HELSINKI UNIVERSITY OF TECHNOLOGY
Department of Automation and Systems Technology

Jarkko Tikka

Learning linear dependency trees from multivariate data

Master's thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Technology

Espoo, 21st May 2004

Supervisor: Professor Jaakko Hollmén
Instructor: Professor Jaakko Hollmén

Teknillinen korkeakoulu	**Diplomityön tiivistelmä**
Automaatio- ja systeemitekniikan osasto	
Tekijä Jarkko Tikka	**Päiväys** 21.5.2004
	Sivumäärä 67
Työn nimi Lineaaristen riippuvuuspuiden oppiminen moniulotteisesta datasta	
Professuuri Informaatiotekniikka	**Koodi** T-122
Valvoja: Ma. Prof. Jaakko Hollmén	
Ohjaaja: Ma. Prof. Jaakko Hollmén	

Tässä työssä tutkitaan lineaarisia riippuvuuksia moniulotteisessa datassa. Muuttujista muodostetaan usean selittäjän lineaarisia regressiomalleja. Myös regressiomallien välisiä riippuvuuksia tutkitaan. Lopullinen riippuvuusrakenne on puu tai metsä. Riippuvuusrakenteesta voidaan saada täydentävää ja odottamatonta informaatiota datan tuottaneesta prosessista.

Aluksi esitellään kaksi datan esikäsittelymenetelmää. Ensin näytetään kuinka aikasarjoista voidaan valita samanlaiset tilat. Muuttujien välillä oletetaan olevan lineaarinen riippuvuus valitun tilan aikana. Mittaukset sisältävät tyypillisesti kohinaa. Kohina voi häiritä mallien estimointia tai peittää aikasarjan mielenkiintoiset ominaisuudet. Tämän vuoksi mittauksiin sovelletaan diskreettiin aallokemuunnokseen perustuvaa kohinanpoistomentelmää.

Seuraavaksi estimoidaan niin monta regressiomallia kuin datassa on muuttujia. Jokainen muuttuja on vuorollaan selitettävä muuttuja ja loput muuttujat ovat mahdollisia selittäjiä. Merkitsevimmät selittäjät etsitään käyttämällä Least Angle Regression mallinvalinta-algoritmia ja bootstrap-menetelmää. Selittäville muuttujille lasketaan suhteelliset painot käyttämällä bootstrap-menetelmän toistoja parametreista. Suhteellinen paino mittaa uskomusta, että vastaava selittäjä kuuluu estimoituun malliin.

Uskomusgraafi muodostetaan suhteellisista painoista. Graafiin käytetään moralisointioperaatiota ja kynnysarvoa. Lopullinen riippuvuusrakenne konstruoidaan modifioidusta uskomusgraafista.

Esiteltyä menetelmää testataan kolmella erilaisella datajoukolla ja kaikista joukoista saadut tulokset ovat vakuuttavia. Menetelmää voidaan periaatteessa soveltaa hyvin erilaisiin datajoukkoihin tutkittaessa lineaaristen riippuvuuksien olemassaoloa.

Avainsanat
kohinanpoisto, lineaarinen harvaregressio, bootstrap-menetelmä, suhteellinen paino, klikki, riippuvuuspuu, riippuvuusmetsä

Helsinki University of Technology	**Abstract of Master's thesis**
Department of Automation and Systems Technology	
Author Jarkko Tikka	**Date** 21.5.2004
	Number of pages 67
Title Learning linear dependency trees from multivariate data	
Professorship Computer and information science	**Code** T-122
Supervisor: Prof. (pro tem) Jaakko Hollmén	
Instructor: Prof. (pro tem) Jaakko Hollmén	

In this work, linear dependencies in multivariate data are studied. Multiple linear regression models are constructed from the variables. Also, the dependencies between the different regression models are studied. The final dependency structure is a tree or a forest. Additional and unsuspected information from the underlying process can be obtained from the dependency structure.

In the beginning, two preprocessing techniques of the data are presented. First, it is shown how similar states can be selected from time series. It is assumed that there exists a linear dependency between the variables during the selected state. Secondly, measurements include typically noise. The noise can disturb the estimation of the models or cover interesting features of time series. Thus, a noise reduction technique based on discrete wavelet transform is applied to the measurements.

Next, as many regression models as there are variables in the data set are estimated. Each variable is in turn the dependent variable and the rest of the variables are possible independent variables. The most significant independent variables are sought using the Least Angle Regression model selection algorithm and bootstrap. Relative weights are calculated for the independent variables using the bootstrap replications. The relative weight measures the belief that the corresponding independent variable belongs to the estimated model.

A belief graph is constructed using the relative weights. A threshold value and moralizing operation are applied to the graph. The final dependency structure is constructed from the modified belief graph.

The proposed method is tested with three data sets and the results from all sets are convincing. Basically, the method can be applied to very different data sets to explore the existence of linear dependencies.

Keywords
noise reduction, linear sparse regression, bootstrap, relative weight, clique, dependency tree, dependency forest

Preface

This work has been carried out at the Laboratory of Computer and Information Science of Helsinki University of Technology. I have been honored to be the member of the Intelligent Data Engineering research group. I am grateful to Professor Olli Simula for making the necessary financial arrangements.

First of all, I want to thank my supervisor and instructor, Professor Jaakko Hollmén, for encouragement, guidance and patience. He had always time for my questions and without his invaluable advice this work would not have been possible.

I also want to thank my former instructor Dr. Esa Alhoniemi for guiding me when I started to work in the Intelligent Data Engineering research group. He initiated me into habits and mysteries of scientific research. I also want to thank Dr. Esa Alhoniemi for developing the idea of selecting similar time windows from time series.

I also want to express my gratitude and appreciation to M.Sc Timo Similä, M.Sc Mika Sulkava and M.Sc Pasi Lehtimäki. Their comments have been valuable and insightful. They have also made the daily life and the atmosphere enjoyable.

After all, I want to thank my parents and sister for a supportive and positive attitude towards my work and studying. At last, I want to enounce my enormous compliments to all my friends, you know who you are.

Espoo, 21st May 2004

Jarkko Tikka

Contents

Abbreviations and Notations

DWT	Discrete Wavelet Transform
IDWT	Inverse Discrete Wavelet Transform
RSS	Residual Sum of Squares
OLS	Ordinary Least Squares
Lasso	Least absolute shrinkage and selection operator
LARS	Least Angle Regression
MDL	Minimum Description Length
CPU	Central Processing Unit
D	data set
N	number of observations
k	number of independent variables
$\mathbf{y}$	dependent variable
$\hat{\mathbf{y}}$	estimate of the dependent variable
$\mathbf{x}$	independent variable
$\boldsymbol{\epsilon}$	random error term
$\boldsymbol{\beta}$	estimator of regression coefficients
$\mathbf{b}$	estimate of regression coefficients
R^2	coefficient of determination
α, τ	constants
B	number of bootstrap replications
$\mathbf{w}$	relative weights of regression coefficients
λ	threshold value
G_b	belief graph
G_u	undirected graph
G_m	moral graph
c_n^i	ith n-clique
V_i	ith node of graph

Chapter 1

Introduction

There are lots of data available nowadays. Data are collected from different sources, for example from industrial processes, economy, mobile phone communication and environment. The data are typically multidimensional i.e. there are many variables in the data. The deeper understanding of the underlying process can be achieved by exploring or analyzing the data. Economical and ecological benefits are great motivation for the data analysis. The following definition of the data mining is given in the book [17, p. 1].

> Data mining is the analysis of (often large) observational data sets to find unsuspected relationships and to summarize the data in novel ways that are both understandable and useful to the data owner.

That definition describes pretty well the approach of this study. In this study, dependencies between the variables in the data set are analyzed. The purpose is to construct multiple linear regression models. In the end, a linear dependency tree or forest is constructed from the variables. One variable can belong to several regression models in the final structure. From the dependency structure can be clearly seen how a change in a value of one variable induces changes in values of other variables. This can be useful information in many cases. Especially, if some vulnerable variable is dependent on some other variables, then the state of the vulnerable variable can be controlled by other variables.

The multiple linear regression models have a couple of advantages. The dependencies in linear models are easy to interpret. In addition, processes might

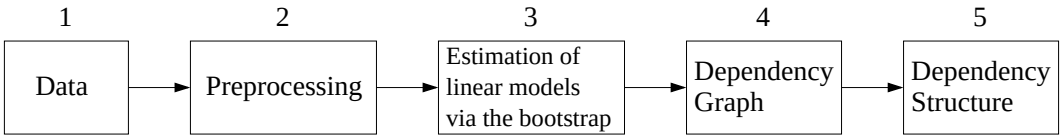

Figure 1.1: The flow chart of the proposed method.

be inherently linear or over short ranges many processes can be approximated by a linear model.

A flow chart of the methodology proposed in this study is presented in Figure (1.1). The method consists of five phases. The first phase is that there should be available some data which can be analyzed or measurements have to be made from some process.

The second phase deals with preprocessing of data. Some operations have to be usually performed on measurements before they can be analyzed mathematically. Some measurements might be missing or measurements might be noisy. In this study, preprocessing includes a noise reduction and a selection of interesting parts of time series.

In third phase, as many multiple linear regression models as there are variables in the data are constructed. Each variable is a dependent variable in turn and the rest of the variables are possible independent variables. The most significant independent variables for each model are selected by sparse regression algorithm. The bootstrap is also applied to the selection of the independent variables and the estimation of the regression coefficients in each case. The relative weights of the regression coefficients are computed from the bootstrap replications. The relative weight of the regression coefficient is a measure of belief that the corresponding independent variable belongs to the estimated linear model.

In the fourth phase a belief graph is constructed from the relative weights of the regression coefficients. The belief graph represents the strength of the dependencies between the variables. In the belief graph there are as many nodes as there are variables in the data. The relative weights define arcs of the belief graph. The weights of the arcs are obtained from the relative weights. The weights represent the strength of the dependency between two nodes or variables. The weakest dependencies are ignored using a proper threshold value. The moralizing operation is also applied to the belief graph and the result is a moral graph or a final dependency graph.

In the fifth phase, a dependency structure of the variables is constructed from the dependency graph. A set of variables, which form a multiple linear regression model, belongs to a same maximal clique in the dependency graph. One of the variables is selected to be the dependent variable. One variable can belong to several multiple linear regression models, but it can only be the dependent variable in one model. Two linear models are dependent on each other if a same variable belongs to the both models. Two models can only have one variable in common. The dependencies cannot form circles in a final dependency structure i.e. the variable cannot be dependent on itself through the other variables. Thus, the final dependency structure is a tree or a forest.

As far as I know, the proposed method is a novel way to model a structure of the linear dependencies in a multivariate data. In the article [8] dependency trees are also used. There is a method shown which approximates optimally n-dimensional probability distributions. Each variable can only be dependent on at most one variable in that model.

In the article [37], an algorithm which models dependencies between inputs and outputs is presented. The algorithm can be applied for general class of objects of inputs and outputs. The objects can for example be vectors or images. In the algorithm, kernel functions are used to measure similarities between the inputs as well as the outputs.

Belief networks is discussed in the article [22]. In that article is shown how the belief network can be constructed. In the end, the network induces a probability distribution over its variables. The conditional probabilities of the variables are presented by a belief network. The belief networks are directed and acyclic. In the article [20] dependency networks, which might be cyclic, are presented. In both belief and dependency networks the variables are conditioned upon its parents.

In the article [26] independent variable group analysis (IVGA) is proposed. In that approach variables are clustered. Variables in one cluster are dependent on each other but they are independent on variables which belong to other clusters. In IVGA, the dependencies between the groups or clusters are ignored and the dependencies in each group can be modeled in different ways.

Structural equation modeling (SEM) [30] is another technique to investigate relationships between the variables. SEM provides a methodology to test a plausibility of hypothesized models. The predefined dependencies between the variables are investigated using the SEM, when the dependencies are learned from the data in this study proposed method. Structural Equation models

can consist of both observed and latent variables. The latent variables can be extracted from the observed ones using the factor analysis. The observed or measured variables are only modeled in this study.

The rest of the thesis is organized as follows. In Chapter 2 is presented how the similar states of time series can be selected. Also, the noise reduction technique based on the discrete wavelet transform is shown. The multiple linear regression model and sparse regression algorithms are introduced in the beginning of Chapter 3, followed by an application of the bootstrap in multiple linear regression and the computation of the relative weights of the regression coefficients. The construction of the linear dependency tree or forest is proposed thereafter. The proposed method is applied to three different data sets. One of the data sets is generated synthetically and the other two consist of real measurements. The results of the experiments are shown in Chapter 4. Conclusion and final remarks are in Chapter 5.

Chapter 2

Preprocessing of data

2.1 Selection of the windows

Data sets contain usually many variables. A number of measurements from each variable can be very large, even many thousands. The variables describe different features of the underlying process. In the time series data each measurement has a time label and the information of the data progresses in time. The time series consist of a deterministic and a stochastic part. The deterministic part of the time series can be modeled mathematically. The stochastic part is usually random noise or white noise.

The process might be in different states during the measurements. For example, from a paper machine a measurement might be the production speed. The production speed can vary depending on which kind of paper is under production. The state of the process can be assumed to be same when a certain paper is produced.

The similar states of the process can be sought using for example the production speed. The production speed is called a reference signal in that case. The reference signal can be any of the signals in the data set depending on which feature is wanted to explore.

A query window is selected from the reference signal. The query window of the reference signal should include information or the measurements of the feature i.e. the interesting state of the time series, which is under exploration. The similar states of the reference signal can be located mathematically in many ways. In this study, the sum of squares of differences between the query window and a candidate window is minimized. The candidate window is a part

of the reference signal which is as long as the query window. The candidate windows are not allowed to overlap with each other or with the query window. The candidate windows which have the smallest sum of squares of differences between the query window are chosen.

The sum of squares of differences between the query window and the candidate window can be calculated as follows

$$E_c = \sum_{i=1}^{M} (y_{q,i} - y_{c,i})^2. \tag{2.1}$$

y_q is the query window and y_c is the candidate window in the Equation (2.1). M is a number of the measurements which are included in the query window or the length of the query window.

The candidate windows which minimize the Equation (2.1) are chosen. The number of the chosen candidate windows can be decided, for example, setting a threshold value to the sum of squares of differences between the candidate window and the query window or how many data points are at least needed in further calculations. The data in the chosen candidate windows and in the query window from the reference signal are chosen and rest of the measurements are excluded from further calculations. The parts, which has the same time label than chosen candidate windows and query window, are also selected from the rest of the signals. A diagram of the selection of the windows is presented in Figure (A.2) in Appendix A.

New time-series are got when the selected windows of the original signals are put one after another. The original measurements include many times a noise. The level of the noise might be disturbing high. If this is the case, noise reduction techniques can be applied to the selected windows.

2.2 Discrete wavelet transform

Signals or time series can be analyzed in a time or in a frequency domain. The time domain representation is not always the best way to illustrate features of the signal. Some features might not be distinguishable from each other in the time domain. Frequencies of the signal are illustrated in the frequency domain representation. The time domain information is lost in the frequency domain representation. In the frequency representation is assumed that all the frequencies are present in the signal all the time, which might not always be true.

The discrete wavelet transform (DWT) gives a time-frequency representation of the signal. The DWT divides the frequency band of the signal to subbands. The different features of the signal are shown in the different frequency bands. Noise is present mainly in subbands, which consist of information from higher frequencies. Thus, the DWT can be applied for example to a noise reduction. The DWT is presented briefly in the Sections 2.2.1-2.2.2. An application of the DWT to the noise reduction is presented in the Section 2.2.3. The more detailed presentation of the Sections 2.2.1-2.2.3 can be found from the book [23].

2.2.1 DWT in general

Discrete wavelet transform consists of a direct transform and its corresponding inverse transform. An input for the direct transform is a sequence or a signal $\mathbf{s_j}$ and outputs are signals $\mathbf{s_{j+1}}$ and $\mathbf{d_{j+1}}$ and vice versa for the inverse transform. The symbol $\mathbf{T_a}$ represents the analysis part that is the DWT and the symbol $\mathbf{T_s}$ represents synthesis part that is the corresponding inverse discrete wavelet transform (IDWT). These building blocks are drawn in Figure (2.1). Contents of the blocks are explained later.

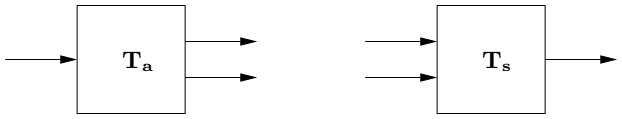

Figure 2.1: DWT on left and IDWT on right

The DWT is computed by combining these building blocks. The signal is transformed j times. The building blocks $\mathbf{T_a}$ are put one after another j times. This procedure is shown in the top panel of Figure (2.2) over two scales. The DWT over j scales means that there are j consecutive direct transforms. The corresponding inverse transform is got putting the building blocks $\mathbf{T_s}$ one after another. This is shown in the lower panel of Figure (2.2). $\mathbf{s_j}$ is a original signal if $j = 1$ otherwise $\mathbf{s_j}$ contains a half of the wavelet coefficients of level j. $\mathbf{s_{j+1}}$ and $\mathbf{d_{j+1}}$ are wavelet coefficients of level $j + 1$.

DWT works properly if it is used for signals of infinite length $\mathbf{s_j} = \{s_j[n]\}_{n\in\mathbb{Z}}$.

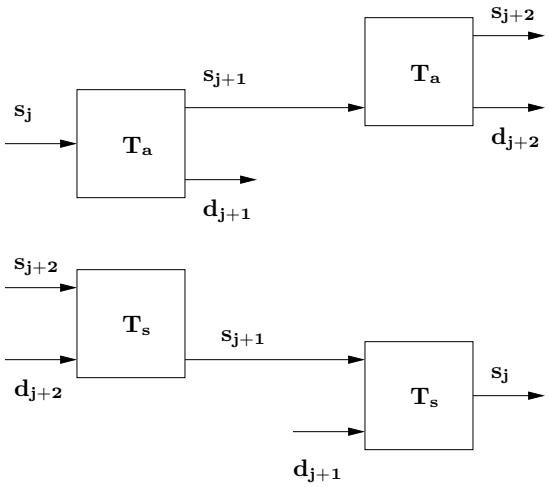

Figure 2.2: DWT (top) and IDWT (bottom) over two scales

The signal $\mathbf{s_j}$ must have finite energy. The condition for finite energy is

$$\sum_{n=-\infty}^{\infty} \left|s[n]\right|^2 < \infty. \tag{2.2}$$

A finite signal of length N can be expressed as a series

$$s[0], s[1], \ldots, s[N-1].$$

A signal of finite length can be seen as an infinite length if zeros are added to the beginning and to the end of the signal. For some integers $n_{first} < n_{last}$, where n_{first} is the first index and n_{last} is the last index of the finite signal, it can be assumed that $s[n] = 0$ when $n < n_{first}$ or $n > n_{last}$. Signals of finite length always satisfy condition of the Equation (2.2), because there is a finite number of non-zero terms in the sum. Each non-zero $s[n]$ must also be finite. In this study signals of finite length are used, but the DWT can be applied according to the previous assumption.

2.2.2 DWT using filters and filter banks

There are at least two possible techniques to define the DWT. The techniques are lifting and filters [23]. These two techniques are actually identical. In

Table 2.1: The Daubechies 4 filter coefficients

	a_0	a_1	a_2	a_3
$\mathbf{h_0}$	-0.1294	0.2241	0.8365	0.4830
$\mathbf{h_1}$	-0.4830	0.8365	-0.2241	-0.1294
$\mathbf{g_0}$	0.4830	0.8365	0.2241	-0.1294
$\mathbf{g_1}$	-0.1294	-0.2241	0.8365	-0.4830

the end, they are just two different ways to represent the same thing. Filters are used in this study and this definition is introduced briefly. Understanding some of the results requires knowledge of the signals and filters frequency representation. These results can be found from the book [32].

The definition starts from a two channel filter bank with the perfect reconstruction property. The filter bank consists of two analysis and two synthesis filters. The analysis filters are $\mathbf{h_0}$ and $\mathbf{h_1}$. The synthesis filters are $\mathbf{g_0}$ and $\mathbf{g_1}$. The all filters are finite impulse response (FIR) filters and index 0 refers to low pass filters and index 1 refers to high pass filters. The building block $\mathbf{T_a}$ consists of the filters $\mathbf{h_0}$ and $\mathbf{h_1}$ and the building block $\mathbf{T_s}$ consists of the filters $\mathbf{g_0}$ and $\mathbf{g_1}$. Several filter banks exist for different purposes. In this study, Daubechies 4 filters are used. The filter coefficients are listed in Table (2.1). These filters construct a two channel multi-rate filter bank. The wavelet packet decomposition is done by filtering signals with the above analysis filters and it is the multi-resolution representation of the signal.

The frequency response $H(e^{j\omega})$ of the filter $\mathbf{h}$ is defined

$$H(e^{j\omega}) = \sum_{n=-\infty}^{\infty} h[n]e^{-j\omega}, \tag{2.3}$$

where ω is the input frequency. $H(e^{j\omega})$ provides a frequency domain description of the filter $\mathbf{h}$. $H(e^{j\omega})$ is the discrete-time Fourier transform of the filter $\mathbf{h}$ and it is a complex series. The associated magnitude response is denoted by $|H(e^{j\omega})|$. From the magnitude response, it can be seen how the different frequencies pass through the filter. The magnitude responses $\left|H_0(e^{j\omega})\right|$ and $\left|H_1(e^{j\omega})\right|$ of the analysis filters $\mathbf{h_0}$ and $\mathbf{h_1}$ are plotted in Figure (2.3) and correspondingly the magnitude responses $\left|G_0(e^{j\omega})\right|$ and $\left|G_1(e^{j\omega})\right|$ of the synthesis filters $\mathbf{g_0}$ and $\mathbf{g_1}$ in Figure (2.4).

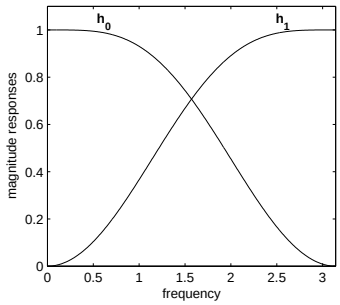

Figure 2.3: The magnitude responses of the analysis filters

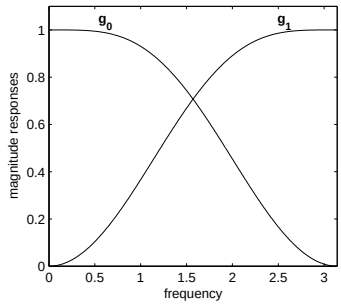

Figure 2.4: The magnitude responses of the synthesis filters

There exists also longer filters in the Daubechies family. Filters approach ideal low and high pass filters as filter length increases. The Daubechies 4 filters are used in this study because the transformed signals are short. The signals length N is only few dozens of points.

Signals $\mathbf{s_{j+1}}$ and $\mathbf{d_{j+1}}$ are calculated by filtering a signal $\mathbf{s_j}$ and down sampling by two. The signals $\mathbf{s_{j+1}}$ and $\mathbf{d_{j+1}}$ consist of approximately a half of the frequency band of the signal $\mathbf{s_j}$. The signal $\mathbf{s_{j+1}}$ consists of a lower part and the $\mathbf{d_{j+1}}$ consists of a higher part of the original frequency band. Down sampling by two can be performed because the frequency band has been halved. This procedure can be continued for the signals $\mathbf{s_{j+1}}$ and $\mathbf{d_{j+1}}$ and the result is the wavelet packet decomposition of the original signal $\mathbf{s_j}$. The filtering and down sampling are represented mathematically as follows

$$s_{j+1}[k] = \sum_n h_0[n] s_j[2k - n], \tag{2.4}$$

$$d_{j+1}[k] = \sum_n h_1[n] s_j[2k - n]. \tag{2.5}$$

The perfect reconstruction property means that the signal $\mathbf{s_j}$ can be recovered exactly from the signals $\mathbf{s_{j+1}}$ and $\mathbf{d_{j+1}}$ by up sampling by two and using the filters $\mathbf{g_0}$ and $\mathbf{g_1}$. The reconstruction is shown in the mathematical form as follows

$$s_j[k] = \sum_n g_0[k - 2n] s_{j+1}[n] + g_1[k - 2n] d_{j+1}[n] \tag{2.6}$$

All filters in the Daubechies family are orthogonal filters. The orthogonal filters have certain advantages. For example, in the transition $\mathbf{s_j} \rightarrow \mathbf{s_{j+1}}, \mathbf{d_{j+1}}$

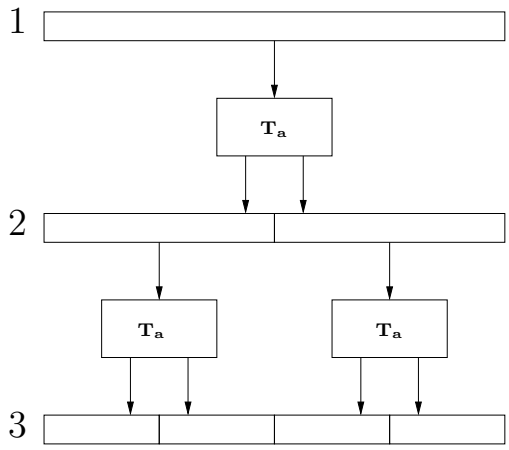

Figure 2.5: The wavelet packet decomposition over three levels or two scales

energy is conserved. The filters cannot be designed independently if the perfect reconstruction property is desired. Filters must satisfy two conditions that the perfect reconstruction property is achieved. The conditions are

$$G_0(z)H_0(z) + G_1(z)H_1(z) = 2, \tag{2.7}$$

$$G_0(z)H_0(-z) + G_1(z)H_1(-z) = 0, \tag{2.8}$$

where $H_0(z), H_1(z), G_0(z)$ and $G_1(z)$ are z-transforms of the filters $\mathbf{h_0}$, $\mathbf{h_1}$, $\mathbf{g_0}$ and $\mathbf{g_1}$. The filters listed in Table (2.1) satisfy the conditions (2.7) and (2.8).

Let a box next to number 1 in Figure (2.5) represent the original signal. The length of the box is proportional to the length of the signal. There are two consecutive transforms $\mathbf{T_a}$ in the Figure (2.5). Both signals from the first transform are transformed again. The procedure could be continued onward. The Figure (2.5) represents the full wavelet packet decomposition over two scales. The signals or wavelet coefficients at the level three characterize different frequency bands of the original signal. Let us assume that the frequency band at the level one is $[0 \ldots \pi]$. Then at the level two the box on left represents approximately the frequency band $[0 \ldots \pi/2]$ and the box on right represents the frequency band $[\pi/2 \ldots \pi]$ of the original signal. Finally, the boxes at the level three from left to right represent approximately the frequency bands $[0 \ldots \pi/4]$, $[\pi/4 \ldots \pi/2]$, $[\pi/2 \ldots 3\pi/4]$ and $[3\pi/4 \ldots \pi]$. If directions of the arrows are inverted and the building blocks $\mathbf{T_a}$ are replaced with the blocks $\mathbf{T_s}$ can be computed the IDWT from the wavelet coefficients of the third level.

The slow and the fast variations in the signal can be separated by the full wavelet packet decomposition. It can also be applied to the noise reduction of the original signal.

2.2.3 Noise reduction

One way how the DWT can be applied to the noise reduction is introduced through an example. First, the noise reduction technique is used to a synthetic signal. In the end, it is applied to a signal which consists of real measurements from an industrial process.

The synthetic signal is a sine function $\sin(4\pi t)$ with $0 \leq t < 1$. The signal is sampled at 512 equidistant points in $0 \leq t < 1$. Normally distributed random noise with a mean $\mu = 0$ and a standard deviation $\sigma = 0.17$ is added to the signal. The signal is plotted on the top panel of the Figure (2.6). The horizontal axis is labeled by the sample index. In addition, three impulses are also added to the signal. The impulses indices are 60, 200 and 400. A symbol $\mathbf{s_1}$ represents the generated signal.

The full wavelet packet decomposition of the signal $\mathbf{s_1}$ over two scales is in the Figure (2.6). The procedure is same as it was in the Figure (2.5). The objective is to remove the noise from the signal. This is done by approximating the wavelet coefficients at the third level. However, the coefficients $\mathbf{s_3}$ are left unchanged since they have information about the original shape of the signal. In other words, $\mathbf{s_3}$ represents the lowest quarter of the signal's frequency band. The noise is mainly concentrated in higher frequencies. In each signal $\mathbf{d_3}$, $\mathbf{d_2^{low}}$ and $\mathbf{d_2^{high}}$ $p\%$ of the coefficients are left unchanged and the rest are set to zero. The unchanged coefficients have the largest absolute values of each signal.

In this example, 5% of the absolute values are left unchanged and the result is the approximated signals $\mathbf{d_3'}$, $\mathbf{d_2^{'low}}$ and $\mathbf{d_2^{'high}}$. The plots of these signals and $\mathbf{s_3}$ are at the first row in the Figure (2.7). The signals $\mathbf{s_2'}$ and $\mathbf{d_2'}$ are calculated from the signals $\mathbf{s_3}$, $\mathbf{d_3'}$, $\mathbf{d_2^{'low}}$ and $\mathbf{d_2^{'high}}$ by the IDWT. The plots of the signals $\mathbf{s_2'}$ and $\mathbf{d_2'}$ are depicted at the middle row in the Figure (2.7). The denoised signal $\mathbf{s_1'}$ is computed from the signals $\mathbf{s_2'}$ and $\mathbf{d_2'}$ by the IDWT. The denoised signal $\mathbf{s_1'}$ is plotted in Figure (2.7) at the bottom panel. When the signals $\mathbf{s_1}$ and $\mathbf{s_1'}$ are compared, it can be seen that the sine shape and the impulses are preserved which are the most important parts of the signal $\mathbf{s_1}$. The noise has instead diminished significantly.

Another example is similar than before except now the signal $\mathbf{s_1}$ consists

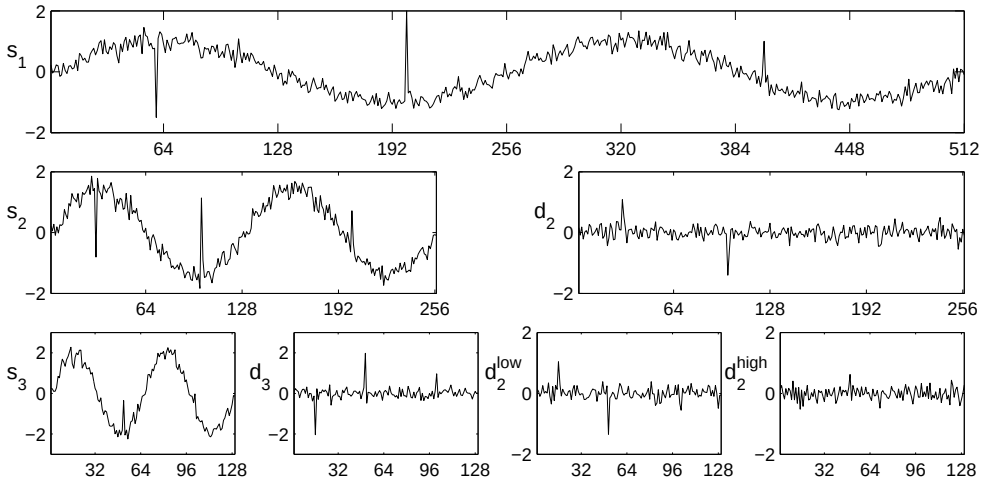

Figure 2.6: The full wavelet packet decomposition over two scales of the synthetically generated signal

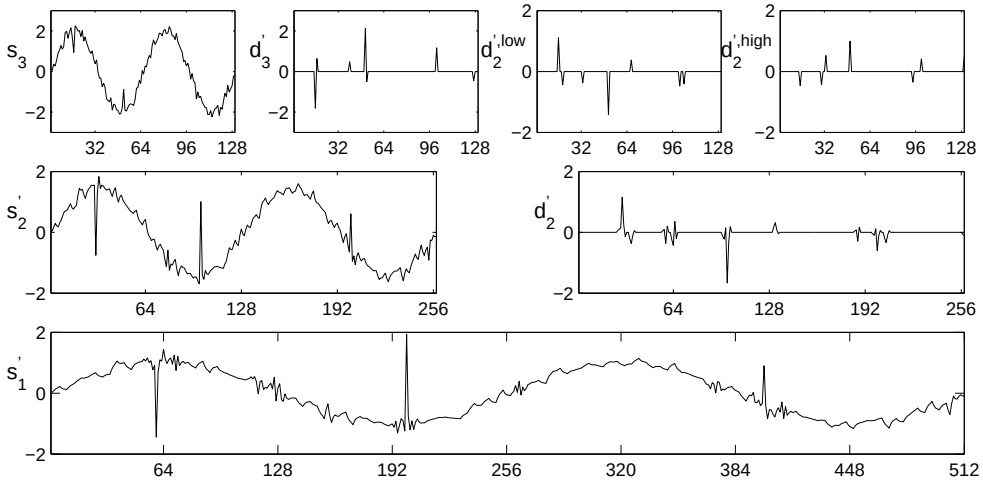

Figure 2.7: The wavelet coefficients which were chosen and the denoised signal s_1'

of real measurements. The measurements are pH values from an industrial process. The noise is assumed to be Gaussian distributed with zero mean

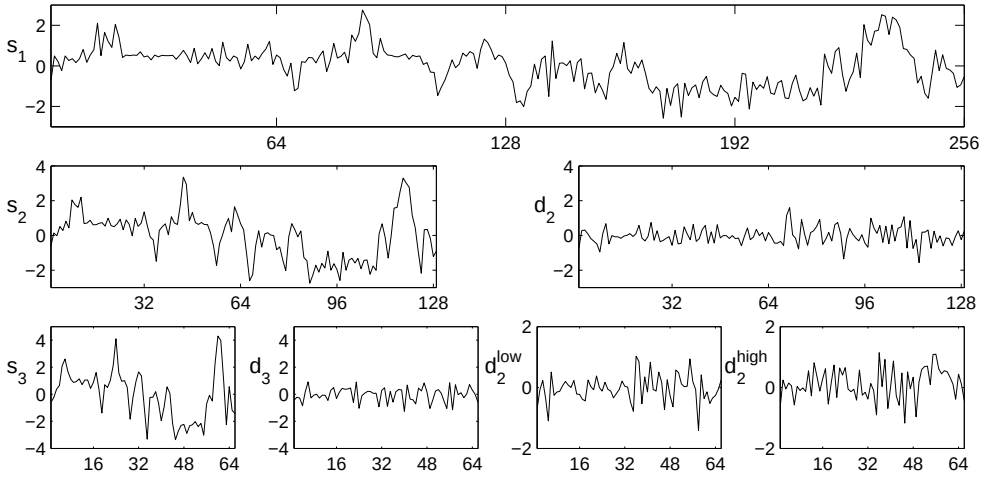

Figure 2.8: The full wavelet packet decomposition over two scales of the real signal from an industrial process.

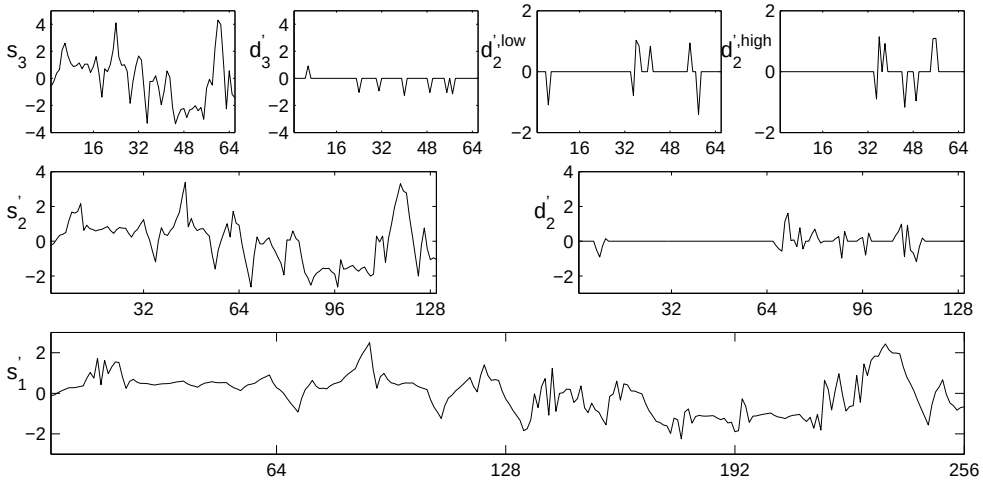

Figure 2.9: The wavelet coefficients which were chose and the denoised signal s_1'

and an unknown standard deviation. The full wavelet decomposition over two scales is illustrated in Figure (2.8). In this case, the $\mathbf{s_3}$ and 10% of the absolute values of the wavelet coefficients $\mathbf{d_3}$, $\mathbf{d_2^{low}}$ and $\mathbf{d_2^{high}}$ are left unchanged and

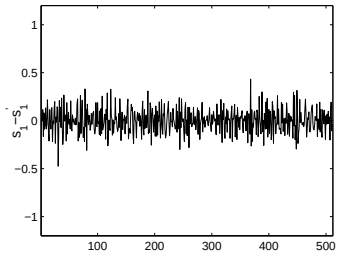

Figure 2.10: Residuals between the synthetic signal and the denoised signal

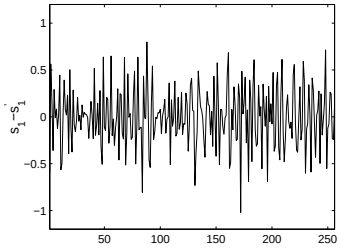

Figure 2.11: Residuals between the real measurements and the denoised estimates

the rest are set to zero. The unchanged coefficients have the largest absolute values. The plots of the approximated signals are drawn in the Figure (2.9). In the bottom panel of the same figure is a plot of the approximated signal $\mathbf{s'_1}$. Comparing $\mathbf{s'_1}$ with $\mathbf{s_1}$ can be seen that the structure of the signal $\mathbf{s_1}$ is preserved and the noise has reduced.

Residuals for the original signal $\mathbf{s_1}$ and the denoised signal $\mathbf{s'_1}$ are

$$\mathbf{r} = \mathbf{s_1} - \mathbf{s'_1}. \tag{2.9}$$

A plot of the residuals $\mathbf{r}_{synt}$ for the synthetically generated signal is in the Figure (2.10) and a plot of the residuals $\mathbf{r}_{real}$ for the real measurements is in the Figure (2.11). The sequences of residuals are removed noise. In the sequences of the residuals do not exist a significant structure or any important information are not lost in the noise reduction process.

There exist also more sophisticated methods for the noise removal in the literature such as soft- and hard-thresholding [11], [12]. The noise level is estimated from the wavelet coefficients at the highest frequency band. In the hard-thresholding, wavelet coefficients whose absolute values are under approximated noise level are set to zero. In the soft-thresholding, all wavelet coefficients are modified. The coefficients under the noise level are decreased and the coefficients above the noise level are increased. A generalized version of the previous hard-thresholding method is presented in the article [27]. It is called shift invariant nonorthogonal wavelet transform. It is possible to achieve smooth and accurate estimates simultaneously using the generalized hard-thresholding. There is a trade-off between the smoothness and the accuracy in the hard-thresholding method [12]. An increased computational complexity is the disadvantage of the method [27].

Chapter 3

Methods

3.1 Multiple linear regression

Using the multiple linear regression the linear dependency can be analyzed between a single dependent variable and several independent variables. The values of the single dependent variable are explained by the values of the independent variables. The dependent variable have to be selected beforehand. The objective is to find weights or regression coefficients for the independent variables that those linear combination explains the variance of the dependent variable as well as possible according to some criterion. The weights describe the relative importance of the each independent variable in the multiple linear regression model [16].

The multiple linear regression model can be written as [33]

$$y_t = \beta_0 + \beta_1 x_{t,1} + \beta_2 x_{t,2} + \ldots + \beta_k x_{t,k} + \epsilon_t. \tag{3.1}$$

In the Equation (3.1), y_t is the dependent variable, $x_i, i = 1, \ldots, k$ are the independent variables, $\beta_i, i = 1, \ldots, k$ are the corresponding regression coefficients, β_0 is the constant or the intercept term of the equation, and ϵ_t is the error term. The index $t = 1, \ldots, N$ represents the tth observation of the variables y and x_i and N is the sample size.

The Equation (3.1) can be written in the matrix form as follows [19]

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}. \tag{3.2}$$

$\mathbf{X}$ is the $N \times (1 + k)$ matrix. In the matrix $\mathbf{X}$, the tth row consists of the tth input vector with a 1 in the first position, so the tth row of the matrix $\mathbf{X}$ is

$[1\ x_{t,1}\ x_{t,2}\ \ldots\ x_{t,k}]$. The $(N \times 1)$-vectors $\mathbf{y}$ and $\boldsymbol{\epsilon}$ contain the outputs and the error terms, respectively. $\boldsymbol{\beta}$ is the $(k+1) \times 1$-vector including the constant term and the regression coefficients, $\boldsymbol{\beta} = [\beta_0\ \beta_1\ \ldots\ \beta_k]^T$.

The regression coefficients are estimated from the data set $(\mathbf{x}_1, \ldots, \mathbf{x}_k, \mathbf{y})$. Let the vector $\mathbf{b} = [b_0\ b_1\ \ldots\ b_k]^T$ be the estimate of the estimator $\boldsymbol{\beta}$. The estimate of the dependent variable is then

$$\hat{\mathbf{y}} = \mathbf{X}\mathbf{b}. \tag{3.3}$$

The most common estimation technique is least squares. In the least squares method the residual sum of squares is minimized . The residuals are the difference of the original values of $\mathbf{y}$ and the estimated values $\hat{\mathbf{y}}$. The least squares condition can be written as follows

$$\begin{aligned} RSS(\mathbf{b}) &= \sum_{t=1}^{N}(y_t - \hat{y}_t)^2 = \sum_{t=1}^{N}(y_t - b_0 - \sum_{j=1}^{k} b_j x_{t,j})^2 \\ &= ||\mathbf{y} - \hat{\mathbf{y}}||^2 = ||\mathbf{y} - \mathbf{X}\mathbf{b}||^2. \end{aligned} \tag{3.4}$$

The Equation (3.4) is a quadratic function in the $k+1$ parameters. Differentiating with respect to $\mathbf{b}$ one obtains

$$\frac{\partial RSS}{\partial \mathbf{b}} = -2\mathbf{X}^T(\mathbf{y} - \mathbf{X}\mathbf{b}). \tag{3.5}$$

Setting Equation (3.5) equal to zero and solving respect to $\mathbf{b}$ the estimates of the regression coefficients can be computed. The unique solution is

$$\mathbf{b} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}. \tag{3.6}$$

The Equation (3.4) does not make any assumptions about the validity of the model (3.1). By solving the Equation (3.5) the best linear fit to the available data [19] is found. The residuals $\mathbf{r} = \mathbf{y} - \hat{\mathbf{y}}$ provide valuable information about goodness of the linear fit. The regression model is good if a large portion of the variance of $\mathbf{y}$ is explained. In the good models, the variance of the residuals is remarkably lower than the variance of the dependent variable $\mathbf{y}$. The goodness of the fit can be measured using a R-squared (R^2) value or a coefficient of determination, which is defined as follows

$$R^2 = 1 - \frac{\sum_{t=1}^{N}(y_t - \hat{y}_t)^2}{\sum_{t=1}^{N}(y_t - \overline{y})^2} = 1 - \frac{\mathbf{r}^T\mathbf{r}}{||\mathbf{y} - \overline{\mathbf{y}}||^2}, \tag{3.7}$$

where $\overline{y}$ is the mean of the samples $y_t, t = 1, \ldots, N$. The R^2-value is the proportion of the explained variation in $\mathbf{y}$. The R^2-value ranges between 0 and 1. The R^2 is 1 only when all the sample points and the estimated regression plane combine exactly. The R^2-value is 0 when the estimated linear regression model does not explain at all variation in $\mathbf{y}$ [33].

3.1.1 Assumptions of the linear regression model

The sample size N has an impact on estimation. Small sample sizes, for example under 50 sample points, are only appropriate for simple regression models. In that case, computed statistical values can be unreliable, because there are not enough data for accurate results. Very large sample size also causes difficulties. The statistical significance tests can make some variables significant though they actually should be rejected from the model, so statistical tests might become too sensitive.

The sample size affects also generalizability of the result of the regression model. The ratio of observations to number of parameters should be suitable. There does not exists a general rule how much the ratio should be. It should be at least over 20 and in some cases even more. This means that there should be at least 20 observations for each parameter. The sufficient ratio changes according to the data which is in use [16].

The signal to noise ratio has also impact on the estimation. The estimates of the parameters are inaccurate if the level of noise is too high. Some independent variables can also be rejected wrongly from the model because they are covered under noise. Noise can be reduced from the dependent variable before the estimation of the regression coefficients. Also, the noise reduction can be done for the independent variables if they include noise. In this study, the DWT is used to reduce noise. This technique is presented in Chapter 2.

The multiple linear regression model involves a couple of crucial assumptions. Under these assumptions the least squares estimates from Equation (3.6) are unbiased. The following assumptions are needed for the linear regression model (3.1).

- The model (3.1) is valid i.e. there exists the linear dependency between the dependent variable $\mathbf{y}$ and the independent variables $\mathbf{x}_i, i = 1, \ldots, k$.
- The variables $\mathbf{x}_i$ are non-stochastic variables. This assumption is not fulfilled in this study because the variables $\mathbf{x}_i$ are also measurements

and they include noise. The independent variables can be stochastic if two assumptions hold. The distribution of each independent variable is independent on the true regression coefficients. The distribution of each independent variable is independent on the distribution of the error term. These two assumptions can be assumed to hold in this study. The parameters have to be interpreted to be conditional on the given values of $\mathbf{x}_i$.

- There are not exact linear dependency between any $\mathbf{x}_i$ and $\mathbf{x}_j$, $i \neq j$. This assumption guarantees that the matrix $\mathbf{X}^T\mathbf{X}$ is a positive definite and then its inverse matrix exists and there exists an unique solution for the Equation (3.6).

- The error term has expected value of zero and the equal variance for all observations. Mathematically this is $E(\epsilon_t) = 0$ and $E(\epsilon_t^2) = \sigma^2, \forall\ t$.

- The random error terms are also assumed to be uncorrelated i.e $E(\epsilon_i \epsilon_j) = 0, \forall\ i \neq j$.

- The error terms are normally distributed $\epsilon_t \sim N(0, \sigma^2)$.

The assumption of the normality of the error term is important for the statistical testing of the model. The Gauss-Markov theorem states that even without this assumption the parameter estimates are the best linear unbiased estimates of the parameters. The best linear unbiased estimate means that it has the minimum variance of all linear unbiased estimators [33].

The statistical assumptions are crucial for statistical testing. The statistical tests cannot be used to validate model if the previous statistical assumptions are not fulfilled. The multiple linear regression model can also be estimated even if the assumptions are not valid, but then the validation of the model have to be done in another way. In this study, the statistical tests are not used to validate model. Descriptions of the tests is found from the book [33].

3.2 Sparse regression

The usual situation is that the available data are $(x_{t,1}, \ldots, x_{t,k}, y_t)$, where $t = 1, \ldots, N$, and the linear regression model should be estimated. $\mathbf{y}$ is the

dependent variable and $(\mathbf{x}_1, \ldots, \mathbf{x}_k)$ are a set of the possible independent variables. The ordinary least squares (OLS) estimates are calculated by minimizing Equation (3.4) using the all independent variables in the model. The OLS estimates are not typically satisfactory. The number of possible independent variables might be large and there can be non-informative variables among them.

The OLS estimates have a low bias but a large variance. The large variance impairs the prediction accuracy. The prediction accuracy can sometimes be improved setting some regression coefficients to zero. The models with too many independent variables are difficult to interpret. The objective is many times to find a smaller subset of independent variables that have the strongest effect in the regression model [35].

The familiar algorithms for improving OLS estimates are forward selection [16], backward elimination [16], subset selection [19], ridge regression [21] and nonnegative garrote [4], which all have some drawbacks.

In the forward selection algorithm one variable is added to the model at each stage. The backward elimination starts from the model where all independent variables are included. During the algorithm one independent variable is eliminated at each stage. It is not possible to change any earlier decisions at the later stage in either algorithm. Both algorithms are also too greedy or they take too long steps towards the final model at each stage during the execution of the algorithm [16].

In the subset selection only a subset of the independent variables is included to the model. The objective is to find the best subset of size $l < k$. The goodness of the subset is measured by the least squares criterion. The subset which minimize the value of the Equation (3.4) is chosen. This is an inefficient approach if k is large. The subset selection is not robust because small changes in the data can result in very different models [19].

Ridge regression is a continuous process that shrinks the regression coefficients. There is added a penalty on the magnitude of the regression coefficients. In this approach, a penalized residual sum of squares is minimized. The penalty term is the sum of the squares of the regression coefficients. Mathematically this is

$$RSS(\mathbf{b}) = ||\mathbf{y} - \mathbf{X}\mathbf{b}||^2 + \alpha \sum_{j=1}^{k} |b_j|^\gamma, \tag{3.8}$$

where $\gamma = 2$ and α is a positive constant. The solution for the Equation (3.8)

is

$$\mathbf{b} = (\mathbf{X}^T\mathbf{X} + \alpha\mathbf{I})^{-1}\mathbf{X}^T\mathbf{y}, \tag{3.9}$$

where $\mathbf{I}$ is an identity matrix. The larger α compels the regression coefficients to shrink more. Ridge regression does not necessarily set any of the coefficients exactly to zero, so the models are still hard to interpret. The ridge regression solution is not sensitive for little changes in the data. Ridge regression is presented in the article [21].

The similar penalization term than in ridge regression is used with the artificial neural networks (ANN) [2]. In context of the ANNs it is called weight decay. The penalty is calculated over all adaptive parameters or weights of the network. It has been justified empirically that the penalization can lead to improvements in network generalization.

Nonnegative garrote has similar features than subset selection and ridge regression. Nonnegative garrote is presented in the article [4]. In the garrote is minimized following Equation

$$RSS(\mathbf{b}) = \sum_{t=1}^{N}(y_t - \sum_{j=1}^{k} c_j b_j x_{t,j})^2, \tag{3.10}$$

under the constraints $c_j \geq 0$ and $\sum_{j=1}^{k} c_j \leq s$. s is some positive constant. Nonnegative garrote sets some regression coefficients to zero and shrinks other. A smaller s means that more of the regression coefficients are set to zero. The nonnegative garrote estimates are also relatively stable. The constant term b_0 is excluded in the Equations (3.8) and (3.10) because the variables are assumed to have zero mean.

3.2.1 Lasso

The lasso algorithm is almost similar than ridge regression but in the lasso some coefficients are set to zero. Lasso is abbreviation from the words least absolute shrinkage and selection operator. The lasso algorithm is presented by Tibshirani in the article [35].

The lasso minimizes the residual sum of squares subject to sum of the absolute values of regression coefficients being less than a predefined constant. The object function is still Equation (3.4), but the minimum is calculated

subject to

$$\sum_{j=1}^{k} |b_j| \leq \tau, \tag{3.11}$$

where $\tau \geq 0$ is a tuning parameter. The equivalent expression for the problem is Equation (3.8) when $\gamma = 1$. There does not exist a simple solution like in Equation (3.9) when $\gamma = 1$, because the penalty term is not continuous in this case. The parameters α and τ are related to each other by a one-to-one mapping [19]. There exists an unique pair of τ and λ for each solution. In the extreme case, same solutions are obtained when $\alpha = \infty$ and $\tau = 0$. The solution is $\mathbf{b} = \mathbf{0}$ in that case.

It is assumed that available data $(\mathbf{x}_1, \ldots, \mathbf{x}_k, \mathbf{y})$ are standardized. The dependent variable $\mathbf{y}$ is standardized to have zero mean or

$$\frac{1}{N} \sum_{t=1}^{N} y_t = 0. \tag{3.12}$$

All independent variables are standardized to have zero mean and unit length

$$\frac{1}{N} \sum_{t=1}^{N} x_{i,t} = 0 \quad \text{and} \quad \frac{1}{N} \sum_{t=1}^{N} x_{i,t}^2 = 1, \forall\ i. \tag{3.13}$$

The intercept term b_0 is zero due to standardization of the variables. It can be dropped out from the model (3.1) without loss of generality.

The tuning parameter τ controls the amount of the shrinkage that is applied to the coefficients. $\tau_0 = \sum |b_j^0|$, where b_j^0 is OLS estimates. The parameter values $\tau < \tau_0$ cause some coefficients to shrink towards zero. The value of τ have to be set before estimation. The value of τ, which causes some coefficients to be exactly zero, is strongly dependent on the data. The article [35] presents some data-based methods to estimate the value of τ.

Computation of the lasso estimates is a quadratic programming problem. The objective function is (3.4) and the linear inequality constraint is (3.11). The efficient and stable algorithm for solving the problem is presented in the article [35]. The algorithm is briefly described below.

The condition $\sum |b_j| \leq \tau$ is equivalent to $\boldsymbol{\delta}_i \mathbf{b} \leq \tau$, where $\boldsymbol{\delta}_i, i = 1, \ldots, 2^k$, are the k-tuples of the form $(\pm 1, \pm 1, \ldots, \pm 1)$. For given $\mathbf{b}$, let $E = \{i : \boldsymbol{\delta}_i \mathbf{b} = \tau\}$. The set E is the equality set corresponding to those constraints which are exactly met. Let $\mathbf{G}_E$ be the matrix whose rows are $\boldsymbol{\delta}_i, \forall\ i \in E$, and let $\mathbf{1}$ be

a vector of ones of length equal to the number of rows of $\mathbf{G}_E$. The algorithm proceeds as follows.

1. Start with $E = \{i_0\}$ where $\boldsymbol{\delta}_{i_0} = \text{sign}(\hat{\mathbf{b}}^0)$, where $\hat{\mathbf{b}}^0$ is OLS estimates without constraints.
2. Find $\hat{\mathbf{b}}$ to minimize $g(\mathbf{b}) = \sum_{t=1}^{N}(y_t - \sum_{j=i}^{k} b_j x_{t,j})^2$ subject to $\mathbf{G}_E\mathbf{b} \leq \tau\mathbf{1}$.
3. While $\sum_{j=1}^{k} |b_j| > \tau$,
4. add i to the set E, where $\boldsymbol{\delta}_i = sign(\hat{\mathbf{b}})$. Find $\hat{\mathbf{b}}$ to minimize $g(\mathbf{b})$ subject to $\mathbf{G}_E \leq \tau\mathbf{1}$.

3.2.2 Forward stagewise linear regression

The lasso algorithm cannot be used if the number of possible independent variables is large. Forward stagewise method can be used instead of lasso in that case. Forward stagewise method approximates the effect of the lasso constraint (3.11).

New independent variables are added sequentially to the model in the forward stagewise algorithm. Only one parameter is adjusted at each iteration. Forward stagewise linear regression is presented in the book [19]. The algorithm is described below.

1. Initialize $b_j = 0, j = 1, \ldots, k$. Set $\epsilon > 0$ to a small constant and M a large one.
2. for $m = 1$ to M
   (a) $(\beta^*, j^*) = \arg\min_{\beta,j} \sum_{t=1}^{N}(y_t - \sum_{l=1}^{k} b_l x_{t,l} - \beta x_{t,j})^2$
   (b) $b_{j^*} \leftarrow b_{j^*} + \epsilon \cdot sign(\beta^*)$
3. $\hat{\mathbf{y}} = \sum_{l=1}^{k} b_l \mathbf{x}_l$

It is assumed that the dependent and the independent variables are scaled according to Equations (3.12) and (3.13). Two constants ϵ and M have to be set before iterations. ϵ should be a small constant. One parameter is adjusted amount of ϵ at each iteration. M is the number of the iterations. Typically ϵ is about 0.05 and M can be from a few hundred to several thousands. The all

parameters $b_j, j = 1, \ldots, k$ are set to zero in the beginning. This means that the tuning parameter τ is zero in the lasso.

The expression $\sum_{l=1}^{k} b_l x_{t,l}$ is the estimated output and $r_t = y_t - \sum_{l=1}^{k} b_l x_{t,l}$ is the current residual at the index t. At each iteration step the independent variable x_{j^*} and the coefficient β^* which minimize the equation on the line (2a) are found. The corresponding regression coefficient b_{j^*} is increased or decreased by ϵ on the line (2b). Other coefficients b_j, $j \neq j^*$ are left unchanged. This procedure is continued M iteration steps or until $\beta^* = 0$. $\beta^* = 0$ can occur if $k < N$ and then values of the coefficients are equal to OLS estimates. This corresponds to $\tau = \tau_0$ in the lasso algorithm.

Many of the coefficients b_j might be zero after M iterations. It means that corresponding independent variables are not yet added to the regression model. This M iteration solution is almost similar than the lasso solution with some tuning parameter τ. The forward stagewise and the lasso solutions are even identical in some cases [19].

3.2.3 Least angle regression

Least angle regression (LARS) is a new model selection algorithm. It is presented in the article [14]. LARS is closely related to the traditional forward selection algorithm but it is less greedy version of it. LARS is also very similar as the lasso and the forward stagewise linear regression algorithms.

In the forward stagewise algorithm numerous tiny steps are taken towards a final model. LARS moves on the same path as the forward stagewise but steps are much larger. The steps are not as large as in the forward selection algorithm, though.

In the Figure (3.1), the progress of the LARS algorithm is presented. It is assumed that the dependent variable is scaled according to Equation (3.12) and the independent variables according to Equations (3.13). In the beginning, all coefficients b_j are zero and a estimate of the $\mathbf{y}$ is $\hat{\mathbf{y}}_0$. The independent variable which is most correlated with $\mathbf{y}$ is found at first. Let it be $\mathbf{x}_{j_1}$. The largest possible step is taken in the direction of $\mathbf{u_1}$ until some other independent variable $\mathbf{x}_{j_2}$ is as much correlated with current residuals as $\mathbf{x}_{j_1}$. $\mathbf{u}_1$ is the unit vector in the direction of $\mathbf{x}_{j_1}$. The estimate of the $\mathbf{y}$ is $\hat{\mathbf{y}}_1$ and the residuals are $\mathbf{y} - \hat{\mathbf{y}}_1$ at this point. LARS proceeds in a direction equiangular between $\mathbf{x}_{j_1}$ and $\mathbf{x}_{j_2}$. Let this direction be $\mathbf{u}_2$. LARS proceeds in the direction of $\mathbf{u}_2$ until some third independent variable $\mathbf{x}_{j_3}$ is as much correlated with the current

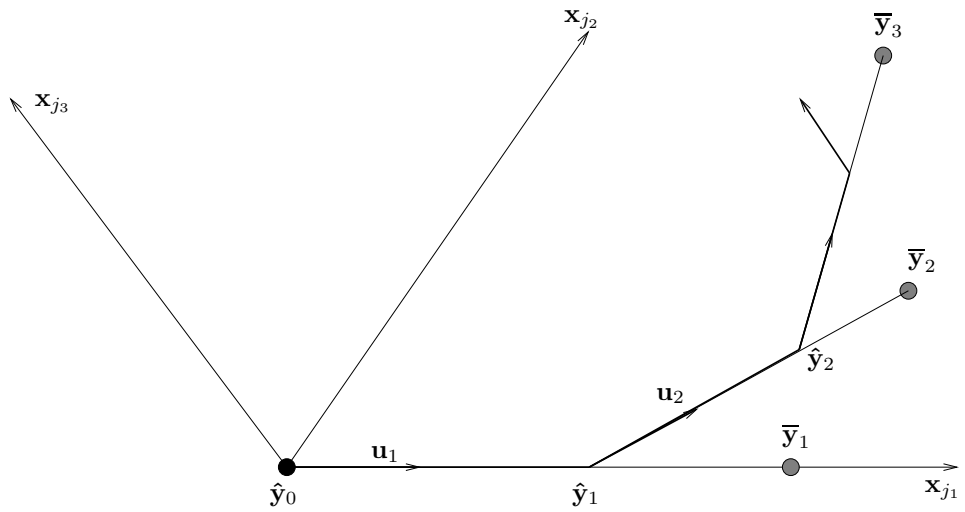

Figure 3.1: Progress of the LARS algorithm

residuals as $\mathbf{x}_{j_1}$ and $\mathbf{x}_{j_2}$. This point is $\hat{\mathbf{y}}_2$. Then LARS proceeds equiangularly between $\mathbf{x}_{j_1}$, $\mathbf{x}_{j_2}$ and $\mathbf{x}_{j_3}$ until the fourth independent variable can be taken into the model. This procedure can be continued as long as there are still independent variables left.

In the LARS algorithm k steps are only needed for a full set of solutions, where k is the number of the independent variables. It is assumed that the independent variables $(\mathbf{x}_1, \ldots, \mathbf{x}_k)$ are linearly independent. First, a couple of variables which are needed in the algorithm are described.

Let $\mathcal{A}$ be a subset of the indices $\{1, 2, \ldots, k\}$. $\mathcal{A}$ includes indices of the independent variables which are already added to the model. $\mathcal{A}$ defines the matrix

$$\mathbf{X}_{\mathcal{A}} = (\cdots \; s_j \mathbf{x}_j \cdots)_{j \in \mathcal{A}}, \tag{3.14}$$

where the signs s_j equal ± 1.

$$\mathcal{G}_{\mathcal{A}} = \mathbf{X}_{\mathcal{A}}^T \mathbf{X}_{\mathcal{A}} \;\text{ and }\; A_{\mathcal{A}} = (\mathbf{1}_{\mathcal{A}}^T \mathcal{G}_{\mathcal{A}}^{-1} \mathbf{1}_{\mathcal{A}})^{-\frac{1}{2}}, \tag{3.15}$$

where $\mathbf{1}_{\mathcal{A}}$ is a vector of 1's. The length of the vector $\mathbf{1}_{\mathcal{A}}$ is same as the size of $\mathcal{A}$. The equiangular vector between the added independent variables is

$$\mathbf{u}_{\mathcal{A}} = X_{\mathcal{A}} w_{\mathcal{A}} \;\text{ where }\; w_{\mathcal{A}} = A_{\mathcal{A}} G_{\mathcal{A}}^{-1} \mathbf{1}_{\mathcal{A}}. \tag{3.16}$$

$\mathbf{u}_\mathcal{A}$ is the unit vector and it makes equal angles with the columns of $\mathbf{X}_\mathcal{A}$. The angles are less than 90°. Mathematically this is

$$X_\mathcal{A}^T \mathbf{u}_\mathcal{A} = A_\mathcal{A} \mathbf{1}_\mathcal{A} \quad \text{and} \quad ||\mathbf{u}_\mathcal{A}||^2 = 1. \tag{3.17}$$

In the beginning $\hat{\mathbf{y}} = \mathbf{0}$. Let $\hat{\mathbf{y}}_\mathcal{A}$ be the current LARS estimate of $\mathbf{y}$. Let

$$\hat{\mathbf{c}} = \mathbf{X}^T(\mathbf{y} - \hat{\mathbf{y}}_\mathcal{A}) \tag{3.18}$$

be the vector of current correlations between the independent variables and the current residuals. The set $\mathcal{A}$ contains indices whose corresponding independent variables have the greatest current absolute correlations,

$$\hat{C} = \max_j \{|\hat{c}_j|\} \quad \text{and} \quad \mathcal{A} = \{j : |\hat{c}_j| = \hat{C}\}. \tag{3.19}$$

Signs in the Equation (3.14) are

$$s_j = \text{sign}\{\hat{c}_j\} \quad \text{for} \quad j \in \mathcal{A}. \tag{3.20}$$

$\mathbf{X}_\mathcal{A}$, $A_\mathcal{A}$ and $\mathbf{u}_\mathcal{A}$ can now be computed as in Equations (3.14)-(3.16). An inner product vector $\mathbf{a}$ is defined as follows

$$\mathbf{a} = \mathbf{X}^T \mathbf{u}_\mathcal{A}. \tag{3.21}$$

The new estimate of $\mathbf{y}$ is now

$$\hat{\mathbf{y}}_{\mathcal{A}_+} = \hat{\mathbf{y}}_\mathcal{A} + \hat{\gamma} \mathbf{u}_\mathcal{A}, \tag{3.22}$$

where

$$\hat{\gamma} = \min_{j \in \mathcal{A}^c} \left\{ \frac{\hat{C} - \hat{c}_j}{A_\mathcal{A} - a_j}, \frac{\hat{C} + \hat{c}_j}{A_\mathcal{A} + a_j} \right\} \tag{3.23}$$

The minimum is taken over only positive components for each j in the Equation (3.23).

The estimates of the regression coefficients b_j are updated at each step for each $j \in \mathcal{A}$.

$$b_{\mathcal{A}_j}^{new} = b_{\mathcal{A}_j}^{old} + \hat{\gamma} s_j w_{\mathcal{A}_j}, \tag{3.24}$$

where $\mathcal{A}_j$ refers to index j in the set $\mathcal{A}$. $b_{\mathcal{A}_j}$ is the regression coefficient and $w_{\mathcal{A}_j}$ is the component of the vector $w_\mathcal{A}$ of the corresponding independent variable. s_j is obtained from Equation (3.20) and $\hat{\gamma}$ is same as in the Equation (3.23).

The complete LARS algorithm requires k times the procedure of the Equations (3.18)-(3.24). One independent variable is added to the model at each iteration step. There is exception when the last independent variable is added to the model. The Equation (3.23) is not defined and $\hat{\gamma} = \hat{C}_k / A_\mathcal{A}$ and $\hat{\mathbf{y}}$ and $\mathbf{b}$ are equal to the OLS estimates for the full set of k independent variables.

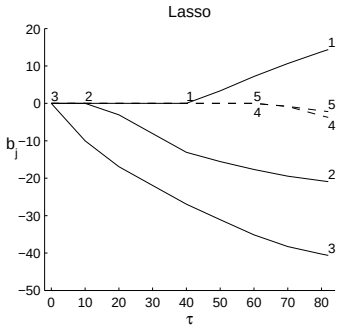

Figure 3.2: Lasso estimates

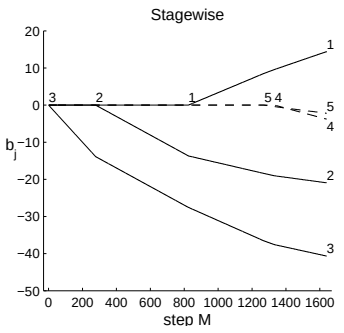

Figure 3.3: Forward stagewise estimates

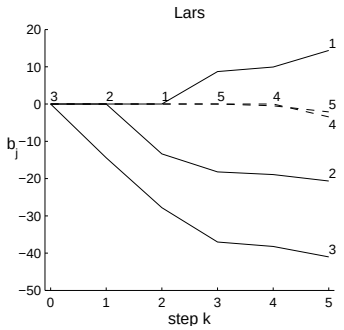

Figure 3.4: LARS estimates

3.2.4 Example of sparse regression

The presented sparse regression algorithms are illustrated with a simple example. The multiple linear regression model was generated synthetically. The model is

$$\mathbf{y} = \beta_1 \mathbf{x}_1 + \beta_2 \mathbf{x}_2 + \beta_3 \mathbf{x}_3 + \boldsymbol{\epsilon}, \tag{3.25}$$

where the error term is normally distributed with zero mean. Two additional variables $\mathbf{x}_4$ and $\mathbf{x}_5$ were taken into the set of possible independent variables.

Lasso estimates are plotted as a function of $\tau = \sum |b_j|$ in the Figure (3.2). In the Figure (3.3), forward stagewise estimates are plotted as a function of the iteration round M. In this case, the number of the iteration rounds was $M = 1640$ and the parameters were adjusted by $\epsilon = 0.05$. In the figure (3.4), LARS estimates are drawn as a function of the step k. The plots of the values

of the regression coefficients are almost identical. The independent variables are added to the model in the same order $3, 2, 1, 5, 4$ in all three algorithms. The variables which actually belong to the model were added first. In the end, the estimates of the regression coefficients from each algorithm are identical to the corresponding OLS estimates.

The disadvantage of the lasso and the forward stagewise algorithms is that parameters τ and M and ϵ have to be set beforehand. It may be difficult to find suitable values for these parameters in some cases. In the LARS algorithm is not any such parameters. LARS is also computationally more efficient than the lasso or the forward stagewise algorithm.

The problem is typically to find the best choice of $\mathbf{b}$ from all the possible solutions which are got from the LARS algorithm. A C_p-type selection criterion is used in the article [14]. The C_p selection criterion is discussed in detail in the articles [28] and [29]. The Akaike information criterion (AIC) is also popular criterion for model selection. AIC and the corrected AIC are presented in the article [6]. An application of the minimum description length (MDL) model selection criterion for the linear regression is demonstrated in the article [18]. The MDL criterion for the linear regression is

$$MDL(k) = \frac{N}{2} \ln ||\mathbf{y} - \hat{\mathbf{y}}||^2 + \frac{k}{2} \ln N. \tag{3.26}$$

k represents kth step of the LARS algorithm in the Equation (3.26). $\hat{\mathbf{y}}$ is the estimate of the dependent variable $\mathbf{y}$ and N is the sample size. The value of the Equation (3.26) is computed at each step in LARS algorithm. The model which has the minimum value for the Equation (3.26) is selected.

The C_p-type and the AIC criteria are closely related to the MDL criterion. The second term of the Equation (3.26) is a penalty on the selected parameters. C_p and AIC differ only in the size of this penalty. MDL is only used for rough selection of the number of the independent variables. The selection of the final number of the variables is done by bootstrapping.

3.3 Bootstrap

The bootstrap is a relatively new statistical method and it was introduced by Efron in the article [13]. The idea of the bootstrap is to use sample data to estimate some statistics of the data. There are no assumptions about the model or the form of the probability distribution in the bootstrap procedure.

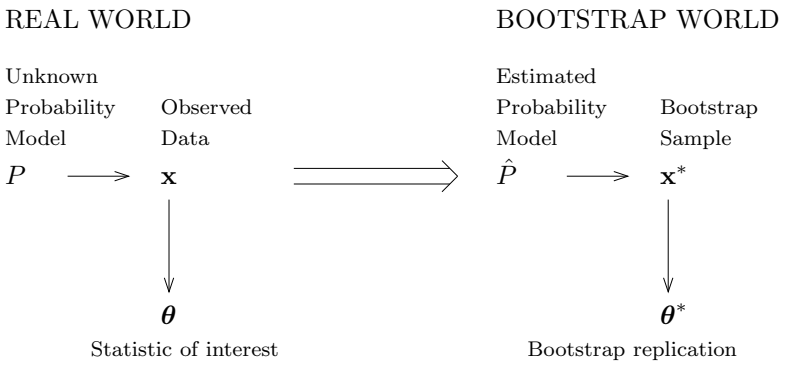

Figure 3.5: A general diagram of the bootstrap method

The statistic of interest and its distribution are computed by resampling the original data with replacement. The bootstrap requires a lot of computing time so it is a computer-intensive method. It is less of a problem nowadays due to the availability of efficient computers.

The jackknife is a less computer-intensive resampling method. The resampling in the jackknife is done without replacement, so the size of the subsamples are smaller than the size of the original data. There are only finite number of subsamles in the data. A review of the jackknife is presented in the article [31].

The bootstrap principle is presented for instance in the book [15]. Figure (3.5) is a general diagram of the bootstrap method. The real world is on the left. $\mathbf{x}$ is the observed sample data, which is got from the unknown probability distribution P. $\boldsymbol{\theta}$ is the statistic of the interest which is function of $\mathbf{x}$. The statistical behavior of $\boldsymbol{\theta}$, for example its standard deviation or the statistical significance, is usually under exploration.

The bootstrap world is on the right side in the Figure (3.5). The estimated probability model $\hat{P}$ gives bootstrap samples $\mathbf{x}^*$ by random sampling. The bootstrap replication $\boldsymbol{\theta}^*$ of the statistic of interest $\boldsymbol{\theta}$ is calculated from $\mathbf{x}^*$ according to a same function as in the real world case. The advantage of the bootstrap is that the replications $\boldsymbol{\theta}^*$ can be calculated as many times as wanted or the number of times which is computationally feasible. The replications $\boldsymbol{\theta}^*$ can be used directly to estimate the statistic of the interest. The observed variability of $\boldsymbol{\theta}^*$ can be used as an estimate of the unknown true standard deviation of $\boldsymbol{\theta}$. The confidence intervals of $\boldsymbol{\theta}$ can also be estimated using the

replications $\boldsymbol{\theta}^*$. Constructing the bootstrap confidence intervals in general situations are discussed in the article [34]. Many examples of applications of constructing the bootstrap confidence intervals in signal processing are found from the article [38].

The double arrow $\Rightarrow$ is a crucial step in the bootstrap process in the Figure (3.5). The unknown probability model P needs to be estimated from the observed data $\mathbf{x}$, this is indicated by $\mathbf{x} \Rightarrow \hat{P}$. There is no general prescription for this process. There are various processes for different data structures. Many examples are found from the book [15]. The step $\hat{P} \rightarrow \mathbf{x}^*$ represents a simulation of the bootstrap data from the estimated probability model $\hat{P}$. An application of the bootstrap in the multiple linear regression is presented in the next section.

3.3.1 Application of the bootstrap in multiple linear regression

Bootstrapping a regression model can be done in two different ways. The methods are called bootstrapping residuals and bootstrapping pairs. Bootstrapping the residuals is briefly described at first. Bootstrapping the pairs is used in this study and it is described after bootstrapping the residuals. These two methods are presented in the book [15].

The multiple linear regression model is presented in Equations (3.1) and (3.2). The probability model $P \rightarrow \mathbf{x}$ for the multiple linear regression has two components $P = (\boldsymbol{\beta}, F)$, where $\boldsymbol{\beta}$ is the parameter vector and F is the probability distribution of the error terms $\boldsymbol{\epsilon}$. The assumptions of the probability distribution of the error terms were introduced in Section 3.1.1. The error terms are assumed to be normally distributed and uncorrelated.

The parameter vector $\boldsymbol{\beta}$ can be estimated by $\mathbf{b}$ according to Equation (3.6). The error terms $\boldsymbol{\epsilon}$ can be approximated by the residuals $\mathbf{r} = \mathbf{y} - \mathbf{X}\mathbf{b}$. The estimate of F is still required for the estimate of P in the bootstrap method in the Figure (3.5). The probability distribution of the error terms F can be estimated by an empirical distribution of the residuals $\mathbf{r}$. The empirical distribution is assumed to be

$$\hat{F} \ : \ \text{probability } \frac{1}{N} \text{ on } r_t \text{ for } t = 1, \ldots, N, \tag{3.27}$$

where N is the sample size.

The bootstrap data sets are now calculated according to the probability model $\hat{P} = (\mathbf{b}, \hat{F})$. The bootstrap error terms are selected randomly at first. The random selection is done with replacement. The bootstrap error terms are

$$\hat{F} \rightarrow (r_1^*, \ldots, r_N^*) = \mathbf{r}^*. \tag{3.28}$$

Each r_t^* equals some of N values of residuals r_t with probability $1/N$ in Equation (3.28). The bootstrap replication of the dependent variable $\mathbf{y}^*$ is calculated as follows

$$\mathbf{y}^* = \mathbf{X}\mathbf{b} + \mathbf{r}^*. \tag{3.29}$$

The bootstrap replication $\mathbf{b}^*$ of the estimate $\mathbf{b}$ is computed respectively than in Equation (3.6). The bootstrap replication is

$$\mathbf{b}^* = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}^*. \tag{3.30}$$

The procedure of Equations (3.28)-(3.30) can be repeated for instance B times. There are B replications of $\mathbf{b}$ in use then. The statistic of the interest is calculated from these B replications.

The independent variables $(\mathbf{x}_1, \ldots, \mathbf{x}_k)$ are treated as fixed quantities in the bootstrapping residuals approach. It is assumed that the error between the dependent variable and its estimate $\hat{\mathbf{y}}$ is not dependent on the independent variables $(\mathbf{x}_1, \ldots, \mathbf{x}_k)$. This is a strong assumption and it can fail even if the Equation (3.1) for the multiple linear regression model is correct. In the bootstrapping pairs approach weaker assumptions about validity of the Equation (3.1) are made as in the bootstrapping residuals approach.

In the bootstrapping pairs, $\hat{F}$ is assumed to be empirical distribution of the observed data vectors $(x_{t,1}, \ldots, x_{t,k}, y_t)$, where $t = 1, \ldots, N$. $\hat{F}$ puts probability $1/N$ on each vector $(x_{t,1}, \ldots, x_{t,k}, y_t)$. A bootstrap sample is now a random sample of size N drawn with replacement from the population of N vectors $(x_{t,1}, \ldots, x_{t,k}, y_t)$. The bootstrap sample is

$$\hat{F} \rightarrow \begin{pmatrix} x_{1,1}^*, & \ldots, & x_{1,k}^*, & y_1^* \\ x_{2,1}^*, & \ldots, & x_{2,k}^*, & y_2^* \\ \vdots, & \vdots, & \vdots, & \vdots \\ x_{N,1}^*, & \ldots, & x_{N,k}^*, & y_N^* \end{pmatrix} = (\mathbf{X}^*, \mathbf{y}^*). \tag{3.31}$$

$(\mathbf{X}^*, \mathbf{y}^*)$ is the resampled version of the original data $(\mathbf{X}, \mathbf{y})$. Some data vectors from the original data $(\mathbf{X}, \mathbf{y})$ can appear zero times or once or twice etc. in

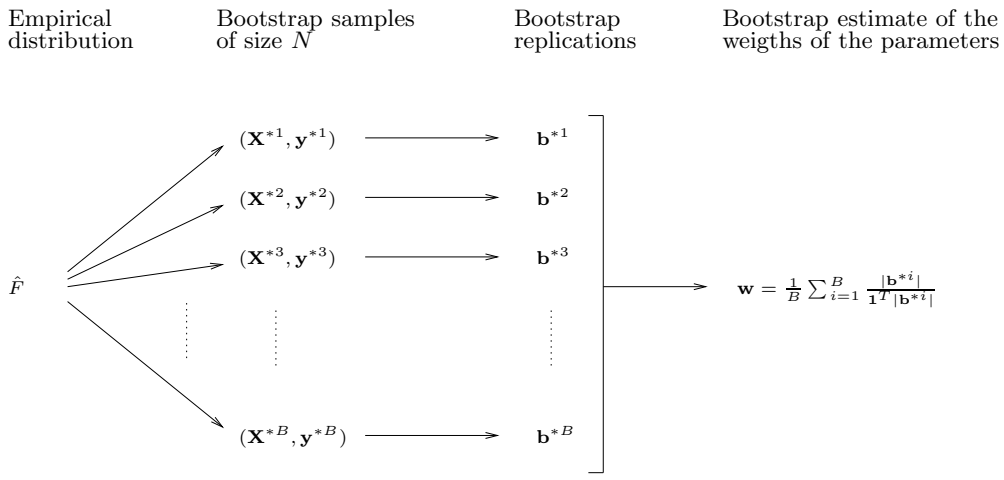

Figure 3.6: A general diagram of bootstrapping pairs

the bootstrap sample $(\mathbf{X}^*, \mathbf{y}^*)$. The bootstrap replication of the estimates of the parameters $\mathbf{b}$ are computed as follows

$$\mathbf{b}^* = (\mathbf{X}^{*T}\mathbf{X}^*)^{-1}\mathbf{X}^{*T}\mathbf{y}^*. \tag{3.32}$$

The order of the samples $(x^*_{t,1}, \ldots, x^*_{t,k})$ are irrelevant in the matrix $\mathbf{X}^*$. The only restriction is that the corresponding values of the independent variables $(x^*_{t,1}, \ldots, x^*_{t,k})$ and the dependent variable y^*_t have to be on the same row in the matrix $\mathbf{X}^*$ and in the vector $\mathbf{y}^*$.

A general diagram of the bootstrapping pairs approach is presented in the Figure (3.6). The empirical distribution $\hat{F}$ is on left. B independent bootstrap samples are drawn from the distribution $\hat{F}$ according to Equation (3.31). The bootstrap replications $\mathbf{b}^{*i}$, $i = 1. \ldots, B$ of the estimates $\mathbf{b}$ are computed according to Equation (3.32). The statistic of interest or some other features of the parameters $\mathbf{b}$ can be calculated from these B bootstrap replications.

In this study, relative weights of the parameters of the multiple linear regression model are computed. The relative weights are computed from the bootstrap replications as follows

$$\mathbf{w} = \frac{1}{B}\sum_{i=1}^{B} \frac{|\mathbf{b}^{*i}|}{\mathbf{1}^T|\mathbf{b}^{*i}|}. \tag{3.33}$$

B is the number of the bootstrap replications and $\mathbf{b}^{*i}$ is ith bootstrap replication of the parameters $\mathbf{b}$. The absolute values are taken over all the components of the parameter vector $\mathbf{b}^{*i}$ in the Equation (3.33). There is a sum of the absolute values of the parameters in the denominator. $\mathbf{1}$ is a vector of ones and the length of the vector $\mathbf{1}$ is the same as the length of the vector $\mathbf{b}^{*i}$. All components of the vector $|\mathbf{b}^{*i}|$ are divided by the previous sum. These operations are done for every bootstrap replication and these scaled bootstrap replications are added together. This sum is divided by the number of the bootstrap samples B. The result is a vector $\mathbf{w}$, which includes the relative weights of the parameters $\mathbf{b}$.

There is a relative weight w_i for the each possible independent variable $\mathbf{x}_i, i = 1, \ldots, k$ in the vector $\mathbf{w}$. The value of the each w_i is within the range $w_i \in [0, 1]$ and the sum of the weights is 1. The relative weight of the independent variable is a measure of the belief that the independent variable belongs to the estimated linear model. The independent variable can be rejected from the estimated model if the value of w_i is zero or under a predefined threshold value. The most significant independent variables have the largest relative weights w_i.

3.4 Learning linear dependency structure

3.4.1 Belief graph

Let us assume now that there are data D available. In the data D there are $k + 1$ variables and N measurements for each variable. The objective is to find multiple linear regression models among the variables. Each variable is selected to be the dependent variable in turn and the rest of the variables are the possible independent variables.

For each dependent variable the best independent variables are searched using the LARS algorithm. The LARS algorithm is presented in the Section 3.2.3. The data are resampled B times according to Equation (3.31) and the LARS algorithm is applied to each bootstrap sample. The rough selection of the independent variables are done for the each bootstrap replication using Equation (3.26). The bootstrap replications of the regression coefficients of the rejected variables are zero. The relative weights of the regression coefficients are computed from the bootstrap replications according to Equation (3.33). The previous steps are repeated for all $k + 1$ linear models. The result of

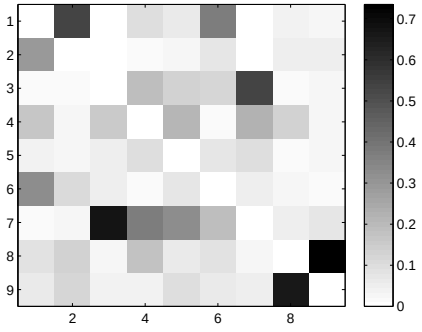

Figure 3.7: Belief graph G_b

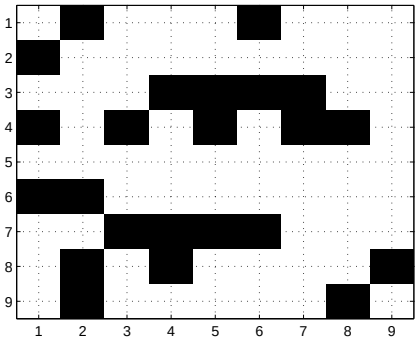

Figure 3.8: Directed graph G_d

the previous procedure is $k + 1$ vectors of the relative weights i.e. the vector of the relative weights for each estimated linear model. A belief graph G_b is constructed from these vectors of the relative weights.

The each signal or variable of the data D is presented as a node in the belief graph G_b. There are $k + 1$ nodes in G_b. The arcs between the nodes are obtained from the nonzero relative weights. The arc between two nodes represents that there exists a linear dependency between these two nodes and the corresponding variables belong potentially to the same multiple linear regression model. The weights of the arcs are got from the values of the relative weights. The weights are a measure of the belief that there exists the linear dependency between two nodes. The belief graph can be represented as an adjacency matrix. The elements of the matrix are weights of the arcs.

In Figure (3.7) is an example of the adjacency matrix of the belief graph G_b. In this case, in the data D have been nine variables. The relative weights of the parameters of the ith model are in the ith column in the adjacency matrix. The ith variable has been the dependent variable in the ith model and rest of the variables have been the possible independent variables. The relative weights for the variables $2, \ldots, 8$, when the variable 1 is the dependent variable, are presented in the first column of the adjacency matrix in the Figure (3.7). The other columns are constructed in a corresponding way. Dark colors refer to strong belief that these variables are significant in the multiple linear regression model. For example, in the first column or in the first regression model the variables 2, 4 and 6 are the most significant independent variables.

The dependencies or the number of the arcs in the adjacency matrix can be reduced by setting some threshold value λ for the relative weights. The relative

weight of the parameter is set to zero if it is below the threshold λ. All the relative weights which are above the threshold λ are set to unity. This means that these dependencies are equally important. The belief graph G_b becomes unweighted directed graph G_d after using the threshold λ. G_d is presented in Figure (3.8). The value of the threshold was $\lambda = 0.1$. In this study, the value of the threshold λ is not estimated according to some defined principle. The suitable value for λ is decided by exploring values of the adjacency matrix. The purpose is to find such a value for λ that a little change in the value of λ would not cause major changes in the graph G_d.

3.4.2 Construction of moral graph

The following idea of constructing an undirected and a moral graph from the belief graph is adapted from the article [22]. The arcs represent in the graph G_d that some variables are independent variables in certain linear models. Let V_i stand for a node or a variable. There are nine nodes $V_i, i = 1, \ldots, 9$ in the graph G_d. The dependencies have directions in the graph G_d, but the directions of the dependencies can be discarded after processing of the graph G_d. When the directions of the dependencies are removed from the graph G_d, an unweighted undirected graph G_u is obtained. An adjacency matrix of the directed graph G_d is in the Figure (3.8) and the corresponding adjacency matrix of the undirected graph G_u is in Figure (3.11). It can be assumed now that two signals belong to the same linear model if they are connected by the arc.

There is a part of the graph G_u in Figure (3.9). When a variable 7 is the dependent variable both variables 5 and 6 are relevant independent variables. Other independent variables can be seen from the adjacency matrix in Figure

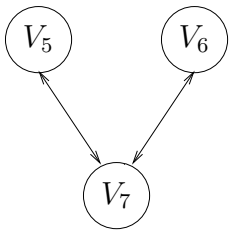

Figure 3.9: A part of the graph G_u

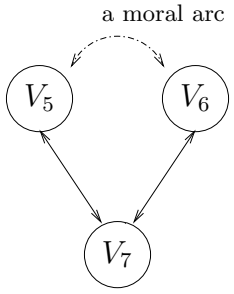

Figure 3.10: A moral arc addition

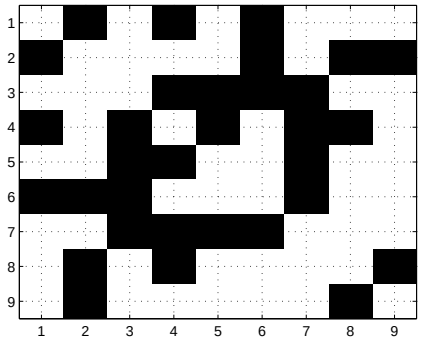

Figure 3.11: Undirected graph G_u

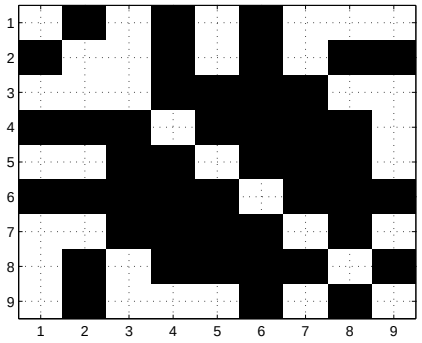

Figure 3.12: Moral graph G_m

(3.11). When the variable 5 or 6 is the dependent variable the variable 7 is one of the independent variables. Let us assume that the variable 7 is the actual dependent variable so $x_7 = \beta_5 x_5 + \beta_6 x_6 + \phi$, where ϕ is a function of the other independent variables and noise. It is possible that all relevant variables are not found by sparse regression algorithms if some independent variable is chosen to be the dependent variable. It is highly possible in this case that all three variables 7, 5 and 6 belong to the same model. An arc can be added to connect the nodes V_5 and V_6. The added arc is called a moral arc. An addition of the moral arc is represented in Figure (3.10). The moral arcs can be added to the unweighted undirected graph G_u according to the following two steps.

1. Create the unweighted undirected graph G_u from the graph G_b by using the threshold value λ and ignoring the directions of the dependencies.

2. Create a moral graph G_m from G_u. For each node V_i, find its parents $\mathcal{P}_{V_i}$ in G'_u. Connect each pair of nodes in $\mathcal{P}_{V_i}$ by adding undirected arcs to G_u.

The parents are sought for each node V_i from the adjacency matrix of G'_u in the step 2. When a parent and child relationships are sought, the directions must be interpret in G_u such that there do not exist cycles. This adjacency matrix is denoted by G'_u. The restriction in this case is that the parent V_i must have a smaller index than the child V_j i.e. $i < j$. This restriction confirms that the relationships can be interpreted correctly and the number of the added moral arcs is reasonable. The graph G_u is called a moral graph G_m after all moral arcs have been added. The adjacency matrix of the moral graph G_m is demonstrated in Figure (3.12).

The probability propagation in trees of clusters method for probabilistic inference is described in the article [22]. It is a method for performing probabilistic inference on a belief network. The belief network and in this study used belief graph have some analogy. The belief network induces a probability distribution over its variables, when the belief graph represents the strength of the dependencies between the variables. The relative weights of the belief graph can be seen as probabilities in a loose sense. The moral graph construction has the same purpose in both cases i.e. to add arcs to the undirected graph G_u and to the undirected graph of the belief network.

3.4.3 Maximal cliques

The objective is to find multiple linear regression models among the variables in the data D. The linear models or the sets of variables are sought from the unweighted undirected graph G_u or from the moral graph G_m. The variables, which are interpreted to belong to the same model, are part of the same maximal clique. A subgraph of G_u or G_m is called a clique if the subgraph is complete and maximal. The subgraph is complete if every pair of nodes in the subgraph are connected by an arc. The clique is maximal if it is not a subgraph of the larger complete subgraph. This definition of the complete and maximal cliques is presented in the article [22].

The interactions between different metabolites are studied in the article [25]. In the article, the linear correlation coefficients are used as estimates how tight interactions are between the metabolites. An undirected correlation matrix is constructed from the correlation coefficients. The interactions are significant between the metabolites which belong to the same maximal clique. An algorithm, which can be used to generate all maximal cliques from arbitrary undirected graph, is presented in detail in the article [25]. The short description of the algorithm is given in the next two paragraphs.

Let n-clique stand for a clique which includes n nodes and C_n for a list of all n-cliques. The algorithm starts by forming all 2-cliques. All pairs of nodes which are connected by an arc are 2-cliques. There exists 3-clique if two 2-cliques have one node in common and two sole nodes are connected. For example, if there are cliques $\{V_1, V_2\}, \{V_1, V_3\}$ and $\{V_2, V_3\}$ in the graph then there exists 3-clique $\{V_1, V_2, V_3\}$. All 3-cliques are collected to the list C_3.

All $(n+1)$-cliques can be constructed from the list C_n. Two n-cliques c_n^1 and c_n^2, which have already $(n-1)$ nodes in common, are tested if they could

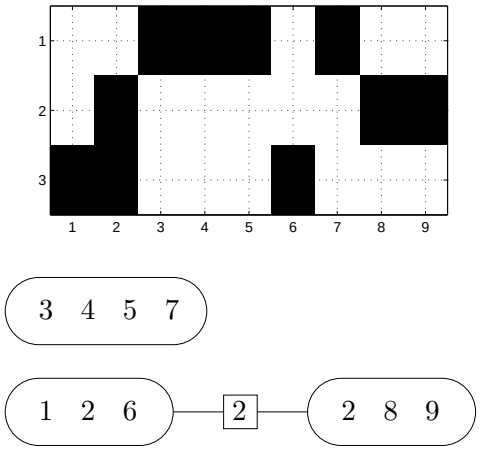

Figure 3.13: Cliques from G_u

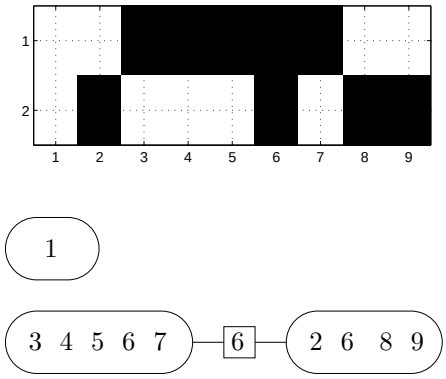

Figure 3.14: Cliques from G_m

form a new $(n+1)$-clique c_{n+1}. There has to exist n-clique c_n^3, which has $(n-2)$ nodes in common with cliques c_n^1 and c_n^2, in the list C_n. Additionally, $(n-1)$th node of c_n^3 has to be equivalent to nth node of c_n^1 and nth node of c_n^3 has to be equivalent to nth node of c_n^2, then there is $(n+1)$-clique c_{n+1} in the graph. For example, if there exist cliques $c_4^1 = \{V_1, V_2, V_3, V_4\}$, $c_4^2 = \{V_1, V_2, V_3, V_5\}$ and $c_4^3 = \{V_1, V_2, V_4, V_5\}$, then there exist 5-clique $c_5 = \{V_1, V_2, V_3, V_4, V_5\}$ in the graph. This procedure is repeated as long as new cliques can be constructed. All lists $C_i, i = 1, \ldots, n_{max}$ are tested in the end, that any n-clique is not a subclique of $(n+m)$-clique, $m > 0$. If there exist subcliques they can be eliminated.

The problem to find all maximal cliques is known to be the NP-hard problem [24]. This means that the computational time for a solution is nondeterministic and the number of cliques can increase exponentially. Several algorithms for solving the clique problem are introduced and analyzed in the article [24]. The computational burden is not too heavy if the number of variables in the data D is not large. The number of variables can be a few hundred. The number of arcs in the moral graph G_m also affects on the computational efficiency.

The maximal cliques from the graphs G_u and G_m are in Figures (3.13) and (3.14). The adjacency matrices of G_u and G_m are in Figures (3.11) and (3.12). On the each row in matrices in the top panels of Figures (3.13) and (3.14), are listed the variables which belong to a certain clique. The numbers of cliques are in the y-axis and the numbers of the variables are in the x-axis.

The variables, which belong to ith clique, are marked by black in ith row. The same information is represented in the bottom panels of the same figures, but the dependencies between the variables and the cliques can be seen more clearly.

Two maximal 3-cliques and one 4-clique or the regression models were found from the graph G_u. The cliques are $c_4^1 = \{V_3, V_4, V_5, V_7\}, c_3^1 = \{V_1, V_2, V_6\}$ and $c_3^2 = \{V_2, V_8, V_9\}$. The variables within a clique are interpreted to be dependent on each other. The cliques c_3^1 and c_3^2 are dependent on each other through the variable 2. The best linear models are achieved according to the R^2-value if the signals 7, 1 and 8 are selected to be the dependent variables. The R^2-value is introduced in Equation (3.7).

From the graph G_m 5-clique $c_5^1 = \{V_3, V_4, V_5, V_6, V_7\}$ and 4-clique $c_4^1 = \{V_2, V_6, V_8, V_9\}$ were found. These cliques are dependent on each other through the variable 6. If the variables 7 and 8 are selected to the dependent variables then the R^2-values for the cliques c_5^1 and c_4^1 are 0.96 and 0.94. The R^2-values indicate that the search of the linear models was successful.

More maximal cliques can be found in both cases, but there are additional criteria how final cliques are selected. The first criterion is that two cliques can have only one variable or node in common. The second one is that the common variable cannot be a dependent variable in both cliques. Finally, cycles are not allowed in the dependency structure. The dependency structure is a dependency tree or a forest under these restrictions. More detailed description of the results and the data D of this example can be found from Section 4.2 in Chapter 4.

Chapter 4

Results

This chapter presents how the previous algorithms work in practice. The algorithms are tested by three different data sets. One of the data sets is generated synthetically and the rest of the data sets consist of real measurements. The real-world data sets are called the System data and the Boston Housing data. First, results from the synthetic data are presented followed by results from the System data and from the Boston housing data.

4.1 Synthetic data

The synthetically generated data sets are a convenient way to test how the method works in practice, since the dependency structure of the data is known beforehand. Thus, the dependency structure which the method returns can be compared to the actual dependency structure. The other qualities of the method can also be studied, because the all features of the data are known. For example, the quality of the noise reduction algorithm can be identified.

In the synthetic data set there are 28 variables. The number of samples in each variable is $N = 1000$. In the data set there are 8 multiple linear regression models. The dependency structure of the variables is presented in Figure (A.1) in Appendix A. The arrows point from the independent variables to the corresponding dependent variables. The dependent variables are 18, 5, 19, 8, 13, 22, 25 and 28. The dependent variables are linear combinations of the independent variables as in the Equation (3.1). The regression coefficients were taken from the uniform distribution $[0.2, 1.2]$ or $[-0.2, -1.2]$. The sign of the coefficient is irrelevant in this case. All the variables are scaled to have

zero mean and equal variance.

Noise was added to all dependent variables. As noise were used random samples, which were normally distributed. The mean and the variance of the random samples were $\mu = 0$ and $\sigma^2 = 0.05$. The noise reduction technique described in the Section (2.2.3) was used. The original noiseless signal $\mathbf{s}$, the noise $\boldsymbol{\epsilon}$ and the noisy signal $\mathbf{s}_n = \mathbf{s} + \boldsymbol{\epsilon}$ are all known in this case. After the noise reduction, the estimate of the noisy signal $\hat{\mathbf{s}}_n$ is also known. The noise, which is left after the noise reduction, can be computed as a difference $\hat{\boldsymbol{\epsilon}} = \hat{\mathbf{s}}_n - \mathbf{s}$. The variance of $\hat{\boldsymbol{\epsilon}}$ is approximately half of the original variance $\sigma^2 = 0.05$ in each 8 cases. This indicates that the power of the noise is halved.

In the method proposed in the Chapter 3 one does not make any assumptions of the number of the dependent variables. The dependent variables are sought during the execution of the algorithm. Each variable of total 28 is kept in turn as a dependent variable. The rest of the variables are possible independent variables. The most significant independent variables are sought using the model selection algorithm LARS. The LARS algorithm is presented in the Section 3.2.3. The number of independent variables is selected according to the MDL criterion. The MDL criterion is presented in Equation (3.26). In each of the 28 cases the number of the bootstrap replications of the regression coefficients is $B = 1000$. The application of the bootstrap in the multiple linear regression is described in the Section 3.3.1. The relative weights of the regression coefficients are computed from the bootstrap replications according to Equation (3.33).

The belief graph G_b is constructed from the relative weights. The adjacency matrix of the belief graph G_b is presented in Figure (4.1). There is one node for each variable so there are 28 nodes in total. Arcs between the nodes are obtained from the relative weights. The arcs represent the relative strength of the dependency between two variables. The darker colors correspond to stronger dependency between variables in the Figure (4.1). The relative weights for the model where jth variable has been the dependent variable are in the jth column.

The number of the dependencies can be reduced by setting a threshold λ. The dependencies whose relative weight are under the threshold λ can be rejected. In this case, the threshold is selected to be $\lambda = 0.06$. The threshold can also be seen as a significance level. In this case, the relative weights whose value are above the threshold $\lambda = 0.06$ are set to unity. Those dependencies are seen after that equally important. The belief graph is called a directed

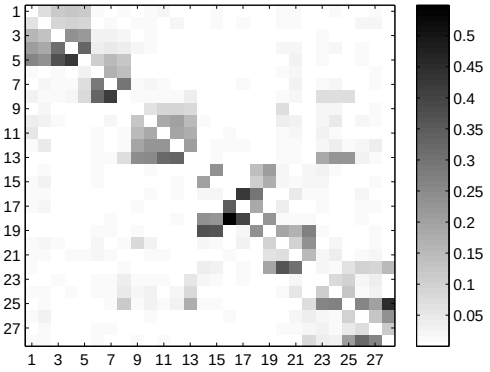

Figure 4.1: Belief graph G_b

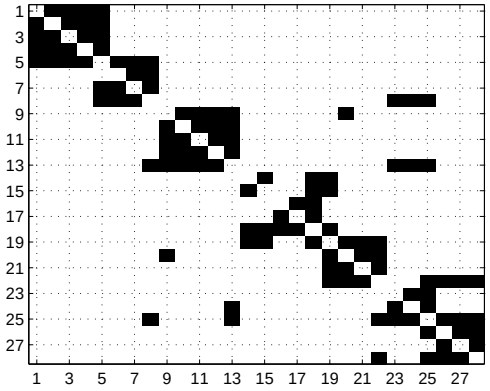

Figure 4.2: Directed graph G_d

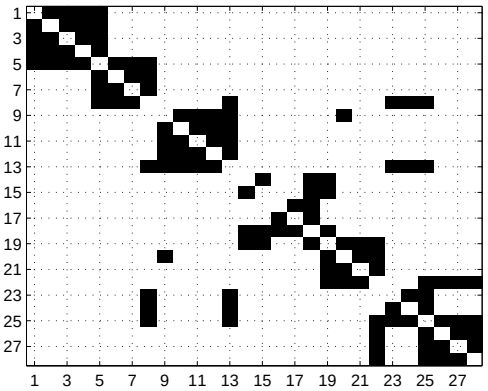

Figure 4.3: Undirected graph G_u

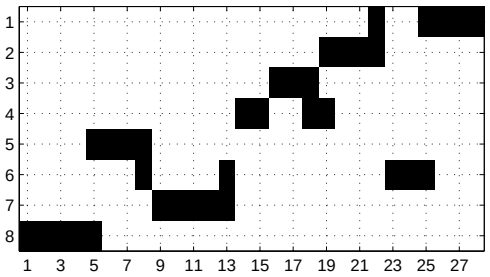

Figure 4.4: Cliques from G_u

graph G_d after these operations. The adjacency matrix of G_d is in Figure (4.2). The directions of the dependencies are also ignored for a while. When the directions are canceled the undirected graph G_u whose adjacency matrix is in Figure (4.3) is achieved.

The maximal cliques are sought from the graph G_u. The maximal cliques are found by using an algorithm which is presented in the Section 3.4.3. The variables which belong to the same maximal clique form a multiple linear regression model. The maximal cliques are presented in Figure (4.4). Each clique is on its own row in the Figure (4.4). The variables which belong to ith clique are marked by black on ith row. For example, the signals 22, 25, 26, 27 and 28 belong to the first clique. The largest R^2-value is achieved if

the variable 28 is the dependent variable and the rest of the variables are the independent variables. The directions of the dependencies is restored now. The change in the values of the independent variables causes a change in the dependent variable. Thus variable 28 is dependent on variables 22, 25 26, 27. This dependency was presented in the Figure (A.1) by the arrows.

Eight maximal cliques were found in total from the graph G_u. The number of the found linear models equals to the actual number of the linear regression models. When the Figures (A.1) and (4.4) are compared to each other can be seen that the all regression models were found. The variable 22 belongs also to the second clique. Thus the cliques 1 and 2 are dependent on each other through the variable 22. The equivalent relationship can be seen from the original dependency structure (A.1). The other relationships between the variables are also managed to find exactly.

In this case, the construction of the moral graph G_m from the undirected graph G_u was not necessary. If the threshold λ had been larger, the moralizing operation might have been necessary. The relative weight of the some dependencies might have been under the threshold then. The undirected graph G_u is almost always different with the different λ. In this case, some cliques might have had additional variables if λ had been smaller.

4.2 System data

The first real-world data set which is used in this study is called the System data. The system data consist of nine measurements from a single computer which is connected to the network. The computer is used, for example, to edit programs or publications and to calculate computationally intensive tasks.

Four of the measurements or the variables describe the network traffic. Rest of the variables are measurements from the central processing unit (CPU). All variables are in relative measures in the data set. The variables are `blks/s` (read blocks per second (network)), `wblks/s` (written blocks per second (network)), `usr` (time spent in user processes (CPU)), `sys` (time spent in system processes (CPU)), `intr` (time spent handling interrupts (CPU)), `wio` (CPU was idle while waiting for I/O (CPU)), `idle` (CPU was idle and not waiting for anything (CPU)), `ipkts` (the number of input packets (network)) and `opkts` (the number of output packets (network)). The variables are also listed in Table A.1 in Appendix A. The variables which describe the network traffic are marked by (network) and the variables which consist of the measurements

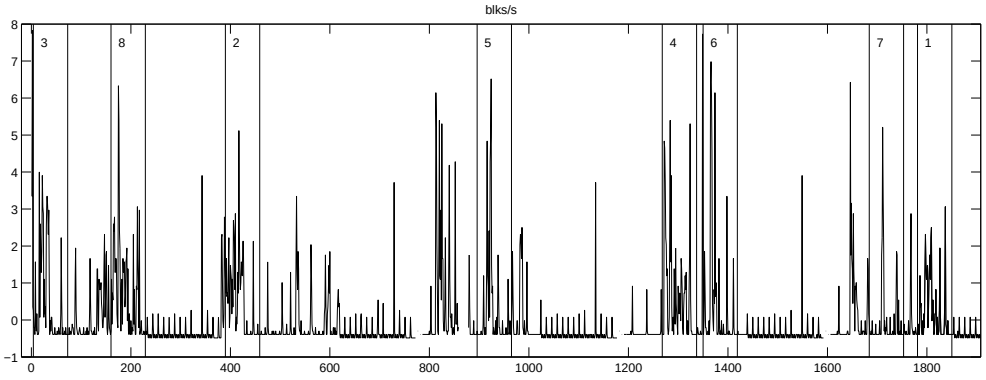

Figure 4.5: The reference signal `blks/s` and the selected windows.

from the central processing unit are marked by (CPU).

The System data is also used in the articles [36] and [1]. The first article presents how cluster structures and contents of the clusters can be described. In the second article, the System data are used as a basic example how high-dimensional data can be analyzed and visualized by the Self-Organizing Map. The description of the System data set is found from the article [36].

The System data is collected during a one week of computer operation. The first measurement is done at 10.02 am on Monday and the last one is done at 11.55 pm on Friday. The measurements are done every two minutes during the day and every five minutes during the night. The measurements are done from every nine variable each time. Some measurements have not been able to accomplished every time. There are missing values in all nine variables.

The selection of the time windows is used in this case. The procedure is presented in the Section 2.1. The infinite cost can be set for the missing values in the Equation (2.1), so any missing values are not in the selected windows.

In this case, the variable `blks/s` (read blocks per second) is selected to the reference signal. A plot of the reference signal is in Figure (4.5). The selected windows are marked by vertical lines. The window number one is the query window. The measurements in the query window are done in the afternoon on Friday. The windows $2 - 8$ are the chosen candidate windows. Smaller numbers of candidate windows refer to smaller values of the Equation (2.1). In the Section 2.2 presented noise reduction technique is applied to each window before the selected windows are put one after another. There are

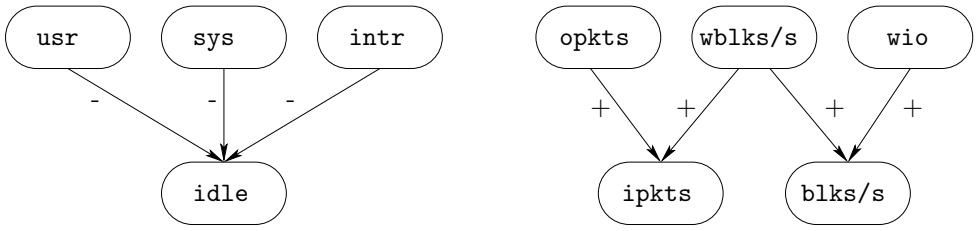

Figure 4.6: The dependency forest from the graph G_u

70 data points in each selected window so in the further operations there are $N = 560$ data points in total from every variable in use.

The next step is to find the best multiple linear regression models from the preprocessed data set. The process proceeds as it is described in Sections 3.4.1-3.4.3. The first task is to construct the belief graph G_b. The adjacency matrix of the belief graph G_b is presented in Figure (3.7). The number of the bootstrap replications was $B = 1000$ in each of the nine cases. Values of the bootstrap replications of the regression coefficients were estimated by LARS algorithm. The number of the independent variables in each bootstrap replication is obtained from the MDL criterion. The MDL criterion is presented in Equation (3.26). The relative weights of the parameters were calculated according to Equation (3.33). The directed graph G_d in Figure (3.8) is computed using the threshold $\lambda = 0.1$.

The final linear models are sought from the undirected graph G_u and from the moral graph G_m. The adjacency matrices of G_u and G_m are presented in Figures (3.11) and (3.12). The variables which belong to the same multiple linear regression model are part of the same maximal clique in the graphs G_u or G_m. The maximal cliques can be found by the algorithm which is presented in the Section 3.4.3.

The maximal cliques which were found from the graph G_u are plotted in Figure (3.13). The dependency forest of these cliques is in Figure (4.6). In this case, three maximal cliques $c_4^1 = \{\texttt{idle}, \texttt{usr}, \texttt{sys}, \texttt{intr}\}$, $c_3^1 = \{\texttt{ipkts}, \texttt{opkts}, \texttt{wblks/s}\}$ and $c_3^2 = \{\texttt{blks/s}, \texttt{wblks/s}, \texttt{wio}\}$ were found. The best R^2-values are achieved if the variables `idle`, `ipkts` and `blks/s` are chosen to be the dependent variables. The R^2-values are then 0.95, 0.94 and 0.82.

All variables in the clique c_4^1 are measurements from the CPU. All regression coefficients were negative in this model. If there is a positive change in some independent variable, the value of the dependent variable `idle` will decrease.

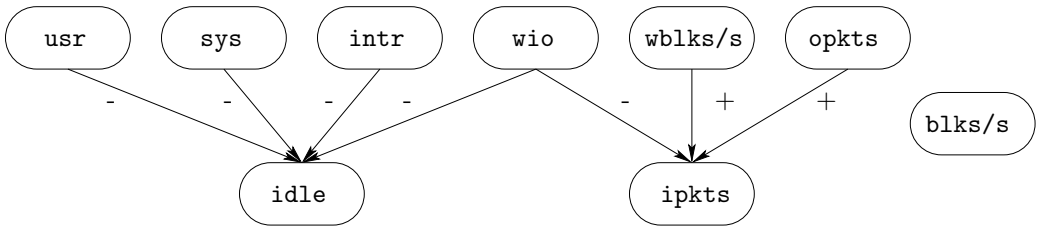

Figure 4.7: A dependency tree from the graph G_m

The cliques c_3^1 and c_3^2 are dependent on each other through the signal `wblks/s`, which is one of the independent variables in both cliques or models. When a positive change occurs in the variable `wblks/s` also the values of the dependent variables `ipkts` and `blks/s` increase. All variables in the clique c_3^1 are the measurements from the network traffic. In the clique c_3^2, the variable `blks/s` is the measurement from the network traffic and the variable `wio` is the measurement from the CPU.

Another dependency structure from the graph G_m is sought. The graph G_m was constructed from the graph G_u by adding the moral arcs. The moral arc addition is discussed in Section 3.4.2. The maximal cliques from the graph G_m are plotted in Figure (3.14) and the dependency tree of the variables is in Figure (4.7). There were two maximal cliques and the variable `blks/s` was left alone. The cliques are $c_5^1 = \{\texttt{idle}, \texttt{usr}, \texttt{sys}, \texttt{intr}, \texttt{wio}\}$ and $c_4^1 = \{\texttt{ipkts}, \texttt{wio}, \texttt{wblks/s}, \texttt{opkts}\}$.

All variables in the clique c_5^1 are measurements from the CPU. The best R^2-value is obtained, when `idle` is the dependent variable. Then R^2 value is 0.96, which indicate that the linear model describes the dependencies between the variables very well. The values of the regression coefficients are negative in the model c_5^1. The independent variables describe how much of the power of the CPU is spent to different activities and the dependent variable describes how much the CPU power is unused at the moment. According to the estimated linear model the value of `idle` decreases when the values of `usr`, `sys`, `intr` and `wio` increase. This is very intuitive result because the processes which need CPU time obviously diminish available power of the CPU.

The clique c_4^1 formulates another multiple linear regression model. When the variable `ipkts` is the dependent variable is achieved the best R^2-value. The R^2-value is 0.94. The variable `ipkts` consists of measurements from the

network traffic. The variables `wblks/s` and `opkts` describes also the network traffic and `wio` is the same measurement from the CPU as in the previous clique. The regression coefficients of the independent variables `wblks/s` and `opkts` are positive. This means that when the number of the written blocks per second and the number of the output packets increase the number of the input packets also increases. This is a natural situation in the bidirectional network traffic. The packets are sent to the both directions when, for example, a file is downloaded.

The cliques c_5^1 and c_4^1 are dependent on each other through the variable `wio`. Changes in `wio` has effect on both dependent variables `idle` and `ipkts`. The value of the regression coefficient of `wio` is also negative in the latter case. A positive change in `wio` decreases the value of `ipkts`.

4.3 Boston housing data

The second real world data set is called the Boston housing data. The data are got from the UCI repository of the databases [3]. The data set concerns housing values in suburbs of Boston in the USA. The data were collected in early 1970's. The data are not time series data i.e. the order of the samples is insignificant.

In the data set are 14 variables. The variables are listed in the Table (A.2) in Appendix A. The description of the data is found from the article [5]. There are 506 measurements from the each variable in the dataset. The variable `mv` represents the median value of owner-occupied homes in thousands of US dollars. The important feature of the variable `mv` is that the values larger or equal to 50 have been recorded as 50. This feature of the data is explained in the article [7]. The truncation of the range of the values of `mv` might put those extreme measurements in the unequal position i.e. the dependencies might be distorted in the upper part of the range. The measurements whose value is 50 in the variable `mv` and the corresponding measurements from other variables are excluded in this study. Sixteen measurements are excluded so there are $N = 490$ measurements from each variable left for the analysis.

A variable `chas` is a dummy variable. The value of `chas` is 1 if a tract bounds the river and 0 otherwise. Values of other variables are continuous. The Boston housing data are studied in many articles, for example in [5], [10], [7] and [9]. In the articles [10] and [9], the quantile regression is used to model dependencies. In the article [5], nonlinear transformations of inde-

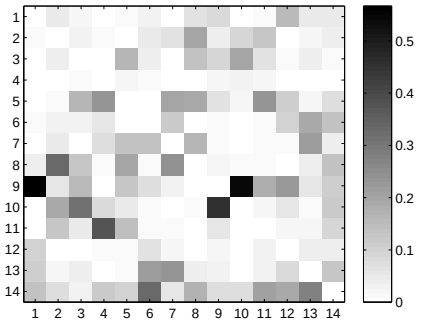

Figure 4.8: Belief graph G_b

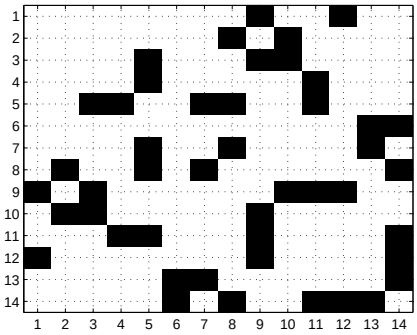

Figure 4.9: Undirected graph G_u

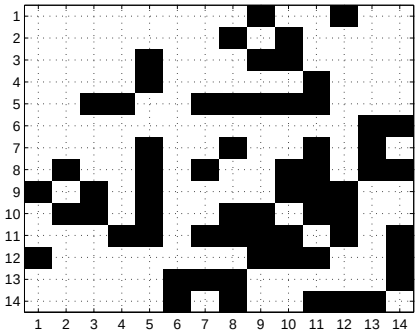

Figure 4.10: Moral graph G_m

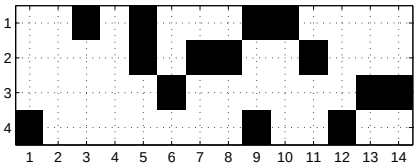

Figure 4.11: Cliques from G_m

pendent variables in regression model are studied. The explanatory power of the independent variables is studied in the article [10]. The dependent variable is selected beforehand in the all previous articles. The dependent variable have been the median value of owner-occupied homes `mv`. In this study, the dependent variable or variables are not selected beforehand. The most proper dependent variables are selected during the execution of the algorithm.

Each variable is selected to be the dependent variable in turn. The rest of the variables are the possible independent variables. The model selection algorithm LARS is used to select the most significant independent variables. In this case, 14 multiple linear regression models have to be estimated. The number of the bootstrap replications of the regression coefficients was $B = 1000$ in each 14 cases. The relative weights of the coefficients are computed according to Equation (3.33).

The belief graph G_b is constructed from the relative weights. The belief

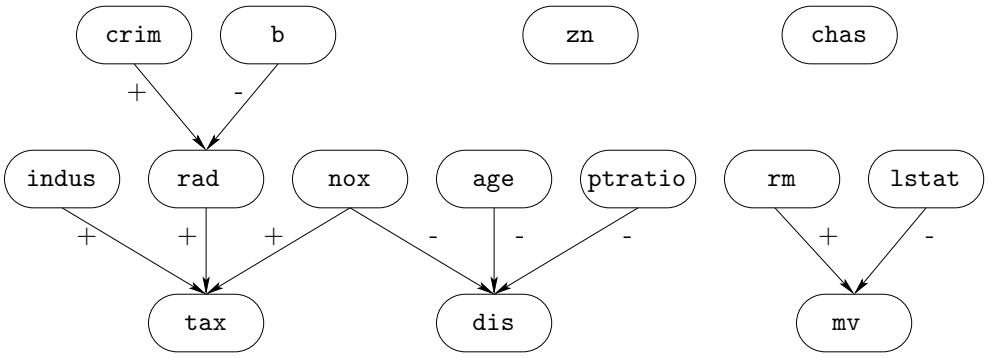

Figure 4.12: A dependency forest from the Boston housing data.

graph is in Figure (4.8). The darker the color in the Figure the more stronger the dependency is between the two variables. The dependencies whose relative weight are under 0.15 are ignored i.e. the threshold is $\lambda = 0.15$. The dependencies which are remained are treated equally important so the value of the relative weight is set to unity. The undirected graph G_u is obtained when the weakest dependencies and the directions of the dependencies are ignored. The adjacency matrix of the graph G_u is presented in Figure (4.9).

The moral graph G_m is constructed from the graph G_u as it is described in Section 3.4.2. The maximal cliques are sought from the graph G_m. The variables which belong to the same maximal clique form a multiple linear regression model. Four maximal cliques were found in this case. The maximal cliques are presented in Figure (4.11). The dependency structure of the variables is more clearly seen in Figure (4.12).

The cliques are $c_4^1 = \{\texttt{tax}, \texttt{indus}, \texttt{rad}, \texttt{nox}\}$, $c_4^2 = \{\texttt{dis}, \texttt{nox}, \texttt{age}, \texttt{ptratio}\}$, $c_3^1 = \{\texttt{mv}, \texttt{rm}, \texttt{lstat}\}$ and $c_3^2 = \{\texttt{rad}, \texttt{crim}, \texttt{b}\}$. The best multiple linear regression models according to the R^2-value are achieved if the variables `tax`, `dis`, `mv` and `rad` are the dependent variables. The R^2-values are 0.87, 0.66, 0.66 and 0.45 respectively.

In this dependency structure the variables `zn` and `chas` are independent on other variables. If there is a change in a value of `zn` or `chas` it has not effect on values of other variables.

The models c_4^1 and c_4^2 are dependent on each other through the variable `nox`. In the model c_4^1, the all regression coefficients are positive. This means that the value of `tax` increases when the value of the independent variables `indus`,

rad or nox increase. The all regression coefficients are instead negative in the model c_4^2. When the values of the independent variables nox, age or ptratio increase the value of the dependent variable dis decreases. The positive change in the variable nox causes the positive change in the variable tax and negative change in the variable dis.

The models c_4^1 and c_3^2 are also dependent on each other. The variable rad belongs to the both models. The regression coefficients of the variables crim and b are positive and negative, respectively. A change in the values of the variables crim and b cause a change in the value of the variable tax indirectly through the variable rad. The change in the variable tax occurs to the same direction as in the variable rad when the value of the independent variable crim or b changes. This is the case because the regression coefficient of rad is positive in the model c_4^1.

The model c_3^1 is independent on the other models. In the model c_3^1 the dependent variable is mv and the independent variables are rm and lstat. The value of the regression coefficient of rm is $b_{\texttt{rm}} = 4.2$ and lstat is $b_{\texttt{lstat}} = -0.60$, when the data are not scaled to zero mean and unit length. This indicates that when the average number of rooms per dwelling rm increases by one, the median value of owner-occupied homes mv increases by 4200 US dollars. When the value of the lstat increases by one unit, the mv decreases by 600 US dollars i.e. when the percent lower status of the population lstat increases the median value of owner-occupied homes decreases.

In the article [10], it is shown that the variables rm and lstat have the strongest explanatory power when the mv is selected to be the dependent variable. In that study the dependent variable was selected beforehand. In this study the corresponding relationship between the previous variables were identified although the dependent variable was not selected beforehand.

In the article [9], the quantile regression is used. In that model, mv is selected to be the dependent variable. In that analysis is four independent variables crim, rm, dis and lstat. The conclusion is also that rm and lstat are the most significant independent variables. In the article, it is also shown that rm is more significant in the upper range of values of the variable mv than in the lower range. Thus, the linear model is not necessarily the best model to describe the dependencies between the variables mv, rm and lstat, although it works adequately.

Chapter 5

Summary and conclusions

In this study, the method for analyzing linear dependencies in multivariate data is proposed. The result of the method is a linear dependency tree or a forest. The dependencies of the variables can be clearly seen from the final dependency structure. The linear dependencies are modeled by multiple linear regression models.

Two preprocessing techniques are presented in the beginning. Similar states of time series are selected by using the Euclidean distance between a reference signal. A single regression model is constructed to model that selected state. It might be difficult or even impossible to construct a single regression model to time series, which consists of many different states. Every state would require its own model. Secondly, the noise reduction technique based on the discrete wavelet transform is introduced. The real measurements are usually noisy and thus interesting features of the signals might be covered by noise. The variance of the noise was succeeded to halve in the case where the original noise was known.

The multiple linear regression models were constructed using sparse regression algorithm LARS. The bootstrap was also applied to the selection of the variables. The each variable was the dependent variable in turn and the most significant independent variables were sought. This study proposes how the relative weights of the regression coefficients can be calculated from the bootstrap replications. The relative weight of the regression coefficient is the measure of the belief that corresponding independent variable belongs to the certain regression model. In the experiments it was shown that the most significant variables have the highest relative weights. The relative weights seem to be appropriate to measure significance of the independent variables.

The final dependency structure was constructed from the belief graph. The belief graph represents the variables and the relative strength of the dependencies between the variables. In this this study, the threshold value was used to reduce the dependencies in the belief graph. The chosen threshold value has a strong impact on the final dependency structure. The minor change in the threshold value might cause the major changes in the final dependency structure. Thus, the special attention to the threshold value should be paid. The dependency graph or the moral graph is obtained using the threshold value and the moralizing operation on the belief graph. The maximal cliques in the dependency graph are interpreted to be the multiple linear regression models. The maximal cliques form the final dependency tree or the dependency forest.

The proposed method was tested using three different data sets. The first data set was synthetically generated whose actual dependency structure was known. Despite the rather complicated structure of the linear relationships the dependency structure was reconstructed perfectly using the proposed method. The rest of the data sets consist of real measurements. The data sets are called the System data and the Boston Housing data. The constructed dependency structures were convincing from both data sets. Over 90% of the variation of the dependent variable were succeeded to explain by the regression models which were constructed from the System data. In the Boston housing data case, 66% of the variation of the dependent variables are explained on average.

In addition to relative weights, the R^2-value and the MDL criterion is used to measure the goodness of the multiple linear regression models. In this study, whole data set are used when the dependencies are estimated so the reported results describe mainly the static dependencies between the variables or explain the structure of the data. The prediction accuracy of the models is not validated in this study.

5.1 Future work

Basically, the methods of the phases of the proposed algorithm are nearly independent on each other. Every method can be replaced by another method which is the most proper in a certain problem.

The selection of the interesting states can be improved. Now the selection is simple and straightforward. Similarities between the different parts of the time series can be measured more complicated way than using just the Euclidean distance. The noise reduction algorithm can be also improved for example by

soft-thresholding [11].

The goodness of the regression models can also be measured by the prediction accuracy. The data could be divided into training and validation sets. The division of data could be done for example using cross-validation [2]. The prediction accuracy analysis could be performed then. Then the models could be used to predict behavior in future and not merely to analysis of the seen behavior.

It would be beneficial to automate the selection of the threshold value λ. One possibility might be include it somehow in the moralizing operation. Another possibility could be to construct the final dependency structure from the belief graph and ignore the weakest dependencies such that a tree or a forest structure is achieved.

Although the linear dependencies describe the dependencies well in many cases some processes cannot be described by linear models at all. In the future, the linear model is meant to be replace by some nonlinear model. The purpose is to find such a nonlinear model that the relative weights of the regression coefficients can also be used in that case. Then the belief and the dependency graphs and the final dependency structure can be constructed as in this study. The ultimate goal is that the final dependency structure could consists of both linear and nonlinear models.

Appendix A

Figures and Tables

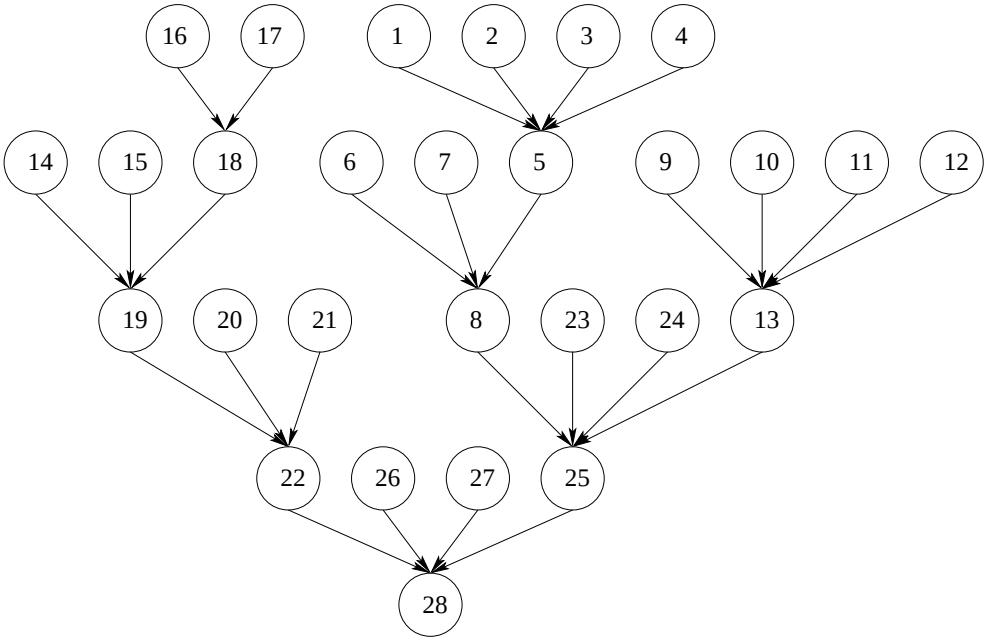

Figure A.1: A tree structure of the synthetic data

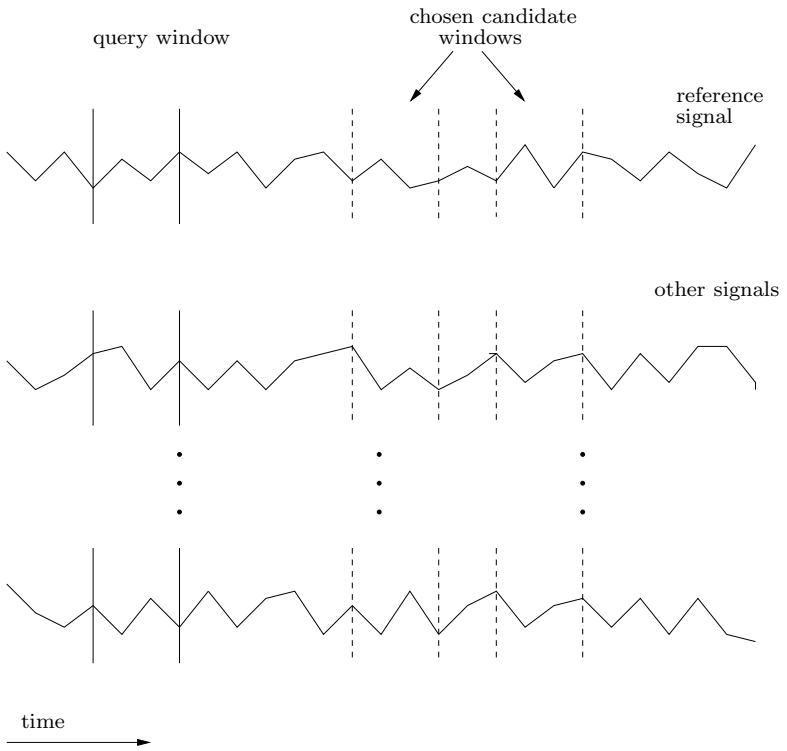

Figure A.2: The diagram of the selection of the windows. The data from the query and from the chosen candidate windows are selected for the further calculations.

Table A.1: The descriptions of the variables in the system data

No.	Name	Description
1	`blks/s`	read blocks per second (network)
2	`wblks/s`	written blocks per second (network)
3	`usr`	time spent in user processes (CPU)
4	`sys`	time spent in system processes (CPU)
5	`intr`	time spent handling interrupts (CPU)
6	`wio`	CPU was idle while waiting for I/O (CPU)
7	`idle`	CPU was idle and not waiting for anything (CPU)
8	`ipkts`	the number of input packets (network)
9	`opkts`	the number of output packets (network)

Table A.2: The descriptions of the variables in the Boston housing data

No.	Name	Description
1	`crim`	per capita crime rate by town
2	`zn`	proportion of residential land zoned for lots over 25,000 sq.ft.
3	`indus`	proportion of non-retail business acres per town
4	`chas`	Charles River dummy variable (= 1 if tract bounds river; 0 otherwise)
5	`nox`	nitric oxides concentration (parts per 10 million)
6	`rm`	average number of rooms per dwelling
7	`age`	proportion of owner-occupied units built prior to 1940
8	`dis`	weighted distances to five Boston employment centers
9	`rad`	index of accessibility to radial highways
10	`tax`	full-value property-tax rate per $10,000
11	`ptratio`	pupil-teacher ratio by town
12	`b`	$1000(B_k - 0.63)^2$ where B_k is the proportion of blacks by town
13	`lstat`	% lower status of the population
14	`mv`	Median value of owner-occupied homes in $1000's

Bibliography

[1] E. Alhoniemi, J. Hollmén, O. Simula, and J. Vesanto. Process monitoring and modeling using the self-organizing map. *Integrated Computer-Aided Engineering*, 6(1):3–14, 1999.

[2] C. M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press Inc., New York, 1995.

[3] C. L. Blake and C. J. Merz. UCI repository of machine learning databases. `http://www.ics.uci.edu/~mlearn/MLRepository.html`, 1998. University of California, Irvine, Dept. of Information and Computer Sciences.

[4] L. Breiman. Better subset regression using the nonnegative garrote. *Technometrics*, 37(4):373–384, November 1995.

[5] L. Breiman and J. H. Friedman. Estimating optimal transformations for multiple regression and correlation. *Journal of the American Statistical Association*, 80(391):580–598, September 1985.

[6] J. E. Cavanaugh. Unifying the derivations for the Akaike and corrected Akaike information criteria. *Statistics & Probability Letters*, 33:201–208, 1997.

[7] P. Chaudhuri, K. Doksum, and A. Samarov. On average derivative quantile regression. *The Annals of Statistics*, 25(2):715–744, April 1997.

[8] C. Chow and C. Liu. Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory*, 14(3):462–467, May 1968.

[9] J. G. De Gooijer and D. Zerom. On additive conditional quantiles with high-dimensional covariates. *Journal of the American Statistical Association, Theory and Methods*, 98(461):135–146, March 2003.

[10] K. Doksum and A. Samarov. Nonparametric estimation of global functionals and measure of the explanatory power of covariates in regression. *The Annals of Statistics*, 23(5):1443–1473, October 1995.

[11] D. L. Donoho. De-noising by soft-thresholding. *IEEE Transactions on Information Theory*, 41(3):613–627, May 1995.

[12] D. L. Donoho and I. M. Johnstone. Ideal spatial adaptation by wavelet shrinkage. *Biometrika*, 81(3):425–455, December 1994.

[13] B. Efron. Bootstrap methods: Another look at the Jackknife. *The Annals of Statistics*, 7(1):1–26, January 1979.

[14] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. Technical Report 220, Department of Statistics, Stanford, University, May 2002. The Annals of statistics 32, In press.

[15] B. Efron and R. J. Tibshirani. *An Introduction to the Bootstrap*. Chapman & Hall, 1993.

[16] J. F. Hair, R. E. Anderson, R. L. Tatham, and W. C. Black. *Multivariate Data Analysis*. Prentice Hall, 5 edition, 1995.

[17] D. Hand, H. Mannila, and P. Smyth. *Principles of Data Mining*. The MIT Press, 2001.

[18] M. H. Hansen and B. Yu. Model selection and the principle of minimum description length. *Journal of the American Statistical Association*, 96(454):746–774, June 2001.

[19] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer, 2001.

[20] D. Heckerman, D. M. Chickering, C. Meek, R. Rounthwaite, and C. Kadie. Dependency networks for inference, collaborative filtering, and data visualization. *Journal of Machine Learning Research*, 1:49–75, October 2000.

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

[22] C. Huang and A. Darwiche. Inference in belief networks: A procedural guide. *International Journal of Approximate Reasoning*, 15(3):225–263, 1996.

[23] A. Jensen and A. la Cour-Harbo. *Ripples in Mathematics The Discrete Wavelet Transform*. Springer, 2001.

[24] I. Koch. Enumerating all connected maximal common subgraphs in two graphs. *Theoretical Computer Science*, 250(1-2):1–30, January 2001.

[25] F. Kose, W. Weckwerth, T. Linke, and O. Fiehn. Visualizing plant metabolomic correlation networks using clique-metabolite matrices. *Bioinformatics*, 17(12):1198–1208, 2001.

[26] K. Lagus, E. Alhoniemi, and H. Valpola. Independent variable group analysis. In G. Dorffner, H. Bischof, and K. Hornik, editors, *Proceedings of the International Conference on Artificial Neural Networks*, number 2130 in Lecture Notes in Computer Science, pages 203–210. Springer, 2001.

[27] M. Lang, H. Guo, J. E. Odegard, C. S. Burrus, and R. O. Wells. Noise reduction using an undecimated discrete wavelet transform. *IEEE Signal Processing Letters*, 3(1):10–12, January 1996.

[28] C. L. Mallows. More comments on C_p. *Technometrics*, 37(4):362–372, November 1995.

[29] C. L. Mallows. Some comments on C_p. *Technometrics*, 42(1):87–94, February 2000.

[30] G. M. Maruyama. *Basics of Structural Equation Modeling*. SAGE Publications, Inc., 1997.

[31] R. G. Miller. The Jackknife - A Review. *Biometrika*, 61(1):1–15, April 1974.

[32] S. K. Mitra. *Digital Signal Processing A Computer-Based Approach*. McGraw-Hill, 2002.

[33] R. S. Pindyck and D. L. Rubinfeld. *Econometric Models and Economic Forecasts*. McGraw-Hill, Inc, 1991.

[34] D. N. Politis. Computer-intensive methods in statistical analysis. *IEEE Signal Processing Magazine*, 15(1):39–55, January 1998.

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

[36] J. Vesanto and J. Hollmén. An automated report generation tool for the data understanding phase. In A. Abraham, L. Jain, and B. J. van der Zwaag, editors, *Innovations in Intelligent Systems: Design, Management and Applications*, Studies in Fuzziness and Soft Computing. Springer (Physica) Verlag, 2003.

[37] J. Weston, O. Chapelle, A. Elisseeff, B. Schölkopf, and V. Vapnik. Kernel dependency estimation. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems*, volume 15, pages 873–880. MIT Press, 2003.

[38] A. M. Zoubir and B. Boashash. The bootstrap and its application in signal processing. *IEEE Signal Processing Magazine*, 15(1):56–76, January 1998.