaged output. Bagging reduces variance by voting/averaging and usually reduces the overall expected error.

Boosting (Kearns 1988, Schapire 1990, Freund & Schapire 1996) also uses voting/averaging, but the models are weighted according to their performance. It is an iterative procedure in the sense that the new models are influenced by performance of previously built ones. The new model is “encouraged” to become an expert for the instances incorrectly classified by earlier models. The intuitive justification of the method is that the models should be “experts” that complement each other. Boosting often produces classifiers that are significantly more accurate on novel/unseen data than bagging. Nevertheless, sometimes it fails in practical situations for the reason that the combined classifier over-fits the data.

Stacking (Wolpert 1992) is used to combine forecasts produced by different models. The basic idea behind the algorithms is that instead of evaluating several methods and selecting one, it is better to combine them. To combine the classifiers, stacking uses a meta learner instead of voting. The goal of the meta learner is to learn which classifiers are the reliable ones. In other words, the input into the meta learner is the output of the individual basic classifiers. Weka’s implementation of combining multiple methods is called the “multi scheme”. It selects a classifier from among several using cross validation on the training data or the performance on the training data. Performance is measured based on percent correct (classification) or mean-squared error (regression). We tested all three of the composite methods, and found that the bagging of our best-performing basic models, as well as the multi scheme with the top three basic classifiers delivered the best performance.

MT Bagging: Bagging of the model tree resulted in a model similar to the basic model tree. It consists of 217 nodes. It differs from the basic method model starting at the node level 2, but it does produce a very similar tree with just slightly different linear models in the leaves. It took anywhere from 330 to 593 seconds to build the bagged model.

Regression Tree Bagging: Bagged regression trees model gave a fairly different output. Compared to the basic model's 4589-node tree, it gave a shorter, 3889-node regression tree at the output. The performance improvement was most likely due to the basic tree's overfitting tendencies. By virtue of being shorter, the bagged tree most likely tends to overfit less. It took on average 2-3 seconds to build the composite model.

ANN Bagging: Bagged ANN created a different neural network model in 530-600 seconds of training time on a 2GHz with 2GB ram Pc. The out-of-sample forecasting performance was improved. The models are difficult to compare as both are "black-box" models. The training time was the longest of all the composite methods, almost as long as the SVM model training time.

Multi Scheme: We used the MT, RT and ANN models as the input to the multi scheme. Multi scheme took on average 96-327 seconds to complete the task on a 2GHz with 2GB ram Pc. The method selected the same basic regression tree model as the greatest performer (see table 6.7).

Overall, the model tree and regression tree methods gave the best non-MineTool, basic method results, with the out-of-sample RMSE of 0.0074 and 0.0098 respectively. The model tree, having given a smaller output evaluated slightly better, most likely due to offering a more general output model that tends to overfit less than the more detailed regression tree. Bagging of the top three basic methods (MT, RT and ANN) definitely increased their performance, especially in the case of ANNs-RMSE decreased from 0.0110 to 0.0068.

The best overall performers are bagged model trees and bagged regression trees, evaluating at the RMSE of 0.0060. These two models are similar in how they choose the attributes to split the search space and best distinguish among the split instances, and therefore, when bagged, gave identical composite results.

Chapter 7

Conclusions and directions for future research

7.1 Conclusions

The literature on prediction and model selection is highly interdisciplinary and interest in this field is fast growing. The review presented in this dissertation in by no means exhaustive. Automatic modeling and data-mining methods may help in many real-world problems where time constraints may have an important impact on the development of predictive models. However the automation of algorithms does not remove the need for human direction of data mining, and “data snooping” (White 2000) should be avoided.

We propose automated procedures that focus on the issue at hand: out-of-sample forecasting. In practical terms we try to find a balance between the following principles:

- Flexibility
- Parsimony
- Reverse engineering ability
- Computational speed

We follow a coherent strategy that balances all these aspects. We are inspired by a Specific to General philosophy, going from the “simple” to the “sophisticatedly simple” avoiding unnecessary complexity.

We assume that statistical and econometric models can be regarded as convenient approximations of an unknown data generation process. Because of this assumption, any model is inherently misspecified. Nonetheless, since we are interested in prediction, automated methods are still useful for our purposes. We propose automatic model building and selection procedures that behave well under the miss-specification hypothesis and discuss solutions to problems that one typically faces in applied research.

If prediction is the goal, and miss-specification is assumed, *asymptotic loss efficiency*, rather than *consistency*, is a desirable property of any model selection procedure. Among others, AIC, AICC, Mallows C and GCV benefit of this property, while BIC does not, since it typically selects under-parameterized models that have a high bias and poor out-of-sample performance assessed by the RMSE.

Another aspect of the prediction problem is the choice of an approximating function. Many approximating methods have been developed since computing power has become available. In the third chapter we reviewed briefly some of them with especial emphasis on different types of basis functions used in the Artificial Neural Network Literature, focusing especially on radial basis and ridgelets. Since non-linear optimization methods suffer the fact that the likelihood function may have many local minima, we consider approximations that non-linearities only in the inputs. A convenient class of such functions useful to build libraries of approximants are the Generically Comprehensive Revealing functions. Libraries of different basis may be useful where, as in economic data occurs, the smoothness degree of the target variable is usually unknown. Nonetheless other practical issues arise in empirical building of such flexible functional forms. These are related to the choice of the inputs and the selection of the approximation bases.

Throughout this dissertation, we referred many times to a subset selection tool useful for forecasting purposes called RETINA (Pérez-Amaral et al. 2003). The method implements an automated strategy for specification search and out-of-sample model validation and testing. We reviewed in detail its main characteristics and presented a new implementation for real data sets called RETINA Winpack. This is a standalone software, fully documented, with features which are relevant in applied research:

1. It is designed for immediate use by non-specialist applied researchers.

2. It reads data in the Excel format, allowing fast and easy data input.
3. It has a simple extremely user-friendly Graphical User Interface of just one window frame.
4. It handles extreme observations in both response and predictors set using the (Peña & Yohai 1999) method.
5. It allows for distinctive treatment of categorical predictors prior to input transformations. This feature allows to build flexible functional forms that include specific constants and specific slopes like an analysis of covariance.
6. It delivers an informative output by summarizing out-of-sample predictive statistics of proposed specifications and allowing the user to easily compare among them.

We also fill a gap present in the literature on comparing RETINA with other methods (Pérez-Amaral et al. 2003, Castle 2005, Pérez-Amaral et al. 2005) since we explicitly assess its validity as an automatic modeling tool focusing exclusively on the out-of-sample forecasting ability against a variety of methods (Stepwise regression, Non-negative Garrote, LARS, LASSO, Ridge and the General to Specific methodology). A striking fact which emerges from the experiments is that a similar behavior results in terms of forecasting ability, although RETINA seems to be specially well suited for cases where the ratio N/k is large. Tests for forecast equality show that RETINA does equally well or better than other methods, under different settings in which sample sizes (even small, eg. 50 observations), the number of candidate predictors, and the nature of the data (time series or cross-section) vary systematically across experiments.

We applied RETINA Winpack to an empirical case. We use it to forecast business telecommunications demands for local, intra-LATA and inter-LATA services using US Telecommunications data. We obtained specifications that proved to be useful for out-of-sample prediction. RETINA generates an expanded input set using the firm group membership as a heterogeneity parameter to estimate specific constants and specific slopes. All suggested specifications include interactions and nonlinear transformations of the original predictors. As a result, out-of-sample forecasting ability significantly improves over alternative formulations. We also find that telephone equipment variables are almost always selected as relevant first order effects. Semi-parametric demand elasticities, evaluated for the relevant variables at

Table 7.1: The main contributions of this dissertation.

1. RETINA Winpack: An user friendly software for real data sets which incorporates: outlier detection, normalization of variables, Graphical User Interface, user guide, treatment of categorical variables, informative output.
2. Comparison between RETINA and other subset regression methods: Stepwise method, The LASSO, Non-negative Garrote, Ridge Regression, Gets modelling, LARS. RETINA Winpack show to be as valid as all these methods, based on out-of-sample forecasting ability.
3. Empirical Application to Telecommunications demand using firm-level data:
    - We show the potential of RETINA Winpack in finding suitable approximations that behave well out-of-sample in comparison with alternative linear baseline models.
    - The reverse engineering capabilities of RETINA account for possible substitution patterns among telephone equipments.
4. RETINET:
    - Generalizes RETINA.
    - Automatizes the process of building flexible functional forms from simple linear specifications to highly non-linear in the inputs specifications, using a Specific To General methodology.
    - Alternative methodology or flexible model building to non-linearly estimated ANN.

the average values, suggest substitution patterns between different types of telephone equipment.

Finally we presented a new automated modeling tool called RETINET, which provides a heuristic method to build flexible functional forms by using libraries of highly non-linear transformations of the original predictors. The procedure avoids technical difficulties related to non-linear estimation procedures to compute the weights of the ANN transformations. As an advantage over traditional ANN empirical building strategies, the method also provides guidance about 1) the selection of inputs X_t 2) the selection of the hidden nodes of the network. Being the specified equations always linear in the parameters, OLS estimation techniques can be used, and computations, even if intensive, can be performed reasonably fast. The algorithm presented here differs from White's (2006) Quicknet in that we use RETINA to select the inputs and to control for over-fitting and model evaluation method. Finally the procedure chooses among three different types of squashing functions, namely Logistic, Radial Basis and Ridgelets transformation. These provide a sufficiently rich library of approximants and more parsimonious models can be obtained. Based on the data, the algorithm chooses which basis function to adopt in any given instance, and parsimonious approximations are usually delivered. These are important advantages compared to networks estimated by non-linear methods. Another advantage of RETINET is that suggested specifications retain, to some extent, analytical interpretability and allow reverse engineering. This facilitates easier dissemination of the model as well as exploration of the effects of various terms. Based on two different simulation examples the method provides favorable evidence with respect to the out-of-sample forecasting ability provided by other simpler or more sophisticated methods. Even though these results cannot be considered as a definitive proof of the superiority of the method, we can state that it doesn't perform worse while offering the relevant advantages commented above. More experimentation is needed under controlled settings using Monte Carlo methods.

7.2 Future Research directions

We are currently applying RETINET-Minetool in a number of projects using financial and geophysical spacecraft data. These include the analysis and prediction of foreign exchange rate, stock market returns and other macro-economic variables forecast such as inflation. In the area of geophysical sciences there is a special interest in using these techniques in the development of a 3D model of magnetopause, identification of flux transfer events and traveling compression regions, among others.

In its earliest version, the RETINET algorithm has been written by the author in Gauss language. After its successful application to geophysics data, Sciberquest Inc. a scientific consultancy agency based in Solana Beach, California decided to include it in a data-mining package called Mine-tool. Currently the software is being re-written in C language and soon a public licensed version of it should be available to the scientific community. These represent important directions in our research agenda, as well as further developments in order to extend the method to:

- Window roll estimation and multiple step ahead based forecasts.

- Multiple output forecasts.
- Include a richer variety of highly non-linear libraries.
- Include richer non-linear dynamic structure detection ability by means of recurrent networks (like the Elman (1990) network. network) or stochastic volatility models.
- Provide user-friendly interfaces and user documentation.
- Improve the computational efficiency.
- Extend the method to panel data

These are just a few of the many possibilities that actually exist to improve the algorithm. Nonetheless our main hope, and at the same time our main concern, is that the method will generate future interest and will be of some usefulness in the research community especially where real-time predictions, as in financial markets, are of big interest to analysts and practioneers. In this context we agree with the point of view of McAleer (2005) who points out:

> *An automated method of inference that is never used has zero value.*

Bibliography

Akaike, H. (1973), Information theory and an extension of the likelihood principle, *in* B. Petrov & F. Csaki, eds, 'Proceedings of the Second International Symposium on Information Theory', Akademiai Kiado, Budapest, pp. 267–281.

Arrow, K., Chenery, H., Minhas, B. & Solow, R. (1961), 'Capital-Labor Substitution and Economic Efficiency', *The Review of Economics and Statistics* **43**(3), 225–250.

Artle & Averous (1973), 'The Telephone System as a Public Good', *Bell Journal of Economics and Management Science* **4**, 89–100.

Barron, A. (1993), 'Universal approximation bounds for superpositions of a sigmoidal function', *Information Theory, IEEE Transactions on* **39**(3), 930–945.

Baxt, W. & White, H. (1995), 'Bootstrapping Confidence Intervals for Clinical Input Variable Effects in a Network Trained to Identify the Presence of Acute Myocardial Infarction', *Neural Computation* **7**(3), 624–638.

Bellman, R. (1961), *Adaptive control processes: a guided tour*, Princeton University Press.

Ben-Akiva, M. & Gershenfeld, S. (1989), *Analysis of Business Establishment Choice of Telephone Systems*, Cambridge Systematics, Cambridge , MA.

Berthold, M. & Hand, D. (2003), *Intelligent Data Analysis: an introduction*, Springer.

Bierens, H. (1990), 'A consistent Conditional Moment Test of Functional Forms', *Econometrica* **1**(58), 1443–1458.

Binner, J. M., Elger, C. T., Nilsson, B. & Tepper, J. (2006), 'Predictable nonlinearities in U.S. inflation', *Economics Letters* **93**(3), 323–328.

Bollerslev, T. (1986), 'Generalized autoregressive conditional heteroskedasticity', *Journal of Econometrics* **31**(3), 307–327.

Box, G. (1979), Robustness in the strategy of scientific model building, *in* R. Launer & G. Wilkinson, eds, 'Robustness in Statistics', Academic Press, New York.

Breiman, L. (1992), 'The Little Bootstrap and Other Methods for Dimensionality Selection in Regression: X-Fixed Prediction Error', *Journal of the American Statistical Association* **87**(419), 738–754.

Breiman, L. (1995), 'Better Subset Regression using the Nonnegative Garrote', *Technometrics* **37**(4), 373–384.

Breiman, L. (1996*a*), 'Bagging Predictors', *Machine Learning* **24**(2), 123–140.

Breiman, L. (1996*b*), 'Heuristics of instability and stabilization in model selection', *Annals of Statistics* **24**(6), 2350–2383.

Breiman, L., Friedman, J., Olshen, R. & Stone, C. (1984), 'Classification and Regression Trees', *Wadsworth, Belmont, CA* pp. 75–80.

Brooks, C. (1997), 'Linear and Non-linear(Non-) Forecastability of High-frequency Exchange Rates', *Journal of Forecasting* **16**(2), 125–145.

Brownlees, C. (2005), RETINA for Matlab, Technical report, Dipartimento di Statistica - Università di Firenze, Italy.

Brusco, M. & Stahl, S. (2005), *Branch-and-Bound Applications in Combinatorial Data Analysis*, Springer.

Burman, P., Chow, E. & Nolan, D. (1994), 'A cross-validatory method for dependent data', *Biometrika* **81**(2), 351.

Burnham, K. & Anderson, D. (2002), *Model Selection and Multi-Model Inference. A Practical Information-Theoretic Approach*, 2nd edn, Springer-Verlag, New York.

Campos, J., Ericsson, N. & Hendry, D. (2005), *General to Specific Modeling*, Edward Elgar.

Candès, E. (1998), Ridgelets: Theory and Applications, PhD thesis, Stanford University.

Candès, E. (1999), 'Ridgelets: a key to higher-dimensional intermittency?', *Philosophical Transactions: Mathematical, Physical and Engineering Sciences* **357**(1760), 2495–2509.

Candès, E. (2003), 'Ridgelets: estimating with ridge functions', *Ann. Statist* **31**(5), 1561–1599.

Castle, J. (2005), 'Evaluating PcGets and RETINA as Automatic Model Selection Algorithms', *Oxford Bulletin of Economics and Statistics* **67**, 837.

Chen, X., Racine, J. & Swanson, N. (2001), 'Semiparametric ARX neural-network models with an application toforecasting inflation', *Neural Networks, IEEE Transactions on* **12**(4), 674–683.

Cheng, B. & Titterington, D. (1994), 'Neural Networks: A Review from a Statistical Perspective', *Statistical Science* **9**(1), 2–30.

Cobb, C. & Douglas, P. (1928), 'A Theory of Production', *The American Economic Review* **18**(1), 139–165.

Craven, P. & Wahba, G. (1979), 'Smoothing noisy data with spline functions: estimating the correct degree of smoothing by the method of generalized cross-validation', *Numer. Math* **31**(4), 377–403.

Cybenko, G. (1989), 'Approximation by superpositions of a sigmoidal function', *Mathematics of Control, Signals, and Systems (MCSS)* **2**(4), 303–314.

Dempster, A. (1969), *Elements of continuous multivariate analysis*, Addison-Wesley Reading, Mass.

Diebold, F. & Mariano, R. (1995), 'Comparing Predictive Accuracy', *Journal of Business & Economic Statistics* **13**(3), 253–263.

Dmitriev, A. & Suvorova, A. (2000), 'Three-dimensional artificial neural network model of the dayside magnetopause', *Journal of Geophysical Research* **105**(A8), 18909–18918.

Draper, N. & Smith, H. (1966), 'Applied Regression', *Analysis* .

Efron, B. (1983), 'Estimating the error rate of a prediction rule: improvement on cross-validation', *Journal of the American Statistical Association* **78**(382), 316–331.

Efron, B. (1986), ‘How biased is the apparent error rate of a prediction rule’, *Journal of the American Statistical Association* **81**(394), 461–470.

Efron, B., Hastie, T., Johnstone, I. & Tibshirani, R. (2004), ‘Least angle regression’, *Annals of Statistics* **32**(2), 407–499.

Efron, B. & Tibshirani, R. (1993), *An Introduction to the Bootstrap*, London: Chapman & Hall.

Elman, J. (1990), ‘Finding Structure in Time’, *Cognitive Science* **14**(2), 179–211.

Engle, R. (1982), ‘Autoregressive conditional heteroskedasticity with estimates of the variance of UK ination’, *Econometrica* **50**, 987–1008.

Farlow, S. (1984), *Self-Organizing Methods in Modeling: Gmdh Type Algorithms*, Marcel Dekker.

Fine, T. (1999), *Feedforward Neural Network Methodology*, Springer.

Fletcher, G. (1996), ‘Adaptive internal activation functions and their effect on learning in feed forward networks’, *Neural Processing Letters* **4**(1), 29–38.

Frank, E., Hall, M., Trigg, L., Holmes, G. & Witten, I. (2004), ‘Data mining in bioinformatics using Weka’, *Bioinformatics* **20**(15), 2479–2481.

Franses, P. & van Dijk, D. (2000), *Nonlinear time series models in empirical finance*, Cambridge University Press New York.

Freund, Y. & Schapire, R. (1996), ‘Experiments with a new boosting algorithm’, *Machine Learning: Proceedings of the Thirteenth International Conference* **148**, 156.

Gilbert, C. (1989), ‘LSE and the British Approach to Time Series Econometrics’, *Oxford Economic Papers* **41**(1), 108–128.

Golub, G. H., Heath, M. & Wahba, G. (1979), ‘Generalized Cross-Validation as a Method for Choosing a Good Ridge Parameter’, *Technometrics* **21**, 215–223.

Goutte, C. (1997), ‘Note on free lunches and cross-validation’, *Neural Computation* **9**(6), 1245–1249.

Granger, C. & Andersen, A. (1978), ‘An Introduction to Bilinear Time Series’, *Gottingen: Vandenhock and Ruprecht* .

Haggan, V. & Ozaki, T. (1981), 'Modelling nonlinear random vibrations using an amplitude-dependent autoregressive time series model', *Biometrika* **68**(1), 189.

Harvey, D., Leybourne, S. & Newbold, P. (1997), 'Testing the equality of prediction mean squared errors', *International Journal of Forecasting* **13**(2), 281–291.

Hastie, T., Tibshirani, R. & Friedman, J. (2001), *The elements of statistical learning: data mining, inference, and prediction*, Springer.

Hendry, D. & Krolzig, H. (2001), *Automatic Econometric Model Selection with PcGets.*, Timberlake Consultants Press, London.

Hendry, D. & Krolzig, H. (2005), 'The Properties of Automatic Gets Modelling', *Economic Journal* **115**(502), 32–61.

Hill, T., O'Connor, M. & Remus, W. (1996), 'Neural Network Models for Time Series Forecasts', *Management Science* **42**(7), 1082–1092.

Hjorth, J. (1994), *Computer Intensive Statistical Methods: Validation, Model Selection, and Bootstrap*, London: Chapman & Hall.

Hoerl, A. & Kennard, R. (1970), 'Ridge Regression: Biased Estimation for Nonorthogonal Problems', *Technometrics* **12**(1), 55–67.

Hoover, K. D. & Perez, S. J. (1999), 'Data mining reconsidered: encompassing and the general-to-specific approach to specification search', *Econometrics Journal* **2**, 167–191.

Hornik, K., Stinchcombe, M. & White, H. (1989), 'Multilayer feedforward networks are universal approximators', *Neural Networks* **2**(5), 359–366.

Jankovicová, D., Dolinskỳ, P., Valach, F. & Vörös, Z. (2002), 'Neural network-based nonlinear prediction of magnetic storms', *Journal of Atmospheric and Solar-Terrestrial Physics* **64**(5-6), 651–656.

Jhee, W. & Lee, J. (1993), 'Performance of neural networks in managerial forecasting', *International Journal of Intelligent Systems in Accounting and Finance Management* **2**(1), 55–71.

Jones, L. (1997), 'The computational intractability of training Sigmoidal Neural Networks', *IEEE Transactions on Information Theory* **43**, 167–173.

Karimabadi, H., Sipes, T., White, H., Marinucci, M., Dmitriev, A., Chao, J., Driscoll, J. & Balac, N. (2007), 'Data mining in space physics: MineTool algorithm', *Journal of Geophysical Research* **112**(A11).

Kearns, M. (1988), 'Thoughts on hypothesis boosting', *Unpublished manuscript, December* .

Kearns, M. (1997), 'A bound on the error of cross validation using the approximation and estimation rates, with consequences for the training-test split', *Neural Computation* **9**(5), 1143–1161.

Kodogiannis, V. & Lolis, A. (2002), 'Forecasting Financial Time Series using Neural Network and Fuzzy System-based Techniques', *Neural Computing & Applications* **11**(2), 90–102.

Kohavi, R. (1995), A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection, *in* 'Proceedings of the 14th International Joint Conference on Artificial Intelligence', Vol. 2, IJCAI, Montreal, Quebec, Canada, pp. 1137–1145.

Kridel, D.J. Rappoport, P. & Taylor, L. (1997), 'IntraLATA Long-Distance Demand: Carrier Choice, Usage Demand and Price Elasticities', *International Communications Forecasting Conference* .

Kuan, C. & Liu, T. (1995), 'Forecasting Exchange Rates Using Feedforward and Recurrent Neural Networks', *Journal of Applied Econometrics* **10**(4), 347–364.

Kuan, C.-M. & White, H. (1994), 'Artificial Neural Networks: An Econometric Perspective', *Econometric Reviews* **13**(1-92).

Leeb, H. & Pötscher, B. (2005), 'Model Selection and Inference: Facts and Fiction', *Econometric Theory* **21**(01), 21–59.

Leisch, F. (2003), FlexMix: A general framework for finite mixture models and latent class regression in R, Technical Report 86, Adaptive Information Systems and Modelling in Economics and Management Science. SFB.

Lendasse, A., Lee, J., de Bodt, E., Wertz, V. & Verleysen, M. (2002), 'Approximation by radial-basis function networksapplication to option pricing', *Proceedings of the ACSEG* pp. 201–212.

Levy, A. (1998), Semi-parametric Estimation of Demand for Intra-LATA Telecommunications, Master's thesis, Department of Economics, North Carolina State University, Raleigh, NC 27695-8110.

Li, K. C. (1987), 'Asymptotic Optimality for C_p, C_l Cross-Validation and Generalized Cross-Validation: Discrete Index Set', *The Annals of Statistics* **15**(3), 958–975.

Longley, J. (1967), 'An Appraisal of Least Squares Programs for the Electronic Computer from the Point of View of the User', *Journal of the American Statistical Association* **62**(319), 819–841.

Looney, C. (1997), *Pattern recognition using neural networks: theory and algorithms for engineers and scientists*, Oxford University Press, Inc. New York, NY, USA.

Lovell, M. (1983), 'Data Mining', *The Review of Economics and Statistics* **65**(1), 1–12.

Lundstedt, H. (1992), 'Neural networks and predictions of solar-terrestrial effects', *Planetary and Space Science* **40**(4), 457–464.

MacQueen, J. (1967), Some Methods for Classification and Analysis of Multivariate Observations, Proc. Symp. Math. Statist. And Probability, 5th, Univ. of California Press, Berkeley, CA, pp. 281–297. AD 669871.

Mallows, C. L. (1973), 'Some comments on C_P', *Technometrics* **15**, 661–675.

Marinucci, M. (2005), *RETINA Winpack for Real Data: A quick Guide for Automatic Model Selection*, Universidad Complutense de Madrid, Somosaguas, Madrid, Spain.

Masters, T. (1995), *Advanced Algorithms for Neural Networks: A C++ Sourcebook*, John Wiley & Sons Inc.

McAleer, M. (2005), 'Automated Inference and Learning in Modeling Financial Volatility', *Econometric Theory* **21**(01), 232–261.

McLachlan, G. & Peel, D. (2000), *Finite Mixture Models*, Wiley Series in Probability & Statistics, John Wiley & Sons, New York.

McNelis, P. (2005), *Neural Networks in Finance: Gaining Predictive Edge in the Market*, Academic Press.

McQuarrie, A. & Tsai, C. (1998), *Regression and time series model selection*, World Scientific River Edge, NJ.

Medeiros, M., Teräsvirta, T. & Rech, G. (2006), 'Building Neural Network Models for Time Series: A Statistical Approach', *Journal of Forecasting* **25**, 49–75.

Miller, A. (2002), *Subset Selection in Regression*, Monographs on Statistics and Applied Probability, 2 edn, Chapman & Hall/CRC.

Mizon, G. (1995), 'Progressive Modelling of Macroeconomic Time Series: The LSE Methodology'.

Nakamura, E. (2005), 'Inflation forecasting using a neural network', *Economics Letters* **86**(373-378).

Nishii, R. (1984), 'Asymptotic Properties of Criteria for Selection of Variables in Multiple Regression', *The Annals of Statistics* **12**(2), 758–765.

O'Brien, T. & McPherron, R. (2003), 'An empirical dynamic equation for energetic electrons at geosynchronous orbit', *Journal of Geophysical Research (Space Physics* **108**(A3).

Pagan, A. (1987), 'Three econometric methodologies: A critical appraisal', *Journal of Economic Surveys* **1**(1), 3–24.

Peña, D. & Yohai, V. (1999), 'A Fast Procedure for Outlier Diagnostic in Large Regression Problems', *Journal of the American Statistical Association* **94**, 434–445.

Pérez-Amaral, T., Gallo, G. & White, H. (2003), 'A Flexible Tool for Model Building: the Relevant Transformation of the Inputs Network Approach (RETINA)', *Oxford Bulletin of Economics and Statistics* **65**, 821–838.

Pérez-Amaral, T., Gallo, G. & White, H. (2005), 'A comparison of Complementary Automatic Modeling Methods: RETINA and PCGets', *Econometric Theory* **21**, 262–277.

Pérez-Amaral, T. & Marinucci, M. (2002), Econometric Modelling of Business Telephone Toll Demand for Individual Firms using a new model selection approach, RETINA. 13th Regional Conference of the International Telecommunications Society-Madrid.

Pesaran, M. & Timmermann, A. (1992), 'A Simple Nonparametric Test of Predictive Performance', *Journal of Business & Economic Statistics* **10**(4), 461–465.

Phillips, P. & Perron, P. (1988), 'Testing for a unit root in time series regression', *Biometrika* **75**(2), 335.

Plutowski, M. & White, H. (1993), 'Selecting Concise Traning Sets from Clean Data', **4**(2), 305–318.

Powell, M. (1987), 'Radial basis functions for multivariable interpolation: a review', *Clarendon Press Institute Of Mathematics And Its Applications Conference Series* pp. 143–167.

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

R Development Core Team (2007), *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0.

Racine, J. (1997), 'Feasible cross-validatory model selection for general stationary processes', *Journal of Applied Econometrics* **12**(382), 169–179.

Racine, J. (2000), 'A Consistent Cross-Validatory Method For Dependent Data: hv-Block Cross-Validation', *Journal of Econometrics* (99), 39–61.

Raftery, A. (1995), 'Bayesian model selection in social research', *Sociological Methodology* . with discussion by Andrew Gelman, Donald B. Rubin, and Robert M. Hauser, and a rejoinder.

Ramsey, J. (1969), 'Tests for Specification Errors in Classical Linear Least Square Regression Analysis', *Journal of the Royal Statistical Society* **B**(31), 350–371.

Ripley, B. (1996), *Pattern Recognition and Neural Networks*, Cambridge University Press.

Rissanen, J. (1978), 'Modelling by shortest data description', *Automatica* **14**, 465–471.

Roh, T. (2007), 'Forecasting the volatility of stock price index', *Expert Systems with Applications* **33**(4), 916–922.

Rohlfs, J. (1974), 'A Theory of Interdependent Demand for a Communications Service', *Bell Journal of Economics and Management Science* **5**, 16–37.

Schapire, R. (1990), 'The strength of weak learnability', *Machine Learning* **5**(2), 197–227.

Schölkopf, B., Smola, A., Burges, C. & Soentpiet, R. (1999), *Advances in Kernel Methods: support vector learning*, MIT Press.

Schwarz, G. (1978), 'Estimating the Dimension of a Model', *The Annals of Statistics* **6**(2), 461–464.

Shannon, C. & Weaver, W. (1963), *Mathematical Theory of Communication*, University of Illinois Press.

Shao, J. (1997), 'An asymptotic Theory for Linear Model Selection', *Statistica Sinica* **7**, 221–264.

Shao, J. & Tu, D. (1995), *The Jackknife and Bootstrap*, New York:Springer-Verlag.

Shue, J., Kokubun, S., Song, P., Russell, C., Steinberg, J., Chao, J., Zastenker, G., Vaisberg, O., Singer, H. & Detman, T. (1998), 'Magnetopause location under extreme solar wind conditions', *Journal of Geophysical Research* **103**(A8), 17691–17700.

Smola, A. & Schölkopf, B. (2004), 'A tutorial on support vector regression', *Statistics and Computing* **14**(3), 199–222.

Smola, A., Scholkopf, B. & Muller, K. (1998), 'The connection between regularization operators and support vector kernels', *Neural Networks* **11**(4), 637–649.

Solow, R. (1956), 'A Contribution to the Theory of Economic Growth', *The Quarterly Journal of Economics* **70**(1), 65–94.

Stinchcombe, M. & White, H. (2000), 'Consistent Specification Testing with Nuisance Parameters present only under the Alternative', *Econometric Theory* **14**(03), 295–325.

Stone, M. (1977), 'An asymptotic equivalence of choice of model by cross-validation and Akaike's criterion', **39**, 44–47.

Sugiura, N. (1978), 'Further analysis of the data by Akaikes information criterion and the finite corrections', *Communications in Statistics: Theory and Methods* **7**(1), 13–26.

Swanson, N. & White, H. (1997), 'A Model Selection Approach to Real-Time Macroeconomic Forecasting Using Linear Models and Artificial Neural Networks', *The Review of Economics and Statistics* **79**(4), 540–550.

Takeuchi, K. (1976), 'Distribution of informational statistics and a criterion of model fitting', *Suri-Kagaku (Mathematical Sciences)* **153**, 12–18.

Tang, Z. & Fishwich, P. (1993), 'Back-Propagation neural nets as models for time series forecasting', *ORSA Journal on Computing* **5**(4), 374–385.

Taylor, L. (1996), Competitive Own- and Cross-Price Elasticities in the Intralata Toll Market: Estimates from the Bill Harvesting ® II Database, Technical report, Department of Economics, University of Arizona, Tucson AZ.

Terasvirta, T. (1994), 'Specification, Estimation, and Evaluation of Smooth Transition Autoregressive Models', *Journal of the American Statistical Association* **89**(425), 208–218.

Tibshirani, R. (1996*a*), 'A Comparison of Some Error Estimates for Neural Network Models', *Neural Computation* **8**(1), 152–163.

Tibshirani, R. (1996*b*), 'Regression Shrinkage and Selection via the Lasso', *Journal of the Royal Statistical Society* **58**(1), 267–288.

Tikhonov, A. & Arsenin, V. (1977), *Solutions of Ill posed problems*, Winston.

Tong, H. (1978), 'On a threshold model', *Pattern Recognition and Signal Processing* pp. 101–141.

Tong, H. & Lim, K. (1980), 'Threshold Autoregression, Limit Cycles and Cyclical Data', *Journal of the Royal Statistical Society. Series B (Methodological)* **42**(3), 245–292.

Trippi, R. & Turban, E. (1992), *Neural Networks in Finance and Investing: Using Artificial Intelligence to Improve Real World Performance*, McGraw-Hill, Inc. New York, NY, USA.

Troyanskaya, O., Cantor, M., Sherlock, G., Brown, P., Hastie, T., Tibshirani, R., D., B. & Altman, R. B. (2001), 'Missing Value Estimation Methods for DNA Microarrays', *Bioinformatics* **17**(6), 520–525.

Tschernig, R. & Yang, L. (2000), 'Nonparametric Lag Selection for Time Series', *Journal of Time Series Analysis* **21**(4), 457–487.

Vanden Berghen, F. (2005), LARS Library: Least Angle Regression Stagewise Library, Technical report, IRIDIA-Universitè Libre de Bruxelles, Bruxelles.

Von Rabenau, B. & Stahl, K. (1974), 'Dynamic A of Public Goods: a further Analysis of the Telephone System', *Bell Journal of Economics and Management Science* **5**(2), 651–669.

Vu, V. (1998), 'On the infeasibility of training neural networks with small mean squared error', *IEEE Transactions on Information Theory* **44**, 2892–2900.

Wahba, G. (1990), *Spline models for observational data*, Society for Industrial and Applied Mathematics Philadelphia, Pa.

Wahba, G. & Wold, S. (1975), 'A completely automatic French curve: fitting spline functions by cross-validation', *Communications in Statistics* **4**(1), 1–17.

Wang, Y. (2000), A New Approach to Fitting Linear Models in High Dimensional Spaces, PhD thesis, The University of Waikato.

Wang, Y. & Witten, I. (2002), 'Modeling for Optimal Probability Prediction', *Proceedings of ICML* .

Ward Jr., J. (1963), 'Hierarchical Grouping to Optimize an Objective Function', *Journal of the American Statistical Association* **58**(301), 236–244.

Weiss, S. & Kulikowski, C. (1991), *Computer Systems that Learn*, Morgan Kaufmann.

White, H. (1984), *Asymptotic Theory for Econometricians*, Academic Press Orlando.

White, H. (1988), 'Economic prediction using neural networks: the case of IBM dailystock returns', *Neural Networks, 1988., IEEE International Conference on* pp. 451–458.

White, H. (1998), *Artificial Neural Networks and alternative methods for assessing naval readiness*, NRDA, San Diego.

White, H. (2000), 'A reality Check for Data Snooping', *Econometrica* **68**(5), 1097–1126.

White, H. (2006), Approximate Nonlinear Forecasting Methods, *in* G. Elliott, C. Granger & A. Timmermann, eds, 'Handbook of Economic Forecasting', Vol. I, Elsevier, chapter 9, pp. 459–512.

White, H. & Racine, J. (2001), 'Statistical inference, the bootstrap, and neural-network modeling with application to foreign exchange rates', *Neural Networks, IEEE Transactions on* **12**(4), 657–673.

Wolpert, D. (1992), 'Stacked generalization', *Neural Networks* **5**(2), 241–259.

Yao, Q. & Tong, H. (1994), 'On subset selection in non-parametric stochastic regression', *Statistica Sinica* **4**, 51–70.

Zhang, G., Patuwo, B. & Hu, M. (1998), 'Forecasting with artificial neural networks: The state of the art', *International Journal of Forecasting* **14**(1), 35–62.

Zhang, G., Patuwo, B. & Hu, M. (2001), 'A simulation study of artificial neural networks for nonlinear time-series forecasting', *Computers and Operations Research* **28**(4), 381–396.

Zhu, H. & Rohwer, R. (1996), 'No free lunch for cross-validation', *Neural Computation* **8**(7), 1421–1426.