NTNU
Norwegian University of
Science and Technology

A Framework for Constructing and Evaluating Probabilistic Forecasts of Electricity Prices
A Case Study of the Nord Pool Market

Kim Stenshorne

Master of Science in Physics and Mathematics
Submission date: June 2011
Supervisor: Ingelin Steinsland, MATH

Problem Description

For power producers good forecasts for the electricity prices are important for optimal scheduling. The methods currently used are deterministic models incorporating much system knowledge, but which do not include any quantification of uncertainty. To account for uncertainty, and based on a deterministic model, the candidate shall build a framework for constructing and evaluating a 10-days ahead probabilistic forecast of the electricity prices.

Assignment given: 18. January 2011
Supervisor: Ingelin Steinsland, MATH
Co-supervisor: Turid Follestad, SINTEF Energy Research

Preface

This report is the result of my master thesis in statistics, the final assignment of my four years study at the Norwegian University of Science and Technology (NTNU). The work presented is done in cooperation with SINTEF Energy Research during the spring of 2011. The thesis seeks to incorporate uncertainty into deterministic power prices of the Nord Pool market. It also explores the concept of evaluation, when the forecast provides a probability distribution. This exploration lead us to a greater consciousness in the use of the word *forecast*. At the same time new methods proposed in the literature regarding the evaluation of probability forecast are considered.

Several people have influenced this report in various ways and it is my pleasure to express my thanks to them. I would like to express my gratitude to Ingelin Steinsland, Department of Mathematical Sciences, who has been my advisor. I would also like to thank Turid Follestad, SINTEF Energy Research, my co-advisor during this project.

I am also grateful to Andreas Sylte, Statkraft, and Eirik Mo, Statkraft, who have shared their valuable insight into the industry of forecasting power prices and the Nord Pool market. To Gro Klæboe, Department of Electric Power Engineering, express my appreciation for introducing me to the concept of seemingly unrelated regression models and our fruitful discussions of the use of probabilistic evaluation criteria. Finally, I would thank SINTEF Energy Research which has been kind to provide me with office space.

Kim Stenshorne
Trondheim, June 2011

Abstract

A framework for a 10-day ahead probabilistic forecast based on a deterministic model is proposed. The framework is demonstrated on the system price of the Nord Pool electricity market. The framework consists of a two-component mixture model for the error terms (ET) generated by the deterministic model. The components assume the dynamics of "balanced" or "unbalanced" ET respectively. The label of the ET originates from a classification of prices according to their relative difference for consecutive hours.

The balanced ET are modeled by a seemingly unrelated model (SUR). For the unbalanced ET we only outline a model. The SUR generates a 240-dimensional Gaussian distribution for the balanced ET. The resulting probabilistic forecast is evaluated by four point-evaluation methods, the Talagrand diagram and the energy score.

The probabilistic forecast outperforms the deterministic model both by the standards of point and probabilistic evaluation. The evaluations were performed at four intervals in 2008 consisting of 20 days each. The Talagrand diagram diagnoses the forecasts as underdispersed and biased. The energy score finds the optimal length of training period and set of explanatory variables of the SUR model to change with time.

The proposed framework demonstrates the possibility of constructing a probabilistic forecast based on a deterministic model and that such forecasts can be evaluated in a probabilistic setting. This shows that the implementation and evaluation of probabilistic forecasts as a scenario generating tools in stochastic optimization are possible.

Contents

Chapter 1

Introduction

This work is motivated by the need for next generation short-term optimization methodology for hydropower producers. Previously the producers have used numerical optimization of the power prices in their production planning. The deterministic optimization procedure is now replaced by stochastic optimization. This change in methodology of the optimization has also an impact on the type of forecasts which are admissible. While the deterministic forecasts were capable to generate input for the deterministic optimization, the new method requires a forecast distribution which scenarios can be generated from(Fleten and Kristoffersen, 2008). The reason for this change is that the deterministic forecast, which only takes the expected values into account, fails to consider the uncertainty found in the probability distribution of the prices. This error in methodology might lead the optimization to find a non-optimal solution (Kall and Wallace, 2003; Higle, 2005). This stems from the fact that the previous methodology optimizes the function after the expected values of the parameters have been found, rather than the expected value of the function itself.

Today, the power producing companies employ several of these deterministic forecast, which in the industry is called fundamental models. These forecasts are made on different time scales, the longest of which are spanning several years. In this report we choose to study the short-term planning horizon, which only considers prices for the next ten days. The predictions are made for hourly prices, since this is the frequency which the prices are observed in the market. The market information structure is respected when predictions are made. This information structure means that new price data is only received once a day. Even this short forecast horizon of only 10 days produces a very complex system. This is because the forecast depends on variables that themselves are hard to predict. Examples of such variables are the temperature and consumption. In addition, the fundamental models must also take into account the structure of the power grid, since the electricity has to be transported to the consumer. This structure combined with the technical constraints of the supply lines have an effect on the power prices (Fu and Li, 2006). Further, the geographical division of the market has also an impact on the prices (Løland and Dimakos, 2010). In addition to fulfill these system based constraints, should the forecast of the short-term model be calibrated towards the forecast with the medium-time horizon.

The power producing companies have developed forecast algorithms which take the

complexity of the system into account. However, as the forecast is an essential part of the decision making process, they are closely guarded trade secrets. As we do not have access to any fundamental model, we have to construct one. To do so a detailed study of the power production facilities and other complex physical systems such as drainage basins had to have been made. In addition, a good insight into the power price market had to have been acquired. We do however not study these physical systems. The reason for this is that the motivation of this project is to make a framework by which any fundamental model may be used. Therefore, we chose a simple deterministic model which only depends on past prices, since the framework should be able to handle any deterministic (and suitable) model that are chosen. The deterministic model that we chose is throughout the project taken for granted and is not a topic of discussion. Thus, we are aware that the optimal fundamental model is not used, but rather one that is chosen by convenience.

In this report we present a statistical method of the error terms (differences) generated by the deterministic model and the observed prices. The prices are divided into two groups "balanced" and "unbalanced". This classification is done with regards to the relative size of the difference of prices for consecutive hours. The reason for classifying the power prices into two groups is the hypothesis that the error terms are generated by power prices which follow two different sets of dynamics. The prices in the balanced group are assumed to come from power prices that are effected by ordinary scenarios. By ordinary scenarios we mean that the prices are generated by a power system working under usual conditions and that none extraordinary external factors come into play. The prices which fall into the unbalanced category is assumed to be generated from scenarios that are unusual. These scenarios may originate from a break down of important infrastructure in the power system, which leads to shortage in supply. This shortage does in turn invoke prices that cannot be explained by the usual model. In addition would a single model that tries to capture both groups be dominated by the unbalanced prices. The error terms are also classified into two groups. The group that the error term is classified into is the same as its respective power price. Thus, if the price difference of time t and $t-1$ is classified as unbalanced, then the error term at time t is also classified as unbalanced.
A model both

A model which takes into account both groups of the error terms is proposed. However, this proposal does not in reality consist of a single model, since it is the finite mixture model that we propose. The mixture model consist of two components meant to represent a model for each of the classified groups. In addition to the components the mixtures consists of another feature. This final feature is a function which indicates the weight that the two error terms models should be given. The theoretical framework of the entire mixture model is described, but only for the component of the balanced error terms is a model proposed and results given. It is the seemingly unrelated regression (SUR) model which is used for the error terms that are labeled as "balanced". The SUR model was initially proposed by Zeller (1962) in his seminal paper and has since been applied on a wide variety of problems. These problems involve property prices, domestic incomes and crime rates among others (Wooldridge, 2010).

The SUR model is used to model a system of equations which through applying linear

regression on each of the equation. The noise term of the equations are assumed to be Gaussian distributed. Furthermore is the covariates of the equations assumed to be independent. This independence of covariates is both with respect to other covariates of the same equation and those of the other equations. However, the equations are assumed to be related though a correlation in the covariance matrix of the noise term. In this report the coefficients are fitted through a feasible generalized least square (FGLS) (Srivastava and Giles, 1987), which compromise of fitting the equations by the generalized least square. The feasibility comes from the fact that the first estimate of the covariance matrix of the FGLS is found by independently fitting the regression model of each equation by the use of ordinary least squares. The coefficients found by the SUR are used as parameters in the multivariate Gaussian distribution which is used to sample the price forecast.

In order to be able to evaluate the forecasts we introduce scoring functions. A scoring function evaluates the forecast with respect to the realization by giving it a value. This value should be as small as possible (or large as possible depending on the definition.) The scoring functions can be divided into two classes by the way they consider the probabilistic forecasts. This division is quite similar to that of the deterministic and stochastic optimization, which we saw at the start of this chapter. The first class of scoring functions, the class that we consider to contain the "classical" scoring functions, uses a single predicted value to evaluate the forecast. This single predicted value is deduced from the probabilistic forecast by a forecast function. We reproduce the results from the paper of Gneiting (2010) which shows that the variety of forecast functions that are used in the literature today. An example of one such forecast function and classical scoring function are the mean and the absolute error respectively. Many papers today seem to choose the forecast function at random (Cuaresma et al., 2004). However, we can optimize the choice of forecast function given the predicted distribution and scoring function(Gneiting, 2010).

The second class of scoring functions consider the entire distribution of the forecast without the application of the forecast function. This class of scoring functions is entitled probabilistic scoring functions. These scoring functions have in recent years gained momentum and have been used to evaluate a number of forecast scenarios (Berrocal et al., 2008; Gneiting et al., 2008; Panagiotelis and Smith, 2008). We introduce the continuous rank probability score (CRPS), which is a probabilistic scoring function for the uni-variate case. In addition to explaining the concepts of the probabilistic forecast we give a small tutorial of its use. Further, the energy score is introduced, which is the multivariate generalization of the CRPS. A second tool for evaluating a probabilistic forecast is the rank histogram. The method of the rank histogram is introduced and explained. The method measures the calibration of two probabilistic density functions with regard to each other (Gneiting et al., 2007). This calibration is found by evaluating to which degree the quantiles of the predicted and observed coincide.

This report is organized as follows. In Chapter 2 the Nord Pool market is presented. This presentation includes a brief historic summary of the years 2005-2009 of the prices and their pattern(s). In addition, we introduce our deterministic forecast, which is followed by an analysis of the error terms. In Chapter 3 we review the existing literature in the field of forecasting and modeling of power prices. In Chapter 4 the theoretical foundations of the

statistical regression, prediction and evaluation methods used in this report are presented. In Chapter 5 we present our own stochastic forecast model for the power prices. The focus is on the model for the unbalanced error terms, for which also the method and manner of parameter estimation are explained. In Chapter 6 we look at the result and performance of our own spot price model that was presented in the last chapter. This presentation is given through plots of predicted prices, use of scoring functions and rank histograms. In 7 we discuss the the results provided in the last chapter. The report closes with a discussion and a outline of further work in Chapter 8.

Chapter 2

Spot Price Characteristics

In this chapter the Nord Pool data, which we later are to use in a case study, is presented. This presentation includes the characteristics found in the electricity price such as patterns and spikes. In addition, the deterministic forecast model is revealed. After the presentation of the deterministic model a short analysis of the error terms follow. In the end we study the correlation structure of the prices and error terms. This study is later used when we is to determine which model to use.

2.1 Available Data

In this section we present the basic facts of the data used in this chapter. The main interest is the spot prices of electricity on the Nord Pool market. The market is divided into different areas, each with their own price. In addition to these a system price exists. The system price represents the ideal price [1] of the entire market. The division of the market areas changes over time. The prices for the different areas are also vulnerable to local effects caused by break downs in transmission lines and other technical infrastructure. Therefore, we choose to consider the system price in this report. It is however important to note that the data analysis could have been done on specific market areas. The final spot price model can be employed on the spot price of a single market area. The system price of Nord Pool, the prices of the market areas and other variables, can be found at the home page of Nord Pool (Nord Pool Spot AS, 2011). The other variables which are considered are the consumption of power in the market, the mean temperature[2] of Norway and the reservoir level. The data for the reservoir level are found at the home page of the Norwegian Water Resources and Energy Directorate (NVE , 2011). Months, weekdays and Norwegian holidays are also used as explanatory variables. The power prices of the market are quoted in euro (€) per megawatt-hour (MWh). We work with the prices from 2005

---

[1]The ideal price is the price that the market would have had if none bottlenecks or other technical constraints existed.

[2]The mean temperature is calculated as an average over the different locations, weighted by their respective population size.
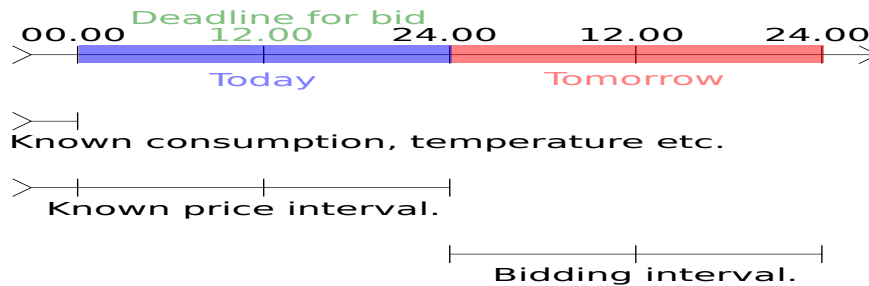
Figure 2.1.1: *Overview of available information at 12 pm yesterday. The upper time-line gives the important hours in which different information are revealed and/or updated. The second time-line gives the interval for which information about various variables, which may have an impact upon the electricity price, are known with certainty. The third time-line gives us the interval in which the electricity price is already settled. The last time-line gives us the time window for which prices are to be predicted.*

until the summer of 2010. The prices are given on hourly intervals giving us almost 50.000 data points[3]. The hours of the day will be denoted h, which takes the values $0, ..., 23$. Hour 0 is the time interval from 12 pm to 1 am and so forth.

The Nord Pool market is a day-ahead market. In such a market the price on a commodity for a given day is, as the name suggests, settled upon the day before. This is done in the following manner: Before noon the power producing companies (the sellers) send their price bids for hourly intervals the following operational day to the market operator. At the same time the power supplying companies (the buyers) send in their demands for hourly intervals for the next day. The market prices are then determined by the equilibrium between the supply and the demand. The market operator then reveals the equilibrium in form of prices and the amount of power to the respective producer. For the Nord Pool market the market prices are announced at 1 pm each day. This fact does that the power price for all hours today were already known from 1 pm yesterday, and that we do not have any more information about hour 0 than for hour 23. This information structure is depicted in 4.2.3. However, other parameters such as temperatures, consumption etc are only known up to and including the day before.

2.2 Pattern(s)

In everyday life the modern man uses a number of items powered by electricity. This creates patterns in electricity demand, which in turn create patterns in the power prices. As the existence of such patterns may influence the choice of model, we study the spot price history in order to find them. The structure of this pattern is found to be dominant on three time scales; daily, weekly and yearly. In Figure 2.2.1 (c) we have a plot of an usual weekday. The most prominent structures are the morning and afternoon peaks. The morning peak occurs at around 9-10 am. After the peak hour the power price decreases until around
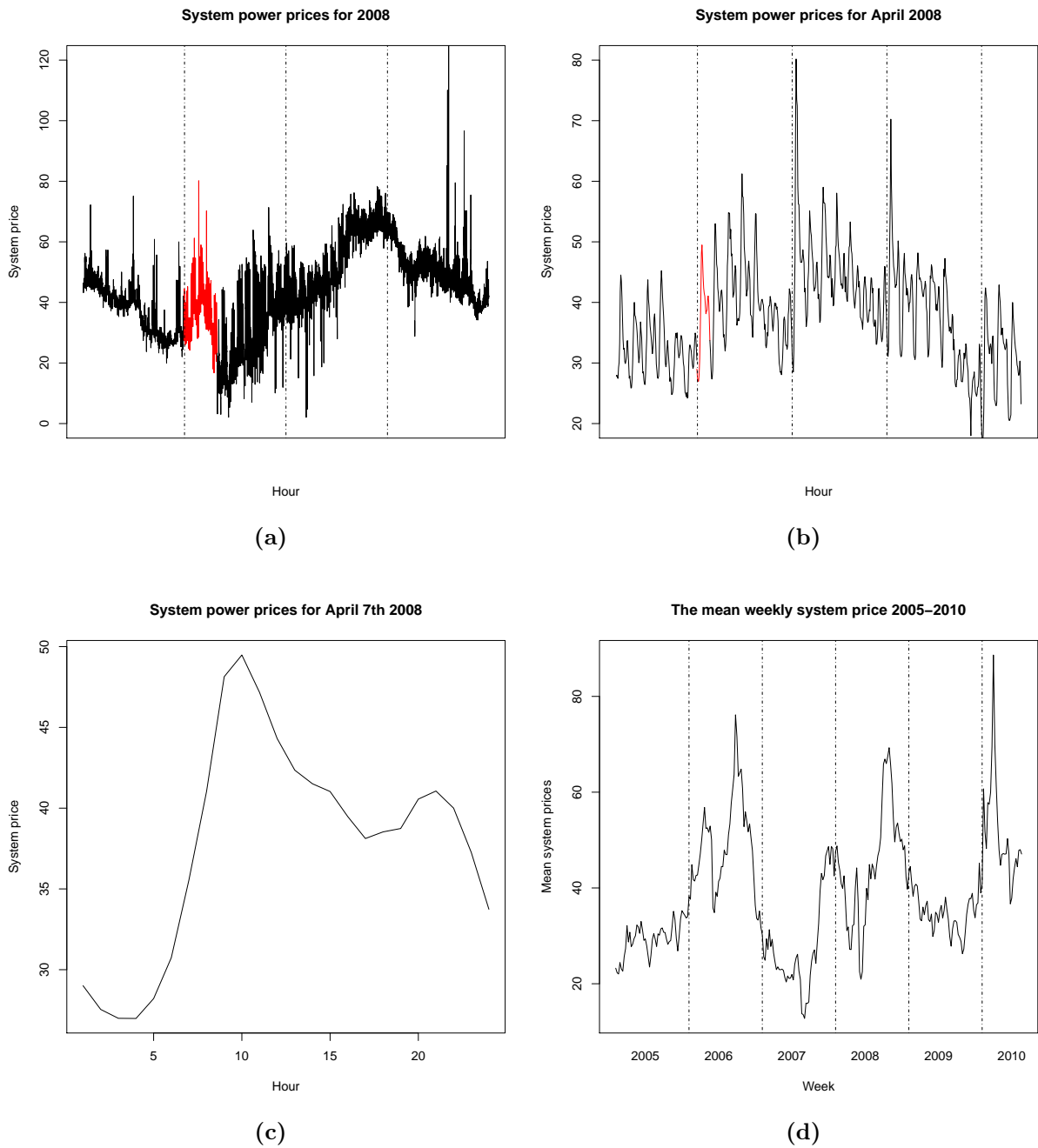
---

[3]However, as the site is updated daily, new data points are added each day.

3-4 pm when it again starts to rise and subsequently reaches the afternoon peak around 7 in the evening. Thereafter the price decreases and attain a minimum around 2 am, after which it once again starts to increase towards the morning peak. Another observation we make of the prices leading up to either of the two daily peaks is that they seem to be more volatile than the other day prices of the day. This is observed in Figure 2.2.2 (a) to (f) where the time series and box are plotted for the respective hours. In the figure the prices for the hours 7-9 are more ragged than for the hours 22-0, suggesting that they are more uncertain. Supporting this hypothesis is the box plot in Figure 2.2.2 where the whiskers are larger for the pre-peak hours than for the hours around midnight. This is especially true for the intervals from 7-9 am, where more outliers than for the hours 22-0 are seen. In Figure 2.2.1 (b) we can see the plot of the system price for April 2008. In addition to the observation of the two daily peaks we observe that the weekdays Monday through Friday seem to have a similar daily shape. For the weekends the maximum value of the morning spike tends to be much smaller. This is seen in the first week of April where the morning and afternoon spikes for the Saturday are of the same size. In addition to not having the same spike levels as the weekdays, the general price level of the weekends seems lower than for the weekdays.

An overview of the price for the entire year 2008 can be seen in Figure 2.2.1 (a). Here we can witness that the price decreases in the spring, picks up again during the late summer months and throughout the autumn. In the early winter months it starts to decrease once again. In the lower right panel of Figure 2.2.1 the averaged system price over weeks for the entire time period is plotted. The impression of a yearly cycle, given by the plot of the the prices of year 2008, seems to hold in general. Except for the winters of 2006/2007 and 2004/2005 (where we do not have data points before New Year.) The highest electricity price occurs sometime during the winter months. In addition, the price seems to drop during the spring whereupon it starts to increase again during the summer. Even thought the yearly fluctuation of the prices are considered, the level of the prices for the years differ. In Figure 2.3.1 the prices from the different years are plotted. From early May to late June and in December the price level seem to be the same. The level between these dates are however a different story. In late August we have a longer period where the prices differ with as much as 60€. Moreover, the price-level in this period of the year seem to alternate. In 2005, 2007 and 2009 the price level is "low", while it for 2006 and 2008 are "high". Even though this is too few years to draw any solid conclusion, this shows that one cannot expect the prices of a given year to be the same as those of the previous year. In addition, it clearly shows that no such thing as "ordinary" power prices for a given time of year exists.

2.3   Unbalanced prices

In this section an analysis is made of the extreme upward and downward movements in the hourly power prices, otherwise know as jumps. Jumps in the upward movement are to be called "spikes", while the downward movement is called a "dip". Together they will be

System power prices for 2008 (a)

System power prices for April 2008 (b)

System power prices for April 7th 2008 (c)

The mean weekly system price 2005–2010 (d)

Figure 2.2.1: *The Panels (a)-(c) are plots of the hourly system price over the same period, but with different time scales. Panel (a) is a plot of year 2008. The vertical dotted lines represent the different quarters of the year, while the red segment is the prices of April 2008. Panel (b) is a plot of April 2008. The Mondays in the month are specified with dotted vertical lines to ease their identification. The red segment is April 7th. Panel (c) is a plot of April 7th of 2008. Panel (d) is the weekly mean for the duration of the data period, that is from the start of 2005 until the summer of 2010. It thereby captures the fluctuation of the system price over the entire time period. The dotted vertical lines in this panel separate the different years. Note that the y-axes is constantly changing.*

8
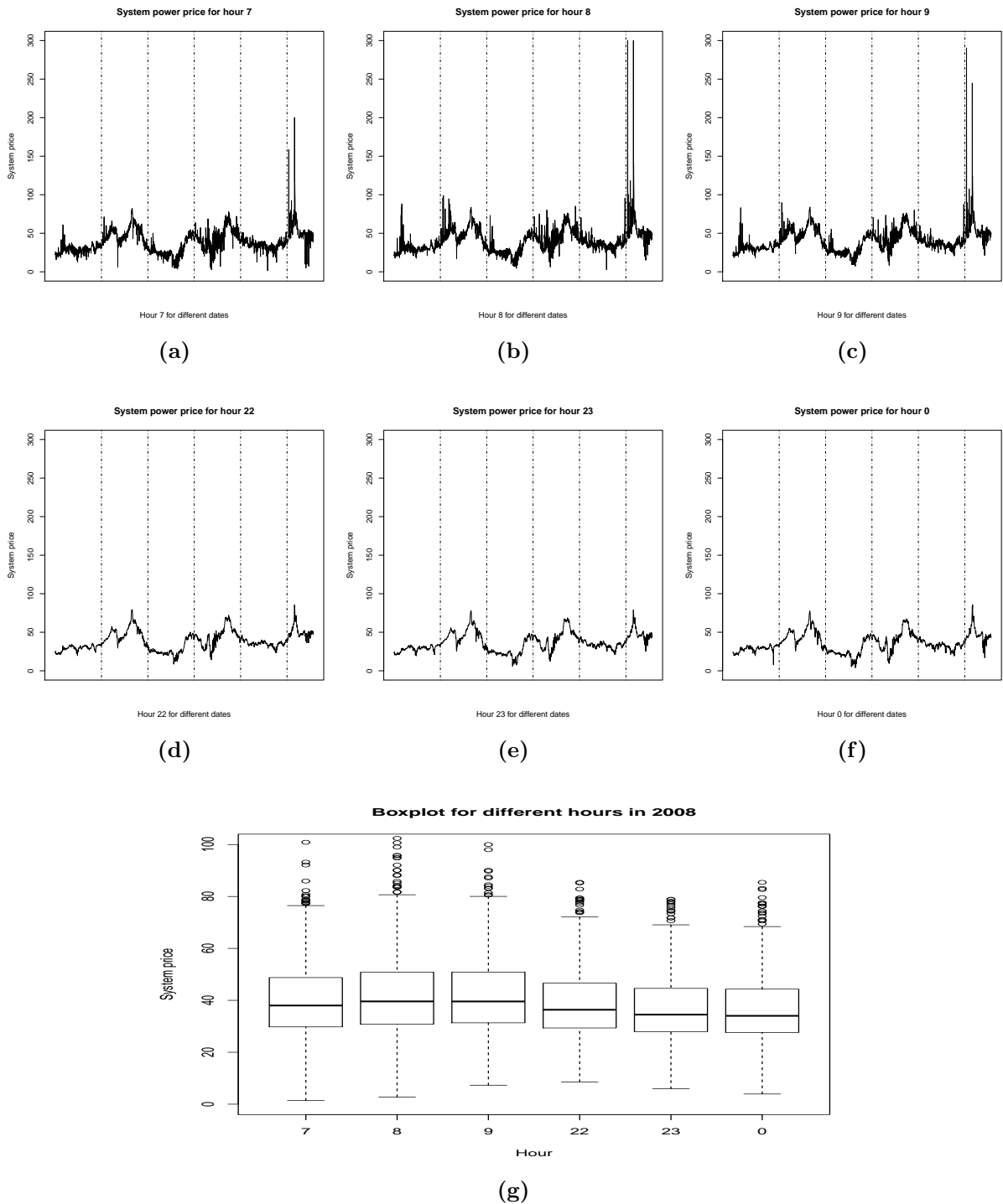
Figure 2.2.2: *The Panels (a)-(f) are time series plots of the system price for a single hour for the duration of the time period 2005-2010. Hour 7 is the interval from 7 am to 8 am etc. The dotted vertical lines mark the years. Panel (g) is a box plot of the same hours, but restricted to the year 2008. The prices plotted in the panels at the top show greater variance than the prices found in the figure in the panel of the middle row. This can also be observed in the box plot, where the whiskers of the hours 7-9 are larger than for the hours of 10-12. It should be noted that the y-axis of the time series plots have the same scale.*

9

referred to as "unbalanced" power prices, the complement of which being "balanced" power prices. Models that are proposed to capture the dynamics of power prices are usually unable to capture both unstable and stable prices at the same time. As we will see in Chapter 5, where a separate model for the unstable prices is considered, therefore an analysis of the phenomenon is done. In Figure 2.3.1 a plot of the hourly power prices for the years 2005 to 2009 is seen. In the figure we notice a set of characteristics concerning prices; (i) they have changing volatility, (ii) extreme jumps in the price occur and (iii) the upward jumps is larger in absolute value than the downward jumps. In Figure 2.3.1 we see that it is not always unforeseen spikes that causes the highest prices. The build up for the August and September peaks for 2006 and 2008 happen after a longer period of (relative) stable prices. By stable we mean that we observe (relatively) few cases of unbalanced prices.

To identify the unbalanced prices we follow a procedure from Clewlow and Strickland (2000). A spike (dip) can be defined as when the difference in price for successive hours are greater (lower) than the mean plus (minus) three times the standard deviation of differences for the same year. To identify the unbalanced prices we first take the differences and calculate their mean and standard error for the period in question. We first separate the differences into two classes: The "normal" differences (C1) and the "extreme" differences (C2). We then assign the difference into the class in which it belong. After this the new mean and standard deviation of C1 are calculated and the spikes found to be extreme are separated into the C2 class. This process is continued until no more C2 class differences are found from C1. When this is completed all the prices in C2 are unbalanced prices and the C1 class consists of only balanced prices.

In Table 2.3.1 the number of spikes and dips for the years 2005-2009 are found by the method stated in the last paragraph. The method calculates the number of spikes relative to the mean and standard deviation of the differences. As a result it cannot be used to compare directly the number of unbalanced prices between years. It might however give a fairly good representation of the fraction of yearly unbalanced prices. The reason for why the number of unstable price cannot be compared directly is discussed in de Lande et al. (2002). The main reason is that the iterative process of "sorting out" C2-values is stopped too early as a result of low standard deviation in the price difference for the period. In addition, the paper raises concerns in the use of the mean value as "center point" of the sample. The reason for this concern lies in the fact that a few extreme unbalanced prices may dominate the C2-group, thereby classifying prices that should be in C2 fall into C1. Signs of this can be found by comparing Table 2.3.1 and Figure 2.3.2 for the year 2007. The table tells us that 2007 is the year with the fewest unbalanced prices. This is however not obvious by looking at Figure 2.3.1, which suggests that this should be 2005, because the time series plot of the spot prices for that year is very flat and have relatively few spikes. In addition, the large fluctuation in the level of jump sizes for the unbalanced prices, as given in Figure 2.3.2, suggests that a single group of spikes fools the model. The largest spike sizes are in February making this seems plausible, as a large group of spikes is found here in the Figure 2.3.1. However, as none of the other years shows sign of the same degeneracy, we will use the proposed model by Clewlow and Strickland (2000).

We also note from Table 2.3.1 that there are more spikes than dips. In Figure 2.3.2 a

Year	2005	2006	2007	2008	2009
# Spikes	387	370	312	350	299
# Dips	289	231	182	199	112
# Total	676	601	494	549	511
Total frequency	7.71	6.86	5.63	6.28	5.83

Table 2.3.1: *The number of spikes and dips for the years 2005 to 2009. The spike is a large upward jump in the price, while the dip is a downward jump. In the third row the total number of spikes and dips is found. At the bottom the (relative) frequency of the occurrences of spikes and dips are found. The frequency is given as spikes and dips per hundred hour. The overall trend is that both the number of spikes and dips are decreasing.*

plot of the number of spikes and dips for each hour of the day is reported. It is clear from these plots that the occurrence of spikes and dips within a day is not uniformly distributed. In Figure 2.3.2a we see that the majority of peaks happen in the morning hours from 4 to 7. In addition, there is a cluster of spikes in the hours 3-5 in the afternoon. For the dips, which are seen in Figure 2.3.2b, there is a completely different story. An increase in dips is seen in the morning hours, but less dominant than for the spikes. However, a consistently larger degree of dips does occur through the early evening hours to the early morning. A conclusion of Figure 2.3.2 is that the number of spikes and dips in a given year is not the same, and that they are not uniformly distributed through the day. In Figure 2.3.3 a 30-day moving average (MA) of the size of spikes and dips for each of the years 2005-2009 are plotted. The MA value were calculated by counting the number of spikes or dips which had occurred in the 30 past days, and dividing it by the number of days, i.e. 30. From the figures we note that there seem to be a lower jump size for the dips than for the spikes. The difference in jump size, together with the observation of Table 2.3.1 that more spikes than dips occur, suggests that the prices have a skewed distribution. In addition we notice that the spread in the MA for the extreme events seems not to be that great, with December 2008 being an exception.

In Figure 2.3.4 a plot of the hourly frequency of jumps and dips for a 30-day MA is seen. At first sight we notice that the frequency for both the spikes and dips seem to change throughout the year(s). A further study of Figure 2.3.4a reveals that there in 2007 were four period with increased spike frequency, which occurred in March, June, September and December respectively. In 2006 we also observe four periods with an increase in frequency. These periods are February, May/June, October and December. In 2008 we have a single such period stretching from the start of May to the en of July. In addition we observe a second period with increased spike frequency in December of 2008. In 2009 we have 3 periods with increased frequency in June, September and December. The trend of the intensity of the dips which are seen in Figure 2.3.4b seem to more or less follow that of the spikes. The intensity of dips are however smaller than that of the spikes as fewer dips than spikes occur. The increase in intensity for the late spring and early summer might be more or less expected. This increase comes from the fact that this is a time of year when the power producing companies tries to minimize their water reservoir level. The
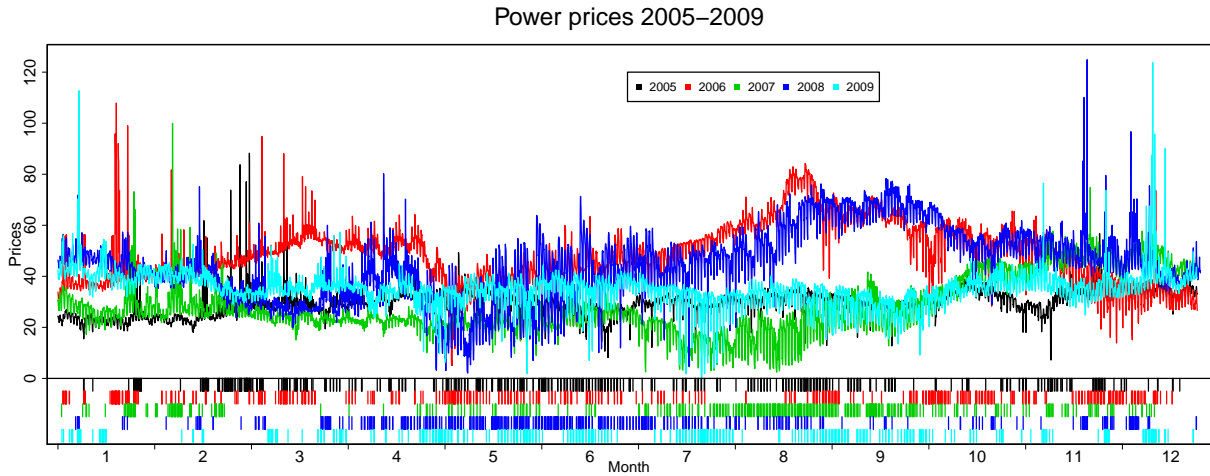
Figure 2.3.1: *A plot of the power prices for the years 2005-2009. The line segments at the bottom of the plot indicate spikes and dips. A single segment indicate either a spike or a dip. The color of the segment indicate the year. How the spikes and dips are found is described in 2.3.*

reason for doing this is that they prepare to fill up their reservoir with "new" water coming from snow melting and precipitation. Thus, a delay in warmer weather or precipitation may lead the level of the water reservoirs lower than expected which invoke uncertainty in form of spikes in the market. However, if a warmer weather and much precipitation comes in combination, this might lead to an increase in dips as the water cannot be stored. The increase in frequency in the early fall might be explained by the same dynamics as the spring. However, the increase in frequency during late fall and winter are harder to explain. In this period we might expect an increase in unbalanced prices in response to a cold period. However, as cold weather rarely set in for the whole Nordic market at the same time for longer periods, this cannot be used as the only explanation. Some of these increases might be the cause of system failure or other dynamics.

2.4 Our Deterministic Forecast

In this section we introduce our deterministic forecast. The deterministic forecast is, as clearly stated in Chapter 1, not a topic of interest in this report. This since it will be consider as in input to the framework that we are to present, namely the statistical model of the error terms. Thus, we choose to present the deterministic model here, even though the discussion could arguably have been done in a later chapter.

The power price is considered as the sum of a deterministic and the stochastic variable:

$$X_v(t,h) = D_v(t,h) + \epsilon_v(t,h), \tag{2.4.1}$$

where the variable $X_v(t,h)$ is the power price(s), the variable $D_v(t,h)$ is the deterministic forecast and the variable $\epsilon_v(t,h)$ the stochastic forecast. The subscript v denotes the last

The number of spikes in hours

The number of dips in hours

(a)

(b)
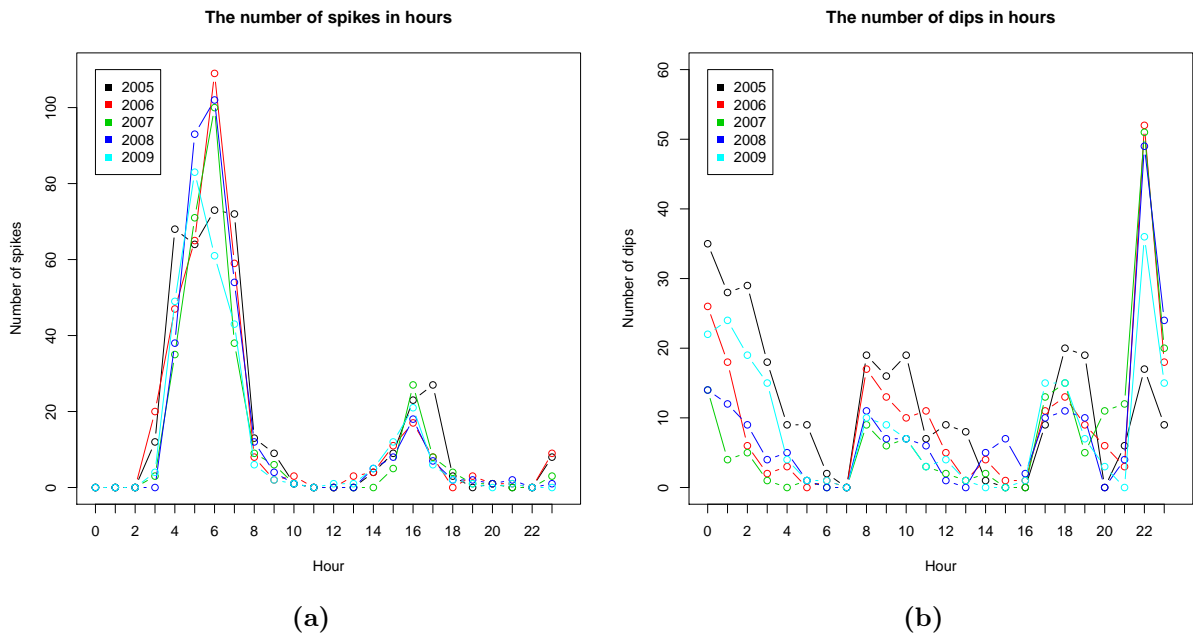
Figure 2.3.2: *Number of unbalanced prices. In Panel (a) the number of spikes for a given hour in the years 2005-2009 is plotted. In Panel (b) we see the respective plot of the dips in the same period. Note that the scales on the y-axis of the plots are not the same.*

Moving 30–days average of jumps

Moving 30–days average of dips

(a)

(b)
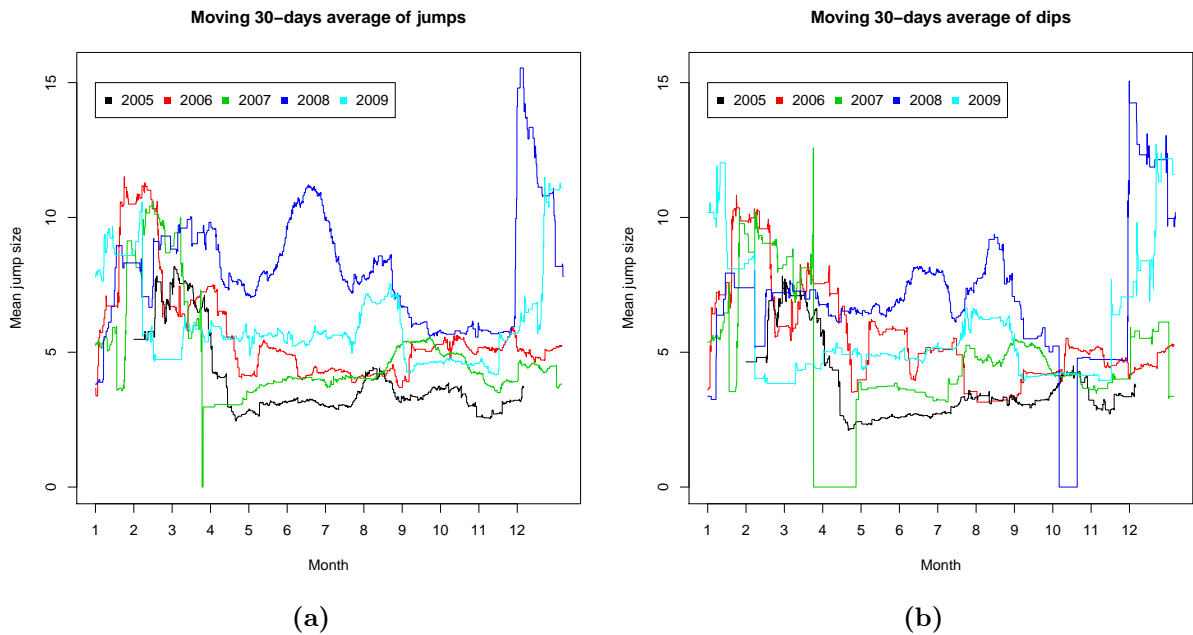
Figure 2.3.3: *The 30-days moving average of the size of a price jump for the years 2005-2009. The value at 1st of February is the sum of the spikes (dips) in January divided by the number of spikes (dips) that occurred. In Panel (a) a plot of the spikes is given, while in Panel (b) we have the dips.*
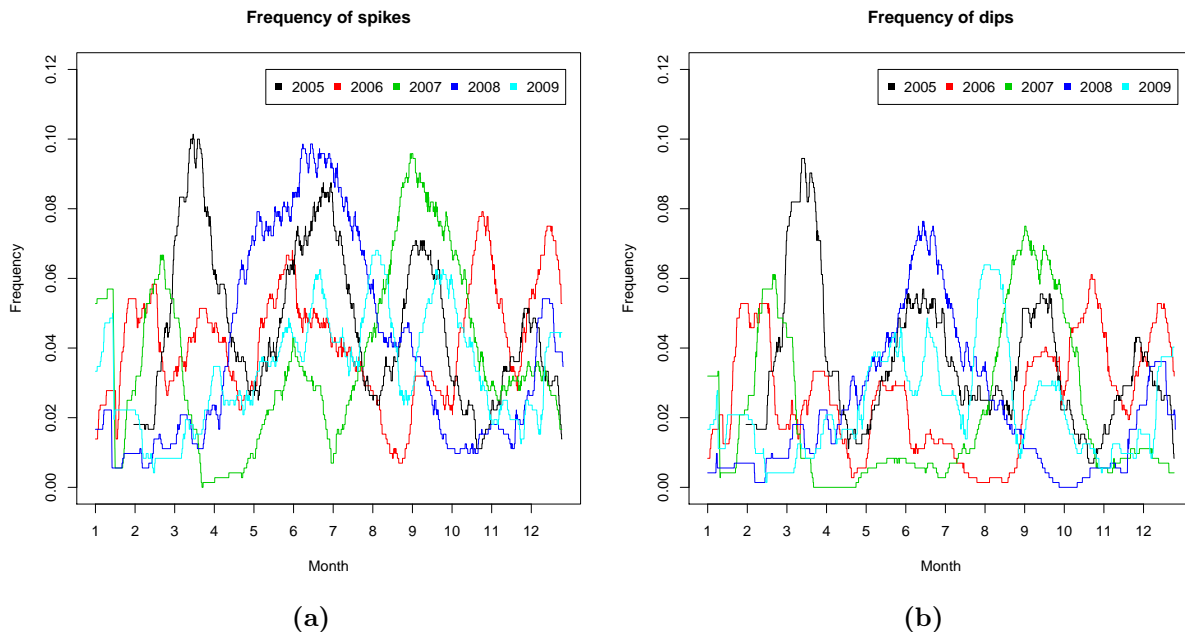
Figure 2.3.4: *The 30-days moving average of the intensity of jumps for the years 2005-2009. The intensity is given as the number of unbalanced prices per hour. In Panel (a) a plot of the spikes is given, while in Panel (b) we have the dips.*

day for which we known the power prices (and other data.) Thus, the parameter v will be referred to as today's date. The parameter t denotes a number of days relative to v, and will be referred to as the lead time of the forecast. The index h is the same as encountered at the end of Section 2.1. The deterministic model used in this report will be

$$\ln[D_v(t,h)] = \ln[X_v(-1,h)] + \operatorname*{median}_{i \in h}(\ln[X_v(-7,i)] - \ln[X_v(-8,i)]), \qquad (2.4.2)$$

where the ln denotes the natural logarithm. This model ensures that the forecast does not produce negative prices, which is unheard of in the Nord Pool market area[4]. This transformation does only apply to the deterministic forecast. Thus, the deterministic forecast consist of the logarithm of the power price yesterday plus the sum of the difference of the logarithm of the prices the same weekdays last week.

2.5   The Error Terms

In this section an analysis of the error terms introduced by the deterministic forecast of last section is given. In Figure 2.5.1b a plot of the deterministic one day-ahead forecast of the

---

[4]Negative power prices occur in the German market, where coal plants are in use. Coal plants cost sometimes more to shut down than to keep in production.

year 2008 can be seen together with the realized values of that year. The one day-ahead forecast in the figure does not depict the error terms of the other leading days, but it does however give a good insight into characteristics of the error terms. One observation that we make is that the deterministic prediction follows the observed prices quite well through the year. Another observation is that the deterministic forecast does not give any negative prediction of the prices as promised in Section 2.4. This is of course expected as the image of the logarithmic function do not contain negative values. At the same time it is still impressive that such a simple model as presented in Equation (2.4.2) manages to capture a sharp drop in the prices, as seen at the end of June, without dropping below zero.

The deterministic forecast presented in Equation (2.4.2) does however have some weaknesses. One of these weaknesses is that it induces some rather large errors after a large jump in the power prices have occurred. This is seen in Figure 2.5.1a, where a large negative error is observed at the end of October. This is right after a large spike in the prices, as seen from Figure 2.5.1b. Thus, the only explanation for this large negative spike in the model is that the deterministic forecast predicts a large spike on the basis of another which fails to be observed. Disregarding this flaw in the deterministic model we see from Figure 2.5.1a that it produces deseasonalized error terms. The error terms are however heteroskedastic. This can be observed from Figure 2.5.1a as a change in absolute value of the error terms throughout the year. For instance is the variance of the error terms greater from March to the end of July, than from the end of July to the end of October.

2.6   Correlation Structures

Before a model is fitted it is interesting to know if the patterns in the power price, which was discussed in Section 2.2, still exist after the deterministic model has been applied. In this section the correlation of the power prices and the error terms generated by the deterministic model is studied. As the information of power prices on the Nord Pool market is updated only once a day[5] any forecast has to be made for 24-hour blocks. Thus, the correlation of the power prices on daily intervals, both the one that takes place within and between days, are of interest.

In Figure 2.6.1a the correlation of the power prices among hours for the year 2008 is plotted. We can see that the intra-day dependence is strong, but very different among hours. From the plot there seems to be three "periods" of the day with extra strong correlation. The first period is for the early morning hours, another for the lunch-time hours and a third for the evening hours. These three groups have large within-group dependence. The dependence between the groups are however low. In Figure 2.6.1 the correlation matrix for the stochastic part of Equation 2.4.1 is shown. Therefore, before the correlation was calculated the deterministic forecast was subtracted from the observed prices. The figure indicates that the intra-group dependencies are sustained while the inter-group dependencies have decreased somewhat. However, as the range of the minimum value of the z-axis of the plot indicates a correlation of 0.7, the overall correlation is not low.

---

[5]The prices are published by the market operator at 1 pm every day.

The plot of $\varepsilon_v(1, h)$ for the year 2008

(a)

The plot of $D_v(1, h)$ and $X_v(1, h)$ for the year 2008
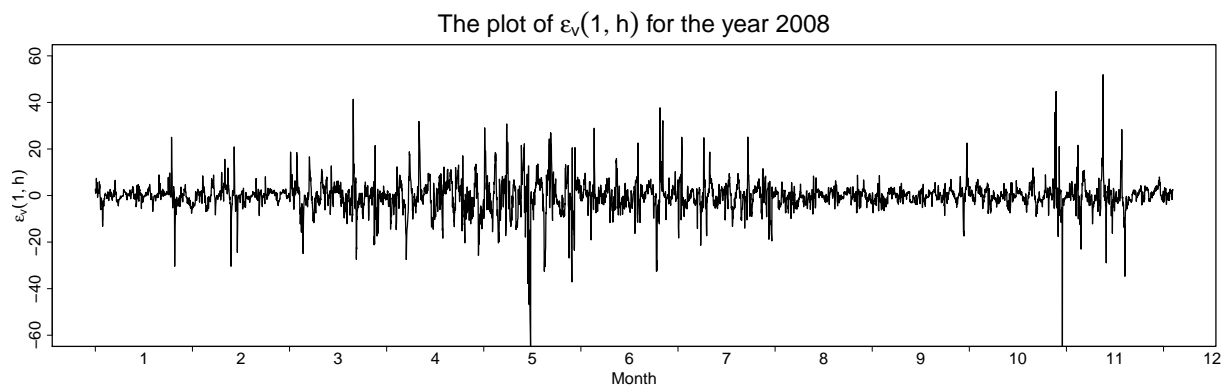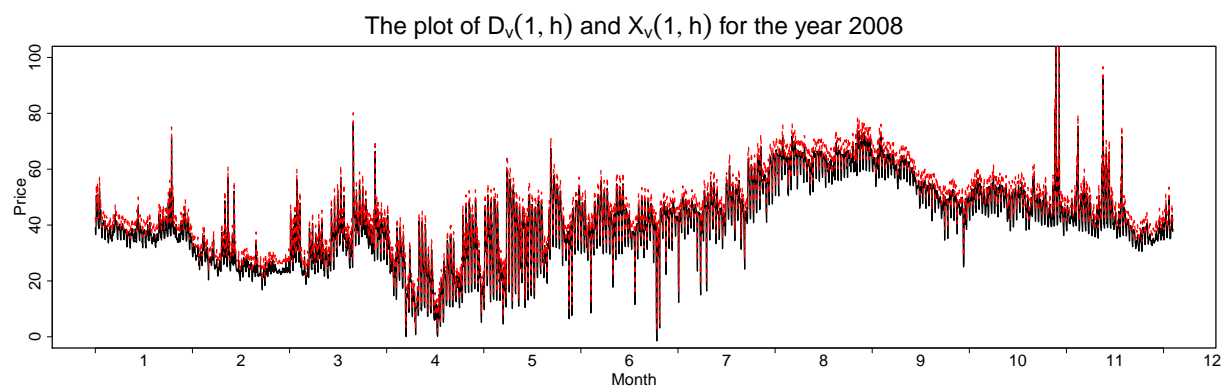
(b)

Figure 2.5.1: *(a) The plot of the error terms, $\epsilon_t(h)$, for the year 2008. (b) The plot of the deterministic one day-ahead forecast (red) and the true price data (black) for the year 2008.*
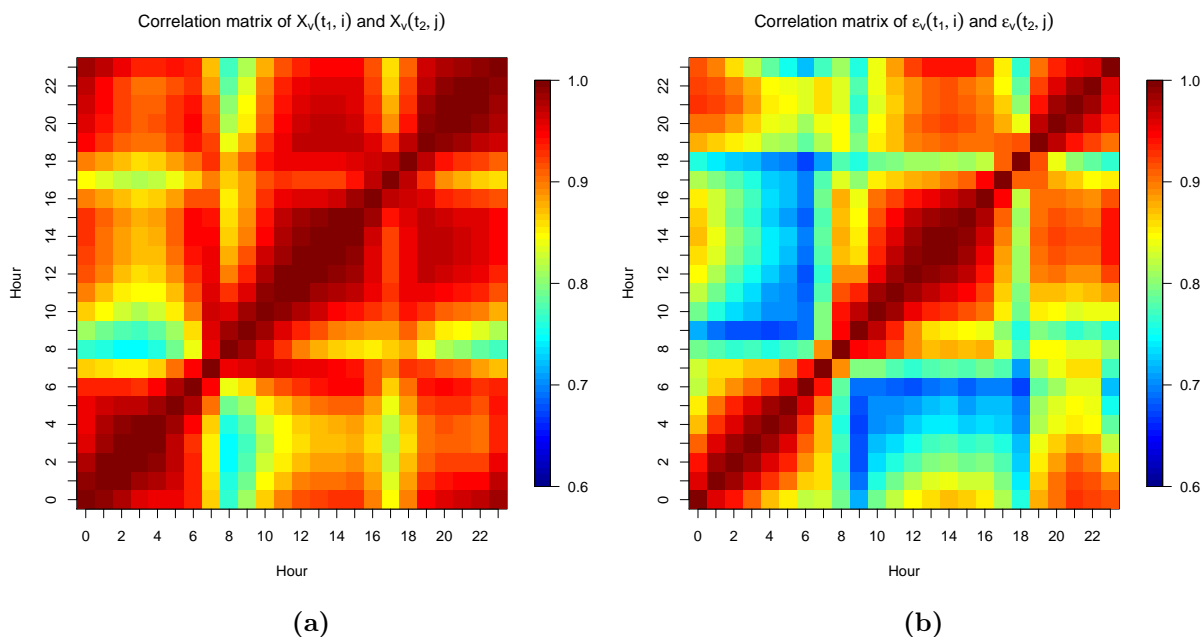
Figure 2.6.1: *The correlation matrices of Panel a) is the untreated price date for 2005-2008. In Panel b) is however the correlation matrix of the error terms, which originate from the difference of the deterministic model and the observed prices, as given in Equation (2.4.1).*

In Figure 2.6.2 the correlation plot for the power prices and the error terms are plotted for a ten lead-day interval. The motivation is to study the inter-day dependency. In Figure 2.6.2 we see the power prices, and for this plot the Figure 2.6.1a would alomst be the same as the diagonal matrix. This is because the diagonal of in Figure 2.6.2 is the within day correlation. In the plot we can observe that there seem, in addition to the within-day correlation to be correlation for the same hour for different lead days. Further, we notice that the correlation between a given hour and its neighbors seems to smooth out as the difference in lead time increases. This manifest itself through a more uniform correlation. We also observe that the correlation among the seven first lead days have more in common than that of lead day eight to ten. Moving to Figure 2.6.2b we see that the correlation displays a more significant pattern. The correlation does not discriminating among the lead days. In addition, we notice that the correlation structures between the inter-day hours is sustained. We also notice a smoothing of the error terms as the difference in lead time increases. This suggest that the error terms for become more similar with increasing lead times. This is plausible since the bias of the prediction should increase with increasing lead time. This means that while the error terms at small lead times are explained better with external variables, is the error terms at large lead times explained better by the systematic error made by the model.

17
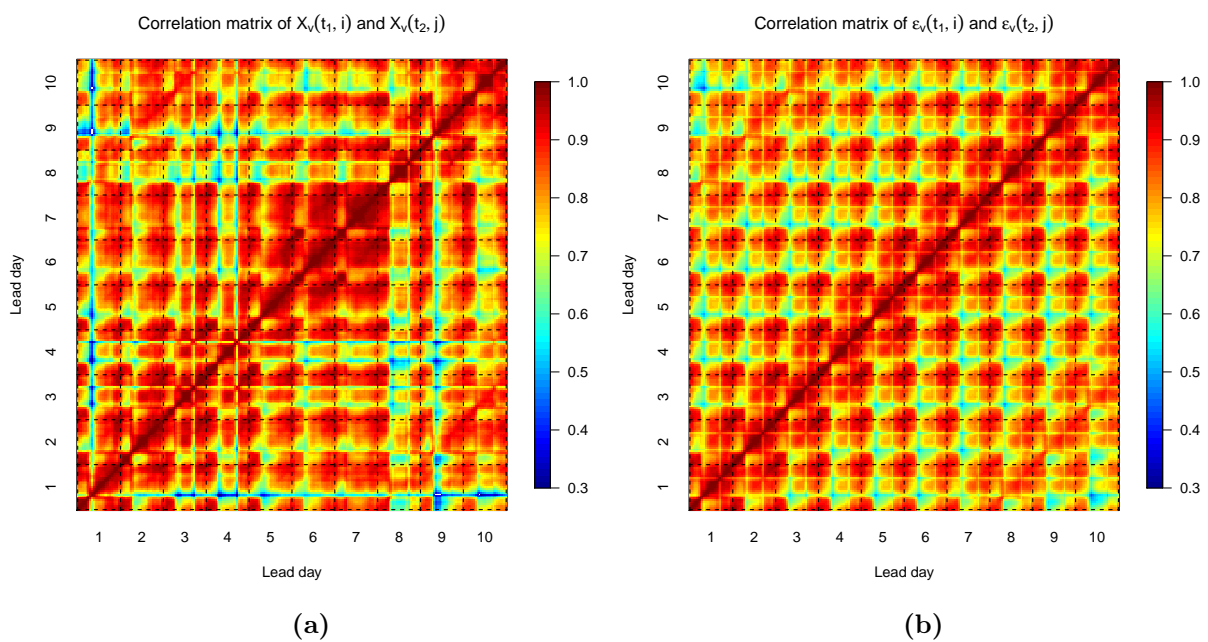
Figure 2.6.2: *The correlation matrices of Panel a) the untreated price date for 2005-2008 and Panel b) the error term of the power prices as given in Equation (2.4.1). The correlation matrices consider the lead time from 1 to 240 hours. The panels of Figure 2.6.1 can be seen as the mean diagonal matrix of each of the panels corresponding to the same quantity here.*

Chapter 3

Price Models in the Literature

In this chapter we perform a literature study in the field of forecasting and estimation methods of power prices. With this study we seek to explore the solution(s) and problem(s) found in competing strategies. The number of articles written on the subject of power prices is quite large (Gonzales et al., 2005). A number which have risen substantially since the liberalization of many of the power markets in the 1990's (Fleten and Kristoffersen, 2008). In addition, the articles in the literature display a wide variety of solution methods and approaches. This is caused by many factors: Firstly, the markets have different features, e.g. Worthington et al. (2005), which considers the different electricity markets found in Australia. Secondly the different structure of the market itself is a factor worth considering (Knivflå and Rud, 1995). Thirdly, exterior variables of the power system may be highly influential, as seen in McSharry et al. (2005), where the climatology changes the daily cycle of the prices. Fourth, the articles often try to capture a certain characteristic of the power price (Mullins et al., 2010), which there are many of, and thereby tailor the model to the question at hand. These models may be good at capturing certain dynamics, but by focusing on a (single) detail they may lose the whole picture altogether. Finally, many of the articles do only have the power price as a secondary objective (Lucia and Torro, 2005), or may only seek to explain (and thereby not forecast) the power prices (Løland and Dimakos, 2010).

The power price is a stochastic variable and as a result it changes with time. As a result an intuitive approach is the time series methodology. The simplest and thereby often the first approach when the time series methodology is considered is the uni-variate case. In such models the price is considered a function of single hour in the past or as a sequence of previous hours. In Contreras et al. (2003) the power prices of the Spanish and Californian market are modeled in sequence as an ARIMA series. In the ARIMA model the variance is the same for all points in time, which is something that is not necessarily true as seen in Chapter 2. Taking this into account Garcia et al. (2005) fits a GARCH model in addition to the ARIMA model to the same markets. The problems with these models are that they do not respect the information structure of the markets.

In Wu and Shahidehpour (2010) the authors expand the time series model by introducing an hybrid model by adding the possibilities granted by neural networks and wavelets

for extra flexibility. This paper does also work on the day-ahead prices forecasts. By this we mean that they do not consider the prices up to and including yesterday as known. This corresponds to the information structure in Figure 4.2.3. In the article returns of prices are modeled both in sequence and as separate hours. The approach of modeling prices in sequence consist of only applying a time series model to a single hour at the time. Thus, the paper disregard any dependence between different hours and consider the hours separately. In Section 2.2 we saw that the different hours of the day displayed their own characteristics. Indeed, better results are experienced with this model than with the time series approach alone. However, as the prices are considered as independent for different hours. This is disturbing, given the outcome of the analysis made in Section 2.6, as the correlation of the prices between hours in the same day are lost.

In Huisman et al. (2007) another attempt is made to capture the dynamics of the hourly data by its true information structure as we saw in Figure 4.2.3. However, the paper does this by combining the concepts of panel data and time series. A group of 24 hours, that is a day, is considered as a panel. These panels are in turn treated as time series. Thus the model captures the fact that the power prices have to be predicted for an entire day at the time, and that each such forecast depends on the same information set. That is, the information used to model 23rd hour of the forecast does not differ from the information available to forecast the 1st hour. However, the main advantage of the panel model approach is that it can also consider the structure within the panels. This is an improvement from Wu and Shahidehpour (2010), where each of the hours was considered to be independent. In Andersson and Lillestøl (2007) a further expansion of the fusion of panel and time series models are proposed. In this paper the authors propose a multivariate time series model is proposed. The paper also uses different price profiles for daily, monthly, and yearly power prices as a basis to model the panel of power prices. The panel approach is the same as that ofHuisman et al. (2007). However, by decomposition of the price structure the authors are able to analyze the impact on the power price for the different time scales as discussed in Section 2.2. Moreover the article study the volatility structure and also fit a GARCH process to the data. Also in the volatility structure is the paper able to connect the structure of the different time scales and that of the panel.

To deal with the occurrence of spikes in power prices, as mentioned in Section 2.3, an adaptive neural network is applied. In Nogales et al. (2002) time series featuring both power price and demand are used with dynamic regression and transfer function models. Another approach to tackle the system-shock induced by spikes is to introduce a mixture model. The mixture model is a popular model[1] that we will explore in more detail later. However, the main philosophy of the method is briefly stated to use many (separate) models as components. By using different models as components it is able to captured dynamics which (any) single models cannot capture. For the power prices the mixture model has previously been constructed with an Ornstein-Uhlenbeck process (Seifert and Uhrig-Homburg, 2006) and time series (Garcia-Martos et al., 2011; Cuaresma et al., 2004)

---

[1]The popularity of the model is shown in McLachlan and Peel (2000), where several names for the model are quoted. These names include mixed, component and switching model among others.

as components. However these papers do only make an one hour-ahead forecast, since they assume that the information is updated on an hourly basis. Consequentially they do not consider the information structure of the market. Additionally the goal of these papers is to find the model which best fits the data. As a result none of them seek to make a probabilistic forecast. The reason for a lack of interest in these matters can however be related to the fact that none of the papers stated is written for use in production planning.

The discrete approach of the time series models is not the only strategy found in the literature. In Skantze et al. (2000) a mean reverting processes is fitted to spot prices. A mean reverting process assumes that the prices have a mean level to which they will always return. Thus, the philosophy of a mean reverting process is to capture the rate and the manner of this return. This can be done by imposing structures on both the prices and its variance as seen in the paper. The mean revering methodology has been tried on a wide variety of other commodities, and as such it is supported by a large mathematical toolbox. However, the models are often hard to calibrate and the answers they provide are often difficult to interpret (Cartea and Figueroa, 2005). Continuous models are also suggested for power prices. One such suggestion is stochastic partial differential equations (SPDE) (Erlwein et al., 2010). The SPDE is a generalization of both the time series and the mean reverting processes in a continuous setting. The price bids on the Nord Pool market are however only placed for single hours, and as a result are discrete. Therefore we do not discuss the SPDE approach further.

Chapter 4

Statistical Models

This chapter explores statistical models already found in the literature, which later serve as the basis (or part of) the error term model. The chapter first turns its attention towards the regression of observations. Then the focus is shifted towards the prediction of intensities observation through point process. Thereafter we look at the mixture models which takes up both the regression of observation and the prediction of intensity of the observations. Finally, in the last part of the chapter we study how to evaluate predictions which are given both as point and probability distributions.

The following notation is used: The uppercase letters represent matrices. In addition, is the uppercase letters that are emphasized stochastic. The lowercase letters that are emphasized denote column vectors. The letters written in calligraphic style are sets. The letter $\mathcal{N}$ is the exception, since it is the Gaussian distribution. The Gaussian distribution is written $\mathcal{N}(\mu, \sigma^2)$, where μ and σ^2 are the mean and variance respectively. The superscript T denotes the transpose. Further, the letter I denotes the identity matrix, and a subscript on the identity matrix denotes its dimension. Subscripts are however only written on the identity matrix in cases where it is strictly necessarily. The symbol $\otimes$ marks the Kronecker product operator. The indicator function is written $\mathbb{I}(\cdot)$. The indicator function of a set $\mathcal{A} \subseteq \Omega$, is a function on the sample space Ω such that $\mathbb{I}_{\mathcal{A}}(x) = 1$ if $x \in \mathcal{A}$ and $\mathbb{I}_{\mathcal{A}}(x) = 0$ if $x \notin \mathcal{A}$. Thus, the indicator function indicates whether an outcome x is in the respective set of events $\mathcal{A}$ or not.

4.1   Regression and Prediction Methods

In this section we discuss different linear regression models. We start of considering a regression model for a single equation and expand this to include a system of regression equations. This enables us to first consider the equations as independent, while we in the end have a system of equations where there exists a correlation structure. This correlation between equation is captured by the covariance matrix. After we have introduced the models a longer discussion of their inference is given. When we are finished with multivariate regression and its inference we look at point processes in one dimension. Thus, we

are able to model an event which only happens at an instantaneous point in time. In the end we introduce mixture models. In this part a discussion fitting different components of the finite mixture model are highlighted, with special attention given to the component modeling of the balanced prices.

4.1.1 Seemingly Unrelated Regression

In regression a model of the relationship between the response variable y and the explanatory variable(s) x is considered. The relation can be written as $y = f(x) + \epsilon$, where the variable ϵ is the noise term. If the function f is assumed to be linear, we have then a linear regression model as expressed by

$$\mathbf{y}_i = X_i^{\mathrm{T}} \boldsymbol{\beta}_i + \boldsymbol{\epsilon}_i, \quad i \in \{1, ..., M\} \tag{4.1.1}$$

can be used to model the relationship. The dimension of the variables are respectively: The response variable $\mathbf{y}_i$ is $(T \times 1)$, the model matrix X is $(T \times K)$, the coefficients $\boldsymbol{\beta}$ is $(K \times 1)$ and the noise terms $\boldsymbol{\epsilon}$ is $(T \times 1)$. The parameter T is the number of observations of the response variables in the ith equation. The parameter M is the number of response variables. In the regression one assumes that each of the response variables is related to a linear combination of K explanatory variables. The weight of each explanatory variables is expressed through the regression coefficients $\boldsymbol{\beta}_i$. The relationship between the response variable and the linear coefficients have a disturbance term that is assumed to be $\mathcal{N}(0, \sigma_\epsilon^2)$ distributed. The parameter σ_ϵ^2 is the variance, while the mean is zero.

A multivariate regression model of M response variables each observed T times and K explanatory variables is expressed

$$\mathbf{y} = Z\boldsymbol{\beta} + \boldsymbol{\epsilon}, \tag{4.1.2}$$

where the response $\mathbf{y}$ is $(MT \times 1)$, the model matrix Z is a $(MT \times K)$, the coefficients $\boldsymbol{\beta}$ is $(K \times 1)$ and the noise terms ϵ is $(MT \times 1)$. The noise terms are $\mathcal{N}(0, \sigma_{ij}I_T)$ distributed. The parameter σ_{ij} is the variance of each of the different response variables as indicated by the index i. The index j on the other hand refer to correlation among equations. By defining $X_i = ZJ_i$, where J_i is a $(K \times K_i)$ we get a more general model, since the selection matrix picks out the explanatory variables of a restricted model. These restrictions, in the form of excluded explanatory variables, may be different for the equations. The restricted model is written

$$\mathbf{y}_i = ZJ_i\boldsymbol{\beta} + \boldsymbol{\epsilon}_i = X_i\boldsymbol{\beta} + \boldsymbol{\epsilon}_i. \tag{4.1.3}$$

This is a more flexible model compared to the one presented in Equation (4.1.2) where the same number of response variables were used. In Equation (4.1.3) we assumed that no correlation existed among the different explanatory variables of each response variable. If a correlation between the response variables in the multivariate case is suspected, then another model should be selected. An independent model assumes that the only terms in the variance matrix that is non-zero are the ones found on the diagonal.

In the rest of this section we talk about the inference for the three regression model

given in Equation (4.1.1), (4.1.2) and (4.1.3). The model in the first equation can be fitted by the ordinary least square (OLS) estimator. The estimate of the regression coefficients of this model is $\hat{\boldsymbol{\beta}}^{\text{ls}} = (Z^{\text{T}}Z)^{-1}Z^{\text{T}}y$. It should be noted that we fit a model where all the explanatory variables are present in every equation. For the model of Equation (4.1.2) we have to Take into consideration the new information represented by the possibility of correlation among response variables. If any of the response variables truly is correlated this is manifested through off-diagonal elements of the covariance matrix which differ from zero. This new assumption is written $\mathbb{E}(\epsilon^{\text{T}}\epsilon) = \Sigma \otimes I_T = \Psi$, where the old covariance matrix is $\Sigma \in \mathbb{R}^{M \times M}$. The dimension of Ψ, which is new covariance matrix, is $(MT \times MT)$. The least square estimation method for the new model is called the generalized least squares (GLS) estimator, and is written $\hat{\boldsymbol{\beta}}^{\text{GLS}} = (X^{\text{T}}\Psi^{-1}X)^{-1}X^{\text{T}}\Psi^{-1}\mathbf{y}$. The coefficient vector $\boldsymbol{\beta}^{\text{GLS}}$ has the dimension $(K' \times 1)$, the noise terms ϵ has the dimension $(MT \times 1)$ and the model matrix X has the dimension $(MT \times K')$. The parameter K' is less than K, since it represents a reduction in the number of explanatory variables. We do however not know the new correlation matrix. As a result we cannot estimate $\boldsymbol{\beta}^{\text{GLS}}$. Following the results of Zeller (1962) we replace the correlation matrix Σ with another matrix S. This replacement matrix has to be positive definite, symmetric, non-singular and of same dimension as the old covariance matrix. In other words: The replacement matrix has to be a feasible choice. This estimation method with the choice of covariance matrix just described is called the feasible least squares (FLS). The corresponding estimator is

$$\hat{\boldsymbol{\beta}}^{\text{FLS}} = [X^{\text{T}}(S^{-1} \otimes I_T)X]^{-1}X^{\text{T}}(S^{-1} \otimes I_T)\mathbf{y}. \qquad (4.1.4)$$

Two estimates of the S matrix are discussed in Chapter 2 of Srivastava and Giles (1987). The first estimate uses the covariance matrix of the residuals found by the full linear regression model, which were found by employing a OLS on each of the equations separately as we did in Equation (4.1.1). The residuals used to construct the covariance matrix are $\tilde{\epsilon} = \mathbf{y} - Z(Z^{\text{T}}Z)^{-1}\mathbf{y} = P_Z\mathbf{y}$, where the matrix $P_Z = I - Z(Z^{\text{T}}Z)^{-1}$. These residuals are the same as were found in the unrelated case of Equation (4.1.2). Thus, a consistent estimator for the covariance matrix S is $\tilde{S} = \frac{1}{T}\tilde{\epsilon}^{\text{T}}\tilde{\epsilon}$. By introducing this estimate into Equation (refeq:FLSest) we get the seemingly unrelated unrestricted estimator (SUUR), which then can be written

$$\hat{\boldsymbol{\beta}}^{\text{SU}} = [X^{\text{T}}(\tilde{S}^{-1} \otimes I_T)X]^{-1}X^{\text{T}}(\tilde{S}^{-1} \otimes I_T)\mathbf{y}. \qquad (4.1.5)$$

The reason why this estimate is called SUUR, is that all of the explanatory variables are used to estimate the feasible covariance matrix. Another approach is to considered the restricted version of the regression model of Equation (4.1.2). This restricted model was found in Equation (4.1.3), and the residuals of this model is $\hat{\epsilon} = \mathbf{y} - X(X^{\text{T}}X)^{-1}\mathbf{y} = P_X\mathbf{y}$, where the matrix $P_X = I - X(X^{\text{T}}X)^{-1}$. The same method used to find an unrestricted estimate is used again. Thus, a consistent estimate for the covariance matrix is $\hat{S} = \frac{1}{T}\hat{\epsilon}^{\text{T}}\hat{\epsilon}$

and the respective restricted estimator of Equation (4.1.4) is

$$\hat{\boldsymbol{\beta}}^{\text{SR}} = [X^{\text{T}}(\hat{S}^{-1} \otimes I_T)X]^{-1}X^{\text{T}}(\hat{S}^{-1} \otimes I_T)\mathbf{y}. \tag{4.1.6}$$

This is called the seemingly unrelated restricted regression (SURR), since the parameters can change among equations thereby giving a restricted parameter space. Thus, the restriction comes from the exclusion of explanatory variable(s) in the model.

We see that the approach of fitting the SUUR and SURR estimator follows more or less the same procedure. The only difference is the SUUR uses the full model, while the SURR have excluded one or more explanatory variables from equation(s). To recapitulate: We first fit a separate OLS model to each of the M equations. Second the residuals from this estimate are used as a basis to calculate a feasible covariance matrix. Finally this covariance matrix is used in the GLS estimation finding an estimate of the coefficients $\boldsymbol{\beta}$.

Two special cases reduces the SUUR and SURR models back to the ordinary linear regression, where the OLS estimate would be sufficient. The first of this cases is when all the explanatory variables are used in every equation, that is $X_1 = X_2 = ... = X_T = Z$ and $K_1 = K_2 = ... = K_T$. The model is then reduced to the multivariate model of Equation (4.1.2). The second case is when the estimated covariance matrix Ψ only has elements on the diagonal. The model is then equivalent to the uni-variate regression model of Equation 4.1.1. However, these two special cases are substantially different in the way they occur. The first case is the result of that model that was chosen, and we can therefore avoid such cases before any calculations are done. The second case on the other hand can sometimes only be identified only after the result of the calculation were acquired. This reason for this is that the preliminary exploration of the covariance structure may hint to a correlation structure. However, the existence of this structure may be rejected or found to be non-significant. Thus, leading the us to chose the model of Equation 4.1.1 as the preferred model. The advantages of the SUR model is its ability to identify any correlation that is found between the equations which cannot be explained systematically, something that enables it to capture a wide variety of problems.

Above we have assumed that all of the response variables y consist of an equal number of observation T. In real world application this might not be the case. In the paper of Schmidt (1977) an analysis of the SUR model, where the number of observations T_i for each response variable y_i differ is discussed. In the literature this is referred to as an unbalanced SUR models (Wooldridge, 2010). In the article four ways of tackling the unbalanced SUR model is discussed. Of these methods only the two cases is discussed in this report. The first option is to reduce the total number of observations $T = \sum_{\forall i} T_i$ to $T = MT_j$, where the variable $T_j = \min_{\forall i} T_i$. Thus, we reduce the number of observations of *all* the responses variables to the same number as the response variable with the fewest number of observations. This solution method does that we get back to the ordinary balanced SUR model. Another approach is to keep all of the observations. This results in a more accurate prediction, while the calculation and thereby the implementation gets harder.

4.1.2 Temporal Point Processes

The following section is inspired by Rasmussen (2011) and Daley and Vere-Jones (1988). A temporal point process describes a point pattern on the real line. This could be considered as a list of times when events take place. Thus, the terms point and event are used interchangeably. The term temporal describes the instantaneous nature of the point, which only exists at a single moment in time. In addition, the total number of points in the process is not known. Thus, given a time series Y_t, $t \in \{1, ..., T\}$, where an event happens at some time $j \in t$. Let $\mathcal{H}_j$ denote the history up to, but not including, the event. The history is defined as the training period of the process, that is $\mathcal{H}_j = \{Y_t\}_{t=1}^{t=(j-1)}$.

The simplest model of the stochastic process is the binomial model. This model considers the probability that an event occur at a single point in time to be p. In addition each point in time is considered as independent, making the process a Bernoulli trial. More advanced do however exist. The models can usually be classified into one of the two categories. The fist type model the stochastic process by the distribution of the time interval between events. The second type of model consider the distribution of the number of events in an arbitrary interval. The time interval between events are often denoted as the interevent time. A stationary interevent time is expressed as $f(t_1, ...t_j) = \prod_j (t_j | t_{j-1}...t_1) = \prod_j f(t_j)$, where f is the probability density function. A stationary process is a process which joint probability does not change in time.

An example of a stationary process is the homogenous Poisson process, where the intensity function $\lambda(t)$ is constant. This process is a special case of the renewal processes, where the interevent time is independent identically distributed, that is $f(t_j) = f(t_j - t_{j-1})$. In Figure 4.1.1 three renewal processes are seen. In the figure three Gamma-distributions with different parameters are plotted. We observe that the different parameters gives three (apparent) different point processes with regards to the clustering of the points.

A further generalization leads to the Wold process where $f(t_j) = f(t_j - t_{j-1} - t_{j-2})$ which is a first order Markov chain. This concept is further generalized into point processes where a larger fraction of the entire history is considered. The conditional intensity function is defined as $\lambda(t) = \frac{f(t)}{1-F(t)}$, where F is the cumulative distribution function of the distribution. A brief explanation for the previous definition of the conditional point process is

$$
\begin{aligned}
\lambda(t) &= \frac{f(t)}{1 - F(t)} \\
&= \frac{\Pr(\text{point is in dt} | \mathcal{H}_t)}{\Pr(\text{point not before t} | \mathcal{H}_t)} \\
&= \frac{\Pr(\text{point is in dt, point not before t} | \mathcal{H}_t)}{\Pr(\text{point not before t} | \mathcal{H}_t)} \\
&= \Pr(\text{point in dt , point not before t} | \mathcal{H}_t) \\
&= \Pr(\text{point in dt} | \mathcal{H}_t) \\
&= \mathbb{E}(N(dt) | \mathcal{H}_t),
\end{aligned}
$$
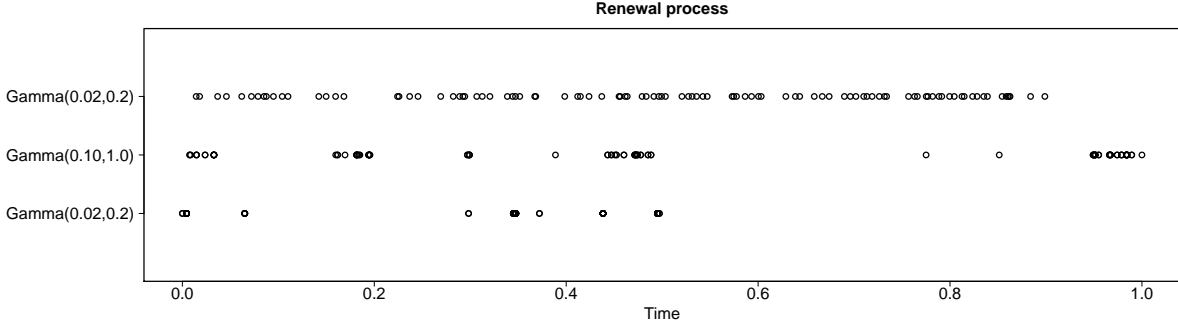
Figure 4.1.1: *A plot of three different renewal processes with different degree of clustering. The interevent time of the process displayed dare respectively Gamma(0.02,0.2), Gamma(0.1,1) and Gamma(2,20) distributed. Due not that the Gamma(0.1,1) distribution is the same as the Poisson process with intensity 0.1. From each of the processes 100 points are sampled. The three processes are however standardized such that the (global) minimum value is put at time zero and the (global)maximum value is put at time one.*

where Pr denotes the probability and $N(\mathrm{d}t)$ the number of points in the interval $\mathrm{d}t$. A specific point processes is the Hawkes process

$$\lambda(t) = \mu + \int_{-\infty}^{t} g(t-s)\mathrm{d}s(N) \qquad (4.1.7)$$

$$= \mu + \sum_{t_i \in \mathcal{H}_t} g(t - t_i)$$

where $\mu > 0$. The μ is the baseline intensity function throughout the year. The function g is the so called trigger function, which may be expressed on the general form $g(z) = \sum_{i=1}^{K} \alpha_i z^{i-1} \exp(-ic)$, where $z = t - t_i$. The parameters α_i decides the degree of clustering of points, c the damping of the intensity function after an event and K is the order of the trigger function. Thus for the simplest case when $K = 1$ the Hawkes process becomes $\lambda(t) = \mu + \alpha \sum_{t_i \in \mathcal{H}_t} \exp(-(t - t_i))$. In this case the expected level of the intensity is μ. However, when a new point occur the intensity function increases by α, after which it instantly starts to decreases exponentially towards μ again.

4.1.3 Mixture Models

In this section the framework of the mixture model is studied. The main idea of mixture models is to let the probability density function be a linear combination of other probability density functions. The weights of each of the linear terms have to sum to one, so the probability of an event being in the (entire) sample space is one. The advantage that the mixture models have when compared against other traditional models functions, is that they enable more exotic (probabilistic) functions. Some of these functions are seen in Figure 4.1.2. In the figure many of the functions seen are consists of several components. Indeed theory for finite models where the number of components in the model is not known exists (Fruhwirth-Schnatter, 2006)[Chapter 4]. However, we will focus on the cases where

Finite mixture densities
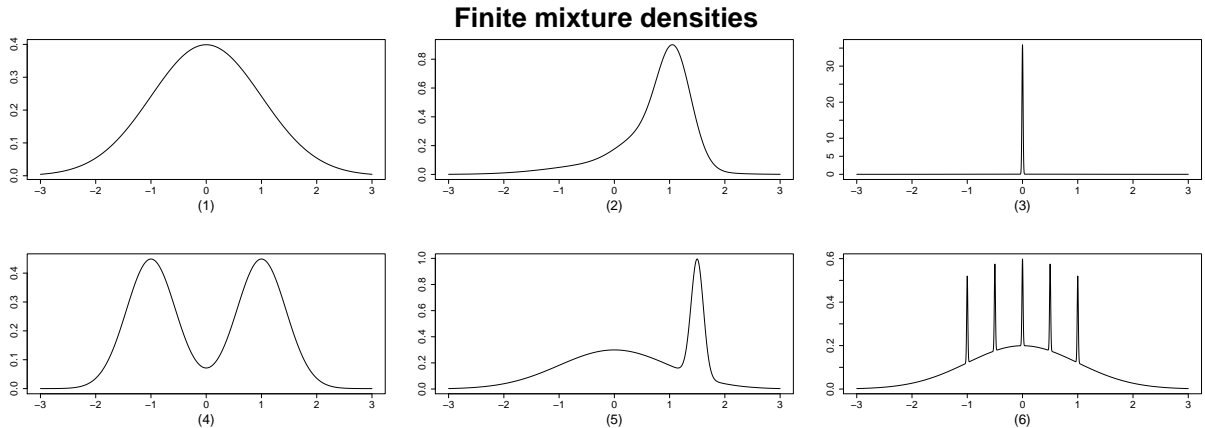
Figure 4.1.2: *Figure 1 of Marron and Wand (1992). In Panel 1-6 the probabilistic density of the standard Gaussian, skewed unimodal, outlier, bimodal, skewed binomal and claw are plotted. The closed form expression of the densities are found in Table 1 of the same paper from which the figure is taken. The Gaussian distribution is the only probability distribution used as component of the mixtures. The scale of the y-axes changes between panels.*

the number of components are assumed to be known and especially the cases where the number of components is assumed to be two.

The finite mixture model is used when the events in the sample space consist of sub-populations. These sub-populations may be group of events that share a expectation and variance which is different from other sub-populations. Thereby, one can assume the stochastic variable Y is composed of several probability distributions. These probability distributions are expressed as the stochastic variable conditioned on which of the sub-populations it originated. If S is the indicator of the sub-population from which Y originates the distribution which may be written $\Pr(y, S) = \Pr(y|S)p(S) = \Pr(y|\theta_S)\Pr(S)$, with θ the parameter(s) of the distribution (McLachlan and Peel, 2000)[Chapter 1].

Two questions arise from the new representation. The first of which is how one is to model the stochastic variable given that the sub-population is known. To solve this methods such as the linear regression is used. The second question is how to model the group indicator S. A simple choice of group indicator is to give it the relative frequency of the occurrence of the sub-populations. That is to say, the probability that the stochastic variable is in the group is the same as the number of times events have been realized from the respective group in the past. The representation of such a finite mixture distribution is

$$\mathbf{Y} \sim \sum_{l=1}^{k} \omega_l f_{\sigma_l}(x - \mu_l), \tag{4.1.8}$$

where k is the number of components in the mixture models. The function f is the probability distribution function of the components. The component distributions are any feasible distribution function. In Equation (4.1.8) the parameters σ_l and μ_l are the shape

28

and location parameters[1] of the components of the distribution function of the different components. The parameter ω is the group indicator of the mixture, in Equation (4.1.8) put to be the relative rate of the respective group. However, this choice assumes that the relative population size of the different groups within the sample space remains fixed.

Another approach with regards to the group indicator variable is the regime switching model. In the switching regression model the choice of group is linked to a value τ, called the breaking value (Fruhwirth-Schnatter, 2006)[Chapter 8]. The model can be expressed as

$$\mathbf{Y} \sim \begin{cases} X_i \boldsymbol{\beta}_1 + \epsilon_1 & \epsilon_i \sim \mathcal{N}(0, \sigma_{\epsilon_1}^2) \quad , \tau < i \\ X_i \boldsymbol{\beta}_2 + \epsilon_2 & \epsilon_i \sim \mathcal{N}(0, \sigma_{\epsilon_2}^2) \quad , \tau \geqslant i. \end{cases} \tag{4.1.9}$$

However, a re-parametrization can be done with $\mathbf{Y}_i \sim X_i \boldsymbol{\beta}_1 \mathbb{I}(\tau < i) + X_i \boldsymbol{\beta}_2 \mathbb{I}(\tau \geqslant i) + \sigma^2$, with $\sigma^2 = \mathbb{I}(\tau < i)\sigma_{\epsilon_1}^2 + \mathbb{I}(\tau \geqslant i)\sigma_{\epsilon_2}^2$. This model relies on the fact that the breaking value is known. Otherwise, the breaking value itself must be considered as a stochastic variable. Two such models are the variance inflating model of Abraham and Box (1978) and the local shift model of Box and Tiao (1968). These models both uses a mixture of two Gaussian densities. The first model uses as it names suggests a (constant) shift k_1 in variance of $\mathbf{Y}_i \sim (1-\eta)\mathcal{N}(X_i\beta, \sigma_\epsilon^2) + \eta\mathcal{N}(X_i\beta, k_1\sigma_\epsilon^2)$. The second model incorporate a shift in expected value k_2 of the stochastic variable $\mathbf{Y}_i \sim (1-\eta)\mathcal{N}(X_i\beta, \sigma_\epsilon^2) + \eta\mathcal{N}(X_i\beta + k_2, \sigma_\epsilon^2)$. This enables the models to capture either an increase in uncertainty or a change in the expected value. Both of the models can easily be expanded into other distributions than the Gaussian which is employed in these examples. We could in addition envision an expansion of the models by making the parameters k_1 and k_2 functions of explanatory variables. The inference of the models mentioned by Abraham and Box (1978); Box and Tiao (1968) are done by MCMC as explained in Fruhwirth-Schnatter (2006)[Section 8.3.4]. There, a two stage Gibbs sampler is proposed to estimate the parameters.

4.2 Evaluation Methods

In this section the evaluation of both point and probabilistic forecasts are considered. As the evaluation methods give a score indicating the performance of the evaluation they are called scoring functions. The scoring functions are written $S(x, y)$, where the parameter x denotes the observation and the parameter y denotes the forecast. The point forecast uses, as the name suggest, only a single value to evaluate. Several values of the forecast ensemble might be chosen to represent the point forecast. Examples of such values are the median or the mean of the forecasting samples. Since multiple values might get chosen, it is important to specify which function that is used to represent the forecast. The choice of forecast function is studied in the next section. The probabilistic forecast have only

---

[1]The reason we write shape and scale parameters is that we have the normal distribution in mind. However, we could just as easily written that these parameters were the shape and scale parameters, which are used by the beta and gamma distributions among others. The essence is that the parameters needed to define the distribution function are given, not which or how many they are.

recently gained momentum as a preferred prediction method. Consequentially, specific methods evaluating these forecasts have only come to attention recently. Therefore we give a short tutorial in this field. In this tuition both uni-variate and multivariate methods are discussed, even though we only will use the latter in later chapter. On the other hand are the discussion of the uni-variate cases important as they highlight important points. In addition consist many of the multivariate procedures in reducing the problem into a setting where it can be solved by the uni-variate method.

4.2.1 Point Forecast Evaluation

Four classical approaches to evaluate forecasts are the root mean square error (MSE) (Contreras et al., 2003; Garcia et al., 2005), mean absolute error (MAE), mean absolute percentage error (MAPE) and mean relative error (MRE) (Wu and Shahidehpour, 2010). The scoring functions for these evaluation criteria are

$$S_{\mathrm{MSE}}(\mathbf{x}, \mathbf{y}) = \frac{1}{T} \sum_{t=1}^{T} (x_t - y_t)^2, \tag{4.2.1a}$$

$$S_{\mathrm{MAE}}(\mathbf{x}, \mathbf{y}) = \frac{1}{T} \sum_{t=1}^{T} |x_t - y_t|, \tag{4.2.1b}$$

$$S_{\mathrm{MAPE}}(\mathbf{x}, \mathbf{y}) = \frac{1}{T} \sum_{t=1}^{T} \left| \frac{x_t - y_t}{y_t} \right|, \tag{4.2.1c}$$

$$S_{\mathrm{MRE}}(\mathbf{x}, \mathbf{y}) = \frac{1}{T} \sum_{t=1}^{T} \left| \frac{x_t - y_t}{x_t} \right|, \tag{4.2.1d}$$

where T the number of observations for which the model is to be evaluated. We note that the $S_{\mathrm{MAPE}}(x, y)$ is exposed to small observed values. When an observation is small (in absolute value) the denominator is small. In this scenario we are unreasonably punished when predicting values that in reality lie close to the observed ones (Nogales et al., 2002). This scenario where instability in the MAPE score occur is detected by another point evaluation methods. The $S_{\mathrm{MRE}}(x, y)$ is constructed just in such a case. This scoring function put the observed rather than the predicted value in the denominator. Thus, if the predicted value is small something that would lead to an unjustified penalization in the $S_{\mathrm{MAPE}}(x, y)$, then this can be observed by the value of $S_{\mathrm{MRE}}(x, y)$ staying low. If however both scoring functions give the predicted value a bad score, then we should conclude that a bad prediction has been made.

The possibility of making a more fundamental errors in the use of point evaluation methods are discussed in Gneiting (2010). In addition the article gives a summary of statistics of the number of times the different evaluation methods are used by other articles. The evaluation methods considered are the same as those presented in Equation (4.2.1). It also includes a case study, where the different scoring functions are employed. The essential parts of the case study is revisited here. We consider the stochastic process $Y_t = Z_t^2$,

where $Z_t \sim \mathcal{N}(0, \sigma_t^2)$. In addition, the variance is a conditional heteroscedastic time series $\sigma_t^2 = 0.20Z_{t-1}^2 + 0.75\sigma_{t-1}^2 + 0.05$. For this time series three strategies of predicting an one time-ahead forecast are studied. The different strategies are labeled: The Statistician, Optimist and Pessimist. The strategy of the Statistician is to predict the true conditional mean $\hat{x} = \mathbb{E}(Y_t|\sigma_t^2) = \sigma_t^2$. The Optimist always predicts the value 5, while the Pessimist always predict the value 0.05. The case study then performs 100,000 simulation of 200 points in time of the time series. Table 4 of the paper is here reproduced as Table 4.2.1. As we can see from the table the Pessimist performs best in half of the evaluation criteria (MAE and MAPE,) while the Statistician and Optimist perform best in a single evaluation criteria each (respectively the MSE and MRE.) This is somewhat disturbing, since the Statistician uses information about the underlying model. The point that the article emphases, is that in the evaluation of the prediction we always use the mean to represent the point forecast. By doing so, we have done what is sometimes done by forecasters. However, the article argues that the scoring function, here the MSE, MAE, MAPE and MRE, is to be disclosed before the forecast is made. This is because the choice of scoring function influence the forecast functions.

A forecast function such as the mean, quantile or fractional moment has to be chosen on the basis of scoring function. The best forecast function is the one that minimize the expected loss function with respect to the predictive distribution. For the MAPE and MRE a new moment has to be defined, namely the fractional moment, which is defined as: "...If the predictive distribution F has density f on the the positive real half-axis and a finite fractional moment of order β, the optimal point forecast under the scoring function, $S_\beta(y, x) = \left| 1 - \left(\frac{x}{y}\right)^\beta \right|$, where $\beta \neq 0$, is the median of a random variable whose density is proportional to $y^\beta f(y)$. We call this the β-median of F and write $\text{med}^\beta(F)$. The traditional median arises when $\beta \to 0$." (Gneiting, 2010). The resulting optimal forecast functions for the scoring functions in question is reproduced in Table 4.2.2. In this table the forecast functions for the respective scoring function are used to calculate new and improved values for the time series example. We see that the new forecast functions greatly improve the old values obtained for the scoring functions that were found in Table 4.2.1. The only exception is the MAE, where the Statistician used the appropriate forecasting function in the first calculation. Thus, the example shows the importance of choosing the right forecast function. This forecast function can be found by a procedure including the maximization of the scoring function.

4.2.2 Univariate Rank Histograms

In the following sections we will explore evaluation methods which use the entire probabilistic forecast. This method include the rank histogram, the energy score among others. By applying the evaluation methods Gneiting et al. (2007) defines the objective of a predictive performance to be the "*maximization of the sharpness of the predicted distribution subject to calibration*". The calibration being the measure of consistency between the predictive distribution and the observed realization. This could be that 90% and 50% of the observed

Forecaster	SE	AE	APE	RE
Statistician	**5.07**	0.97	2.58×10^5	0.97
Optimist	22.73	4.35	13.96×10^5	**0.87**
Pessimist	7.61	**0.96**	$\mathbf{0.14 \times 10^5}$	19.24

Table 4.2.1: *Table 4 of Gneiting (2010). The table shows values of different evaluation criteria. The scoring functions are applied on an one time-ahead forecast, where different prediction strategies are tried. These strategies are labeled the Statistician, Optimist and Pessimist. The boldface values are the strategy with the best performance for the given scoring function.*

Scoring Function	Bayes Rule	Point Forecast in Simulation Study
$S_{\mathrm{MSE}}(\mathbf{x}, \mathbf{y})$	$\hat{x} = \mathrm{mean}(F)$	σ_t^2
$S_{\mathrm{MAE}}(\mathbf{x}, \mathbf{y})$	$\hat{x} = \mathrm{median}(F)$	$0.455\sigma_t^2$
$S_{\mathrm{MAPE}}(\mathbf{x}, \mathbf{y})$	$\hat{x} = \mathrm{med}(^{(-1)}F)$	ϵ
$S_{\mathrm{MRE}}(\mathbf{x}, \mathbf{y})$	$\hat{x} = \mathrm{med}(^{(1)}F)$	$2.366\sigma_t^2$

Table 4.2.2: *Table 5 of Gneiting (2010). In the table the appropriate forecast functions according to the pre-determined scoring functions are given. The scoring function are the mean squared error (MSE), mean absolute error (MAE), mean absolute percentage error (MAPE) and mean relative error (MRE). The ϵ denotes a very small positive number. By $med^{(\beta)}$ we mean the fractional moment of order β.*

|	SE	AE	APE	RE
Value	5.07	0.86	1.00	0.75

Table 4.2.3: *Table 6 of Gneiting (2010). In the table the value of different scoring function under their respective optimized forecast function is seen. The optimal forecast function of the different scoring functions are found in Table 4.2.2. The predictions evaluated are the same as in Table 4.2.1.*

values lie within the respective 90% and 50% interval of the predictive distribution. The sharpness on the other hand refers to the size of the prediction interval and is thereby a quality of the predictive distribution alone. These two objective functions are used to prevent the predictive distribution to be too wide, and thus get it to cover a larger fraction of the observed variables without being penalized.

When forecasting future values the objective is to obtain a distribution that is as close as possible to the distribution of the observed values. In other words we are trying to calibrate our distribution to the distribution that nature has chosen to our stochastic variable. One way of quantitatively measure the degree in which this is satisfied is to use a rank histogram as first proposed by Talagrand et al. (1997). The rank histogram is used to evaluate discrete ensemble forecast by assessing the threshold value and the ordered observed values. For continuous forecasts the *probability integral transform* (PIT) is used (Gneiting et al., 2004). However, for the PIT the cumulative density function (CDF) is used instead of the ordered observed values. The PIT is frequently used in the community of statistical meteorology (Gneiting et al., 2008; Gneiting and Raftery, 2007).

The PIT is a method for deciding whether the predictive distribution used is calibrated for the outcomes. Alternatively, the PIT shows why a calibration of the predictive distribution is not achieved. As the name suggests the values of the PIT is $p_i = F_i(y_i)$, where F_i is the predictive distribution function and y_i a single outcome. This distribution function is usually the same for all of the outcomes, but may also change for different outcomes. If the distribution function is continuous and calibrated the histogram generated by several outcomes of the p_i should be uniform. The uniformity of a rank histogram, or the lack thereof, is an indication of the degree of calibration between the distribution of the predicted and the observed values. To give an example we consider the calibration of some Gaussian distributions with different parameters. The PIT of these examples are presented in Figure 4.2.1. Peaked histogram as the ones that are in the top row of Figure 4.2.2 indicate uncalibrated predictions. From these histograms it we conclude that the predictive distributions have a smaller variance than the distribution of the observed values. On the contrary is the case of the U-shaped rank histogram, as seen in the bottom row of Figure 4.2.2. This case occur when the distribution of the observed values has a greater variance than that of the predictive distribution. An histogram which has an increasing (decreasing) slope from left to right indicates that the mean of the predictive distribution is greater (smaller) than the distribution of the observed values. This is observed in the histogram in the right column of Figure 4.2.2.

The method of calibration is however not entirely fool proof. This stems from the fact that a calibrated distribution will display a uniform rank histogram, but the opposite needs not to be true. That is, a uniform rank histogram does not necessarily originate from a calibrated distribution. In Hamill (2001) two cases which highlight this issue is presented. In the first case the predicted values are drawn uniformly from one of three different uncalibrated distributions. In the example the observed values are drawn from another distribution. The paper shows that the predicted values are able to produce a uniform rank histogram when compared to the observed values. The second case presented in the paper is when a requirement is added to the predicted values after the sampling is
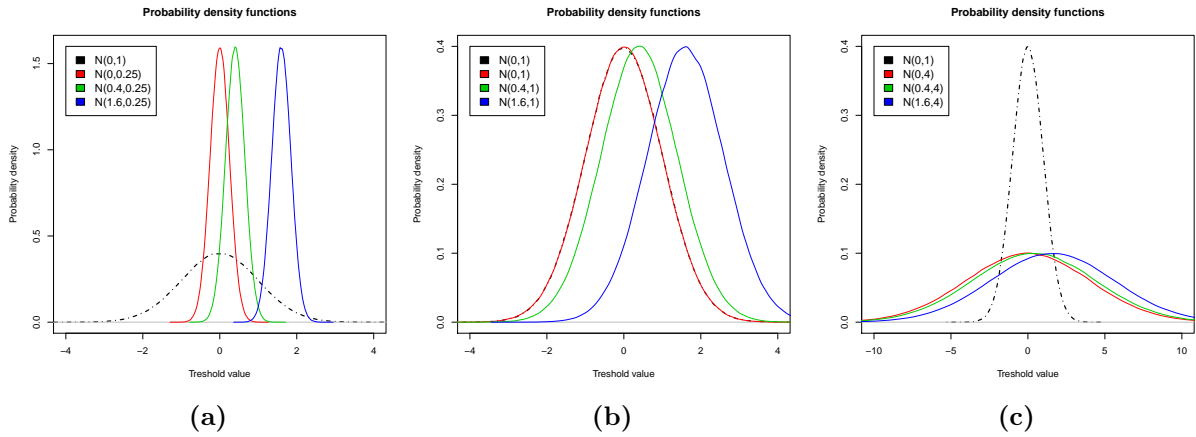
Figure 4.2.1: *In Panel (a)-(c) the standard Gaussian distribution is the thick black dotted line. The other lines are density functions of the Gaussian distribution with different parameters as indicated in the legends. These functions are the used in the evaluation of the PIT histograms in Figure 4.2.2. The panels are (from left to right) the predictive distribution of different rows (from top to bottom) in Figure 4.2.2. It is important to note that the respective y- and x-axis do not have the same scales in the different panels.*

done. Thus, by post-processing the predicted values an uniform rank histogram is achieved. These two examples are important to consider when analyzing the rank histogram, and shows that one cannot rely on this evaluation method alone. In addition, the second case where the predicted value is post-process may be used to improve forecasts. In Raftery et al. (2005) and Berrocal et al. (2008) a post-processing of the predictions are performed. This post-processing is based on the rank histograms, and is used to improve the predictive distribution.

4.2.3 The Continuous Rank Probability Score

In this section the method evaluating the sharpness of the prediction in one dimension is presented. The sharpness of a forecast is an important question. When a probabilistic forecast is made a minimization of the uncertainty in the prediction is often preferred. When dealing with predictive distributions the uncertainty are linked to the width of the distribution function. In other words, a sharp function will give a small interval of simulated values from the distribution. Thus, we would like to minimize the width of the predictive distribution. Further it is important that an evaluation method always gives the forecast that predicts the true value the best score. Functions that have this property are called *bena fide* scoring functions. It is also beneficial if the scoring function is stable. By this we mean two things. Firstly, the function should not be dominated by a single value in the forecast. That is, if a single predicted value is far off its observed value, then this should not make us reject the entire forecast. Second, the scoring function should not denote forecast with infinite values if they give a finite prediction.

As we have seen previously in this chapter a scoring rule can be defined which evaluates
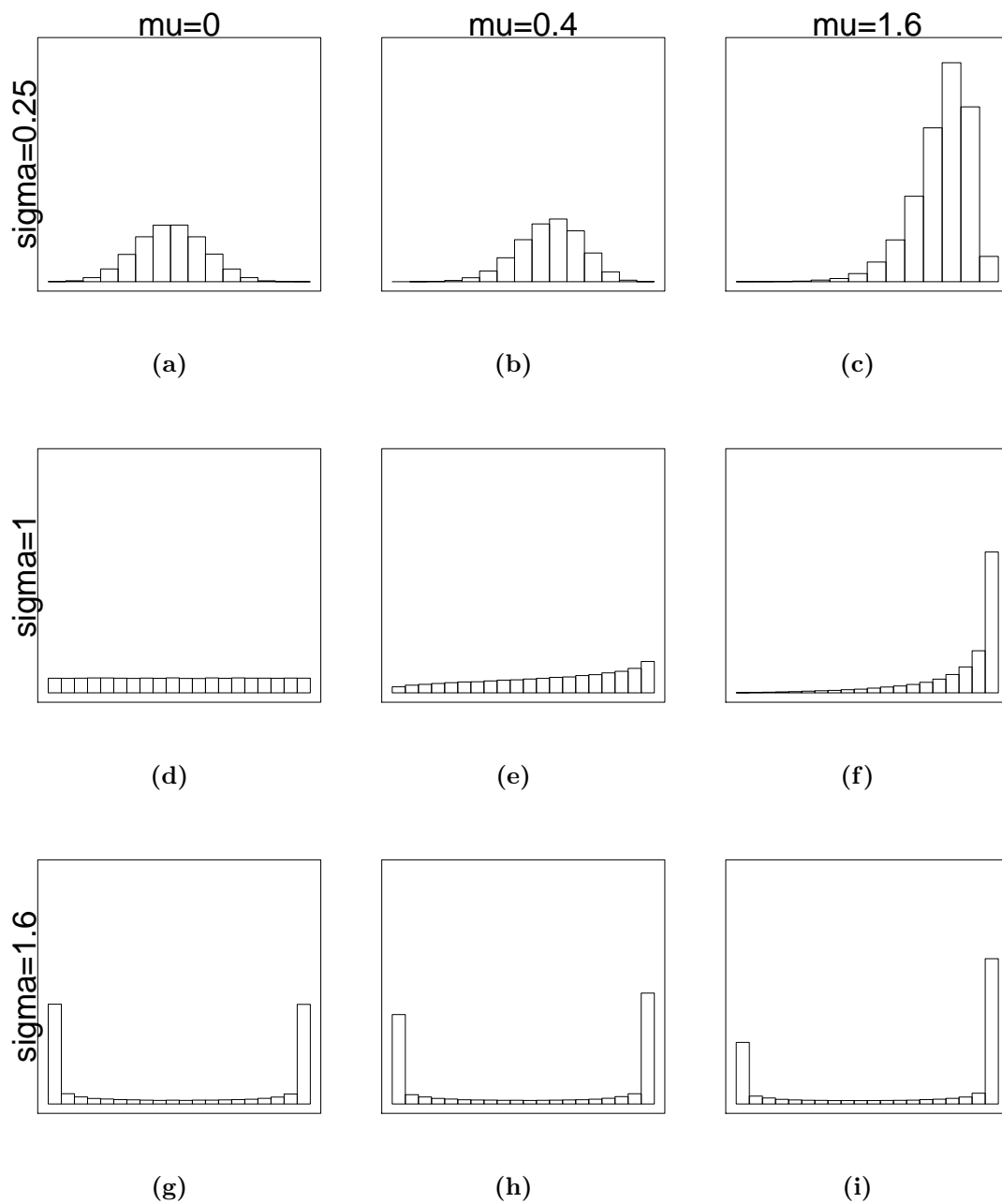
Figure 4.2.2: *Theoretic probability integral transform (PIT) histograms. The observed values are sampled from the standard Gaussian distribution with different means (μ) and standard errors (σ). The different means and standard errors are indicated to the left of the rows and at the top of the columns. The predictive distribution function is at all times that of the standard Gaussian distribution. Due note that the y-axis are the same in each of the figures. That is to say with values between 0 and 16. The range of the x-axis is also the same for all figures, namely from 0 to 1. In the figures we note that if the observed values have greater mean than the predictive distribution the histograms are right skewed. On the other hand if the standard error is to great the PIT histograms are "smiling". Too low variance gives the opposite effect, which results in a hump.*

the predictive distribution function F. Two such scoring functions are the *Brier score* (BS) and the *continuous rank probability score* (CRPS). The choice of the these scoring functions is based on the fact that they are bena fide and that they are stable (Gneiting et al., 2007). A theoretical walk-through of these scoring functions are given in Gneiting and Raftery (2007) and case studies are provided in Gneiting et al. (2007, 2008). The BS is a verification tool for the prediction of a specific event. The BS is used to evaluate predictions of binary events. Thus the value of the BS is either zero or one. However, the continuous predictions can also be evaluated once a threshold value is introduced. In Gneiting et al. (2007) Brier score is defined to be

$$\text{BS}(z) = \frac{1}{T} \sum_{t=1}^{T} \{F_t(z) - \mathbb{I}(z \geqslant x_t)\}^2, \tag{4.2.2}$$

where x_t is the observed value at time t. The variable z is a threshold value, while the function F_t is the cumulative predictive distribution (CDF) at time t.[2] The parameter T denotes the length of the time period that is to be evaluated by the Brier function. The Brier score function evaluates the performance of the predicted distribution towards the empirical distribution. The function evaluated to with degree the predictive distribution captures real distribution. This is done by measuring the probability that an event occur with respect to the frequency with which it is observed in nature. To do this we have to consider all possible threshold values. The continuous rank probability score does exactly this. The CRPS is defined as

$$\text{CRPS} = \frac{1}{T} \sum_{t=1}^{T} \text{crps}(F_t, x_t) = \int_{-\infty}^{\infty} \text{BS}(z) \, \mathrm{d}z. \tag{4.2.3}$$

The objective is to minimize the CRPS, which is the mean of several crps evaluated for the same predictive forecasting model over different times. It is important to note that the CRPS is scoring function which depends on the unit of the predicted variable(Berrocal et al., 2008)[3]. As a result, two models can be compared directly using the CRPS once the physical quantity is taken into account. However, this is not an attempt to say that all stochastic variables are equally easy to model, just that models can be compared. The CRPS can also be used to evaluate deterministic forecasts. It then reduces to the mean absolute error (MAE) described in Section 4.2.1. This can be shown by a simple calculation which is done in the Appendix.

An example of the BS and CRPS for a single point in time and a given threshold is shown in Figure 4.2.3. The BS is the distance from the dotted line and the intersection of the red and the blue curve. The CRPS is the square of the light blue area between the red and blue curve. The perfect forecast would be a deterministic forecast which would result in a vertical cumulative distribution function at the same place as the observed value. That is, the blue area and thus the CRPS would then be zero, which is the minimum value for the

---

[2]Using the previous notation we have that F_t is the (empirical) CDF of the predicted values $\mathbf{y}_t$ at time t.

[3]Unlike the log scoring function like the maximum likelihood among others.

A realization of CRPS for a single value
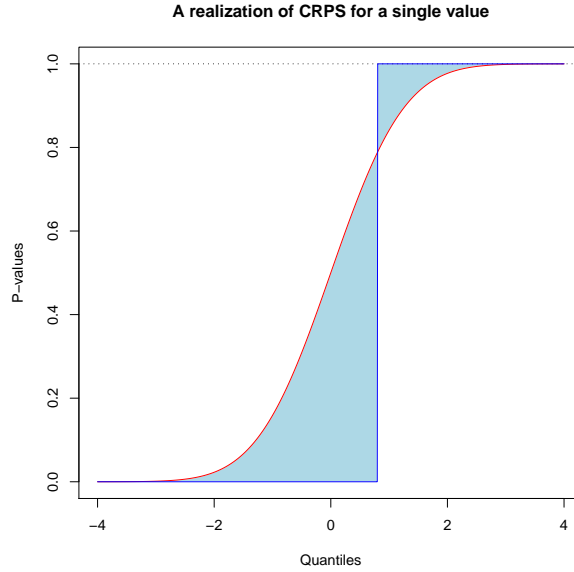
Figure 4.2.3: *A figure of the empirical cumulative distribution function (ECDF) of predicted values (red) and the ECDF of a single threshold value with respect to the observation, i.e a Heaviside function, which is to be evaluated (blue). The light blue area highlighted between the curves squared is the continuous rank probability score (CRPS) score of the single threshold value which is evaluated.*

CRPS. A noteworthy fact is that the CRPS may alternatively be defined as the negative BS (Gneiting and Raftery, 2007), which is important if the reader chooses to do further literature studies. An example of theoretical Brier score plot is given in Figure 4.2.4, where the predictive density functions are the same as the ones used the example of evaluating PIT histograms in Figure 4.2.2. The corresponding CRPS values are given in Table 4.2.4. The examples evaluate the score functions of the predictive Gaussian distribution with different parameters. The observed values are however sampled sampled from the standard Gaussian distribution. As one may notice the different BS of two predictive distributions are hard to compared. As a result the calculation of the BS should only be considered as a step to finding the CRPS. The CRPS of the example is found in Table 4.2.4. The prediction which has the same distribution as the observed values, is the distribution which has the lowest CRPS.

To give further insight into the CRPS another example is given. This time we consider a prediction of a 24 hour period seen in Figure 4.2.5a. The forecast is divided into 4 groups of equal size. The observed value is a sinus function. All of the groups in the forecast consist of the value of the true function plus some noise. The noise terms in the four groups are different. In group 1 the noise terms are sampled from a $\mathcal{N}(0, 0.1)$ distribution. In group 2 the noise terms are sampled from a $\mathcal{N}(-0.5, 0.1)$ distribution. In group 3 the noise terms are sampled from the $\mathcal{N}(0, 0.4)$ distribution. In group 4 however the noise terms are zero, thus for this last group we have no noise. In Figure 4.2.5b the corresponding Brier function for a single forecast from each groups are shown. We notice that the Brier function for

$\sigma \setminus \mu$	0	0.4	1.6
0.25	0.6547	1.0592	1.2617
1.0	0.5625	1.0350	1.2770
4.0	1.0305	1.0506	1.2773

Table 4.2.4: *The table of the continuous rank probability score (CRPS) values for the Brier score (BS) functions of Figure 4.2.4. The BS functions are in turn realizations of the probability integral transform (PIT) plots in Figure 4.2.2. The columns and rows are the CRPS value for different predicted Gaussian densities with different mean and variance. All of the predicted densities are all evaluated toward a observed density. The observed density is the standard Gaussian. All of the densities are found in Figure 4.2.1.*

the perfect forecast (group 4) is everywhere zero. We also notice that the biased forecast of group 2 is most penalized, with a maximum BS function value of one. Remembering Equation (4.2.2) this can be expected, since the indicator function dominates the function completely until the empirical distribution kicks in. This intervention of the empirical distribution function is much "smoother" for the Brier function of group 1 and 3, which in addition have a symmetric BS function. This comes from the fact that their forecast ensembles are sampled from distributions that are unbiased and symmetric with regard to the true value. For these groups we notice that the Brier function is wider for the function with the most variance in the noise term. Thus, the BS penalizes the least sharp distribution the most. The observations concerning the BS are reflected in the value of the CRPS. The CRPS value for group 1, 2, 3 and 4 is 0.024, 0.445, 0.095 and 0 respectively.

The last example confirms the promises made with respect to the CRPS. In addition it showed that the CRPS can in fact be used to evaluate deterministic forecasts. Further, the example illustrated the range of values the CRPS might take. From Equation (4.2.2) we know that the maximum value of the BS is one. As seen in Figure 4.2.3 this is the case. As the crps denotes the integral over all points the worst possible score will be infinite. However, when the crps is used in a setting where the reasonable evaluation value of the function is unknown, a naive method is used as a benchmark. As deterministic forecasts can be evaluated these are often used. An example of such benchmark is given in paper Gneiting et al. (2008). Here the mean of recent wind observations are used as the benchmark. Thus, the more advanced statistical method used in the paper are evaluated against this. However, this method of using deterministic forecasts as benchmark is only done if a competing statistical prediction does not exist.

4.2.4 Multivariate Calibration

In the previous sections evaluation of one-dimensional forecasts was discussed. In this section calibration of forecasts in more than one dimension is discussed. Thus, in the following sections the forecasts are given in vector form. To evaluate the calibration of a forecast the same Talagrand histogram as seen in Figure 4.2.2 is used. The only difference is the forecasts have to be pre-processed in order to use the same framework. The main goal of the pre-prosessing is to get the predicted values into a setting where they are of internal order is in a one-dimensional setting. That we need to have a framework where we can
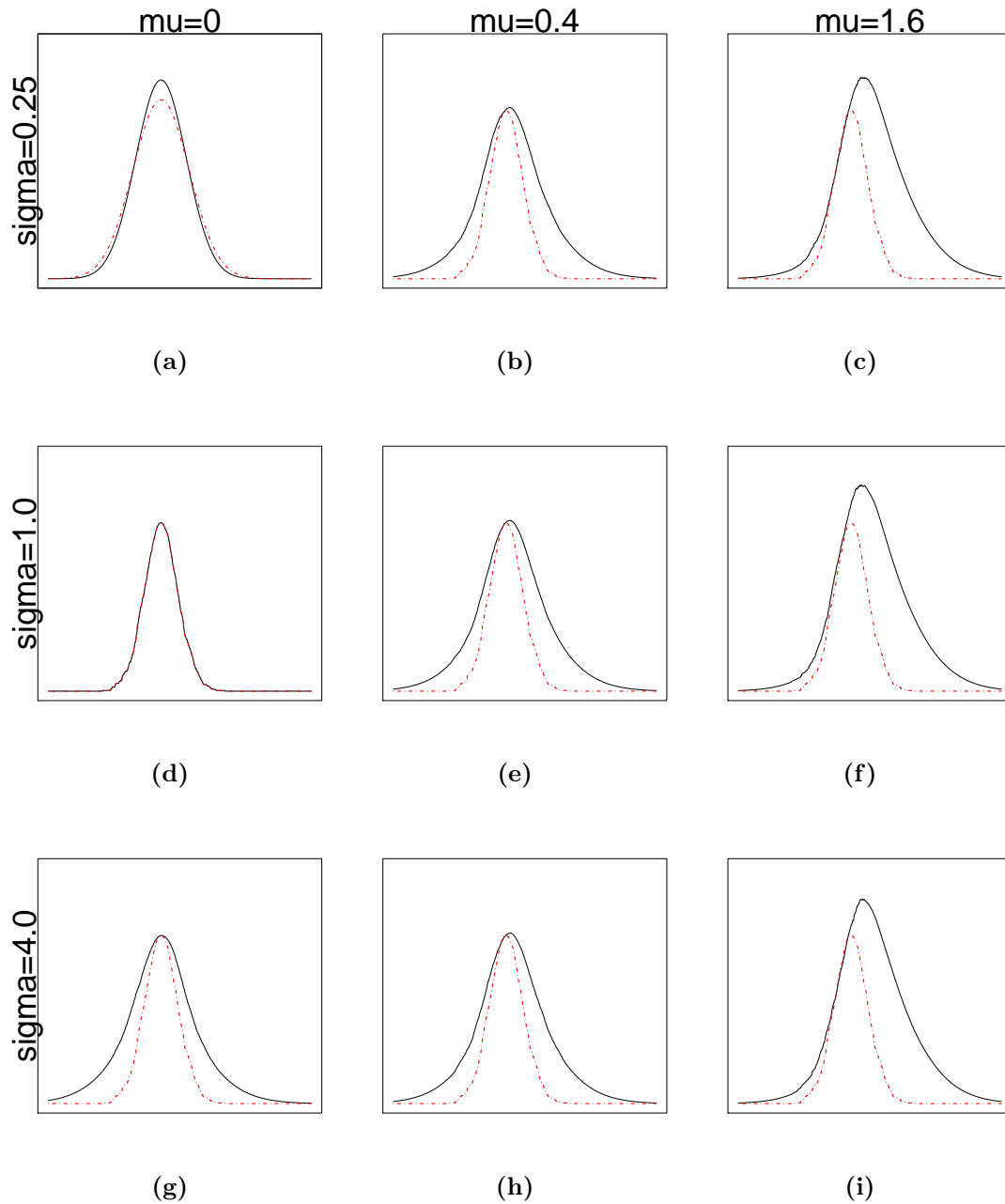
Figure 4.2.4: *Theoretic Brier score functions. The values on the x-axis corresponds to different threshold values, while the values on the y-axis are the corresponding value of the Brier score. The plots are each the mean over 1000 time periods T. The observed values are sampled from the standard Gaussian distribution. The predicted values are also sampled from the Gaussian distribution. However, the parameters of the predictive distribution functions vary. These parameters are indicated to the left and at the top of the panels. Each row has the same mean value, while each column has the same standard deviation. The dotted red line is the ideal predictive function, namely that the predictive distribution being the same as the distribution function of the observed values. It is important to note that scales of the y- and x-axis have been kept unchanged for all the plots. The range of values on the x-axis being from -10 to 10, and the values on the y-axis being from 0 to 0.35. Also note that the Brier score for (d) is that of the ideal forecast. In addition the Brier scores corresponds to the PIT plot given in Figure 4.2.2 and the corresponding CRPS values are given in Table 4.2.4.*

39
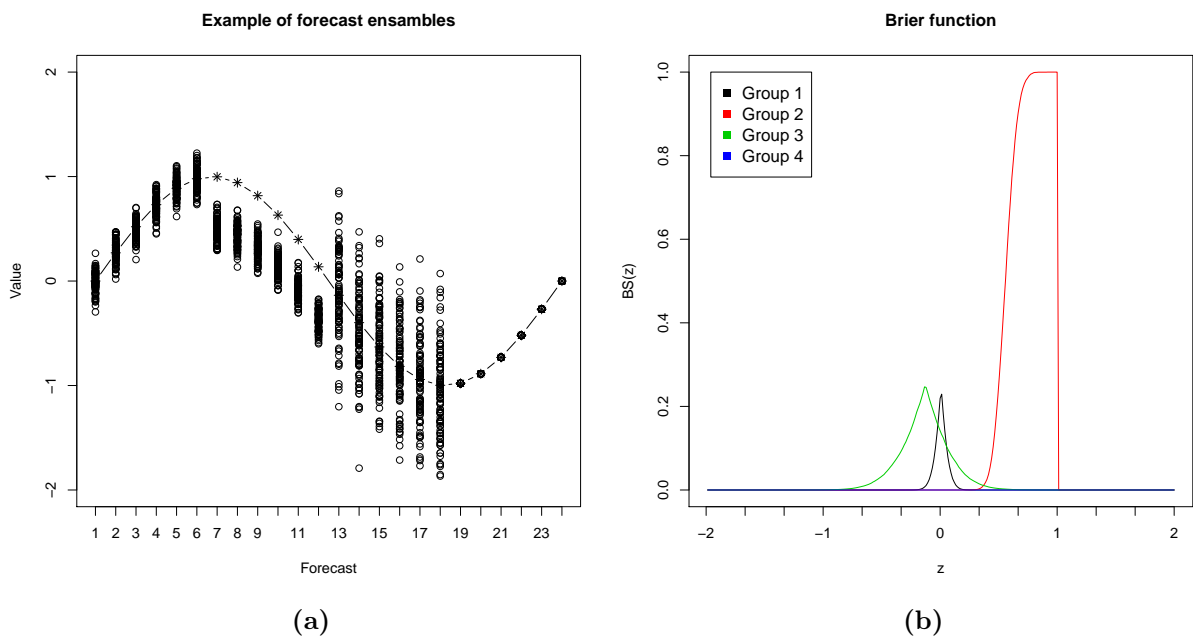
Figure 4.2.5: *In Panel a) an artificial forecast of 24 time periods is shown. The forecasts are divided into 4 groups, where group 1 is at time 1-6, group 2 is at time 7-12 etc. All the time periods have a forecast ensemble size of 100 which is marked with dots. The observed values are marked with an asterisk. Each of the time periods in the same groups share the same parameters in the Gaussian distribution of the error terms . Group 1 is unbiased and have low variance, group 2 is biased and low variance, group 3 are unbiased with large variance and group 4 are unbiased with zero variance. In Figure b) the corresponding brier score for one of the time periods from each groups is shown.*

40

map the predicted values to the natural numbers. The algorithm for doing this given by Gneiting et al. (2008): When considering a vector $\mathbf{x} = (x_1, x_2, ..., x_d)$ and $\mathbf{y} = (x_1, x_2, ..., x_d)$ in $\mathbb{R}^d$ we define

$$\mathbf{x} \prec \mathbf{y} \quad \text{iff} \quad x_i \leqslant y_i \quad i \in \{1, ..., d\}.. \tag{4.2.4}$$

Thus, for an ensemble forecast $\mathbf{x}_j \in \mathbb{R}^d$, $j \in \{1, ..., K\}$, where K is the size of the ensemble, a multivariate Talagrand histogram can be constructed by the following steps:

i. Standardize: First an optional standardization of the ensemble members are performed using the principal component method.

ii. Pre-rank assignment: Members are pre-ranked:

$$\rho_i = \sum_{j=1}^{K} \mathbb{I}(\mathbf{x}_j \prec \mathbf{x}_i), \tag{4.2.5}$$

where $\mathbb{I}$ denotes the indicator function. Thus all the members of the ensemble are given a non-distinct value between 1 and $m+1$. This is depicted in Figure 4.2.6a.

iii. Find multivariate rank: The multivariate rank r of the ensemble is found. The multivariate rank is the same as the pre-rank. However, for cases with members of same rank, the tie is resolved by giving them a rank between $(s^< + 1, ..., s^< + s^=)$ at random. The variables $s^<$ and $s^=$ are the number of ensemble members which have lower and the same pre-rank as the member that are to be consider respectively. These variables are found by calculating

$$s^< = \sum_{j=1}^{K} \mathbb{I}(\rho_j < \rho_0) \quad \text{and} \quad s^= = \sum_{j=1}^{K} \mathbb{I}(\rho_j = \rho_0). \tag{4.2.6}$$

The variable ρ_0 is the pre-rank of the ensemble member which is to be evaluated.

The second and third step of the procedure are depicted in Figure 4.2.6b. The same Talagrand histogram that was used in the uni-variate case can be used once the algorithm is employed on the prediction. Thus, we can use the diagnostic tools as the once presented in Section 4.2.2.

4.2.5 Evaluation of Sharpness in a Multivariate Setting

The CRPS cannot evaluate sharpness of a multivariate prediction as it is defined. This comes from the fact that the Brier score function uses the indicator function, and we then come into the same problem of ordering as in the calibration. The CRPS encountered in Section 4.2.3 is however a special case of the energy score (ES). Let $\mathbf{X}$ and $\mathbf{X}'$ be two independent copies of a random vector with distribution P in $\mathbb{R}^m$. Then, as long as $\mathbb{E}||X||^{\beta}$
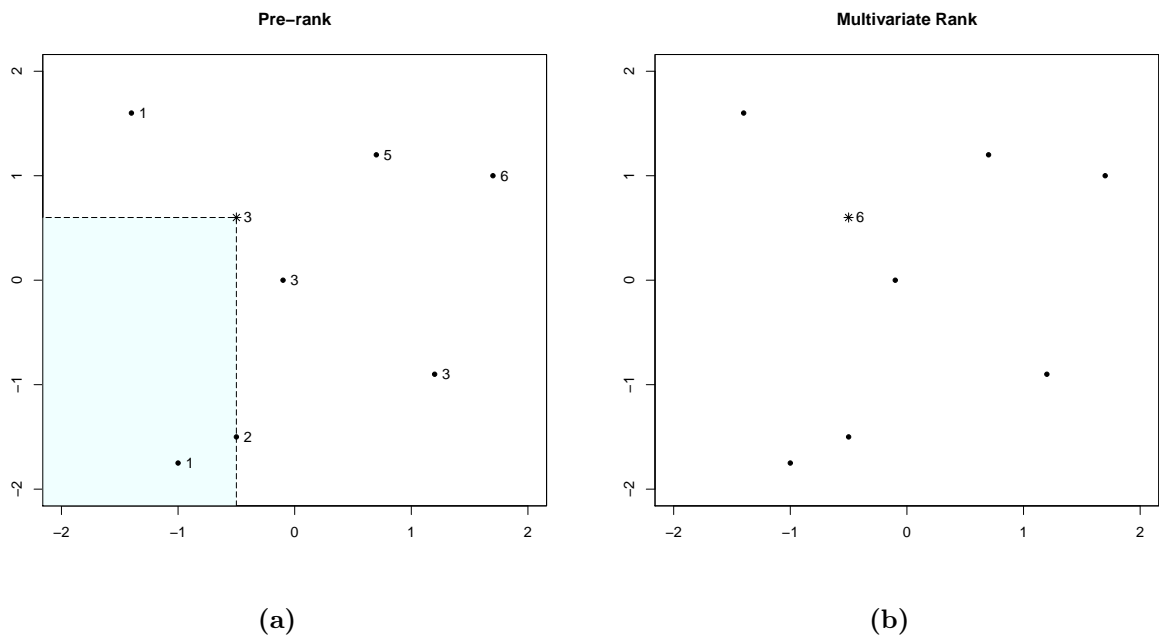
Figure 4.2.6: *Figure 2 of Gneiting et al. (2008). In Panel a) the ensemble members are pre-ranked following Equation 4.2.5. The light blue area is the area which have a smaller rank than the member marked with an asterisk. Any member in this area is outranked by the ensemble member with an asterisk. Three such ensemble members have the same rank as the asterisk member. This has to be resolved using the procedure given in Equation 4.2.6.By performing the calculations of Equation 4.2.6 the ensemble members with asterisk is found to have the values $s^< = 3$ and $s^= = 3$, which are the number of ensemble members with equal and smaller pre-rank respectively. Thus, the possible ranks of the ensemble member are $\{4, 5, 6\}$. Sampling from uniformly from the possible ranks the rank for the asterisk member becomes six. In Panel b) the pre-processing of the ensemble members are finished. The ensemble is then ready to by plotted in a Talagrand histogram.*

is finite [4] where $|| \cdot ||$ is the Euclidean norm and the parameter $\beta \in (0,2)$, then the ES is defined

$$\text{ES}(P, \mathbf{x}) = \frac{1}{2} E_P ||\mathbf{X} - \mathbf{X}'||^\beta - E_P ||\mathbf{X} - \mathbf{x}||^\beta, \qquad (4.2.7)$$

where $\mathbf{x}$ is a vector of observed values (Gneiting et al., 2008). Thus, the evaluation of sharpness in a multivariate setting uses the energy score. The CRPS is the special case when beta equals one. Then the Euclidean norm of Equation refeq:energyscore becomes the absolute value. The definition of the ES is then $\text{ES}(P,x) = E_P|X - x| - \frac{1}{2} E_P|X - X'|$, which is shown in Lemma 2.2 of Baringhaus and Franz (2004) or the identity of Equation (17) of Szekely (2003) to be identical to Equation (4.2.3).

The expression in Equation (4.2.7) has no closed form. However if we assume that the density of the predictive ensemble is $P = P_{\text{ens}}$, P_{ens} a numeric expression can be found. The predictive density proposed gives point mass $1/K$ to each of the ensemble members $\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_K \in \mathbb{R}^m$, then following Gneiting et al. (2008), the energy score becomes

$$\text{ES}(P_{\text{ens}}, \mathbf{x}) = \frac{1}{K} \sum_{j=1}^{K} ||\mathbf{x}_j - \mathbf{x}|| - \frac{1}{2K^2} \sum_{i=1}^{K} \sum_{j=1}^{K} ||\mathbf{x}_i - \mathbf{x}_j||. \qquad (4.2.8)$$

In addition the ES may also be simplified to be the negative squared error. This simplification happens when $\beta = 2$. The ES is then expressed as $\text{ES}(P, \mathbf{x}) = -||\boldsymbol{\mu}_P - \mathbf{x}||^2$. The parameter μ_P is the multivariate mean of the probability distribution P. We thus see that the ES is a generalization of many of other well known evaluation criteria, and that it thus may be used in a wide variety of settings. In addition we have seen that the evaluation methods of calibration and sharpness encountered in Section 4.2.2 can be generalized into the multivariate case.

---

[4]When this requirement fails the $\text{ES}(P, \mathbf{x})$ can be defined as

$$\text{ES}(P, x) = -\frac{\beta 2^{\beta-2} \Gamma(\frac{m}{2} + \frac{\beta}{2})}{\pi^{m/2} \Gamma(1 - \frac{\beta}{2})} \int_{\mathbb{R}^m} \frac{|\varphi_P(\mathbf{y}) - e^{i<\mathbf{x},\mathbf{z}>}|^2}{||\mathbf{y}||^{m+\beta}} d\mathbf{z},$$

where $|| \cdot ||$ is the Euclidean norm, the parameter $\beta \in (0,2)$, the function $\Gamma(\cdot)$ is the gamma function, the operator $< \cdot, \cdot >$ is the inner product space and the function $\varphi_P(\cdot)$ is the characteristic function of probability distribution P (Gneiting and Raftery, 2007). This score evaluates the weighted distance between the characteristic function of P and the observed value

Chapter 5

Model Specification

In this chapter a model of the error terms of the deterministic electricity spot price model, as introduced in Section 2.4, is proposed. The framework of the model is based on a two-component mixture model as seen in Section 4.1.3. Later we focus on the balanced part of the model which uses the seemingly unrelated regression which we saw in Section 4.1.1. Then we discuss the estimation of the SUR model parameters . Finally the evaluation method of the is discuss, unveiling the optimization procedure that is used to find the best set of explanatory variables.

5.1 Spot Price Forecast Model

In this Section we will describe how the full spot price model is built up. By this we mean before any filtering of prices as described in Section 2.3 is done. The framework is as described in Section 2.4, reproduced here

$$X_v(t, h) = D_v(t, h) + \epsilon_v(t, h) \quad h \in \{0, ..., 23\}, \tag{5.1.1}$$

where $X_v(t, h)$ is the power price at hour h at relative day t to a given date v. The $D_v(t, h)$ is the deterministic forecast as given in Equation (2.4.2). The ϵ is the error term, which we are going to find a model. The error term is described by the mixture model, as described in Section 4.1.3. As already indicated in that section the model consist of two components. Each of the two components is a model of either one of the two groups of error terms that are labeled balanced and unbalanced. The error terms are classified according to the criterion presented in Section 2.3. In other words are the model of the error terms conditioned on the classification of its respective power price. Accordingly is the full model written $\Pr(\epsilon_v(t, h)|X_v(t, h) = \text{Balanced}) = f_1(\theta_1)$ and $\Pr(\epsilon_v(t, h)|X_v(t, h) = \text{Unbalanced}) = f_2(\theta_2)$, where f_i denotes the probability distribution function while θ_i is the associated parameters. These two components are put into a mixture model, where an inflated variance model or mean shift is considered the for the unstable prices, as mentioned in Section 4.1.3

$$\epsilon_v(t, h) \sim (1 - \eta)f_1(X_i\beta, \sigma_u^2) + \eta f_2(X_i\beta + k_1, k_2\sigma_u^2). \tag{5.1.2}$$

This is to be considered as the full model of the error terms. In further analysis of the model is the three separate parts[1] handled individually. Indeed is only the a model for f_1 specified in the of this chapter, Chapter 6 and 7. The discussion of the component model of f_2 and the group-selection function are not taken up again until Chapter 8, and there only in the form of further studies. ***Nota bene*** that we henceforth will denote the balanced error term(s) as simply error term(s) for the sake of simplicity in writing.

5.2   The Unbalanced Spot Price Forecast Model

In this section the model for the balanced error is revealed. In Equation refeq:mixtureEMP the probabilistic distribution together with its expected value and covariance matrix is not given. However, on the basis of the correlation both for within-day hours seen in Figure 2.6.1, the information structure as depicted in Figure 4.2.3 and our deterministic model (which is stated in Equation (2.4.2)) we consider the model to consist of 24 hour blocks. Equally important is the between day correlation which is seen in Figure 2.6.2. To account for this dependency and induce bias correction in the model the probability distribution of the forecast horizon is given conditionally for each lead day. Thus, the joint forecast distributions can be expressed as the conditional marginal distribution for each lead day $\pi(y_1, ... y_{10}) = \pi(y_1)\pi(y_2|y_1)\pi(y_3|y_1, y_2)...\pi(y_{10}|y_1...y_9)$. The marginal distributions are

$$X_v(1, h) \sim \mathcal{N}(D_v(1, h) + X_1\beta_1, \Sigma_1)$$

$$X_v(i, h)|\epsilon_v(i-1, h) \sim \mathcal{N}\left(D_v(i, h) + \binom{X_i}{u_v(i-1, h)}\beta_i, \Sigma_{i|i-1}\right) \quad i \in \{2, ..., 10\}, \tag{5.2.1}$$

where $u_v(t, h) = X_v(t, h) - D_v(t, h) - \epsilon_v(t, h)$ and Σ_i is the within-day variance matrix. The index i refers to the lead day, that is to say the number of days-ahead from the date that we forecast error terms. Inn addition is the variable $u_v(t, h)$ the residuals of the combined deterministic and stochastic model. The model given for the mean and variance model of Equation 5.2.1 is the seemingly unrelated model (SUR). Thus we assume that the error terms are independent between different hours in the day in the expectation, but that the dependence is captured by the covariance matrix Σ. The SUR model is multivariate a linear regression and is explained in detail in Section 4.1.1. To explain the error terms we introduce the relation

$$\begin{aligned}
\epsilon_v(t, h) = {} & C_v(t, h) + \Phi_v(t, h)\epsilon_v(-1, h) + T_v(t, h) + q_v(t, h) + R_v(t, h)+ \\
& + g_v(t, h)\mathbb{I}_{\mathcal{G}}(d_v(t, h)) + p_v(t, h)\mathbb{I}_{\mathcal{P}}(d_v(t, h))+ \\
& + w_v(t, h)\sum_{j\in\mathcal{W}}\mathbb{I}_j(d_v(t, h)) + m_v(t, h)\sum_{i\in\mathcal{M}}\mathbb{I}_i(d_v(t, h))+ \\
& + u_v(t, h).
\end{aligned} \tag{5.2.2}$$

---

[1]The two mixture components and the group-indicator function.

The variables in this full model are the intercept ($C_v(t, h)$), bias correcting term $\Phi_v(t, h)$, temperature ($T_v(t, h)$), consumption ($q_v(t, h)$) and reservoir level ($R_v(t, h)$). The variable $d_v(t, h)$ is a dummy variable, consisting of the information of a given day, and used in the case of the categorical explanatory variables which are linked to the calender. These are explanatory variables such as Norwegian holidays ($g_v(t, h)$), daylight savings ($p_v(t, h)$), weekday ($w_v(t, h)$) and month ($m_v(t, h)$) are also in the model. In Equation (5.2.2) $\mathbb{I}(\cdot)$ is the indicator function and $\mathcal{W} = \{$Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday$\}$ and $\mathcal{M} = \{$February, March, April, May, June, July, August, September, October, November, December$\}$ are the sets weekdays and months, while $\mathcal{G}$ and $\mathcal{P}$ are the days with Norwegian holidays and daylight savings respectively. The number of months and weekdays considered is chosen to avoid any co-linearity in the model.

The variable $d_v(t, h)$ is the information about a given day. Thus if we use it with regards to the set of summer time, we check if the summer time is used for on that day. In the same way is the day check if it is a holiday. For the terms corresponding to the effect of weekday and month we sum over all possible outcomes of the set. However, as the day, $d_v(t, h)$, only can be a member of a single day or month of the time, all except one of the terms in the summation are put to zero. Sometimes are even all of the terms put to zero. This happens for the weekday and month on Monday and in January respectively. On the other hand is the notation chosen used to harmonized with the earlier notation. In addition do the 216^2 parameters of the weekday, $w_v(t, h)$, vary with the day of the forecast, v. As a result is the notation used to emphasis this fact. The same reason is used in the case of the month explanatory variable.

5.3 Parameter Estimation for the Unbalanced Model

In this section we give a short explanation of some the values of the parameters of the Equation (5.2.2). In addition we go through the estimation procedure of the response variable in the same equation. The consumption and reservoir level are in terms of their relative size according to the mean value of the years 2005 and 2006. As the reservoir level value is only reported once a week, a linear interpolation between data points is used to get hourly values. The reservoir level is a physical value that rarely makes jumps, since it represents quantities with large geographical dispersion and vast size. As a result this seems to be a safe assumption enabling us to generate hourly values. Another simplification is that all of the variables mentioned are considered known. However, we do not know any the future values and the values should have been estimated. The implementation of a prediction model for any of the variables is however a project in itself. However, as will be discussed in the next section, not all factors in the model will necessarily be utilized.

The length of the training data period is also consider to be essential, a length which greatly influence the procedure (and probably value) of estimating variables. The motivation of considering different length in the training period comes from Figure 2.3.1, where

---

[2]For each forecast there is parameters for each of the 6 days and each their 24 hour.

the power prices changes drastically over time, thus indicating that the vales of the explanatory variables themselves vary much. The length of the period is sometimes called the history and written $\mathcal{H}$, something which is taken from Section 4.1.2. This notation is useful when we are to study the length of the training period. The training data period is given in number of days. The minimal theoretical training data period is 8 days, as this is the time frame needed for the deterministic model to make a forecasts. As we will however see later, the length of the training period will seldom be less than 50 days.

When the SUR model calculates the estimates of the parameters a number of sub-procedures are required. Below a brief explanations is given. For more details the reader is referred to the Appendix, where the source code of the program which does the calculations is found. Thus, the SUR regression consists of the following steps:

i. Deterministic forecasts are made for the training period. The deterministic forecast is described in Equation (2.4.2.)

ii. Error terms between the forecast of (i) and observed prices are calculated.

iii. Error terms belonging to hours with unbalanced prices are filtered out, using the filtration methods described in Section 2.3.

iv. An ordinary least square (OLS) regression of each of Equation (5.2.2) is fitted on the error terms of (iii). Accordingly, a separate model is fitted to each of the hours.

v. The residuals of the OLS in (iv) are used to estimate an empirical variance matrix.

vi. The variance matrix of (v) is used to fit an generalized least square (GLS) model to the data. As with the OLS model an empirical covariance matrix of the GLS residuals estimated. The GLS procedure is described in Section 4.1.1.

vii. The new variance matrix is used as an input to another GLS fitting of the data.

viii. Steps (vi) and (vii) are repeated until all the values for the estimates of the explanatory variables have converged.

In the procedure of finding an estimate for the explanatory variable described above, the OLS and GLS estimators of Section 2.3 are used. In that section the parameters T, K and M are used to denote the dimensions of different matrices and vectors. The parameter M is the number number of response variables, which in our case is 240, as we have one response variable per hour of the forecast horizon. The parameter T which is the number of observations of a single hour depends on the training period chosen. Thus, the T is the length of the training period $\mathcal{H}$, but we also have to subtracts the number of unbalanced error terms which is filtered out of the training set as they are to be explained by the other component of the mixture model. The number K is the number of explanatory variables which is nine when the full set of parameters are used.

The number of explanatory variables may of course vary. The full set of explanatory variables is described in Equation (5.2.2). The model which is reduced to the maximum is

defined to consist only of the intercept and the bias correction term of yesterday $\Phi_v(t, h)$. It is however important to note that even though we specify a set of explanatory variable, all of the variables in the set may not necessarily be estimated. This can come from the fact that the an observations of the explanatory variable may not exist in the specified training period. The explanatory variables of daylight savings and public holiday are this something that happens frequently.

5.4 Subset Selection for the Unbalanced Model

As mentioned in Section 4.1.1 the SUR model opens for the possibility of different explanatory variables for the different equations through the use of a restricted model (SURR). Thus, the number of explanatory variables can differ for each of the 240 equations, which are found in the forecast horizon. The deterministic model as described in Equation (5.2.2) will be considered the bench mark in the evaluation.

As the forecasts consist of ensembles sampled from the multidimensional Gaussian distribution specified in Equation (5.2.1) the probabilistic evaluation methods of Section 4.2 are used. The main evaluation criterion is the Energy score which is described in Section 4.2.5. The energy score is used as criterion in the best subset selection algorithm (Hastie et al., 2008) to select the best set of explanatory variables for the SURR model. The best subset algorithm chooses, as the name suggest, the best set of explanatory variables according to a given evaluation criterion. In our case this criterion is the energy score. In the stepwise form the algorithm always changes the number of explanatory variables by one, either by excluding or including an explanatory variable iteratively into the current set of explanatory variables. The procedure is ended when either no explanatory variable can be added to positively influence the score (forward selection) or nothing is gained by excluding any of the explanatory variables (backward selection).

The number of possible combination of explanatory variables SURR models is however quite large. The total number of combinations of the 7 "free" explanatory variables[3] in 240 equations is 2^{1680}. Even if we restrict ourselves to the same estimation model for all of the 10 leading days we still have 2^{168} combinations, which is a staggering number of (possible) models. As a result we restrict yourself to consider a single estimation model for each of the 10 leading days. In addition we split the hours of the day into 5 groups. These groups are: Hour 0-6, 7-10, 11-15, 16-18 and 19-23. This reduces the number of possible combinations substantially[4]. The reason for reducing the number of combinations of explanatory variable is to make it possible for the algorithm to examine every combination if it should find it to be necessary. Thus, another division of hours into groups could be made. The present choice is motivated by the fitting of a SUR model to power prices by Klæboe (2011). In this paper the coefficients for these hours tended to coincide. We therefore conclude that these hours have the same dynamic and accordingly that they need the same set of explanatory variables.

---

[3]The intercept and the bias correcting term are always present as explanatory variables.

[4]The number of candidates is now reduced to 2^{35}.

Chapter 6

Results

In this chapter we present the result of the probabilistic prediction model proposed in Chapter 5 to forecast the spot prices. The power prices used are from the Nord Pool market, which was discussed discussed in Chapter 2. The predictions presented are for different time intervals of 2008, length in training period and combinations of explanatory variables. Moreover is evaluations of the predictions carried out with methods based on both point and probability methodology.

6.1 Forecasts

In this section we present the results of the forecasts for four different dates in 2008. The first day in our predictions are the 16th of April, June, August and October respectively. Thus, according to the notation introduced in Section 2.4, the variable v is the 15th of this respective months. As we work with a 10-day forecast horizon the electricity prices predicted are those of 16th to the 25th of the respective month. The periods are chosen to test the model for different price dynamics of the year. Moreover are three forecasts for each of the periods performed, with a respective length in the training period of 60, 80 and 120 days.

The forecasts of the power prices are products of the prediction model presented in Section 5.1. Accordingly a categorization of the error terms was executed, and only a model for the balanced terms was build. This model was based on seemingly unrelated regression (SUR) presented in Section 5.2. The inference on the parameters of the SUR model is given in detailed in the same section. The probabilistic prediction of the error terms that followed from this is a multidimensional Gaussion distribution. The figures shortly presented are based on 5000 samples of price realizations from this distribution. In in the figures we present the forecast, represented by the median as the forecast function. Furthermore are the values of the 95% prediction interval reported. These values are found by selecting the 4875th and 225th numbers of the ordered ensembles at each lead time. In addition are the interval of the vertical axis of the forecast figures kept unchanged to facilitate comparison between the figures. On the basis of these figures we do the following

observations:

1. In Figure 6.1.1 the forecasts for 15th of April 2008 are found. The first striking feature is that many of the prices in the forecast horizon would have been classified as unbalanced had they been in the training sample. This is observed by the number of points. Second, we notice that all of the forecasts seem to capture the daily dynamics, by this we mean that the predicted prices capture the characteristic daily pattern. Thirdly one can see that the size of the prediction intervals are effected by the time of day and that it widens with increasing lead time. For instance can this be observed in Figure 6.1.1a where the width of the prediction interval is approximately 40€ around noon and 15€ in the evening. However, in the same figure we observe that the prediction interval increases, and that its value on the 10th lead day ranges from 0 to 90€. Moreover, the prediction interval the 10th lead day changes less through the day than that of the first lead day. Finally, we observe that a decrease in the length of the training period seems to increase the performance of the forecast and to decrease the size of the prediction interval at the 10th lead day. The improved prediction ability can especially be seen on the first day where the observed prices for the forecast with a 120 days training period sometimes falls outside the prediction interval, while it is well predicted by both the forecast with a 60 and 80 days training period. For these predictions do the observed prices fall well inside the prediction interval. It is also observed that the prediction interval for the forecast with a 120 days training period has much wider prediction interval on the 10th lead day than the forecast with a 60 days training period.

2. In Figure 6.1.2 the forecasts for the 15th of June 2008 are found. The most striking feature for this time period is that the unbalanced prices are fewer but more dominating than the ones seen in Figure 6.1.1. An example of this is the unbalanced prices which occur around 170 and 200 hours into the forecast, where big drops make the price fall towards zero. Moreover, these unbalanced prices are producing big dips in the prices. Dips in prices during the summer are to be expected, since precipitation and low consumption during the early morning hours make the power prices fall drastically. However, when it comes to the prediction of prices we observe that these display a flatter structure than the observed prices. Apart from this, we notice that the prediction interval for these prediction also vary within day and increases with lead time. It is however difficult to spot any improvement in the predictions of the prices as the length in the training period increases, while an improvement in the size of the prediction interval is clear.

3. In Figure 6.1.3 the forecasts for the 15th of June 2008 are found. The observed prices have a completely different pattern than the prices seen in Figure 6.1.1 and 6.1.2, namely that the two characteristic daily spikes described in Section 2.3 are very much smaller and sometimes even completely absent. Further is the number of observed prices that would have been characterized as unbalanced prices decreased drastically. Turning to the predictions of the prices we see that they are able to capture the prices

quite well. Furthermore, the prediction intervals are smaller than those previously seen for April and June. However, the change in size of the prediction interval for different hours within a day are not that great for these forecasts. Further is the decrease in size for the prediction interval as the length in training period increases not that apparent as in April and June.

4. In Figure 6.1.4 the forecasts for the 15th October 2008 are found. In comparison to the observed prices of the other months, the prices that are observed in this month have the smallest variation in the daily prices and the fewest number of prices that would have been classified as unbalanced had they been in the training data. When it comes to the predictions of power prices we see an increase in accuracy as the length of the training period decrease. However, the precision of the predictions, as observed through their prediction interval, seem not to improve drastically as the number of training days decreases.

Thus, by looking at the observed and predicted prices for different months in 2008 we see that their dynamics seem to change.

Turning our attention away from the predicted prices we look at the correlation structure of the residuals of the probabilistic forecast. In Section 2.6 we inspected the correlation in the error terms of the deterministic model in in Figure 2.6.1 and 2.6.2 are revisited in Figure 6.1.5d. To facilitate comparison a new plot have been made where the range of the values has been altered to facilitate comparison with the correlation of the residuals. Thus, the correlation structure given in Figure 6.1.5a, where the within-day correlation of the error terms was given, is the same as the one found in Figure 2.6.1b, but a changed range. By comparing these figures we can deduce that the within-day correlation is reduced drastically. The only intervals of the day that seem to uphold correlation in the residuals are the groups of hours 1-5, 9-16 and 20-22. For the between-day correlation found in Figure 2.6.2b we saw strong signs of pattern, while these patterns seem to have vanished in Figure 6.1.5c, which is the correlation after the probabilistic model was applied. From these plot we conclude that the proposed SUR model for the balanced error terms seems to capture the within-day correlation through it use of 24-hour prediction blocks. Additionally, the between-day correlation seems also to have been captured, something which may be attributed to the implementation of the bias correcting terms, which connect the same hour between days.

The values of the coefficients for SUR model of the forecast of October 15th 2008 with a 60 days training period are found in Figure 6.1.6. In the figure we notice that values for three of the explanatory variables changes drastically through the 240-hours (10-days) forecast horizon, namely the intercept, reservoir level and month 2. The month 2 variable is in this case September. While the confidence interval of the values for the reservoir level and September appears to be constant, the confidence interval of the intercept increases with lead time. Further, the values of the coefficient for the consumption and bias correcting term are constantly changing, but do not have any clear trend. The coefficients for the temperature, August, October and Thursday increase with lead time, while the values of Tuesday and Sunday decreases with increasing lead time. The values of the remaining

coefficients: Monday, Wednesday, Saturday and summer time are more or less constant. Many of the confidence intervals are also constant. The exceptions are the confidence interval for the coefficients of Friday, Saturday and the temperature which experience an increase in size during the daytime hours. At the same time do the confidence intervals for the September coefficients appear to increase linearly with lead time.

From Figure 6.1.6 it appears like some of the coefficients have the same dynamics, while others do not. Therefore we have plotted of the correlation of the coefficients of the explanatory variables of the SUR model in Figure 6.1.5d. The model was constructed on the basis of a full SUR model with 2007 and 2008 as training data. The large sample of training data was chosen so we can assess a model where all the explanatory variables are present. From the figure we observe that the intercept is negatively correlated with nearly all of the explanatory variables which take their origin in the calender, that is to say the months and the weekdays. Further is the holidays negatively correlated with several of the months from April to October. moreover, a correlation between the holidays and the months April and May is to expect when a number of (Norwegian) holidays take places at that time. However, the correlation between the holidays and the other months is somewhat unclear as none holidays occur in those months. This may on the other side be a side effect that not really originates from the holiday month relationship, but from the correlation among the months. This is because the months themselves have a positive correlation. Therefore, a lack in correlation of the extra (moving) holidays in some of the months are something that may have been redistributed to the holiday variable. Moreover, we observe a weak negative correlation between the reservoir level and the months. This effect comes naturally as the level together with the rate of change of the reservoir follows a yearly pattern. Finally, we see that the weekdays are positively correlated with each other.

6.2 Evaluation

In this section we first examine the point forecast methods of the deterministic model and the probabilistic forecast. The evaluation of the probabilistic forecast is performed on each of the four same time periods as of the predictions that were presented in the last section. To recapitulate these predictions were the 5th to the 25th of April, June, August and October 2008. Thereafter the energy score for the same deterministic model and probabilistic forecast is presented. Then we study the Talagrand histogram to look for the fit of the predicted distribution with regards to the distribution of the observed value. Finally, the best-subset algorithm, as described in detail in Section 5.4, is used to find the best set of explanatory variables of the SUR model.

In Table 6.2.1 the point evaluation of the deterministic model for the four periods is presented. Before the forecasts, the appropriate forecast function were evaluated as described in Table 4.2.2, was applied on the forecast. We observe that the point evaluations do not give a single period for which the deterministic model gave the best prediction. This stems from the fact that the deterministic model performs best in August according
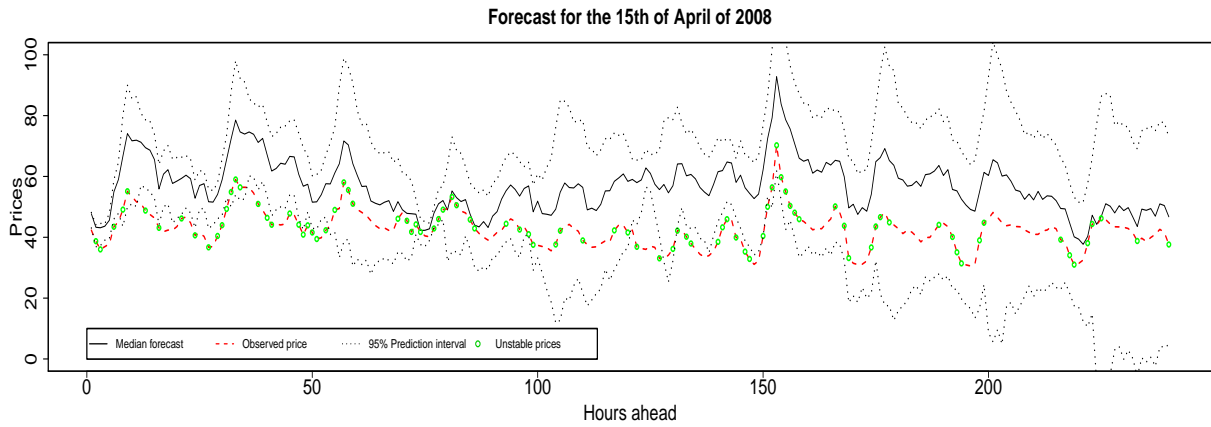
(a)
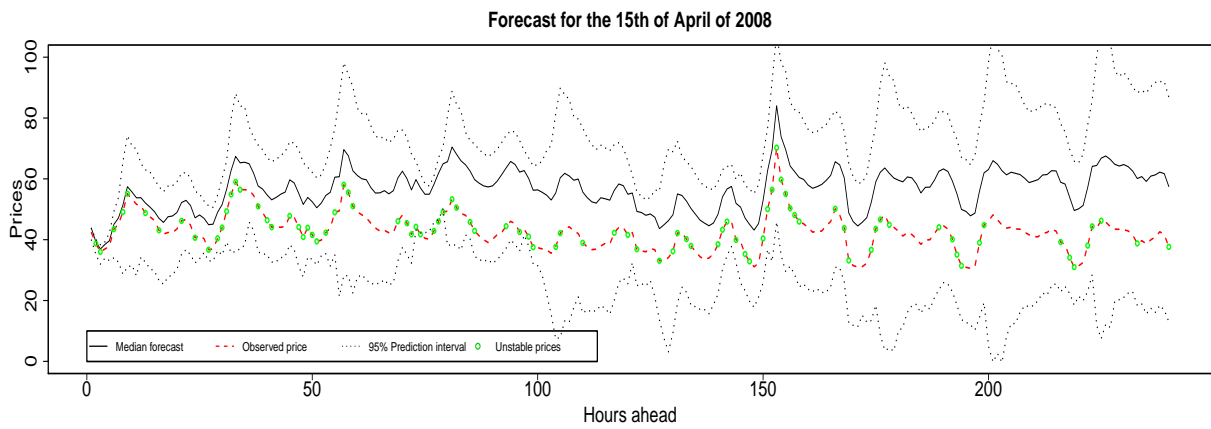
(b)
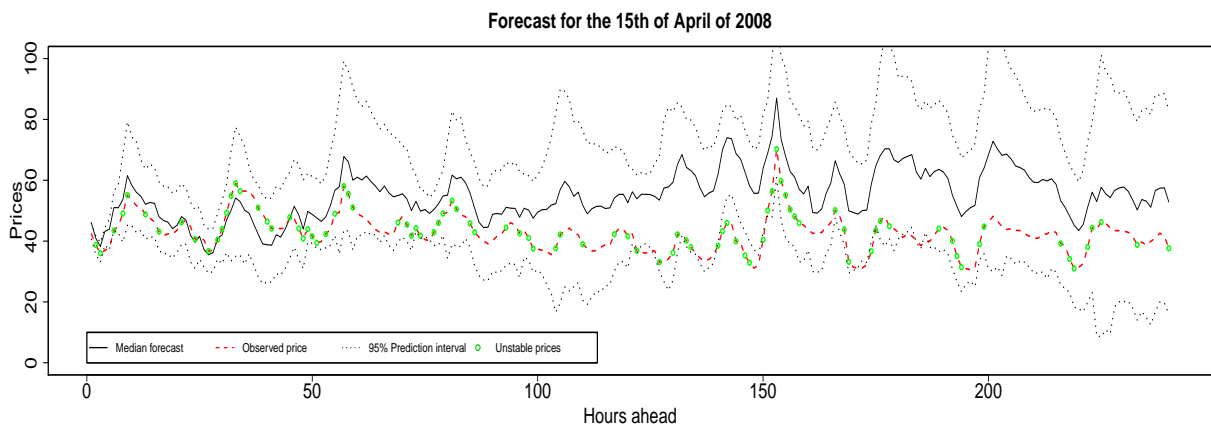
(c)

Figure 6.1.1: *Three 10 day-ahead forecasts from the 15th of April 2008. The length in training period is Panel a) 120 days, Panel b) 80 days and Panel c) 60 days. The explanatory variables of the model are as described in Table 6.2.4. The black line is the predictions values, the dashed red line is the observed prices while the dotted black line is the values of the prediction intervals. The green dots are the hours that would have have classified as spikes if they had been in the in the training sample.*

Forecast for the 15th of June of 2008

(a)

Forecast for the 15th of June of 2008

(b)

Forecast for the 15th of June of 2008

(c)
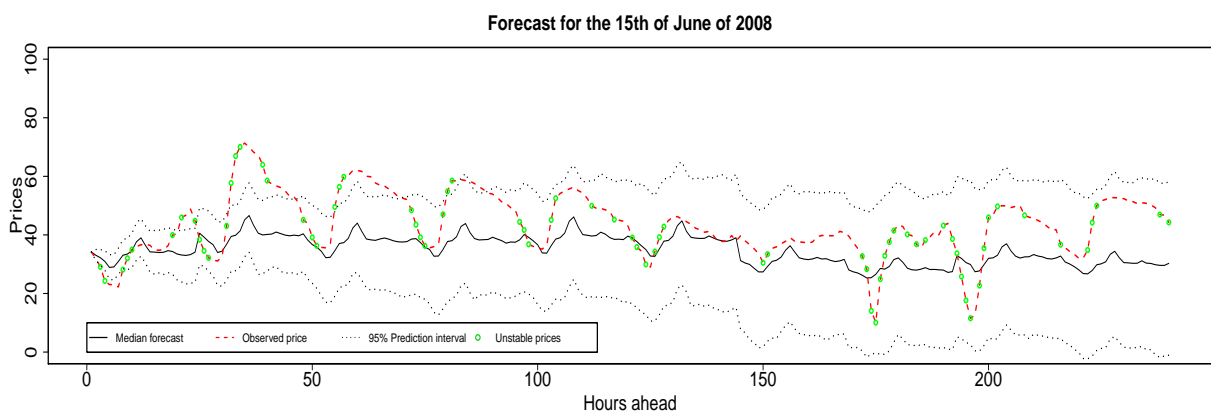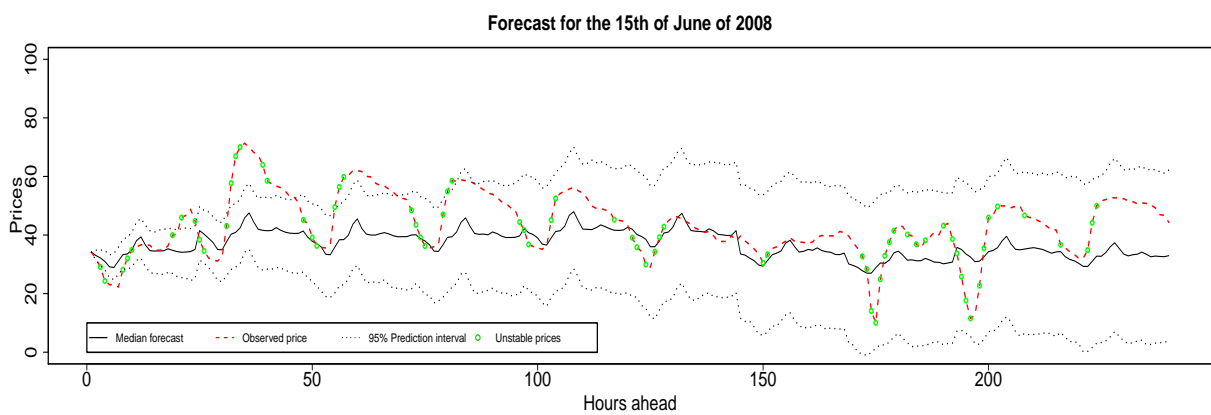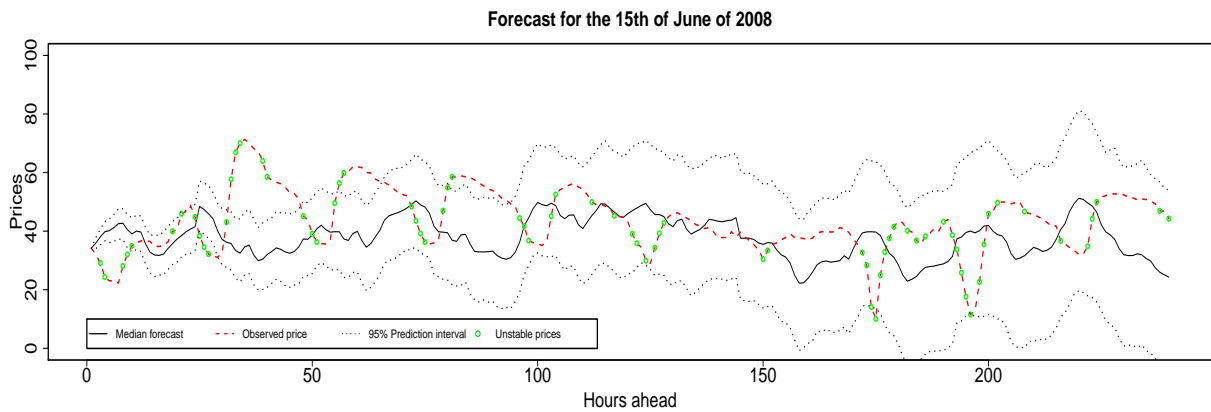
Figure 6.1.2: *Three 10 day-ahead forecasts from the 15th of June 2008. The length in training period is Panel a) 120 days, Panel b) 80 days and Panel c) 60 days. The explanatory variables of the model are as described in Table 6.2.4. The black line is the predictions values, the dashed red line is the observed prices while the dotted black line is the values of the prediction intervals. The green dots are the hours that would have have classified as spikes if they had been in the in the training sample.*
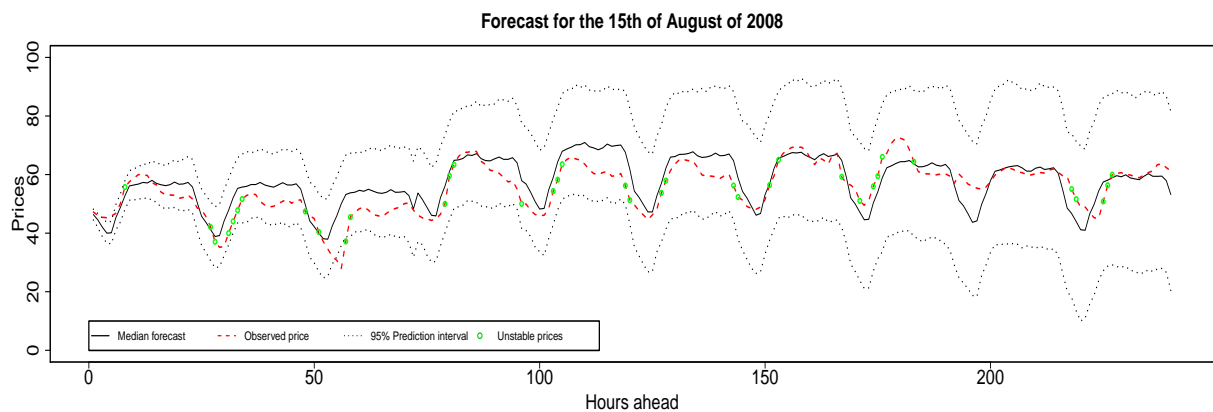
(a)
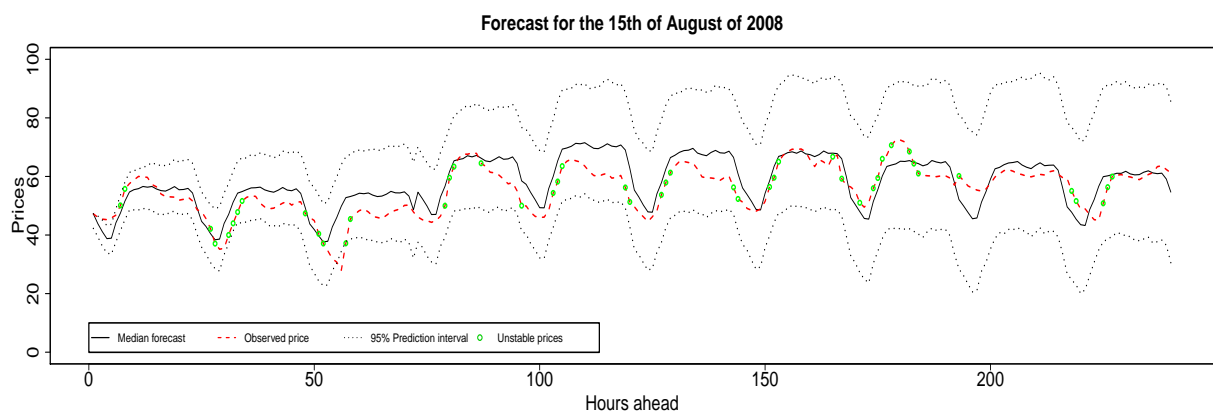
(b)
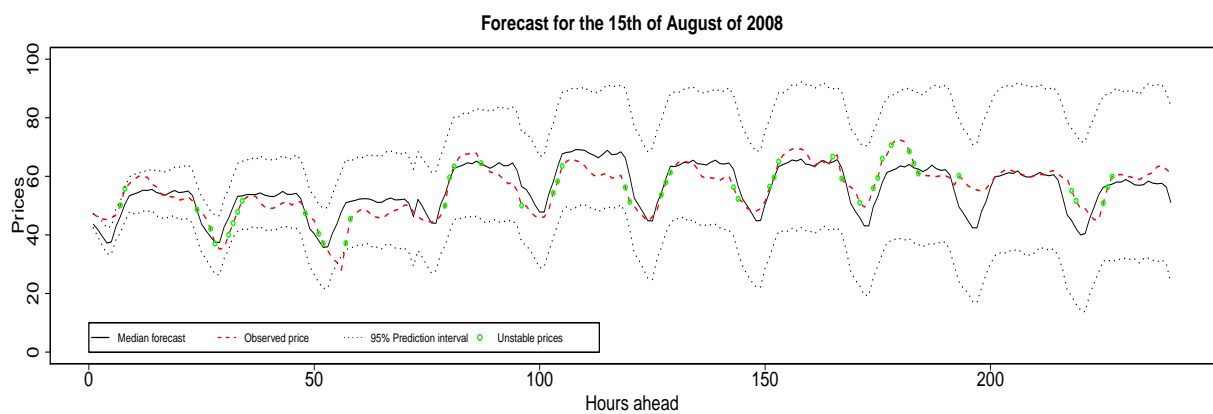
(c)

Figure 6.1.3: *Three 10 day-ahead forecasts from the 15th of August 2008. The length in training period is Panel a) 120 days, Panel b) 80 days and Panel c) 60 days. The explanatory variables of the model are as described in Table 6.2.4. The black line is the predictions values, the dashed red line is the observed prices while the dotted black line is the values of the prediction intervals. The green dots are the hours that would have have classified as spikes if they had been in the in the training sample.*

Forecast for the 15th of October of 2008

(a)

Forecast for the 15th of October of 2008

(b)

Forecast for the 15th of October of 2008
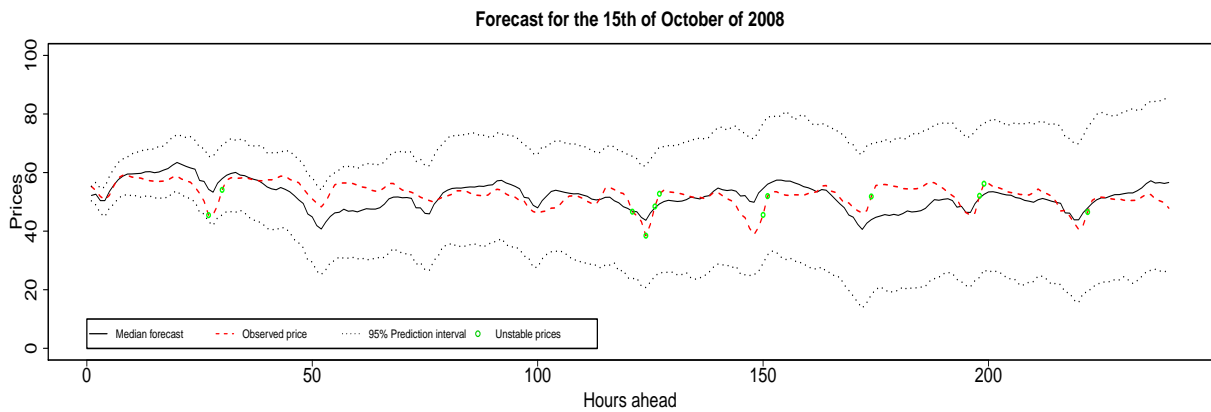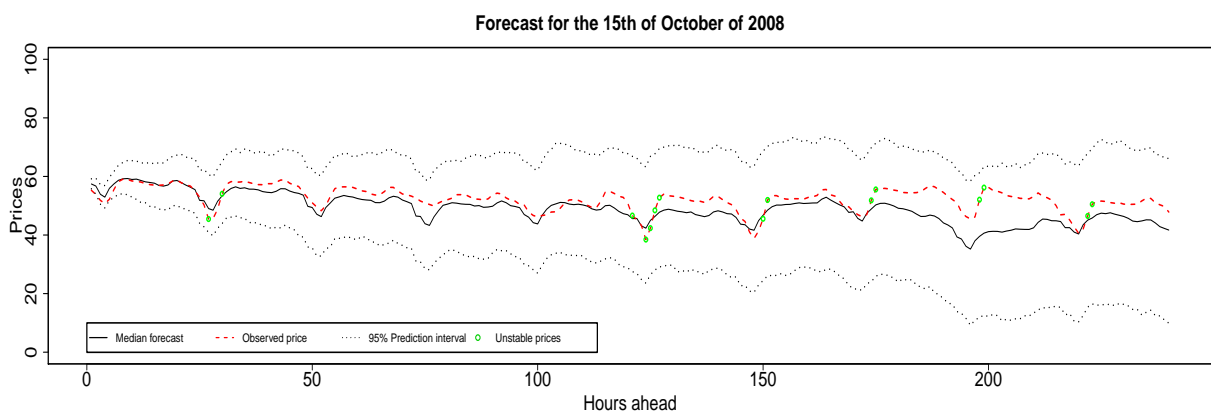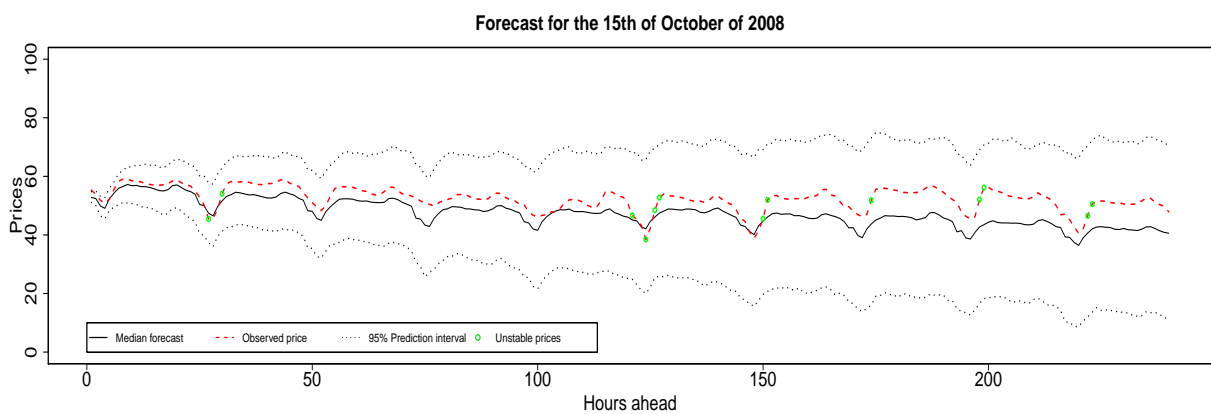
(c)

Figure 6.1.4: *Three 10 day-ahead forecasts from the 15th of October 2008. The length in training period is Panel a) 120 days, Panel b) 80 days and Panel c) 60 days. The explanatory variables of the model are as described in Table 6.2.4. The black line is the predictions values, the dashed red line is the observed prices while the dotted black line is the values of the prediction intervals. The green dots are the hours that would have have classified as spikes if they had been in the in the training sample.*
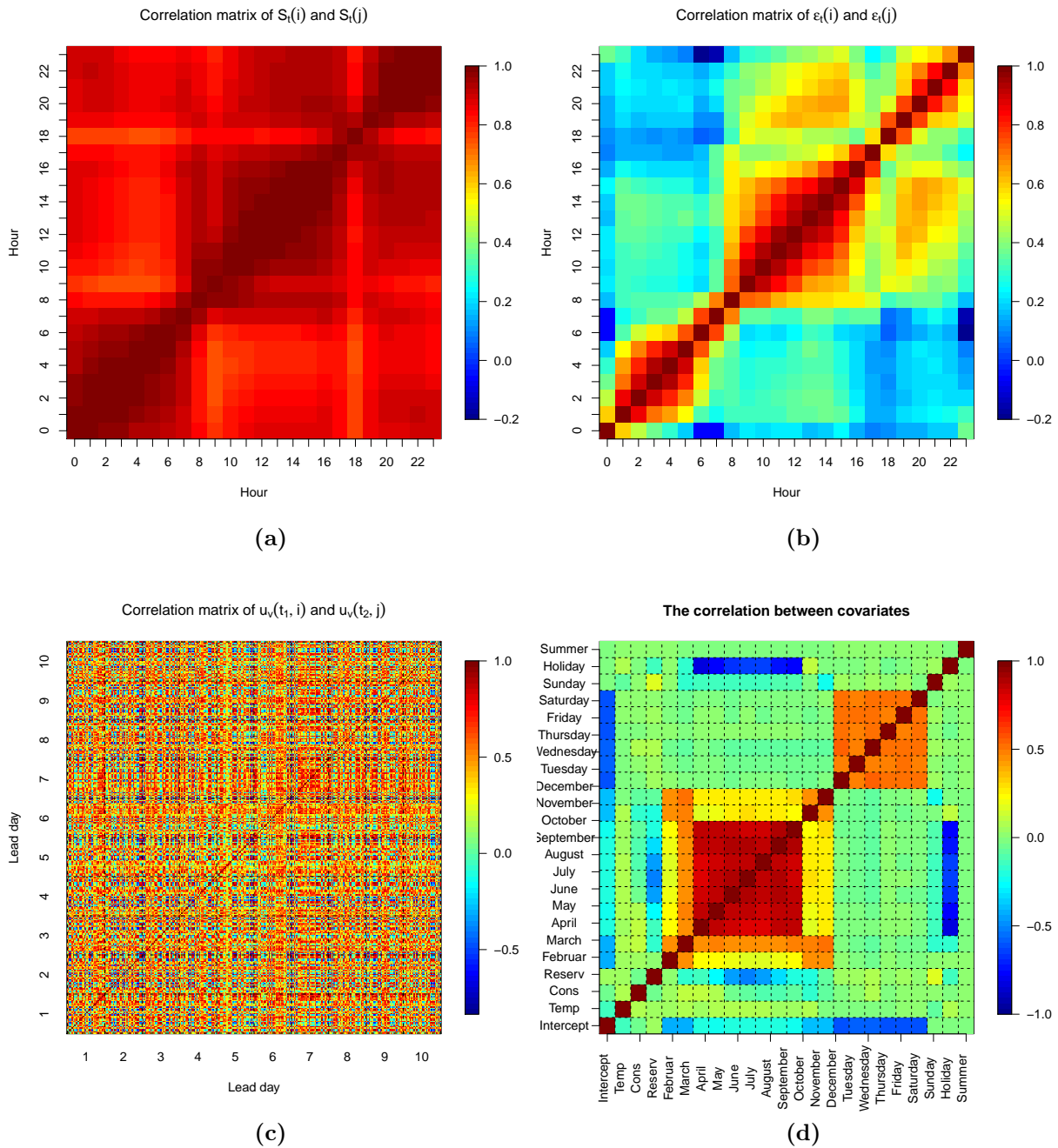
Figure 6.1.5: *In Panel a) is the correlation among hours for the error terms. This correlation is from both the unbalanced and balanced terms as given by Equation (2.4.1). In Panel b) is the correlation among hours of the residuals of both the deterministic and stochastic prediction, as found in Equation (5.2.2), plotted. In Panel c) we have the inter-day correlation among the residuals of the deterministic and probabilistic forecasts combined. In Panel d) is a plot of the correlation of the explanatory variables of the October forecast of 2008 with a 60 days training period. The value and confidence interval of the variables are found in Figure 6.1.6.*
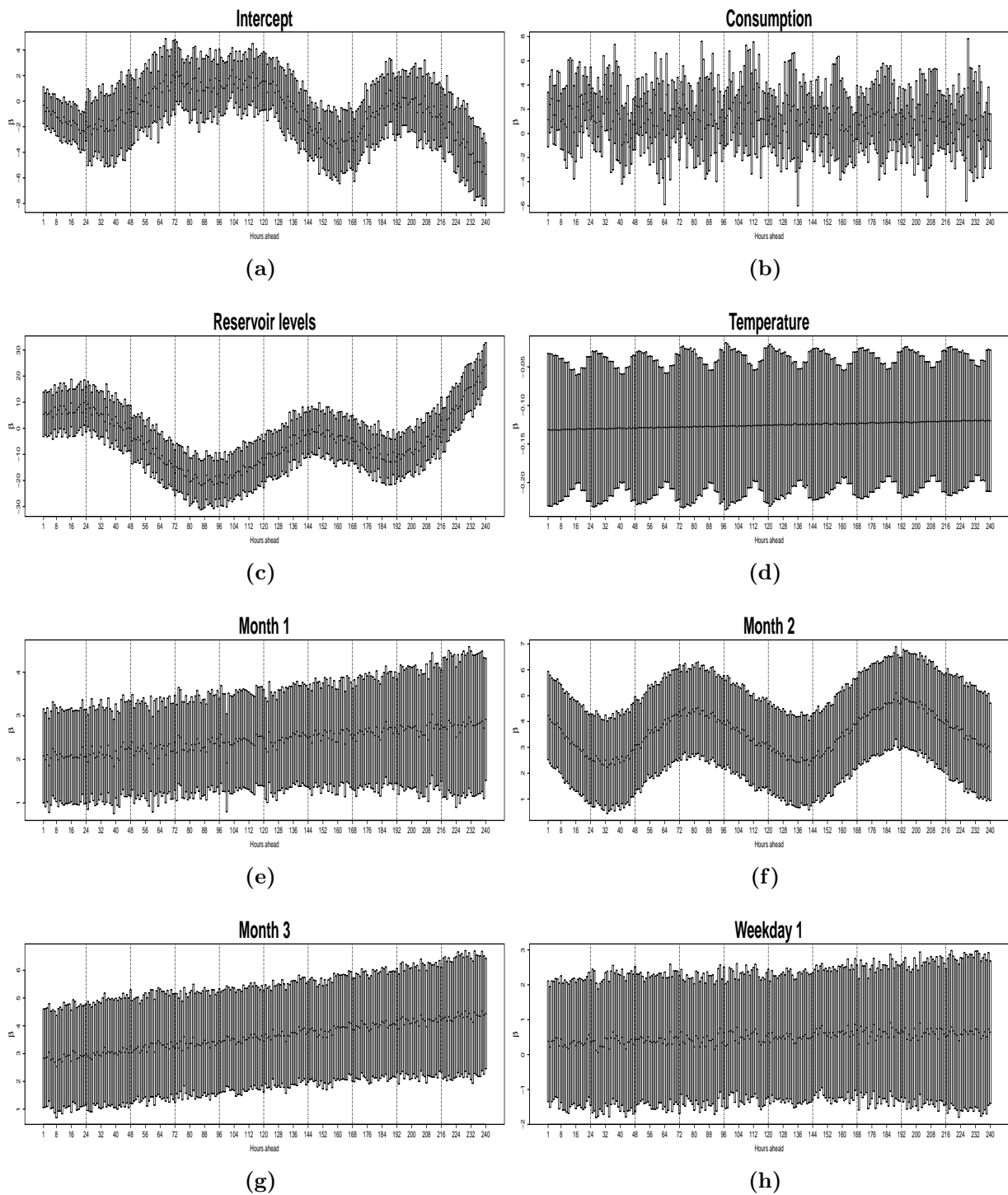
Figure 6.1.6: *This figure continues on the next page. The estimated parameters of the full SUR model for the forecast of the 15th to 25th of October 2008 with a 60 days training period. The parameters are plotted with their respective confidence intervals. In Panel a) through o) we have the following parameters: The intercept, consumption, reservoir level, temperature, October, September, August, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday, summer time and bias correction. The y-axis are not the same for every panel.*
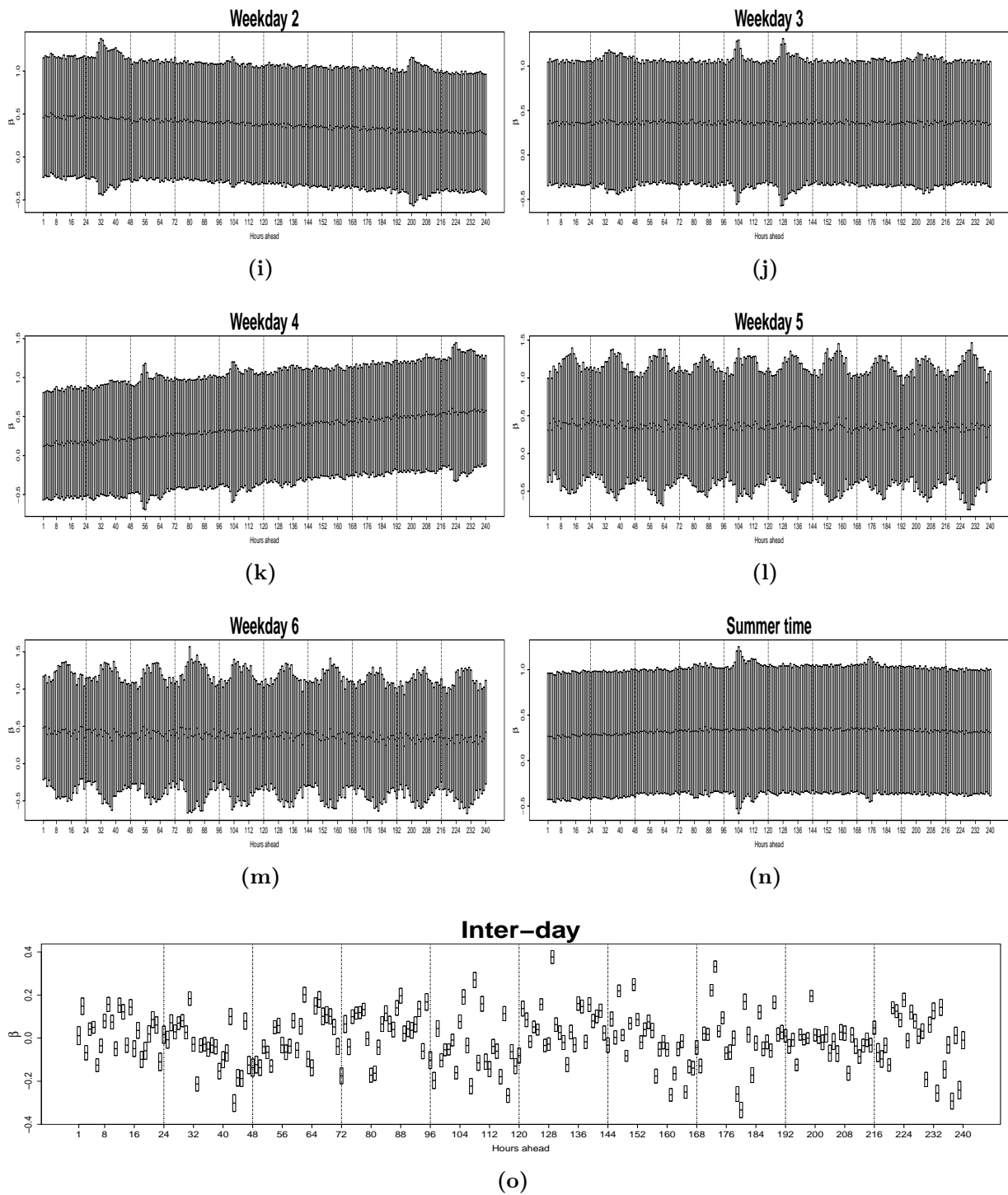
Figure 6.1.6: *This figure is a continuation of the figure on the previous page. The estimated parameters of the full SUR model for the forecast of the 15th to 25th of October 2008 with a 60 days training period. The parameters are plotted with their respective confidence intervals. In Panel a) through o) we have the following parameters: The intercept, consumption, reservoir level, temperature, October, September, August, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday, summer time and bias correction. The y-axis are not the same for every panel.*

to the mean square error (MSE) and the mean absolute error (MAE), while it performs best for October according to the mean absolute percentage error (MAPE) and the mean relative error (MRE).

If we compare the point evaluations of the deterministic model in Table 6.2.1 to those of the probabilistic model which are found in Table 6.2.3, we see that the probabilistic predictions always have the best performance. The point evaluations of the probabilistic predictions were done in the same manner as of the deterministic model. However, for the probabilistic forecast we have also considered different lengths in the training period. The different lengths in the period which are presented in Table 6.2.3 is 60, 80 and 120 days. For the 60 days training period the SUR model performs best in August, and the same is the case for the 120 days training period. The SUR model does however perform best for October, when the training period is set to 80 days.

The probabilistic evaluation criterion of the energy forecast is given for the deterministic model and probabilistic forecast in Table 6.2.2. We do however observe that the energy score of the deterministic model is the same as the mean absolute errors that were presented in Table 6.2.1. This fact was pointed out already in the theoretical walk through of the evaluation method in Section 4.2.3, but it is good to see that the theoretical result holds in practice. Another thing to notice in Table 6.2.2 is that the energy score evaluates the probabilistic forecast to be best always in October regardless of the length of the training period. This is in contrast to the point evaluation methods, which evaluated the probabilistic forecast to be best in August for the 60 and 120 days training period.

In addition to giving the values of the energy score in Table 6.2.2, a plot of the score for more lengths in the training period for the four periods can be found in Figure 6.2.1. From this figure we realize that an even better probabilistic forecast for April can be constructed by a 50 days training period. Moreover, we notice that the other periods have a minimum in energy score for a training period between 60 and 80 training days. The month of October is perhaps an exception, with an always a very small decrease in the score, but the energy score for this month as a function of the training period is very flat below the 110 days training period. What is maybe more important is that the energy score of the mean of the periods evaluates the best length in training period for the probabilistic forecast at all the periods to be 80 days.

Changing the focus away from the evaluations by scoring functions we look at the Talagrand histogram. This histogram is used to evaluate the calibration of the prediction, something which was explained in the theoretical walk through of the multivariate Talagrand histogram was given in Section 4.2.4. The Talagrand histogram presented later were found by drawing 1000 simulation of each of the probabilistic predictions, which were performed of the same periods of 2008 as earlier. In Figure 6.2.2 the histogram for the deterministic model and the probabilistic forecast are presented. We see that the deterministic model displays under-dispersion as it has a smiling shape. Further is the upper half of the rank observations greater than those of the lower ranks, something which implies a bias. However, this may also be a result of the under-dispersion. The same findings manifest themselves in the Talagrand histogram of the probabilistic forecast. On the other hand are the under-dispersion less in this histogram, something which points to a more calibrated

Month	MSE	MAE	MAPE	MRE
April	871.7	16.7	0.98	0.63
June	263.0	20.3	0.93	0.50
August	**41.84**	**12.1**	0.86	0.40
October	87.63	13.6	**0.71**	**0.37**
Mean	316.0	15.7	0.87	0.48

Table 6.2.1: *The table presents the point scoring functions with the use of their optimum forecast functions for the deterministic forecast as stated in Equation (2.4.2).The scoring functions are mean square error (MSE), mean absolute error (MAE), mean absolute percentage error (MAPE) and mean relative error (MRE). The scoring functions are each calculated for four different periods of the year. Each period starts the 5th and ends the 25th of April, June, August and October respectively. The emphasized values are the minimum values of their respective columns. The mean score is reported in the bottom row.*

model. In addition to these histograms are also the histograms of each of the periods April, June, August and October of the stochastic model presented in Figure 6.2.2c to 6.2.2f separately. These histograms may be regarded as a decomposition of the histogram for the histogram of Figure 6.2.2b, since this histogram is the mean of the histograms displayed in Figure 6.2.2c to 6.2.2f. We observe that the histogram for April is heavily under-dispersed, while the histogram of October seems almost calibrated.

To find the optimal set of explanatory variables of the SUR model, and thereby a better probabilistic forecast, we performed the backward leave-one-out best-subset-selection algorithm. The method found that for the 60 days training period only the probabilistic forecast of August was improved by excluding parameters. However, for a training period of 80 days were the SUR model for all the methods improved by excluding parameters. Further was it only August which is not improved for a 120 days training period by excluding parameters. The point and energy scores of the reduced SUR models are found in Table 6.2.4, where also the model which did not get reduced are marked by an asterisk. Moreover is the explanatory variables that were excluded from the SUR model, together with an indicator for which hour they were excluded from, reported in Table 6.2.5. However, as the previously discussed in Section 5.4, the best-subset-selection algorithm is not allowed to choose any combinations of explanatory variables for all the 240 equations. From Table 6.2.5 we see that the reservoir level and the consumption are excluded approximately the same amount of times. However, we notice that they only are excluded one time together. Other explanatory variables that are excluded from the SUR model are the temperature, weekdays and daylight savings.

Month	60	80	120	$D_v(t,h)$
April	8.71	8.09	9.64	16.7
June	14.0	9.35	9.63	20.3
August	10.2	8.23	7.82	12.1
October	**8.28**	**7.08**	**7.06**	**13.6**
Mean	10.3	8.19	8.54	15.8

Table 6.2.2: *The energy score of the probabilistic price forecast including the unbalanced error term with all explanatory variables in the regression model. In the last column is also the energy score of the the deterministic forecast ($D_v(t,h)$) found, which we observe is the same as its mean absolute error. The reason for this is given in Section 4.2.3 and in the Appendix. The energy score of the probabilistic forecast for different length in training period (in days) are presented in the three first columns. The scoring function are calculated for four different periods of the year. Each period starts the 5th and ends the 25th of April, June, August and October respectively. The emphasized values are the minimum values of their respective columns. The mean score is reported in the bottom row.*
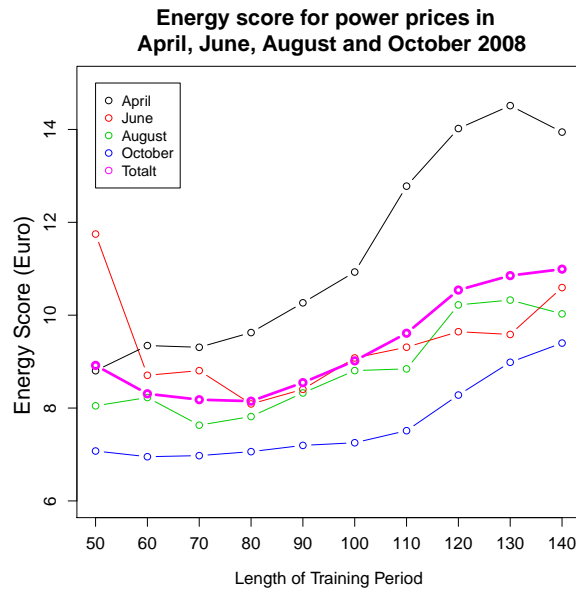
Figure 6.2.1: *The Energy score as a function of the length of the training period for the SUR model with a full set of explanatory variables of the unbalanced error terms. The Energy score is calculated for different length in training period for four periods. These periods are from the 5th to the 25th of April, June, August and October respectively. In addition is the mean of the energy score for these four periods plotted.*

TP	60				80				120			
$S(\mathbf{x},\mathbf{y})$	MSE	MAE	MAPE	MRE	MSE	MAE	MAPE	MRE	MSE	MAE	MAPE	MRE
April	683.8	16.4	0.96	0.62	719	18.3	0.88	0.73	883.8	22.4	0.94	0.70
June	181.7	15.3	0.84	0.48	180.8	17.6	0.76	0.45	120.0	16.1	0.34	0.22
August	**29.97**	**7.12**	**0.61**	**0.02**	32.89	8.56	0.56	0.17	**25.39**	**7.01**	0.13	**0.05**
October	32.34	11.7	0.71	**0.02**	**19.88**	**7.02**	**0.46**	**0.14**	29.11	8.82	**0.07**	0.09
Mean	231.0	12.6	0.78	0.28	238.2	12.9	0.67	0.37	264.6	13.6	0.37	0.27

Table 6.2.3: *The values of the score functions, $S(\mathbf{x},\mathbf{y})$, where the full set of explanatory variables in the SUR model for the unbalanced error terms of Equation (5.2.2). The scoring functions are the mean square error (MSE), mean absolute error (MAE), mean absolute percentage error (MAPE) and mean relative error (MRE). The SUR model is considered for different months of 2008, namely April, June, August and October. Moreover are the scoring functions calculated for the SUR model using different lengths in the training period (TP). The lengths of the training periods in the table are given as days. The scoring functions use the appropriate forecast functions as described in Table 4.2.2. The emphasized values are the minimum values of their respective columns. The mean score is reported in the bottom row.*

TP	60					80					120				
$S(\mathbf{x},\mathbf{y})$	ES	MSE	MAE	MAPE	MRE	ES	MSE	MAE	MAPE	MRE	ES	MSE	MAE	MAPE	MRE
April	*	*	*	*	*	7.76	21.55	10.0	0.10	0.03	9.98	14.61	9.67	0.15	0.04
June	*	*	*	*	*	8.04	26.50	8.03	0.17	0.04	10.1	14.86	11.9	0.18	0.04
August	**7.47**	**24.51**	**5.34**	**0.09**	**0.02**	7.48	17.43	9.05	0.09	0.03	*	*	*	*	*
October	*	*	*	*	*	**6.51**	**15.93**	**4.04**	0.3	**0.02**	**7.49**	**10.80**	7.18	**0.04**	**0.02**
Mean	9.62	230.6	12.2	0.65	0.28	7.44	20.35	7.78	0.09	0.03	8.84	17.42	35.7	0.12	0.04

Table 6.2.4: *The values of the score functions, $S(\mathbf{x},\mathbf{y})$. As opposed to Table 6.2.3, where the full set of explanatory variables in the SUR model for the unbalanced error terms of Equation (5.2.2), this table give scores of the best set of explanatory variables found, rather than the full model. The best set of explanatory variables are found by backward leave-one-out best-subset algorithm, with the energy score as evaluation criterion. The baseline score, which can be found in Table 6.2.3, is however the same. Cells marked with a star have the full model as the best model of choice. The scoring functions presented are the energy score (ES), mean square error (MSE), mean absolute error (MAE), mean absolute percentage error (MAPE) and mean relative error (MRE). The SUR model is considered for different months of 2008, namely April, June, August and October. Moreover are the scoring functions calculated for the SUR model using different length in the training period (TP). The length of the training periods in the table are given as days. The scoring functions use the appropriate forecast functions as described in Table 4.2.2. The emphasized values are the minimum values of their respective columns. The mean score is reported in the bottom row.*
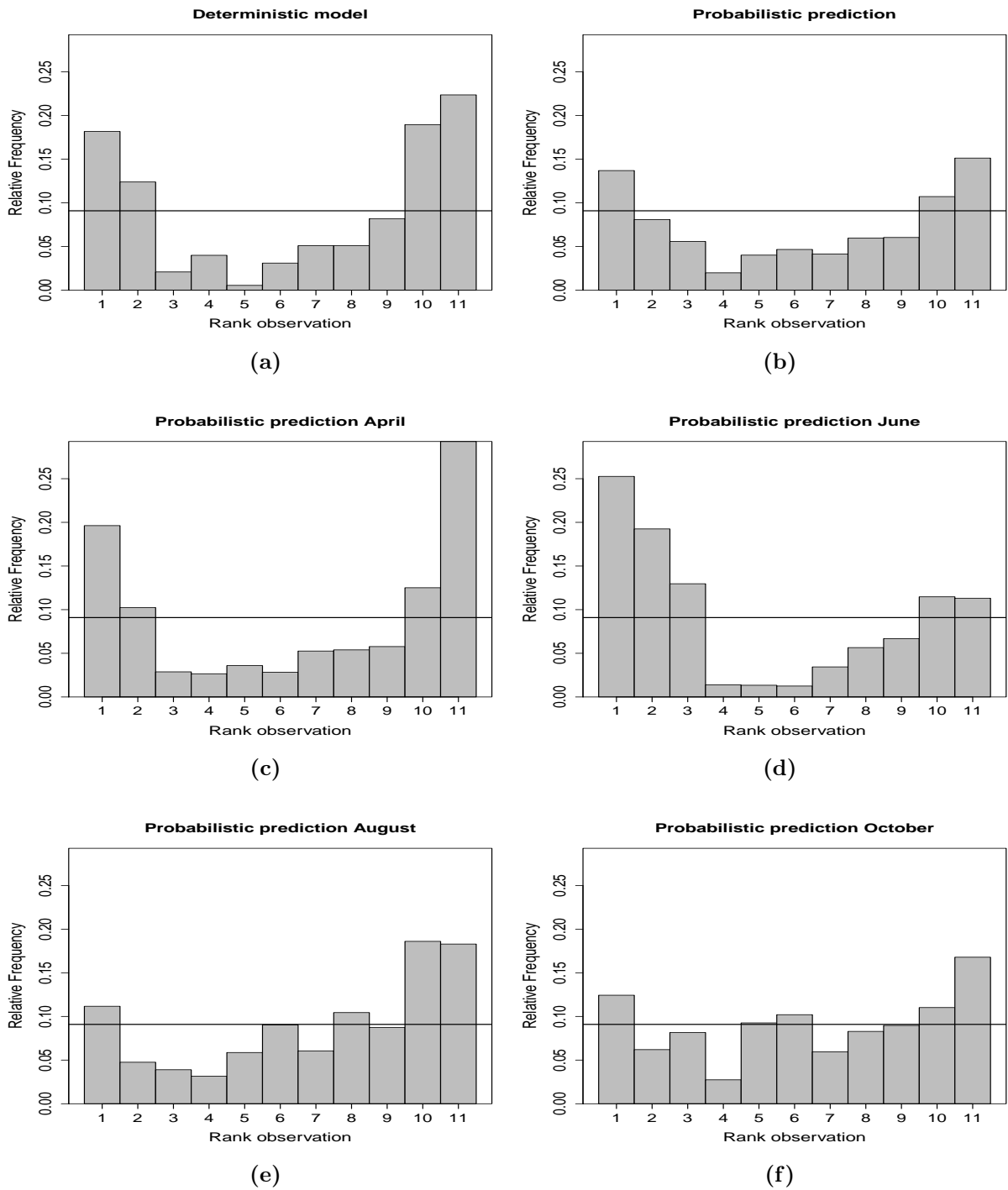
Figure 6.2.2: *In Panel a) through f) are the Talagrand histogram of the deterministic , stochastic, April, June, August and October forecast. The Talagrand histogram displays the calibration of the forecast. The horizontal solid line indicated the level of the bars in a calibrated histogram The histograms of each of the months is calculated from 20 forecasts each (from the 5th to the 25th each month) and tallied over 1000 simulations in each of the 20 forecasts. The Talagrand histogram of the stochastic forecast is a mean of the four months April, June, August and October. The Talagrand histogram is calculated on the basis of forecasts from the same period.*

TP	60	80	120
April	*	$q(1,3,4,5)$	$T + q(1,2,3,4,5)$
June	*	$R(1,5)$	$R(1,5)+w(2,3,4)$
August	$q(1,4,5)+R(1,2,3,4,5)$	$R(1,2,3,4,5)$	*
October	*	$R(1,2,3,4,5)+q(1)$	$q(1,2,3)+m(1,2,3,4,5)$

Table 6.2.5: *The rows display the explanatory variables excluded in the SUR model of the balanced power prices, which first was evaluated by its full set of explanatory variables in Table 6.2.4. The exclusion of variables is done according to the backward leave-one-out best-subset-selection algorithm with the energy score as evaluation criterion. The numbers in the parenthesis are the groups, as outlined in the end of Section 5.4, which have the explanatory variables excluded. The numbers are groups of hours, where 1,2,3,4 and 5 are the hours from 0-6, 7-10, 11-15, 16-18 and 19-23 respectively. The SUR model is considered for different months of 2008, namely April, June, August and October. Moreover are the scoring function calculated for regression model using different lengths in training period (TP). Cells marked with an asterisk have none of the explanatory variables excluded.*

Chapter 7

Discussion

In this chapter a discussion about the results obtained in Chapter 6 is given. When analyzing the data in Chapter 2 we noticed an increase in variance for the spring months. This is something that influenced the optimal training period as seen in Figure 6.2.1. In this figure the best forecast for the spring month April were the one s with a short (50 days) training period. For the other forecast studied, in June, August and October, the optimum training period was not the shortest, being around 60-80 days.

By this we may conclude that the dynamic model used to forecast the prices are only interested in error terms displaying the same regime. This regime of error terms seem to change through the year. While we in April experience a trade of between number of data points and a fast changing dynamics in error terms, the month of October displays no such need for choice. This is because it has a relatively flat energy score function with respect to the training period, as seen in Figure 6.2.1. The month of June is penalized for both the short and long training period. However a training period of 80 days, which is the optimal length according to the energy score, should leave enough data points for a good prediction of the SUR model. The month of August finds itself somewhere between June and October, since it have a relatively flat energy score function with regards to the length in training period, but at the same time displays a local minimum in the function for a 70 days training period.

It is on the other hand not clear what would have been the optimal length of the training period had an model for the unbalanced power prices been included. The component in the mixture model that was to capture the unbalanced prices. As a result an introduction of this into the model could have left us with the same length in training period for all of the periods investigated.

Another factor we saw in Figure 6.2.2 was that the deterministic Talagrand histogram was greatly improved by the use of the probabilistic forecast. On reflection, though, it is not so surprising. There are several sources of uncertainty in the forecast of power prices, including uncertainty in the deterministic model, including uncertainty of the power grid, the price that competitors bid and last but not least the physical system that affects the prices. Most ensembles captures some of these uncertainties, and then probably only partially. Thus it seems inevitable that ensemble based purely on a deterministic model will be

under-dispersive to some extent. Because they do capture some of the important sources of uncertainty, however, it is reasonable to expect a positive spread.error correlation, even when the ensembles is uncalibrated. To obtain a calibrated forecast, therefore it seems necessary to introduce a model for the unbalanced error terms. In addition it may be necessary to include a single more accurate, or several, deterministic forecast in the use of statistical post-processing.

In Table 6.2.4 the optimal set of explanatory variables of a model were studied. We found that the reservoir level and consumption should be excluded in five of twelve cases. The explanatory variable for temperature, weekday and daylight savings should only be excluded one time each. All other explanatory variables were not to be excluded in any of the cases. The fact that the reservoir level and consumption are that often excluded might imply that the explanatory variables of explaining the calendar capture the same dynamic just as well. An other explanation might be that the consumption and reservoir level are mutually exclusion. This is something that is suggested by Table 6.2.4, as the two explanatory variables are rarely excluded simultaneously. (Something which only happens two times.) However, if this were the cases, that the consumption could be explained by the reservoir level, this is something that should have given a high correlation in Figure 6.1.5d. As this correlation fails to materialize we conclude that the consumption and reservoir level are so often excluded is that they have a high impact on the model. Therefore, when this impact simply is negative the explanatory variable is excluded. This might be the case why the reservoir level is excluded for a 80 days training period for June, August and October. In this training period the reservoir level starts to fill up. Thus, the model might be mislead by the sudden increase in reservoir level, which have a negative impact on the model.

Chapter 8

Conclusion

In this report we have made a stochastic model framework for the error terms generated by a deterministic model and the Nord Pool system power prices. First we did an exploratory analysis of the Nord Pool market, from which it was revealed that the system price had daily, weekly and yearly patterns together with a within and between day correlation. In addition, based on the large jumps in the power prices that were found in this analysis something which later lead us to introduce a two-component mixture model was introduced. The components of the mixture model represents the two classes balanced and unbalanced. The error terms were classified in either of the two classes. In the further work of the report we only looked at the balanced error terms. To capture correlation structure of the balanced error terms a seemingly unrelated regression (SUR) model was constructed with a 24-hour block-structure. Moreover these blocks respects the information structure of the market.

In the SUR model eight explanatory variables were assumed to have an impact on the system price. These explanatory variables were: The consumption of electricity on the market, water reservoir level in Norway, temperature in Norway, Norwegian holidays, daylight saving, month and weekday. In addition was a bias correcting explanatory variable introduced which was to capture the between-day dependence.

The selection of explanatory variables of the SUR model was done by the energy score. Similarly was evaluation by point forecasts calculated and analyzed, but since they do not considered the uncertainty structure in the predicted probabilistic forecasts their importance were downplayed. The energy score found that the optimal number of explanatory variables and the length of the training period changed with time. The evaluation was performed at four different time-intervals of 20 days each , which started the 5th of April, June, August and October 2008 and ended the 25th of each the respective months. In addition, the 24-hour block was dived into four groups which were assumed to share significant explanatory variables. this was done to lighten the computational burden.

The energy score preferred the shortest of the studied training periods which were 50 days for April. For the other months the score was optimal for a 70 to 80 days training period. Accordingly was the optimal training period for all the time periods for the model found to be 80 days. The choice of explanatory variables changed with time of year and

length of training period. The reservoir level was the explanatory variable most often left out, while the second most was the consumption. In addition were also in some cases the explanatory variable for the temperature and holidays left out. On the other hand were no more than two explanatory variable left out at once.

The exploratory analysis of the data showed heteroscedasticity and changing level of the power prices which we responded to by constructing a dynamic SUR model. This consisted of constructing a model where the set of optimum explanatory variable could and the length of the training period could change. However, even by always choosing the full set of explanatory variables in the regression model of the balanced error terms, the evaluation always found the stochastic model of the power prices to outperform the deterministic model. The deterministic model was not based on any (confidential) system knowledge. Although, the framework allows for other (system-based) model(s) to be improved by taking into account the uncertainty structure. Furthermore, this framework can also be applied on specific areas of the Nordic market, and thereby exploring more specific scenarios.

In the same way as the energy score is the Talagrand histogram used to perform evaluation on the probabilistic forecasts. The Talagrand histogram of the earlier mentioned forecast periods of April, June, August and October showed the prediction of the power price by a model consisting of the balanced error terms alone to be under-dispersed and biased. Indeed, this shows that the component consisting of the model of the unbalanced error terms is needed, since we by introducing the model unbalanced error term component to the model will expand its possible outcome space and leave it more calibrated. Moreover, the unbalanced error terms showed signs of skewness in the analysis of the observation. We could therefore assume that some of the bias displayed by the Talagrand histogram can be explained by the component of the unbalanced error terms.

In this paper we have focused on the balanced prices. To investigate the unbalanced prices and the group indicator more knowledge concerning the power system have to be acquired. However, the information about the power systems and their models are confidential. On the other hand was possible models based on the attempts by other papers discussed. As the unbalanced prices appeared to be skew distributed a possible candidate for "their" component in the mixture model could be the skewed t-distribution (Panagiotelis and Smith, 2008). The implementation of the model is a rather straight forward task, but the challenge lies in finding a good set of explanatory variables. Likewise would the fusion of the two components through their indicator variable be difficult, perhaps even being the hardest part of the model to deduce. In Chapter 4 we explored the method of temporal point process, which already are used on power prices by Cuaresma et al. (2004); Seifert and Uhrig-Homburg (2006).

The report proposes a model framework for making probabilistic forecasts of power prices and evaluating them. Even if all the details of the model framework are not settled, did the preliminary results show a working procedure. Furthermore the model could serve as a scenario generator which can be used as input to a stochastic optimization algorithm for power prices. In addition could its performance be evaluated in a stochastic setting.

Bibliography

B. Abraham and G.E.P. Box. Linear models and spurious observations. *Applied statistics*, 27:131–138, 1978.

J. Andersson and J. Lillestøl. Multivariate modelling and prediction of hourly one-day ahead prices at Nord Pool. In M. Bjørndal E., Bjørndal and M. Rönnquis, editors, *Energy, Natural Resources and Environmental Economic*, pages 243–268. 2007.

L. Baringhaus and C. Franz. On a new multivariate two-sample test. *Journal of Multivariate Analysis*, 88:190–206, 2004.

V.J. Berrocal, A.E. Raftery, and T. Gneiting. Probabilistic quantitative precipitation field forecasting using a two-stage spatial model. *Annals of Applied Statistics*, 2:1170–1193, 2008.

G.E.P. Box and G.C. Tiao. A bayesian approach to some outlier problems. *Biometrika*, 55:119–129, 1968.

A. Cartea and M. Figueroa. Pricing in electricity markets: A mean reverting jump diffusion model with seasonality. *Applied Mathematical Finance*, 12:313–335, 2005.

L. Clewlow and C. Strickland. *Energy Derivatives: Pricing and Risk Management*, chapter 2. Lacima Group, 2000.

J. Contreras, R. Espinora, F.J. Nogales, and A.J. Conejo. ARIMA models to predict next-day electricity prices. *IEEE Transactions on Power Systems*, 18:1014–1020, 2003.

J.C. Cuaresma, J. Hlouskova, S. Kossmeier, and M. Obersteiner. Forecasting electricity spot-prices using linear univariate time-series models. *Applied Energy*, 77:87–106, January 2004.

D.J. Daley and D. Vere-Jones. *An introduction to the theory of point processes*. Springer, 1988. chapters 3,8,13 and 14.

A.J.P. de Lande, P.H. Schavenmaker, and L. van der Sluis. *Electricity prices: Stochastic or deterministic?* ADL Consultancy, The Netherland, 2002. Working paper.

K. Engeland, B. Renard, I. Steinsland, and S. Kolberg. Evaluation of statistical models for forecast errors from the HBV model. *Journal of Hydrology*, 384:142–155, 2010.

C. Erlwein, F.E. Benth, and R. Mamon. HMM filtering and parameter estimation of an electricity spot price model. *Energy Economics*, 29:240–248, 2010.

S.E. Fleten and T.K. Kristoffersen. Short-term hydropower production planning by stochastic programming. *Computer and Operations Research*, 35:2656–2671, 2008.

S. Fruhwirth-Schnatter. *Finite mixture and Markov switching models*. Springer, 2006.

J. Fu and Z. Li. Different models and properties on LMP calculations. In *Proceedings in Power Engineering Society General Meeting*, Montreal, QC, Canada, June 2006.

R.C Garcia, J. Contreras, M. van Akkeren, and J.B.C. Garcia. GARCH models to predict day-ahead electricity prices. *IEEE Transactions on Power Systems*, 20:867–874, 2005.

C. Garcia-Martos, J. Rodrigez, and M.J. Sanchez. Unobserved component model for forecasting electricity prices and their volatility. `www.forecasters.org/pdfs/FinalReportSAS_GM_R_S.pdf`, 2011. Online; accessed 7th of February 2011.

T. Gneiting. Making and evaluating point forecasts. Technical report, Institut für angewandte mathematik Universität Heidelberg, March 2010.

T. Gneiting and R.E. Raftery. Strictly proper scoring rules, prediction and estimation. *Journal of the American Statistical Association*, 102:359–378, 2007.

T. Gneiting, A.H. Westveld III, A.E. Raftery, and T. Goldman. Calibrated probabilistic forecasting using ensemble model output statistics and minimum CRPS estimation. Technical Report 449, Department of Statistics, University of Washington, June 2004.

T. Gneiting, F. Balabdaoui, and Raftery A.E. Probabilistic forecasts, calibration and sharpness. *Journal of the Royal Statistical Society: Series B*, 69:243–268, 2007.

T. Gneiting, Stanberrym L.I., E.P. Grimit, L. Held, and N.A. Johnson. Assessing probabilistic forecasts of multivariate quantities, with an application to ensamble predictions of surface winds. *TEST*, 17: 211–235, 2008.

A.M. Gonzales, A.M. San Roque, and J. Garcia-Gonzalez. Modeling and forecasting electricity prices with input/output hidden Markov model. *IEEE Transaction of Power Systems*, 20:13–24, 2005.

T.M. Hamill. Interpretation of rank histograms for verifying ensample forecast. *Monthly Weather Review*, 129:550–560, 2001.

T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, Second edition, 2008. chapter 3.3.

J.L. Higle. Stochastic programming: Optimization when uncertainty matters. *Tutorials in operations research*, pages 30–53, 2005. Hamburg, MD.

R. Huisman, C. Huurman, and R. Mahieu. Hourly electricity prices in day-ahead markets. *Energy Economics*, 29:240–248, 2007.

P. Kall and S.W. Wallace. *Stochastic Programming*, chapter 2. John Wiley and Sons, 2nd edition, 2003.

G. Klæboe. Forecasting hourly electricity prices for biding optimization. Norwegian University of Science and Technology, 2011.

K.M. Knivflå and L. Rud. Market for electricity: structure, pricing, and performance. *Pacific Asian Journal of Energy*, 5:261–284, 1995.

A. Løland and X.K. Dimakos. Modeling Nord Pool NO1 area prices. *The Journal of Energy Markets*, 3: 73–92, 2010.

J.J. Lucia and H. Torro. Short-term electricity futures prices at Nord Pool: Forecasting power and risk premiums. University of Valencia, June 2005.

J.S. Marron and M.P. Wand. Exact mean integrated square error. *The annals of statistics*, 20:712–736, 1992.

G. McLachlan and D. Peel. *Finite mixture models.* John Wiley and Sons, 2000.

P.E. McSharry, S. Bouwman, and G. Bloemhof. Probabilistic forecast of the magnitude and timing of peak electricity demand. *IEEE Transctions on power systems*, 20:1166–1172, 2005.

J. Mullins, L. Wagner, and J. Foster. Price spikes in electricity markets: A strategic perspective. Technical Report 05, School of Economics, University of Queensland, 2010.

F.J. Nogales, J. Contreras, A.J. Conejo, and R. Espinola. Forecasting next-day electricity prices by time series models. *IEEE Transactions on Power Systems*, 17:342–348, 2002.

Nord Pool Spot AS. Nordic power exchange information. `http://www.nordpoolspot.com/`, 2011. Online; accessed 7th of February 2011.

NVE . Water reservoirs level. `http://www5.nve.no/magasinfylling/MagfyllTbl_AlleAar.aspx?Tidsenhet=Uke&Omr=NO`, 2011. Online; accessed 7th of February 2011.

A. Panagiotelis and M. Smith. Bayesian density forecasting of intraday electricity prices using multivariate skew t distributions. *International Journal of Forecasting*, 24:710–727, 2008.

A.E. Raftery, T. Gneiting, F. Balabdaoui, and M. Polakowski. Using bayesian model averaging to calibrate forecast ensambles. *Monthly Weather Review*, 133:1155–1174, 2005.

J.G. Rasmussen. Temporal point processes: The conditional intensity function. `http://people.math.aau.dk/~jgr/`, 2011. Online; accessed 5th of March 2011.

P. Schmidt. Estimation of seemingly unrelated regressions with unequal number of observation. *Journal of Econometrics*, 5:365–377, 1977.

J. Seifert and M. Uhrig-Homburg. Modelling jumps in electricity prices theory and empirical evidence. *Review of derivative research*, 10:59–85, 1 2006.

P. Skantze, M. Ilic, and J. Chapman. Stochastic modeling of electric power prices in a multi-market environment. In *Proceedings of Power Engineering Winter Meeting*, pages 1109–1114, January 2000.

V.K. Srivastava and D.E.A. Giles. *Seemingly unrelated regression equations models*. Marcel Dekker, Inc., 1987.

G.J Szekely. E statistiscs: The energy of statistical samples. Technical Report 16, Bowling Green State University, Department of Mathematics and Statistics, 2003.

O. Talagrand, R. Vautard, and B. Strauss. Evaluation of probabilistic evaluation systems. In *Poceedings of a workshop held at ECMWF on predictability*, pages 1–25, Reading, 20–22 October 1997. European Center for Medium Range-Range Forecasts.

J.M. Wooldridge. *Econometric analysis of cross section and panel data*. The MIT Press, 2nd edition, 2010. chapter 7.2-7.7.

A. Worthington, A. Kay-Spratle, and H. Higgs. Transmission of prices and price volatility in Australian electricity spot markets: A multivariate GARCH analysis. *Energy Economics*, 27:337–350, 2005.

L. Wu and M. Shahidehpour. A hybrid model for day-ahead price forecasting. *IEEE transaction of Power Systems*, 3:1519–1529, 2010.

A. Zeller. An efficient method for estimating seemingly unrelated regressions and test for aggregation bias. *Journal of American Statisical Society*, 57:348–368, 1962.

Appendix

Reduction of the CRPS to the MAE

As pointed out in Section 4.2.3 the continuous rank probability score function (CRPS) can be simplified when it evaluates deterministic forecast. To be more precise the CRPS reduces to the mean absolute error (MAE) score function. This can be shown through a simple manipulation of the definition of the CRPS. We use the same variable names as in Chapter 4. To recapitulate: The variable x_t is the observed value and the variable $\mathbf{y}_t$ is the predicted value(s). The index t denotes that we consider the variables at a single moment in time. Thus, the parameter T is the length of the time interval. Further F_t is the distribution function of the predicted values. The variable z is the threshold value. Putting this into the Brier score function found in Equation (4.2.2), where the distribution function of the deterministic forecast reduces to an indicator function

$$
\begin{aligned}
\mathrm{BS}(z) &= \frac{1}{T} \sum_{t=1}^{T} \mathbb{I}(z \geqslant y_t) - \mathbb{I}(z \geqslant x_t) \\
&= \frac{1}{T} \sum_{t=1}^{T} \mathbb{I}(z \geqslant y_t) + \mathbb{I}(z \geqslant x_t) - 2\mathbb{I}(z \geqslant \max(x_t, y_t)). \\
\mathrm{CRPS} &= \int_{-\infty}^{\infty} \mathrm{BS}(z) \, \mathrm{d}z \\
&= \frac{1}{T} \sum_{t=1}^{T} \int_{-\infty}^{\infty} \mathbb{I}(z \geqslant y_t) + \mathbb{I}(z \geqslant x_t) - 2\mathbb{I}(z \geqslant \max(x_t, y_t)) \, \mathrm{d}z \\
&= \frac{1}{T} \sum_{t=1}^{T} |x_t - y_t| \\
&= S_{\mathrm{MAE}}(x, y).
\end{aligned}
$$

Thus we see that the CRPS is reduced to the MAE when deterministic predictions are evaluated.

Month	50	60	70	80	90	100	110	120	130	140
April	8.80	9.35	9.31	9.63	10.3	10.9	12.8	14.0	14.5	13.9
June	11.2	8.71	8.81	8.09	8.40	9.08	9.31	9.64	9.59	10.6
August	8.05	8.23	7.63	7.82	8.32	8.80	8.84	10.2	10.3	10.0
October	7.08	6.95	6.98	7.06	7.20	7.25	7.51	8.28	8.99	9.40

Table I: *A table of the values in Figure 6.2.1. In the table the energy score for the predicted spot power prices from the 15th to the 25th of the months indicated is given. At the top of the columns the training period (in days) of the prediction used to calculate the energy score is given.*

Supplementary tables

In this section we give the Table I which contain the values of Figure 6.2.1.

R code

In the following paragraphs a brief walk through will be given, while the code itself will be presented at the end of the text. The source code is the implementation of the algorithms and statistical methods described in Chapter 5. The language utilized to do the calculations are `R`. The commands are mostly self-implemented, but some off-the-shelf commands are used. A single forecast for a given date is performed by the command `combine_forecast.R`. The command takes approximately 5 minutes and 30 second when run on a computer with the processor `Intel(R) Core(TM) i7 860` with $2.80\mathrm{G}Hz$. The leave-one-out best subset procedure made by the same computer by the command `best_subset.R` on one of the periods described in Section 5.4 takes 28 minutes and 48 seconds to run.

The forecast consists of two separate models, namely the deterministic and stochastic. This is described in Equation (2.4.1). The deterministic forecast is made in the file `detetministic_forecast.R`. The stochastic forecast, which is the main focus of this project, uses considerable more subroutines. The main program however is found in the file `build_model.R`. This model uses a number of internal subroutines. These are labeled with the same name, but with a number added. The following routines are used to make the stochastic forecast: To initialize and filter data of the length of the training period `initialize_data_frame.R` is used. The length of the training period is number of days in backwards that the user wants to use to make the estimation. In addition, the reservoir filling and consumption are scaled in respectively `meanfylling.R` and `meancon.R`. These subroutines are found `initialize_data_frame.R` and `build_model7.R`. The filtration of the unbalanced prices, the spikes, are done in the subroutine `spike_identification.R`. Once this selection and filtering are done the data are passed to file `parameter_estimation`. This function does also have a number of internal subroutines. These have the same name, but are labeled with a number. In this file the model matrix, covariance matrix and parameters are estimated. The data are first sent to the file `parameter_estimation1.R` where

an ordinary least square (OLS) estimation takes place. The covariance of the residuals from this process are then sent to another subroutine `parameter_estimation2`, where a feasible generalized least square (FGLS) estimation is performed. It is important to note that the estimation of both the OLS and FGLS are done such that it starts on lead day one and thereafter consequentially iterates itself through the other nine. When process in `parameter_estimation` are finished the output is passed back to `build_model`. In this file the model matrix and the parameter vector are arranged such that they are compatible with each other and that they take into account the missing data. Much of the work with compatibility is done in the file `beta_fill_max.R`. As the implementation strategy chosen is to fill in the estimated parameters into a vector which size equal that of the full model where no data are missing. This process is also utilized for the variance matrix. The main sub-files of `build_model.R` are `build_model6.R` and `build_model7.R`. The first being the place where all the data for the predicted period is gather and prepared. In `build_model6.R` the model matrix for the period to be predicted is prepared. In the end `build_model.R` estimated parameters of the multidimensional Gaussian distribution, which is then passed to the file `combine_forecast.R`. This is the main file, combining the deterministic and stochastic forecast. It also calculates the prediction intervals. After `build_model.R` the data are passed to `combine_forecast.R`.

In the end all of the predictions are combined in the main function `combine_forecast.R`. However, as mentioned previously the routine `parameter_estimation` takes as input the covariates for the different equation. In Section 4.1.1 and further Section 5.4 the possible choices of covariates are discussed. To evaluate which of the covariates to use the script `best_subset.R` is used. This script perform the best subset calculation. Further this program's subroutine, `forecast_evaluation.R`, calculates a user defined set of forecast which are to be considered in the evaluation. The forecasts are generated by `combined_forecast.R`. The evaluation of the subset of parameters are done in the `energy_score.R`, a sub-function of `forecast_evaluation.R`. This function uses evaluation the Energy score criteria described in Section 4.2.5.

```r
1  ########################################################################################
   ####################----------combined_forecast5.R----------####################
3  ########################################################################################
   comforecast <- function (DAY,MONTH,YEAR,COVARIATES,HISTORY,NUMBER){
5  source("example_det_forecast.r")
   source("forecast.gls.r")
7  print(Sys.time())
   forecast.combined <- matrix(0,nc=NUMBER,nr=240)
9  forecast.det <- det.forecast(DAY,MONTH,YEAR)
   forecast.stoc <- main.pred(DAY,MONTH,YEAR,COVARIATES,HISTORY,NUMBER)
11 spike.jump <- abs(forecast.stoc$spike.indicator)
   spikes.real <- forecast.det$r[c(2:length(forecast.det$r))]-forecast.det$r[c(1:(length(
       forecast.det$r)-1))]
13 spike.classified <- which(abs(spikes.real)>min(spike.jump[which(spike.jump>0)]))+1
   for(i in 1:NUMBER){
15   tmp <- cumsum(rnorm(240,0,1.0))
     forecast.combined[,i] <- as.vector(exp(forecast.det$p)+forecast.stoc$y.pred[,i])
17 }
   five.percent <- dim(forecast.combined)[2]/40
```

```
19  t.sorted <- t(apply(forecast.combined,1,sort))
    conf1 <- t.sorted[,(dim(forecast.combined)[2]/2)]
21  conf2 <- t.sorted[,(dim(forecast.combined)[2]-ceiling(five.percent))]
    conf3 <- t.sorted[,(1+ceiling(five.percent))]
23  print(Sys.time())
    comforecast <- list(forecast=forecast.combined,real=forecast.det$r,det=forecast.det$p,s=
        forecast.stoc$y.pred,sigma=forecast.stoc$sigma,beta=forecast.stoc$beta,spikes=spike.
        classified,conf1=conf1,conf2=conf2,conf3=conf3)
25  }
```

Listing 1: *The main function. Passes arguments to the deterministic forecast and stochastic forecast. In addition it is here that the deterministic forecast and the predicted error terms are combined.*

```
1  ###############################################################################################
   ###################--------deterministic_forecast.R--------###################
3  ###############################################################################################
   det.forecast <- function(day,month,year){
5  source("redigerdato.R")
   data <- read.table("tidsserier_alle_2005-2010.txt",skip=0,header=T,sep="\t")
7  dato.info <-redigerdato(as.character(data[,1]),hundreaar=2000)
   forecast.window <- 10
9  price <- data[which(dato.info$year==year),2]

11 na <- which(is.na(price))
   price[na] <- 0.5*(price[(na+1)]+price[(na-1)])
13
   start <- (which(dato.info$year==year&dato.info$month==month&dato.info$day==day)[1]-which(
       dato.info$year==year)[1])/24
15 memo <- 24*8
   now <- (start-1)*24
17 real.prices <- price[((start)*24+1-memo):(now+24+forecast.window*24)]
   price2 <- real.prices
19
   lag1 <- 24
21 lag2 <- 168
   lag3 <- 192
23 i <- 1

25 while(i<=(10)){
     tmp1 <- rep(now-lag2+(i-1)*24,24)+c(1:24)
27   tmp2 <- rep(now-lag3+(i-1)*24,24)+c(1:24)
     tmp1 <- tmp1-(start-9)*24
29   tmp2 <- tmp2-(start-9)*24
     drift <- median(price2[tmp1]-price2[tmp2])
31   tmp3 <- now+(i)*24+c(1:24)-24
     tmp4 <- now+seq(-23,0,1)+(i-1)*24
33   tmp3 <- tmp3-(start-9)*24
     tmp4 <- tmp4-(start-9)*24
35   price2[tmp3] <- price2[tmp4]+drift
     i <- i+1
37 }
   i <- 1
39 price3 <- log(real.prices)
   while(i<=(10)){
41   tmp1 <- rep(now-lag2+(i-1)*24,24)+c(1:24)
     tmp2 <- rep(now-lag3+(i-1)*24,24)+c(1:24)
43   tmp1 <- tmp1-(start-9)*24
     tmp2 <- tmp2-(start-9)*24
45   drift <- median(price3[tmp1]-price3[tmp2])
     tmp3 <- now+(i)*24+c(1:24)-24
47   tmp4 <- now+seq(-23,0,1)+(i-1)*24
     tmp3 <- tmp3-(start-9)*24
49   tmp4 <- tmp4-(start-9)*24
     price3[tmp3] <- price3[tmp4]+drift
```

```
51    i <- i+1
   }
53
   det.forecast <- list(r=real.prices[-c(1:memo)],p=price3[-c(1:memo)],price2=price2[-c(1:
       memo)])
55 }
```

Listing 2: *The place where the deterministic forecasts are generated.*

```
   ######################################################################################################
2  ####################------------build_model.R------------####################
   ######################################################################################################
4  # INPUT- the current day (DAY), month (MONTH) and year (YEAR). In addition, a
   #binary vector of length 7 (COVARIATES), indicating whetever a covariate should be
6  #present (1) or left out (0), will be the input. COVARIATE 1 is the consumption, 2
   #is the reservoir level, 3 temperature, 4 month, 5 weekday, 6 holidays and 7
8  #daylight saving. OUPUT are the predicted value of the error term (y.pred) and its
   #confidence intercal (y.conf).
10 ######################################################################################################
   ######################################################################################################
12 main.pred <- function(DAY,MONTH,YEAR,COVARIATES,HISTORY,NUMBER){
   library("fSeries")
14 init <- init(DAY,MONTH,YEAR,HISTORY)
   fore <- foreblock(DAY,MONTH,YEAR,init,COVARIATES,HISTORY,NUMBER)
16 y.pred <- matrix(0,nr=240,nc=NUMBER)
   for(number in 1:NUMBER){
18    model.c <- cbind(fore$model,fore$error.term[number,])
      model.c <- model.rear(model.c)
20    model.tmp <- array(0,c(24,576,10))
      for(lead.day in 1:10){
22        index <- c(((lead.day-1)*24+1):(lead.day*24))
          covar.tmp <-model.rear2(model.c[index,])
24        model.tmp[,,lead.day] <- as.matrix(covar.tmp)
      }
26    model.tmp <- as.matrix(model.rear3(model.tmp))
      y.pred[,number] <- mvrnorm(240,model.tmp%*%fore$beta,fore$sigma)
28 }
   y.conf <- model.tmp%*%fore$beta.sigma
30 fore.model <- as.matrix(fore$model)
   main.pred <- list(y.pred=y.pred,sigma=fore$sigma,beta=fore$beta,spike.indicator=fore$spike
       .indicator)
32 }
```

Listing 3: *The main file of the stochastic forecast model. The file gathers separate parts of the code as reading of input data, matching of dimensions and making the model matrices.*

```
   ######################################################################################################
2  ####################------------build_model1.R------------####################
   ######################################################################################################
4  model.day <- function(weekday){
      box <- rep(1,24)
6     v1 <- c(box,rep(0,6*24),box,rep(0,2*24))
      v2 <- c(rep(0,1*24),box,rep(0,6*24),box,rep(0,1*24))
8     v3 <- c(rep(0,2*24),box,rep(0,6*24),box)
      v4 <- c(rep(0,3*24),box,rep(0,6*24))
10    v5 <- c(rep(0,4*24),box,rep(0,5*24))
      v6 <- c(rep(0,5*24),box,rep(0,4*24))
12    v <- matrix(c(v1,v2,v3,v4,v5,v6),nc=6)
      if(weekday!=7){
14       v.tmp <- v[,-c(1:weekday)]
          v <- cbind(v.tmp,v[,c(1:weekday)])
16    }
   model.day <- v
```

```
18 }
```

Listing 4: *Splits the week in half, and puts the current day first.*

```
   ###############################################################################################
 2 ####################———————build_model2.R————————####################
   ###############################################################################################
 4 model.month <- function(MONTH, check, breaks, scope){
     m <- matrix(0,nc=11,nr=scope)
 6   if(MONTH!=1){
       if(check==1) m[,(MONTH-2)] <- 1
 8     if(check==0){
         m[1:(scope-breaks*24-1),(MONTH-2)] <- 1
10       if(MONTH!=12) m[(scope-breaks*24-1):scope,(MONTH-2)] <- 1
         if(MONTH==12)  m[(scope-breaks*24-1):scope,11] <- 1
12     }
     }
14 model.month <- m
   }
```

Listing 5: *Checks if we encountered a new month in the forecast horizon.*

```
 1 ###############################################################################################
   ####################———————build_model3.R————————####################
 3 ###############################################################################################
   model.rear <- function(model){
 5   line <- rep(c(1:24),dim(model)[1]/24)
     model <- cbind(line,model)
 7   model.sort <- model[which(model[,1]==1),]
     for(i in 3:25){
 9     model.sort <- rbind(model.sort,model[which(model[,1]==(i-1)),])
     }
11 model.rear <- model.sort[,-1]
   }
```

Listing 6: *Adds the indicator of between-day correlation to the model matrix.*

```
   ###############################################################################################
 2 ####################———————build_model4.R————————####################
   ###############################################################################################
 4 model.rear2 <- function(model){
   model <- as.matrix(model)
 6   list.tmp <- list(t(model[1,]),t(model[2,]),t(model[3,]),t(model[4,]),t(model[5,]),t(
         model[6,]),t(model[7,]),t(model[8,]),t(model[9,]),t(model[10,]),t(model[11,]),t(
         model[12,]),t(model[13,]),t(model[14,]),t(model[15,]),t(model[16,]),t(model[17,]),t(
         model[18,]),t(model[19,]),t(model[20,]),t(model[21,]),t(model[22,]),t(model[23,]),t(
         model[24,]))
   tmp <- bdiag(list.tmp)
 8 model.rear2 <- tmp
   }
```

Listing 7: *Constructs the (sparse) part of the (same lead day) model matrix.*

```
 1 ###############################################################################################
   ####################———————build_model5.R————————####################
 3 ###############################################################################################
   model.rear3 <- function(model){
 5   list.tmp <- list(model[,,1],model[,,2],model[,,3],model[,,4],model[,,5],model[,,6],model
         [,,7],model[,,8],model[,,9],model[,,10])
   tmp <- bdiag(list.tmp)
 7 model.rear2 <- tmp
   }
```

Listing 8: *Adds the different lead days (of the last sub-function) to an even greater (sparse) matrix.*

```r
###############################################################################
##################---------------build_model6.R---------------################
###############################################################################
foreblock <- function(DAY,MONTH,YEAR,init ,COVARIATES,HISTORY,NUMBER){
source("full.model.matrix.r")
source("beta.fill.max.r")
source("redigerdato.R")
source("beta.sigma.fill.max.r")
data <- read.table("tidsserier_alle_2005-2010.txt",skip=0,header=T,sep="\t")
dato.info <-redigerdato(as.character(data[,1]),hundreaar=2000)
kalender <- redigerdato(as.character(data[which(dato.info$year==YEAR),1]),hundreaar=2000)
today.index <- which(dato.info$month==MONTH&dato.info$day==DAY)
months <- c(kalender$month[(today.index-HISTORY*24)][1]:kalender$month[today.index][1])
t <- main(DAY,MONTH,YEAR,0,COVARIATES,HISTORY)
beta <- t$beta.m
beta.sigma <- t$sigma.mA
end <- 0
count <- 0
for(i in 1:10){
   if(i==1){
      start <- end+1
      end <- end+dim(t$sigma.m1)[1]
      sigma1 <- diag(t$sigma.m1)
   }else{
      count <- count+1
      start <- end+1
      end <- end+dim(beta.sigma[,,(i-1)])[1]
      if(count==1){
         sigma <- diag(beta.sigma[,,(i-1)])
         sigma.m <- matrix(0,nr=length(sigma),nc=9)
         sigma.m[,1] <- sigma
      }else{   sigma.m[,(i-1)] <- diag(beta.sigma[,,(i-1)])}
   }
}
sigma.beta <- beta+1.96*sqrt(c(sigma1,as.vector(sigma.m)))
COVARIATES <- t$COVARIATES
beta <- beta.fill.max(beta,COVARIATES,months,t$month.length)
beta.sigma <- beta.fill.max(sigma.beta,COVARIATES,months,t$month.length)
scope <- 10*24
forecast <- vector(mode="numeric",length=scope)
weekday <- kalender$weekday[which(dato.info$month==MONTH&dato.info$day==DAY)][1]
days <- c("ma","ti","on","to","fr","lo","so")
weekday <- which(days==weekday)
day.dummy <- model.day(weekday)
tmp <- which(dato.info$year==YEAR&dato.info$month==MONTH&dato.info$day==DAY)[1]
check <- as.numeric(MONTH==redigerdato(as.character(data[(tmp+scope),1]))$month)
if(check==0){
   breaks <- 1
   check.copy <- check
   i <- 24
   while(check.copy!=1){
      check.copy <- as.numeric(MONTH==redigerdato(as.character(data[(tmp+scope-i),1]))$month
         )
      i <- i+24
      breaks <- breaks+1
   }
}
month.dummy <- model.month(MONTH,check,breaks,scope)
tmp <- tmp- which(dato.info$year==YEAR)[1]
init <- init[(dim(init)[1]-scope+1):dim(init)[1],]
```

```r
60  model <- cbind(as.matrix(rep(1,scope)),as.matrix(init[,c(1:3)]),month.dummy,day.dummy,as.
        matrix(init[,c(4,5)]))
    group.size <- cumsum(c(1,552,rep(576,9)))
62  error.term <- matrix(0,nr=NUMBER,nc=(24*10))
    counter <- 24
64  for(lead.time in 1:10){
      if(lead.time==1){
66       det.r <- det.d <- matrix(0,nr=24,nc=HISTORY)
         for(history.i in HISTORY:1){
68         day.d <- kalender$day[(today.index-history.i*24)][1]
           month.d <- kalender$month[(today.index-history.i*24)][1]
70         year.d <- kalender$year[(today.index-history.i*24)][1]
           det.forecast.d <- det.forecast(day.d,month.d,year.d)
72         det.r[,(history.i)] <- det.forecast.d$r[1:24]
           det.d[,(history.i)] <- det.forecast.d$p[1:24]
74
         }
76       rep.object <- det.r-det.d-det.s
         rep.object <- rep.object/(0.1*max(abs(rep.object)))
78       sample.elem <- matrix(rep(seq(0,24*(HISTORY-1),HISTORY),NUMBER)+sample(1:HISTORY,24*
             NUMBER,replace=T),nc=24,byrow=T)
         error.term[,c(1:counter)] <- matrix(rep.object[sample.elem],byrow=F,nc=24)
80    }else{
         beta.index <-  rev(seq((group.size[(lead.time+1)]-1),(group.size[(lead.time)]),-24))
82       beta.i <- beta[beta.index]
         rep.object <- beta.i*error.term[,(counter-23):(counter)]
84       error.term[,c((counter+1):(counter+24))] <- rep.object
         counter <- counter+24
86    }
    }
88  forecast <- list(model=model,error.term=error.term,beta=beta,sigma.m=sigma.m,sigma1=sigma1
        ,beta.sigma=beta.sigma,spike.indicator=t$spike.indicator)
    }
```

Listing 9: *Construct the entire model matrix.*

```r
1  ####################################################################################################
   ####################----------build_model7.R------------####################
3  ####################################################################################################
   init <- function(DAY,MONTH,YEAR,HISTORY){
5  source("meancon.R")
   source("meanfylling.R")
7  source("redigerdato.R")
   data <- read.table("tidsserier_alle_2005-2010.txt",skip=0,header=T,sep="\t")
9  dato.info <-redigerdato(as.character(data[,1]),hundreaar=2000)
   scope <- 10*24
11 if(YEAR==2007){
     index <- which(dato.info$year==YEAR)
13 }else{
     index <- which(dato.info$year==YEAR|dato.info$year==(YEAR-1))
15 }
   index.history <- which(dato.info$day==DAY&dato.info$month==MONTH&dato.info$year==YEAR)[1]-
       HISTORY*24
17 index.future <- which(dato.info$day==DAY&dato.info$month==MONTH&dato.info$year==YEAR)[1]+
       scope
   index <- index[which(index>=index.history&index<index.future)/24]
19 con <- data[index,13]
   con <- as.vector(interpNA(con,"linear"))
21 tmp <- meancon()
   tmp <- approx(x=tmp,y=NULL,method="linear",n=length(con))
23 scaledcon <- con/tmp$y-1
   fylling <- read.table("Fyllingsgrad_2002-2010.txt",skip=0,header=T,sep="\t")
25 fyll <- rev(fylling[which(fylling[,1]==YEAR),3])
   fyll <- approx(x=fyll,y=NULL,method="linear",n=length(con))
```

```
27 tmp <- meanfylling()
   tmp <- approx(x=tmp,y=NULL,method="linear",n=length(con))
29 scaledfyll <- fyll$y/tmp$y-1
   temp <- data[index,30]
31 holy <- data[index,23]
   summer <- data[index,29]
33 frame <- data.frame(con=scaledcon,fyll=scaledfyll,temp=temp,holy=holy,summer=summer)
   init <- frame
35 }
```

Listing 10: *Initializes the data for the stochastic forecast model.*

```
1 ###############################################################################################
   ####################------------beta_fill_max.R------------####################
3 ###############################################################################################
   beta.fill.max <- function(beta,COVARIATES,months,month.length.r){
5 size.lead.day1 <- 24*24
   size.lead.day <- 24*24
7 size.beta.full.final <- size.lead.day1+9*size.lead.day
   beta.final <- vector(mode="numeric",length=size.beta.full.final)
9 pos.b <- cumsum(c(1,(size.lead.day1),rep((size.lead.day),9)))
   end.group <- 1
11 for(lead.time in 1:10){
     beta.full <- vector(mode="numeric",length=24*24)
13   COVARIATES.copy <- COVARIATES[,,lead.time]
     month.length <- month.length.r[lead.time]
15   COVARIATES.month <- COVARIATES.copy[,rep(4,11)]
     if(month.length!=0){
17     if(month.length[1]<length(months)){
         COVARIATES.month <- COVARIATES.copy[,rep(4,11)]
19       differ <- length(months)-month.length[1]+1
         COVARIATES.month[,-(c(months[1:differ])-1)] <- 0
21     }
       if(month.length[1]==length(months)){
23       COVARIATES.month <- COVARIATES.copy[,rep(4,11)]
         COVARIATES.month[,(-c(months[-length(months)])+1)] <- 0
25     }
     }
27   for(i in 1:24){
       if(((sum(COVARIATES[,4,lead.time])!=24)&&(COVARIATES[i,4,lead.time]==1))){
29       tmp <- which(COVARIATES.month==1,arr.ind=T)[,2]
         COVARIATES.month[i,tmp[1]] <- 0
31     }
     }
33   COVARIATES.weekday <- COVARIATES.copy[,rep(5,6)]
     if(lead.time==1){
35     COVARIATES.copy <- cbind(rep(1,24),COVARIATES.copy[,c(1:3)],COVARIATES.month,
           COVARIATES.weekday,COVARIATES.copy[,c(6:7)])
     }else{ COVARIATES.copy <- cbind(rep(1,24),COVARIATES.copy[,c(1:3)],COVARIATES.month,
           COVARIATES.weekday,COVARIATES.copy[,c(6:7)],rep(1,24))}
37   counter <- end.group
     for(i in 1:24){
39     line <- 24*(i-1)

41     end <- sum(COVARIATES.copy[i,])-1
       betas <- beta[(counter):(end+counter)]
43     counter <- counter+end+1
       sep.z <- which(COVARIATES.copy[i,]!=0)
45     if(lead.time==1){
         betas <- c(betas,1)
47       sep.z <- c(sep.z,24)
       }
49     beta.full[(line+sep.z)] <- betas
       if(i==24){
```

```
51        if (lead.time==1){
            end.group <- counter
53       }else     end.group <- counter
        }
55    }
    beta.final[pos.b[lead.time]:(pos.b[(lead.time+1)]-1)] <- beta.full
57 }
   return(beta.final)
59 }
```

Listing 11: *As the number of βs used is not equal that of the full model. The estimated βs are put into the right place here, and the ones that are not used aare put to zero. Also used to allocate the standard deviation of the βs.*

```
1 ##########################################################################################
   ####################——————————meancon.R——————————####################
3 ##########################################################################################
   meancon <- function(){
5 library("fSeries")
   source("redigerdato.R")
7 data <- read.table("tidsserier_alle_2005-2010.txt",skip=0,header=T,sep="\t")
   dato.info <-redigerdato(as.character(data[,1]),hundreaar=2000)
9 index1 <- 1
   index2 <- which(dato.info$year==2008)[1]-1
11 con <- data[seq(index1,index2,1),13]
   con <- as.vector(interpNA(con,"linear"))
13 tmp <- apply(matrix(con,nc=(365*24),byrow=T),2,mean)
   meancon <- tmp
15 }
```

Listing 12: *Calculates the mean consumption of 2004 and 2005. This is used to find the relative size of input values according to this level.*

```
1 ##########################################################################################
   ####################——————————meanfylling.R——————————####################
3 ##########################################################################################
   meanfylling <- function(){
5 fylling <- read.table("Fyllingsgrad_2002-2010.txt",skip=0,header=T,sep="\t")
   index1 <- which(fylling[,1]==2002&fylling[,2]==1)
7 index2 <- which(fylling[,1]==2006&fylling[,2]==52)
   index3 <- which(fylling[,2]!=53)
9 index <- rev(intersect((index2:index1),index3))
   niva <- apply(matrix(fylling[index,3],nc=52,byrow=T),2,mean)
11 meanfylling <- niva
   }
```

Listing 13: *Calculates the mean reservoir filling of 2004 and 2005. This is used to find the relative size of input values according to this level.*

```
   ##########################################################################################
2 ####################————————paramter_estimation.R————————####################
   ##########################################################################################
4 main <- function(DAY,MONTH,YEAR,plot,COVARIATES,HISTORY){
   source("beta.fill.max.r")
6 source("redigerdato.R")
   source("example_det_forecast.r")
8 data <- read.table("tidsserier_alle_2005-2010.txt",skip=0,header=T,sep="\t")
   dato.info <-redigerdato(as.character(data[,1]),hundreaar=2000)
10 kalender <- redigerdato(as.character(data[which(dato.info$year==YEAR),1]),hundreaar=2000)
   today.index <- which(dato.info$month==MONTH&dato.info$day==DAY)
12 months <- c(kalender$month[(today.index-HISTORY*24)][1]:kalender$month[today.index][1])
   COVARIATES.a <- array(0,c(24,7,10))
```

```r
14 det.r <- det.d <- matrix(0,nr=240,nc=(HISTORY-10))
   for(history.i in HISTORY:11){
16    day.d <- kalender$day[(today.index-history.i*24)][1]
      month.d <- kalender$month[(today.index-history.i*24)][1]
18    year.d <- kalender$year[(today.index-history.i*24)][1]
      det.forecast.d <- det.forecast(day.d,month.d,year.d)
20    det.r[,(history.i-10)] <- det.forecast.d$r
      det.d[,(history.i-10)] <- det.forecast.d$p
22 }
   fit.ols <- fit.ols(DAY,MONTH,YEAR,COVARIATES,HISTORY,1,NULL)
24 COVARIATES.a[,,1] <- fit.ols$COVARIATES
   fit.gls <- fit.gls(fit.ols$Omega.inv.gls,fit.ols$fit1,fit.ols$model,fit.ols$epsilon.hat)
26 fit.gls1 <- fit.gls
   beta.m <- fit.gls$beta.hat
28 month.length <- fit.ols$month.length[1]
   sigma.m1 <- fit.gls$test
30 for(lead.time in 2:10){
      last.error <-  as.vector(det.r[c(((lead.time-2)*24+1):((lead.time-1)*24)),])-as.vector(
          det.d[c(((lead.time-2)*24+1):((lead.time-1)*24)),])-as.vector(fit.ols$model%*%fit.
          gls$beta.hat)
32    fit.ols.tmp <- fit.ols(DAY,MONTH,YEAR,COVARIATES,HISTORY,lead.time,last.error)
      fit.gls.tmp <- fit.gls(fit.ols.tmp$Omega.inv.gls,fit.ols.tmp$fit1,fit.ols.tmp$model,fit.
          ols.tmp$epsilon.hat)
34    COVARIATES.a[,,lead.time] <- fit.ols.tmp$COVARIATES
      beta.m <- c(beta.m,fit.gls.tmp$beta.hat)
36    if(lead.time==2){
          sigma.mA <- array(0,c(dim(fit.gls.tmp$test)[1],dim(fit.gls.tmp$test)[1],9))
38    }
      sigma.mA[,,(lead.time-1)] <- fit.gls.tmp$test
40    month.length <- c(month.length,fit.ols.tmp$month.length[1])
   }
42 names <- names(fit.ols$fit1$coefficients)
   if(plot==1) plot.beta(names,beta.m[iter,],sigma.m[iter,])
44 main <- list(fit.ols=fit.ols,beta.m=beta.m,COVARIATES=COVARIATES.a,month.length=month.
       length,sigma.m1=sigma.m1,sigma.mA=sigma.mA,spike.indicator=fit.ols$spike.indicator)
   }
```

Listing 14: *The function where the OLS and GLS are done. Calculate the parameters of the multidimensional Gaussian density.*

```r
1 ###############################################################################################
  ###################------------paramter_estimation1.R------------######################
3 ###############################################################################################
  fit.ols <- function(DAY,MONTH,YEAR,COVARIATES,HISTORY,lead.time,last.error){
5 source("fgls.r")
  source("reduced.model.r")
7 library("Matrix")
  library("MASS")
9 source("foo.r")
  epsilon.hat <- NULL
11 frame <- fgls(DAY,MONTH,YEAR,HISTORY)
  if(lead.time==1){
13    full.model <- formula(epsilon~con+fyll+temp+month+weekday+holy+summer)
  }else full.model <- formula(epsilon~con+fyll+temp+month+weekday+holy+summer+last.error.c)
15 forecast.l <- dim(frame)[1]
  frame.c <- frame[-c((forecast.l-240+lead.time*24):forecast.l),]
17 frame.c <- frame.c[-c(1:(lead.time*24-1)),]
  for(i in 1:24){
19 # kill covariates which only have one factor. Delete them and fill in ones (avoiding error
       message)
     frame.tmp <- frame.c[which(frame.c$ind==(i-1)),]
21    epsilon.non.na <- which(!is.na(frame.tmp$epsilon[which(frame.tmp$ind==(i-1))]))
     group.start <- which(frame.tmp$ind==(i-1))[1]
23    epsilon.non.na <- epsilon.non.na+group.start
```

```r
      half <- epsilon.non.na[1:floor(length(epsilon.non.na)/2)]
25    tmp <- subset(frame.tmp,ind==(i-1))$month[1]
      if(length(which(frame.tmp$month==tmp))==dim(frame.tmp)[1]){
27      COVARIATES[,4] <- 0
        frame.tmp$month[(half-1)] <- abs(as.integer(tmp)-2)
29    }

31    tmp <- frame.tmp$holy[1]

33    if(length(which(frame.tmp$holy==tmp))==dim(frame.tmp)[1]){
        COVARIATES[,6] <- 0
35      frame.tmp$holy[(half-1)] <-  abs(as.integer(tmp)-2)
      }
37
      tmp <- frame.tmp$summer[1]
39    if(length(which(frame.tmp$summer==tmp))==dim(frame.tmp)[1]){
        COVARIATES[,7] <- 0
41      frame.tmp$summer[(half-1)] <-  abs(as.integer(tmp)-2)
      }
43 # reduce model
      if(sum(COVARIATES[i,]) !=7){
45      model <- reduced.model(COVARIATES[i,], lead.time)
      }else{
47      model <- full.model
      }
49    nam <- paste("full.model",i,sep="")
      assign(nam,model)
51    nam2 <- paste("fit",i,sep="")
      if(lead.time!=1){
53      last.error.c <- as.vector(last.error[seq(i,length(last.error),24)])
        frame.tmp<- cbind(frame.tmp,last.error.c)
55    }
      fit <- lm(eval(parse(text=paste("full.model",i,sep=""))),frame.tmp,na.action(na.omit))
57    assign(nam2,fit)
      epsilon.hat <- c(epsilon.hat,eval(parse(text=paste("fit",i,"$","residuals",sep=""))))
59 }
   if(length(which(is.na(frame.c$epsilon)))>0){
61   epsilon.na <- vector(mode="numeric",length=length(frame.c$epsilon))
     epsilon.na[-which(is.na(frame.c$epsilon))] <- epsilon.hat
63   epsilon.na[which(is.na(frame.c$epsilon))] <- NA
   }else{
65   epsilon.na <- frame$epsilon
   }
67 fit.ols <- epsilon.na
   S.hat <- cov(matrix(epsilon.na,nc=24,byrow=T),use="pairwise.complete.obs")
69 S.hat[which(is.na(S.hat))] <- S.hat[which(!is.na(S.hat))][1]
   S.hat.inv <- inv((S.hat+diag(0.0001,24)))
71 list <- list(S.hat.inv)
   T <- length(frame.c$epsilon)/24
73 Omega.inv <- bdiag(rep(list,T))
   if(length(which(is.na(frame.c$epsilon)))>0){
75   frame.gls <- frame.c[-which(is.na(frame.c$epsilon)),]
     Omega.inv.gls <- Omega.inv[-which(is.na(frame.c$epsilon)),]
77   Omega.inv.gls <- Omega.inv.gls[,-which(is.na(frame.c$epsilon))]
   }else{
79   frame.gls <- frame.c
     Omega.inv.gls <- Omega.inv
81 }
   model <- model.matrix(fit1)
83 month.length <- vector(mode="numeric",length=24)
   month.length[1] <- foo("month",names(fit1$coefficients))
85 for(i in 2:24){
     model <- bdiag(model,model.matrix(get(paste("fit",i,sep=""))))
87   month.length[i] <- foo("month",names(eval(parse(text=paste("fit",i,"$","coefficients",
         sep="")))))
```

```
    }
89  fit.ols <- list(Omega.inv.gls=Omega.inv.gls,fit1=fit1,model=model,epsilon.hat=epsilon.hat,
        epsilon.na=epsilon.na,epsilon=frame.c$epsilon,COVARIATES=COVARIATES,month.length=month
        .length,spike.indicator=frame$spike.indicator)
    }
```

Listing 15: *The fitting of an OLS for the residuals of the error model.*

```
    ############################################################################################
2   #####################————paramter_estimation2.R————#######################
    ############################################################################################
4   fit.gls <- function(Omega.inv.gls,fit1,model.input,epsilon.hat){
    PLUTO <- 0
6   library("Matrix")
    pre.multiplication <- t(model.input)%*%Omega.inv.gls%*%model.input
8   pre.multiplication <- inv(pre.multiplication)
    sigma <- pre.multiplication
10  epsilon.hat <- matrix(epsilon.hat,nc=1)
    beta <- sigma%*%t(model.input)%*%Omega.inv.gls%*%epsilon.hat
12  names <- names(fit1$coefficients)
    beta.hat <- as.vector(beta)
14  sigma.diag <- diag(sigma)
    test <-inv(t(model.input)%*%model.input)
16  fit.gls <- list(beta.hat=beta.hat,sigma.diag=sigma.diag,sigma=sigma,test=test)
    }
```

Listing 16: *The fitting of a GLS for the feasible variance matrix*

```
1   ############################################################################################
    #####################————paramter_estimation3.R————#######################
3   ############################################################################################
    Kronecher <- function(S.hat,epsilon.hat,epsilon.na){
5   S.hat.inv <- inv(S.hat)
    list <- list(S.hat.inv)
7   T <- length(epsilon.na)/24
    Omega.inv <- bdiag(rep(list,T))
9   if(length(epsilon.hat)!=length(epsilon.na)){
    Omega.inv <- Omega.inv[-which(is.na(epsilon.na)),-which(is.na(epsilon.na))]
11  }
    Omega.inv <- Omega.inv
13  }
```

Listing 17: *Calculates the Kronecker operator.*

```
1   ############################################################################################
    #####################————reduce_model.R————#######################
3   ############################################################################################
    reduced.model <- function(COVARIATES,lead.time){
5   covar <- c("con","fyll","temp","month","weekday","holy","summer")
    cond.number <- sum(COVARIATES)
7   cond <- 7
    if(lead.time==1){
9   if(cond.number<cond){
    exclude1 <- covar[which(COVARIATES==0)][1]
11  red.model <- eval(parse(text=paste("epsilon~con+fyll+temp+month+weekday+holy+summer-",
        exclude1,sep="")))
    }
13  cond <- cond-1
    if(cond.number<cond){
15  exclude2 <- covar[which(COVARIATES==0)][2]
    red.model <- eval(parse(text=paste("epsilon~con+fyll+temp+month+weekday+holy+summer-",
        exclude1,"-",exclude2,sep="")))
```

```r
17 }
   cond <- cond-1
19 if (cond.number<cond){
     exclude3 <- covar[which(COVARIATES==0)][3]
21   red.model <- eval(parse(text=paste("epsilon~con+fyll+temp+month+weekday+holy+summer-",
         exclude1,"-",exclude2,"-",exclude3,sep="")))
   }
23 cond <- cond-1
   if (cond.number<cond){
25   exclude4 <-covar[which(COVARIATES==0)][4]
     red.model <- eval(parse(text=paste("epsilon~con+fyll+temp+month+weekday+holy+summer-",
         exclude1,"-",exclude2,"-",exclude3,"-",exclude4,sep="")))
27 }
   cond <- cond-1
29 if (cond.number<cond){
     exclude5 <- covar[which(COVARIATES==0)][5]
31   red.model <- eval(parse(text=paste("epsilon~con+fyll+temp+month+weekday+holy+summer-",
         exclude1,"-",exclude2,"-",exclude3,"-",exclude4,"-",exclude5,sep="")))
   }
33 cond <- cond-1
   if (cond.number<cond){
35   exclude6 <- covar[which(COVARIATES==0)][6]
     red.model <- eval(parse(text=paste("epsilon~con+fyll+temp+month+weekday+holy+summer-",
         exclude1,"-",exclude2,"-",exclude3,"-",exclude4,"-",exclude5,"-",exclude6,sep="")))
37 }
   cond <- cond-1
39 if (cond.number<cond){
     exclude7 <- covar[which(COVARIATES==0)][7]
41   red.model <- eval(parse(text=paste("epsilon~con+fyll+temp+month+weekday+holy+summer-",
         exclude1,"-",exclude2,"-",exclude3,"-",exclude4,"-",exclude5,"-",exclude6,"-",
         exclude7,sep="")))
   }
43 }else{
   if (cond.number<cond){
45   exclude1 <- covar[which(COVARIATES==0)][1]
     red.model <- eval(parse(text=paste("epsilon~con+fyll+temp+month+weekday+holy+summer+last
         .error.c-",exclude1,sep="")))
47 }
   cond <- cond-1
49 if (cond.number<cond){
     exclude2 <- covar[which(COVARIATES==0)][2]
51   red.model <- eval(parse(text=paste("epsilon~con+fyll+temp+month+weekday+holy+summer+last
         .error.c-",exclude1,"-",exclude2,sep="")))
   }
53 cond <- cond-1
   if (cond.number<cond){
55   exclude3 <- covar[which(COVARIATES==0)][3]
     red.model <- eval(parse(text=paste("epsilon~con+fyll+temp+month+weekday+holy+summer+last
         .error.c-",exclude1,"-",exclude2,"-",exclude3,sep="")))
57 }
   cond <- cond-1
59 if (cond.number<cond){
     exclude4 <-covar[which(COVARIATES==0)][4]
61   red.model <- eval(parse(text=paste("epsilon~con+fyll+temp+month+weekday+holy+summer+last
         .error.c-",exclude1,"-",exclude2,"-",exclude3,"-",exclude4,sep="")))
   }
63 cond <- cond-1
   if (cond.number<cond){
65   exclude5 <- covar[which(COVARIATES==0)][5]
     red.model <- eval(parse(text=paste("epsilon~con+fyll+temp+month+weekday+holy+summer+last
         .error.c-",exclude1,"-",exclude2,"-",exclude3,"-",exclude4,"-",exclude5,sep="")))
67 }
   cond <- cond-1
69 if (cond.number<cond){
     exclude6 <- covar[which(COVARIATES==0)][6]
```

```
71    red.model <- eval(parse(text=paste("epsilon~con+fyll+temp+month+weekday+holy+summer+last
          .error.c-",exclude1,"-",exclude2,"-",exclude3,"-",exclude4,"-",exclude5,"-",exclude6
          ,sep="")))
   }
73 cond <- cond-1
   if(cond.number<cond){
75    exclude7 <- covar[which(COVARIATES==0)][7]
      red.model <- eval(parse(text=paste("epsilon~con+fyll+temp+month+weekday+holy+summer+last
          .error.c-",exclude1,"-",exclude2,"-",exclude3,"-",exclude4,"-",exclude5,"-",exclude6
          ,"-",exclude7,sep="")))
77 }
   }
79 reduced.model <- red.model
   }
```

Listing 18: *Reduces the full OLS model to a model which only the given (input) explanatory variables are considered.*

```
   ## Funksjon for aa hente ut aar,mnd,dag,time fraa datostreng paa forma
2 ##   "fr 16-07-10:02", dvs. "ukedag dag-mnd-aar:time"
   ## time: 0-23
4
   ## datostreng = vektor med datoinformasjon paa forma over
6 ## hundreaar = hundreaar for dato, f.eks 1900, 2000 (kun siste to siffer i streng)
   ##   - ein verdi: alle i samme hundreaar
8 ##   - vektor med lengde lik length(datostreng): i ulike hundreaar
   redigerdato<-function(datostreng,hundreaar=2000)
10 {
       tmp1<-matrix(unlist(strsplit(datostreng,":")),ncol=2,byrow=T)
12     hour<-as.numeric(tmp1[,2])
14     tmp2<-matrix(unlist(strsplit(tmp1[,1]," ")),ncol=2,byrow=T)
       weekday<-tmp2[,1]
16
       datoar<-matrix(unlist(strsplit(tmp2[,2],"-")),ncol=3,byrow=T)
18     year<-as.numeric(datoar[,3])+hundreaar
       day<-as.numeric(datoar[,1])
20     month<-as.numeric(datoar[,2])
22     return(list(year=year, month=month, day=day, hour=hour, weekday=weekday))
   }
```

Listing 19: *A code supplied by Turid Follestad (advisor). Takes as input a text string which represent a date and split in into an object containing the separate parts.*

```
1 ####################################################################################################
   ###################------------check_month_covariates.R---------#####################
3 ####################################################################################################
   foo <- function(a, b){
5    counter <- 0
     for(word in 1:length(b)){
7      a <- strsplit(a, "")[[1]]
       b.tmp <- strsplit(b[word], "")[[1]]
9      nmatch <- sum(outer(a, b.tmp, "=="))
11     if((length(a)) == nmatch) counter <- counter+1
     }
13 return(counter)
   }
```

Listing 20: *Checks the number of months in the OLS fitting. Used as control-mechanism to check that the right number of explanatory variables are considered.*

```r
################################################################################
####################----------initialize_data_frame.R-----------############
################################################################################
fgls <- function(DAY,MONTH,YEAR,HISTORY){
source("meancon.R")
source("meanfylling.R")
library("plm")
library("tseries")
library("fSeries")
source("clewlow_price.R")
source("retrive.r")
source("redigerdato.R")
data <- read.table("tidsserier_alle_2005-2010.txt",skip=0,header=T,sep="\t")
dato.info <-redigerdato(as.character(data[,1]),hundreaar=2000)
fylling <- read.table("Fyllingsgrad_2002-2010.txt",skip=0,header=T,sep="\t")
index <- which(dato.info$year==2008)
index2 <- which(dato.info$year==2007)
price <- data[union(index,index2),2]
skuddaar <- 366
normalaar <- 365
na <- which(is.na(price))
price[na] <- 0.5*(price[(na+1)]+price[(na-1)])
tmp <- length(price)
lag1 <- 24
lag2 <- 168
lag3 <- 192
year=2008
year2 <- 2007
individual <- rep(0:23,(skuddaar+normalaar))
cross <- rep(1:(skuddaar+normalaar),each=24)
kalender <- redigerdato(as.character(data[which(dato.info$year==year),1]),hundreaar=2000)
kalender2 <- redigerdato(as.character(data[which(dato.info$year==year2),1]),hundreaar
=2000)
weekday <- factor(rep(rep(c(0,1,2,3,4,5,6),each=24),104))
tmp <- length(length(price)/24)%%104
weekday <- factor(c(weekday,rep(1,each=(24*(tmp+1)))))
month <- as.factor(c(kalender2$month,kalender$month))
levels(month) <- c(0,1,2,3,4,5,6,7,8,9,10,11)
holy <- factor(c(data[which(dato.info$year==year2),23],data[which(dato.info$year==year)
,23]))
levels(holy) <- c(0,1)
summer <- factor(c(data[which(dato.info$year==year2),29],data[which(dato.info$year==year)
,29]))
levels(summer) <- c(0,1)
temp <- c(data[index2,30],data[index,30])
con <- c(data[index2,13],data[index,13])
con <- as.vector(interpNA(con,"linear"))
tmp <- meancon()
tmp <- approx(x=tmp,y=NULL,method="linear",n=length(con))
scaledcon <- con/tmp$y-1
fyll <- c(rev(fylling[which(fylling[,1]==year2),3]),rev(fylling[which(fylling[,1]==year)
,3]))
fyll <- approx(x=fyll,y=NULL,method="linear",n=length(con))
tmp <- meanfylling()
tmp <- approx(x=tmp,y=NULL,method="linear",n=length(con))
scaledfyll <- fyll$y/tmp$y-1
temp <- as.vector(interpNA(temp,"linear"))
price <- price[-c(1:lag3)]
ind <- individual[-c(1:lag3)]
cross <- cross[-c(1:lag3)]
temp <- temp[-c(1:lag3)]
con <- scaledcon[-c(1:lag3)]
fyll <- scaledfyll[-c(1:lag3)]
month <- month[-c(1:lag3)]
```

```
61 weekday <- weekday[-c(1:(lag2))]
   holy <- holy[-c(1:lag3)]
63 summer <- summer[-c(1:lag3)]
   frame <- data.frame(price=price,temp=temp,ind=ind,cross=cross,con=con,fyll=fyll,month=
        month,weekday=weekday,holy=holy,summer=summer)
65 today <- which(dato.info$day==DAY&dato.info$month==MONTH&dato.info$year==YEAR)[1]-which(
        dato.info$day==1&dato.info$month==1&dato.info$year==2007)[1]
   price.c <- frame[(today-HISTORY*24+1-192):today,]$price
67 frame <- frame[(today-HISTORY*24+1):today,]
   c2007 <- clewlow.price(frame$price)
69 spikes <- c(c2007$spikes)
   spike.indicator <- vector(mode="numeric",dim(frame)[1])
71 spike.indicator[spikes] <- frame$price[spikes+1]-frame$price[(spikes)]
   frame$price[spikes] <- NA
73 frame <- cbind(frame,spike.indicator)
   lag1 <- 24
75 lag2 <- 168
   lag3 <- 192
77 tmp <- length(price.c)
   seq1 <- seq(lag3-lag1,tmp-lag1-1,1)
79 seq2 <- seq(lag3-lag2,tmp-lag2-1,1)
   seq3 <- seq(1,tmp-lag3,1)
81 seq2b <- rep((lag3-lag2):(tmp-lag2-1),each=24)+c(0:23)
   seq3b <- rep(1:(tmp-lag3),each=24)+c(0:23)
83 weekmedian <- apply(matrix(price.c[seq2b]-price.c[seq3b],nc=24,byrow=T),1,median,na.rm=T)
   x.hat <- price.c[seq1]+weekmedian
85 epsilon <- price.c[-c(1:lag3)]-x.hat
   lead.time <- rev(rep(1:(dim(frame)[1]/24),each=24)%%10)
87 frame <- cbind(epsilon,frame,lead.time)
   save(frame,file="frame.Rdata")
89 fgls <- frame
   }
```

Listing 21: *A code that reads the data file, and puts it into a data frame.*

```
   ####################################################################################################
2  ####################----------spike_identification.R----------####################
   ####################################################################################################
4  clewlow.price <- function(price){
   tmp <- which(is.na(price))
6  price[tmp] <- 0.5*(price[(tmp+1)]+price[(tmp-1)])

8  diff <- diff(price)
   n.spikes <- 0
10 dummy <- -2
   std.res <- sd(diff)
12 spikes <- NULL
   up <- NULL
14 while((n.spikes-dummy)>1){
     dummy <- n.spikes
16   spikes <- union(which(abs(diff)>(3*std.res)),spikes)
     up <- union(which(diff>(3*std.res)),up)
18   n.spikes <- length(spikes)
     std.res <- sd(diff[-spikes])
20 }
   clewlow <- list(diff=diff,spikes=spikes,n.spikes=n.spikes,up=up)
22 }
```

Listing 22: *A code that identifies a spike according to the criterion given by Clewlow.*

```
1  ####################################################################################################
   ####################----------best_subset.R----------####################
3  ####################################################################################################
```

```r
  model.evalution <- function(HISTORY,NUMBER){
  start <- Sys.time()
  source("forecast.eval2.r")
  model <- matrix(1,nc=7,nr=24)

# Divide the hours into groups where the same
# covariates should be present
# group 1 Hour 0-6
# group 2 Hour 7-10
# group 3 Hour 11-15
# group 4 Hour 16-18
# group 5 Hour 19-23
  number.of.groups <- 5
  number.of.covariates <- 7
  group.start <- c(1,8,12,17,20)
  group.final.model <- NULL
  ev.criterion <- NULL
  model.cum <- NULL
  best.subset.cum <- NULL
  first.line <- forecast.eval2(model,HISTORY,NUMBER,1)
  best.subset.tmp <- first.line$CRPS
  base.line <- first.line$CRPS.baseline
  best.subset.last <- Inf
  print(paste("base line",base.line))
  for(group in 1:number.of.groups){
  print(paste("group",group))
    zeros <- rep(0,number.of.covariates)
    best.subset <- best.subset.last
    indicator <- NULL
    while(best.subset>best.subset.tmp){
      best.subset.last <- best.subset
      best.subset <- best.subset.tmp
      model.cum <- rbind(model.cum,model)
      for(i in 2:number.of.covariates){
      subset.cum <- rbind(best.subset.cum,best.subset)
        if(zeros[i]==0){
          if(group!=number.of.groups){
            model[c(group.start[group]:(group.start[group+1]-1)),i] <- 0
          }else{
            model[c(group.start[group]:24),i] <- 0
          }
          tmp <- forecast.eval2(model,HISTORY,NUMBER,0)$CRPS
          if(tmp < best.subset.tmp){
            best.subset.tmp <- tmp
            indicator <- c(indicator,i)
            print(indicator)
          }
          if(group!=number.of.groups){
            model[c(group.start[group]:(group.start[group+1]-1)),i] <- 1
          }else{
            model[c(group.start[group]:24),i] <- 1
          }
        }
      }
    }
      zeros[indicator] <- 1
      if(group!=number.of.groups){
        rows.l <- group.start[group+1]-group.start[(group)]
      }else    rows.l <- 24-group.start[group]+1
      change <- 1-matrix(zeros,nc=number.of.covariates,nr=rows.l,byrow=T)
      if(group!=number.of.groups){
        model[c(group.start[group]:(group.start[(group+1)]-1)),] <- change
      }else    model[c(group.start[group]:24),] <- change

  }
  print(paste("best subset tmp",best.subset.tmp))
```

```
69  print(model)
      group.final.model <- rbind(group.final.model,model)
71    ev.criterion <- rbind(ev.criterion,best.subset.tmp)
   }
73  end <- Sys.time()
    output <- list(crps=ev.criterion,best.covariates=group.final.model,start=start,end=end,
        subset.cum=subset.cum)
75  save(output,file="output2.RData")
    model.evaluation <-output
77  }
```

Listing 23: *A algorithm calculating the leave-one-out best-subset, using the Energy score as evaluation criterion.*

```
1  #############################################################################################
   #######################-----------energy_score.R-----------#####################
3  #############################################################################################
   forecast.eval <- function(COVARIATES,HISTORY,SAMPLES,PLUTO){
5  source("comforecast.r")
   source("CRPS.r")
7  DAY <- c(15)
   MONTH <-c(4)
9  YEAR <- c(2008)
   evaluation <- vector(mode="numeric",length=length(MONTH)*length(YEAR)*length(DAY))
11 if(PLUTO==1) evaluation.baseline <- evaluation
   for(time.d in 1:length(DAY)){
13   for(time.y in 1:length(YEAR)){
       for(time.m in 1:length(MONTH)){
15       t <- comforecast(DAY[time.d],MONTH[time.m],YEAR[time.y],COVARIATES,HISTORY,SAMPLES)
         price.obs <- t$real
17       price.fore <- t$forecast
         price.det <- t$det
19       evaluation[((time.y-1)*length(MONTH)+length(DAY)*(time.m-1)+time.d)] <- crps(price.
              fore,price.obs)
         if(PLUTO==1){
21         evaluation.baseline[((time.y-1)*length(MONTH)+length(DAY)*(time.m-1)+time.d)] <-
                crps(price.det,price.obs)
         }
23     }
     }
25 }
   CRPS <- (1/(length(MONTH)*length(YEAR)*length(DAY)))*sum(evaluation)
27 if(PLUTO==1){
     CRPS.baseline <- (1/(length(MONTH)*length(YEAR)*length(DAY)))*sum(evaluation.baseline)
29   forecast.eval.tmp <- list(CRPS=CRPS,CRPS.baseline=CRPS.baseline)
   }else{ forecast.eval.tmp <- list(CRPS=CRPS)}
31 forecast.eval <- forecast.eval.tmp
   }
```

Listing 24: *Calculates the Energy score of several forecasts. Inputs are the dates of the forecast, the covariates, the training period and the number of ensemble members.*