LSE

THE LONDON SCHOOL
OF ECONOMICS AND
POLITICAL SCIENCE

# *Essays in High-dimensional Nonlinear Time Series Analysis*

A DISSERTATION PRESENTED
BY
ALI HABIBNIA

IN FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY
IN THE SUBJECT OF
STATISTICS

LONDON SCHOOL OF ECONOMICS AND POLITICAL SCIENCE
LONDON, UNITED KINGDOM
DECEMBER 2016

Thesis Advisors: Angelos Dassios, Matteo Barigozzi Ali Habibnia

# *Essays in High-dimensional Nonlinear Time Series Analysis*

## Abstract

In this thesis, I study high-dimensional nonlinear time series analysis, and its applications in financial forecasting and identifying risk in highly interconnected financial networks. The first chapter is devoted to the testing for nonlinearity in financial time series. I present a tentative classification of the various linearity tests that have been proposed in the literature. Then I investigate nonlinear features of real financial series to determine if the data justify the use of nonlinear techniques, such as those inspired by machine learning theories. In Chapter 3 & 5, I develop forecasting strategies with a high-dimensional panel of predictors while considering nonlinear dynamics. Combining these two elements is a developing area of research. In the third chapter, I propose a nonlinear generalization of the statistical factor models. As a first step, factor estimation, I employ an auto-associative neural network to estimate nonlinear factors from predictors. In the second step, forecasting equation, I apply a nonlinear function -feedforward neural network-on estimated factors for prediction. I show that these features can go beyond covariance analysis and enhance forecast accuracy. I apply this approach to forecast equity returns, and show that capturing nonlinear dynamics between equities significantly improves the quality of forecasts over current univariate and multivariate factor models. In Chapter 5, I propose a high-dimensional learning based on a shrinkage estimation of a backpropagation algorithm for skip-layer neural networks. This thesis emphasizes that linear models can be represented as special cases of these two aforementioned models, which basically means that if there is no nonlinearity between series, the proposed models will reduce to a linear model. This thesis also includes a chapter (chapter 4, with Negar Kiyavash and Seyedjalal Etesami), which in this chapter, we propose a new approach for identifying and measuring systemic risk in financial networks by introducing a nonlinearly modified Granger-causality network based on directed information graphs. The suggested method allows for nonlinearity and has predictive power over future economic activity through a time-varying network of interconnections. We apply the method to the daily returns of U.S. financial Institutions including banks, brokers and insurance companies to identify the level of systemic risk in the financial sector and the contribution of each financial institution.

# Contents

# List of Figures

# Declaration

I certify that the thesis I have presented for examination for the PhD degree of the London School of Economics and Political Science is solely my own work other than where I have clearly indicated that it is the work of others (in which case the extent of any work carried out jointly by me and any other person is clearly identified in it).

I warrant that this authorisation does not, to the best of my belief, infringe the rights of any third party.

**Statement of conjoint work**

I confirm that chapter 4 is jointly co-authored with Negar Kiyavash and Seyedjalal Etesami, and I contributed 50 % of this work.

# Acknowledgments

My deepest gratitude goes to my advisors, Matteo Barigozzi and Angelos Dassios, for their generous, invaluable support, guidance, patience, and trust. It has been an honor to be Matteo's first Ph.D. student. I thank him for introducing me to the wonders and frustrations of scientific research. He has taught me, both consciously and unconsciously, how real research is done. I appreciate all his contributions of time and ideas to make my Ph.D. experience productive and stimulating.

I am very grateful to fellow researchers. Being a 'child' of both Department of Statistics and Systemic Risk Centre, I was particularly lucky to be surrounded by so many talented, insightful and dedicated people. My time at LSE was made enjoyable in large part due to the many friends and groups that became a part of my life. I gratefully acknowledge the funding sources that made my Ph.D. work possible. I was funded by LSE Systemic Risk Centre and the Economic and Social Research Council (ESRC). My work was also supported by the Department of Statistics. Last but not least, I am truly thankful to my examiners, Qiwei Yao and Esfandiar Maasoumi, for taking the time to engage with my thesis.

I would like to dedicate my thesis to my beloved mother, Fereshteh, who was the source of everlasting love and support over the years and raised me with a love of science. To my father, Nader, who is my biggest mentor in life. To my cherished sister, Fariba, who has been my best friend, my soul mate and the best part of me. To my uncle, Reza, who is not my uncle only, but he has always been a good friend. To my late aunts, Houri and Jahandokht, who played a very important role in creating who I am today. To my sweet nephew, Sina, who recently joined our family.

*Is it the fault of wine if a fool drinks it and goes stumbling into darkness? - Importance of statistical model selection*

Ibn Sīnā (Avicenna)

# 1
# Introduction

World data currently doubles every couple of years with an on-going steady increase in computing power that poses new challenges for economic modelling and forecasting in a big data environment. It challenges state-of-the-art data acquisition, computation and analysis methods. To benefit from many new potential explanatory variables, feature extraction methods (i.e, Principal component analysis - Pearson, 1901; Eckart and Young, 1936; Factor models - Stock & Watson, 2002, 2006; Bai & Ng, 2002; Deistler & Hamann, 2005; Forni et al, 2005; Lam & Yao, 2012), shrinkage techniques (Ridge - Hoerl & Kennard, 1988; LASSO - Tibshirani, 1996; Elastic Net - Zou & Hastie, 2005), and subset selection techniques (Bayesian regression - De Mol, Giannone, Reichlin, 2008; Selecting variables - Bai & Ng, 2008a) are used to handle high-dimensional data but they are mostly linear models.

However linear models are adequate to explain many phenomena in the world, most important economic and financial phenomena are complex and nonlinear in nature. In order to explain nonlinear phenomena, different parametric and nonparametric nonlinear regression models have been developed so far (see Fan & Yao, 2005; Teräsvirta, Tjøstheim & Granger, 2010).

Parametric nonlinear regression models attempt to characterise the relationship between predictors and response with parametric nonlinear functions. The parameters can take the form of a polynomial, exponential, trigonometric, power, or any other nonlinear function. In other words, in parametric nonlinear models the shape of the functional relationships between the response and the predictors are predetermined. In many situations, that relationship is unknown and nonparametric nonlinear regression models should be used. In nonparametric models, the shape of the functional relationships between variables can be adjusted to capture unusual or unexpected features of the data. The main types of nonparametric regression models are kernel-based methods, tree-based regression models and artificial neural networks.

Kernel-based methods can be viewed as a nonlinear mapping from inputs into higher dimensional feature space in the hope that the data will be linearly separable or better structured. It measures distances between observations, then predicts new values based on these distances. Best known example are support vector machines (SVMs), introduced by Vapnik (Chervonenkis and Vapnik, 1964, 1974; Vapnik, 1982, 1995), which provide a structured way to use a linear algorithm in a transformed feature space. The key advantage this so-called kernel trick brings is that nonlinear patterns can be found at a reasonable computational cost. Perhaps the biggest limitation of the kernel-based methods lies in choice of the kernel and tuning model parameters.

Tree-based regression models are alternative (nonparametric and nonlinear) approaches to regression that are not based on assumptions of normality and user-specified model statements.These models originated in the 1960s with the development of AID (Automatic Interaction Detection) by Morgan and Sonquist. In the 1980s, statisticians Breiman et al. (1984) developed CART(Classification And Regression Trees). The fundamental idea is to recursively partition the regressors' space in regions (build a tree) until all the subspaces are sufficiently homogeneous in order to estimate the regression function with the sample average (or the specific local model employed) in each region.

Another class of nonlinear models that we focus on in this paper are neural networks. These are flexible function forms motivated by the way the brain processes information. neural networks consist of a cascade of simple computational units called neurons, which are highly interconnected. Depending on how they are constructed, neural nets can approximate functions that are generally unknown (see Kuan & White, 1994; Bishop, 1995; Hastie, Tibshirani & Friedman, 2009; Teräsvirta, Van Dijk &Medeiros, 2005; Teräsvirta, Tjostheim & Granger, 2010; Varian, 2014).

A neural network is an interesting area of machine learning. It is simultaneously one of the oldest and one of the newest areas. The work on neural networks goes back to the 1940s when researchers tried to build models of the brain. Perceptron, which is an extremely simplified computational model of a biological neuron and a very simple precursor of linear models, goes back to the 50s and people showed amazing performance of the perceptron on a number of problems. Perceptron of course was limited in what it could do, so later on research related to the neural network basically died. It was reborn in the 1980s when researchers figured out how to put multiple perceptrons together into a network and they learnt how network weights with the Backpropagation algorithm worked. Again there was a great deal of excitements because these models finally seemed to be able to solve all kinds of learning problems. At the same time more powerful regression models came along, support vector regressions, so neural networks again fell out of fashion and were shelved. They returned for a second time recently when people finally figured out how to train them reasonable quickly on a massive scale and a big part of that is due to the changes in hardware that have occurred since the 1980s. It is also worth mentioning that there has been a resurgence in the field of artificial neural networks in recent years, known as "Deep neural networks". Deep neural networks use multiple stages of nonlinear computation and have won numerous contests on an array of complex tasks ranging from pattern recognition and machine learning.

Only a few attempts considering nonlinear dynamics in high-dimensional setting exist. Bai & Ng (2008b), and Raviv & Van Dijk (2014), for instance, included the quadratic principal components PCs and the first level cross-products of the original variables to capture nonlinearities. Their model is nonlinear in the variables, but it is linear in the parameters. Meanwhile, Exterkate et al (2013) applied kernel methods to a ridge regression to introduce a nonlinear ridge regression. Giovannetti (2011) improved the factor model by running a nonlinear regression on linear PCs.

With the rise of big data and the real opportunities that machine learning now brings, there is no better time to find out how novel techniques can be used for statistical research. The purpose of this thesis is to develop accurate predictive models based on machine learning techniques for high-dimensional and complex data. To overcome the curse of dimensionality and to manage data complexity, we suggest two high-dimensional nonlinear time series method. First, we introduce a nonlinear forecast model based on a combination of factor models and neural network and then we introduce another forecast model based on a shrinkage estimation of neural networks. Regarding financial forecasting, the proposed approaches can consider the comovement between financial returns and can go beyond the covariance structure of data.

*A hair divides what is false and true.*

Omar Khayyám

# 2

# Testing for Nonlinearity in Financial Time Series

## 2.1 Introduction

Many real-world phenomena and series are nonlinear in nature, however a common assumption that is often made in time series analysis is that the series conforms to a linear model. In this chapter, we provide a tentative classification of the various linearity tests that have been proposed in the literature and we briefly review some of them.

Classification of different statistical approaches which are testing nonlinearity in time series is a challenging task as they entail consideration of various types of nonlinear dynamics and are coming from different disciplines. There are only a few papers available in the literature that try to establish a classification of linearity tests. Granger and Teräsvirta (1993), Teräsvirta, Tjøstheim and Granger (2010) and recently Giannerini (2012) are examples of this.

The main idea behind various linearity tests is a hypothesis testing procedure. Every hypothesis test starts with a null hypothesis ($H_0$) and an alternative ($H_1$). In general, the null hypothesis of linearity tests states that observed series are generated by Gaussian linear stochastic processes against an alternative hypothesis that states observed series are rooted in nonlinear dynamics. To be more precise, $H_0$ tests the hypothesis that the time series is completely specified by its first and second order statistics (i.e. mean, variance, and autocorrelation or its frequency domain counterpart, power spectrum).

In this thesis, first, we classify linearity tests based on their alternative hypothesis into two broad categories. The tests with a specified model as an alternative and those against a nonspecified alternative. Tests with a specified alternative are also usually called Lagrange Multiplier (LM) tests. LM tests refer to those tests focusing on the coefficients of a nonlinear specified model (i.e. threshold autoregressive model: Tong, 1978, 1983 and Tong and Lim, 1980; and autoregressive conditional heteroskedasticity: Engle, 1982) and are parametric. In this case, Wald and Likelihood Ratio are not applicable directly when a specified nonlinear alternative is only identified under the null hypothesis of linearity; see Granger and Teräsvirta (1993, ch 6).

On the contrary, many of the tests proposed in the literature are against a nonspecified alternative. This group of tests has been more popular in the applications when testing linearity is the main aim. However, they can not assist for model building exercises such as forecast problems. In some of the tests in this group, classified as Diagnostic tests, the null hypothesis states that the series is explained by a white noise process and residuals of a properly specified linear fit should be independent versus the alternative of serial dependence. Therefore different aspects of such time series come under the investigation of different diagnostic tests. For example some tests like RESET (Ramsey, 1969), Keenan (Keenan, 1985) and Tsay's F test (Tsay, 1986) consider an auxiliary regression of residuals on a specific function of $X_t$, some other ones such as Ljung-Box (Ljung and Box, 1978) and Mcleod-Li (McLeod and Li, 1983) employ the autocorrelation function of the residuals, some like BDS (Broock et al., 1996) also measure the density structure of such residuals from a linear fit in an embedded phase space and etc.; see Tong (1990, ch 5) and Li (2004) for an overview of diagnostic tests.

In the rest of the tests against a nonspecified alternative, null hypothesis and alternatives are not based on a linear or nonlinear fit and have the ability to detect the presence of specific nonlinear features as we call them here tests for nonlinearity. For instance, tests based on higher order statistics can detect asymmetry and reversibility in time series utilizing the fact that these statistics do not contain new information for linear series and hence for example the bicorrelation of such data over time (or the bispectrum over different frequencies) is constant; see Mendel, 1991; Nikias and Petropulu, 1993 and Petropulu, 1999.

The asymptotic null distribution of the classical nonlinearity test statistics mostly depend on rigorous assumptions and are not always accurate, thus to improve the power of tests, randomization, and bootstrap approaches are introduced in the linearity tests literature, widely known as the method surrogate data, the main idea of which is to compare the value of a discriminating nonlinear measure for observed series to that of surrogates series in order to detect a meaningful deviation ( Theiler et al., 1992 and Schreiber and Schmitz, 2000). A subsection is devoted to different surrogate generating algorithms and also the discussion of appropriate test statistics.

To understand better how linearity tests work, we first explain the definition of a nonlinear process.

## 2.2 Definition of a Nonlinear Process

In contrast to a linear stochastic process which can be defined in terms of an arithmetic sequence of independent and identically distributed random variables in time domain or the power spectrum in the frequency domain, a nonlinear process is generated by a nonlinear dynamic equation of iid random variables consisting of the current and past shocks. Assume that a weakly stationary and purely stochastic process $x_t$ is said to be a linear time series when $t = 1, ..., T$. Hence, Wold's decomposition theorem says that $x_t$ can be represented as an infinite moving average process in time domain as follows

$$x_t = \sum_{i=0}^{\infty} \psi_i \varepsilon_{t-i} \tag{2.1}$$

where $\psi_0 = 1$, and $\varepsilon_t$ is the iid terms with $\mathbb{E}[\varepsilon_t] = 0$, $\mathbb{E}|\varepsilon_t|^2 < \infty$ and $\sum_{i=0}^{\infty} \psi_i^2 < \infty$ . Any stochastic process that does not satisfy the condition of above equation is said to be nonlinear. $x_t$ can also be described in terms of power spectrum in the frequency domain. Given the Fourier transformation of $x_t$ in terms of amplitude spectrum $A(f)$ and phase spectrum $\phi(f)$

$$\begin{aligned} y(f) &= F[x_t] = A(f)\, e^{j\phi(f)} \\ note: \quad e^{j\phi(f)} &= cos(\varphi(f)) + j\, sin(\phi(f)) \end{aligned} \tag{2.2}$$

Fourier transform maps time series into series of their amplitudes and phases (frequency) that composed the time series. Moreover, The power spectrum is retrievable by $A^2(f)$. In general the equation is said to represent a linear process if amplitude, $A(f)$, retains all the information about $x_t$ and even a substitution of the original phase spectrum, $\phi(f)$, with a random sequence between 0 and $2\pi$, preserve all the relevant information contained in the corresponding time series. In other words, given the inverse Fourier transformation of $y(f)$ with randomized phases ,$\hat{\phi(f)}$, returns a seriess, $\hat{x}$, which is indistinguishable from $x_t$ in a statistical sense, otherwise the process is nonlinear. Campbell, Lo, and MacKinlay (1997) define a nonlinear data generating process as on that can written

$$x_t = f(\varepsilon_t, \varepsilon_{t-1}, ...) \tag{2.3}$$

Where$f(.)$ is a nonlinear function. Nonlinearity may arise in different ways. The characteristic of nonlinear time series such as higher-moment structures, time-varying variance, asymmetric fluctuations, thresholds and breaks can be only modelled by an appropriate nonlinear function like $f(.)$ and a linear process is not adequate to model these features. A slightly more specific nonlinear process can be defined as

$$x_t = g(\varepsilon_t, \varepsilon_{t-1}, ...) + \varepsilon_t \sigma^2(\varepsilon_t, \varepsilon_{t-1}, ...) \tag{2.4}$$

Where g is a function of past error terms only (if $g(.)$ is nonlinear, model is nonlinear in mean) and $\sigma^2$ is a variance term (if $\sigma^2(.)$ is nonlinear, model is nonlinear in variance).

## 2.3 Tests: A Review

This subsection covers some of the statistical approaches that can be used to test for nonlinearity in both univariate and multivariate time series.

### 2.3.1 Nonlinearity Detection Using Higher Order Statistics (Spectra)

As mentioned before, if the time series is linear and has a Gaussian probability density function, it can be completely characterized by its first and second order statistics (e.g., mean, variance, power spectrum and autocorrelation) and consequently the higher order statistics (HOS) are either zero or contain redundant information (see Bendat and Piersol, 1993). Nonetheless, many real-world series encountered in practice are non-Gaussian and nonlinear in nature and have non-zero HOS. One may conclude that, second order measures which contains no phase information, cannot adequately describing processes associated with nonlinearities then moments of higher orders are needed to fully describe series properties. [1]

The aim of this section is to give a brief overview of some of the applications of HOS in detecting and caracterizing non-Gaussianity and certain nonlinearity of time series. In practice, a normalised version of polyspectra are usually used to

[1] In practice we usually use the cumulants rather than the moments.

detect and characterize certain nonlinearities. Polyspectra is a term that use to describe the family of all frequency domain spectra, including the second order. Representing series in frequency domain can expose the periodicities of the series and can aid in understanding the data generating process. In terms of nonlinearity detection, our focus is on the bispectrum (third order spectrum) and the trispectrum (fourth order spectrum) and their normalised version, bicoherence and tricoherence spectra.[2] These can be estimated in a way similar to the power spectrum, but to compute the polspectra of high orders, more data is usually needed to get reliable estimates. Rigorous introductions to the Higher order spectral analysis can be obtained by seeing (Mendel, 1991; Nikias and Petropulu, 1993; Petropulu, 1999; Brillinger and Rosenblatt, 1967 for the asymptotic properties of the bispectrum).

A formalized nonparametric frequency domain test of Gaussianity and linearity based on third order moments was initially proposed by Subba Rao and Gabr (1980). They look at the nonconstancy of the bispectrum of a time series as a measure of non-Gaussianity and nonlinear serial dependence in a stochastic process. Shortly afterward Hinich (1982) suggested a more robust statistical test by estimating the normalized bispectrum on a grid of points by averaging the bivariate periodogram at Fourier frequencies around the points of interest, obtaining a chi-squared statistic for testing the significance of individual bispectrum estimates by exploiting its asymptotic distribution. Hinich's test has the advantage of focusing directly on nonlinear serial dependence in contrast to subsequent approaches, which actually test for serial dependence of any kind (nonlinear or linear) on data which have been pre-whitened. It worth noting that a rejection of the null hypothesis of linearity leads automatically to a rejection of Gaussianity. The test can be applied both to the original series and to the residuals and can also be interpreted as testing for the significance of the coefficients associated to the linear terms of the Wiener expansion of the solution of the process (Giannerini, 2012).

In the following, we explain the bispectrum, the trispectrum and their normalised

[2] The third and fourth order moments or cumulants (bicorrelation and tricorrelation), which are the time domain counterparts of Polyspectra, are also found to be useful in analyzing nonlinearities in time series.

versions and we show how these measures estimate the departure of a process from linearity and Gaussianity. Bispectrum is the easiest polyspectra to compute, and hence the most popular. We define bispectrum either as the Double Discrete Fourier Transform (DDFT) of the third order cumulant (or moment) function or as the mathematical expectation of the triple product of Fourier Coefficients at different frequencies (Le Caillec and Garello, 2004).

Let $\{X_t\}_{t=1}^{\infty}$ be a stationary discrete time random process, ($t$ denote discrete time) and assume, without loss of generality, that $\mathbb{E}[X_t] = 0$. The power spectrum is given by

$$S(\omega) = \sum_{n=-\infty}^{+\infty} \mathcal{M}_2^X(n) e^{-j(\omega n)} \tag{2.5}$$

and the bispectrum of $X_t$ is of the form

$$\begin{aligned} B(\omega_1, \omega_2) &= \textstyle\sum_{n_1=-\infty}^{+\infty} \sum_{n_2=-\infty}^{+\infty} \mathcal{M}_3^X(n_1, n_2) e^{-j(\omega_1 n_1 + \omega_2 n_2)} \\ &= \mathbb{E}\{\tilde{X}(\omega_1)\tilde{X}(\omega_2)\tilde{X}^*(\omega_1 + \omega_2)\} \\ &\textit{With} \\ &\mathcal{M}_3^X(n_1, n_2) = \mathbb{E}\{X_t \, X_{t+n_1} \, X_{t+n_2}\} \\ &\textit{and} \;\; \tilde{X}(\omega) = \textstyle\sum_{n=-\infty}^{+\infty} X(n) e^{-j\omega n} \end{aligned} \tag{2.6}$$

Where $\mathbb{E}\{.\}$ denotes expectation and the pair $(\omega_1, \omega_2)$ is called a bifrequency. The bispectrum is complex-valued, contains phase information, and is a function of two independent frequencies. In an analogous manner, we can substitute moment spectrum $\mathcal{M}_3^X$ with third order cumulant spectrum of $X_t$, $\mathcal{C}_3^X$. Since the estimates of the third order momentum and cumulants are hard to interpret, the DDFT is calculated (Brockett, Hinich, and Patterson, 1988). A nice property of the bispectral analysis is its robustness to possible misspecification in the prefiltering linear model. This following from the fact that the squared skewness function, is invariant to linear filtering. (Ashley & Patterson, 2001; Rusticelli et al, 2008)

Bispectrum decomposes the skewness of a time series over frequency and it is sensitive to a particular type of nonlinearity, that is asymmetric nonlinearities, but it cannot detect a nonlinearity that depends upon moments higher than the third. It has been proved (Hinich, 1982), that the skewness function based on the bispectrum is given by

$$Bic^2(\omega_1, \omega_2) = \frac{|B(\omega_1, \omega_2)|^2}{S(\omega_1)\, S(\omega_2)\, S(\omega_1 + \omega_2)} = \frac{\mu_3^2}{\sigma_\varepsilon^6} \tag{2.7}$$

Where $\sigma_\varepsilon^6 = \mathbb{E}(\varepsilon_t^2)$ and $\mu_3 = \mathbb{E}(\varepsilon_t^3)$. Bispectrum is a complex number, therefore the modulus of that is taken. Bicoherence $Bic(\omega_1, \omega_2)$, the normalisation method for the bispectrum, is constant over all frequency pairs if time series is linear. This is the basis of Hinich's test. If the $Bic(\omega_1, \omega_2)$ ratios differ significantly over different frequency pairs, he rejects the linearity of the time series. His bispectral test considers the distribution of the estimated standardized bispectrum $2|Bic(\omega_{n1}, \omega_{n2})|^2$. The asymptotic distribution of such statistics is a noncentral chi-square distribution $\chi_2^2$ under the null hypothesis of linearity. Hinich derives a test statistic based on the sample inner quartile range of $2|Bic(\omega_{n1}, \omega_{n2})|^2$ which is bounded to the choice of a smoothing parameter. A peak in the bicoherence can be interpreted as the presence of Quadratic Phase Coupling, which is a specific type of nonlinearity and shows a quadratic nonlinearity in data generating process. For a review of other versions of the bispectral tests see Berg et al. (2010, 2012).

However, the bicoherence is sensitive to the asymmetric nonlinearities, it does not help to investigate symmetric nonlinearities. Therefore one has the need of the fourth order spectrum, that is the trispectrum. The trispectrum is sensitive to the signal kurtosis; therefore it can detect symmetric nonlinearities. The normalisation of the trispectrum leads to the tricoherence which is sensitive to the cubic phase coupling. A useful feature of the bicoherence and tricoherence functions is that they take values bounded between 0 and 1. (Collis et al., 1998)

### 2.3.2 Diagnostic Tests

A major, well-known and widely utilized group of linearity tests, applied on the residuals of a linear model, are called "diagnostic" tests. The main purpose of such tests is to go through the residuals looking for any sign of model inadequacy using different approaches each related to a specific nonlinear feature. Since any deviation from i.i.d could be a matter of interest and these deviations can be measured in various forms, the literature on these tests is vast and outside the scope of this chapter, for more information see Li (2004), Kuan (2008) and Tong (1990). In this thesis we especially focus on three areas to which the diagnostic tests have been applied. First subcategory is the tests focusing on the residuals' autocorrelation and the fact that if the preliminary filtering was sufficient there should be no further autocorrelation between the residuals. The second subcategory is more general and tries to examine serial independence. A famous test in this category is the BDS test of Brock, Dechert, and Scheinkman (Brock et al., 1987). The third subcategory is what is known as tests of goodness-of-fit which use statistical significance of auxiliary regressions of residuals.

A major drawback of diagnostic tests is that the rejection of their null hypothesis is only an indication for presence of some serial relation in the residuals and hence model inadequacy. However, this becomes a problem when one interprets that as a definite sign of nonlinearity. The reason behind the rejection might simply be that the fitted model is not sophisticated enough and there is a better linear model that can capture the underlying data in a way that the resulting residuals happen to be i.i.d. Having that in mind, we elaborate on the subcategories mentioned above in the coming sections.

#### 2.3.2.1 Tests Based on Autocorrelation Function

the basic idea behind this group of tests is to use the asymptotic distribution of an i.i.d. series' autocorrelation. The main portmanteau statistic which formed a basis for the later tests is known as Box-Pierce Q statistic ($Q_{BP}$) (Box and Pierce, 1970). $Q_{BP}$ is the sum of the squared sample autocorrelations ($\hat{r}$) up to $m$ lag and multiplied by sample size.

$$Q_{BP}(m) = n \sum_{k=1}^{m} \hat{r}_k^2$$

Under the assumption that the series is i.i.d. $Q_{BP}$ has the asymptotic $\chi^2$ distribution with $m$ degrees of freedom (in case of ARMA residuals, $m - p - q$ degrees of freedom). Later, Ljung and Box (1978) propased a modified version of $Q_{BP}$ with the same asymptotic distribution having a better performance with finite samples.

$$Q_{LB}(m) = n(n+2)\sum_{k=1}^{m}\frac{\hat{r}_k^2}{n-k}$$

Davies and Newbold (1979) use $Q_{LB}$ for sample sizes of 50, 100 and 200 and show that despite the modifications it still has low power for small sample sizes. However, as the sample size increases the performance of the test improves dramatically. McLeod and Li (1983) later used the same $Q_{LB}$ on the squared version of the residuals and presented a test statistic ($Q_{ML}$) with the same assymptotic distribution for the null hypothesis of no ARCH which is of higher applicability on financial data and is asymptotically equivalent to LM test of Engle (1982).

#### 2.3.2.2 Tests for Serial Independence

Testing for serial independence is a much broader task since it is not only confined to autocorrelation and one has to check for any kind of relations between the residual and its time delay. The difference between the tests in this subcategory and other tests is that in $Q_{LB}$, for example, one is looking for one of the symptoms of not being i.i.d. which is to have serial correlation and if there are other deviations from randomness there is no guaranty that these tests are going to be capable of recognizing it. Tests of serial independence, on the other hand, are designed to reject the null hypothesis if the series shows any sign of not being i.i.d. Thus, the rejection of null hypotheses in this case is just indicative of the fact that the series under investigation is not i.i.d. and one cannot draw conclusions about the kind of existing deviation. This becomes a problem if one wishes to not only show the inadequacy of the applied linear model, but also to conclude presence of a nonlinear motion.

The most famous test in this subcategory is BDS (Brock et al.,1996) named after the three authors who first developed it. The null hypothesis for BDS That the data are pure noise and it has been argued to have power to detect a variety of departures from randomness - linear or non-linear stochastic processes, deterministic chaos, etc. (see Brock et al., 1991). BDS uses correlation sum as an estimator of correlation integral ($C_{\varepsilon}(m)$) which basically indicates the probability of two points closer than $\varepsilon$ in phase space of time delay embedding. BDS then uses the relation between the correlation integral of the first and other dimensions of an i.i.d series which states $C_{\varepsilon}(m) = C_{\varepsilon}(1)^m$ and turns it into a test to check whether sample estimations of the $C_{\varepsilon}(m)$ and $C_{\varepsilon}(1)$ are sufficiently close.

#### 2.3.2.3 Tests using auxiliary regression

The idea behind the tests in this subcategory is simply an statistical F test in which a linear regression of the series on its lags is the restricted model and by adding a nonlinear part to it the unrestricted model is built. Hence, if the coefficients of the nonlinear part significantly differ from zero the linear part is considered to be inadequate. The regression error specification test (RESET) of Ramsey (1969) uses the higher orders of the fitted value as the nonlinear part. Keenan (1985) later used Volterra expansion and modified RESET, presenting a simplified version which only uses square of the fitted values to deal with the problem of multicollinearity. Tsay (1986) improved the power of the Keenan test, generalizing it by allowing for the all cross products of the lagged values looking for quadratic serial dependence in the data.

### 2.3.3 Specification and Lagrange multiplier tests

As discussed before, linearity tests can be divided into two main categories: the ones against a non-specified alternative and the ones against a specified alternative. These Specification tests are especially interesting since not only they are a test to identify nonlinear time series, but also they help the modeling step. Tests with a specified alternative are also usually called Lagrange Multiplier (LM) tests. LM tests refer to those tests focusing on the coefficients of a nonlinear specified model and are parametric. In this case, Wald and Likelihood Ratio are not applicable directly when a specified nonlinear alternative is only identified under the null hypothesis of linearity and Lagrange multiplier tests only require estimation of the linear model and makes the estimation of the nonlinear model unnecessary (Granger and Teräsvirta, 1993, ch 6; Luukkonen, et al., 1988 and Saikkonen and Luukkonen, 1988). The classic nonlinear models considered for LM test include the threshold autoregressive (TAR) model (Tong 1978, 1983), the exponential autoregressive (EXPAR) model (Ozaki 1982, 1985) and the bilinear model (Granger and Andersen 1978, Subba Rao and Gabr 1984). The TAR model has been especially developed, introducing a verity of other models used as the alternative hypothesis of LM tests. Smooth transition autoregression (STAR) models (Chan and Tong, 1986), self-exciting threshold autoregressive (SETAR) models (Hansen, 1999), logistic smooth transition autoregressive (LSTAR) model (Chan and Tong, 1986), Multiple Regime STAR (MRSTAR) model (Dijk and Franses, 1999) are some advancements on the TAR model, to name a few, see Teräsvirta (1994) and Dijk, et al. (2000). The alternative models of LM tests can also be an ARCH or GARCH model which makes them of a special interest in the field of finance. The famous and widely used Engle's LM test of ARCH effect (Engle, 1982) is an example for such alternatives. It can be shown that many forecasting techniques arise as a natural extension of, and as a complement to, existing specification tests.

### 2.3.4 The method of surrogate data

In traditional linearity tests, the statistic is compared to an asymptotic distribution. This distribution is in most cases based on rigorous assumptions and conditions that are hard to satisfy. The method of surrogate data provides us with an interesting approach which in turn gives a better estimation of the statistic's distribution with assumptions easier to fulfill and enables us to discriminate between the original series and a set of generated linear time series using a vast variety of nonlinear features. The procedure of surrogate data method is essentially similar to parametric bootstrap which has been described by Theiler et al. (1992), considered to be the seminal paper on the subject, and later by Schreiber and Schmitz (2000) in their review paper. In general, the procedure is as follows: 1- One starts with the desired null hypothesis regarding the process of the data, i.e., in case of linearity, the data generating process (DGP) is a linear stochastic one. 2- A set of surrogate series are generated consistent with the null hypothesis and resembling the original series in those aspects but random in all the other aspects. 3- A discriminating statistic is chosen and its value is calculated for both the original series and the surrogate set. In the area of linearity tests, this statistic should be capable of characterizing weak nonlinear signatures. 4- The value of the nonlinear measure calculated for the original series is compared to the distribution of that for the surrogates and if it falls in the critical region and is atypical of the surrogates, the null hypothesis is rejected and hence the presence of a nonlinear movement is inferred.

As it is obvious from the steps mentioned above the method of surrogate data depends heavily on two areas demanding delicacy. First, the algorithm used to generate surrogates and second, the discriminating statistic. The surrogate DGP has to resemble the original data regarding its first and second order statistics as much as possible and simultaneously perturb any other ostensible relations. For the latter, it has been discussed previously in this article that different nonlinear measures focus on different aspects of such series and thus one should be careful about the implied reasoning behind the used statistic. Another point about the test statistic is whether it is "pivotal" and also if the surrogate DGP takes this into account. For more information on that see Theiler and Prichard (1996). Thus, the fact that the null hypothesis is rejected or not might turn out to be spurious if one does not pay attention to these subtle but crucial points. In the next parts we wish to investigate these two areas more thoroughly.

#### 2.3.4.1 Algorithms for the generation of surrogate data

The surrogate DGPs are divided into two broad categories: typical realization and constrained realization. The constrained realization surrogates are limited to take the exact same parameter values as the original series. Typical realization on the other hand assigns a certain type of process to the original series and generates new sets of data according to that. For example, if the null hypothesis is that the data was generated by a Gaussian process, the typical realization approach would be to estimate the mean and standard deviation of the original series and then generate a set of random values with that asymptotic distribution. As it is obvious, the surrogate created in this way does not necessarily have the same sample mean and standard deviation as the original series. The constrained approach, however, is to generate a random series and then rescale it to take the exact same sample mean and standard deviation. Theiler et al. (1992) explain that essentially the typical realization approach is fitting the model and hence good for prediction but the constrained realization surrogates provide an extremely over fit model suitable for hypothesis testing.

In case of linear hypothesis, the desired parameters to be preserved are the first and second order statistics specifically the autocorrelation function or its frequency domain counterpart, the periodogram[3]. Because of their characteristics the former is usually used for the typical realization and the latter is usually utilized for the constrained realizations. There is a vast amount of literature trying out different methods to produce surrogates for the linearity hypothesis each improving a previous model or dealing with certain obstacles.

#### 2.3.4.1.1 Typical realization approach

In Hjellvik and Tjøstheim (1995) authors describe a naïve bootstrap approach which feeds back a resampling of the residuals of a fitted model to the model itself for the purpose of bootstrap test. This residual bootstrap approach uses an AR model and for its model selection relies on the idea of sieve approximation and hence it is also known as AR-sieve bootstrap. See Buhlmann (2002). Two more

[3] Note that the periodogram is not exactly equivalent to autocorrelation function. In other words, peridogram represents "periodical" autocorrelation function in the frequency domain and that is the source of some problem in this field. There are a few ways to circumvent this problem such as windowing and zero padding. See Schreiber and Schmitz (2000)

recent typical realization are statically transformed autoregressive process (STAP) and autoregressive-to-anything (ARTA) in which a time series generated by Gaussian AR(P) with estimated coefficients based on the original time series is transformed to have the given marginal distribution. The only difference between these two methods is in their marginal transform function, see Cario and Nelson (1996) and Kugiumtzis (2002). Later a multivariate version of these two was developed in order to generate multivariate surrogates Kugiumtzis and Bora-Senta (2014).

#### 2.3.4.1.2 Constrained realization approach

As for constrained realization algorithms the most basic one is known as Fourier transform (FT). The idea behind this approach is simple. Using discreet Fourier transform the original series is brought to the frequency domain and is divided to two parts: the amplitude and the phase of each frequency. For generating a surrogate series, the amplitudes are kept and the phases are replaced by a random phase between 0 and $2\pi$ and then the Fourier transform is inverted. Based on Wiener-Khinchin Theorem, by reserving the amplitudes the surrogate series has exactly the same autocorrelation function, however, replacing the phases has destructed any possible extra information in the original series. Later, Theiler et al (1992) improved this process and introduced Amplitude Adjusted Fourier Transform (AAFT) algorithm. The algorithm uses a Gaussian series rank-ordered based on the original series and makes an FT surrogate of it. The original series is then rank-ordered based on the FT surrogate to make an AAFT surrogate of the original series. Schreiber and Schmitz (1996) show that this algorithm has a bias toward a flat spectrum and suggest and iterative algorithm of AAFT (IAAFT) instead. One starts with a random shuffle of the original series (without replacement) and then the process in consisted of two iterative step. First, the amplitudes of the shuffled series are replaced by that of the original series (using a Fourier transform and then inversing it). second, the original data is rank-ordered based on the resulting series. The amplitudes of this new reordered original data is again replaced by the amplitudes of the original series and so on. The iteration continues until no new reordering happens in two consecutive iterations. Venema et al (2006) improved the IAAFT using a stochastic approach (SIAAFT) to only replace a part of the amplitudes in the first stage. This results in a slower convergence and hence a lower chance of getting stuck in a local minimum. There is also an abundance of constrained realization algorithms which do not use the Fourier transform. Simulated annealing, for example, is a famous one which is frequently used and employs concepts of thermodynamic in optimization. In order to achieve that, the problem of creating surrogate series is transformed to an optimization of a single-valued cost function. In the area of linearity testing, this cost function is the discrepancy between the autocorrelation function of the original series and its surrogate. In some areas of study, the data under investigation is not a stationary one, and hence the Fourier based algorithms would not be suitable. In order to evade this hindrance, a set of algorithms are proposed using wavelet transform instead of Fourier. The basics of such algorithms closely follow that of the FT. Wavelet transform breaks the data into two set of coefficients known as Approximation and Detail coefficient and the surrogate generation algorithms keep the approximation coefficient while manipulating the detail coefficients. This manipulation is often done using the Fourier based algorithms. For more information, see Breakspear et al (2003) and Keylock (2006, 2007, 2008 and 2010).

#### 2.3.4.2 Discriminating Statistics

As it was discussed before, virtually any statistic capable of characterizing weak nonlinearity can be utilized in the method of surrogate data with the hypothesis of linearity. Schreiber and Schmitz (1997) in their leading paper investigate the power of a few different statistics used with surrogate data a few of which was discussed in previous sections. Taken's estimator and BDS use correlation dimension and rely on the embedding dimension. They also use bicorrelation as a higher order statistic. There are two other nonlinear measures which we have not discussed prior to this point mainly because their asymptotic distribution is either unknown or hard to estimate and hence are hardly used without surrogate data. The first one is a simple nonlinear prediction error which predicts one step ahead of each point in m dimension by smiley averaging over the future values of all neighboring delay vectors closer than a certain threshold. The next one is a measure of time-reversibility. A time series is regarded as "reversible" if its probabilistic properties do not change when the time is reversed. If this hypothesis is rejected, one can also reject the hypothesis of linear Gaussian stochastic process. Schreiber and Schmitz (1997) conclude that other than nonlinear prediction error which consistently gives good discriminating power, rest of the nonlinear measures give better performance in some cases but completely fail in others. Specifically, time reversal asymmetry measure which has the best power regarding the henon map but cannot result in any significant rejection for noisy Lorenz data. In their paper, Giannerini et al (2015) propose two metric entropy measure based on Bhattacharya-Hellinger-Matusita distance. The null hypothesis for the first one is that the generating process is linear and Gaussian, but the second statistic only assumes linearity as its null hypothesis. In their paper, the proposed statistics show promising result against simulated data specially in finding the lag with nonlinear motion.

## 2.4 Testing for Nonlinearity in Financial Time Series

In this section, we report the results of the implementation of a selection of tests introduced before on financial time series in order to test the presence of nonlinearity in real world financial data based on an eclectic set of tests. The series analyzed are daily returns of $m = 50$ equities on the S&P 500 index from 04.01.2005 through 31.12.2014, for a total of 2517 observations and were taken from the Bloomberg. In all cases, we have applied fifteen tests in four major groups: HOS, Diagnostic, LM and Method of surrogate data. As we can see in Table.2.4.1 and Table.2.4.2, the evidence against linearity is clear in some of the series as almost all tests provide the same outcome. But, in many cases, the result of the tests are different for an individual series which means that the return series follow different level and kind of nonlinearity. As the rational behind tests are different and different tests has been designed in a way to capture a different kind of nonlinearities, we can

**Table 2.4.1:** The results of the application of linearity tests to equity returns. A rejection of the null hypothesis of linearity is shown by ✓.

| ind | HOS | Diagnostic | | | | | | | | | LM | Method of surrogate data | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Serial dependence and serial correlation | | | | | | | | Auxiliary Regression | | AAFT | | Simulated Annealing | |
| | | AR(1) | | | ARMA | | | ARMA-GARCH | | | | | | | |
| | BiSpectral | BDS | LB | ML | BDS | LB | ML | BDS | LB | RESET | SETAR | TA | Srho | TA | Srho |
| 1 | × | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | × | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| 2 | × | ✓ | × | ✓ | ✓ | × | ✓ | × | × | × | ✓ | ✓ | ✓ | ✓ | ✓ |
| 3 | ✓ | ✓ | × | ✓ | ✓ | ✓ | ✓ | ✓ | × | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| 4 | × | ✓ | ✓ | ✓ | ✓ | × | ✓ | × | × | × | ✓ | ✓ | ✓ | ✓ | ✓ |
| 5 | × | ✓ | ✓ | ✓ | ✓ | × | ✓ | ✓ | × | × | ✓ | ✓ | ✓ | ✓ | ✓ |
| 6 | × | ✓ | ✓ | ✓ | ✓ | × | ✓ | ✓ | × | × | × | × | ✓ | ✓ | ✓ |
| 7 | × | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | × | × | ✓ | ✓ | ✓ | ✓ | × | ✓ |
| 8 | × | ✓ | ✓ | ✓ | ✓ | × | ✓ | × | × | ✓ | ✓ | × | ✓ | ✓ | ✓ |
| 9 | × | ✓ | × | ✓ | ✓ | × | ✓ | × | × | × | ✓ | ✓ | ✓ | ✓ | ✓ |
| 10 | × | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | × | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| 11 | × | ✓ | ✓ | ✓ | ✓ | × | ✓ | × | × | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| 12 | × | ✓ | ✓ | ✓ | ✓ | × | ✓ | × | × | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| 13 | × | ✓ | × | ✓ | ✓ | × | ✓ | × | × | × | ✓ | × | ✓ | ✓ | ✓ |
| 14 | × | ✓ | × | ✓ | ✓ | × | ✓ | × | × | × | ✓ | ✓ | ✓ | ✓ | ✓ |
| 15 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | × | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| 16 | × | ✓ | ✓ | ✓ | ✓ | × | ✓ | × | × | × | ✓ | ✓ | ✓ | × | ✓ |
| 17 | × | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | × | × | × | ✓ | ✓ | ✓ | ✓ | ✓ |
| 18 | × | ✓ | ✓ | ✓ | ✓ | × | ✓ | ✓ | × | × | × | × | ✓ | ✓ | × |
| 19 | × | ✓ | ✓ | ✓ | ✓ | × | ✓ | ✓ | × | × | ✓ | ✓ | ✓ | ✓ | ✓ |
| 20 | × | ✓ | × | × | ✓ | × | × | ✓ | × | × | ✓ | ✓ | ✓ | ✓ | ✓ |
| 21 | × | ✓ | ✓ | ✓ | ✓ | × | ✓ | × | × | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| 22 | × | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | × | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| 23 | ✓ | ✓ | ✓ | ✓ | ✓ | × | ✓ | × | × | × | ✓ | ✓ | ✓ | × | ✓ |
| 24 | × | ✓ | × | ✓ | ✓ | × | ✓ | ✓ | × | × | × | ✓ | ✓ | × | ✓ |
| 25 | ✓ | ✓ | × | ✓ | ✓ | × | ✓ | × | × | × | ✓ | × | ✓ | × | ✓ |

guess about the data generating process of the series.

For all the tests up to three lags (in cases with embedded dimensions such as the BDS test up to three dimensions) was tested and if the hypotheses of linearity was rejected at 95% significance level on any of those lags, the time series was considered as nonlinear (shown by check marks in the table). The linear serial dependence is removed from the series through different pre-whitening models in the following way: AR(1) model, ARMA(p,q) model and ARMA-GARCH model for values from p,q = 0 to 5. The optimal lag is chosen to minimise the Akaike information criterion (AIC). LB does not reject the null hypothesis of no serial correlation for 18 of the series in their AR(1) residuals. Oddly enough LB rejects the null hypothesis for two of these 18 series in their Best ARMA residuals. This might mean that a naive approach such as AR(1) might be better capable of filtering serial

**Table 2.4.2:** The results of the application of linearity tests to equity returns. A rejection of the null hypothesis of linearity is shown by ✓.

| ind | HOS | Diagnostic | | | | | | | | | LM | Method of surrogate data | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Serial dependence and serial correlation | | | | | | | | Auxiliary Regression | | AAFT | | Simulated Annealing | |
| | | AR(1) | | | ARMA | | | ARMA-GARCH | | | | | | | |
| | BiSpectral | BDS | LB | ML | BDS | LB | ML | BDS | LB | RESET | SETAR | TA | Srho | TA | Srho |
| 26 | × | ✓ | ✓ | ✓ | ✓ | × | ✓ | × | × | × | ✓ | ✓ | ✓ | ✓ | ✓ |
| 27 | × | ✓ | × | ✓ | ✓ | × | ✓ | ✓ | × | × | ✓ | × | ✓ | ✓ | ✓ |
| 28 | × | ✓ | × | ✓ | ✓ | ✓ | ✓ | ✓ | × | × | ✓ | ✓ | ✓ | ✓ | ✓ |
| 29 | × | ✓ | × | ✓ | ✓ | × | ✓ | × | × | × | × | ✓ | ✓ | ✓ | ✓ |
| 30 | × | ✓ | ✓ | ✓ | ✓ | × | ✓ | × | × | × | ✓ | ✓ | ✓ | × | ✓ |
| 31 | × | ✓ | ✓ | ✓ | ✓ | × | ✓ | × | × | × | ✓ | ✓ | ✓ | ✓ | ✓ |
| 32 | × | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | × | × | ✓ | ✓ | ✓ | ✓ | ✓ |
| 33 | × | ✓ | × | ✓ | ✓ | × | ✓ | × | × | × | ✓ | ✓ | ✓ | × | ✓ |
| 34 | × | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | × | × | × | × | ✓ | ✓ | ✓ | ✓ |
| 35 | × | ✓ | × | ✓ | ✓ | × | ✓ | × | × | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| 36 | × | ✓ | ✓ | ✓ | ✓ | × | ✓ | × | × | × | ✓ | ✓ | ✓ | ✓ | ✓ |
| 37 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | × | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| 38 | × | ✓ | × | ✓ | ✓ | × | ✓ | ✓ | × | × | ✓ | ✓ | ✓ | ✓ | × |
| 39 | × | ✓ | ✓ | ✓ | ✓ | × | ✓ | × | × | × | ✓ | ✓ | ✓ | ✓ | ✓ |
| 40 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | × | × | × | ✓ | ✓ | ✓ | ✓ | ✓ |
| 41 | × | ✓ | × | × | ✓ | × | × | ✓ | × | × | × | ✓ | ✓ | ✓ | ✓ |
| 42 | × | ✓ | ✓ | ✓ | ✓ | × | ✓ | ✓ | × | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| 43 | × | ✓ | ✓ | ✓ | ✓ | × | ✓ | × | × | × | ✓ | ✓ | ✓ | ✓ | ✓ |
| 44 | × | ✓ | ✓ | ✓ | ✓ | × | ✓ | × | × | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| 45 | ✓ | ✓ | × | ✓ | ✓ | × | ✓ | × | × | × | ✓ | ✓ | ✓ | ✓ | ✓ |
| 46 | ✓ | ✓ | ✓ | ✓ | ✓ | × | ✓ | ✓ | × | × | ✓ | × | ✓ | ✓ | ✓ |
| 47 | × | ✓ | ✓ | ✓ | ✓ | × | ✓ | × | × | × | ✓ | × | ✓ | ✓ | ✓ |
| 48 | × | ✓ | × | × | ✓ | × | × | ✓ | × | × | × | ✓ | ✓ | ✓ | ✓ |
| 49 | × | ✓ | × | ✓ | ✓ | × | ✓ | × | × | × | ✓ | × | ✓ | ✓ | ✓ |
| 50 | × | ✓ | ✓ | ✓ | ✓ | × | ✓ | × | × | × | × | ✓ | ✓ | ✓ | ✓ |

correlation from a series than a roundabout approach such as Best ARMA which uses AIC to find the best ARMA lag.

Using ARMA-GARCH pre-whitening, one can capture nonlinear motions of most of financial series in a way which BDS recognizes the residuals as i.i.d and LB is unable to find any serial correlation in them. However, there are some series that show captured nonlinearities in their ARMA-GARCH residuals which is indicative of some nonlinear motion other than that of offered by GARCH model. The fact that the results of different tests overlap but do not cover each other completely can be construed as an indicative of the fact that the verity of measured aspects offered by these different tests is a necessity for any researcher wishing to study nonlinear motions of financial series.

For the method of surrogate data two different algorithms of surrogate generation (AAFT and Simulated annealing) were used accompanied by time asymmetry statistic (TA) and the entropy measure of Giannerini et al (Srho).

We examine the nonlinear dependencies between return series using Kruger et al. (2008) test in the next chapter. The test is designed to detect nonlinearities based on nonlinear principal component analysis and between time series.

*Definition of Statistics: The science of producing unreliable facts from reliable figures.*

Evan Esar

# 3

# Nonlinear Forecasting Using a Large Number of Predictors

## 3.1 Introduction

World data currently doubles every couple of years with an on-going steady increase in computing power that poses new challenges for economic modelling and forecasting in a big data environment. It challenges state-of-the-art data acquisition, computation and analysis methods. To benefit from many new potential explanatory variables, dimension reduction methods (i.e, Factor models - Stock & Watson, 2002, 2006; Bai & Ng, 2002; Deistler & Hamann, 2005; Forni et al, 2005; Lam & Yao, 2012), shrinkage techniques (Ridge - Hoerl & Kennard, 1988; LASSO - Tibshirani, 1996; Elastic Net - Zou & Hastie, 2005), and subset selection techniques (Bayesian regression - De Mol, Giannone, Reichlin, 2008; Selecting variables - Bai & Ng, 2008a) are used to handle high-dimensional data but they are mostly linear models. Since the linear models cannot explain a number of important features common to much economic and financial data, it is logical to think about a nonlinear statistical model to concurrently handle high-dimensionality and nonlinearity. Only a few attempts considering nonlinear dynamics in high-dimensional setting exist. Bai & Ng (2008b), and Raviv & Van Dijk (2014), for instance, included the quadratic principal components PCs and the first level cross-products of the original variables to capture nonlinearities. Their model is nonlinear in the variables, but it is linear in the parameters. Meanwhile, Exterkate et al (2013) applied kernel methods to a ridge regression to introduce a nonlinear ridge regression. Giovannetti (2011) improved the factor model by running a nonlinear regression on linear PCs.

With the rise of big data and the real opportunities that machine learning now brings, there is no better time to find out how novel techniques can be used for economic research. To this end, this chaper aims at addressing high-dimensional econometric models and machine learning techniques to analyse and forecast financial big data. Through this study we introduce a nonlinear statistical factor model based on neural networks (NLFM hereafter).

In theory, the optimal forecast of a variable under quadratic loss is its expectation conditional on information available. In practice, the relevant information set might be very large and a factor-based forecast can systematically handle this information while keeping the dimension of the forecasting model small. Forecasting using factor models has recently received much attention, especially in the macroeconomic literature. However the potential of factor-based forecasts can also be investigated in the realm of finance. Several papers have been devoted to forecasting macroeconomic and financial time series with factor models in a data-rich environment where, in general, the idea is to forecast the series of interest by using the common factors estimated from a large panel of predictors, see Stock & Watson (2002, 2006); Forni et al.(2005); Deistler & Hamann (2005); Stock & Watson (2004),and Boivin & Ng (2005) for a survey of the 'factor approach methods' to forecasting.

In particular, the problem of forecasting with factor models can be divided into two steps: (1) The factor estimation step from a large panel of data and the forecasting step that uses the factor estimates to forecast the series of interest. Factors estimation and forecasting equations can be obtained in different ways. For instance, Kalman filter methods and cross-sectional averaging methods as principal components analysis (PCA) are two kind of estimates of the factors. Concerning forecasting with factor models, several approaches have been proposed based on how the common factors are estimated and how the forecast equations are formulated. There are two leading approaches in the literature that differ primarily because of the methodology used to estimate the factors. The static method, suggested by Stock and Watson (2002) (SW hereafter), and the dynamic method of Forni, Hallin Lippi and Reichlin (2005) (FHLR hereafter). SW performed a forecasting experiment for macroeconomic variables using factors estimated by PCA from a large panel of U.S. monthly predictors. In finance literature, Deistler and Hamann (2005) applied a static factor model (to be more precise, Quasi-static principal component analysis, quasi-static factor models with idiosyncratic errors) when forecasting returns on asset prices. FHLR performed a forecasting experiment based on spectral analysis (a dynamic principal components), and then applied this approach to forecast macroeconomic indicators using a large panel of Euro-area monthly predictors. Moreover, Forni et al. (2015) recently introduced a method that complements FHLR by assuming an infinite-dimensional factor space. Various papers found substantial forecasting improvements based on the mean squared errors of factor models and those obtained from simple autoregressions and more elaborate structural models, see Stock and Watson (1999); Stock, Watson and Marcellino (2003), Forni et al. (2001), and Forni and Reichlin (2001).

This study introduces a nonlinear financial forecasting methodology based on two concepts: factor models and neural networks. As we believe that understanding the comovement between financial returns is crucial for forecasting procedure, we use factor models to describe the covariance structure of financial return and produce a parsimonious representation of the market correlation and demonstrate that a small number of common components accounts for a large proportion of the variability of the equities that we consider. It is worth noting that the nonlinear factor-based forecast model that we propose is based on static factor models, however the proposed model can be extended to a dynamic setting when we deal with macroeconomic forecasting.

The rest of the chapter is organized as follows. Section II provides the basic framework of factor models, the application of factor models in forecasting financial returns, an overview of nonlinear dimensionality reduction and a review of neural networks from a statistical perspective. In Section III we introduce a nonlinear factor model concerning a nonlinear factor estimation and nonlinear forecast equation. Section IV summarizes our empirical findings, concerning data, a quick review of linearity tests in and between series, competing models, tools for validation of the forecasts and trading (portfolio) simulation. Finally, our conclusion are presented in Section V.

## 3.2 Preliminaries

In this chapter we answer the question of whether it is possible to forecast with a large panel of predictors, while considering nonlinear dynamics in data and to show how a model with such features can improve financial forecast accuracy. We propose a forecasting model that is able to handle high-dimensionality and nonlinearity by applying a neural network based principal component analysis to estimate common factors from a large panel of data and nonlinearly forecast the series of interest uses the factor estimates with a feedforward neural network. Such features can go beyond the covariance structure analysis. In the following section we initially review the basic framework of factor models and the dimensionality reduction and the neural network model, we then explain our proposed model.

### 3.2.1 Forecasting financial return series with factor models

Research into modelling and forecasting a financial time series has a long history. Several models are covered in Tsay (2005) and Campbell et al.(1996) that attempt to explain return time series using linear combinations of one or more financial market factors. The most widely studied single factor models is the capital asset pricing model (CAPM) of Sharpe (1964) and Lintner (1965) that relates the expected return of equities to the expected rate of return on a market index such as the Standard and Poor's 500 Index. Although CAPM is attractive for its simple logic and intuitively pleasing predictions about risk and expected return, the empirical record of the model is poor and it can not explain the behaviour of asset returns, see Fama and French (2004). The empirical failure of the CAPM is perhaps because of its simplicity which implies that a model with multiple factors might be required to fully explain return time series. Arbitrage pricing theory (APT) was one general model developed by Ross (1976) to shore up these deficiencies. APT presents a linear approximate model of expected asset returns using an unknown number of unidentified macroeconomic factors or theoretical market indices. The problem then becomes one of identifying the factors. Factors may be classified in terms of identification into either theoretical or statistical. The theoretical models explain asset returns based on specifying observable macro-economic and financial variables (e.g., Chen, Roll and Ross 1986) or observable defined firm-specific variables (e.g., Tsay, 2005) as factors. The relationships between the factors and the return time series are determined using linear regression. The two most frequently used statistical methods, that extract the unobservable or latent factors directly from the portfolios of return time series, are factor analysis and PCA, see Tsay (2005).

Analysis of multiple financial returns often requires high-dimensional statistical models. In practice, the return time series present similar characteristics and share the following stylised facts: comovements, nonlinearity, non-gausianity (skewness and heavy tails), volatility clustering and leverage effect, which makes the modelling of this variable hard, see Hsieh (1991); Bollerslev, Engle and Nelson (1994); Brooks (1996); Cont (2001). Hence, factor models have been proposed in the literature to study common patterns of return series. In statistical factor models, variables are represented as the sum of two mutually orthogonal unobservable components: the common components and the idiosyncratic component. This chapter focuses on statistical factor models to predict financial return series.

#### 3.2.1.1 Linear Factor model general specification

Given a high-dimensional matrix of stationary time series of financial returns, which we denote by $x_{it}$ $(i = 1, ..., m, t = 1, ..., T)$ the factor model enables us to disassemble $x_{it}$ into the few common factors representing the market or the co-movements of the returns and idiosyncratic components representing special features of a given company. $x_{it}$ consists of the $m$ series of financial returns and $T$ time series observations for each company. In terms of matrix notation, let $X$ be the $m \times T$ matrix of observed return series; $x_t$ is a row vector denoting all $m$ observations at time $t$, while $x_i$ is a column vector denoting all $T$ observations for $i^{th}$ company.

The forecast variable of interest is $x_{iT+h|T}$, the $h$-step-ahead out-of-sample forecast of a return series in the panel. It is also assumed that the series hase a zero mean and covariance $\Gamma_X(0)$. In a high-dimensional setting $m$ usually is much greater than T and the factor model reduces the dimension to make the estimation feasible. The idea is that a small number $q$ of factors should be able to explain most of the covariance of the data. A factor model also describe the covariance structure of returns and deconstructs risk and return into systemic and systematic components.

A number of different formulations of factor models are imaginable based on how data is explained by common factors and the idiosyncratic component. If the factors (shown by $u_t$ hereafter) has only a contemporaneous effect on $x_t$ and an idiosyncratic component (shown by $\xi_t$ hereafter) has no cross–sectional dependence, the setting is known as the exact static model; see Spearman (1904); Lawley and Maxwell (1971). This is estimated by

$$x_t = \lambda_i' u_t + \xi_t, \tag{3.1}$$

where $u_t$ is a vector of $q$ common factors, $\lambda_i$ is the corresponding vector of loadings for unit $i$, and $\xi_{it}$ is an idiosyncratic component. In classical factor models, $\xi_{it}$ is serially uncorrelated and iid across $i$. While in the extension to a dynamic setting, the model allows factors and the idiosyncratic component to be serially correlated. For an exact dynamic model see Sargent and Sims (1977) and Geweke (1997). The dynamic factor model representation is given as:

$$x_t = b_i'(L)u_t + \xi_t, \tag{3.2}$$

where $u_t$ is a vector of dynamic factors. In these two last equations, factors are a q-dimensional vector $(q \ll m)$ and $\lambda$ or $B(L)$ are loadings matrices of size $m \times q$. Hereafter, we will refer to $x_t - \xi_t$ as the common component $\chi_t$ in both static and dynamic cases.

Chamberlain and Rothschild (1983) introduced an approximate (or generalized) static factor model allowing weak cross–sectional dependence on the idiosyncratic component $\xi_t$ and for the generalised dynamic factor models (GDFM) see Forni and Lippi (2001); Forni, Hallin, Lippi, and Reichlin (2000). It is also worth noting that $u_t$ and/or $\xi_t$ must be serially correlated to guarantee the predictability of $x_t$.

Since financial returns are often seen to have very low autocorrelation and therefore are often modelled as white noises, this work focuses on a static method of estimating factors. The static factor model seems to be an appropriate choice and unnecessary estimate of the dynamic factor could induce some efficiency loss. The following section starts with an overview of linear and nonlinear dimensionality reduction methods emphasizing PCA and nonlinear PCA as the factor estimation step.

### 3.2.2 Overview of Nonlinear Dimensionality Reduction

A variety of dimensionality reduction methods exist, with a variety of motivations in many fields including statistics, machine learning, and applied fields for over a century. In general, dimensionality reduction methods can be used in discovering low-dimensional structure from high-dimensional and noisy observations that extracts some meaningful features of interest in the data. These methods can be interpreted as a simple optimization framework that can be broadly classified into: linear, nonlinear and proximity (graph) based methods.

In the first class we have principal component analysis PCA (Pearson, 1901; Eckart and Young, 1936), factor analysis FA (Spearman, 1904), canonical correlations analysis CCA (Hotelling, 1936), multidimensional scaling (Torgerson, 1952; Cox and Cox, 2001; Borg and Groenen, 2005), distance metric learning (Kulis, 2012; Yang and Jin, 2006), linear discriminant analysis (Fisher, 1936; Rao, 1948), and several others. Recently, Cunningham and Ghahramani (2015) surveyed the literature on linear dimensionality reduction in their work and gave insights to some rarely discussed shortcomings of traditional approaches. Linear methods are based on a linear projection assuming that the data lives close to a lower dimensional linear subspace. If the data lies on or near a low-dimensional nonlinear manifold, then linear methods, even though computationally efficient, can not model the variability of the data correctly and recover this intrinsic nonlinear structure. However, the basic geometric intuitions behind linear methods play an important role in many algorithms for nonlinear dimensionality reduction.

The algorithms designed to address the problem of nonlinear dimensionality reduction are for instance, Kernel methods (Schölkopf, Smola and Müller, 1997, 1998; Jäkel, Schölkopf and Wichmann, 2007), Neural network methods (Oja, 1982; Kramer, 1991; Hsieh, 2004; Hinton and Salakhutdinov, 2006; Kohonen' 2000). Modern methods for optimization enable a generic nonlinear dimensionality reduction solver, which accepts as input data and an objective that is to be optimised, and returns, as output, an optimal low-dimensional projection of the data. Graph based methods, like the Locally Linear Embedding (Roweis and Saul, 2000), Isomap (Tenenbaum, de Silva and Langford, 2000), Maximum Variance Unfolding (Weinberger and Saul, 2006), Laplacian Eigenmaps (Belkin and Niyogi, 2002) and several related methods are powerful and nonlinear, but their computational cost scales quadratically with the number of observations, so they generally cannot be applied to very large high-dimensional data sets.

The diversity of dimensionality reduction methods makes a direct intercomparison between different methods very difficult. There are very few studies comparing the performance of different methods on the same problem. See Fodor (2002), Burges (2004) and Cayton (2005) for a review of the most common geometrical and statistical methods of dimensionality reduction. This section treats in detail only Linear and Nonlinear PCA methods for the financial data analysis. We will begin by reviewing the notion of PCA and the extension of ideas from PCA to a nonlinear setting. Then we describe the neural networks and its application in nonlinear PCA.

#### 3.2.2.1 Extension of ideas from PCA to a nonlinear setting

Since PCA is perhaps the most popular instance of dimensionality reduction methods and it has been discussed extensively by other authors, only a brief summary is given here. PCA (also known as empirical orthogonal function (EOF), proper orthogonal decomposition (POD) and the Karhunen-Loève decomposition) in a nutshell is a technique for linearly mapping multidimensional data into lower dimensional space while preserving as much information as possible. PCA also transfer a set of correlated variables into a new set of uncorrelated variables.

Given a m-dimensional random vector of the form $\mathbf{x}_t = (x_1, ..., x_m)$, where consists of correlated data points and each variable $\mathbf{x}_i$ $(i = 1, ..., m)$ has $T$ samples labeled by the index $t$, PCA is an optimal matrix factorization of $\mathbf{x}_t$ into two vectors, $\mathbf{u}_t$ called the scores, a q-dimensional orthonormal principal components $(q \ll m)$, and $\lambda$, called the loading matrix of size $m \times q$, plus a matrix of residuals $\xi_t$ when $t$ is simply the time, and each $\mathbf{x}_i$ is a time series containing $T$ observations.

$$\mathbf{x}_t = \boldsymbol{\lambda}_i' \mathbf{u}_t + \boldsymbol{\xi}_t \tag{3.3}$$

PCA searchs for $\mathbf{u}$, a linear combination of the $\mathbf{x}_i$, and an associated vector $\lambda$. The condition of optimality on the factorization is that the Euc1idean norm of the residual matrix, $||\boldsymbol{\xi}_t||^2 = \langle ||\mathbf{x}_t - \boldsymbol{\lambda}_i' \mathbf{u}_t||^2 \rangle$ , must be minimized for the given number of factors. One can easily show that the subspace with minimum reconstruction error is also the subspace with maximum variance. The basis vectors of this subspace are given by the top q eigenvectors of the $m \times m$ covariance matrix.Therefore, to satisfy this criterion, it is known that the columns of $\lambda$ are the eigenvectors Corresponding to the $q$ largest eigenvalues of the covariance matrix of $\mathbf{x}_t$. It is also useful to view PCA as a linear projection of data from $\mathbb{R}^m$ to $\mathbb{R}^q$. PCA finds the projection such that the best linear reconstruction of the data is as close as possible to the original data. Taking $\lambda'\lambda = I$ without loss of generality, the mapping has the form:

$$\mathbf{u}_t = \mathbf{x}_t\boldsymbol{\lambda}_i \tag{3.4}$$

The loadings $\lambda$ are the coefficients for the linear transformation. The information lost in this mapping can be assessed by reconstruction of the measurement vector by reversing the projection back to $\mathbb{R}^m$:

$$\tilde{x}_t = \mathbf{u}_t\lambda' \tag{3.5}$$

where $\tilde{x}_t = x_t - \xi_t$ is the reconstructed measurement vector. In PCA, input data are projected into the q-dimensional subspace that minimizes the reconstruction error. For background and more details on PCA, see Jolliffe (2002). However, PCA provides optimally parsimonious data compression for any dataset whose distribution lies along orthogonal axes, the common drawback of this methods is that only linear structures can be correctly extracted from the data and nonlinear relations are either missed or misinterpreted by this methods and reduction of dimensions for complex distributions may need non linear processing. In nonlinear PCA, the mapping into feature space is generalized to all arbitrary nonlinear continuous function expressible by a feedforward neural network. By analogy to Eq.3.4, we seek a mapping in the form:

linear relation in PCA is now nonlinearly generalized and can be any nonlinear continuous function representable by a feedforward neural network

$$\mathbf{u}_t = \varphi^{(x)}(\mathbf{x}_t) \tag{3.6}$$

where $\varphi^{(x)}$ is a nonlinear vector function, composed of $q$ individual nonlinear functions analogous to the columns of $\boldsymbol{\lambda}$, such that if $u_i$ represents the *ith* element of $u_t$,

$$u_i = \varphi_i^{(x)}(\mathbf{x}_t) \tag{3.7}$$

The inverse transformation, restoring the original dimensionality of the data, analogous to Eq.3.5, is implemented by a second nonlinear vector function $\varphi^{(u)}$

$$\tilde{\mathbf{x}}_t = \varphi^{(u)}(\mathbf{u}_t) \tag{3.8}$$

$\varphi^{(u)}$ nonlinearly generates a continuous open curve in the $x_t$ space and a $q$-dimensional approximation of the original data. The loss of information is again measured by $\xi_t = x_t - \tilde{x}_t$, and analogous to PCA, the functions $\varphi^{(x)}$ and $\varphi^{(u)}$ are selected to minimise $||\xi||$. If $\varphi^{(x)}$ and $\varphi^{(u)}$ are linear, the optimal solution is PCA and if these functions are not linear then we are basically doing nonlinear PCA. In neural network PCA, both mapping (also known as reduction or extraction) and demaping (reconstruction or generation) functions are both approximated by neural networks. PCA is closely related to a particular form of neural network, an autoassociative network method of Kramer (1991) and Scholz et al (2007), which is a neural network whose outputs are its own inputs and the goal is to minimise reconstruction error. An autoassociative neural network has a bottleneck network architecture and consists of a "reduction" network, representing the function $\varphi^{(x)}$ , mapping from m inputs to q outputs, connected directly to a "reconstruction" network, representing the function $\varphi^{(u)}$ , mapping from $q$ inputs to $m$ outputs.

There is also another class of methods that use artificial neural networks, self organizing maps - SOM (Kohonen, 2000), as a means of high-dimensional data projection to a lower-dimensional discrete representation, preserving the locality between data vectors in the original high-dimensional space. However SOM had empirical success especially in one-dimensional cases (see Fort, 2006), the theoretical justification behind the SOM approach is weak. So this work focuses on the autoassociative networks. The aim of next subsections is to cover the key concepts of the methods that use multilayer perceptron neural networks as function approximators (particularly feedforward networks for time series analysis), and neural network PCA method.

### 3.2.3 Neural Networks for Time Series Analysis

When performing time series analysis we would like to characterise how the value of a target variable changes as some predictors are varied. However linear models are adequate to explain many phenomena in the world, most important economic and financial phenomena are complex and nonlinear in nature. In order to explain nonlinear phenomena, different parametric and nonparametric nonlinear regression models have been developed so far.

Parametric nonlinear regression models attempt to characterise the relationship between predictors and response with parametric nonlinear functions. The parameters can take the form of a polynomial, exponential, trigonometric, power, or any other nonlinear function. In other words, in parametric nonlinear models the shape of the functional relationships between the response and the predictors are predetermined. In many situations, that relationship is unknown and nonparametric nonlinear regression models should be used.

In nonparametric models, the shape of the functional relationships between variables can be adjusted to capture unusual or unexpected features of the data. The main types of nonparametric regression models are kernel-based methods, tree-based regression models and artificial neural networks.

Kernel-based methods can be viewed as a nonlinear mapping from inputs into higher dimensional feature space in the hope that the data will be linearly separable or better structured. It measures distances between observations, then predicts new values based on these distances. Best known example are support vector machines (SVMs), introduced by Vapnik (Chervonenkis and Vapnik, 1964, 1974; Vapnik, 1982, 1995), which provide a structured way to use a linear algorithm in a transformed feature space. The key advantage this so-called kernel trick brings is that nonlinear patterns can be found at a reasonable computational cost. Perhaps the biggest limitation of the kernel-based methods lies in choice of the kernel and tuning model parameters.

Tree-based regression models are alternative (nonparametric and nonlinear) approaches to regression that are not based on assumptions of normality and user-specified model statements.These models originated in the 1960s with the development of AID (Automatic Interaction Detection) by Morgan and Sonquist. In the 1980s, statisticians Breiman et al. (1984) developed CART(Classification And Regression Trees). The fundamental idea is to recursively partition the regressors' space in regions (build a tree) until all the subspaces are sufficiently homogeneous in order to estimate the regression function with the sample average (or the specific local model employed) in each region.

Another class of nonlinear models that we focus on in this study are neural networks. These are flexible function forms motivated by the way the brain processes information. neural networks consist of a cascade of simple computational units called neurons, which are highly interconnected. Depending on how they are constructed, neural nets can approximate functions that are generally unknown. However Neural networks can use a variety of topologies, based on the universal approximation theorem, a single hidden layer feedforward network architecture with finite number of neurons can approximate arbitrary well any nonlinear function (poofs have been given by Cybenko (1989), Hornik et al. (1989), White (1990) and Hornik (1991)). For those inetresred in neural network models, see Bishop (1995), Hastie, Tibshirani and Friedman (2009), Teräsvirta, Tjostheim and Granger (2010). Also the following papers provide readers with a statistical approach to neural networks (Varian, 2014; Teräsvirta, Van Dijk and Medeiros, 2005; Kuan and White, 1994).

A neural network is an interesting area of machine learning. It is simultaneously one of the oldest and one of the newest areas. The work on neural networks goes back to the 1940s when researchers tried to build models of the brain. Perceptron, which is an extremely simplified computational model of a biological neuron and a very simple precursor of linear models, goes back to the 50s and people showed amazing performance of the perceptron on a number of problems. Perceptron of course was limited in what it could do, so later on research related to the neural network basically died. It was reborn in the 1980s when researchers figured out how to put multiple perceptrons together into a network and they learnt how network weights with the Backpropagation algorithm worked. Again there was a great deal of excitements because these models finally seemed to be able to solve all kinds of learning problems. At the same time more powerful regression models came along, support vector regressions, so neural networks again fell out of fashion and were shelved. They returned for a second time recently when people finally figured out how to train them reasonable quickly on a massive scale and a big part of that is due to the changes in hardware that have occurred since the 1980s. It is also worth mentioning that there has been a resurgence in the field of artificial neural networks in recent years, known as "Deep neural networks". Deep neural networks use multiple stages of nonlinear computation and have won numerous contests on an array of complex tasks ranging from pattern recognition and machine learning.

In practice, a neural network is an interlinked collection of neurons that the output of some neurons can become inputs to other neurons. A neural network implements a function $y = \Phi(x;\theta)$ ; the 'output' of the network, $y$, is a nonlinear function of the 'inputs' $x$; this function is parametrised by weights and the bias is $\boldsymbol{\theta} = \{w, \beta\}$.

To describe neural networks, we will begin by describing the simplest possible neural network, which is a computational model of a single neuron (known as perceptron). A perceptron follows the "feedforward" model, meaning it takes a set of observed inputs $x_{it} = (x_{1t}, x_{2t}, ..., x_{NT})$, multiplies each of them by their own associated weight $w_i = (w_1, w_2, ..., w_N)$, and sums up the weighted values and also adds a bias $\beta$ (always $+1$) to form a pre-activation $z$. The network then transforms the pre-activation using a nonlinear activation function $\varphi(z)$ (e.g. logistic sigmoid or tanh) to output a final activation $y_t$. So, a single neuron can be formulated as follows:

$$z = \beta + \sum_{i}^{N} x_{it} w_i, \tag{3.9a}$$

$$y_t = \varphi(z) + \varepsilon_t. \tag{3.9b}$$

Single neuron networks with an identity activation function or even a sigmoid function implement linear models, which does not really help us if we want to model nonlinear phenomena. The sigmoid function is almost linear near the mean and has smooth nonlinearity at both extremes. However, by considering the single neuron network to be a basic building block, we can construct more complicated networks, ones that perform powerful nonlinear computations. Instead of a single neuron, we introduce a set of neuron networks. This set of intermediate networks is often referred to as a "hidden" layer, as it does not directly observe input or directly compute the output. By using a hidden layer, we form a multilayered network. A multilayered network with only one hidden layer has two sets of weights: those connecting the inputs to the hidden layer $(w_{ij})$, and those connecting the output of the hidden layer to the output layer$(w_{jk})$.

Multilayer neural networks form compositional functions that map the inputs nonlinearly to outputs. If we associate index *i* with the input layer, index *j* with the hidden layer, and index *k* with the output layer, then an output unit in the network computes an output value $y_t$ given and input $x_t$ via the following compositional function:

$$y_t = \varphi_k\left[\beta_k + \sum_{j}^{L} \varphi_j\left(\beta_j + \sum_{i}^{N} x_{it} w_{ij}\right) w_{jk}\right] + \varepsilon_t \qquad (3.10)$$

where $x_{it}$ is the value of the *i*th input node, which can be a matrix of lagged values of $y_t$ and some exogenous variables. $\varphi_j(.)$ and *j* are activation functions and number of nodes ($L$ neurons) used at the hidden layer. $\varphi_k(.)$ function denotes the output transfer function than can be either linear or a Heaviside step function. As with the single neuron networks, the choice of activation function for the output layer will depend on the task that we would like the network to perform (i.e. categorization or regression). $\beta_j$ and $\beta_k$ are the biases or the weights for the connections between the constant input and the hidden neurons and between the neurons and the output respectively.

From a statistical point of view, formulation of a multilayer feedforward neural network model with more than one hidden layer (i.e. $h$ hidden layers when $h =$

1, ..., $M$) can be generalised to

$$y_t = \varphi_k \left[ \beta_k + \sum_h^M \varphi_h \left( ...\varphi_j \big( \beta_j + \sum_i^N x_{it} w_{ij} \big) \right) w_{hk} \right] + \varepsilon_t, \tag{3.11}$$

Using identity function for the output unit activation function (in conjunction with nonlinear activations amongst the hidden units) allows the network to perform a powerful form of nonlinear regression. So, the network can predict continuous target values using a linear combination of signals that arise from one or more layers of nonlinear transformations of the input. Based on the universal approximation theorem, a single hidden layer feedforward network architecture with a finite number of neurons can approximate arbitrary well any bounded continuous function of $N$ real variables. So, in this study, we have focused on a feedforward network with only one hidden layer. And to show that the neural network models can be seen as a generalisation of linear models, we allowed for direct connections from the input variables to the output layer and we assumed that the output transfer function $\{\varphi_k(.)\}$ is linear, then the model becomes

$$y_t = \beta_k + \sum_i^N x_{it} w_{ik} + \sum_j^L \varphi_j \left( \beta_j + \sum_i^N x_{it} w_{ij} \right) w_{jk} + \varepsilon_t, \tag{3.12}$$

where the first summation represents a linear regression term with a constant. A linear regression term hints the model in a right direction when we know that the data contains a linear component. Moreover, this is more interpretable from a statistical perspective and unraveling a bit of a structure behind the network, which is usually seen merely as a black box. It also has the advantage that, if the problem is essentially linear, the hidden neurons tend to get pruned and we are left with a linear model. Since the functional form of this model is known and we only need to find the number of neurons in hidden layer and to estimate the biases and weights, then feedforward networks are categorized as semiparametric functions. Here, in this approach, we let the data speak for itself as much as possible.

Choosing the appropriate activation function for hidden and output layers is important and depends on the task we would like the network to perform. Activation functions must be differentiable since the learning algorithms such as backpropagation, which determine parameters in neural networks, require the gradient of the activation functions. Most commonly-used activation functions are the identity/linear function, the logistic sigmoid function, and the hyperbolic tangent function. For instance, identity activation function, $\varphi_{linear}(z) = z$ and $y \in (-\infty, \infty)$, is commonly used for the output layer in regression problems. Sigmoidal "S" shape functions like Logistic function, $\varphi_{logistic}(z) = \frac{1}{(1+e^{-z})}$ where $y \in (0, 1)$, and hyperbolic tangent function, $\varphi_{tanh}(z) = \frac{e^{z}-e^{-z}}{e^{z}+e^{-z}}$ where $y \in (-1, 1)$, are other choices for the activation function. What is interesting about these derivatives is that they are either a constant, or are can be defined in terms of the original function. This makes them extremely convenient for efficiently training neural networks, as we can implement the gradient using simple manipulations of the feedforward states of the network. There are several ways that we can think about a derivative of a function. The one that sort of immediately comes to mind is the slope which means how quickly the function is changing around a particular point. Another way to interpret derivative is in relation to the optimum. Suppose that the function is an error function and we are trying to minimise it in some way. What a derivative tells us is whether our current value of parameters is higher or lower than where it should be. In general, estimating the set of network parameters $\theta = \{W, \beta\}$ in a way that minimise the errors that the network makes is known as training/learning neural network. It is equivalent to finding a point in parameter space that makes the height of the error surface small. The error surface gets more and more complicated as we increase the number of layers in the network and the number of units in each hidden layer. In principle, four classes of supervised learning algorithm (desired outputs are available) has been discussed in literature: steepest descent algorithm (also known as backpropagation), Newton's method, Gauss-Newton's algorithm and Levenberg-Marquardt algorithm. It is worth noting that studies in time series and forecasting widely used the conventional feedforward neural network trained with the backpropagation algorithm, however, we explain why the backpropagation is not an efficient algorithm and it converges slowly. Therfore, Levenberg-Marquardt algorithm has been implemented in this study which are fast and have stable convergence. The backpropagation algorithm first introduced by Bryson, Denham and Dreyfus in 1963 and popularised in the field of artificial neural network research by Werbos(1988) and Rumelhart et al. (1986). The goal of backpropagation learning algorithm is to adjust the weights and biases in a way that minimises the network prediction error function.

To solve this problem, the error function's sensitivity to network weights and biases must be quantified based on a gradient descent optimization. Gradient is normally defined as the first order derivative / gradient of the error function with respect to each of the model parameters. This gradient information will give us the direction in parameter space that decreases the height of the error surface. We then take a step in that direction and repeat, iteratively calculating the gradient and taking steps in parameter space.

To explain the math behind backpropagation algorithm, we need to rewrite the neural network formula adding pre-activation signal,$z_j$, for hidden layer and pre-activation signal,$z_k$, for output layer and their corresponding outputs $y_j$ and $y_k$ as these will be used for calculating backpropagated errors and error function gradients.

$$y_t = \varphi_k \overbrace{\left[ \overbrace{\beta_k + \sum_j^L \underbrace{\varphi_j \left( \underbrace{\beta_j + \sum_i^N x_{it} w_{ij}}_{z_j} \right)}_{y_j} w_{jk}}^{z_k} \right]}^{y_k} + \varepsilon_t \tag{3.13}$$

Backpropagation algorithm is based on Widrow-Hoff learning rule (Delta rule) and works like this:

1. Propagat the observed input forward through the network layers toward the outputs. Initial network output/prediction, $y_k$, can be anything, as the initial weights are small random numbers, typically between -1 and 1.

2. Calculate network errors $E$, with respect to a desired target and backpropagate error signal.

   The prediction sum of the squared is a standard way of quantifying error. Given target values and network outputs we can calculate the value of the error function for each setting of weights.

$$E = \frac{1}{2} \sum_{k} (y_k - t_k)^2 \tag{3.14}$$

3. Propagate the errors backward through the network towards the inputs by weighting it by the weights in previous layers and the gradients of the associated activation functions. Unlike the output layer we cannot calculate the error for the neurons in the hidden layer directly (because we do not have target variables), so we backpropagate them from the output layer.

   Output layer parameters directly affect the error function, so the gradients for those parameters can be calculated as follows

$$\frac{\partial E}{\partial w_{jk}} = (y_k - t_k) \frac{\partial}{\partial w_{jk}} (y_k - t_k) \tag{3.15}$$

   Here, based on Chain Rule and having $y_k = \varphi(z_k)$, Thus

$$\frac{\partial E}{\partial w_{jk}} = (y_k - t_k) \frac{\partial}{\partial w_{jk}} y_k \tag{3.16a}$$

$$= (y_k - t_k) \frac{\partial}{\partial w_{jk}} \varphi_k(z_k) \tag{3.16b}$$

$$= (y_k - t_k) \varphi_k'(z_k) \frac{\partial}{\partial w_{jk}} z_k \tag{3.16c}$$

   Recall that $z_k = \beta_j + \sum_j \varphi_j(z_j) w_{jk}$ and $\frac{\partial z_k}{\partial w_{jk}} = \varphi_j(z_j) = y_j$ and again using

Chain Rule we have

$$\frac{\partial E}{\partial w_{jk}} = (y_k - t_k)\varphi_k'(z_k)y_j \tag{3.17}$$

For notation purposes, we define $\delta_k$ to be all the terms that involve index k, and can be interpreted as the network output error after being back-propagated through the output activation function, so we can rewrite the equation above as

$$\frac{\partial E}{\partial w_{jk}} = \delta_k y_j \tag{3.18}$$

Where this equation shows how much each output layer parameter contributes to the error signal.

We follow the same routine for output layer biases, thus the gradient for the biases is

$$\frac{\partial E}{\partial \beta_k} = (y_k - t_k)\varphi_k'(z_k)(1) \quad = \quad \delta_k \tag{3.19}$$

4. Update all the weights and biases in the output layer using the calculated gradients for the parameters. The derivative of the activation function is used to find the rate of change. The weight adjustment is given by

   $$w_{jk}^{new} = w_{jk}^{old} - \eta\, \frac{\partial E}{\partial w_{jk}} \tag{3.20}$$

   Where the constant $\eta$ is the learning rate (step size) and its value falls between zero and one. The direction of search in weight space for the new value of the weights is elected by $\frac{\partial E}{\partial W}$, that shows the sensitivity of the error function to the weights.

5. Having obtained the error for the hidden layer neurons, now using the calculated gradients for the parameters in the hidden layer to update all the

weights in the hidden layer.

The process of calculating the gradients for the hidden layer weights starts just the same but, due to the indirect effect on the output error, the forward and backward signals are used to determine the direction in the parameter space to a move that lowers the network error.

$$\frac{\partial E}{\partial w_{ij}} = \sum_k (y_k - t_k) \frac{\partial}{\partial w_{ij}} y_k \tag{3.21}$$

The sum does not disappear in this case due to the fact that each of the hidden unit outputs affects the state of each output unit. Thus

$$\frac{\partial E}{\partial w_{ij}} = \sum_k (y_k - t_k) \frac{\partial}{\partial w_{ij}} \varphi_k(z_k) \tag{3.22a}$$

$$= \sum_k (y_k - t_k) \varphi_k'(z_k) \frac{\partial}{\partial w_{ij}} z_k \tag{3.22b}$$

Where $z_k$ is indirectly dependent on $w_{ij}$ and by using the Chain Rule we have

$$\frac{\partial z_k}{\partial w_{ij}} = \frac{\partial z_k}{\partial y_j} \frac{\partial y_j}{\partial w_{ij}} \tag{3.23a}$$

$$= \frac{\partial}{\partial y_j} y_j w_{jk} \frac{\partial y_j}{\partial w_{ij}} \tag{3.23b}$$

$$= w_{jk} \frac{\partial y_j}{\partial w_{ij}} \tag{3.23c}$$

$$= w_{jk} \frac{\partial \varphi_j(z_j)}{\partial w_{ij}} \tag{3.23d}$$

$$= w_{jk} \varphi_j'(z_j) \frac{\partial z_j}{\partial w_{ij}} \tag{3.23e}$$

$$= w_{jk} \varphi_j'(z_j) \frac{\partial}{\partial w_{ij}} (\beta_i +_i x_i w_{ij}) \tag{3.23f}$$

$$= w_{jk} \varphi_j'(z_j) x_i \tag{3.23g}$$

Plugging the last two Equations gives

$$\frac{\partial E}{\partial w_{ij}} = \sum_k (y_k - t_k)\varphi'_k(z_k) w_{jk} \varphi'_j(z_j) x_i \tag{3.24a}$$

$$= \varphi'_j(z_j) x_i \sum_k (y_k - t_k)\varphi'_k(z_k) w_{jk} \tag{3.24b}$$

$$= x_i \varphi'_j(z_j) \sum_k \delta_k w_{jk} \tag{3.24c}$$

$$= \delta_j x_i \tag{3.24d}$$

Thus, the gradient for the hidden layer weights is simply the output error signal backpropagated to the hidden layer, then weighted by the input to the hidden layer and this can be interpreted as a proxy for the contribution of the weights to the output error signal. Similar to what we have seen in the output layer, the gradient for the biases is

$$\frac{\partial E}{\partial \beta_i} = \delta_j \tag{3.25}$$

The weight adjustment for hidden layer is given by

$$w_{ij}^{new} = w_{ij}^{old} - \eta \, \frac{\partial E}{\partial w_{ij}} \tag{3.26}$$

6. By repeating iteratively the steps network can be trained in a way that converges to optima. The set of new weights are repeatedly presented to the network until the error value is minimised. Around the optimum point, all the elements of the gradient would be very small, which leads to tiny changes in new weights.

For Newton's method the second-order derivatives of the error function need to be calculated for each elements of gradient vector and the weights update rule is calculated by the following formula:

$$W^{new} = W^{old} - H^{-1} \frac{\partial E}{\partial W} \tag{3.27}$$

Where $H^{-1}$ denotes the inverse of Hessian matrix and is calculated by getting the second-order derivatives of error function. The inverse of Hessian matrix can be too complicated and may be singular.

In Gauss-Newton algorithm, in order to simplify the calculation process, Hessian matrix is approximated by Jacobian matrix and the update rule is defined as:

$$W^{new} = W^{old} - \left(J'J\right)^{-1} \frac{\partial E}{\partial W} \tag{3.28}$$

In this case, it does not require the calculation of second-order derivatives of the error function, but again this approximation of Hessian matrix $J'J$ may not be invertible.

The Levenberg-Madquardt (LM) algorithm algorithm has been proposed by Levenberg (1944) and Marquardt (1963) independently. This learning algorithm is based on a modification of the Gauss-Newton method and provides a numerical solution to iteratively minimise a least square error function without having to compute the Hessian matrix. This algorithm approximates the Hessian matrix and the gradient by calculation of the Jacobian matrix $(J)$, which contains the first derivatives of the model output errors $(e)$ with respect to the network weights and biases. Weight updates rule of the LM algorithm can be formulated as follow:

$$W^{new} = W^{old} - \left(J'J + \mu I\right)^{-1} J'e \tag{3.29}$$

When $\mu$ (combination coefficient) is positive, elements on the diagonal of the approximated Hessian matrix will be larger than zero and guarantee that $(J'J + \mu I)$ is always invertible. It also converges to the Gauss-Newton algorithm when $\mu$ is zero and converges to a gradient descent with a small learning rate when $\mu$ is large enough. The LM algorithm is much faster and more stable than previous learning algorithms since it can switches efficiently between those algorithm during the learning process.

Developments in neural network modelling have further led to the nonlinear generalization of PCA. The next subsection explains how the feed forward neural network can be extended to perform nonlinear PCA (NLPCA).

## 3.3 Nonlinear Factor Model

### 3.3.1 Factor estimation using neural network PCA

The NLPCA basically employs a standard feedforward neural network with four layers of transfer functions mapping from the inputs to the outputs or as explained before one can think of an overall network consisting of two feedforward neural networks (reduction and reconstruction network), which are connected directly and the output of first network is the input for the second one. The layer in the combined network that joins the reduction and reconstruction networks is called the bottleneck layer and contains $q$ neurons, where $(q \ll m)$, giving a reduced dimensionality representation of the input data. By replacing all the layers of transfer functions with the identity function, neural network PCA returns the same result as classical PCA.

Figure.3.3.1 illustrates such a network with three layers of hidden neurons sandwiched between the input layer and the output layer. $m$ dimensional data are compressed to $q$ nonlinear component in the bottleneck reduced layer. This architecture, called an autoassociative network. In this architecture both weight (and bias) parameters of the network and nonlinear component are optimised simultaneously through an unsupervised learning procedure to minimise the reconstruction error. Note when the bottleneck layer has more than one neuron to accomplish the dimension reduction, the residual of the first nonlinear component can be input into the same network to extract the second nonlinear component and so on. Although a better strategy is to penalize the reconstruction error term by adding $(u_1 \cdot u_2)^2$ penalty term to extract a curve surface force the nonlinear components to be uncorrelated with each other (Hsieh, 2004).

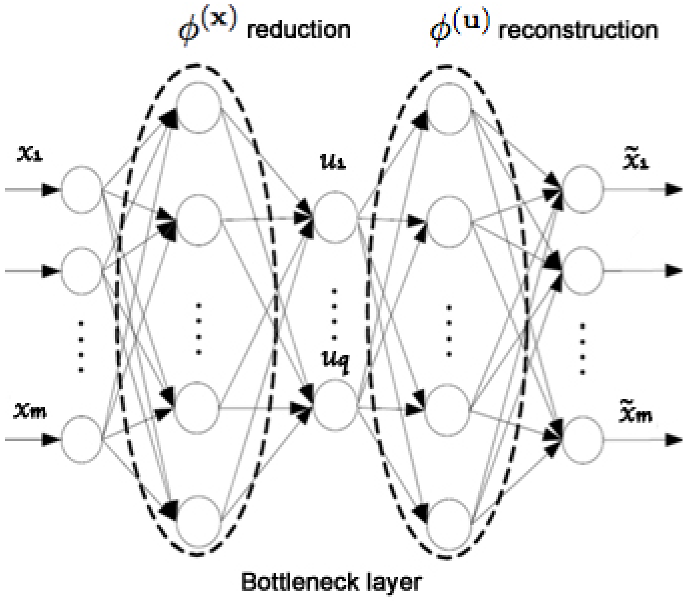

$\phi^{(\mathbf{x})}$ reduction
$\phi^{(\mathbf{u})}$ reconstruction
$x_1$
$x_m$
$u_1$
$u_q$
$\tilde{x}_1$
$\tilde{x}_m$
Bottleneck layer

**Figure 3.3.1:** Schematic diagram of the standard autoassociative neural network architecture for calculating the nonlinear principal component analysis (NLPCA).

Writing $x_t, \tilde{x}_t \in \mathbb{R}^m$ for the input and output data of the overall network, $y_t^{(x)}, y_t^{(u)} \in \mathbb{R}^j$ for the values of the neurons in the input and output hidden layers, $z^{(x)}, z^{(u)}$ for the values of the pre-activation (sum of each input variable multiply by the corresponding weight plus the bias) in the input and output hidden layers and $u \in \mathbb{R}^q$ for the values of the bottleneck neurons, the network transfer functions can be formulated as follow

$$y_t^{(x)} = \varphi_j^{(x)}(z^{(x)}) \tag{3.30a}$$

$$u = \varphi_k^{(x)}(y_t^{(x)}) \tag{3.30b}$$

$$y_t^{(u)} = \varphi_j^{(u)}(z^{(u)}) \tag{3.30c}$$

$$\tilde{x}_t = \varphi_k^{(u)}(y_t^{(u)}) \tag{3.30d}$$

Where the transfer function $\varphi_j^{(x)}$ in Eq.3.30a, maps from the input vector of length $m$, to the first hidden layer (reduction layer). Likewise the second transfer function $\varphi_k^{(x)}$ in Eq.3.30b, which is usually an identity function, maps from the reduction layer to the bottleneck layer to extract nonlinear component values. In the following, $\varphi_j^{(u)}$ in Eq.3.30c, maps from the nonlinear component vector of length $q$, to the final hidden layer (reconstruction layer). $\varphi_k^{(u)}$ in Eq.3.30b, is usually an identity function and produces an approximation of the original data.

Here we briefly describe the model fitting considerations of NLPCA. First of all, note that in contrast to PCA, NLPCA approximation is not an unique analytic solution and must be found through numerical minimisation. In other words, an iterative minimisation procedure carries. Entire samples will be divided into training and validation, and optimisation terminates when either an error over training data stop changing or an error over validation data start increasing. We do not just pick the approximation with the lowest error, but also ensure the approximation is robust. Robustness happens when validation and training share same shape and orientation approximation. It is also worth mentioning that the advantages of NLPCA over the PCA is highly dependent on the data. Noisy and short data can effect on the performance of NLPCA. When underlying relation in data is linear and Gaussian, NLPCA should yield PCA solution. To control the model's complexity and subsequently avoid over fitting, the following strategies can be employed. A model validation based on a missing data approach and regularisation with a weight penalty ensures that NLPCA robustly characterises the underlying structure and it does not describe data in a nonlinear way when the inherent data structure is, in fact, linear (For more details see Christiansen, 2005). A number of runs with random initial weights parameters helps optimization procedure to reach a solution as close as possible to the global minimum in error space. Moreover, Monahan (2001) shows in his work how to measure explained variance of the first nonlinear component in NLPCA, similarly to PCA, by defining residual vectors.

### 3.3.2 Nonlinear forecasting step

Here we compare different linear and nonlinear ways that the forecast equation for the variable of interest $x_{iT+h|T}$ can be formulated in the factor model setting. There are also two different ways to obtain an $h$-step-ahead forecast directly from a long-horizon equation or a sequence of 1-step-ahead forecasts. For instance, Marcellino, Stock and Watson (2006) have shown that the sequential approach outperforms the direct approach (current study focuses on 1-step-ahead forecast of financial return series). Assuming the factors and the loadings are observed, the general formulisation of forecast equation is

$$x_{iT+h|T} = \lambda_i' u_{T+h|T} + \xi_{iT+h|T} \; = \; \chi_{iT+h|T} + \xi_{iT+h|T} \tag{3.31}$$

But in practice, the parameters and the factors are unknown and have to be estimated and the forecast of factors $\hat{u}_{T+h|T}$, idiosyncratic component $\hat{\xi}_{iT+h|T}$, and/or the common component $\hat{\chi}_{iT+h|T}$ are needed to make the forecast feasible. This can be done by an univariate autoregressive model when there is only one factor and a vector-autoregressive when there is more than one factor. A forecast of common components can be obtained by projecting each common component on factors.

Empirical studies, for instance Stock and Watson (2002), Boivin and Ng (2005), Bai and Ng (2006), and Ludvigson and Ng (2009), predict series of interest by using factors ignoring the idiosyncratic component. If this is the case, we only use the information that is common to all predictors, and not idiosyncratic information that might also be relevant for prediction. Here three different types of 1-step-ahead forecasts are considered in order to forecast a series using factors and an idiosyncratic component (given a static approach for factor estimation),

$$\textit{Linear Factor Model (SW)} \begin{cases} \hat{x}_{iT+1|T} = \hat{\beta}_i' \hat{u}_T & (3.32) \\ \hat{x}_{iT+1|T} = \hat{\lambda}_i' u_{T+1|T} & (3.33) \\ \hat{x}_{iT+1|T} = \hat{\lambda}_i' u_{T+1|T} + \hat{\xi}_{iT+1|T} & (3.34) \end{cases}$$

Where $\hat{\beta}_i$ estimated by a linear regression between variable of interest and the lags of factors. One can also add the lags of the variable of interest to the model.

Based on autoassociative neural networks and feedforward neural networks models as explained in the previous section, now we can define the nonlinear extension of factor models. First of all, a factor estimation step in nonlinear setting is done by NLPCA producing $u_t^{(NL)}$.

$$\textit{Nonlinear Factor Model} \begin{cases} \hat{x}_{iT+1|T} = \Phi(\hat{u}_T^{(NL)}) & (3.35) \\ \hat{x}_{iT+1|T} = \varphi_k^{(u^{NL})}(\varphi_j^{(u^{NL})}(\hat{u}_{T+1|T}^{(NL)})) & (3.36) \\ \hat{x}_{iT+1|T} = \varphi_k^{(u^{NL})}(\varphi_j^{(u^{NL})}(\hat{u}_{T+1|T}^{(NL)})) + \hat{\xi}_{iT+1|T}^{(NL)} & (3.37) \end{cases}$$

In 3.35 first we estimate the common factor using NLPCA, then we use them as the input for a feedforward neural network where the targeta variable is the 1-step-ahead forecast of variable of interest. In 3.36 after estimating the factor, we forecast the factor using a feedforward neural network then we propagate a predicted factor through reconstruction layer of NLPA. In 3.37 ,similar to model explain in 3.36, we propagate the forecast value of the factor through a reconstruction layer of NLPA and we also add the forecast value of the corresponding idiosyncratic component obtained from a feedforward neural network. In the next section we illustrate our modelling strategy on a financial empirical example.

## 3.4 Empirical Analysis

### 3.4.1 Data

The data are daily returns of $m = 418$ equities on the S&P 500 index from 04.01.2005 through 31.12.2014, for a total of 2517 observations. We will use the 03.01.2005 - 31.12.2013 period ($T = 2265$ in-sample size) to estimate our forecasting models, and we will use the holdout sample period 02.01.2014 - 31.12.2014 (252 observations) to examine models' out-of-sample forecasting performance. We calculate 1-step (here one day) ahead forecasts of targets ($\hat{x}_{it+1|t}$ return series to be forecast) based on a rolling (moving) estimation window. The estimators of the real value parameters are updated at every time instance. Choice of window size $T$ is a compromise between overly noisy and overly smoothed estimate of parameters and is usually chosen such that $T/N \geq 1$. Our empirical experiments show that different values of T can have an effect on a models' forecasting performance (notably the Random Walk model).

We show the entire series in Figure 3.4.1 , which display clear comovements between the return time series. Heavy tails, leverage effects, nonlinear dependencies and volatility clustering are some other stylized facts that are common to a wide set of high-frequency (e.g., daily) financial return series. Figure 3.4.1 also demonstrate that market become more tightly coupled in volatile periods.

We also zoom in on the 02.01.2014 - 31.12.2014 out-of-sample period in Figure 3.4.2.a in order to reveal the comovement pattern in better detail. Understanding of the comovement between financial equities is crucial for forecasting procedure, therefore we use PCA and NLPCA to produce a parsimonious representation of market correlation (NLPCA is beyond correlation) and demonstrate that a small number of components account for a large proportion of the variability of equities that we consider. In general, we are interested in understanding what factors, lead to movements in an asset's return. One way of identifying these factors, which presumes no knowledge of any factors, and hence is entirely statistical in nature, is via principal component analysis. PCA can be used to identify the underlying statistical factors that explain comovement in asset returns (See Fenn et al. (2011) for a use of principal component analysis and random matrix theory to investigate financial market correlations).

Figure 3.4.2.b visualises the returns' correlation matrix using a heatmap, ranging from black (zero) to white (one). PCA and NLPCA are principally useful when the data under consideration is correlated. In this study PCA and NLPCA are used to analyse the correlated returns of 418 equities.

The first principal component is typically assumed to represent the broad mar-

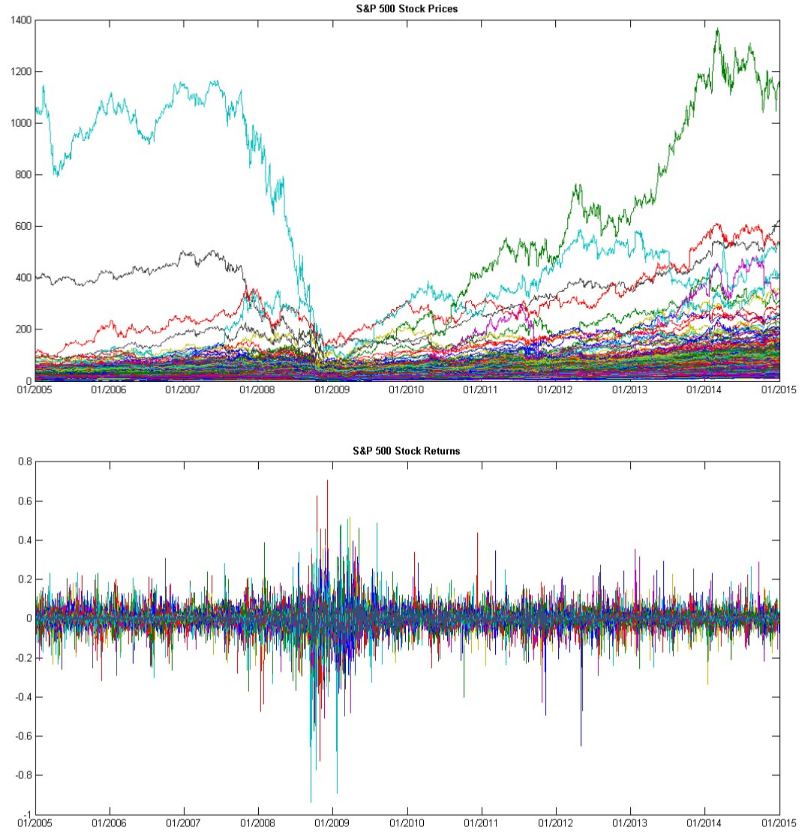


**Figure 3.4.1:** Target time series: prices and returns

ket. The next few are assumed to be the sector/style related factors. The remaining components represent the idiosyncratic properties of stocks. For the given set of equities we can conclude that about 35% of all the variance is because of the broad market factor (systematic risk).

It is worth mentioning that the fraction of the variance in return series explained by the first few linear and nonlinear PCs changes overtime. In Figure 3.4.4, we show how the fraction of the variance explained by first three PCs changes as a function of time for time windows of length 200. Both linear and nonlinear PCs follow the same pattern and in all windows, explained the variance by nonlinear

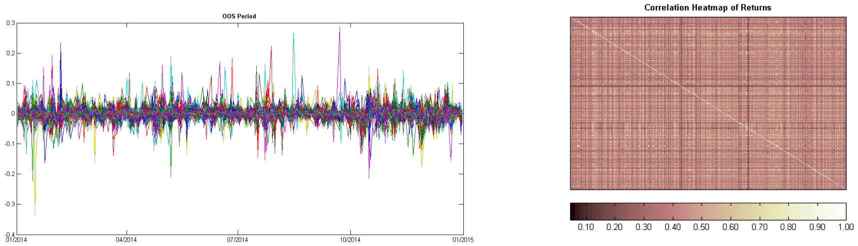


**Figure 3.4.2:** Visualization of comovement between return series; Right: Correlation matrix heatmap, Left: Return series in out-of-sampleperiod

PCA are somewhat higher than the explained variance by linear PCA.

Figure 3.4.4 suggests that market returns have become more correlated during the period of crisis and a sharp increase in the fraction of the variance explained by the first component coincided with major shocks to the financial system (i.e. the collapse of Lehman Brothers in September 2008). It is very instructive for building forecast models and input selection to consider the changes in the variance explained by the first few PCs. We may include the second component (or macro factors) as model predictor when the first component could not account for a large proportion of the variability of data. The square of the first component (a proxy for volatility of the first component) may also improve the forecast performance of the mean returns.

In Figure 3.4.5, we characterize the time-evolving relationships between the different equities by investigating the correlations between the return series and the first three PCs/NLPCs. This figure highlights that there are many time steps at which the correlation between the in the first linear and nonlinear component and return series are significantly large (greater than 0.7). However the correlations decreases between the equity returns and the second and the third components which implies that much of the key structure from the correlation matrices is contained in the first component. Again after Lehman Brothers' collapse, the first component became strongly correlated with nearly all equities implies that many different equities were being driven by the same macroeconomic forces.

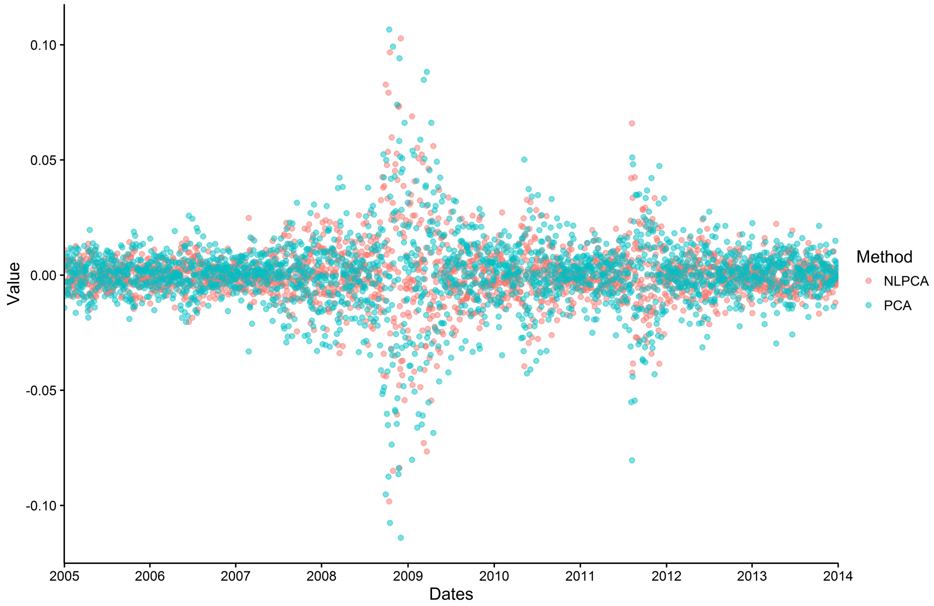


**Figure 3.4.3:** Linear and nonlinear PCA for the first estimation window.

### 3.4.2 Testing for nonlinearity in and between financial return series

Another point this thesis tries to emphasize is that statistical tests to detect nonlinearity in and between time series can assist in terms of choosing the appropriate factor estimation and forecast equation. Before we apply nonlinear models to financial data, it is logical to first ask if the use of such models is justified by the data. To this purpose, we examine the nonlinear dependencies in and between return series and in first principal component series by applying some of the well-known and novel tests introduced in the literature. For instance, applying a nonlinear principal component analysis (NLPCA) for factor extraction step could be justified when the test shows that there is nonlinear dependencies between return series. On the other hand, testing for nonlinearity on the principal component series helps in terms of forecasting them in an appropriate way.

In this chapter we use the following tests to detect nonlinearity in a time series:

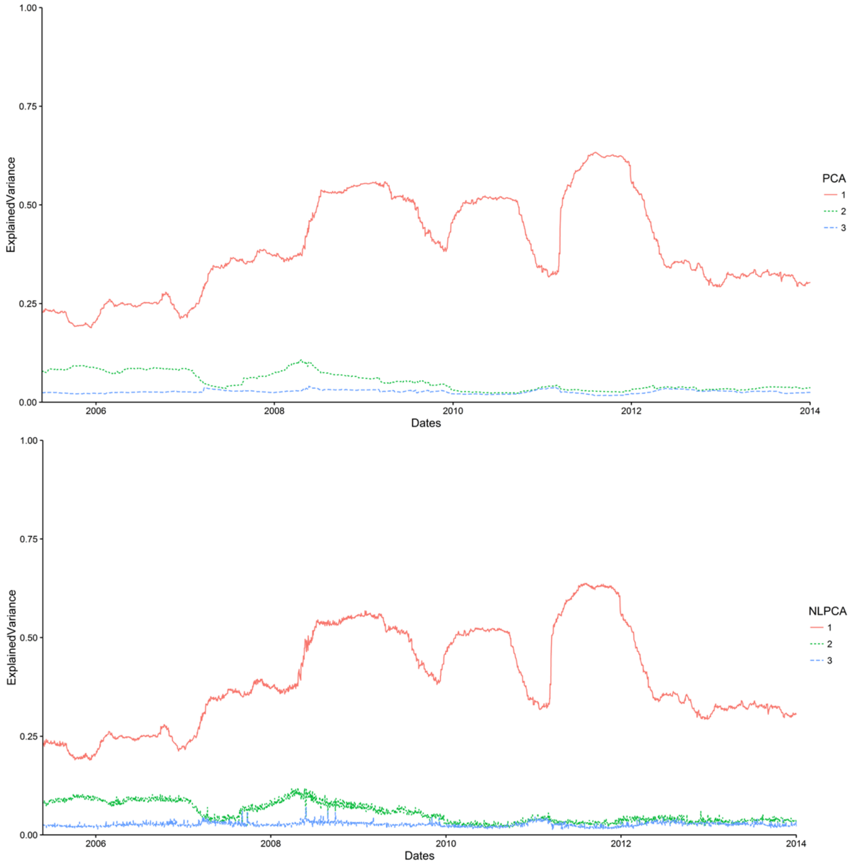


**Figure 3.4.4:** Fraction of the variance in return series explained by the first three PCs

RESET test of Ramsey (1969), that tests for functional form, BDS test of Brock et al. (1987), that tests for the iid series, Teraesvirta NN test of Teraesvirta and Granger (1993), and White NN test of White and Granger (1993), which test for neglected nonlinearity. For the methods of surrogate data, we apply Amplitude Adapted Fourier Transform (AAFT) of Theder et al. (1992), and Iterative Amplitude Adapted Fourier Transform (IAAFT) of Schreiber and Schmitz (1996) algorithms for the generation of surrogate data and Bi-Correlation and Time Reversal as discriminating statistic. See Schreiber and Schmitz (1997) for more details. Finally, we apply Entropy testing for nonlinear serial dependence in a time series based upon the combination of the entropy measure together with resam-

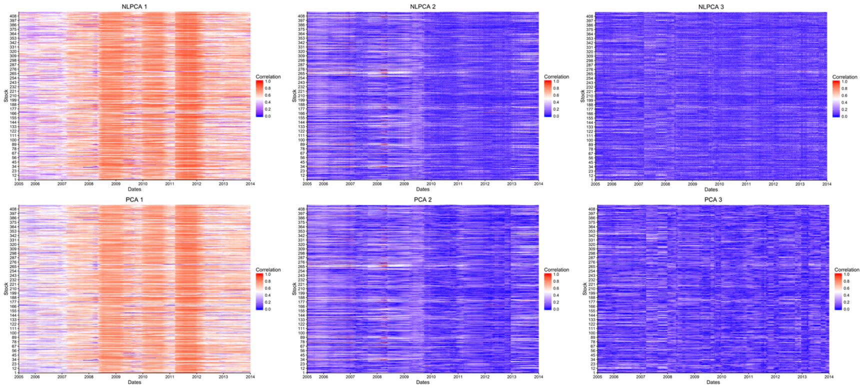


**Figure 3.4.5:** Correlation between each equity and the first three PCs/NLPCs as a function of time. equities are on the vertical axis and the horizontal axis represents the time windows.

pling methods that were recently introduced by Giannerini and Maasoumi (2015). They have proposed a test for identification of nonlinear serial dependence in a time series against the general *H*o of linearity, in contrast to the more widely examined *H*o of "independence".

To shows how the underlying structure within the recorded data can be examined to determine whether a conceptually more demanding NLPCA model is required, a test introduced by Kruger et al. (2008) has been applied in this study. The idea behind this test is explained in the following section.

### 3.4.2.1 Linearity test results

The following subsection presents the results of the linearity tests applied to the sample. The nonlinear dependence in and between return series for 418 companies and their first PCs are examined by applying the tests mentioned above. For the BDS test, an AR model and a best ARMA model is used to remove any serial correlation in the data, and the tests apply to the residual series of the model. It is also difficult to determine if complex real world time series like financial returns behave in a linear or nonlinear fashion. The experimental results that we have got from different nonlinearity tests indicated that the financial time series are rarely pure linear and they consists of nonlinear patterns.

Figure 3.4.6.a, shows the results of the test of nonlinearity based on the Reset test and White NN test. Since Teräsvirta the neural network test for neglected nonlinearity uses a Taylor series expansion of the activation function, the test statistic and the result are quite similar to what we see in the Reset test, therefore we ignore plotting the result of the Teräsvirta test. The evidence against linearity is clear in the series in all lags for Reset test and in first five lags for White NN test. Figure 3.4.6.b, shows the results of the Entropy test of linearity based on surrogate data obtained through the sieve bootstrap to PC series. The evidence against linearity is clear in the series in all lags.

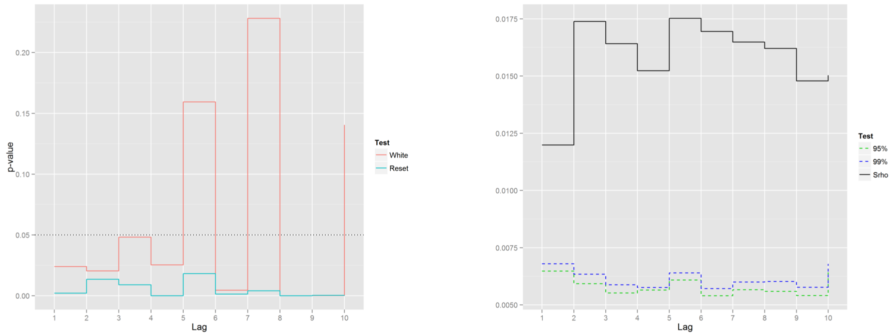


**Figure 3.4.6:** Testing for nonlinearity in PC series. Right: Plot of Entropy test statistic of nonlinearity for the $PC_1$ series. at lags 1:10. The dashed lines indicate the rejection bands at 95% (green/light) and at 99% (blue/dark), Left:Plot of Reset and White test statistics of nonlinearity for the $PC_1$ series. at lags 1:10. y-axis shows the p-value of the test statistics

Figure 3.4.7, shows the results of the linearity test based on the Reset test on three stocks. The evidence against linearity is different in different equities.The major limitation of a linear model is the pre-assumed linear form of the model and therefore, no nonlinear patterns can be captured. A pure nonlinear model is not also adequate to handle fairly both linear and nonlinear patterns, especially when the linear component is superior to the nonlinear component. The performance of the linear model and neural networks is not robust when the time series contains complex linear and nonlinear patterns. To overcome this issue, we add a skip-layer to a neural network and then we penalise the loss function to guarantee that model will reduce to a linear model if no nonlinearity exists in the data.

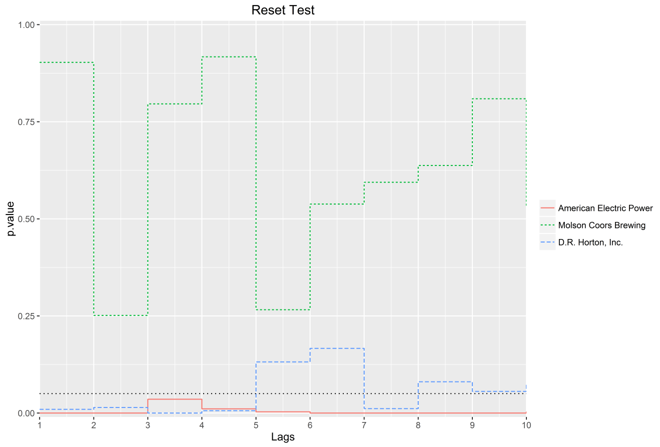


**Figure 3.4.7:** Plot of Reset test statistics of nonlinearity for three different equities

### 3.4.2.2 Testing for nonlinearity between time series

In this section, we examine the nonlinear dependencies between return series to show how the underlying structure within the recorded data can be examined to determine whether a conceptually more demanding NLPCA model is required. The test applied in this section introduced by Kruger et al. (2008) and it is designed to detect nonlinearities based on nonlinear principal component analysis and between time series. This test divides the entire data into several disjunct regions through prior knowledge of the process or by direct analysis of the recorded data, and it takes advantage of the residual variance in each of the regions. The test then examines whether the sum of the residual variances or more precisely its PCA equivalent, the sum of the discarded eigenvalues, are significantly different among these regions. For this, the accuracy bounds are calculated for the sum of the discarded eigenvalues of one of the regions taking advantage of the confidence limits for the mean and variances, and hence each value of the correlation matrix. The sum of the residual variances for other regions are then benchmarked against this accuracy bound. If at least one of the sum of the discarded eigenvalues lies outside this bound, the null hypothesis is rejected and NLPCA approaches should be considered. The power of the test is also increased utilizing the principal of cross-validation and thus calculating the accuracy bounds for each region and comparing sum of the discarded eigenvalues of the other regions with it.

In our case, we divided the sample into 4 disjunct regions. The accuracy bounds for each disjuct region and also sum of the discarded eigenvalues were computed. These bounds were based on thresholds for each element of the correlation matrix corresponding to confidence level of 95Note that the processes were normalized with respect to the mean and variance of the regions for which the accuracy bounds were computed. After performing the test, the third region falls out of accuracy bounds of the first region. The third region contains the data starting from 6 July 2009 until 29 September 2011. One can speculate that the nonlinearity could be traced back to the aftermath of the 2008 financial crisis. Fig 3.4.8 Benchmarking of the residual variances against accuracy bounds of each disjunct region and illustrates that the relationship between recorded financial data is nonlinear.

### 3.4.3 Proposed and competing models

In this study we attempt to answer the question of whether it is possible to forecast with a large panel of predictors, while considering nonlinear dynamics in a high-dimensional dataset and to show how a model with such features can improve financial forecasting accuracy. We proposed a nonlinear high-dimensional forecasting model that is able to handle high-dimensionality and nonlinearity by applying a neural network based principal component analysis to estimate common factors from a large panel of data and nonlinearly forecasts the series of interest uses the factor estimates with a feedforward neural network. We compare our proposed nonlinear factor model NLFM with several competing models and benchmarks. As a benchmark for comparison, we use the sample mean of $x_t$ in the in-sample

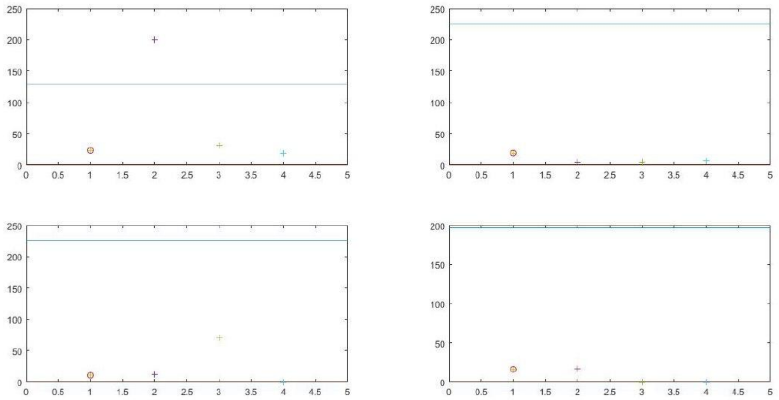


**Figure 3.4.8:** Accuracy bounds and residual variances. Sample is divided into smaller disjunct regions; and accuracy bounds are determined for the sum of the discarded eigenvalues of each region. If this sum is within the accuracy bounds for each region, the process is assumed to be linear. Conversely, if at least one of these sums is outside, the process is assumed to be nonlinear. As the figure illustrates, the recorded financial data is nonlinear.

window as the 1-step ahead forecast. This corresponds to assuming that the log daily price of stocks follows a random walk with drift and it is almost equivalent to the "zero forecast" when the in-sample window is large enough. Furthermore, a buy-and-hold strategy in the market portfolio has been considered as another benchmark. To understand whether considering the comovemen of returns improves forecast accuracy we compare univariate forecasting methods with high-dimensional forecasting methods. The univariate models that we estimate here can be categorised in two groups, linear univariate models: specifically, AR(1) model and best ARMA(p,q) model, and nonlinear univariate model: feedforward neural networks. The high-dimensinal models are also in two groups, linear static factor models and nonlinear neural network factor models. Since the information inside an idiosyncratic component might also be relevant for prediction, we compare the forecast accuracy of linear/nonlinear factor models considering an idiosyncratic component to forecast a return series with models, ignoring the idiosyncratic component.

### 3.4.4 Forecast evaluation

Since the predictability of financial returns is used to guide decisions, forecast accuracy and comparing the forecastability of alternative models is of obvious importance to finance managers to discriminate among forecast models and choose an adequate model that is inextricably linked to the predictive performance of the model. Forecast error can be defined simply as $\hat{e}_{it+1} = x_{it+1} - \hat{x}_{it+1}$, when $x_{t+1}$ denotes the observation at time $t+1$ and $\hat{x}_{it+1}$ denotes the forecast of $x_{it+1}$. Among various models, the one that has the minimal forecast error is often deemed optimal. However, the model with minimum forecast error does not necessarily guarantee profit maximisation, which is the ultimate objective of making predictions of returns in financial markets. There are some forecast accuracy measures from the time series approach, of which the most commonly used are scale-dependent measures based on the absolute errors or squared errors like the Mean Square Error (MSE). These are useful and generally straightforward especially when comparing different models on the same set of data. Root Mean Square Error (RMSE) is often preferred as it is on the same scale as the data. As financial models with minimal direct measures, such as RMSE, do not necessarily guarantee maximised investment profits, an alternative approach, which explicitly addresses these issues, is to evaluate the merits of the alternative financial forecasting models based on indirect measures such as performance of a trading strategy. Armstrong and Collopy (1992), Pesaran and Timmermann (1995, 2000), Granger and Pesaran (2000) and Engle and Colacito (2006) argue that a forecast evaluation criterion should be related to decision making and judge predictability of financial returns in terms of portfolio simulation. More specifically, a trading (portfolio) simulation approach assumes that all competing models are applied with stock market virtual investment decisions, and out-of-sample portfolio performances are used to evaluate the predictability of alternative models. The accuracy measures that this study employs to discriminate between a competing set of forecasting models are from both time series approach (the out-of-sample RMSE and the out-of-sample coefficient of determination ($R_i^2 = 1 - \frac{(\hat{e}_i - \bar{\hat{e}}_i)'(\hat{e}_i - \bar{\hat{e}}_i)}{(x_i - \bar{x}_i)'(x_i - \bar{x}_i)}$)) and trading simulation approach (the out-of-sample hit rate - also called the probability of detection given by $h_i = \frac{\sum_{t=T_1+1}^{T_2} sign(x_{it}\ \hat{x}_{it|t-1})}{T_2 - T_1}$ - and the portfolio performance during the out-of-sample period). Moreover, to test the equality of forecast accuracy between competing models, we employ the Diebold Mariano (DM) test, which can easily be applied to a wide variety of criteria including RMSE (see Diebold and Mariano (1995) and Harvey, Leybourne, and Newbold (1997)). The portfolio construction is based on passive equally weighted ($1/N$) portfolios with short sell that are known to be a very stringent benchmarks and that many optimization models fail to outperform (see DeMiguel et al., 2009). We compute the portfolio's out-of-sample return and volatility as well as the Sharpe ratio[1]. It worth mentioning that the hit rate shows the proportion of correctly predicted signs of returns and it is sensitive only to missed events rather than false forecasts and should be interpreted with care, but, in contrast to hit rate, portfolio simulation can count in false forecasts.

### 3.4.5 Results

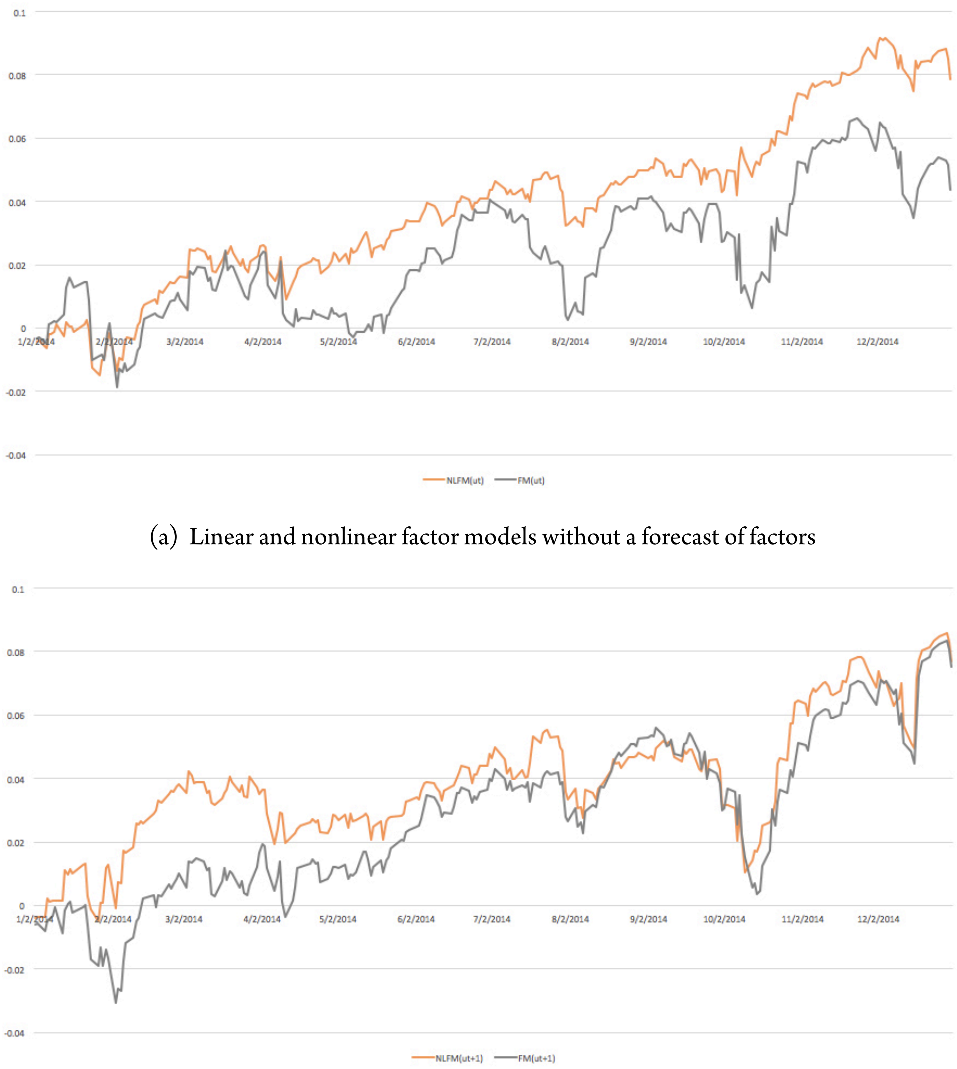
This section compares our proposed nonlinear factor model (NLFM) with several competing models and benchmarks. As mentioned in previous section, three different types of 1-step-ahead forecast are considered in this study in order to forecast return series using factors and idiosyncratic components. First we compare linear/nonlinear factor models estimating a regression between the return series and the lags of estimated factor[2] ($\hat{x}_{iT+1|T} = \hat{\beta}_i^{'} \hat{u}_T$ shown by FM$(u_t)$ , $\hat{x}_{iT+1|T} = \Phi(\hat{u}_T^{(NL)})$ shown by NLFM$(u_t)$). Then we compare linear/nonlinear factor models multiplying estimated loadings (and weights in nonlinear setting) by 1-step-ahead forecast of factors($\hat{x}_{iT+1|T} = \hat{\lambda}_i^{'} \hat{u}_{T+1|T}$ shown by FM$(u_{t+1})$ , $\hat{x}_{iT+1|T} = \varphi_k^{(u^{NL})}(\varphi_j^{(u^{NL})}(\hat{u}_{T+1|T}^{(NL)}))$ shown by NLFM$(u_{t+1})$).

Figure 3.4.9.a illustrates portfolio returns during an out-of-sample period for

[1] In this work we have also constructed portfolios including only one stock for each competing model. In general, trading simulation results from these portfolios were aligned with the results we get from portfolio simulation, including all 418 stocks and other criterion.

[2] Only the first principal component has been chosen as the predictor in the models.

a linear factor model FM$(u_t)$ and a nonlinear factor model NLFM$(u_t)$ without a forecast of factors. Also Figure 3.4.9.b illustrates portfolio returns during an out-of-sample period for the linear factor model FM$(u_{t+1})$ and the nonlinear factor model NLFM$(u_{t+1})$ with a forecast of factors. In both figures nonlinear factor models significantly outperform linear factor models in terms of portfolio return and Sharpe ratio. Sharpe ratio is a measure for calculating risk-adjusted return and a portfolio with a greater Sharpe ratio gives more returns for the same risk. If a portfolio with lower Sharpe ratio has returned better over a time period than another portfolio with a comparatively higher ratio, it means that the risk of losing by investing in the former fund will be higher. (see table 3.4.1)



(a) Linear and nonlinear factor models without a forecast of factors

(b) Linear and nonlinear factor models with a forecast of factors

**Figure 3.4.9:** Comparison of linear and nonlinear factor models based on the performance of the portfolio return during an out-of-sample period.

Figure 3.4.10 illustrates portfolio returns during an out-of-sample period for all four linear/nonlinear factor models with and without a forecast of factors. Between two nonlinear factor models, one without a forecast of factor could outperform the alternative model on the contrary, however, a linear factor model with a forecast of factor outperforms the alternative. So for the rest of the chapter we keep FM$(u_{t+1})$ as representative of linear factor models and NLFM$(u_t)$ as representative of nonlinear factor models. One of the preferential property that a nonlinear factor model has is its low fluctuating portfolio performance during an out-of-sample period, which is appealing for investors.

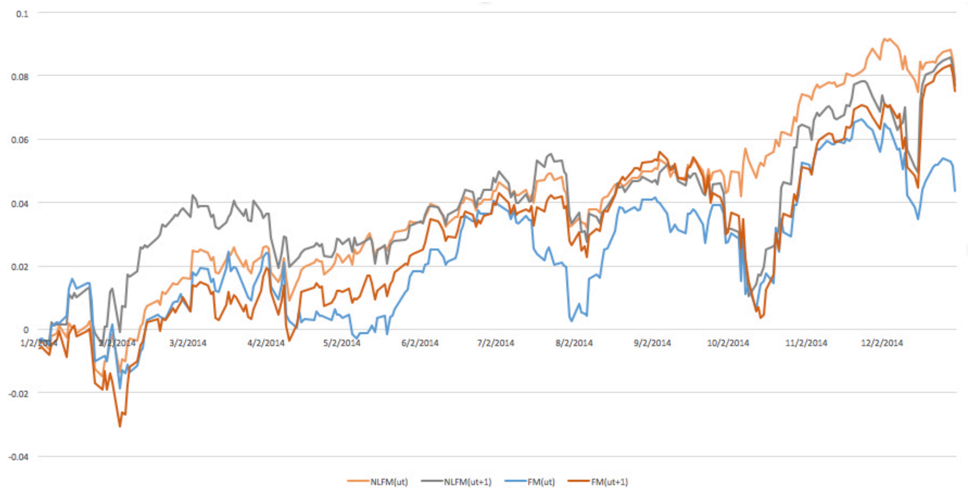


**Figure 3.4.10:** Comparison of linear and nonlinear factor models based on the performance of the portfolio simulation during an out-of-sample period.

**Table 3.4.1**

| Portfolio | Return | Sharp Ratio |
|---|---|---|
| FM$(u_t)$ | 4.35% | 9.1770 |
| FM$(u_{t+1})$ | 7.51% | 17.4927 |
| NLFM$(u_t)$ | 7.87% | 25.0019 |
| NLFM$(u_{t+1})$ | 7.41% | 18.8963 |

Based on DM test 55 out of 418 predictions were significantly different between FM$(u_t)$ and FM$(u_{t+1})$, which in 48 cases FM$(u_{t+1})$ returned lower forecast RMSE. In 280 stocks FM$(u_{t+1})$ showed higher $R^2$ than the alternative model. In a nonlinear factor model 74 out of 418 prediction were significantly different which for

NLFM($u_t$) 55 times could bit NLFM($u_{t+1}$) in terms of RMSE. NLFM($u_t$) also showed in 310 stocks a higher $R^2$ than the alternative model. We also used the DM test to compare nonlinear factor models with linear factor models. In 77 out of 418 the predictions were significantly different between two models, in 61 cases of which NLFM returned a lower forecast RMSE.

Two benchmark forecasts that we compare our models with are random walk with drift (RW) and a buy-and-hold strategy in the market portfolio (S&P 500 Index). We also compare the NLFM model with AR(1) later in this section.

Figure **??** illustrates linear and nonlinear factor models against a buy-and-hold investment strategy on S&P 500 index. The portfolio return for this strategy at the end of the out-of-sample period is equal to 4.68% and the corresponding Sharpe ratio is 15.0060, which means that both linear and nonlinear factor models outperform the market.

Figure **??** shows linear and nonlinear factor models against Random Walk. However RW leads to a higher return (10.42%) at the end of out-of-sample period, the Sharpe ratio corresponding to this model (17.4608) is lower than nonlinear factor models.

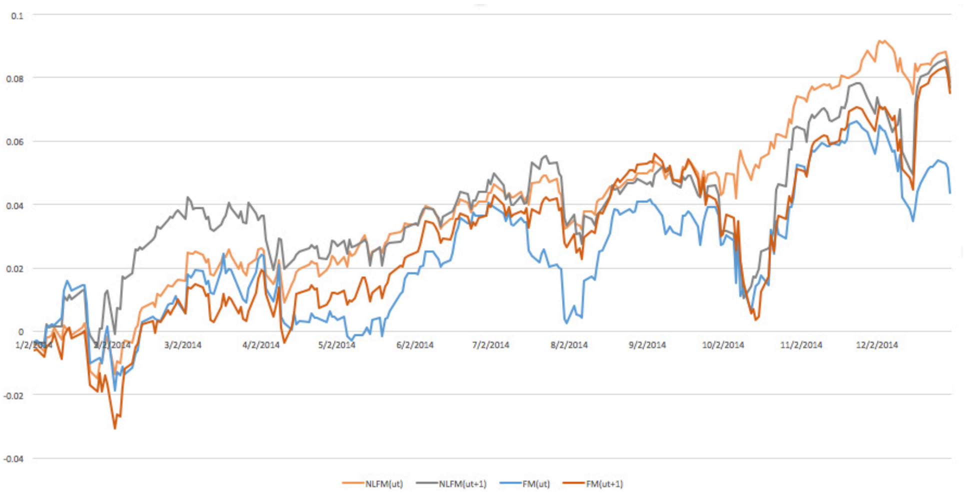


**Figure 3.4.11:** Comparison of linear and nonlinear factor models against an investment on S&P 500 index based on the performance of the portfolio simulation.

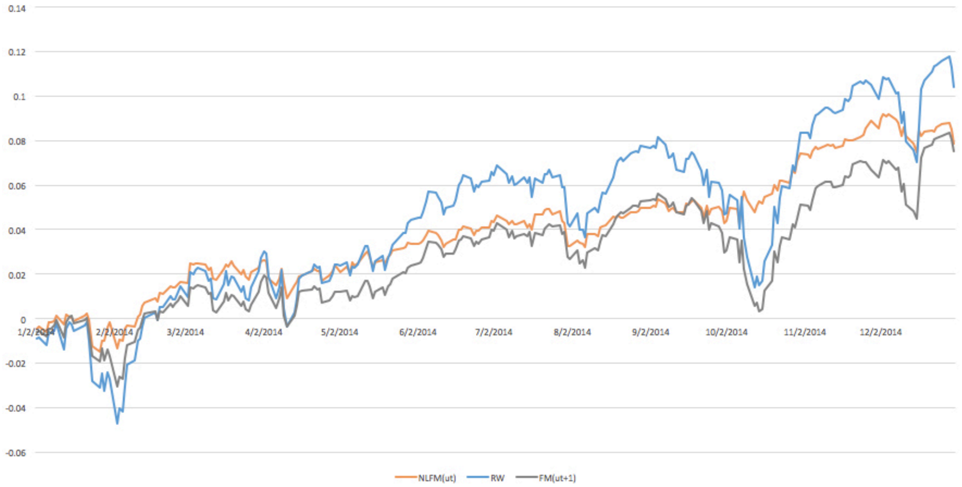


**Figure 3.4.12:** Comparison of linear and nonlinear factor model against Random walk based on the performance of the portfolio simulation.

The result shows that a nonlinear factor model significantly outperforms the linear factor model, nevertheless the result does not tell us whether the nonlinear factor estimation caused this improvement or whether it was the nonlinear forecast equation. So we decided to also compare our linear/nonlinear factor models with two competing models; a model with nonlinear factor estimation still with a linear forecast equation, and a model with linear factor estimation benefits with a nonlinear forecast equation.

Figure 3.4.13 first shows that neither a model with nonlinear factor estimation nor a model with a nonlinear forecast equation could outperform a proposed nonlinear factor model, however they could outperform the linear factor model. On the other hand, the result shows that factor estimation in a nonlinear way is more important than having a nonlinear forecast equation on linear components. (see table 3.4.2)

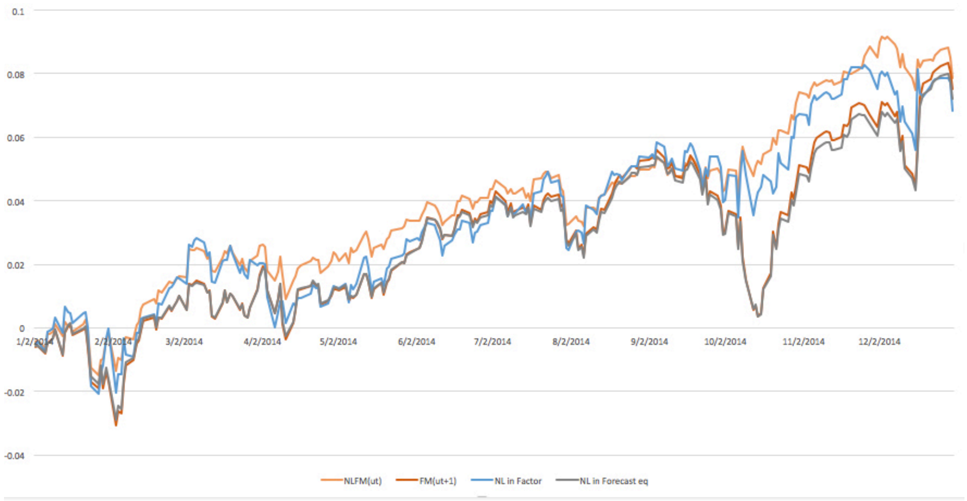


**Figure 3.4.13:** Comparison of linear and nonlinear factor models, and the models with only one nonlinear step based on the performance of the portfolio simulation

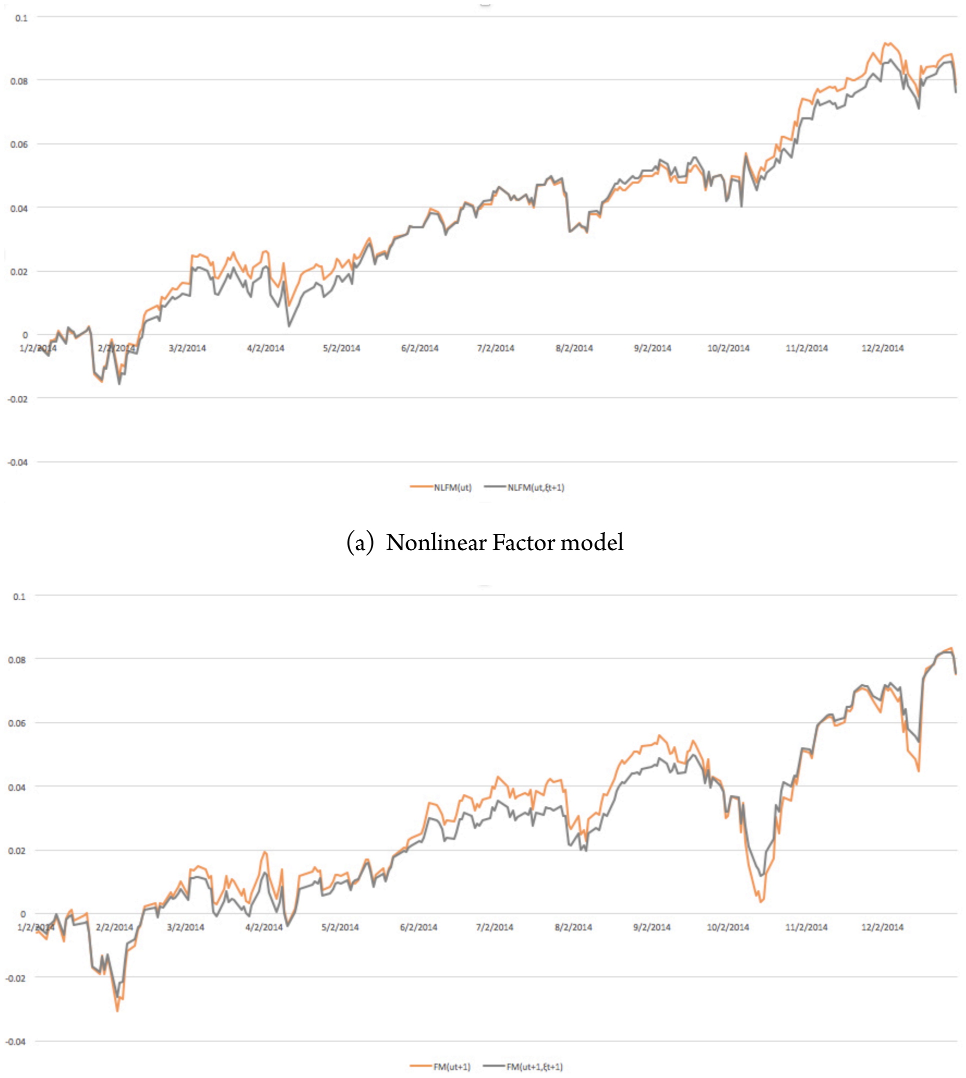
**Table 3.4.2**

| Portfolio | Return | Sharp Ratio |
|---|---|---|
| Linear FM | 7.51% | 17.4927 |
| Nonlinear FM | 7.87% | 25.0019 |
| Nonlinear in factor estimation step | 7.61% | 18.6259 |
| Nonlinear in forecast equation step | 6.83% | 17.8577 |

The fraction of the variance in financial series explained by the first few components are much less than the fraction of the variance in the macroeconomic series explained by the first few linear and nonlinear components. It indicates that the idiosyncratic component is not just a noise and there is information inside idiosyncratic components that might also be relevant for financial forecasting. For this purpose, we compare the forecast accuracy of linear/nonlinear factor models considering idiosyncratic components ($\hat{x}_{iT+1|T} = \hat{\lambda}_i' u_{T+1|T} + \hat{\xi}_{iT+1|T}$, $\hat{x}_{iT+1|T} = \varphi_k^{(u^{NL})}(\varphi_j^{(u^{NL})}(\hat{u}_{T+1|T}^{(NL)})) + \hat{\xi}_{iT+1|T}^{(NL)}$) to forecast return series with models ignoring the idiosyncratic component[3].

Figure 3.4.15 illustrates the effect of adding idiosyncratic components in forecast models. By looking at the individual RMSEs and hit rates, we will see improvements in many stocks, however, in terms of Portfolio simulation, it was more effective to improve a linear factor model. It can be interpreted that NLPCA could extract more information from data and less information remained in the idiosyncratic component. (See table 3.4.3)

[3] We forecast the idiosyncratic component with an AR(1) process in the linear factor model and a feedforward neural network in the nonlinear factor model



(a) Nonlinear Factor model

(b) Linear Factor model

**Figure 3.4.14:** Considering idiosyncratic component to forecast return series and its effect on portfolio return

**Table 3.4.3**

| Portfolio | Return | Sharp Ratio |
|---|---:|---:|
| Linear FM | 7.51% | 17.4927 |
| Nonlinear FM | 7.87% | 25.0019 |
| Linear FM + $\hat{\xi}_{iT+1\|T}$ | 7.55% | 22.4096 |
| Nonlinear FM + $\hat{\xi}_{iT+1\|T}$ | 7.61% | 23.5331 |

A hybrid forecast model is also proposed in this chapter to show how linear factor models can be improved. In this approach, first we estimate the linear model and we collect the residuals obtained from the fitted model. Then we let neural network to model the residuals which contain information about the nonlinearity. In theory, the hybrid model can be an effective tool with a superior forecast when both linear model and neural network model are specified well and are suboptimal. But, in practice we are combining two model specification errors and it is assumed that the time series has only a linear structure in the first step and a nonlinear structure in the second step which can be imprecise.

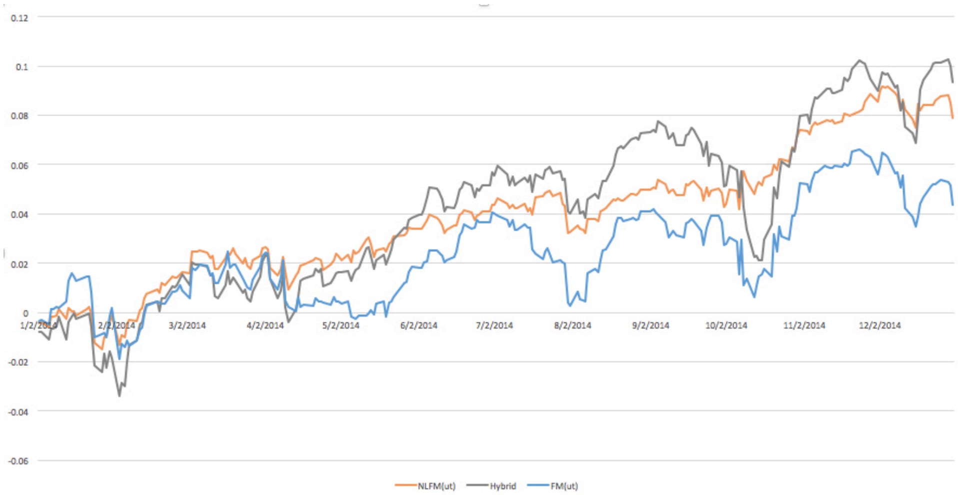
Figure 3.4.15 illustrates linear and nonlinear factor models against the hybrid model. The hybrid model outperforms linear factor model however it can not outperform nonlinear factor model in terms of Sharpe ratio. Then again, if a fund with lower Sharpe ratio has returned better over a time period than another fund with a comparatively higher ratio, it means that the risk of losing by investing in the former fund will be higher. (see table 3.4.4)

To understand whether considering the covariance structure of returns improve forecast accuracy we compare linear and nonlinear univariate forecasting methods with factor models as well.

Figure 3.4.16.a shows how trading strategies based on AR(1) model noticeably outperform best ARMA(p,q) model[4], in terms of portfolio return during out-of-sample period. Portfolio return and Sharpe ratio calculated for the best ARMA

[4] ARMA model is obtained from a minimization of the penalized AICc and MLE, but it seems overfit.

**Table 3.4.4**

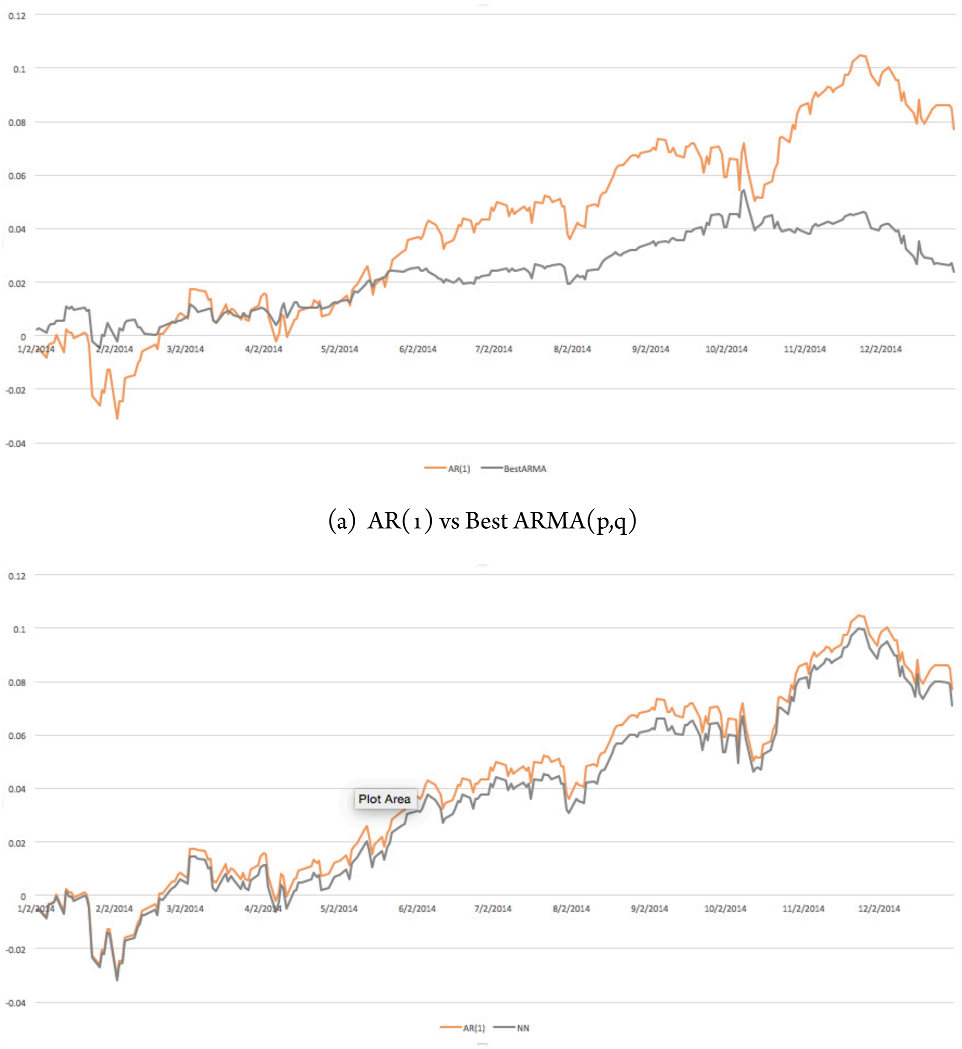


| Portfolio | Return | Sharp Ratio |
|---|---|---|
| Linear FM | 7.51% | 17.4927 |
| Nonlinear FM | 7.87% | 25.0019 |
| Hybrid model | 9.32% | 19.2152 |

model is %2.3 and 10.3660 against %7.7 and 18.3059 for AR(1) model. Therefore an AR(1) model is chosen as representative of linear univariate models. AR(1) model is more successful than a best ARMA model based on all other criterion we used for forecast evaluation.

Figure 3.4.16.b Complares AR(1) model with a feedforward neural network which we used to forecast return series based on their first lags. Portfolio return follows almost the same pattern during an out-of-sample period. Portfolio return and Sharpe ratio calculated for the neural network model is %7.3 and 18.4610. It is worth mentioning that financial series follow a different level of nonlinearity and testing for nonlinearity can help us to choose the appropriate forecast model. In our study, in %86[5] when test shows that time series follows a linear/nonlinear pattern, a linear/nonlinear model returns a better result. However figure 3.4.16.b illustrate portfolios constructed with linear and nonlinear models separately.

[5] This number is based on linearity test results on 418 stocks and AR(1) and NN forecasts.



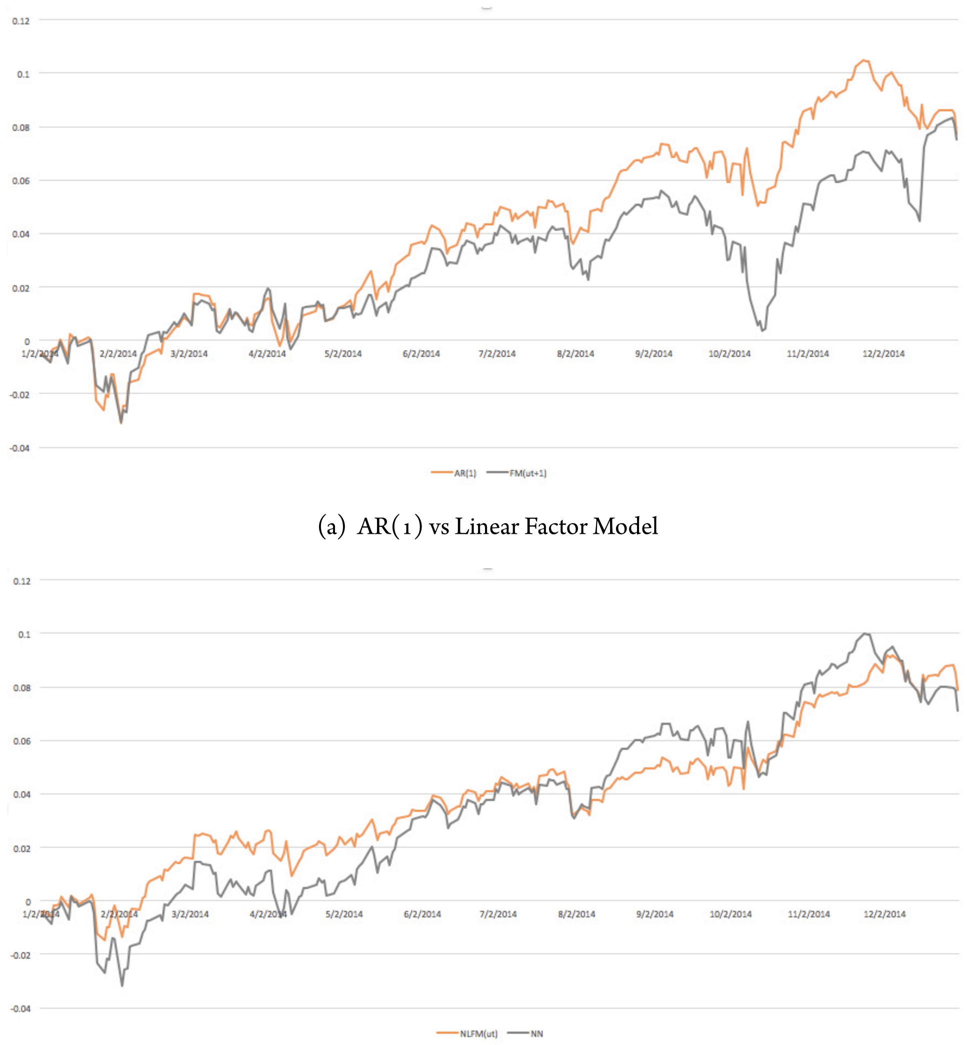
(a) AR(1) vs Best ARMA(p,q)

(b) AR(1) vs Neural Network

**Figure 3.4.16:** Comparison of linear and nonlinear univariate models based on the performance of the portfolio return during the out-of-sampleperiod.

Figure 3.4.17.a illustrates the linear univaritae model , AR(1), against the linear factor model. It seems as having the first lag of each stock return as the predictor returns better results in terms of portfolio simulation. However the DM test result for this two competing models show that in only 35 stocks the RMSEs are significantly different and 24 out of 35 stocks have a lower RMSE when a linear factor model is used for forecasting. Furthermore 235 out of 418 stocks have a higher $R^2$ when a linear factor model is used for forecasting. AR(1) outperforms FM in terms of hit rate.

Figure 3.4.17.b illustrates a neural network as the representative of nonlinear univaritae models , against the nonlinear factor model. Here nonlinear factor models notably beat univariate model. DM test result for these two competing models shows that in only 40 stocks the RMSEs are significantly different and 36 out of 40 stocks has lower RMSE when NLFM is used. Furthermore, 310 out of 418 stocks has higher $R^2$ when NLFM is used. Another notable issue here is the comparison of NLFM and AR(1) which can be seen as a benchmark in financial forecasting. The NLFM model count also beat the AR(1) model in terms of portfolio simulation criterion. (See Table 3.4.5)



(a) AR(1) vs Linear Factor Model

(b) Neural Network vs Nonlinear Factor model

**Figure 3.4.17:** Comparison of linear and nonlinear univariate and factor models based on the performance of the portfolio return during an out-of-sample period.

Based on the univariate model one can build a forecast model by considering both common factors and the lags of target variable as the predictors. Also macroeconomic variables can be added to the model.

**Table 3.4.5**

| Portfolio | Return | Sharp Ratio |
|---|---|---|
| AR(1) | 7.55% | 22.4096 |
| Linear FM | 7.51% | 17.4927 |
| Neural Network | 7.12% | 17.4610 |
| Nonlinear FM | 7.87% | 25.0019 |

## 3.5 Conclusion

Forecasting with many predictors has received a good deal of attention in recent years. The most common approach for forecasting with many predictors is to linearly extract common factors, and use them as predictors in a linear forecast equation. Such methods are not efficient because they disregard nonlinear dynamics among predictors and between common factors and the target variable. In this chapter, our focus was on answering the question whether "it is possible to forecast with a high-dimensional panel of predictors while considering nonlinear dynamic among variables?"

To answer this question, we proposed a nonlinear generalization of the statistical factor model which at the first step (factor estimation) employs an autoassociative neural network to estimate nonlinear factors from predictors and at the second step (forecasting equation) applies a nonlinear function -feedforward neural network- on estimated factors to predict the dependent variable. This model can go beyond the covariance structure analysis. The method also behave noticeably better in the empirical analysis. The empirical application to forecasting daily returns of equities on the S&P 500 index from 2005 to 2014 provide support for the out-of-sample forecasting ability of this model vis-à-vis exist competing approaches both in terms of the time series approach and the trading simulation approach. We showed how adopting statistical tests to detect nonlinearity in and between time series helps for model selection. Our empirical results encourage further research toward other possible applications of nonlinear factor models.

*Risk is like fire: If controlled it will help you; if uncontrolled it will rise up and destroy you.*

Theodore Roosevelt

# 4

# Econometric Modeling of Systemic Risk: Going Beyond Pairwise Comparison and Allowing for Nonlinearity

## 4.1 Introduction

Understanding the interconnection between the financial institutions is of great importance. In principle, there are two main approaches to measure such interconnections between financial institutions in the literature. One is based on a mathematical model of financial market participant relations as a graph using a combination of information extracted from financial statements like the market value of liabilities of counterparties, and the other one that is also the approach of this work is based on statistical analysis of financial series.

Most of the existing approaches in the literature are built on pairwise comparison or assuming linear relationship between the time series. For instance the authors in Billio et al. (2012) propose several measures of systemic risk to capture the connections between the daily returns of different financial institutions (hedge funds, banks, brokers, and insurance companies) based on Granger-causality tests. They propose a definition of systemic risk as "any set of circumstances that threatens the stability of or public confidence in the financial system". This definition implies that the risk of such events is unlikely to be captured by any single metric that ignores the connections between the financial institutions. Billio et al. (2012) uses principle component analysis to estimate the number and importance of common factors driving the returns of financial institutions, and it uses *pairwise* Granger-causality tests to identify the network of Granger-causal relations among those institutions.

Another related work is Diebold and Yılmaz (2014). In this work, the authors propose a connectedness measure based on generalized variance decomposition (GVD) and consequently, define a weighted, directed network. However, the measure introduced in this work is limited to linear dynamical systems, more precisely, data-generating processes (DGPs). Moreover, as we will discuss later in Section 4.3.2, their measure suffers from disregarding the entire network akin to pairwise analysis commonly used in the literature.

In Barigozzi and Hallin (2016), the authors focus on one particular network structure: the long-run variance decomposition network (LVDN). Similar to Diebold and Yılmaz (2014), the LVDN defines a weighted and directed graph where the weight that is associated with edge $(i, j)$ represents the proportion of h-step-ahead forecast error variance of variable $i$ which is accounted for by the innovations in variable $j$. LVDNs are characterized by the infinite vector moving average (VMA) that limits them to linear systems.

Connectedness measures based on correlation remain widespread, however, they measure only pairwise association and are mainly used for linear Gaussian dynamics. This makes them of limited value in financial-market contexts. Different approaches have been developed to relax these conditions. For example, equicorrelation approach in Engle and Kelly (2012) uses average correlations across all pairs. The CoVaR approach of Adrian and Brunnermeier (2008) measures the value-at-risk (VaR) of financial institutions conditional on other institutions experiencing financial distressand. The marginal expected shortfall (MES) approach of Acharya et al. (2010) measures the expected loss to each financial institution conditional on the entire set of institutions, poor performance. Although these measures rely less on linear Gaussian methods and are certainly of interest, they measure different things, and a general framework that can be used to capture the connectedness in different networks remains elusive. Introducing such measure is the main purpose of this work.

In this work, we develop a method that allows for nonlinearity of the data and does not depend on pairwise relationships among time series. We also show how the model improve the measurement of systemic risk and explain the connection between Granger-causality and variance decompositions method.

### 4.1.1 Organization

The rest of this chapter is organized as follows. In Section 4.2, we review the literature on graphical models, Granger causality, and introduce directed information graphs. In Section 4.3, we study the causal network of linear models. Section 4.4 studies the causal network of non-linear models. In Section 4.5, we apply our nonlinear method to learn the causal network of set of financial institutions and compare it with the linear regression method in the literature. Finally, we conclude in Section 4.6.

## 4.2 Causal Network

In order to investigating the dynamic of systemic risk, it is important to measure the causal relationship between financial institutions. In this section, we propose a statistical approach to learn such causal interconnections using Granger causality Granger (1969).

### 4.2.1 Graphical Models and Granger Causality

Researchers from different fields have developed various graphical models suitable for their application of interest to encode interconnections among variables or processes. Markov Networks, Bayesian networks (BNs), and Dynamic Bayesian networks (DBNs) are three example of such graphical models that have been used extensively in the literature. In these particular graphical models, nodes represent random variables Koller and Friedman (2009), Murphy (2002).

Markov networks are undirected graphs that represent the conditional independence between the variables. On the other hand BNs and DBNs are directed acyclic graphs (DAGs) that encode conditional dependencies in a reduced factorization of the joint distribution.

Since the size of such graphical models depends on the time-homogeneity and the Markov order of the random processes. Therefore, in general, the graphs can grow with time. As an example, the DBN graph of a vector autoregressive (VAR) with $m$ processes each of order $L$ requires $mL$ nodes Dahlhaus and Eichler (2003). As such they are not suitable for succinct visualization of relationships between the time series such as systemic risks.

In this work, we use directed information graphs (DIGs) to represent interconnections among the financial institutions in which each node represents a time series Massey (1990), Quinn et al. (2015). Below, we formally introduce this type of graphical models. We use an information-theoretical generalization of the notion of Granger causality to determine the interconnection between time series. The basic idea in this framework was originally introduced by Wiener Wiener (1956), and later formalized by Granger Granger (1969). The idea reads as follows: "we say that $X$ is causing $Y$ if we are better able to predict the future of $Y$ using all available information than if the information apart from the past of $X$ had been used."

Granger formulated this framework for practical implementation using multivariate autoregressive (MVAR) models and linear regression. This version has been widely adopted in econometrics and other disciplines Dufour and Taamouti (2010), Granger (1963). More precisely, in order to identify the influence of $X_t$ on

$Y_t$ in a MVAR comprises of three time series $\{X, Y, Z\}$, Granger's idea is to compare the performance of two linear regressions: the first predictor is non-nested that is it predicts $Y_t$ given $\{X^{t-1}, Y^{t-1}, Z^{t-1}\}$, where $X^{t-1}$ denotes the time series $X$ up to time $t-1$ and the second predictor is nested that is it predicts $Y_t$ given $\{Y^{t-1}, Z^{t-1}\}$. Clearly, the performance of the second predictor is bounded by the first predictor and if they have the same performance, then we say $X$ does not cause $Y$. In this framework, since the dynamic between time series is linear, linear regression has been chosen. Next, we introduce directed information (DI), an information-theoretical measure that generalized Granger causality beyond linear models Quinn et al. (2011a).

DI has been used in many applications to infer causal relationships. For example, it has been used for analyzing neuroscience data Kim et al. (2011), Quinn et al. (2011b) and market data Etesami and Kiyavash (2014).

### 4.2.2 Directed Information Graphs (DIGs)

In the rest of this section, we describe how the DI can capture the interconnections in causal[1] dynamical systems (linear or non-linear) and formally define DIGs.

Consider a dynamical system comprised of three time series $\{X, Y, Z\}$. To answer whether $X$ has influence on $Y$ or not over time horizon $[1, T]$, we compare the average performance of two particular predictors with predictions $p$ and $q$ over this time horizon. The first predictor uses the history of all three time series while the second one uses the history of all processes excluding process $X$. On average, the performance of the predictor with less information (the second one) is upper bounded by the performance of the predictor with more information (the first one). However, when the prediction of both predictors, i.e., $p$ and $q$ are close over time horizon $[1, T]$, then we declare that $X$ does not cause $Y$ in this time horizon; otherwise, $X$ causes $Y$.

In order to measure the performance of a predictor, we consider a nonnegative loss function, $\ell(p, y)$, which defines the quality of the prediction. This loss func-

[1] In causal systems, given the full past of the system, the present of the processes become independent. In other words, there are no simulations relationships between the time series.

tion increases as the prediction $p$ deviates more from the true outcome $y$. Although there are many candidate loss functions, e.g. the squared error loss, absolute loss, etc, for the purpose of this work we consider the logarithmic loss.

Moreover, in our setting, the prediction $p$ lies in the space of probability measures over $y$. More precisely, we denote the past of all processes up to time $t1$ by $\mathcal{F}^{t-1}$ that is the $\sigma$-algebra generated by $\{X^{t-1}, Y^{t-1}, Z^{t-1}\}$, where $X^{t-1}$ represents the time series $X$ up to time $t-1$, and denote the past of all processes excluding process $X$, up to time $t-1$ by $\mathcal{F}^{t-1}_{-X}$.

The prediction of the first predictor that is non-nested at time $t$ is given by $p_t := P(Y(t)|\mathcal{F}^{t-1})$ that is the conditional distribution of $Y(t)$ given the past of all processes and the second predictor which is nested is given by $q_t := P(Y_t|\mathcal{F}^{t-1}_{-X})$.

Given a prediction $p$ for an outcome $y \in \mathcal{Y}$, the log loss is defined as $\ell(p, y) := -\log p(y)$. This loss function has meaningful information-theoretical interpretations. The log loss is the Shannon code length, i.e., the number of bits required to efficiently represent a symbol $y$ drawn from distribution $p$. Thus, it may be thought of the description length of $y$.

When the outcome $y_t$ is revealed for $Y_t$, the two predictors incur losses $\ell(p_t, y_t)$ and $\ell(q_t, y_t)$, respectively. The reduction in the loss (description length of $y_t$), known as regret is defined as

$$r_t := \ell(q_t, y_t) - \ell(p_t, y_t) = \log \frac{p_t}{q_t} = \log \frac{P(Y_t = y_t|\mathcal{F}^{t-1})}{P(Y_t = y_t|\mathcal{F}^{t-1}_{-X})} \geq 0.$$

Note that the regrets are non-negative. The average regret over the time horizon $[1, T]$ given by $\frac{1}{T}\sum_{t=1}^{T} \mathbb{E}[r_t]$, where the expectation is taken over the joint distribution of $X$, $Y$, and $Z$ is called *directed information* (DI). This will be our measure of causation and its value determines the strength of influence. If this quantity is close to zero, it indicates that the past values of time series $X$ contain no significant information that would help in predicting the future of time series $Y$ given the history of $Y$ and $Z$. This definition may be generalized to more than 3 processes as follows,

**Definition 1** *Consider a network of m time series $\underline{R} := \{R_1, ..., R_m\}$. We declare $R_i$ influences $R_j$ over time horizon $[1, T]$, if and only if*

$$I(R_i \to R_j || \underline{R}_{-\{i,j\}}) := \frac{1}{T} \sum_{t=1}^{T} \mathbb{E} \left[ \log \frac{P(R_{j,t} | \mathcal{F}^{t-1})}{P(R_{j,t} | \mathcal{F}^{t-1}_{-\{i\}})} \right] > 0, \tag{4.1}$$

*where $\underline{R}_{-\{i,j\}} := \underline{R} \setminus \{R_i, R_j\}$. $\mathcal{F}^{t-1}$ denotes the sigma algebra generated by $\underline{R}^{t-1} := \{R_1^{t-1}, ..., R_m^{t-1}\}$, and $\mathcal{F}^{t-1}_{-\{i\}}$ denotes the sigma algebra generated by $\{R_1^{t-1}, ..., R_m^{t-1}\} \setminus \{R_i^{t-1}\}$.*

**Definition 2** *Directed information graph (DIG) of a set of m processes $\underline{R} = \{R_1, ..., R_m\}$ is a weighted directed graph $G = (V, E, W)$, where nodes represent processes ($V = \underline{R}$) and arrow $(R_i, R_j) \in E$ denotes that $R_i$ influences $R_j$ with weight $I(R_i \to R_j || \underline{R}_{-\{i,j\}})$. Consequently, $(R_i, R_j) \notin E$ if and only if its corresponding weight is zero.*

**Remark 1** *Pairwise comparison has been applied in the literature to identify the causal structure of time series Allen et al. (2010), Billio et al. (2010, 2012). Such comparison is not correct in general and fails to capture the true underlying network as we will see in the next example. For more details please see Quinn et al. (2015).*

**Example 1** *As an example, consider a network of three times series $\{X, Y, Z\}$ with the following linear model:*

$$\begin{aligned} X_t &= a_1 X_{t-1} + a_2 Z_{t-1} + \varepsilon_{x_t}, \\ Z_t &= a_3 Z_{t-1} + \varepsilon_{z_t}, \\ Y_t &= a_4 Y_{t-1} + a_5 Z_{t-1} + \varepsilon_{y_t}, \end{aligned} \tag{4.2}$$

*where $\varepsilon_x$, $\varepsilon_y$, and $\varepsilon_z$ are three independent white noise processes, and $\{a_1, ..., a_5\}$ are non-zero coefficients of the model. Due to the functional relationships between these time series, we have that the causal network of this model is $X \leftarrow Z \to Y$, i.e., there is an arrow from Z to X and Z to Y because $X_t$ and $Y_t$ depend on $Z_{t-1}$, respectively. This* *can also be inferred using the DIs in (4.1), it is straight forward to show that*

$$I(X \to Y||Z) = 0, \quad I(X \to Z||Y) = 0,$$
$$I(Y \to X||Z) = 0, \quad I(Y \to Z||X) = 0,$$
$$I(Z \to Y||X) > 0, \quad I(Z \to X||Y) > 0.$$

*Notice that none of the above DIs are pairwise as they have conditioned on the remaining time series. However, considering the pairwise causal relationships, for instance between X and Y will give us*

$$I(X \to Y) = \frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\log\frac{P(Y_t|Y^{t-1}, X^{t-1})}{P(Y_t|Y^{t-1})}\right] > 0.$$

*Hence, looking into pairwise causal relationships, we obtain that X directly causes Y that is not the case in this example.*

A causal model allows a factorization of the joint distribution in some specific ways. It was shown in (Quinn et al., 2015) that under a mild assumption, the joint distribution of a causal discrete-time dynamical system with $m$ time series can be factorized as follows,

$$P_{\underline{R}} = \prod_{i=1}^{m} P_{R_i||\underline{R}_{B_i}}, \tag{4.3}$$

where $B_i \subseteq -\{i\} := \{1, ..., m\} \setminus \{i\}$ is the minimal[2] set of processes that causes process $R_i$, i.e., parent set of node $i$ in the corresponding DIG. Such factorization of the joint distribution is called minimal generative model. In Equation (4.3), $P(\cdot||\cdot)$ is called causal conditioning and defined as follows

$$P_{R_i||\underline{R}_{B_i}} := \prod_{t=1}^{T} P_{R_{i,t}|\underline{\mathcal{F}}^{t-1}_{B_i \cup \{i\}}},$$

and $\mathcal{F}^{t-1}_{B_i\cup\{i\}} = \sigma\{\underline{R}^{t-1}_{B_i\cup\{i\}}\}$.

It is important to emphasize that learning the causal network using DI does not require any specific model for the system. There are different methods that can

[2] Minimal in terms of its cardinality.

estimate (4.1) given i.i.d. samples of the time series such as plug-in empirical estimator, k-nearest neighbor estimator, etc Frenzel and Pompe (2007), Jiao et al. (2013), Kraskov et al. (2004).

In general, estimating DI in (4.1) is a complicated task and has high sample complexity. However, knowing some side information about the system can simplify the learning task. In the following section, we describe learning the causal network of linear systems. Later in Section 4.4, we discuss generalization to non-linear models.

### 4.2.3 Quantifying Causal Relationships

The purpose of this section is to justify that the DI introduced in (4.1) also quantifies the causal relationships in a network. We do so using a simple linear model and then generalize it to nonlinear systems.

Consider a network of three time series $\vec{X}_t = (X_{1,t}, X_{2,t}, X_{3,t})^T$ with the following dynamic

$$\vec{X}_t = \begin{pmatrix} 0 & 0.1 & 0.3 \\ 0 & 0 & -0.2 \\ 0 & 0 & 0 \end{pmatrix} \vec{X}_{t-1} + \vec{\varepsilon}_t, \tag{4.4}$$

where $\vec{\varepsilon}_t$ denotes a vector of exogenous noises that has normal distribution with mean zero and covariance matrix **I**. Figure 4.2.1 shows the corresponding DIG of this network. Note that in this particular example that the relationships are linear, the support of the coefficient matrix also encodes the corresponding DIG of the network.

In order to compare the strength of causal relationships $X_2 \to X_1$ and $X_3 \to X_1$ over a time horizon $[1, T]$, we compare the performance of two linear predictors of $X_{1,t}$ over that time horizon. The first predictor $(L_1)$ predicts $X_{1,t}$ using $\{X_1^{t-1}, X_3^{t-1}\}$ and the other predictor $(L_2)$ uses $\{X_1^{t-1}, X_2^{t-1}\}$. If $L_1$ shows better performance compared to $L_2$, it implies that $X_3$ contains more relevant information about $X_1$ compared to $X_2$. In other words, $X_3$ has stronger influence on $X_1$ compared to $X_2$.

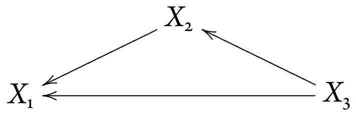


**Figure 4.2.1:** Corresponding DIG of the system in (4.4).

To compare the performance of $L_1$ and $L_2$, we consider their mean squared errors over the time horizon $[1, T]$.

$$L_1 : e_1 := \frac{1}{T}\sum_{t=1}^{T} \min_{y_t\in\mathcal{A}_t} \mathbb{E}||X_{1,t}-y_t||^2, \text{ where } \mathcal{A}_t := \text{span}\{X_1^{t-1}, X_3^{t-1}\},$$

$$L_2 : e_2 := \frac{1}{T}\sum_{t=1}^{T} \min_{z_t\in\mathcal{B}_t} \mathbb{E}||X_{1,t}-z_t||^2, \text{ where } \mathcal{B}_t := \text{span}\{X_1^{t-1}, X_2^{t-1}\}.$$

It is easy to show that $e_1 = 1 + 0.1^2$ and $e_2 = 1 + 0.3^2$. Since $e_1 < e_2$, we infer that $X_3$ has stronger influence on $X_1$ compared to $X_2$.

Analogous to the directed information graphs, we can generalize the above framework to non-linear systems. Consider a network of $m$ time series $\underline{R} = \{R_1, ..., R_m\}$ with corresponding DIG $G = (V, E, W)$. Suppose $(R_i, R_j)$ and $(R_k, R_j)$ belong to $E$, i.e., $R_i$ and $R_k$ both are parents of $R_j$. We say $R_i$ has stronger influence on $R_j$ compared to $R_k$ over a time horizon $[1, T]$ if $P(R_{j,t}|\mathcal{F}^{t-1}_{-\{k\}})$ is a better predictor for $R_{j,t}$ compared to $P(R_{j,t}|\mathcal{F}^{t-1}_{-\{i\}})$ over that time horizon. In other words, $R_i$ has stronger influence on $R_j$ compared to $R_k$, if

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\log\frac{P(R_{j,t}|\mathcal{F}^{t-1}_{-\{k\}})}{P(R_{j,t}|\mathcal{F}^{t-1}_{-\{i\}})}\right] > 0.$$

The above inequality holds if and only if $I(R_i \to R_j||\underline{R}_{-\{i,j\}}) > I(R_k \to R_j||\underline{R}_{-\{k,j\}})$. Thus, the DI in (4.1) can quantify the causal relationships in a network. For instance, looking again at the system in (4.4), we obtain

$$I(X_2 \to X_1||X_3) = \frac{1}{2}\log(1 + 0.1^2) < \frac{1}{2}\log(1 + 0.3^2) = I(X_3 \to X_1||X_2).$$

## 4.3 DIG of Linear Models

Herein, we study the causal network of linear systems. Consider a set of $m$ stationary time series, and for simplicity assume they have zero mean, such that their relationships are captured by the following model:

$$\vec{R}_t = \sum_{k=1}^{p} \mathbf{A}_k \vec{R}_{t-k} + \vec{\varepsilon}_t, \tag{4.5}$$

where $\vec{R}_t = (R_{1,t}, ..., R_{m,t})^T$, and $\mathbf{A}_k$s are $m \times m$ matrices. Moreover, we assume that the exogenous noises, i.e., $\varepsilon_{i,t}$s are independent and also independent from $\{R_{j,t}\}$. For simplicity, we assume that the $\{\varepsilon_{i,t}\}$ have mean zero. For the model in (4.5), it was shown in Etesami and Kiyavash (2014) that

$$I(R_i \to R_j || \underline{R}_{-\{i,j\}}) > 0,$$

if and only if $\sum_{k=1}^{p} |(\mathbf{A}_k)_{j,i}| > 0$, where $(\mathbf{A}_k)_{j,i}$ is the $(j,i)$th entry of matrix $\mathbf{A}_k$. Thus, to learn the corresponding causal network (DIG) of this model, instead of estimating the DIs in (4.1), we can check whether the corresponding coefficients are zero or not. To do so, we use the Bayesian information criterion (BIC) as the model-selection criterion to learn the parameter $p$ Schwarz et al. (1978), and use F-tests to check the null hypotheses that the coefficients are zero Lomax and Hahs-Vaughn (2013).

Wiener filtering is another alternative approach that can estimate the coefficients and consequently learn the DIG Materassi and Salapaka (2012). The idea of this approach is to find the coefficients by solving the following optimization problem,

$$\{\hat{\mathbf{A}}_1, ..., \hat{\mathbf{A}}_p\} = \arg\min_{\mathbf{B}_1,...,\mathbf{B}_p} \mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T} ||\vec{R}_t - \sum_{k=1}^{p} \mathbf{B}_k \vec{R}_{t-k}||^2\right].$$

This leads to a set of Yule-Walker equations that can be solved efficiently by Levinson-Durbin algorithm Musicus (1988).

### 4.3.1 DIG of GARCH models

The relationship between the coefficients of the linear model and the corresponding DIG can easily be extended to the financial data in which the variance of $\{\varepsilon_{i,t}\}_{t=1}^{T}$ are no longer independent of $\{R_{i,t}\}$ but due to the heteroskedasticity, they are $\mathcal{F}_i^{t-1}$-measurable. More precisely, in financial data, the returns are modeled by GARCH that is given by

$$\begin{aligned} R_{i,t}|\mathcal{F}^{t-1} &\sim \mathcal{N}(\mu_{i,t}, \sigma^2_{i,t}), \\ \sigma^2_{i,t} &= \alpha_0 + \sum_{k=1}^{q} \alpha_k (R_{i,t-k} - \mu_{i,t})^2 + \sum_{l=1}^{s} \beta_l \sigma^2_{i,t-l}, \end{aligned} \tag{4.6}$$

where $\alpha_k$s and $\beta_l$s are nonnegative constants.

**PROPOSITION 1** *Consider a network of time series whose dynamic is given by (4.6). In this case, there is no arrow from $R_j$ to $R_i$ in its corresponding DIG, i.e., $R_j$ does not cause $R_i$ if and only if*

$$\mathbb{E}[R_{i,t}|\mathcal{F}^{t-1}] = \mathbb{E}[R_{i,t}|\mathcal{F}^{t-1}_{-\{j\}}], \ \forall t. \tag{4.7}$$

See Appendix 4.7.1.

Multivariate GARCH models are a a generalization of (4.6) in which the variance of $e_{i,t}$ is $\mathcal{F}^{t-1}$-measurable. In this case, not only $\mu_{i,t}$ but also $\sigma^2_{i,t}$ capture the interactions between the returns. More precisely, in multivariate GARCH, we have

$$\begin{aligned} \vec{R}_t|\mathcal{F}^{t-1} &\sim \mathcal{N}(\vec{\mu}_t, \mathbf{H}_t), \\ vech[\mathbf{H}_t] &= \Omega_0 + \sum_{k=1}^{q} \Omega_k vech[\vec{\varepsilon}_{t-k}\vec{\varepsilon}^{T}_{t-k}] + \sum_{l=1}^{p} \Gamma_l vech[\mathbf{H}_{t-l}], \end{aligned}$$

where $\vec{\mu}_t$ is an $m \times 1$ array, $\mathbf{H}_t$ is an $m \times m$ symmetric positive definite and $\mathcal{F}^{t-1}$-

measurable matrix, and $\vec{\varepsilon}_t = \vec{R}_t - \vec{\mu}_t$. Note that *vech* denotes the vector-half operator, which stacks the lower triangular elements of an $m \times m$ matrix as an $(m(m+1)/2) \times 1$ array.

**PROPOSITION 2** *Consider a network of time series whose dynamic is captured by a multivariate GARCH model. In this case, there is no arrow from $R_j$ to $R_i$ in its corresponding DIG, i.e., $R_j$ does not influence $R_i$ if and only if both the condition in Proposition 1 and the following condition hold*

$$\mathbb{E}[(R_{i,t} - \mu_{i,t})^2 | \mathcal{F}^{t-1}] = \mathbb{E}[(R_{i,t} - \mu_{i,t})^2 | \mathcal{F}^{t-1}_{-\{j\}}], \ \forall t. \tag{4.8}$$

See Appendix 4.7.2.

Next example demonstrates the connection between the DIG of a multivariate GARCH and its corresponding parameters.

**Example 2** *Consider the following multivariate GARCH(1,1) model*

$$\begin{pmatrix} R_{1,t} \\ R_{2,t} \end{pmatrix} = \begin{pmatrix} 0.2 & 0.3 \\ 0 & 0.2 \end{pmatrix} \begin{pmatrix} R_{1,t-1} \\ R_{2,t-1} \end{pmatrix} + \begin{pmatrix} \varepsilon_{1,t} \\ \varepsilon_{2,t} \end{pmatrix},$$
$$\begin{pmatrix} \sigma^2_{1,t} \\ \rho_t \\ \sigma^2_{2,t} \end{pmatrix} = \begin{pmatrix} 0 \\ 0.3 \\ 0.1 \end{pmatrix} + \begin{pmatrix} 0.2 & 0 & 0.3 \\ 0 & 0.2 & 0.7 \\ 0.1 & 0.4 & 0 \end{pmatrix} \begin{pmatrix} \varepsilon^2_{1,t-1} \\ \varepsilon_{1,t-1}\varepsilon_{2,t-1} \\ \varepsilon^2_{2,t-1} \end{pmatrix} + \begin{pmatrix} 0.3 & 0.5 & 0 \\ 0.1 & 0.2 & 0 \\ 0 & 0 & 0.4 \end{pmatrix} \begin{pmatrix} \sigma^2_{1,t-1} \\ \rho_{t-1} \\ \sigma^2_{2,t-1} \end{pmatrix}, \tag{4.9}$$

*where $\rho_t = \mathbb{E}[\varepsilon_{1,t}\varepsilon_{2,t}]$. The corresponding DIG of this model is $R_1 \leftrightarrow R_2$. This is because $R_2$ influences $R_1$ through the mean and variance and $R_1$ influences $R_2$ only through the variance.*

**Remark 2** *Recall that as we mentioned in Remark 1 and Example 1, the pairwise Granger-causality calculation, in general, fails to identify the true causal network. It was proposed in Billio et al. (2012) that the returns of the ith institution linearly depend on the past returns of the jth institution, when*

$$\mathbb{E}[R_{i,t}|\mathcal{F}^{t-1}] = \mathbb{E}\left[R_{i,t}|R_{j,t-1}, R_{i,t-1}, \{R_{j,\tau} - \mu_{j,\tau}\}_{\tau=-\infty}^{t-2}, \{R_{i,\tau} - \mu_{i,\tau}\}_{\tau=-\infty}^{t-2}\right].$$

*This test is obtained based on pairwise Granger-causality calculation and does not consider non-linear causation through the variance of $\{\varepsilon_i\}$. For instance, if the returns of two institutions $R_j$ and $R_k$ cause the returns of the ith institution, then the above equality does not hold, because $R_k$ cannot be removed from the conditioning.*

### 4.3.2 DIG of Moving-Average (MA) Models

The model in (4.5) may be represented as an infinite moving average (MA) or data-generating process (GDP), as long as $\vec{R}(t)$ is covariance-stationary, i.e., all the roots of $|\mathbf{I} - \sum_{k=1}^{p} \mathbf{A}_k z^k|$ fall outside the unit circle Pesaran and Shin (1998):

$$\vec{R}_t = \sum_{k=0}^{\infty} \mathbf{W}_k \vec{\varepsilon}_{t-k}, \tag{4.10}$$

where $\mathbf{W}_k = 0$ for $k < 0, \mathbf{W}_0 = \mathbf{I}$, and $\mathbf{W}_k = \sum_{l=1}^{p} \mathbf{W}_{k-l}\mathbf{A}_l$. In this representation, $\{\varepsilon_i\}$s are called shocks and if they are independent, they are also called orthogonal Diebold and Yılmaz (2014).

In this section, we study the causal structure of a MA model of finite order $p$. Consider a moving average model with orthogonal shocks given by

$$\vec{R}_t = \sum_{k=0}^{p} \mathbf{W}_k \vec{\varepsilon}_{t-k}, \tag{4.11}$$

where $\mathbf{W}_i$s are $m \times m$ matrices such that $\mathbf{W}_0$ is non-singular with nonzero diagonals and without loss of generality, we can assume that $diag(\mathbf{W}_0)$ is the identity matrix. Equation (4.11) can be written as $\vec{R}_t = \mathbf{W}_0 \vec{\varepsilon}_t + \mathcal{P}(L)\vec{\varepsilon}_{t-1}$, where $\mathcal{P}(L) := \sum_{k=1}^{p} \mathbf{W}_k L^{k-1}$. Subsequently, we have

$$\mathbf{W}_0^{-1}\vec{R}_t = \vec{\varepsilon}_t + \sum_{k=1}^{\infty} (-1)^{k-1} \left(\mathbf{W}_0^{-1}\mathcal{P}(L)\right)^k \mathbf{W}_0^{-1}\vec{R}_{t-k}. \tag{4.12}$$

This representation is equivalent to an infinite AR model. Hence using the result in Etesami and Kiyavash (2014), we can conclude the following corollary.

**COROLLARY 1** *Consider a MA model described by (4.11) with orthogonal shocks such that $\boldsymbol{W}_0$ is non-singular and diagonal. In this case, $R_j$ does not influence $R_i$ if and only if the corresponding coefficients of $\{R_{j,t-k}\}_{k>0}$ in $R_i$'s equation are zero.*

In the interest of simplicity and space, we do not present the explicit form of these coefficients, but we show the importance of this result using a simple example.

**Example 3** *Consider a MA(1) with dimension three such that $\boldsymbol{W}_0 = \boldsymbol{I}$, and*

$$\boldsymbol{W}_1 = \begin{pmatrix} 0.3 & 0 & 0.5 \\ 0.1 & 0.2 & 0.5 \\ 0 & 0.4 & 0.1 \end{pmatrix}, \quad \boldsymbol{W}_1^2 = \begin{pmatrix} 0.09 & 0.2 & 0.2 \\ 0.05 & 0.24 & 0.2 \\ 0.04 & 0.12 & 0.21 \end{pmatrix},$$

*Using the expression in (4.12), we have $\vec{R}_t = \vec{\varepsilon}_t + \sum_{k=1}^{\infty}(-1)^{k-1}\mathbf{W}_1^k\vec{R}_{t-k}$. Because, $\boldsymbol{W}_1^2$ has no nonzero entry, the causal network (DIG) of this model is a complete graph.*

We studied the DIG of a MA model with orthogonal shocks. However, the shocks are rarely orthogonal in practice. To identify the causal structure of such systems, we can apply the whitening transformation to transform the shocks into a set of uncorrelated variables. More precisely, suppose $\mathbb{E}[\vec{\varepsilon}_t\vec{\varepsilon}_t^T] = \Sigma$, where the Cholesky decomposition of $\Sigma$ is $\mathbf{V}\mathbf{V}^T$ Horn and Johnson (2012). Hence, $\mathbf{V}^{-1}\vec{\varepsilon}_t$ is a vector of uncorrelated shocks. Using this fact, we can transform (4.11) with correlated shocks into

$$\vec{R}_t = \sum_{k=0}^{p} \tilde{\mathbf{W}}_k \vec{\tilde{\varepsilon}}_{t-k}, \tag{4.13}$$

with uncorrelated shocks, where $\vec{\tilde{\varepsilon}}_t := \mathbf{V}^{-1}\vec{\varepsilon}_t$ and $\tilde{\mathbf{W}}_k := \mathbf{W}_k\mathbf{V}$.

**Remark 3** *The authors in Diebold and Yılmaz (2014) applied the generalized variance decomposition (GVD) method to identify the population connectedness or in another word the causal structure of a MA model with correlated shocks. Using this method, they monitor and characterize the network of major U.S. financial institutions during 2007-2008 financial crisis. In this method, the weight of $R_j$'s influence on $R_i$ in (4.11)* *was defined to be proportional to*

$$d_{i,j} = \sum_{k=0}^{p} \left( (\boldsymbol{W}_k \Sigma)_{i,j} \right)^2 , \tag{4.14}$$

*where $(\boldsymbol{A})_{i,j}$ denotes the $(i,j)$-th entry of matrix $\boldsymbol{A}$. Recall that $\mathbb{E}[\vec{\varepsilon}_t \vec{\varepsilon}_t^T] = \Sigma$. Applying the GVD method to Example 3, where $\Sigma = \boldsymbol{I}$, we obtain that $d_{1,2} = d_{3,1} = 0$. That is $R_2$ does not influence $R_1$ and $R_1$ does not influence $R_3$. This result is not consistent with the Granger-causality concept since the corresponding causal network (DIG) of this example is a complete graph, i.e., every node has influence on any other node. Thus, GVD analysis of Diebold and Yılmaz (2014) is also seems to suffer from disregarding the entire network akin to pairwise analysis commonly used in traditional application of the Granger-causality.*

## 4.4 DIG of Non-linear Models

DIG as defined in Definition 2 does not require any linearity assumptions on the model. But, similar to Billio et al. (2010), side information about the model class can simplify computation of (4.1). For instance, let us assume that $\underline{R}$ is a first-order Markov chain with transition probabilities:

$$P(\underline{Y}_t | \underline{R}^{t-1}) = P(\underline{R}_t | \underline{R}_{t-1}).$$

In this setup, $I(R_i \to R_j || \underline{R}_{-\{i,j\}}) = 0$ if and only if

$$P(R_{j,t} | \underline{R}_{t-1}) = P(R_{j,t} | \underline{R}_{-\{i\},t-1}), \forall t.$$

Recall that $\underline{R}_{-\{i\},t-1}$ denotes $\{R_{1,t-1}, \dots, R_{m,t-1}\} \setminus \{R_{i,t-1}\}$. Furthermore, suppose that the transition probabilities are represented through a logistic function again as in Billio et al. (2010). More specifically, for any subset of processes $\mathcal{S} := \{R_{i_1}, \dots, R_{i_s}\} \subseteq$

$\underline{R}$, we have

$$P(R_{j,t}|R_{i_1,t-1},\ldots,R_{i_s,t-1}) := \frac{\exp(\vec{a}_{\mathcal{S}}^T\vec{U}_{\mathcal{S}})}{1+\exp(\vec{a}_{\mathcal{S}}^T\vec{U}_{\mathcal{S}})},$$

where $\vec{U}_{\mathcal{S}}^T := \bigotimes_{i\in\mathcal{S}}(1,R_{i,t-1}) = (1,R_{i_1,t-1})\otimes(1,R_{i_2,t-1})\otimes\cdots\otimes(1,R_{i_s,t-1})$, $\otimes$ denotes the Kronecker product, and $\vec{a}_{\mathcal{S}}$ is a vector of dimension $2^s \times 1$. Under these assumptions, the causal discovery in the network reduces to the following statement: $R_i$ does not influence $R_j$ if and only if all the terms of $\vec{U}_{\underline{R}}$ depending on $R_i$ are equal to zero. More precisely:

$$\vec{U}_{\underline{R}} = \vec{U}_{\underline{R}_{-\{i\}}}\otimes(1,R_{i,t-1}) = (\vec{U}_{\underline{R}_{-\{i\}}},\vec{U}_{\underline{R}_{-\{i\}}}R_{i,t-1}).$$

Let $\vec{a}_{\underline{R}}^T = (\vec{a}_1^T,\vec{a}_2^T)$, where $\vec{a}_1$ and $\vec{a}_2$ are the vectors of coefficients corresponding to $\vec{U}_{\underline{R}_{-\{i\}}}$ and $\vec{U}_{\underline{R}_{-\{i\}}}R_{i,t-1}$, respectively. Then $R_i \nrightarrow R_j$ if and only if $\vec{a}_2 = 0$.

Another such non-linear models are Multiple chain Markov switching models (MCMS)-VAR Billio and Di Sanzo (2015), in which the relationship between time series $\underline{Y}_t$ is given by

$$Y_{i,t} = \mu_i(S_{i,t}) + \sum_{k=1}^{p}\sum_{j=1}^{m}(\mathbf{B}_k(S_{i,t}))_{i,j}Y_{j,t-k} + \varepsilon_{i,t},\ \text{for } i\in\{1,\ldots,m\}, \tag{4.15}$$

and $\vec{\varepsilon}_t := (\varepsilon_{1,t},\ldots,\varepsilon_{m,t}) \sim \mathcal{N}(0,\Sigma(\vec{S}_t))$, where the mean, the lag matrices, and the covariance matrix of the error terms all depend on a latent random vector $\vec{S}_t$ known as the state of the system. $S_{i,t}$ represents the state variable associated with $Y_{i,t}$ that can take values from a finite set $\mathcal{S}$. The random sequence $\{\vec{S}_t\}$ is assumed to be a time-homogenous first-order Markov process with one-step ahead transition probability $P(\vec{S}_t|\underline{S}^{t-1},\underline{Y}^{t-1}) = P(\vec{S}_t|\underline{S}_{t-1})$. Furthermore, we assume that given the past of the states, their presents are independent, i.e.,

$$P(\vec{S}_t|\underline{S}_{t-1}) = \prod_j P(S_{j,t}|\underline{S}_{t-1}).$$

Next result stresses a set of conditions under which by observing the time series $\underline{Y}_t$, we are able to identify the causal relationships between them.

**PROPOSITION 3** *Consider a MCMS-VAR in which $\Sigma(\vec{S}_t)$ is diagonal for all $\vec{S}_t$. In this case, $I(Y_j \to Y_i||\underline{Y}_{-\{i,j\}}) = 0$ if*

- $(\boldsymbol{B}_k(s_{i,t}))_{i,j} = 0$ *for all realizations $s_{i,t}$,*
- $(\Sigma(\vec{S}_t))_{i,i} = (\Sigma(S_{i,t}))_{i,i}$,
- $P(S_{k,t}|\underline{S}^{t-1}, \underline{S}_{-\{k\},t}) = P(S_{k,t}|S_{k,t-1})$ *for every k.*

See Appendix 4.7.3. Note that the third condition in this proposition seems strong compared to the condition in Billio and Di Sanzo (2015). But notice that Billio and Di Sanzo (2015) studies the causal relationships between the time series given the state variables, which is not realistic as they are hidden. Below, we show a simple example in which $Y_1$ does not functionally depend on $Y_2$ and $S_1$ is statistically independent of $S_2$. However, in this example, observing the states leads to $Y_2$ has no influence on $Y_1$, but without observing the states we infer differently.

**Example 4** *Consider a bivariate MCMS-VAR $\{Y_1, Y_2\}$ in which the states only take binary values and*

$$
\begin{aligned}
Y_{1,t} &= b_{1,1}(S_{1,t})Y_{1,t-1} + 0.1\varepsilon_{1,t},\\
Y_{2,t} &= \mu_2(S_{2,t}) + 0.5Y_{1,t-1} + 0.1\varepsilon_{2,t},
\end{aligned}
$$

*where $(\varepsilon_{1,t}, \varepsilon_{2,t}) \sim \mathcal{N}(0, I)$, $\mu_2(0) = 10$, $\mu_2(1) = -5$, $b_{1,1}(0) = 0.5$, and $b_{1,1}(1) = -0.5$. Moreover, the transition probabilities of the states are $P(S_{1,t}|S_{1,t-1}, S_{2,t-1}) = P(S_{1,t}|S_{1,t-1}) = 0.8$ whenever $S_{1,t} = S_{1,t-1}$, and $S_{2,t}$ equals to $S_{1,t-1}$ with probability 0.9. Based on Billio and Di Sanzo (2015), in this setup, $Y_{2,t-1}$ does not Granger-cause $Y_{1,t}$ given $Y_{1,t-1}, S_{1,t-1}$, i.e.,*

$$P(Y_{1,t}|Y_{2,t-1}, Y_{1,t-1}, S_{1,t-1}) = P(Y_{1,t}|Y_{1,t-1}, S_{1,t-1}).$$

*Note that in this example, $P(Y_{1,t}|Y_{2,t-1}, Y_{1,t-1}) \neq P(Y_{1,t}|Y_{1,t-1})$. This is because, $Y_{2,t-1}$ has information about $S_{2,t-1}$ which contains information about $S_{1,t-2}$.*

## 4.5 Experimental Result

In we have introduced tools for identifying the causal structure in a network of time series. In this section, we put those tools to work and use them to identify and monitor the evolution of connectedness among major financial institutions during 2006-2016.

### 4.5.1 Data

We obtained the data for individual banks, broker/dealers, and insurers from bloomberg, from which we selected the daily returns of all companies listed in Table 4.5.1.

**Table 4.5.1:** . List of companies in our experiment.

| **Banks** | |
|---|---|
| FNMA US | BNS US |
| AXP US | STI US |
| FMCC US | C US |
| BAC US | MS US |
| WFC UN | SLM US |
| JPM US | BBT US |
| DB US | USB US |
| NTRS US | TD US |
| RY US | HSBC US |
| PNC US | BCS US |
| STT US | GS US |
| COF US | MS US |
| BMO US | CS US |
| CM US | |
| RF UN | |

| **Insurances** | |
|---|---|
| MET US | PFG US |
| ANTM US | LNC US |
| AET US | AON US |
| CNA US | HUM US |
| XL US | MMC US |
| SLF US | HIG US |
| MFC US | CI US |
| GNW US | ALL US |
| PRU US | BRK/B US |
| AIG US | CPYYY US |
| PGR US | AHL US |
| CB US | |
| BRK/A US | |
| UNH US | |
| AFL US | |

| **Brokers** | |
|---|---|
| MS US | WDR US |
| GS US | EV US |
| BEN US | ITG UN |
| MORN US | JNS US |
| LAZ US | SCHW US |
| ICE US | ETFC US |
| AINV US | AMTD US |
| SEIC US | |
| FII US | |
| RDN US | |
| TROW US | |
| AMP US | |
| GHL US | |
| AMG US | |
| RJF US | |

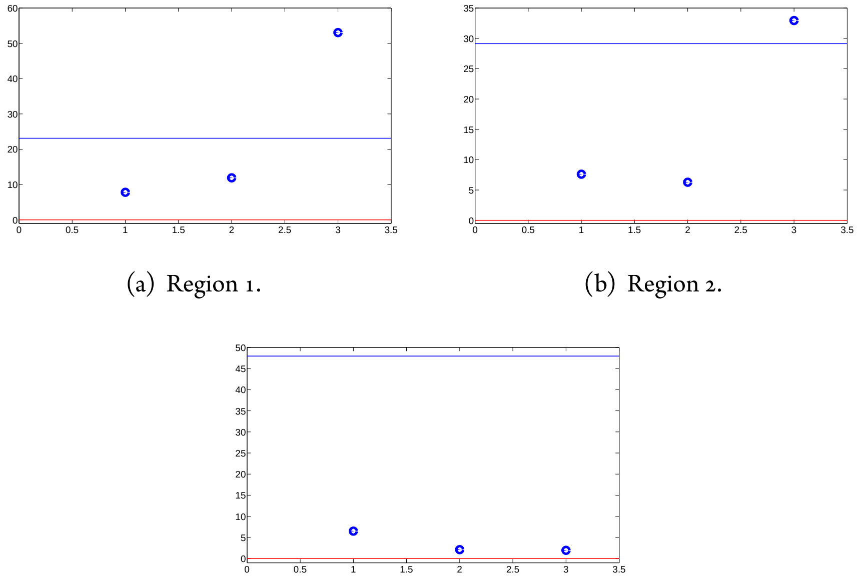
We calculated the causal network for different time periods that will be considered in the empirical analysis: 2006-2008, 2009-2011, 2011-2013, and 2013-2016.

### 4.5.2 Non-linearity Test

In this section, we applied a non-linearity test on the data to determine whether the underlying structure within the recorded data is linear or nonlinear. The non-linearity test applied in this section is based on nonlinear principle component analysis (PCA) Kruger et al. (2008). This test is based on two principles: the range of recorded data is divided into smaller disjunct regions; and accuracy bounds are determined for the sum of the discarded eigenvalues of each region. If this sum is within the accuracy bounds for each region, the process is assumed to be linear. Conversely, if at least one of these sums is outside, the process is assumed to be nonlinear.

More precisely, the second principle in this test requires computation of the correlation matrix for each of the disjunct regions. Since the elements of this matrix are obtained using a finite dataset, applying *t*-distribution and $\chi^2$-distribution establish confidence bounds for both estimated mean and variance, respectively. Subsequently, these confidence bounds can be utilized to determine thresholds for each element in the correlation matrix. Using these thresholds, the test calculates maximum and minimum eigenvalues relating to the discarded score variables, which in turn allows the determination of both a minimum and a maximum accuracy bound for the variance of the prediction error of the PCA model. This is because the variance of the prediction error is equal to the sum of the discarded eigenvalues. If this sum lies inside the accuracy bounds for each disjunct region, a linear PCA model is then appropriate over the entire region. Alternatively, if at least one of these sums is outside the accuracy bounds, the error variance of the PCA model residuals then differs significantly for this disjunct region and hence, a nonlinear model is required. For more details see Kruger et al. (2008).

We divided the operating region into 3 disjunct regions. The accuracy bounds for each disjuct region and also sum of the discarded eigenvalues were computed. These bounds were based on thresholds for each element of the correlation matrix corresponding to confidence level of 95%. Note that the processes were normalized with respect to the mean and variance of the regions for which the accuracy bounds were computed. Figure 4.5.1 shows the accuracy bounds and the sum of the discarded eigenvalues. As figures 4.5.1-(a) and 4.5.1-(b) illustrate, the recorded financial data is nonlinear.



(a) Region 1.

(b) Region 2.

(c) Region 3.

**Figure 4.5.1:** Benchmarking of the residual variances against accuracy bounds of each disjunct region.

### 4.5.3 Estimating the DIs

As we mentioned earlier, there are different methods that can be used to estimate (4.1) given i.i.d. samples of the time series. Plug-in empirical estimator and k-nearest neighbor estimator are such two methods Frenzel and Pompe (2007), Jiao et al. (2013), Kraskov et al. (2004). For our experimental results, we used k-nearest method to estimate the DIs since it shows relatively better performance compared to the other non-parametric estimators. To do so, we used the fact that

$$I(R_i \to R_j||\underline{R}_{-\{i,j\}}) = \frac{1}{T}\sum_{t=1}^{T} I(R_{j,t}; R_i^{t-1}|\underline{R}_{-\{i,j\}}^{t-1}, R_j^{t-1}),$$

where $I(X;Y|Z)$ denotes conditional mutual information between $X$ and $Y$ given $Z$ Cover and Thomas (2012). Then, we estimated each of the above conditional mutual information using k-nearest method in Sricharan et al. (2011). Below, we describe the steps of k-nearest method to estimate $I(X;Y|Z)$.

Suppose that $N+M$ i.i.d. realizations $\{\mathbf{X}_1, ..., \mathbf{X}_{N+M}\}$ are available from $P_{X,Y,Z}$, where $\mathbf{X}_i$ denotes the $i$th realization of $(X,Y,Z)$. The data sample is randomly divided into two subsets $S_1$ and $S_2$ of $N$ and $M$ points, respectively. In the first stage, an k-nearest density estimator $\widehat{P}_{X,Y,Z}$ at the $N$ points of $S_1$ is estimated using the $M$ realizations of $S_2$ as follows: Let $d(\mathbf{x},\mathbf{y})$ denote the Euclidean distance between points $\mathbf{x}$ and $\mathbf{y}$ and $d_k(\mathbf{x})$ denotes the Euclidean distance between a point $\mathbf{x}$ and its k-th nearest neighbor among $S_2$. The k-nearest region is $S_k(\mathbf{x}) := \{\mathbf{y} : d(\mathbf{x},\mathbf{y}) \leq d_k(\mathbf{x})\}$ and the volume of this region is $V_k(\mathbf{x}) := \int_{S_k(\mathbf{x})} dn$. The standard k-nearest density estimator Sricharan et al. (2011) is defined as

$$\widehat{P}_{X,Y,Z}(\mathbf{x}) := \frac{k-1}{MV_k(\mathbf{x})}.$$

Similarly, we obtain k-nearest density estimators $\widehat{P}_{X,Z}$, $\widehat{P}_{Y,Z}$, and $\widehat{P}_Z$. Subsequently, the $N$ samples of $S_1$ is used to approximate the conditional mutual information:

$$\widehat{I}(X;Y|Z) := \frac{1}{N}\sum_{i\in S_1} \log \widehat{P}_{X,Y,Z}(\mathbf{X}_i) + \log \widehat{P}_Z(\mathbf{X}_i) - \log \widehat{P}_{X,Z}(\mathbf{X}_i) - \log \widehat{P}_{Y,Z}(\mathbf{X}_i).$$

For more details corresponding this estimator including its bias, variance, and confidence, please see Loftsgaarden et al. (1965), Sricharan et al. (2011).

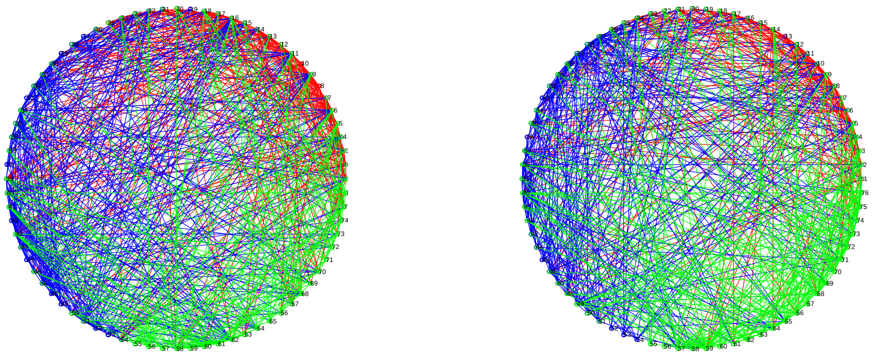
### 4.5.4 DIG of the Financial Market

In this section, we learned the DIG of the aforementioned financial institutions by estimating the directed information quantities in (4.1). To do so, we divided the data into four sectors each of length almost 36 months, 2006-2008, 2009-2011, 2011-2013, and 2013-2016. We assumed that the DIG of the network did not change over each of these time periods. Furthermore, the data collected per working day are assumed to be i.i.d.. Hence, in this experiment the length of each time series was almost 36 and for each time instance we had nearly 19 independent realizations.

As we discussed in Section 4.2.2, in order to identify the influence from node $i$ on node $j$, we need to estimate $I(R_i \to R_j||\underline{R}_{-\{i,j\}})$, which in this experiment, required estimating a joint distribution of dimension 76. In general, without any knowledge about the underlying distribution, estimating such object requires a large amount of independent samples. Unfortunately, in this experiment, we had limited number of independent samples. Thus, we reduced the dimension by instead of conditioning on $\underline{R}_{-\{i,j\}}$ that is a set of size 74, we conditioned on a smaller subset $\underline{K}_{i,j}$ of $\underline{R}_{-\{i,j\}}$ with size 7. This set contained only those institutions with highest correlation with $R_j$. In another words, we ordered the institutions in $\underline{R}_{-\{i,j\}}$ based on their correlation value with $R_j$, and picked the first 7 of them. Afterward, we estimated $I(R_i \to R_j||\underline{K}_{i,j})$ to identify the connection between $R_i$ and $R_j$.

Figures 4.5.2 and 4.5.3 show the resulting graphs. Note that the type of institution causing the relationship is indicated by color: green for brokers, red for insurers, and blue for banks.

In order to compare our results with other methods in the literature, we also learned the causal network of these financial institutions by assuming linear relationships between the institutions and applying linear regression. Similarly, we reduced the dimension of the regressions by bounding the number of incoming arrows of each node to be a subset of size 20. More precisely, we picked 20 most correlated institutions with node $i$, let say $\{R_{j_1}, \dots, R_{j_{18}}\}$ and obtained the parents of $i$ by solving $\min_{a_j} \sum_t |R_{i,t} - \sum_{k=1}^{18} a_k R_{j_k,t-1}|^2$ The resulting graphs are depicted in Figures 4.5.4 and 4.5.5.



(a) January 2006 to December 2008

(b) January 2009 to December 2011

**Figure 4.5.2:** Recovered DIG of the daily returns of the financial companies in Table 4.5.1. The type of institution causing the relationship is indicated by color: green for brokers, red for insurers, and blue for banks.

From these networks, we constructed the following network-based measures of systemic risk.

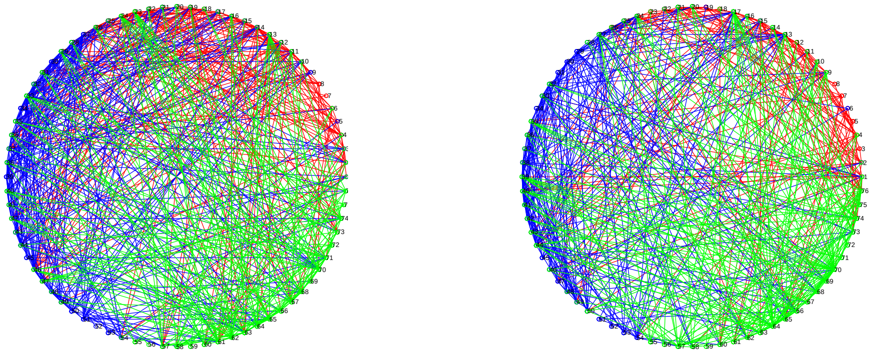
We calculated the fraction of statistically significant Granger causality relationships among all pairs of financial institutions. This is known as the degree of Granger causality (DGC) and it is a measure of the risk of a system event Billio et al. (2012). Table 4.5.2 presents the DGC values and total number of connections of the DIGs and the networks obtain by linear regression.

**Table 4.5.2:** . DGC values and total number of connections.

| **DIGs** | | |
|---|---|---|
| 2006-2008 | 0.1225 | 698 |
| 2009-2011 | 0.1114 | 635 |
| 2011-2013 | 0.1065 | 607 |
| 2013-2016 | 0.0930 | 530 |

| **Linear Models** | | |
|---|---|---|
| 2006-2008 | 0.1453 | 828 |
| 2009-2011 | 0.1288 | 734 |
| 2011-2013 | 0.1174 | 669 |
| 2013-2016 | 0.1216 | 693 |

In order to assess the systemic importance of single institutions, we computed the number of financial institutions that are caused by institution *i* and also the



(a) January 2011 to December 2013

(b) January 2013 to June 2016

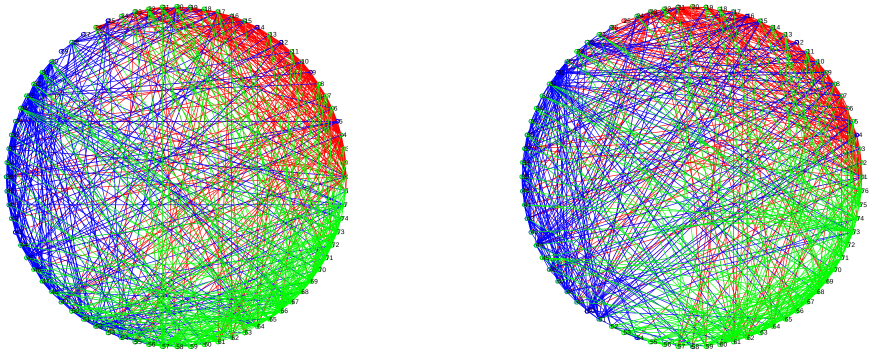
**Figure 4.5.3:** Recovered DIG of the daily returns of the financial companies in Table 4.5.1. The type of institution causing the relationship is indicated by color: green for brokers, red for insurers, and blue for banks.

number of financial institutions that are causing institution *i*. Figure 4.5.6 demonstrates the average number of out-degree and in-degree distributions of the DIGs. Correspondingly, Figure 4.5.7 shows these quantities for the networks obtain by linear regression.

Tables 4.5.3 and 4.5.4 represent the average number of connections between the sectors e.g., 0.1719 fraction of connections are from Banks to Insurances during 2006-2008 in the DIG.

**Table 4.5.3:** . Average number of connections between different sectors in the DIGs.

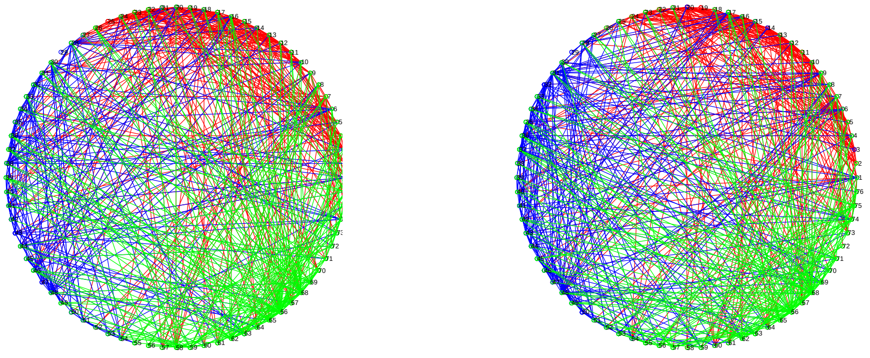
| | **2006-2008** | | | **2009-2011** | | | **2011-2013** | | | **2013-2016** | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Ins. | Ba. | Br. | Ins. | Ba. | Br. | Ins. | Ba. | Br. | Ins. | Ba. | Br. |
| Insurance | .1390 | .1719 | .1074 | .1291 | .1575 | .1213 | .1054 | .1301 | .1104 | .1075 | .1151 | .1340 |
| Bank | .1361 | .1332 | .0702 | .0866 | .1402 | .1039 | .1417 | .1631 | .1021 | .0774 | .1830 | .1302 |
| Broker | .0774 | .1017 | .0630 | .0740 | .929 | .0945 | .0906 | .0873 | .0692 | .0774 | .0774 | .0981 |



(a) January 2006 to December 2008

(b) January 2009 to December 2011

**Figure 4.5.4:** Recovered network of the daily returns of the financial companies in Table 4.5.1 using linear regression. The type of institution causing the relationship is indicated by color: green for brokers, red for insurers, and blue for banks.



(a) January 2011 to December 2013

(b) January 2013 to June 2016

**Figure 4.5.5:** Recovered network of the daily returns of the financial companies in Table 4.5.1 using linear regression. The type of institution causing the relationship is indicated by color: green for brokers, red for insurers, and blue for banks.

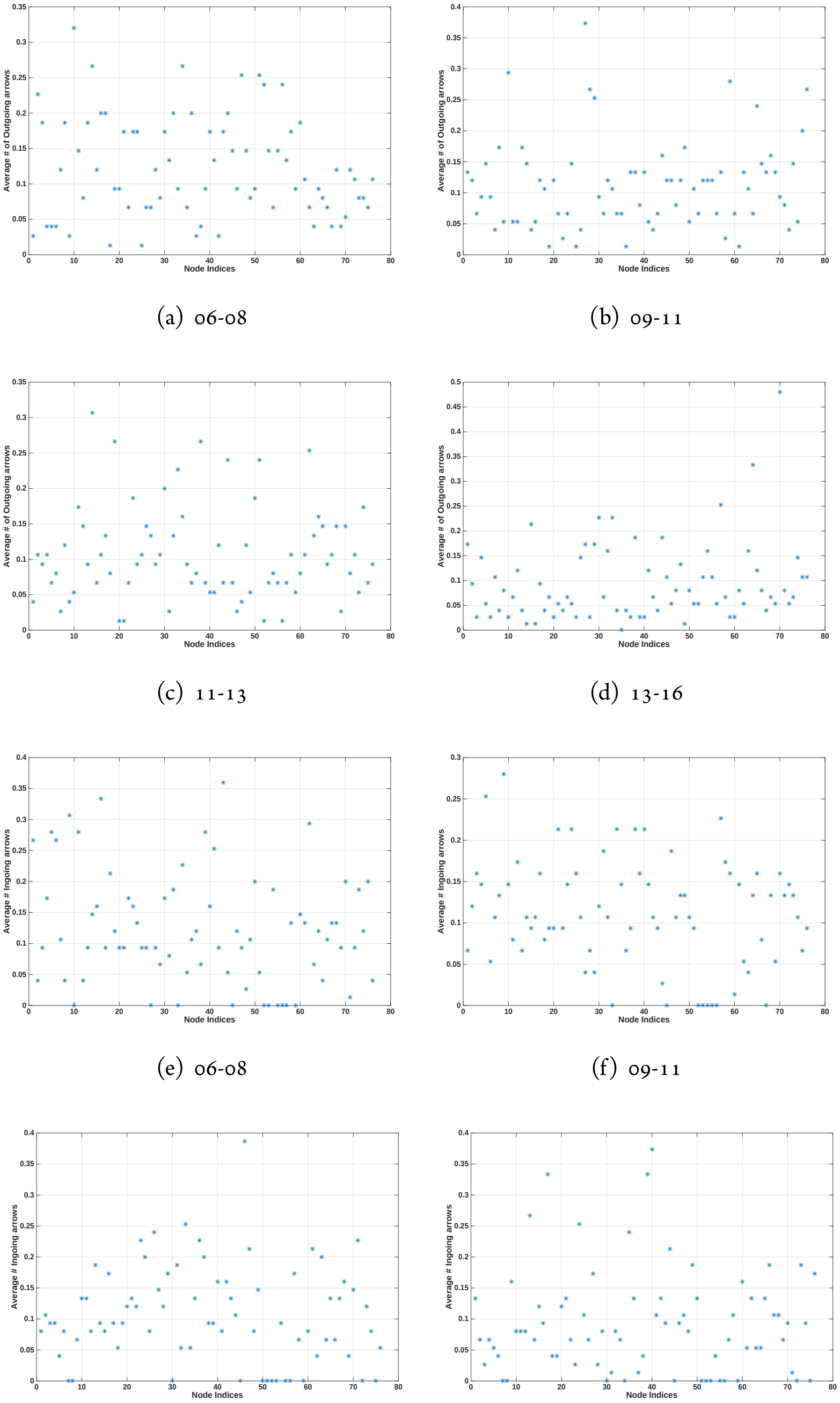
## 4.6 Conclusion

In this work, we developed a data-driven econometric framework to understand the relationship between financial institutions using a non-linearly modified Granger-

**Table 4.5.4:** . Average number of connections between different sectors in the networks obtained using regression.

| | 2006-2008 | | | 2009-2011 | | | 2011-2013 | | | 2013-2016 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Ins. | Ba. | Br. | Ins. | Ba. | Br. | Ins. | Ba. | Br. | Ins. | Ba. | Br. |
| Insurance | .1896 | .0688 | .0737 | .1785 | .1076 | .0640 | .2033 | .0792 | .1016 | .2107 | .0851 | .0678 |
| Bank | .0906 | .1872 | .0809 | .1322 | .1431 | .0899 | .1136 | .1226 | .1001 | .1010 | .1515 | .1053 |
| Broker | .0857 | .1063 | .1171 | .0790 | .0708 | .1349 | .1226 | .0673 | .0897 | .1082 | .0895 | .0808 |

causality. Unlike existing literature, it is not focused on a linear pairwise estimation. The proposed method allows for nonlinearity and it does not suffer from pairwise comparison to identify the causal relationships between financial institutions. We also show how the model improve the measurement of systemic risk and explain the link between Granger-causality and variance decomposition. We apply the model to the daily returns of U.S. financial Institutions including banks, broker, and insurance companies to identify the level of systemic risk in the financial sector and the contribution of each financial institution.



(a) 06-08

(b) 09-11

(c) 11-13

(d) 13-16

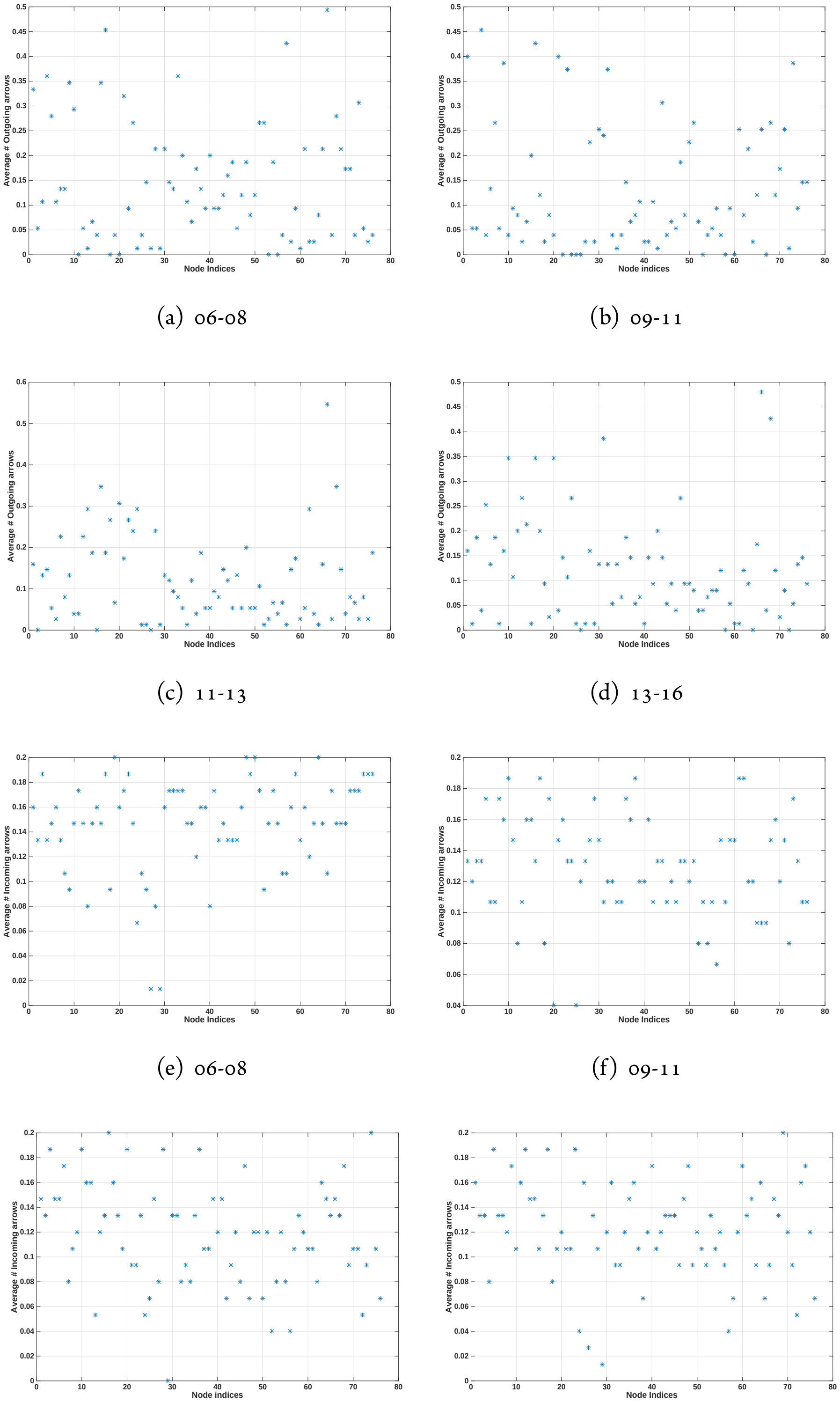
(e) 06-08

(f) 09-11

(g) 11-13

(h) 13-16

**Figure 4.5.6:** Out and In degree distributions of the DIGs obtained in Section 4.5.4.



(a) 06-08

(b) 09-11

(c) 11-13

(d) 13-16

(e) 06-08

(f) 09-11

(g) 11-13

(h) 13-16

**Figure 4.5.7:** Out and In degree distributions of the networks obtained using linear regression.

## 4.7 Appendix

### 4.7.1 Appendix A. Proof of Proposition 1

Note that in this model, since the variance of each $e_{i,t}$ is $\mathcal{F}_i^{t-1}$-measurable, the only term that contains the effect of the other returns on the $i$-th return is $\mu_{i,t}$. Hence, if (4.7) holds, then $\mu_{i,t}$ is independent of $R_j$. This implies the result. Moreover, when $\mu_{i,t} = \sum_{k=1}^{p}\sum_{l=1}^{m} a_{i,l}^{(k)} R_{l,t-k}$, using the result in Etesami and Kiyavash (2014), we declare $R_j$ affects $R_i$ if and only if $\sum_{k=1}^{p}\sum_{l=1}^{m} |a_{i,l}^{(k)}| > 0$, where $a_{i,l}^{(k)}$ denotes the $(j,l)$-th entry of matrix $\mathbf{A}_k$ in (4.5).

### 4.7.2 Appendix B. Proof of Proposition 2

First, we need to show that if there is no arrow from $R_j$ to $R_i$ in the corresponding DIG, then (4.7) and (4.8) hold. This case is straight forward, since when $I(R_j \to R_i||\underline{R}_{-\{i,j\}}) = 0$, then for all $t$, $R_{i,t}$ is independent of $R_j$ given $\mathcal{F}_{-\{j\}}^{t-1}$. This concludes both (4.7) and (4.8).

To show the converse, we use the fact that in multivariate GARCH model, $\vec{R}_t|\mathcal{F}^{t-1}$ is a multivariate Gaussian random process. Thus, if the corresponding mean and variance of $R_{i,t}$ do not contain any influence of $R_j^{t-1}$ given the rest of the network, then $R_{i,t}$ is independent of $R_j^{t-1}$ given $\underline{R}_{-\{j\}}^{t-1}$. This holds if both conditions in (4.7) and (4.8) that are corresponding to the mean and the variance, respectively, are satisfied.

### 4.7.3 Appendix C. Proof of Proposition 3

Suppose the conditions in Proposition 3 hold. We show that $I(Y_j \to Y_i||\underline{Y}_{-\{i,j\}}) = 0$.

$$P(Y_{i,t}|\underline{Y}^{t-1}) = \sum_{S_{i,t}} P(Y_{i,t}|\underline{Y}^{t-1}, S_{i,t})P(S_{i,t}|\underline{Y}^{t-1})$$
$$= \sum_{S_{i,t}} P(Y_{i,t}|\underline{Y}_{-\{j\}}^{t-1}, S_{i,t})P(S_{i,t}|\underline{Y}^{t-1}).$$

The second equality holds because given $S_{i,t}$, $Y_{i,t}$ is a linear function of $\left(\mu_i(S_{i,t}), \vec{Y}_{t-p}, \ldots, \vec{Y}_{t-1}\right)$ plus the error term $\varepsilon_{i,t}$. From the first and second conditions in Proposition 3, we have the coefficients corresponding to $Y_j$ are zero and also the error term is independent of $Y_j$. Thus, $Y_{i,t}$ is independent of $Y_j^{t-1}$ given $\underline{Y}^{t-1}_{-\{j\}}, S_{i,t}$.
If we show $P(S_{i,t}|\underline{Y}^{t-1}) = P(S_{i,t}|\underline{Y}^{t-1}_{-\{j\}})$, using the above equality, we obtain that $P(Y_{i,t}|\underline{Y}^{t-1}) = P(Y_{i,t}|\underline{Y}^{t-1}_{-\{j\}})$ for all $t$. This implies $I(Y_j \to Y_i||\underline{Y}_{-\{i,j\}}) = 0$. To do so, we have

$$
\begin{aligned}
&P(S_{i,t}|\underline{Y}^{t-1}) = \sum_{S_{i,t-1}} P(S_{i,t}|\underline{Y}^{t-1}, S_{i,t-1})P(S_{i,t-1}|\underline{Y}^{t-1})\\
&= \sum_{S_{i,t-1}} P(S_{i,t}|\underline{Y}^{t-1}_{-\{j\}}, S_{i,t-1})P(S_{i,t-1}|\underline{Y}^{t-1})\\
&= \sum_{S_{i,t-1}} P(S_{i,t}|\underline{Y}^{t-1}_{-\{j\}}, S_{i,t-1})P(S_{i,t-1}|\underline{Y}^{t-1}_{-\{j\}}) = P(S_{i,t}|\underline{Y}^{t-1}_{-\{j\}}).
\end{aligned}
$$

The second equality is due to condition three and the fact that $\vec{S}_t$ is conditionally independent of $\underline{Y}_{t-1}$ given $\underline{S}_{t-1}$. The third equality is due to the following

$$
\begin{aligned}
&P(S_{i,t-1}|\underline{Y}^{t-1}) = P\left(S_{i,t-1}|\underline{Y}^{t-2}, Y_{i,t-1}, \underline{Y}_{-\{i,j\},t-1}, Y_{j,t-1}\right)\\
&= P\left(S_{i,t-1}|\underline{Y}^{t-2}, F_i(\underline{Y}^{t-2}_{-\{j\}}, S_{i,t-1}), \underline{Y}_{-\{i,j\},t-1}, F_j(\underline{Y}^{t-2}, S_{j,t-1})\right)\\
&= P\left(S_{i,t-1}|\underline{Y}^{t-2}, F_i(\underline{Y}^{t-2}_{-\{j\}}, S_{i,t-1}), \underline{Y}_{-\{i,j\},t-1}\right),
\end{aligned}
$$

where $F_j$s represent the functional dependency between time series given in (4.15), i.e., $Y_{m,t-1} := F_m(\underline{Y}^{t-2}, S_{m,t-1})$. The above equality holds due to the third condition that states are mutually independent and the fact that all the $Y_j$'s coefficients are zero in $Y_i$'s equation. Same reasoning implies

$$\begin{aligned}
&P\left(S_{i,t-1}|\underline{Y}^{t-2}, F_i(\underline{Y}^{t-2}_{-\{j\}}, S_{i,t-1}), \underline{Y}_{-\{i,j\},t-1}\right) \\
&= P\left(S_{i,t-1}|\underline{Y}^{t-3}, F_i(\underline{Y}^{t-2}_{-\{j\}}, S_{i,t-1}), Y_{i,t-2}, \underline{Y}^{t-1}_{-\{i,j\},t-2}, Y_{j,t-2}\right) \\
&= P\left(S_{i,t-1}|\underline{Y}^{t-3}, F_i(\underline{Y}^{t-2}_{-\{j\}}, S_{i,t-1}), F_i(\underline{Y}^{t-3}_{-\{j\}}, S_{i,t-2}), \underline{Y}^{t-1}_{-\{i,j\},t-2}, F_j(\underline{Y}^{t-3}, S_{j,t-2})\right) \\
&= P\left(S_{i,t-1}|\underline{Y}^{t-3}, F_i(\underline{Y}^{t-2}_{-\{j\}}, S_{i,t-1}), F_i(\underline{Y}^{t-3}_{-\{j\}}, S_{i,t-2}), \underline{Y}^{t-1}_{-\{i,j\},t-2}\right) \\
&\quad \vdots \\
&= P\left(S_{i,t-1}|F_i(\underline{Y}^{t-2}_{-\{j\}}, S_{i,t-1}), F_i(\underline{Y}^{t-3}_{-\{j\}}, S_{i,t-2}), \ldots, \underline{Y}^{t-1}_{-\{i,j\}}\right) = P\left(S_{i,t-1}|\underline{Y}^{t-1}_{-\{j\}}\right).
\end{aligned}$$

Recall that $\underline{Y}^{t}_{\mathcal{K},t'}$ denotes the time series with index set $\mathcal{K}$ from time $t'$ up to time $t$.

*The world is one big data problem.*

Andrew McAfee

# 5

# Forecasting in Big Data Environments with a Shrinkage Estimation of Skip-layer Neural Networks

## 5.1 Introduction

One of the important steps in designing the modern predictive models is to cope with high-dimensional data, which contain large numbers of correlated variables and present complex properties. Big data is not just an increase in the number of samples collected over time, but also is an increase in the number of potential explanatory variables and predictors that are simultaneously measured on a process. When the dimension grows, the specificities of high-dimensional spaces and data must then be taken into account in the design of predictive models. While this statement is valid in general, its importance is even higher when using nonlinear tools such as artificial neural networks. Most nonlinear models involve more parameters than the dimension of the data space which may result in a lack of model identifiability, instability, and overfitting (Huber (2011);Cherkassky et al. (1994); Moody (1991)). Therefore, selection of the significant predictors and the level of model complexity are the key tasks of designing accurate predictive models in data-rich environments.

Feature extraction (i.e, principal component analysis - Pearson (1901); Eckart and Young (1936); factor analysis - Spearman (1904); canonical correlations analysis - Hotelling (1936)) and feature selection (i.e, Ridge - Hoerl and Kennard (1970a); LASSO - Tibshirani (1996); Elastic Net - Zou and Hastie (2005)) are broadly the two general approaches for dimensionality reduction, the former transforms the original features into a lower dimensional space preserving all its fundamental characteristics whereas the latter selects a small subset of the original features without a transformation. In this work, our focus is in particular on feature selection techniques, and we apply shrinkage approaches (In machine learning, this is called regularization). We embed feature selection in backpropagation algorithm as part of its overall operation. To be more precise, we penalize the neural network loss function with the $L_1$ norm for weights in the hidden layer and $L_2$ norm for weights in skip-layer[1].

Doing shrinkage is, therefore, an implicitly embedded manner of doing feature selection, which is an example of model selection problem, since only a subset of variables contributes to the final predictor. It has frequently been observed that $L_1$ regularization in many models causes many parameters to equal zero and can result in dropping some features and getting a sparse model, so that only those parameters whose impact on the empirical risk is considerable and consequently appear in the fitted model Ng (2004). This makes it a proper candidate for the nonlinear part to control the complexity of the model and from an optimization point of view it is equivalent to a neural network learned/estimated by LASSO. It prevent hidden

[1] Direct connections from each of the input variables to each of the output variables. This part of the model is equivalent to a linear regression.

units getting stuck near zero and weights exploding. On the other hand, we apply $L_2$ regularization on the skip-layer connections (linear part of the model), which penalizes groups of parameters and encourages the sum of the squares of the parameters to be small. Therefore we will not drop specific features from linear part and there is the possibility of directly interpreting the marginal impact of predictors on target variable. It is worth mentioning that the linear part of the model can be seen as a Ridge regression.

In fact, that's not the only benefit of using regularization. Empirically, regularization is also a way to reduce overfitting and to increase prediction accuracies Ng (2004). This is especially true in modern networks, which often have very large numbers of weights. The proposed algorithm combines the neural network's advantage of describing the nonlinear process with the superior accuracy of feature selection that is provided by penalized loss function combining $L_1$ and $L_2$ norms.

Although many time series studies have suggested neural networks as a promising alternative to the linear regression models, there are empirical evidences showing deceptive results in terms of their superiority in out-of-sample forecasting performance. It is also challenging to determine if complex real world time series behave in a linear or nonlinear fashion. Hence, it is not wise to apply neural networks blindly to any type of data. The experimental results from different linearity tests indicated that the real world series are rarely pure linear or nonlinear. They consists of both linear and nonlinear patterns. If that is the case, we can assume that the financial time series $(y_t)$ are composed of a linear structure $(\mathcal{L}_t)$ plus a nonlinear component $(\mathcal{N}_t)$.

$$y_t = \mathcal{L}_t + \mathcal{N}_t \tag{5.1}$$

The major limitation of a linear model is the pre-assumed linear form of the model and therefore, no nonlinear patterns can be captured. The neural network is not also adequate to handle fairly both linear and nonlinear patterns especially when the linear component is superior to the nonlinear component. The performance of the linear model and neural networks is not remarkable when the time series contain complex linear and nonlinear patterns.

Two different approaches to model and forecast time series with both linear and nonlinear patterns are imaginable. The first approach is a hybrid methodology with combined linear time series models and neural network models which is capable to capture different aspects of the data. In this approach, first we estimate the linear component using a linear model and then we collect the residuals obtained from the fitted model $\hat{e}_t = y_t - \hat{\mathcal{L}}_t$. Finally we let a nonlinear approach (i.e, GARCH family models, Neural networks) to model the residuals which are representing the nonlinear component and will contain information about the nonlinearity. In theory, the hybrid model can be an effective tool with a superior predictive ability when both linear model and neural network model are specified well and are suboptimal. But, in practice we are combining two model specification errors.

There is another approach to model time series with complex patterns that we are proposing in this paper and is based on a neural network with skip-layer connections including both linear and nonlinear structure. Therefore, we optimize a neural network model containing a linear part simultaneously. The model considers both linear and nonlinear patterns in the time series at the same time.

## 5.2 The Model

In this study, we focus on feedforward neural networks with only one hidden layer. And to show that the neural network models can be seen as a generalisation of linear models, we allow for direct connections from the input variables to the output layer and we assume that the output transfer function is linear[2], then the model

[2] Using linear function for the output unit activation function (in conjunction with nonlinear activations amongst the hidden units) allows the network to perform a powerful form of nonlinear regression. So, the network can predict continuous target values using a linear combination of signals that arise from one layer of nonlinear transformations of the input.

becomes

$$y_t = \Phi(x; w) = \sum_{i \to k} x_{it} w_{ik} + \sum_{j \to k} \varphi_j \left( \sum_{i \to j} x_{it} w_{ij} \right) w_{jk} + \varepsilon_t, \qquad (5.2)$$

where $\Phi$ describe network by a vector function. We associate subscript $i$ with the input layer, subscript $j$ with the hidden layer, and subscript $k$ with the output layer. $x_{it} = (x_{1t}, x_{2t}, ..., x_{mn})$ is the value of the $i$th input node, which can be a constant input representing biases, a matrix of lagged values of $y_t$ and some exogenous variables. $\varphi_j(.)$ and $J$ are activation functions and number of neurons used at the hidden layer. A single-hidden-layer neural network with skip-layer connections is shown in Figure 5.2.1. A network with only one hidden layer and skip-layer connections has three sets of weights: those direct connections between the inputs and the output $(w_{ik})$, those connecting the inputs to the hidden layer $(w_{ij})$, and those connecting the output of the hidden layer to the output layer$(w_{jk})$.

First summation of the Eq.(5.2) represents a linear regression term. A linear regression term hints the model in a right direction when we know that the data contains a linear component. Moreover, this is more interpretable from a statistical perspective and unraveling a bit of a structure behind the network, which is usually seen merely as a black box. It also has the advantage that, when we apply shrinkage techniques to estimate network parameters, if the problem is essentially linear, the hidden neurons tend to get pruned and we are left with a linear model.

In general, estimating the set of network elementary parameters (weights, $w$) in a way that minimise the errors that the network makes is known as training/learning neural network. It is equivalent to finding a point in parameter space that makes the height of the error surface small. The error surface gets more and more complicated as we increase the number of input variables in the network and the number of units in hidden layer. The mean squared prediction error, $E = \frac{1}{n} \sum_{t=1}^{n} (y_t - \hat{y}_t)^2$, is a standard way of quantifying error. Given target values and network outputs we can calculate the value of the error function for each setting of weights.

In principle, studies in time series and forecasting widely use the conventional

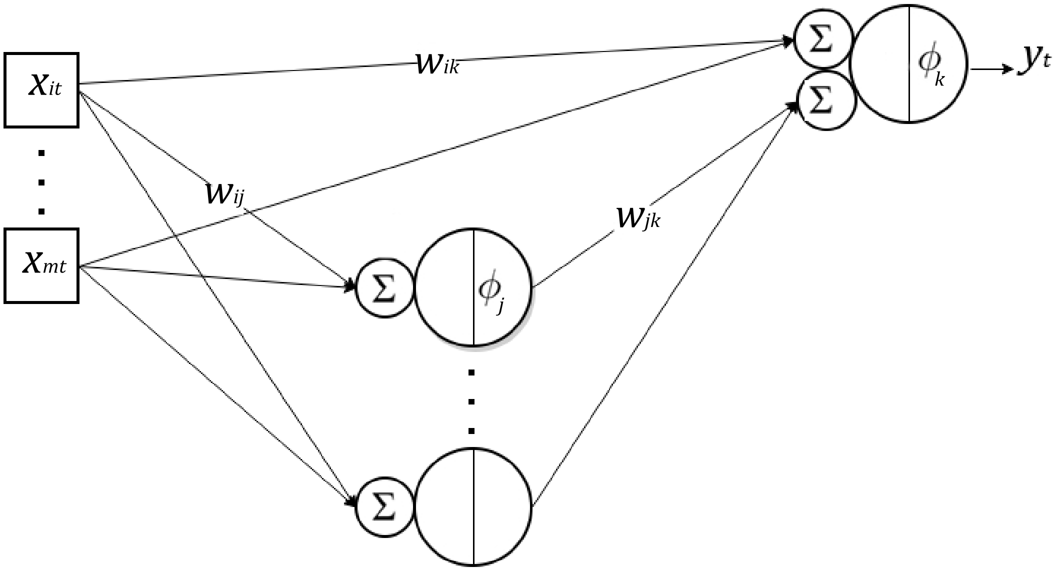


**Figure 5.2.1:** A single-hidden-layer neural network with skip-layer connections

feedforward neural network trained with the gradient descent type algorithm (also known as backpropagation). The backpropagation algorithm first introduced by Bryson et al. (1979) and popularised in the field of artificial neural network research by Werbos (1988) and Rumelhart et al. (1986). The goal of backpropagation learning algorithm is to adjust the weights in a way that minimises the network prediction error function.

To solve this problem, the error function's sensitivity to network weights must be quantified based on a gradient descent optimization. Gradient is normally defined as the first order derivative /gradient of the error function with respect to each of the model parameters. This gradient information will give us the direction in parameter space that decreases the height of the error surface. We then take a step in that direction and repeat, iteratively calculating the gradient and taking steps in parameter space. The weight adjustment is given by

$$w^{new} = w^{old} - \eta \frac{\partial E(w)}{\partial w} \tag{5.3}$$

Where the constant $\eta$ is the learning rate (step size) and its value falls between zero and one. The direction of search in weight space for the new value of the weights is elected by $\frac{\partial E(w)}{\partial w}$, that shows the sensitivity of the error function to the weights. By repeating iteratively the steps network can be trained in a way that converges to optima. The set of new weights are repeatedly presented to the network until the error value is minimised. Around the optimum point, all the elements of the gradient would be very small, which leads to tiny changes in new weights.

Implementing $L_1$ and $L_2$ regularization in a backpropagation algorithm of a neural network is relatively easy. In particular, method error returns the total error plus penalties or constraints and the objective of learning is minimization of the regularized loss function. If weight values are included in the total error term that's being minimized, then smaller weight values will generate smaller error values. Therefore, network parameters are the solutions to the following optimization problem

$$w^* = \underset{w}{\operatorname{argmin}}\, E(w) \; + \; \lambda\, \Omega(w) \tag{5.4}$$

where the regularization term $\Omega(w)$ is multiplied by a shrinkage/regularization hyperparameter $\lambda$. Assuming a fixed $\lambda$, to learn network parameters $w^*$ using backpropagation algorithm, we just need to add derivative of penalty term to the gradient and iteratively update parameters[3].

$$\begin{cases} \Delta = \frac{\partial E(w)}{\partial w} + \lambda\, \frac{\partial \Omega(w)}{\partial w} \\ w^{new} = w^{old} \; - \; \eta \Delta \end{cases} \tag{5.5}$$

Where $\Delta$ is the gradient of regularized loss function and $\lambda \; > \; 0$ is known as the regularization hyperparameter. In many practical applications, the simplest method to set $\lambda$ is to train the neural network with a number of different $\lambda$ values, and then choose the model having the smallest validation error. A more attractive approach is to use an optimization algorithm to adapt the model hyperparameters $\lambda$ automatically. In the next section we will explain how to estimate a set of $\lambda$ for each network weight in order to minimize a validation error during training of

[3] It is worth noting that the regularization term does not include the biases.

neural networks.

$L_1$ sparsity norm and $L_2$ smoothing norm are two closely related regularization that can be used by machine learning training algorithms to select model features and reduce model overfitting. Regularization in neural networks limits the magnitude of network parameters by adding a penalty for weights to the model error function. $L_1$ weight regularization penalizes weight values by adding the sum of their absolute values to the error term. $L_2$ regularization uses the sum of the squared values of the weights. In this study, $L_2$ regularization penalizes weight values in skip-layer connections by adding the sum of their squared values to the error term and $L_1$ regularization penalizes weight values in hidden layer to encourage the topology of the learned network to be sparse.

$$w^* = \underset{w}{\operatorname{argmin}}\, E(w) \;+\; \frac{\lambda_2}{2}\sum_{i\to k} w_{ik}^2 \;+\; \lambda_1\Big(\sum_{i\to j}|w_{ij}| + \sum_{j\to k}|w_{jk}|\Big) \tag{5.6}$$

Then the learning rule for the weights becomes:

$$\begin{cases} w_{ik}^{new} = w_{ik}^{old} \;-\; \eta\left(\frac{\partial E(w)}{\partial w_{ik}} + \lambda_2\, w_{ik}^{old}\right) \\ w_{ij}^{new} = w_{ij}^{old} \;-\; \eta\left(\frac{\partial E(w)}{\partial w_{ij}} + \lambda_1\, sgn(w_{ij}^{old})\right) \\ w_{jk}^{new} = w_{jk}^{old} \;-\; \eta\left(\frac{\partial E(w)}{\partial w_{jk}} + \lambda_1\, sgn(w_{jk}^{old})\right) \end{cases} \tag{5.7}$$

This is exactly the same as the usual gradient descent learning rule, except we rescale the weights that makes them smaller. Intuitively, the effect of penalty terms is to lead the network to learn small weights, all other things being equal. Large weights will only be allowed if they considerably improve the first part of the cost function. The relative importance of the compromise between finding small weights and minimizing the original loss function depends on the size of $\lambda$. When $\lambda$ is small we prefer to minimize the original loss function, but when it is large we prefer small weights. There are different ways to tune hypeparameters such as $\lambda$. We explain later how to estimate $\lambda$ instead of setting that manually using grid search or cross-validation.

To use $L_2$ regularization, we add a $\lambda_2 w$ term to the gradient as the derivative of $w^2$ is $2w$. $L_2$ regularization works with all forms of learning algorithm, but does not provide implicit feature selection. The derivative of the absolute value of $w$ is $w/|w|$, however $L_1$ norm is not differentiable at 0 and hence poses a problem for gradient-based methods.

The problem can be solved using the exact gradient, which is discontinuous at 0. We can also solve the problem by the smooth approximation approach which will allow us to use gradient descent. To smooth out the $L_1$ norm using an approximation, we use $\sqrt{w^2 + \varepsilon}$ place of $|w|$ , where $\varepsilon$ is a smoothing parameter which can also be interpreted as a sort of sparsity parameter. When $\varepsilon$ is large compared to $w$, the $w + \varepsilon$ is dominated by $\varepsilon$ and taking the squared root yields approximately $\sqrt{\varepsilon}$. Lee et al. (2006)

## 5.3 Gradient-based Hyperparameter Optimization

The major downsides of using shrinkage method is that it introduces additional hyperparameters that must be determined. In practice we have two set of parameters: model elementary parameters (network weights), and learning algorithm hyperparameters (lambdas; size of an $L_1$ or $L_2$ penalty). We would ideally like to determine these hyperparameters to get optimal generalization[4]. There are different approaches to this problem. Grid search and manual search are the most widely used strategies for hyperparameter optimization in the literature. However, in many practical applications manually searching the space of hyperparameter settings is tedious and tends to lead to unsatisfactory outcomes. Bergstra and Bengio (2012) show empirically and theoretically that random search is more efficient than grid search for hyperparameter optimization in the case of several learning algorithms on several data sets. Statistical techniques such as cross-validation Wahba (1990), bootstrapping Efron and Tibshirani (1994), and the Bayesian method MacKay (1992) can also assist in terms of determining the hyperparameters.

[4] Generalization means building a model on one set of training data and hope that it makes effective predictions on a different set of test data.

Cross-validation (CV) is popular practical method to select hyperparameters. The rationale behind CV is to split the data into the training sample used for learning the algorithm, and the validation sample (once or several) for estimating the risk of each algorithm and for evaluating its performance. Validation sample is not used for training, but are used to evaluate how well the model generalizes to data it hasn't seen before. In brief, CV consists in averaging several hold-out estimators (folds) of the risk corresponding to different splits of the data and selects the algorithm with the smallest estimated risk. Within each fold, hyperparameters are fixed and we only estimate model elementary parameters. The validation sample plays the role of new data as soon as data are i.i.d.[5] For a general description of the CV strategy See **?**, and Arlot and Celisse (2010) for a comprehensive review on cross-validation procedures and their applications in different algorithms and frameworks. However, Several studies such as Rivals and Personnaz (1999) show that CV performance is not always good.

Recently, automated approaches for estimation of the hyperparameters has been proposed in the literature and led to substantial improvements specially when the researcher did not have a strong intuition regarding good values to try for the hyperparameters. There are a number of gradient-free automated optimization methods (Hutter et al. (2011); Bergstra et al. (2011); Bergstra et al. (2013); Snoek et al. (2012)), all of which rely on multiple complete training runs with varied fixed hyperparameters, with the hyperparameter selection based on the validation set performance. Hyperparameters are chosen to optimize the validation loss after complete training of the model parameters.

In the late 1990s, gradient-based automated approaches proposed by Larsen et al. (1996) and Andersen et al. (1997). They formulate an iterative gradient descent scheme for adapting the hyperparameters by minimizing the validation error calculated from a single validation set. They treat hyperparameters similar to elementary parameters during training and simultaneously update both sets of parameters. Following this scheme, we can estimate a single regularization parameter or separate regularization parameter for each individual weight in the network.

[5] This assumptions can be relaxed. see: Chu and Marron (1991).

The details of this approach can be summarized in few steps as follows:

(1) In the first step, we split the data set into two disjoint sets: a validation set for estimation of hyperparameters and optimization of network architecture, and a training set for estimation of network elementary parameters. we will refer to the training set as $\mathcal{T}$, with $n_{\mathcal{T}}$ observations, and to the validation set (used exclusively for training the hyperparameters) as $V$, with $n_V$ observations.

(2) Then we initialize the weights and $\lambda$ to train the network with fixed $\lambda$ to achieve $\hat{w}(\lambda)$. The validation error $E_V$ of the trained network is given by

$$E_V(\hat{w}(\lambda)) = \frac{1}{n_V} \sum_{t=1}^{n_V} (y_t - \hat{y}_t)^2 \tag{5.8}$$

Where $\hat{w}(\lambda)$ is the $\lambda$-dependent vector of weights estimated from the training set $\mathcal{T}$. Validation error is a function of network weights, and the network weights are affected by the hyperparameters through the regularized loss function. Therefore, the validation error is an implicit function of $\lambda$.

(3) The last step includes updating $\lambda$ using the gradient of validation error until the relative change in validation error is below a small percentage. After each update, the network is again trained to convergence. The optimal $\lambda$ can be found by a gradient descent scheme as follow

$$\lambda^{new} = \lambda^{old} - \gamma \frac{\partial E_V}{\partial \lambda} (\hat{w}(\lambda^{old})) \tag{5.9}$$

Where $\gamma > 0$ is the gradient step size (learning rate) and $\hat{w}(\lambda^{old})$ is the estimated weight vector using $\lambda$ from previous iteration.

Similar approaches have been proposed since the late 1990s; Larsen et al. (2012) extended their framework to a multi-fold validation sets. Chen and Hagan (1999) extended the gradient descent hyperparameter optimization by introducing a second derivative of validation error based regularization algorithm using the Gauss-Newton approximation to the Hessian. Bengio (2000) shows that the implicit function theorem can be used to derive a formula for the hyperparameter gradient involving second derivatives of the training criterion. His method require computation of the inverse Hessian. Snoek et al. (2012) optimize separate regularization parameters for each layer in a neural network, and found that it improved performance. Maclaurin et al. (2015) compute exact gradients of cross-validation performance with respect to all hyperparameters by chaining derivatives backwards through the entire training procedure. They compute hyperparameter gradients by exactly reversing the dynamics of stochastic gradient descent with momentum.

## 5.4 Concluding Remarks

In this study, we suggested a high-dimentional learning algorithm of a neural network with skip-layer connections as an accurate predictive model in a data-rich environment. We explained how skip-layer connections hints the model in a right direction when the data contains both linear and nonlinear components. To overcome the curse of dimensionality and to manage model complexity, we penalized the model loss function with $L_1$ and $L_2$ norms. Setting the size of regularization is still an open question. Recent studies proposed automated approaches for estimation of algorithm hyperparameters. We briefly explained the gradient-based automated approaches which treats shrinkage hyperparameters similar to the network weights during training and simultaneously optimize both sets of parameters.

# Bibliography

Acharya, V. V., Pedersen, L. H., Philippon, T., and Richardson, M. P. (2010). Measuring systemic risk.

Adrian, T. and Brunnermeier, M. C. (2008). Staff report no348. *Federal Reserve Bank of New York.*

Allen, F., Babus, A., and Carletti, E. (2010). Financial connections and systemic risk. Technical report, National Bureau of Economic Research.

Andersen, L. N., Larsen, J., Hansen, L. K., and Hintz-Madsen, M. (1997). Adaptive regularization of neural classifiers. *Proceedings of the IEEE Workshop on Neural Networks for Signal Processing,* VII:24–33.

Arlot, S. and Celisse, A. (2010). A survey of cross-validation procedures for model selection. *Statistics Surveys,* 4(0):40–79.

Armstrong, J. and Collopy, F. (1992). Error measures for generalizing about forecasting methods: Empirical comparisons. *International Journal of Forecasting,* 8(1):69–80.

Arranz, M. A. (2005). Portmanteau test statistics in time series, tol-project.

Ashley, R. A. and Patterson, D. M. (2001). Nonlinear model specification diagnostics: Insights from a battery of nonlinearity tests. *Working Paper E99-05, Department of Economics, Virginia Tech.*

Bacry, E., Dayri, K., and Muzy, J.-F. (2012). Non-parametric kernel estimation for symmetric hawkes processes. application to high frequency financial data. *The European Physical Journal B,* 85(5):1–12.

Bacry, E., Delattre, S., Hoffmann, M., and Muzy, J.-F. (2013a). Modelling microstructure noise with mutually exciting point processes. *Quantitative Finance,* 13(1):65–77.

Bacry, E., Delattre, S., Hoffmann, M., and Muzy, J.-F. (2013b). Some limit theorems for hawkes processes and application to financial statistics. *Stochastic Processes and their Applications*, 123(7):2475–2499.

Bacry, E. and Muzy, J.-F. (2014a). Hawkes model for price and trades high-frequency dynamics. *Quantitative Finance*, 14(7):1147–1166.

Bacry, E. and Muzy, J.-F. (2014b). Second order statistics characterization of hawkes processes and non-parametric estimation. *preprint arXiv:1401.0903*.

Bai, J. and Ng, S. (2002). Determining the number of factors in approximate factor models. *Econometrica*, 70(1):191–221.

Bai, J. and Ng, S. (2006). Evaluating latent and observed factors in macroeconomics and finance. *Journal of Econometrics*, 131(1-2):507–537.

Bai, J. and Ng, S. (2008a). Forecasting economic time series using targeted predictors. *Journal of Econometrics*, 146(2):304–317.

Bai, J. and Ng, S. (2008b). Large dimensional factor analysis. *Foundations and Trends®in Econometrics*, 3(2):89–163.

Barigozzi, M. and Hallin, M. (2015). Networks, dynamic factors, and the volatility analysis of high-dimensional financial series. *arXiv preprint arXiv:1510.05118*.

Barigozzi, M. and Hallin, M. (2016). A network analysis of the volatility of high dimensional financial series. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*.

Belkin, M. and Niyogi, P. (2003). Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396.

Bendat, J. S. and Piersol, A. G. (1993). *Engineering Applications of Correlation and Spectral Analysis*. Wiley, John & Sons, New York, 2 edition.

Bengio, Y. (2000). Gradient-based optimization of hyperparameters. *Neural Computation*, 12(8):1889–1900.

Berg, A., McMurry, T., and Politis, D. N. (2012). Testing time series linearity. In *Time Series Analysis: Methods and Applications*, pages 27–42. Elsevier BV.

Berg, A., Paparoditis, E., and Politis, D. N. (2010). A bootstrap test for time series linearity. *Journal of Statistical Planning and Inference*, 140(12):3841–3857.

Bergstra, J., Bardenet, R., Bengio, Y., and Kégl, B. (2011). Algorithms for hyper-parameter optimization. *Advances in Neural Information Processing Systems*, 24:2546–2554.

Bergstra, J. and Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13:281–305.

Bergstra, J., Yamins, D., and Cox, D. (2013). Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. *International Conference on Machine Learning*, page 115–123.

Billio, M. and Di Sanzo, S. (2015). Granger-causality in markov switching models. *Journal of Applied Statistics*, 42(5):956–966.

Billio, M., Getmansky, M., Lo, A. W., and Pelizzon, L. (2010). Measuring systemic risk in the finance and insurance sectors.

Billio, M., Getmansky, M., Lo, A. W., and Pelizzon, L. (2012). Econometric measures of connectedness and systemic risk in the finance and insurance sectors. *Journal of Financial Economics*, 104(3):535–559.

Bishop, C. M. and Hinton, G. (1995). *Neural networks for pattern recognition*. Oxford University Press, New York.

Black, F. and Scholes, M. (1973). The pricing of options and corporate liabilities. *The journal of political economy*, pages 637–654.

Boivin, J. and Ng, S. (2005). Understanding and comparing factor-based forecasts. *International Journal of Central Banking*, 1:3:117–152.

Bollerslev, T., Engle, R. F., and Nelson, D. B. (1994). Arch models. *Handbook of Econometrics*, Volume IV(Elsevier Science):2961–3031.

Borg, I. and Groenen, P. J. F. (2005). Modern multidimensional scaling: Theory and applications. *Springer-Verlag, New York*, pages 277–280.

Bowsher, C. G. (2007). Modelling security market events in continuous time: Intensity based, multivariate point process models. *Journal of Econometrics*, 141(2):876–912.

Box, G. E. P. and Pierce, D. A. (1970). Distribution of residual autocorrelations in autoregressive-integrated moving average time series models. *Journal of the American Statistical Association*, 65(332):1509–1526.

Breakspear, M., Brammer, M., and Robinson, P. A. (2003). Construction of multivariate surrogate sets from nonlinear data using the wavelet transform. *Physica D: Nonlinear Phenomena*, 182(s 1–2):1–22.

Breiman, L., Friedman, J., and Stone, C. J. (1984). *Classification and regression trees*. Chapman and Hall/CRC, New York, NY.

Brémaud, P. and Massoulié, L. (1996). Stability of nonlinear hawkes processes. *The Annals of Probability*, pages 1563–1588.

Brillinger, D. R. and Rosenblatt, M. (1967). Asymptotic theory of estimates of k-th order spectra. *Proceedings of the National Academy of Sciences*, 57(2):206–210.

Brock, W. A., Dechert, W. D., and Scheinkman, J. A. (1987). A test for independence based on the correlation dimension. *WP University Wisconsin*.

Brock, W. A., Hsieh, D. A., and LeBaron, B. D. (1991). *Nonlinear dynamics, chaos, and instability: Statistical Theory and Economic Evidence*. MIT Press, Cambridge, MA.

Brock, W. A., Scheinkman, J. A., Dechert, W. D., and LeBaron, B. (1996). A test for independence based on the correlation dimension. *Econometric Reviews*, 15(3):197–235.

Brockett, P. L., Hinich, M. J., and Patterson, D. (1988). Bispectral-based tests for the detection of gaussianity and linearity in time series. *Journal of the American Statistical Association*, 83(403):657.

Brooks, C. (1996). Testing for non-linearity in daily sterling exchange rates. *Applied Financial Economics*, 6(4):307–317.

Bryson, A. E., Denham, W. F., and Dreyfus, S. E. (1963). Optimal programming problems with inequality constraints. *AIAA Journal*, 1(11):2544–2550.

Bryson, A. E., Ho, Y.-C., and Siouris, G. M. (1979). Applied optimal control: Optimization, estimation, and control. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(6):366–367.

Burges, C. J. C. (2004). Geometric methods for feature extraction and dimensional reduction: A guided tour. *Technical Report*, MSR-TR-2004-55, Microsoft Research.

Bühlmann, P. (2002). Bootstraps for time series. *Statistical Science*, 17(1):52–72.

Caillec, J.-M. L. and Garello, R. (2004). Comparison of statistical indices using third order statistics for nonlinearity detection. *Signal Processing*, 84(3):499–525.

Campbell, J. Y., Lo, A. W., and MacKinlay, C. A. (1997). *The Econometrics of Financial Markets*. Princeton University Press, United States, 2 edition.

Cario, M. C. and Nelson, B. L. (1996). Autoregressive to anything: Time-series input processes for simulation. *Operations Research Letters*, 19(2):51–58.

Cayton, L. (2005). Algorithms for manifold learning. *UCSD Technical Report*, CS2008-0923.

Cesa-Bianchi, N. and Lugosi, G. (2006). *Prediction, learning, and games*. Cambridge university press.

Chamberlain, G. and Rothschild, M. (1983). Arbitrage, factor structure, and mean-variance analysis on large asset markets. *Econometrica*, 51(5):1281.

Chan, K. S. and Tong, H. (1986). On estimating thresholds in autoregressive models. *Journal of Time Series Analysis*, 7(3):179–190.

Chávez, M., Martinerie, J., and Le Van Quyen, M. (2003). Statistical assessment of nonlinear causality: application to epileptic eeg signals. *Journal of neuroscience methods*, 124(2):113–128.

Chen, D. and Hagan, M. T. (1999). Optimal use of regularization and cross-validation in neural network modeling. page 1275–1289, International Joint Conference on Neural Networks.

Chen, N.-F., Roll, R., and Ross, S. A. (1986). Economic forces and the stock market. *The Journal of Business*, 59(3):383.

Cherkassky, V., Friedman, J. H., and Wechsler, H., editors (1994). *From statistics to neural networks: Theory and pattern recognition applications*. Springer-Verlag Berlin and Heidelberg GmbH & Co. K, Berlin.

Christiansen, B. (2005). The shortcomings of nonlinear principal component analysis in identifying circulation regimes. *Journal of Climate*, 18(22):4814–4823.

Chu, C.-K. and Marron, J. S. (1991). Comparison of two bandwidth selectors with dependent errors. *The Annals of Statistics*, 19(4):1906–1918.

Chui, C. K. and Li, X. (1992). Approximation by ridge functions and neural networks with one hidden layer. *Journal of Approximation Theory*, 70(2):131–141.

Collis, W., White, P., and Hammond, J. (1998). Higher-order spectra: the bispectrum and trispectrum. *Mechanical Systems and Signal Processing*, 12(3):375–394.

Cont, R. (2001). Empirical properties of asset returns: Stylized facts and statistical issues. *Quantitative Finance*, 1(2):223–236.

Cover, T. M. and Thomas, J. A. (2012). *Elements of information theory*. John Wiley & Sons.

Cox, M. A. A. and Cox, T. F. (2001). Multidimensional scaling. *Springer Handbooks Comp.Statistics*, pages 315–347.

Cunningham, J. P. and Ghahramani, Z. (2015). Linear dimensionality reduction: Survey, insights, and generalizations. *Journal of Machine Learning Research*, 16:2859–2900.

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems*, 2(4):303–314.

Dahlhaus, R. and Eichler, M. (2003). Causality and graphical models in time series analysis. *Oxford Statistical Science Series*, pages 115–137.

Davies, N. and Newbold, P. (1979). Some power studies of a portmanteau test of time series model specification. *Biometrika*, 66(1):153–155.

Deistler, M. and Hamman, E. (2005). Identification of factor models for forecasting returns. *Journal of Financial Econometrics*, 3(2):256–281.

DeMiguel, V., Garlappi, L., and Uppal, R. (2009). Optimal versus naive diversification: How inefficient is the 1/ n portfolio strategy? *Review of Financial Studies*, 22(5):1915–1953.

Diebold, F. X. and Mariano, R. S. (1995). Comparing predictive accuracy. *Journal of Business & Economic Statistics*, 13(3):253.

Diebold, F. X. and Yılmaz, K. (2014). On the network topology of variance decompositions: Measuring the connectedness of financial firms. *Journal of Econometrics*, 182(1):119–134.

Dufour, J.-M. and Taamouti, A. (2010). Short and long run causality measures: Theory and inference. *Journal of Econometrics*, 154(1):42–58.

Eckart, C. and Young, G. (1936). The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218.

Efron, B. and Tibshirani, R. J. (1994). *An introduction to the Bootstrap, Vol. 57*. Chapman & Hall/CRC, Boca Raton, FL.

Engle, R. and Colacito, R. (2006). Testing and valuing dynamic correlations for asset allocation. *Journal of Business & Economic Statistics*, 24(2):238–253.

Engle, R. and Kelly, B. (2012). Dynamic equicorrelation. *Journal of Business & Economic Statistics*, 30(2):212–228.

Engle, R. F. (1982). Autoregressive conditional heteroscedasticity with estimates of the variance of united kingdom inflation. *Econometrica*, 50(4):987.

Etesami, J. and Kiyavash, N. (2014). Directed information graphs: A generalization of linear dynamical graphs. In *American Control Conference (ACC)*, pages 2563–2568. IEEE.

Exterkate, P., Groenen, P. J. F., Heij, C., and Dijk, D. J. C. V. (2013). Nonlinear forecasting with many predictors using kernel ridge regression. *CREATES Aarhus University and Erasmus University*, Working paper.

Fama, E. F. and French, K. R. (2004). The capital asset pricing model: Theory and evidence. *Journal of Economic Perspectives*, 18(3):25–46.

Fan, J. and Yao, Q. (2005). *Nonlinear Time Series: Nonparametric and Parametric Methods (Springer Series in Statistics)*. Springer-Verlag New York.

Fenn, D. J., Porter, M. A., Williams, S., McDonald, M., Johnson, N. F., and Jones, N. S. (2011). Temporal evolution of financial-market correlations. *Physical Review E*, 84(2).

Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7:179–188.

Fodor, I. K. (2002). A survey of dimension reduction techniques. *Technical Report UCRL-ID-148494, Lawrence Livermore National Laboratory*.

Forni, M., Hallin, M., Lippi, M., and Reichlin, L. (2000). The generalized dynamic-factor model: Identification and estimation. *Review of Economics and Statistics*, 82(4):540–554.

Forni, M., Hallin, M., Lippi, M., and Reichlin, L. (2001). Coincident and leading indicators for the euro area. *Economic Journal*, 111(471):82–5.

Forni, M., Hallin, M., Lippi, M., and Reichlin, L. (2005). The generalized dynamic factor model. *Journal of the American Statistical Association*, 100(471):830–840.

Forni, M., Hallin, M., Lippi, M., and Zaffaroni, P. (2014). The generalized dynamic factor model. *Journal of Econometrics*, forthcoming.

Forni, M. and Lippi, M. (2001). The generalized dynamic factor model : representation theory. *Econometric Theory*, 17:1113–1141.

Forni, M. and Reichlin, L. (2001). Federal policies and local economies: Europe and the us. *European Economic Review*, 45(1):109–134.

Fort, J. (2006). Som's mathematics. *Neural Networks*, 19(6-7):812–816.

Frenzel, S. and Pompe, B. (2007). Partial mutual information for coupling analysis of multivariate time series. *Physical review letters*, 99(20):204101.

Friedman, J., Hastie, T., and Tibshirani, R. (2009). *The elements of statistical learning: Data mining, inference, and prediction, Second edition - 2nd edition*. Springer-Verlag New York, New York, NY, 2 edition.

Geweke, J. (1977). The dynamic factor analysis of economic time series. *Latent Variables in Socio-Economic Models*, Amsterdam: North-Holland.

Giannerini, S. (2012). The quest for nonlinearity in time series. *Time Series Analysis: Methods and Applications*, pages 43–63.

Giannerini, S., Maasoumi, E., and Dagum, E. B. (2015). Entropy testing for nonlinear serial dependence in time series. *Biometrika*, 102(3):661–675.

Giovannetti, B. C. (2011). Nonlinear forecasting using factor-augmented models. *Journal of Forecasting*, 32(1):32–40.

Granger, C. W. (1969). Investigating causal relations by econometric models and cross-spectral methods. *Econometrica: Journal of the Econometric Society*, pages 424–438.

Granger, C. W. J. (1963). Economic processes involving feedback. *Information and control*, 6(1):28–48.

Granger, C. W. J. and Andersen, A. P. (1978). *An introduction to bilinear time series models*. Vandenhoeck und Ruprecht, Göttingen.

Granger, C. W. J. and Pesaran, M. H. (2000). Economic and statistical measures of forecast accuracy. *Journal of Forecasting*, 19(7):537–560.

Granger, C. W. J. and Teräsvirta, T. (1993). *Modelling non-linear economic relationships*. Oxford University Press, Oxford, United Kingdom.

Hansen, B. (1999). Testing for linearity. *Journal of Economic Surveys*, 13(5):551–576.

Harvey, D., Leybourne, S., and Newbold, P. (1997). Testing the equality of prediction mean squared errors. *International Journal of Forecasting*, 13(2):281–291.

Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The elements of statistical learning data mining, inference, and prediction: With 200 full-color illustrations*. Springer-Verlag New York, New York, 4 edition.

Hawkes, A. G. (1971). Spectra of some self-exciting and mutually exciting point processes. *Biometrika*, 58(1):83–90.

Heston, S. L. (1993). A closed-form solution for options with stochastic volatility with applications to bond and currency options. *Review of financial studies*, 6(2):327–343.

Hinich, M. J. (1982). Testing for gaussianity and linearity of a stationary time series. *Journal of Time Series Analysis*, 3(3):169–176.

Hinton, G. and Salakhutdinov, R. (2006). Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507.

Hjellvik, V. and Tjostheim, D. (1995). Nonparametric tests of linearity for time series. *Biometrika*, 82(2):351–368.

Hoerl, A. E. and Kennard, R. W. (1970a). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55.

Hoerl, A. E. and Kennard, R. W. (1970b). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67.

Horn, R. A. and Johnson, C. R. (2012). *Matrix analysis*. Cambridge university press.

Hornik, K. (1991). Approximation capabilities of multilayer feedforward networks. *Neural Networks*, 4(2):251–257.

Hornik, K., Stinchcombe, M., and White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366.

Hotelling, H. (1936). Relations between two sets of variates. *Biometrika*, 28(3/4):321.

Hsieh, D. A. (1991). Chaos and nonlinear dynamics: Application to financial markets. *The Journal of Finance*, 46(5):1839–1877.

Hsieh, W. W. (2004). Nonlinear multivariate and time series analysis by neural network methods. *Reviews of Geophysics*, 42(1).

Huber, P. J. (2011). *Data analysis: What can be learned from the past 50 years*. John Wiley & Sons, United States.

Hutter, F., Hoos, H. H., and Leyton-Brown, K. (2011). Sequential model-based optimization for general algorithm configuration. *Proceedings of the 5th international conference on Learning and Intelligent Optimization*, pages 507–523.

Jäkel, F., Schölkopf, B., and Wichmann, F. A. (2007). A tutorial on kernel methods for categorization. *Journal of Mathematical Psychology*, 51(6):343–358.

Jeffrey, A. (2005). *Complex analysis and applications*, volume 10. CRC Press.

Jiao, J., Permuter, H. H., Zhao, L., Kim, Y.-H., and Weissman, T. (2013). Universal estimation of directed information. *Information Theory, IEEE Transactions on*, 59(10):6220–6242.

Jolliffe, I. T. (2016). *Principal component analysis*. Springer-Verlag, New York, 2nd edition.

Keenan, D. M. (1985). A tukey nonadditivity-type test for time series nonlinearity. *Biometrika*, 72(1):39.

Keylock, C. J. (2006). Constrained surrogate time series with preservation of the mean and variance str. *Physical Review E*, 73(3).

Keylock, C. J. (2007). A wavelet-based method for surrogate data generation. *Physica D: Nonlinear Phenomena*, 225(2):219–228.

Keylock, C. J. (2008). Improved preservation of autocorrelative structure in surrogate data using an initial wavelet step. *Nonlinear Processes in Geophysics*, 15(3):435–444.

Keylock, C. J. (2010). Characterizing the structure of nonlinear systems using gradual wavelet reconstr. *Nonlinear Processes in Geophysics*, 17:615–632.

Kim, S., Putrino, D., Ghosh, S., and Brown, E. N. (2011). A granger causality measure for point process models of ensemble neural spiking activity. *PLoS computational biology*, 7(3):e1001110.

Kohonen, T. (2001). *Self-organizing maps*. Springer-Verlag Berlin Heidelberg.

Koller, D. and Friedman, N. (2009). *Probabilistic graphical models: principles and techniques*. MIT press.

Koop, G., Pesaran, M. H., and Potter, S. M. (1996). Impulse response analysis in nonlinear multivariate models. *Journal of econometrics*, 74(1):119–147.

Kramer, G. (1998). *Directed information for channels with feedback*. PhD thesis, University of Manitoba, Canada.

Kramer, M. A. (1991). Nonlinear principal component analysis using autoassociative neural networks. *AIChE Journal*, 37(2):233–243.

Kraskov, A., Stögbauer, H., and Grassberger, P. (2004). Estimating mutual information. *Physical review E*, 69(6):066138.

Kruger, U., Zhang, J., and Xie, L. (2008). Developments and applications of nonlinear principal component analysis-a review. In *Principal manifolds for data visualization and dimension reduction*, pages 1–43. Springer.

Kuan, C.-M. and White, H. (1994). Reply to comments on "artificial neural networks: An econometric perspective". *Econometric Reviews*, 13(1):139–143.

Kuan, G. (2008). Lecture on time series diagnostic tests. *Institute of Economics Academia.*

Kugiumtzis, D. (2002). Statically transformed autoregressive process and surrogate data test for nonlinearity. *Physical Review E*, 66(2).

Kugiumtzis, D. and Bora-Senta, E. (2014). Simulation of multivariate nongaussian autoregressive time series with given autocovariance and marginals. *Simulation Modelling Practice and Theory*, 44:42–53.

Kulis, B. (2012). Metric learning: A survey. *Foundations and Trends in Machine Learning*, 5(4):287–364.

Lam, C. and Yao, Q. (2012). Factor modeling for high-dimensional time series: Inference for the number of factors. *The Annals of Statistics*, 40(2):694–726.

Larsen, J., Hansen, L., Svarer, C., and Ohlsson, M. (1996). Design and regularization of neural networks: The optimal use of a validation set. *Proceedings of the IEEE Workshop on Neural Networks for Signal Processing*, VI(Piscataway, New Jersey):62–71.

Larsen, J., Svarer, C., Andersen, L. N., and Hansen, L. K. (2012). Adaptive regularization in neural network modeling. *Neural Networks: Tricks of the Trade*, pages 111–130.

Larson, S. C. (1931). The shrinkage of the coefficient of multiple correlation. *Journal of Educational Psychology*, 22(1):45–55.

Lee, S.-I., Lee, H., Abbeel, P., and Ng, A. Y. (2006). Efficient l1 regularized logistic regression. Proceedings of the Twenty-First National Conference on Artificial Intelligence (AAAI-06).

Levenberg, K. (1944). A method for the solution of certain non-linear problems in least squares. *Quarterly of Applied Mathematics*, 2(2):164–168.

Lewis, E. and Mohler, G. (2011). A nonparametric em algorithm for multiscale hawkes processes. *Preprint.*

Li, W. K. (2003). *Diagnostic checks in time series*. Chapman & Hall, London.

Linderman, S. W. and Adams, R. P. (2014). Discovering latent network structure in point process data. *preprint arXiv:1402.0914*.

Liniger, T. J. (2009). *Multivariate hawkes processes*. PhD thesis, Diss., Eidgenössische Technische Hochschule ETH Zürich, Nr. 18403, 2009.

Lintner, J. (1965). The valuation of risk assets and the selection of risky investments in stock portfolios and capital budgets. *The Review of Economics and Statistics*, 47(1):13.

Liu, Y. and Jin, R. (2006). Distance metric learning: A comprehensive survey. *Technical report, Department of Computer Science and Engineering, Michigan State University,*.

Ljung, G. M. and Box, G. E. P. (1978). On a measure of lack of fit in time series models. *Biometrika*, 65(2):297–303.

Loftsgaarden, D. O., Quesenberry, C. P., et al. (1965). A nonparametric estimate of a multivariate density function. *The Annals of Mathematical Statistics*, 36(3):1049–1051.

Lomax, R. G. and Hahs-Vaughn, D. L. (2013). *Statistical concepts: A second course*. Routledge.

Ludvigson, S. C. and Ng, S. (2009). Macro factors in bond risk premia. *Review of Financial Studies*, 22(12):5027–5067.

Luukkonen, R., Saikkonen, P., and Teräsvirta, T. (1988). Testing linearity in univariate time series models. *Scandinavian Journal of Statistics*, 15(3):161–175.

MacKay, D. J. C. (1992). Bayesian interpolation. *Maximum Entropy and Bayesian Methods*, pages 39–66.

Maclaurin, D., Duvenaud, D., and Adams, R. P. (2015). Gradient-based hyperparameter optimization through reversible learning. *Arxiv preprint*, arXiv:1502.03492.

Marček, D. (1998). Stock price prediction using autoregressive models and signal processing procedures. In *Proceedings of the 16th Conference MME*, volume 98, pages 114–121.

Marcellino, M., Stock, J. H., and Watson, M. W. (2003). Macroeconomic forecasting in the euro area: Country specific versus area-wide information. *European Economic Review*, 47(1):1–18.

Marcellino, M. G., Stock, J. H., and Watson, M. W. (2005). A comparison of direct and iterated multistep ar methods for forecasting macroeconomic time series. *SSRN Electronic Journal*.

Marko, H. (1973). The bidirectional communication theory–a generalization of information theory. *Communications, IEEE Transactions on*, 21(12):1345–1351.

Marquardt, D. W. (1963). An algorithm for least-squares estimation of nonlinear parameters. *Journal of the Society for Industrial and Applied Mathematics*, 11(2):431–441.

Massey, J. (1990). Causality, feedback and directed information. In *Proc. Int. Symp. Inf. Theory Applic.(ISITA-90)*, pages 303–305. Citeseer.

Materassi, D. and Salapaka, M. V. (2012). On the problem of reconstructing an unknown topology via locality properties of the wiener filter. *Automatic Control, IEEE Transactions on*, 57(7):1765–1777.

Maxwell, A. E. and Lawley, D. (1971). *Factor analysis as a statistical method*. Butterworth & Co Publishers, London, 2nd edition.

McLeod, A. I. and Li, W. K. (1983). Diagnostic checking arma time series models using squared-residual autocorrelations. *Journal of Time Series Analysis*, 4(4):269–273.

Mendel, J. (1991). Tutorial on higher-order statistics (spectra) in signal processing and system theory: Theoretical results and some applications. *Proceedings of the IEEE*, 79(3):278–305.

Mohler, G. O., Short, M. B., Brantingham, P. J., Schoenberg, F. P., and Tita, G. E. (2011). Self-exciting point process modeling of crime. *Journal of the American Statistical Association*, 106(493).

Mol, C. D., Giannone, D., and Reichlin, L. (2008). Forecasting using a large number of predictors: Is bayesian shrinkage a valid alternative to principal components? *Journal of Econometrics*, 146(2):318–328.

Monahan, A. H. (2001). Nonlinear principal component analysis: Tropical indo–pacific sea surface temperature and sea level pressure. *Journal of Climate*, 14(2):219–233.

Moody, J. E. (1991). Note on generalization, regularization and architecture selection in nonlinear learning systems. *Neural Networks for Signal Processing Proceedings of the 1991 IEEE Workshop*.

Morgan, J. N. and Sonquist, J. A. (1963). Problems in the analysis of survey data, and a proposal. *Journal of the American Statistical Association*, 58(302):415.

Muni Toke, I. and Pomponio, F. (2011). Modelling trades-through in a limited order book using hawkes processes. *Economics discussion paper*, (2011-32).

Murphy, K. P. (2002). *Dynamic bayesian networks: representation, inference and learning*. PhD thesis, University of California, Berkeley.

Musicus, B. R. (1988). *Levinson and fast Choleski algorithms for Toeplitz and almost Toeplitz matrices*. Citeseer.

Ng, A. Y. (2004). Feature selection, l1 vs. l2 regularization, and rotational invariance. In ICML.

Nikias, C. L. and Petropulu, A. P. (1993). *Higher order spectra analysis: A non-linear signal processing framework*. PTR Prentice Hall, New York, NY, United States.

Ogata, Y. (1999). Seismicity analysis through point-process modeling: A review. *Pure and Applied Geophysics*, 155(2-4):471–507.

Oja, E. (1982). Simplified neuron model as a principal component analyzer. *Journal of Mathematical Biology*, 15(3):267–273.

Ozaki, T. (1979). Maximum likelihood estimation of hawkes' self-exciting point processes. *Annals of the Institute of Statistical Mathematics*, 31(1):145–155.

Ozaki, T. (1982). The statistical analysis of perturbed limit cycle processes using nonlinear time series models. *Journal of Time Series Analysis*, 3(1):29–41.

Ozaki, T. (1985). Non-linear time series models and dynamical systems. *Time Series in the Time Domain*, pages 25–83.

Pearl, J. (1988). *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann.

Pearl, J. (2009). *Causality*. Cambridge university press.

Pearson, K. (1901). Liii. on lines and planes of closest fit to systems of points in space. *Philosophical Magazine Series 6*, 2(11):559–572.

Pesaran, H. H. and Shin, Y. (1998). Generalized impulse response analysis in linear multivariate models. *Economics letters*, 58(1):17–29.

Pesaran, M. H. and Timmermann, A. (1995). Predictability of stock returns: Robustness and economic significance. *The Journal of Finance*, 50(4):1201.

Pesaran, M. H. and Timmermann, A. (2000). A recursive modelling approach to predicting uk stock returns. *The Economic Journal*, 110(460):159–191.

Petropulu, A. (1999). Higher-order spectral analysis. *Electrical Engineering Handbook*.

Piersol, A. G. (1993). Optimum resolution bandwidth for spectral analysis of stationary random vibration data. *Shock and Vibration*, 1(1):33–43.

Pinto, J. C. L., Chahed, T., and Altman, E. (2015). Trend detection in social networks using hawkes processes. In *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015*, pages 1441–1448. ACM.

Quinn, C., Kiyavash, N., and Coleman, T. P. (2015). Directed information graphs. *Transactions on Information Theory*, 61(12):6887–6909.

Quinn, C. J., Cole, T. P., and Kiyavash, N. (2011a). A generalized prediction framework for granger causality. In *Computer Communications Workshops (INFOCOM WKSHPS), 2011 IEEE Conference on*, pages 906–911. IEEE.

Quinn, C. J., Coleman, T. P., Kiyavash, N., and Hatsopoulos, N. G. (2011b). Estimating the directed information to infer causal relationships in ensemble neural spike train recordings. *Journal of computational neuroscience*, 30(1):17–44.

Quinn, C. J., Kiyavash, N., and Coleman, T. P. (2011c). Equivalence between minimal generative model graphs and directed information graphs. In *Information Theory Proceedings (ISIT), 2011 IEEE International Symposium on*, pages 293–297. IEEE.

Ramsey, J. B. (1969). Tests for specification errors in classical linear least-squares regression analysis. *Journal of the Royal Statistical Society. Series B (Methodological)*, 31(2):350–371.

Rao, R. C. (1948). The utilization of multiple measurements in problems of biological classification. *Journal of the Royal Statistical Society. Series B (Methodological)*, 10(2):159–203.

Rao, S. T. and Gabr, M. M. (1984). *An introduction to bispectral analysis and bilinear time series models*. Springer-Verlag New York, New York.

Rao, T. S. and Gabr, M. M. (1980). A test for linearity of stationary time series. *Journal of Time Series Analysis*, 1(2):145–158.

Raviv, E. and Dijk, D. J. C. V. (2014). Forecasting with many predictors: Allowing for non-linearity. *Available at SSRN: https://ssrn.com/abstract=2565288*.

Reynaud-Bouret, P., Schbath, S., et al. (2010). Adaptive estimation for hawkes processes; application to genome analysis. *The Annals of Statistics*, 38(5):2781–2822.

Rissanen, J. and Wax, M. (1987). Measures of mutual and causal dependence between two time series (corresp.). *Information Theory, IEEE Transactions on*, 33(4):598–601.

Rivals, I. and Personnaz, L. (1999). On cross validation for model selection. *Neural Computation*, 11(4):863–870.

Ross, S. A. (1976). The arbitrage theory of capital asset pricing. *Journal of Economic Theory*, 13(3):341–360.

Roweis, S. T. and Saul, L. K. (2000). Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, 323(6088):533–536.

Rusticelli, E., Ashley, R. A., Dagum, E. B., and Patterson, D. M. (2008). A new bispectral test for nonlinear serial dependence. *Econometric Reviews*, 28(1-3):279–293.

Saikkonen, P. and Luukkonen, R. (1988). Lagrange multiplier tests for testing nonlinearities in time series models. *Scandinavian Journal of Statistics*, 15(1):55–68.

Sargent, T. J. and Sims, C. A. (1977). Business cycle modeling without pretending to have too much a-priori economic theory. *New Methods in Business Cycle Research*, Federal Reserve Bank of Minneapolis.

Schölkopf, B., Smola, A., and Müller, K.-R. (1997). Kernel principal component analysis. *Artificial Neural Networks — ICANN'97*, 1327(of the series lecture notes in computer science):583–588.

Schölkopf, B., Smola, A., and Müller, K.-R. (1998). Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5):1299–1319.

Scholz, M., Fraunholz, M., and Selbig, J. (2008). Nonlinear principal component analysis: Neural network models and applications. In *Principal Manifolds for Data Visualization and Dimension Reduction*, pages 44–67. Springer Science, Berlin Heidelberg.

Schreiber, T. (2000). Measuring information transfer. *Physical review letters*, 85(2):461.

Schreiber, T. and Schmitz, A. (1996). Improved surrogate data for nonlinearity tests. *Physical Review Letters*, 77(4):635–638.

Schreiber, T. and Schmitz, A. (1997). Discrimination power of measures for nonlinearity in a time series. *Physical Review E*, 55(5):5443–5447.

Schreiber, T. and Schmitz, A. (2000). Surrogate time series. *Physica D: Nonlinear Phenomena*, 142(s 3–4):346–382.

Schwarz, G. et al. (1978). Estimating the dimension of a model. *The annals of statistics*, 6(2):461–464.

Schweitzer, F., Fagiolo, G., Sornette, D., Vega-Redondo, F., Vespignani, A., and White, D. R. (2009). Economic networks: The new challenges. *science*, 325(5939):422.

Sharpe, W. F. (1964). Capital asset prices: A theory of market equilibrium under conditions of risk. *The Journal of Finance*, 19(3):425.

Snoek, J., Larochelle, H., and Adams, R. P. (2012). Practical bayesian optimization of machine learning algorithms. *Advances in Neural Information Processing Systems*, 25:2960–2968.

Spearman, C. (1904). General intelligence objectively determined and measured. *The American Journal of Psychology*, 15(2):201–292.

Sricharan, K., Raich, R., and Hero, A. O. (2011). k-nearest neighbor estimation of entropies with confidence. In *Information Theory Proceedings (ISIT), 2011 IEEE International Symposium on*, pages 1205–1209. IEEE.

Stock, J. H. and Watson, M. W. (1999). Forecasting inflation. *Journal of Monetary Economics*, 44(2):293–335.

Stock, J. H. and Watson, M. W. (2002a). Forecasting using principal components from a large number of predictors. *Journal of the American Statistical Association*, 97(460):1167–1179.

Stock, J. H. and Watson, M. W. (2002b). Macroeconomic forecasting using diffusion indexes. *Journal of Business & Economic Statistics*, 20(2):147–162.

Stock, J. H. and Watson, M. W. (2006). Chapter 10 forecasting with many predictors. *Handbook of Economic Forecasting*, pages 515–554.

Tenenbaum, J., vin de Silva, and john Langford (2000). A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323.

Teräsvirta, T. (1994). Testing linearity and modelling nonlinear time series. *Kybernetika*, 30(3):319–330.

Teräsvirta, T., Tjøstheim, D., and Granger, C. W. J. (2010). *Modelling nonlinear economic time series*. Oxford University Press, Oxford.

Teräsvirta, T., van Dijk, D., and Medeiros, M. C. (2005). Linear models, smooth transition autoregressions, and neural networks for forecasting macroeconomic time series: A re-examination. *International Journal of Forecasting*, 21(4):755–774.

Teräsvirta, T., Tjøstheim, D., and Granger, C. W. J. (2010). Modelling nonlinear economic time series.

Theiler, J., Eubank, S., Longtin, A., Galdrikian, B., and Farmer, D. J. (1992). Testing for nonlinearity in time series: The method of surrogate data. *Physica D: Nonlinear Phenomena*, 58(1-4):77–94.

Theiler, J. and Prichard, D. (1996). Constrained-realization monte-carlo method for hypothesis testing. *Physica D: Nonlinear Phenomena*, 94(4):221–235.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B*, 58(1):267–288.

Tong, H. (1978). On a threshold model. In Chen, C., editor, *Pattern recognition and signal processing*, pages 575–586. Sijthoff & Noordhoff, Netherlands.

Tong, H. (1983). *Threshold models in non-linear time series analysis*. Springer-Verlag New York, New York, Berlin, Heidelberg [usw.].

Tong, H. (1990). *Non-linear time series: A dynamical system approach*. Clarendon Press, Oxford.

Tong, H. and Lim, K. S. (1980). Threshold autoregression, limit cycles and cyclical data. *Journal of the Royal Statistical Society. Series BMethodological)*, 42(3):245–292.

Torgerson, W. S. (1952). Multidimensional scaling: I. theory and method. *Psychometrika*, 17(4):401–419.

Tsay, R. S. (1986). Nonlinearity tests for time series. *Biometrika*, 73(2):461.

Tsay, R. S. (2005). *Analysis of Financial Time Series*. Wiley, 3rd edition.

van Dijk, D. and Franses, P. H. (1999). Modeling multiple regimes in the business cycle |macroeconomic dynamics |cambridge core. *Macroeconomic Dynamics*, 3(3):311–340.

van Dijk, D., Teräsvirta, T., and Franses, P. H. (2007). Smooth transition autoregressive models - a survey of recent developments. *Econometric Reviews*.

Vapnik, V. (1982). *Estimation of Dependences based on empirical data: Springer series in statistics*. Springer-Verlag New York, New York, Heidelberg, Berlin.

Vapnik, V. N. (1995). *The nature of statistical learning theory*. Springer-Verlag New York, New York, 2 edition.

Vapnik, V. N. and Chervonenkis, A. Y. (1964). A note on one class of p erceptrons. *Automation and Remote Control*, 25.

Vapnik, V. N. and Chervonenkis, A. Y. (1974). Theory of pattern recognition: Statistical problems of learning in russian. *Nauka Moscow*, English translation: Academic, New York(2).

Varian, H. R. (2014). Big data: New tricks for econometrics †. *Journal of Economic Perspectives*, 28(2):3–28.

Venema, V., Ament, F., and Simmer, C. (2006). A stochastic iterative amplitude adjusted fourier transform algorithm with improved accuracy. *Nonlinear Processes in Geophysics*, 13(3):321–328.

Wahba, G. (1990). *Spline models for observational data*. Society for Industrial & Applied Mathematics,U.S., Philadelphia, PA, 4 edition.

Weinberger, K. Q. and Saul, L. K. (2006). An introduction to nonlinear dimensionality reduction by maximum variance unfolding. *National Conference on Artificial Intelligence (AAAI), Nectar paper, Boston MA*.

Weissman, T., Kim, Y.-H., and Permuter, H. H. (2013). Directed information, causal estimation, and communication in continuous time. *Information Theory, IEEE Transactions on*, 59(3):1271–1287.

Werbos, P. J. (1988). Generalization of backpropagation with application to a recurrent gas market model. *Neural Networks*, 1(4):339–356.

White, H. (1990). Connectionist nonparametric regression: Multilayer feedforward networks can learn arbitrary mappings. *Neural Networks*, 3(5):535–549.

Wiener, N. (1956). The theory of prediction. *Modern mathematics for engineers*, 1:125–139.

Yang, S.-H. and Zha, H. (2013). Mixture of mutually exciting processes for viral diffusion. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pages 1–9.

Zhou, K., Zha, H., and Song, L. (2013). Learning triggering kernels for multidimensional hawkes processes. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pages 1301–1309.

Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320.