PREDICTING SUPPLY CURVE OF ELECTRICITY IN AN INTRA-DAY MARKET USING STATE-SPACE MODELS AND SEQUENTIAL MARKOV CHAIN MONTE CARLO METHODS

by
MOHAMMAD KHALAFI

Submitted to the Graduate School of Engineering and Natural Sciences
in partial fulfillment of
the requirements for the degree of Master of Industrial Engineering

Sabancı University
July 2021

PREDICTING SUPPLY CURVE OF ELECTRICITY IN AN INTRA-DAY MARKET USING STATE-SPACE MODELS AND SEQUENTIAL MARKOV CHAIN MONTE CARLO METHODS

Approved by:

Date of Approval: July 12, 2021

ABSTRACT

PREDICTING SUPPLY CURVE OF ELECTRICITY IN AN INTRA-DAY MARKET USING STATE-SPACE MODELS AND SEQUENTIAL MARKOV CHAIN MONTE CARLO METHODS

MOHAMMAD KHALAFI

Industrial Engineering, Master's Thesis, July 2021

Thesis Supervisor: Asst. Prof. Dr. Sinan Yıldırım

Keywords: Monte Carlo methods, Electricity market, Hidden Markov models, Sequential Markov chain Monte Carlo, Supply curve

In a free market, the price of a commodity is based on the relation between demand and supply. There is no exception in the case of the electricity market. Power companies use various load forecasting techniques to predict how much supply will be needed for a particular amount of demand. In the market equilibrium where demand and supply curves intersect, the price of a given commodity is realized. However, due to the special characteristics of the electricity market, daily and hour-by-hour prediction of the electricity price is more important. In this study, we will predict the daily supply curve of the electricity market in Turkey. These predictions are for each hour of the day. We have developed a hidden Markov model (HMM) to predict the supply curve in an intra-day market. The most popular approaches in dealing with the hidden Markov models or state-space models are sequential Monte Carlo methods (SMC) which are called *particle filtering* methods. However, in the case of high-dimensionality, standard particle filtering algorithms fail and are not efficient. In our article, the latent variables of the model are approximated by a sequential Markov chain Monte Carlo (SMCMC) method, which is an innovation in load forecasting, especially when dealing with a high-dimensional problem. We propose two different kernels for our algorithm to sample from the target distribution. Moreover, we use an expectation-maximization (EM) algorithm to update the hyperparameters of the model, such as the variances of latent variables and observations in our hidden Markov model.

ÖZET

DURUM-UZAY MODELLERİ VE SIRALI MARKOV ZİNCİRİ MONTE CARLO YÖNTEMLERİ KULLANILARAK BİR GÜN İÇİ PİYASADA ELEKTRİK ARZ EĞRİSİNİN TAHMİN EDİLMESİ

MOHAMMAD KHALAFI

Endüstri Mühendisliği, Yüksek Lisans Tezi, Haziran 2021

Tez Danışmanı: Asst. Prof. Dr. Sinan Yıldırım

Anahtar kelimeler: Monte Carlo yöntemleri, Elektrik piyasası, Saklı Markov modelleri, Sıralı Markov chain Monte Carlo, Arz eğrisi

Serbest piyasada, bir malın fiyatı arz ve talep arasındaki ilişkiye dayanır. Elektrik piyasasında bir istisna yoktur. Enerji şirketleri, belirli bir talep miktarı için ne kadar arzın gerekli olacağını tahmin etmek için çeşitli yük tahmin teknikleri kullanır. Talep ve arz eğrilerinin kesiştiği piyasa dengesinde, belirli bir malın fiyatı gerçekleşir. Ancak elektrik piyasasının kendine has özelliklerinden dolayı elektrik fiyatının günlük ve saatlik tahmini daha önemlidir. Bu çalışmada Türkiye elektrik piyasasının günlük arz eğrisi tahmini yapılacaktır. Bu tahminler günün her saati içindir. Bir gün içi piyasada arz eğrisini tahmin etmek için bir saklı Markov Modeli geliştirdik. Saklı Markov Modelleri veya Durum-Uzay modelleri ile ilgili en popüler yaklaşımlar, *parçacık filtreleme* yöntemleri olarak adlandırılan Sıralı Monte Carlo yöntemleridir. Ancak yüksek boyutluluk durumunda standart parçacık filtreleme algoritmaları başarısız olur ve verimli olmaz. Makalemizde, Modelin saklı değişkenleri, özellikle yüksek boyutlu bir problemle uğraşırken yük tahmininde bir yenilik olan Sıralı Markov chain Monte Carlo yöntemi ile tahmin edilmektedir. Algoritmamızın hedef dağılımdan örneklenmesi için iki farklı çekirdek öneriyoruz. Ayrıca, saklı Markov modelimizde saklı değişkenlerin varyansları ve gözlemler gibi modelin hiper parametrelerini güncellemek için bir beklenti-maksimizasyon algoritması kullanıyoruz.

ACKNOWLEDGEMENTS

First and foremost, I would like to express my sincere gratitude for all the support and patient of my research supervisor Dr. Sinan Yıldırım. His patience, motivation, guidance, and profound knowledge were always inspiring for me. Dr.Yıldırım was a great advisor during my thesis and a great mentor in my life.

I want to thank all the help and support of great people in ***Technix GİP*** especially Dr. Tayyar Guzel. His help in providing a valuable data set and his advice on the technical issues was a crucial part of this thesis.

I would also like to express my appreciation to the jury members, Dr Kemal Kılıç and Dr. Ilker Birbil, for their insightful comments to improve the quality of my work.

My next appreciation goes to all the faculty members at Sabanci university who helped me enhance my experience in the academic area during the last two years. Their patient and support during my study in Sabanci was a true motivation for me.

Last but not least, I want to thank my family for their endless support and kindness. They are the most important reason why I have made this path till now.

To my family

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

i.i.d: Independent and identically distributed

MCMC: Markov chain Monte Carlo

MH: Metropolis-Hastings

SMC: Sequential Monte Carlo

SIS: Sequential importance sampling

SISR: Sequential importance sampling resampling

HMM: Hidden Markov models

HO-HMM: Higher order Hidden Markov models

HSMM: Hidden semi-Markov models

FFBS: Forward filtering backward smoothing

EKF: Extended Kalman filtering

UKF: Unscented Kalman filtering

DKF: Discriminative Kalman filtering

SMCMC: Sequential Markov chain Monte Carlo

BSISR: Block sequential importance sampling resampling

EM: Expectation maximization

EPF: Electricity price forecasting

ANN: Artificial neural networks

MCP: Market clearing price

ML: Machine learning

1. INTRODUCTION AND LITERATURE REVIEW

Electricity price prediction has increasingly become one of the fundamental aspects for the companies to set their strategies. Despite some differences in electricity pricing, the nature of load forecasting stems from the relationship between supply and demand curves. These two curves are not necessarily dependent on each other, and multiple factors can affect each curve. Some factors that can influence the demand and supply of electricity can be temperature, prices of other energy resources, life quality of the population, solar radiation, wind speed, level of technology in the energy and electricity industry, and special days. The way these factors cause the demand and supply is not the same. For example, on holidays, electricity generation slows down, but the energy demand may increase.

In this thesis, we will focus on the daily supply curve of electricity for each hour. In the electricity market, the supply curve is obtained by summating the supply curves of different energy sources. We can mention some energy resources: solar power, wind energy, fossil fuels, thermal energy, steam turbines, tidal energy. In this study, we have broken down the supply curve of energy into six different resources. We developed a piece-wise linear function for each resource where each source's supply curve depends on some latent variables in a hidden Markov model. The latent variables of the HMM are approximated by a sequential Markov chain Monte Carlo algorithm. An expectation-maximization algorithm estimates the hyperparameters of the model.

1.1 Scope of the research and contribution

As we will discuss in the literature review, most of the works in electricity price forecasting (EPF) are done using artificial neural networks (ANN). However, to the best of our knowledge, we are using a novel method in dealing with load forecasting

problems. In addition, our model uses many parameters that capture the characteristics of the electricity supply curve. These factors can contribute to better resource prediction. However, having a lot of factors can cause some problems. To tackle these problems, we incorporated an SMCMC method which can reduce these issues in the approximation. Moreover, we presented two different approaches in our SMCMC algorithm. In the first approach, we use Metropolis-Hastings (MH) moves for the updates of the algorithm, and in the second approach, we use a combination of Gibbs and MH moves in our algorithm steps. We will mention some of the related works that tend to predict the price of electricity by using various methods. Just a few of them use a prediction based on Monte Carlo methods, and none of them used an SMCMC method where the dimensionality of the model is high. In the following, we will elaborate on some of the popular techniques that are used for EPF.

The rest of the thesis is organized as follows: In Chapter 2, we will give an overview on Monte Carlo methods. In Chapter 3, we will introduce the hidden Markov models and some of the approaches to approximate the latent variables. Chapter 4 will discuss the model which we developed for the supply curve as the summation of supplies from 6 different resources and a constant value which will be added to that summation. In this chapter, we will also mention the results for two different scenarios, and lastly, we will state our conclusion about the thesis and related future researches on this subject in Chapter 5.

1.2 Literature review

1.2.1 Time frame

Load forecasting is the predicting of electrical power required to meet the short-term, medium-term or long-term demand. The forecasting helps the utility companies in their operation and management of the supply to their customers. In the literature electricity price forecasting is broken into three categories:

- long-term forecasting
- mid-term forecasting
- short-term forecasting

Long-term forecasting is aimed for planning purposes [Hong, 2014]. Spatial load forecasting determines how the electricity market will develop and is used for transmission and distribution planning. After companies' focus on operational performance, short-term forecasting started to grow gradually [Hong et al., 2020]. Besides, due to the difference between the electricity market and other commodities, short-term forecasting has a particular interest for the companies. Unlike the other commodities, the electricity market spot price is a day-ahead market and is not continuous. The reason behind this is that the system operators require advance notice to make sure that the schedule is feasible [Weron, 2014]. In the literature, there are some works related to mid-term forecasting. Compared to the short-term electricity market clearing price (MCP), there have been few works for the mid-term forecasting (from one month to 6 months) of electricity MCP. The main difference between short-term forecasting and mid-term forecasting is that in the latter one, we cannot use the trends from the immediate past, while it is the case in short-term forecasting. However, in the midterm forecasting of the electricity MCP, because of the unavailability of data from the immediate past, locating the peak prices becomes extremely difficult [Yan & Chowdhury, 2015]. As we mentioned earlier, the short-term framework is much more extensive than other time frames in the probabilistic approach. One of the popular papers that concentrates on short-term load forecasting is Hong [2010] where the techniques, the variables being used, and the representative work being done by several major research groups are discussed. In the long-term time frame, we can allude to other works such as Hong et al. [2013] where we can see

a linear regression model for a short-term time frame augmented with a macroeconomic indicator in a probabilistic approach. The results showed that the models based on hourly data had smaller forecasting errors than those found on monthly or daily data.

To sum up, due to the regulation and the particular structure of the electricity market, price forecasting has attracted much interest, and most of the work has been done in the short-term time frame.

1.2.2 Energy resources prediction

The most critical resources in our model to generate electricity are wind, solar radiation, hydro, coal (imported, domestic), and gas.

To investigate wind power, we have to study wind speed as a dominant factor. One paper that takes the irregularity of the wind speed into account and presents the forecasting techniques for the wind speed, such as statistical methods, ANN approaches, and hybrid techniques for different time frames is Soman et al. [2010]. Moreover, there is another paper about wind power prediction focused on short-term forecasting for three decades. In this paper, the two schools of thought (statistical and physical) are used simultaneously [Costa et al., 2008]. As we mentioned before, wind power had been studied in two different approaches. These approaches are statistical and physical. However, in recent papers, these two approaches have been considered together. ANN, as one of the popular approaches, has been used again for the wind power prediction in the electricity market [Filik & Filik, 2017]. Time series analysis and Kalman filter algorithm are another approaches to estimate wind speed [Pan et al., 2008].

The significant improvement in solar power estimation started in early 2010. Inman et al. [2013] provide a comprehensive study of solar forecasting and provides some successful applications of solar forecasting methods for both the solar resource and the power output of solar plants. Camera-based or satellite-based data are used for solar prediction in works such as Hammer et al. [1999]; Jang et al. [2016]; Lorenz et al. [2004]. Filik et al. [2017] used a polynomial model and minimum least square optimization method for predicting solar power for three different parts of the day (Morning, noon, night). In the literature, there are other approaches such as time series or fuzzy methods to predict the solar power [He et al., 2019; Jang et al., 2016]. However, the most common approach to forecast solar energy is ANN again [Prema

& Rao, 2015].

The next resource that is taken into account in our study is hydropower. This resource has a significant role in the electricity market of the countries that do not have abundant petroleum and natural gas resources. In Turkey, due to the high potential for using hydropower, it has attracted a lot of attention [Yuksek et al., 2006]. Monteiro et al. [2013] proposes a novel method named H4C for hydropower estimation for small hydropower plants. Statistical approaches are used to estimate hydroelectric power. For example, AR, ARIMA, and ARIMAX bivariate approaches are utilized in hydroelectric power application [Barzola-Monteses et al., 2019]. Johnsen [2001] uses hydro inflow, snow, and temperature conditions to explain spot price formation. Since the significant resource for electricity production is hydropower in Scandinavia, much of the studies have been there. Vehviläinen & Pyykkönen [2005] created a stochastic multi-factor model that takes into account the hydro flow and snow-pack developments.

The spot price in the electricity market depends on many factors such as temperature, winds speed, solar radiation and snow level. The other factor that plays a significant role in electricity price forecasting concerns fuel-related factors such as gas price and coal supply, including imported coal and domestic coal. The fuel-related factors are the factors where the regulations play a more important role and depend on how the regulations are acting in the market. This factors are mostly considered as fuel price and its impact on electricity prices [Aggarwal et al., 2009; Fan et al., 2007; Li et al., 2005].

1.2.3 Modeling approaches

To start with, we want to consider two main papers, Weron [2014] and [Hong et al., 2020], for our classification of the approaches. Therefore, here is our outline for the classification:

a. **Multi-agent and fundamental approaches**: These two categories are different. However, since they are less popular than other approaches, we merged them into one category. Multi-agent approaches mainly consist of agent-based simulation methods. The agents (companies) interact with each other to come to an equilibrium in supply and demand. Fundamental (structural) methods describe the price dynamics by modeling the impacts of significant physical and economic factors on the price of electricity.

b. **Reduced-form (quantitative, stochastic) models**: They characterize the overall statistical characteristics of electricity prices over time, focusing on derivative valuation and risk analytics.

c. **Statistical and probabilistic approaches**: They involve the direct application of statistical techniques in the electricity market. Moreover, they use econometric models in the context of electricity price forecasting.

d. **Computational intelligence techniques**: They combine elements of learning, evolution, and fuzziness to create approaches capable of adapting to complex dynamic systems. Machine learning and deep learning methods are some examples of these techniques.

1.2.3.1 Multi-agent and fundamental models

Multi-agent approaches are typically used for long-term and mid-term EPF. Equilibrium approaches may be viewed as generalizations of cost-based models with strategic bidding assumptions.The Nash-Cournot framework and supply function equilibrium are examples of these approaches. On the other hand, the increasingly popular adaptive agent-based simulation techniques can address features of electricity markets that static equilibrium models ignore. Agent-based computational economics has become a widely accepted approach to solving both theoretical and practical problems in energy economics [Guerci et al., 2010; Sensfuß et al., 2007; Kowalska-Pyzalska et al., 2014].

Agent-based simulations present many flexible tools where this feature can be considered an advantage and a disadvantage at the same time. We consider this freedom as a weakness because it needs all the assumptions in the simulation to be justified.

Fundamental models try to capture the primary physical and economic relationships present in the production and trading of electricity. Besides, other approaches such as non-linear programming and dynamic programming approaches use these fundamental factors in their EPF [Gonzalez et al., 2011; Karakatsani & Bunn, 2008; Kristiansen, 2012].

1.2.3.2 Reduced-form approaches

These approaches aim to capture the main features of the daily electricity market rather than having an accurate prediction for each hour. Reduced-form models generally do not forecast hourly prices precisely. Still, they are expected to recover the main characteristics of electricity spot prices, most of the time at the daily time scale. Such models provide a simplified yet reasonably realistic picture of the price dynamics and are commonly used for derivatives pricing and risk analysis [Benth et al., 2008; Eydeland & Wolyniec, 2002].

1.2.3.3 Statistical and probabilistic approaches

Reduced-form models are good at derivatives valuation and risk analytics. However, a model's simplicity can be a serious drawback. Statistical methods forecast the current price by previous or current values of exogenous factors, typically consumption and production figures, or weather variables in additive or multiplicative forms. Statistical models are attractive because some physical interpretation may be attached to their components, which is interesting for the system operators and engineers.

Probabilistic forecasts can be mentioned in the forms of distributions, intervals, etc. Generally, these approaches are done by point forecasts which are discussed in Gneiting [2011]. Among all of the resources used in electricity production, wind forecasting is the pioneer in using a probabilistic approach. To summarize the popular models in the context of statistical and probabilistic methods, we can refer to the regression models, autoregressive (AR) time series models, ARX time series models, GARCH models. Since our model uses a statistical approach, we will delve into this approach in the following section in a more specific manner.

1.2.3.4 Computational intelligence approaches

Recent developments in artificial intelligence (AI) and machine learning (ML) approaches have made this field more popular than before. Expert systems, ANN, and genetic algorithms are some examples of these approaches [Metaxiotis et al., 2003]. Machine learning techniques allow us to use a large amount of data in our decision processes. In energy and electricity, these algorithms and methods will enable us to predict when we need more power and give us this chance to adapt our energy policies. Various ML and AI algorithms such as reinforcement learning and deep learning have been adopted in energy forecasting. Each of these methods has some advantages and disadvantages. For example, the deep learning methods require a lot of time and are much more complex than the regression method. Moreover, deep learning methods are dependent on the computational power and the amount of collected data. Support vector regression is utilized to perform classifications in datasets whose output variable is continuous. Kavaklioglu [2011] developed a method based on support vector regression to predict the consumption of energy in Turkey. It made a model of each variable such as gross national product, imports and exports and used them to produce consumption prediction values. Another popular algorithm discussed in the energy market context is the k-nearest neighborhood due to its simplicity in problems, especially in high-dimensional situations. Random forest and Gaussian process regression are other examples of using Machine learning techniques in energy forecasting. The most important strength of using a computational intelligence approach is their ability to handle non-linearity in which they are better than statistical models.

1.2.4 A dynamic supply-demand model for EPF

A similar work to this study has been done in the statistical model context in Buzoianu et al. [2005]. In this paper, the electricity price is predicted based on the principle of supply and demand equilibrium. The model is a nonlinear non-Gaussian HMM. The supply and demand curves are a function of some explanatory variables and some unknown variables which are assumed to be the latent variables of the HMM. It was also assumed that supply and demand curves were not observable but the prices and quantities traded were. The data used for the model is the California electricity market from 1998 to 2000, which includes the "California power crisis" in which we experienced extremely high prices. The approach for parameter estimation

stems from particle filter methods that fit the data suitably expect during the crisis period. Supply and demand curves are functions of the explanatory variables for a quantity with a vector of parameters in a Markov process. The paper considers a state-space model in which the supply and demand curves are the latent variables. Quantities and prices at equilibrium are the observed variables in the model. Since the number of parameters in the model that need to be estimated is not much (9 parameters), a simple particle filter algorithm using weighted samples is working fine (the during-crisis data, which is out of the scope of our discussion here). However, in our model, we are dealing with a high-dimensional model. The number of parameters to estimate is much more (approximately 30 parameters). Therefore, the simple particle filter techniques are not working as well as the California electricity market. In this thesis, we will propose a sequential Markov chain Monte Carlo algorithm to remedy this problem.

2. MONTE CARLO METHODS

2.1 Introduction

Monte Carlo methods are a set of computational algorithms which use random samples from a given distribution to solve a mathematical or statistical problem. Here are the main steps in Monte Carlo methods:

a. Generating a finite number of independent and identically distributed (i.i.d) random numbers followed from a given distribution with a probability density or mass function $\pi(x)$.

$$X^{(1)}, X^{(2)}, ..., X^{(N)} \overset{i.i.d}{\sim} \pi(x).$$

b. Calculating the expectation of a general function in the domain of the generated random numbers. This expectation can be approximated by the average of realized values of the function.

$$\mathbb{E}_\pi(\varphi(X)) = \int_{\mathcal{X}} \varphi(x)\pi(x)dx \approx \frac{1}{N}\sum_{i=1}^{N} \varphi(X^{(i)}) \tag{2.1}$$

One can show that Monte Carlo methods give us an unbiased estimation of $E_\pi(\varphi(X))$. Moreover, it almost surely converges to the real expectation, and the variance of the estimation decreases as the sample size increases.

In some cases, we can generate random numbers from the target distribution π, using exact sampling methods such as composition methods, transformation methods, inversion, and rejection sampling. However, sometimes, π is unknown, or drawing samples from π is challenging and expensive. In that case, we resort to approximate

sampling methods, among which the most popular ones are known as Markov chain Monte Carlo (MCMC) methods.

2.2 Markov chain Monte Carlo

Markov chain Monte Carlo methods are the most common class of methods for approximate sampling from complex distribution. The core idea behind MCMC methods is to construct an ergodic Markov chain whose stationary distribution is the target distribution $\pi(x)$ from which we want to sample. If we run an MCMC algorithm long enough, its Markov chain eventually converges to the target distribution π. In the following sections, we will elaborate on some of the most popular MCMC algorithms, including Metropolis-Hastings and Gibbs methods.

2.2.1 Metropolis-Hastings

The most commonly used MCMC algorithm in the literature is the Metropolis-Hastings. This algorithm was first introduced in Metropolis et al. [1953] and later on, Hastings [1970] developed it in more general context. A single iteration of MH is as follows:

In MH, we need a Markov transition kernel $q(\cdot|x)$ to generate the proposal values.

We propose x' based on the previous sample $X_{t-1} = x$. This proposed value follows from a transition density $q(\cdot|x)$ which is accepted as $X_t = x\prime$ with the following probability:

$$\alpha(x, x') = \min\left\{1, \frac{\pi(x')q(x|x')}{\pi(x)q(x'|x)}\right\} \quad x, x' \in \mathcal{X}$$

If the proposal is not accepted we stick to the previous sample and set $X_t = x$.

In Algorithm 1, we will present the MH algorithm.

Algorithm 1: Metropolis-Hastings algorithm

Begin with an initial value as $x_0 \in \mathcal{X}$
for $t = 1, \ldots, T$ **do**
    Sample $x' \sim q(x'|x_{t-1})$
    Set $x_t = x'$ with probability: $\alpha(x, x')$; else set $x_t = x_{t-1}$

One can prove that the stationary distribution of this algorithm is indeed π. In the literature, there are many approaches in terms of choosing the proposal distribution. For example, one can select a proposed sample independent of the previous sample. Another approach is to use a random walk distribution for proposing new samples. In the literature, there are some works in which we can use an adaptive proposal distribution in MH algorithm [Haario et al., 2001; Griffin & Walker, 2013].

2.2.2 Gibbs sampling

Another popular in MCMC algorithm is the Gibbs sampling. When the random variable which we are sampling has more than one dimension, MH sampling is not a efficient method. The idea behind the Gibbs sampling is that one can sample each component of the random variable $X \in \mathcal{X}^d$ once at a time by using the low dimensional conditional distribution of π. In other words, we are using $\pi_k(\cdot|x_{1:k-1}, x_{k+1:d})$ to sample $X_k \in X$ at a time where $\pi_k(\cdot|x_{1:k-1}, x_{k+1:d})$ is the full conditional distribution of the k'th component. After sampling from all the components of the random variable, we can move to the next iteration in the algorithm. In Algorithm 2, we present the Gibbs sampling algorithm.

Algorithm 2: Gibbs sampling method

Begin with an initial value
for $t = 1, \ldots, T$ **do**
    for $k = 1, \ldots, d$ **do**
        $x_{t,k} \sim \pi_k(\cdot|x_{1:k-1}, x_{k+1:d})$

2.2.3 Metropolis-Hastings within Gibbs sampling

One of the most effective sampling approaches in the context of MCMC which captures advantages of both methods(MH and Gibbs) is Metropolis-Hastings within Gibbs sampling. If we cannot obtain $\pi_k(\cdot|x_{1:k-1}, x_{k+1:d})$ one can use an MH move that targets this conditional density. In Algorithm 3 we show the Metropolis-Hastings within Gibbs sampling algorithm.

Algorithm 3: Metropolis-Hastings within Gibbs sampling

```
Begin with an initial value
for t = 1, ..., T do
    for k = 1, ..., d do
        update x_{t-1,k} to obtain x_{t,k} using an MH move that targets π_k(·|x_{1:k-1}, x_{k+1:d})
```

2.3 Bayesian framework

In the Bayesian inference context, we are primarily interested in finding a good approximation of our posterior distribution $p(x|y)$. Once we have a suitable approximation of our posterior distribution, we can identify other quantities of interest such as posterior mean, etc.

Consider two random variables X, Y, which are conditionally related to each other. For example, in hidden Markov models, X can be our latent variables, and Y is the observations. According to the Bayes theorem, one can construct the posterior distribution as follows:

$$\begin{aligned} p(x|y) &= \frac{p(x,y)}{p(y)} \\ &= \frac{p(y|x) \times p(x)}{p(y)} \\ &\propto p(y|x) \times p(x) \end{aligned} \tag{2.2}$$

In the Bayesian setting, we call $p(y|x)$ likelihood, $p(x)$ prior and $p(y)$ evidence. In other words, likelihood is the probability distribution of our data given our unknown parameters, and the prior is an assumption about the distribution of unknown variables. Sometimes, the posterior distribution is intractable, therefore we need to use

Monte Carlo methods to handle it.

3. HIDDEN MARKOV MODELS AND PARTICLE FILTERING

3.1 Introduction

We can allude to hidden Markov models as a modification of Markov chains. In addition to the Markovian relationship between the variables, we have observed variables that depend on our Markov chain. In many cases, we do not have access to the variables of our Markov chain. We call these variables *latent varibles* or *hidden variables*. Therefore, to estimate our hidden variables, we have to rely on our observations. In signal processing, image processing, intrusion and fraud detection and speech recognition [Picone, 1990; Mor et al., 2021; Nefian & Hayes, 1998],HMMs are one of the main modeling approaches. Moreover, we can see HMM applications in bioinformatics, finance, and weather forecasting in the literature [Krogh et al., 1994; Zhang, 2004; Khiatani & Ghose, 2017].

In this thesis, we focus on first-order hidden Markov models, which are the most common models in the context of HMM. In first-order hidden Markov model, each latent variable is dependent only on the previous latent variable, and the observations are conditionally independent given the latent variables. Figure 3.1 shows the relations between the latent variables and the observations in a first-order hidden Markov model.

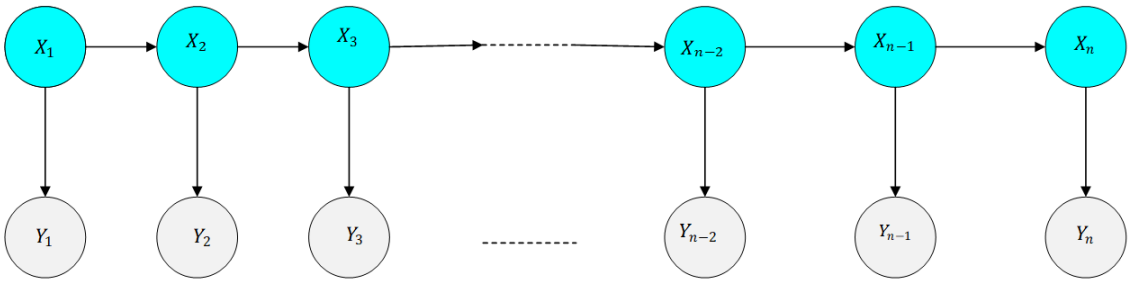

Figure 3.1 A scheme of a general hidden Markov model (frist order HMM)

3.2 Mathematical formulation

Suppose $\{X_t\}_{t\geq 1}$ be a homogeneous Markov chain defined in space $\mathcal{X}$. Assume the initial value $X_1 \sim \eta_1$ and the state transition density is $f(x_t|x_{t-1})$. Moreover, there is another process in which we obtain our observations $\{Y_t\}_{t\geq 1}$. In first-order HMM we assume that Y_t conditioned on all other random variables of the model only depends on X_t and follows $g(y_t|x_t)$. Hence, the stochastic process $\{X_t, Y_t\}_{t\geq 1}$ forms the hidden Markov model. Due to the structure of the HMM one can write the joint probability density as follows:

$$p(x_{1:n}, y_{1:n}) = \eta_1(x_1) \prod_{t=1}^{n-1} f(x_{t+1}|x_t) \prod_{t=1}^{n} g(y_t|x_t) \tag{3.1}$$

One of the important distributions in HMM is called the full posterior distribution. This distribution refers to the probability of observing latent variables $x_{1:n}$ given the observations $y_{1:n}$ up to time n. Using the Bayes theorem one can write the following:

$$\begin{aligned} p(x_{1:n}|y_{1:n}) &= \frac{p(x_{1:n}, y_{1:n})}{p(y_{1:n})} \\ &= \frac{\eta_1(x_1) \prod_{t=1}^{n-1} f(x_{t+1}|x_t) \prod_{t=1}^{n} g(y_t|x_t)}{\int p(x_{1:n}, y_{1:n}) dx_{1:n}} \end{aligned} \tag{3.2}$$

3.3 Posterior distribution in HMM

In a Bayesian framework, we are interested in calculating the posterior distributions in a HMM. In particular, we are interested in calculating the following:

$$p(x_{1:t'}|y_{1:n}) := \begin{cases} p(x_{1:n}|y_{1:n}) \prod_{\tau=n+1}^{n'} f(x_\tau|x_{\tau-1}) & n' \geq n \\ \int_{\mathcal{X}} p(x_{1:n}|y_{1:n}) dx_{n'+1:n} & n' \leq n \end{cases} \tag{3.3}$$

In the following, we will delve into three different cases in Bayesian optimal filtering based on sampling from the posterior distribution with respect to n.

3.3.1 Filtering, smoothing and prediction

In theory, the ideal approach that enables us to answer many quantities of interests in HMM is to identify $p(x_{1:n'}|y_{1:n})$. This distribution will give us all the needed information on the model. Nevertheless, due to computational complexity and other reasons, this distribution is hard to obtain. Sometimes, it suffices to consider the marginal posterior of a random latent variable X_t and we are interested in calculating the marginal posterior distribution $p(x_t|y_{1:n})$. In the literature there are three main cases for calculating posterior distribution $p(x_t|y_{1:n})$ with respect the relative position of t and n.

In the first case, if $t \leq n$ the identifying $p(x_t|y_{1:n})$ is called **smoothing**, if $t = n$ then obtaining $p(x_t|y_{1:n})$ is called **filtering** and if $t \geq n$ calculating $p(x_t|y_{1:n})$ is called **prediction**.

3.3.1.1 Forward calculation of filtering and prediction densities

Suppose the filtering posterior at step 1 is calculated as the following:

$$p(x_1|y_1) = \frac{\eta_1(x_1)g(y_1|x_1)}{\int \eta_1(x_1)g(y_1|x_1)dx_1}$$

The rest of the filtering and prediction densities can be obtained in a forward manner by recursion as

$$\begin{aligned} p(x_t|y_{1:t-1}) &= \int p(x_t, x_{t-1}|y_{1:t-1})dx_{t-1} \\ &= \int f(x_t|x_{t-1})p(x_{t-1|y_{1:t-1}})dx_{t-1} \end{aligned} \tag{3.4}$$

As one can observe, the prediction density can be calculated by knowing the filtering distribution from the previous step. Now, we aim to calculate the filtering distribution at step t. We have the filtering distribution as

$$\begin{aligned} p(x_t|y_{1:t}) &= \frac{p(x_t, y_{1:t})}{p(y_{1:t})} \\ &= \frac{g(y_t|x_t)p(x_t|y_{1:t-1})}{p(y_t|y_{1:t-1})} \end{aligned} \tag{3.5}$$

Note that $p(y_t|y_{1:t-1})$ does not depend on x_t so it can be treated as a normalizing constant which can also be written as

$$p(y_t|y_{1:t-1}) = \int g(y_t|x_t)p(x_t|y_{1:t-1})dx_t$$

The above equation can be used in calculating the evidence and predictive density with respect to Y_{n+1}. These two equations (3.4) (3.5) can be used recursively to obtain the filtering and predictive distribution up to time n. In the literature, these two related steps are called forward filtering and prediction.

Moreover, to alleviate the computational complexity of the algorithms that aim to track the posterior, some revised methods have been developed [Douc et al., 2011; Fearnhead et al., 2010]. Tracking the required posteriors can be broken into two main categories: first, tracking the exact posteriors in finite state-space models or in linear Gaussian HMMs and second, approximating the posterior using computational methods. Among popular methods for tracking the exact posterior distribution, one can refer to *Viterbi algorithm*, *Kalman filtering* and forward filtering backward sampling (FFBS) methods.

3.4 Sequential Monte Carlo

Sequential Monte Carlo methods are a set of algorithms or approaches that aim to approximate a sequence of distributions $\{\pi_n\}_{n\geq 1}$. SMC methods first were introduced in Liu & Chen [1998]. Let us assume $X_i \sim \pi_i$ for $i = \{1, ..., n\}$ and φ_n is a real valued function on domain of $\{X_n\}_{n\geq 1}$ with respect to π_n. Now we want to calculate the following quantity of interest:

$$\mathbb{E}_{\pi_n}\left[\varphi_n(x_{1:n})\right] = \int \pi_n(x_{1:n})\varphi_n(x_{1:n})dx_{1:n} \tag{3.6}$$

As one can see, the computation complexity increases by n. Therefore, the usual MCMC methods cannot handle the problem. The main focus of SMC approaches is to approximate this expectation. In state-space models, particle filtering approaches are one of the popular methods to track the posterior density. Like what we discussed in MCMC methods, there are similar approaches in SMC. Among these methods, we can mention sequential importance sampling (SIS) methods , sequential importance sampling resampling (SISR) methods. In the literature, there are some approaches when it comes to the resampling step in SMC methods.

3.4.1 Sequential importance sampling

Importance sampling in Monte Carlo methods is based on sampling from an instrumental distribution $q(x)$ instead of our target distribution π and weight the samples as the following:

$$w(x) := \begin{cases} \frac{\pi(x)}{q(x)} & q(x) \neq 0 \\ 0 & q(x) = 0 \end{cases} \tag{3.7}$$

Therefore, if we want to calculate $E_\pi(\varphi(X))$ where $X \sim \pi$, we can simply calculate $E_q(\varphi(X)w(X))$ where $X \sim q$. The key factor in importance sampling is to choose the instrumental distribution such a way that it is close to the target distribution.

Likewise, in SIS one can use the similar approach and weight the samples as $w_n(x) = \frac{\pi_n(x_{1:n})}{q_n(x_{1:n})}$. However, it is really expensive to sample from the whole sample path $x_{1:n}$ and approximate π_n by using q_n for each $n \geq 1$. Consequently, it is necessary to

reconstruct the instrumental distribution as follows:

$$q_n(x_{1:n}) = q(x_1)\prod_{t=2}^{n} q(x_t|x_{1:t-1})$$

Now we can reform the weights in recursive form:

$$\begin{aligned} w_n(x_{1:n}) &= \frac{\pi_n(x_{1:n})}{q_n(x_{1:n})} \\ &= \frac{\pi_n(x_{1:n})\pi_{n-1}(x_{1:n-1})}{q_n(x_{1:n})\pi_{n-1}(x_{1:n-1})} \\ &= \frac{\pi_n(x_{1:n})\pi_{n-1}(x_{1:n-1})}{q_{n-1}(x_{1:n-1})q(x_n|x_{1:n-1})\pi_{n-1}(x_{1:n-1})} \\ &= w_{n-1}(x_{1:n-1})\frac{\pi_n(x_{1:n})}{\pi_{n-1}(x_{1:n-1})q(x_n|x_{1:n-1})} \end{aligned} \tag{3.8}$$

In some cases, we do not have $\pi_n(x_{1:n})$ and consider it as $\pi_n(x_{1:n}) = \frac{\hat{\pi}_n(x_{1:n})}{C}$ where the constant is unknown. Hence by replacing the $\pi_n(x_{1:n})$ with $\hat{\pi}_n(x_{1:n})$ we have to normalize the weights. The self-normalized weights for N samples are

$$W_n^{(i)} = \frac{w_n(x_{1:n}^{(i)})}{\sum_{i=1}^{N} w_n(x_{1:n}^{(i)})} \tag{3.9}$$

Most of the time, the proposal distribution only depends on the last sample, not the whole sample path up to $n-1$. In the literature, there are some works in terms of choosing the best proposal distribution. Del Moral et al. [2006] showed the optimum proposal distribution in terms of reducing the variance is:

$$q^{opt}(x_n|x_{1:n-1}) = \pi_n(x_n|x_{1:n-1}) = \frac{\pi_n(x_{1:n})}{\int \pi_n(x_{1:n})dx_n} \tag{3.10}$$

One of the main problems of SIS methods is an issue called weight degeneracy. As we discussed, the best proposal distribution is the target distribution, and the closer these two distributions are to each other, the better performance we will have in sampling. However, if our proposal is not closed to the target distribution, we will end up with a few samples with high wights, and the rest of the samples will have negligible weights. Eventually, just one of the samples will have a high weight ≈ 1, and the rest's weight will be ≈ 0. To handle this problem, we have to develop a modification to make a resampling step in our SIS. This approach is called sequential importance sampling resampling.

3.4.2 Sequential importance sampling resampling

Roughly speaking, the sequential importance sampling resampling approach creates an equally weighted distribution by the weighted distribution which SIS generates. Suppose we have obtained $X_{1:n-1}^{(i)}$ with a array of weights such as $W_{n-1}^{(i)}$ for $i = 1, ..., N$. These samples are replaced by $\tilde{X}_{1:n-1}^{(i)}, i = 1, \ldots, N$ with equal weights and are drawn from $X_{1:n-1}^{(i)}, i = 1, \ldots, N$ as follows:

$$\mathbb{P}(\tilde{X}_{1:n-1}^{(j)} = X_{1:n-1}^{(i)}) = W_{n-1}^{(i)} \quad i, j = 1, \ldots, N$$

As one can observe, this corresponds to N draws from a multinomial distribution whose probability vector is $W_{n-1} = (W_{n-1}^{(1)}, W_{n-1}^{(2)}, ..., W_{n-1}^{(N)})$. This is called multinomial resampling. After doing the resampling step we will sample $X_n^{(i)} \sim q(X_n^{(i)} | \tilde{X}_{1:n-1}^{(i)})$ and continue as in SIS.

Although SISR reduces the weight degeneracy problem, it produces another problem which is similiar to weight degeneracy. After a while, the algorithm leads to generate many copies of the highly weighted samples. Eventually, we are left with one sample to approximate our target distributions. This problem is called *path degeneracy*. Moreover, Since we are introducing another sampling step, it creates more variance. In the literature, there have been some works to solve this problem to some extent. Doucet et al. [2006] uses an efficient way called *block sampling* to reduce the effect of *path degeneracy*. Moreover, we can mention alternative methods to standard multinomial sampling such as residual sampling, stratified sampling, and adaptive sampling methods [Fearnhead et al., 2013].

3.5 Sequential Monte Carlo in hidden Markov models

In section 3.4, we discussed SMC to track a sequence of distributions in general. We introduced SIS and SISR as two main popular SMC methods to track stochastic processes. One of the main applications of SMC methods in the literature is related to HMM models. In this section, we will show the application of SMC approach in HMM. Note that although there is a difference between *state-space model* and *hidden Markov model* in the literature [Fahrmeir & Tutz, 2001], we use these two terms as equivalent to each other.

Kalman filtering approaches are among the most popular methods to identify the posterior distribution when the state-space model is considered linear-Gaussian. However, it fails if we ignore the assumption that there is a linear relationship between the variables. In literature, there have been some attempts to modify the Kalman filtering approach to deal with non-linearity, such as extended Kalman filtering (EKF), which was first introduced in Kopp & Orford [1963]. EKF linearizes a nonlinear model. One flaw of this approach is the error related to the degree of non-linearity in the model. Unscented Kalman filter (UKF) is another approach which was presented by Julier & Uhlmann [1997]. UKF is similar to EKF in terms of using a linear Gaussian to approximate the latent Markov chain. However, this time a minimal set of carefully chosen sample points are used for the sampling and approximates the nonlinear HMM. This approach can also be used in the artificial neural network when there is non-linearity in the model [Wan & Van Der Merwe, 2000]. Besides, there has been some other modification in Kalman filtering called discriminative Kalman filtering (DKF). It enables us to have faster performances and also is easier to implement when the observation's dimensionality is greater than that of latent variable [Burkhart et al., 2016]. However, due to the deterministic nature, all these methods are deficient in a high degree of non-linearity. On the other hand, SMC methods have a good performance in highly nonlinear models compared to the deterministic models.

In the following, we will show how SMC methods can be applied to HMM. According to the structure of HMM, one can show that the filtering path of latent variables given the observation up to time t can be reformulated as a recursive equation. In other words, we can consider $p(x_{1:n}|y_{1:n})$ as:

$$\begin{aligned} p(x_{1:n}|y_{1:n}) &= \frac{p(x_{1:n}, y_{1:n})}{p(y_{1:n})} \\ &= \frac{f(x_n|x_{n-1})g(y_n|y_{n-1})p(x_{1:n-1}|y_{1:n-1})}{p(y_n|y_{1:n-1})} \end{aligned} \tag{3.11}$$

Now let us consider our target distribution to be $p(x_{1:n}|y_{1:n})$

$$\pi_n(x_{1:n}) = p(x_{1:n}|y_{1:n}) \quad n \geq 1$$

There are many options for choosing the proposal density. For example one can assume that the proposed particle is only dependent to the last latent variable $q(x_t|x_{t-1}) = f(x_t|x_{t-1})$ (*bootstrap filtering*) [Gordon et al., 1993]. Another option is to choose the proposal distribution independent from the other variables but the observations $q(x_t) = p(x_t|y_t)$ when the observations provide significant information about the latent variables (*independent particle filter*) [Lin et al., 2005]. A more

sophisticated way to construct the proposal distribution is to choose $q(x_{1:n}|y_{1:n})_{n\geq 1}$ is as follows:

$$q(x_{1:n}|y_{1:n})_{n\geq 1} = p(x_1|y_1)\prod_{t=2}^{n} q(x_t|x_{t-1}, y_{1:n}) = p(x_1|y_1)\prod_{t=2}^{n} q(x_t|x_{t-1}, y_{1:t})$$

In the literature we use $q(x_t|x_{t-1}, y_t)$ instead of $q(x_t|x_{t-1}, y_{1:t})$. Therefore, the proposal distribution and can be simplified as

$$q(x_{1:n}|y_{1:n}) = p(x_1|y_1)\prod_{t=2}^{n} q(x_t|x_{t-1}, y_t) \quad n = 1, \dots$$

According to the (3.8) Since our target distribution is the filtering path distribution we can show the incremental weights as:

$$\begin{aligned}
\frac{\pi_n(x_{1:n})}{\pi_{n-1}(x_{1:n-1})q(x_n|x_{1:n-1})} &= \frac{p(x_{1:n}|y_{1:n})}{p(x_{1:n-1}|y_{1:n-1})q(x_n|x_{n-1}, y_n)} \\
&= \frac{p(x_{1:n}, y_{1:n})p(y_{1:n-1})}{p(x_{1:n-1}y_{1:n-1})p(y_{1:n})q(x_n|x_{n-1}, y_n)} \\
&= \frac{p(x_{1:n}, y_{1:n})}{p(x_{1:n-1}y_{1:n-1})p(y_n|y_{1:n-1})q(x_n|x_{n-1}, y_n)} \\
&= \frac{f(x_n|x_{n-1})g(y_n|x_n)}{p(y_n|y_{1:n-1})q(x_n|x_{n-1}, y_n)}
\end{aligned} \tag{3.12}$$

Since calculating the conditional evidence $p(y_n|y_{1:n-1})$ is expensive and its density is free of x_n, we can use the self normalized version of incremental weights.

$$w_{n|n-1}(x_n|x_{n-1}, y_n) = \frac{f(x_n|x_{n-1})g(y_n|x_n)}{q(x_n|x_{n-1}, y_n)} \tag{3.13}$$

In Algorithm 4, we present the SISR algorithm also known as the particle filter, for a generic HMM

Algorithm 4: SISR (particle filtering) for HMM

for $i = 1, \ldots, N$ **do**

  Begin with an intial sample $X_1^{(i)} \sim q(x_1)$ and calculate the initial weight

$$w_1(x_1^{(i)}) = \frac{\eta_1(x_1^{(i)}) g(x_1^{(i)} | Y_1 = y_1)}{q(x_1^{(i)} | Y_1 = y_1)}$$

for $i = 1, \ldots, N$ **do**

$$W_1^{(i)} = \frac{w_1(x_1^{(i)})}{\sum_{i=1}^{N} w_1(x_1^{(i)})}$$

for $t = 2, 3, \ldots, n$ **do**

  Resample from $X_{1:t-1}^{(i)}$ for $i = 1, 2, ..., N$ based on their weights W_t^i and obtain $\tilde{X}_{1:t-1}^{(i)}$ with equal wights $\frac{1}{N}$

  for $i = 1, \ldots, N$ **do**

    Sample $X_t^{(i)} \sim q(x_t^{(i)} | \tilde{x}_{1:t-1}^{(i)})$

$$w_t(x_{1:t}^{(i)}) = \frac{1}{N} \frac{f(x_t^{(i)} | x_{t-1}^{(i)}) g(y_t | x_t^{(i)})}{q(x_t^{(i)} | x_{t-1}^{(i)}, y_t)}$$

    for $i = 1, \ldots, N$ **do**

      Calculate the self-normalized weight as (3.9)

$$W_t^{(i)} = \frac{w_t(x_{1:t}^{(i)})}{\sum_{i=1}^{N} w_t(x_{1:t}^{(i)})}$$

3.6 Sequential Markov chain Monte Carlo methods

As discussed previously, the most popular method to approximate the filtering distribution in non-linear non-Gaussian state-space models is SMC method or particle filtering. However, in some cases, such as dealing with high-dimensional models, this algorithm fails. The curse of dimensionality can be seen in a wide range of problems, such as weather forecasting. In these situations, we can use alternatives such as sequential Markov chain Monte Carlo methods. The literature results for these algorithms are promising and can be considered an alternative to standard SMC approaches. In the literature, there are some methods to alleviate the curse of dimensionality in state-space models. For example, one can apply a simple MH move on each particle after the resampling step. Another alternative is to break down the state-space model into a set of subspaces and do an SMC algorithm on each subspace. This approach is called block sequential importance sampling resampling (BSISR) [Doucet et al., 2006]. Furthermore, there are other methods called space-time particle filter (STPF) which, unlike the standard particle filtering algorithms, aim to move through space and time rather than just time [Beskos et al., 2014]. In recent years another promising method that seeks to remedy the problems that stem from SMC methods is called sequential Markov chain Monte Carlo [Septier & Peters, 2015]. It is important to note that, unlike the usual resample-move approaches, we do not use importance sampling or resampling steps in sequential Markov chain Monte Carlo methods [Septier et al., 2009]. The main reason that standard SMC methods fail in high-dimensional problems is weight degeneracy. In other words, we see poor performance of SMC approaches in high-dimensional problems for the same reason we observe inefficiency in dealing with issues with large time horizons. This weight degeneracy can be expressed as a quantity called *effective sample size* which can be defined as follows:

$$ESS_n = \frac{1}{\sum_{j=1}^{N} (W_n^{(j)})^2} \tag{3.14}$$

In SMCMC methods, we use an MCMC kernel to target the distribution $\pi_n(x_{1:n})$ which is usually the full posterior distribution. Since we do not have the exact full posterior $\pi_n(x_{1:n}) = p(x_{1:n}|y_{1:n})$, we approximate it by $\tilde{\pi}_n(x_{1:n})$. One can express the approximation to target distribution as follows:

$$\tilde{\pi}_n(x_{1:n}) = p(x_{1:n}|y_{1:n}) \propto g(y_{1:n}|x_{1:n}) f(x_n|x_{1:n-1}) \hat{\pi}_n(x_{1:n-1}) \tag{3.15}$$

where $\hat{\pi}_n(x_{1:n-1})$ is

$$\hat{\pi}_{n-1}(x_{1:n-1}) = \frac{1}{N} \sum_{j=M+1}^{M+N} \delta_{X^{(j)}_{n-1,1;n-1}} dx_{1:n-1} \tag{3.16}$$

Note that in (3.16) $\{X^{(j)}_{n-1,1;n-1}\}_{j=M+1}^{M+N}$ denotes to the last N particles which are drawn from the approximation of target density $\tilde{\pi}_n(x_{1:n-1})$ at time $n-1$.

The first M particles in SMCMC are generated during the burning time of the algorithm and will be discarded at the end of sampling. The SMCMC algorithm can be presented in Algorithm 5:

Algorithm 5: SMCMC in HMM

```
for i = 1, ..., M + N do
    Begin with an intial sample X^(i)_{1,1} ~ K_1(X^(i-1)_{1,1})
for n = 2, 3, ... do
    for i = 1, ..., N + M do
        Sample X^(i)_{n,1:n} ~ K_n(X^(j-1)_{n,1:n}) where K_1 is the MCMC kernel of
          invarient distribution π̃_n(x_{1:n-1}) as mentioned in (3.15)
```

To summarize Algorithm 5, one can say at the first step, we can sample from the initial distribution $\pi_1(x_{1:n})$ using the kernel $\mathcal{K}_1$. At iteration n we generate $M+N$ particles using the kernel $\mathcal{K}_n$. At the end of nth iteration, we discard M particles and keep the last N particles for approximating the filtering distribution:

$$p(x_n|y_{1:n}) = \frac{1}{N} \sum_{j=M+1}^{M+N} \delta_{X^{(j)}_{n,n}} dx_{n-1}$$

Note that the approximation of target distribution converges to the real target distribution:

$$\tilde{\pi}_n(x_{1:n-1}) \to \pi_n(x_{1:n-1}) \quad \text{As} \quad \hat{\pi}_{n-1}(x_{1:n-1}) \to \pi_{n-1}(x_{1:n-1})$$

Although there are some innovations in terms of choosing MCMC kernel [Septier & Peters, 2015], in the literature, most of the MCMC kernels used are based on Metropolis-Hasting updates. However, we show a combined kernel consisting of MH move and Gibbs sampling method in this study. The following section introduce a

non-linear non-Gaussian model and use an SMCMC algorithm to target the filtering distribution. First, we will use MH kernels as our MCMC kernel, and secondly, we will use an MH-based kernel and Gibbs-based kernel as our MCMC kernel.

3.6.1 A non-linear non-Gaussian state-space model

This section will show four different scenarios for a non-linear, non-Gaussian state-space model for different levels of non-linearities. These four methods are:

- Standard sequential Monte Carlo method
- Resample move approach: In this method, we make a MH move after resampling step in our SMC algorithm
- Sequential Markov chain Monte Carlo method using composition of Gibbs and MH sampling as the MCMC Kernel
- Sequential Markov chain Monte Carlo method using MH sampling as MCMC Kernel

3.6.1.1 Model formulation

First, let us present the latent Markov chain $\{X_n\}_{n\geq 1}$ as follows:

$$X_n = \frac{X_{n-1}}{2} + \frac{25X_{n-1}}{1+X_{n-1}^2} + 8\cos(1.2n) + e_n \tag{3.17}$$

$$\text{where } e_n \sim \mathcal{N}(0, \sigma_x^2)$$

The observation stochastic process $\{Y_n\}_{n\geq 1}$ is as the following:

$$Y_n = X_n + e_n \quad \text{where } e_n \sim \mathcal{N}(0, \sigma_y^2) \tag{3.18}$$

3.6.1.2 Results for SMC method

Table 3.1 Model, algorithm parameters and performance measures n=500

Parameters and Performance measures	**Exper 1**	**Exper 2**	**Exper 3**	**Exper 4**
σ_x^2	10	10	10	10
σ_y^2	10	10	10	10
n	100	100	100	100
N	100	1000	10000	100000
adjusted R^2	0.9240	0.9442	0.9540	0.8557
MSE	6.2794	6.59425	5.1276	4.6590

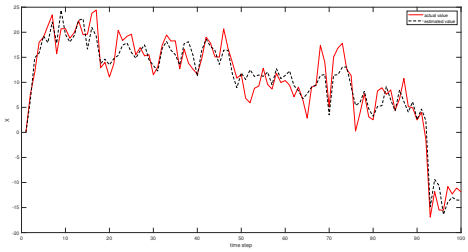

Figure 3.2 Non-linear non-Gaussian Model using SMC n=100 and N=100

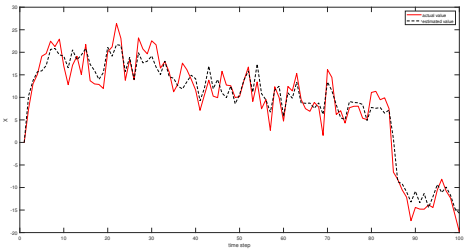

Figure 3.3 Non-linear non-Gaussian Model using SMC $n = 100$ and $N = 1000$

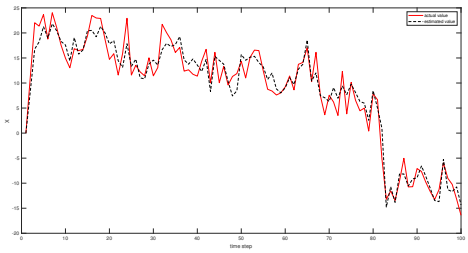

Figure 3.4 Non-linear non-Gaussian Model using SMC $n = 100$ and$N = 10000$

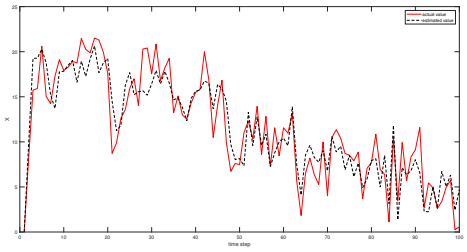

Figure 3.5 Non-linear non-Gaussian Model using SMC $n = 100$ and $N = 100000$

3.6.1.3 Results for SMC using resample move approach

Table 3.2 Model, algorithm parameters and performance measures n=100

Parameters and Performance measures	**Exper 1**	**Exper 2**	**Exper 3**	**Exper 4**
σ_x^2	10	10	10	10
σ_y^2	10	10	10	10
n	100	100	100	100
Number of MH moves	10000	10000	1000	100
N	100	1000	10000	100000
σ_q^2	1	1	1	1
adjusted R^2	0.9273	0.9447	0.9540	0.8558
MSE	6.0125	6.5242	5.1240	4.6569

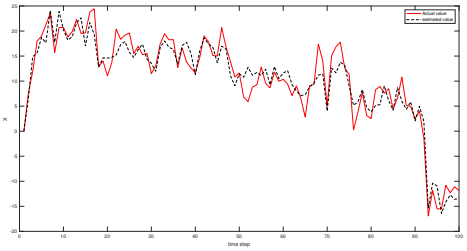

Figure 3.6 Non-linear non-Gaussian Model using SMC $n = 100$ and$N = 100$ (Resample Approach)

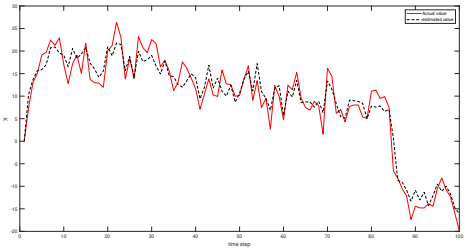

Figure 3.7 Non-linear non-Gaussian Model using SMC $n = 100$ and $N = 1000$ (Resample Approach)

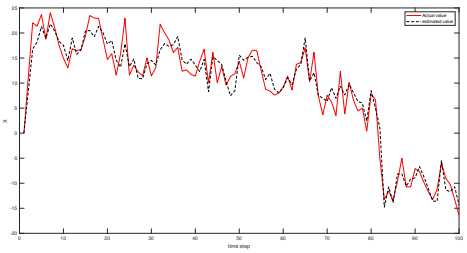

Figure 3.8 Non-linear non-Gaussian Model using SMC $n = 100$ and $N = 10000$ (Resample Approach)

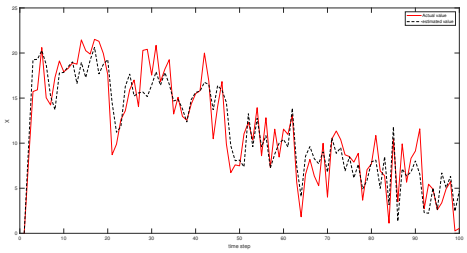

Figure 3.9 Non-linear non-Gaussian Model using SMC $n = 100$ and $N = 100000$ (Resample Approach)

As it can be observed from the results, since we are dealing with a model with a high degree of non-linearity, the algorithm's performance degrades. Now let us discuss the non-linearity Model in the sequential Markov chain Monte Carlo context.

3.6.1.4 SMCMC using Gibbs and MH sampling

In this section, we will not use the resampling step. Instead, we will use an MCMC kernel to sample from the filtering distribution up to $n = 100$. In SMCMC, one of the most important factors to good performance is choosing the proposal distribution close to the target distribution. In highly non-linear models having a poor choice of proposal distribution will result in bad performance. In the following, we will show the results for SMCMC with a kernel that is composed of Gibbs and MH updates

Table 3.3 Model, algorithm parameters and performance measures n=100

Parameters and Performance measures	**Exper 1**	**Exper 2**	**Exper 3**	**Exper 4**
σ_x^2	10	10	10	10
σ_y^2	10	10	10	10
n	100	100	100	100
M	1000	10000	15000	20000
N	100	1000	1500	2000
σ_q^2	1	1	1	1
adjusted R^2	0.8736	0.9548	0.9397	0.9421
MSE	7.1232	4.7218	4.4470	4.3015

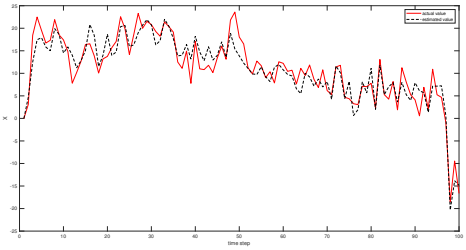

Figure 3.10 Non-linear non-Gaussian Model using SMCMC $n = 100, M = 1000$ and $N = 100$

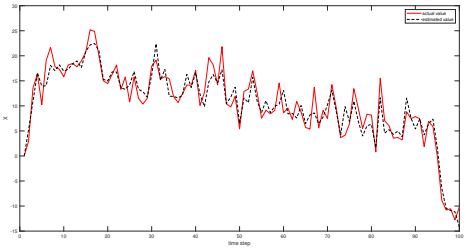

Figure 3.11 Non-linear non-Gaussian Model using SMCMC $n = 100, M = 10000$ and $N = 1000$

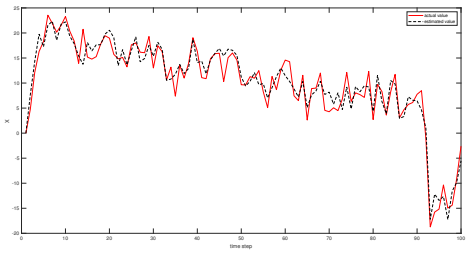

Figure 3.12 Non-linear non-Gaussian Model using SMCMC $n = 100$,$M = 15000$ and $N = 1500$

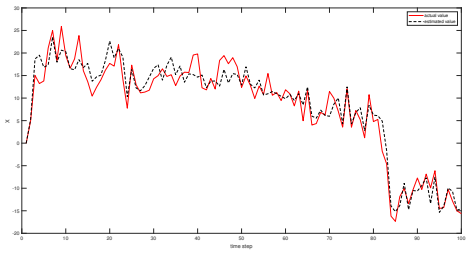

Figure 3.13 Non-linear non-Gaussian Model using SMCMC $n = 100$, $m = 20000$ and $N = 2000$

3.6.1.5 SMCMC using just MH sampling

Table 3.4 Model, algorithm parameters and performance measures n=100

Parameters and Performance measures	**Exper 1**	**Exper 2**	**Exper 3**	**Exper 4**
σ_x^2	10	10	10	10
σ_y^2	10	10	10	10
n	100	100	100	100
M	1000	10000	15000	20000
N	100	1000	1500	2000
σ_q^2	1	1	1	1
adjusted R^2	0.9003	0.9087	0.9379	0.9413
MSE	7.7144	4.8850	4.5843	4.4712

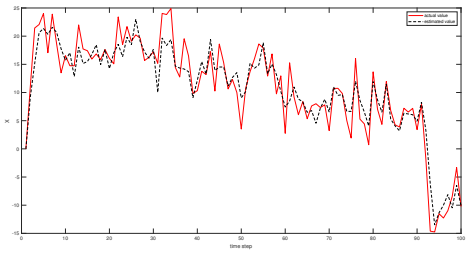

Figure 3.14 Non-linear non-Gaussian Model using SMCMC $n = 100$,$M = 1000$ and $N = 100$

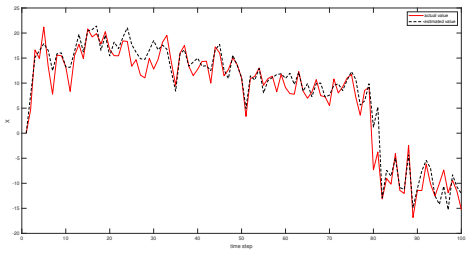

Figure 3.15 Non-linear non-Gaussian Model using SMCMC $n = 100$,$M = 10000$ and $N = 1000$

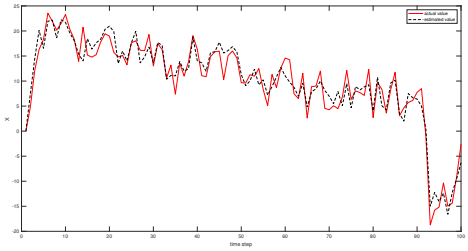

Figure 3.16 Non-linear non-Gaussian Model using SMCMC $n = 100$, $M = 15000$ and $N = 1500$

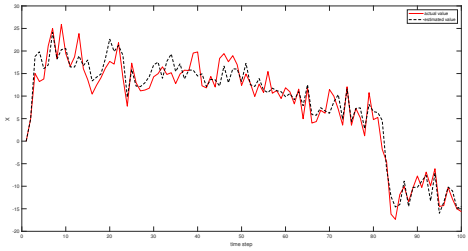

Figure 3.17 Non-linear non-Gaussian Model using SMCMC$n = 100$, $M = 20000$ and $N = 2000$

One can observe as we increase the number of particles, the performance of the algorithm improves (3.1) (3.2) (3.3) (3.4). As discussed earlier, SMCMC models are better choices if we are dealing with highly non-linear, high-dimensional models. The first reason is that they can act better in terms of accuracy when the model becomes high-dimensional. The second reason is that the computational complexity of the usual remedies of SMC models such as resample approach is such that they can not handle high-dimensional problems. Therefore, choosing alternatives such as SMCMC can be a good decision. in previous example, we have the hyperparameters vector $\Theta = (\sigma_x^2, \sigma_y^2, \mu_{initial}, \sigma_{initial}^2)$ in (3.17) and (3.18). In the real-life problems discussed in Chapter 4, we do not have information about the hyperparameters of the model. Therefore we need to use a hyperparameter estimation method to approximate them. In our model, we use *expectation-maximization* method, which is one of the most popular methods in state-space models to approximate the hyperparameters of the model. We will mention the EM algorithm to give an overview of how we are going to approximate the hyperparameters in a hidden Markov model

3.7 Expectation-maximization algorithm

When we have missing or hidden variables in our data, the usual parameter estimation approaches such as maximum likelihood estimation are not working. Therefore, we have to use other techniques that can estimate the hyperparameters of our model. Expectation-maximization algorithms are one of the most popular methods to approximate hyperparameters through a two-phase iterative algorithm. These methods can be used in the hidden Markov model based on the fact that finding the likelihood $p_\theta(y_{1:n})$ is challenging to obtain.

Algorithm 6: General expectation maximization algorithm

1 begin with a initial hyper parameter vector θ_0 **for** $t = 1, \ldots, T$ **do**

2 do the expectation step as the following:

$$E_{\theta^{t-1}}(\log p_\theta(x_{1:n}, y_{1:n})|y_{1:n})$$

do the maximization step as follows:

$$\theta^t = \arg\max_\theta E_{\theta^{t-1}}(\log p_\theta(x_{1:n}, y_{1:n})|y_{1:n})$$

Now let us focus of the case where our joint distribution is has an exponential. For exponential families, the expectation step reduces to calculating the posterior expectation of some additive summary statistics of the form [Dempster et al., 1977]

$$S(X_{1:n}, Y_{1:n}) = \sum_{t=1}^{n} s(x_{t-1}, x_t, y_t)$$

Consequently, the maximization step reduces to a function of those posterior expectations. In the rest of the thesis, we will introduce the application of SMCMC methods in the electricity market. We will present our model on the daily supply curve of electricity for each our of the day. And lastly, we will mention the results for using SMCMC predict the supply curve.

4. PROBLEM DEFINITION AND RESULTS

4.1 Model

As we discussed before, we propose a supply curve as the summation of supplies from different resources. These supply curves are a function of some of the explanatory factors and their coefficients for each time step. Moreover, since the coefficients are changing over time, the supply curves change over time. Since we deal with a non-linear model, the standard models of sequential monte Carlo are not efficient enough. Therefore, Markov chain Monte Carlo is a suitable method for modeling

In the following, we will delve into the details of the model and the summation of supply curves from the different sources that shape the total supply curve:

4.1.1 A dynamic supply model

For each time step, combine several supply curves (quantity vs price) $S_{t,1}, \ldots, S_{t,K}$ together: Let the inverse of the aggregate supply curve be S:

$$S_t(p) = S_{t,0} + \sum_{k=1}^{K} S_{t,k}(p), \quad p \in (0, p_{\max}). \tag{4.1}$$

Now we present a simple piece-wise function for each of the supply curves of resources $S_{t,k}(\cdot)$. We assume there is a minimum and maximum supply that a given resource can provide. Let the minimum and maximum quantities that can be supplied by source k be $q_{t,k}$ and $Q_{t,k}$, respectively. Given those variables, we model $S_{t,k}$ as a linear function between the points $(b_{t,k}, q_{t,k})$ and $(e_{t,k}, Q_{t,k})$, where $b_{t,k}$ is

the minimum price for the source to supply and $e_{t,k}$ is the minimum price for the maximum quantity, so that $m_{t,k} = (Q_{t,k} - q_{t,k})/(e_{t,k} - b_{t,k})$ is the slope of the line. To the left of the line, $S_{t,k}$ is 0 since the source does not supply for prices smaller than $b_{t,k}$. To the right of the line, $S_{t,k}$ is constant and has a value $Q_{t,k}$, which is the maximum amount the source can supply. Putting all those in mathematical terms, we have:

$$S_{t,k}(p;\beta_{t,k}) = \begin{cases} 0, & \text{for } p < b_{t,k} \\ q_{t,k} + m_{t,k}(p - b_{t,k}), & \text{for } b_{t,k} < p < e_{t,k} \\ Q_{t,k}, & \text{for } p \geq e_{t,k} \end{cases} \tag{4.2}$$

the function $\beta_{t,k} = (b_{t,k}, e_{t,k}, q_{t,k}, Q_{t,k})$ to indicate the parameters of $S_{t,k}$.

The function $\beta_{t,k}$ is itself defined as a function of some explanatory variables, or factors, and this function is defined through a $d_k \times 1$ vector of time-varying parameters, which we will denote as $x_{t,k} \in \mathcal{X}^{d_k}$ for some $d_k \geq 0$. Let

$$z_t = (z_{t,1}, \ldots, z_{t,I}) \in \mathbb{R}^I, \quad t \geq 1$$

note the vector of $I \geq 1$ explanatory variables at time t and the set $A_k \in \subseteq \{1, \ldots, I\}$ have the indices of those explanatory variables that have an impact on $\beta_{t,k}$. Furthermore, let the function $\varphi_k : \mathbb{R}^{|A_k|} \times \mathcal{X}^{d_k} \mapsto [0, \infty)^4$ be such that, given the parameter vectors at time for $x_{t,k} \in \mathcal{X}^{d_k}$, for $k = 1, \ldots, K$, the critical points of the supply functions are determined as

$$\beta_{t,k} = \varphi_k(z_{t,A_k}, x_{t,k}), \quad k = 1, \ldots, K. \tag{4.3}$$

where $z_{t,A_k} = (z_i : i \in A_k)$ is the subvector formed by those elements of $z_{t,k}$ whose indices are in A_k. The specific forms of φ_k, proposed to model the supply curve in the Turkish electricity market is given in the following Section 4.1.2.

4.1.2 HMM for the relation between the supply and latent variables

The explanatory variables $z_{t,k}$ are observed, whereas the time-varying latent parameters $x_{t,k}$ are unknown. The observed supply curves are also observed. We want to predict the supply curves of the future, for example, the supply curves of the next day for each hour. For prediction, the explanatory variables for the next step are either available or can be predicted with high accuracy. That is why the task of prediction of the supply curves of the following time steps requires tracking $x_{t,k}$'s as well as predicting $x_{t',k}$ for $t' > t$. Once $x_{t',k}$ is predicted, the supply curve can be predicted by using the relation in (4.3). Therefore, we formulate the problem as tracking the parameters $x_{t,k}$ given the supply curves as the observed variables$(p_{t,1}, Y_{t,1})$.

Let us combine the parameters $x_{t,k}$ of K sources from time t into a single vector as

$$x_t := (x_{t,1}, \ldots, x_{t,K}) \in \mathbb{R}^d.$$

where $d = d_1 + \ldots + d_K$.

We propose a hidden Markov model for modelling the supply curves in time. The hidden variables of HMM are $\{X_t\}_{t \geq 1}$ with the following transition

$$x_t = x_{t-1} + v_t, \quad v_t \sim \mathcal{N}(0, \Sigma) \tag{4.4}$$

where Σ is a $d \times d$ covariance matrix. The variable x_t, combined with the explanatory variables z_t, gives the lead to the critical points of supply curves as in (4.3).

By 'observed supply curves', we refer to a collection of (price, quantity) pairs, of size n_t,

$$(p_{t,1}, Y_{t,1}), \ldots, (p_{t,n_t}, Y_{t,n_t})$$

Note $p_{t,i}$ is the price, and $Y_{t,i}$ is the quantity offered at that price for the i'th pair. We assume that, given x_t, each $Y_{t,i}$ is independent and a noisy version of the supply curve in (4.1) calculated at $p = p_{t,i}$,

$$Y_{t,i} = \sum_{k=1}^{K} S_{t,k}(p_i; \beta_{t,k}) + w_{t,i}, \quad w_{t,i} \sim \mathcal{N}(0, \sigma_y^2), \quad i = 1, \ldots, n_t. \tag{4.5}$$

where $\beta_{t,k} = \varphi_k(z_{t,A_k}, x_{t,k})$ is the critical points of the supply curves of source k. Therefore, we have constructed the relationship between latent variables and the observations, which are the supply quantities. Now let us focus on the specific

structure of our model used for the electricity supply curve in Turkey. We are using six main resources that provide the supply. These resources are as the following: (i) solar, (ii) wind, (iii) domestic coal, (iv) imported coal, (v) gas, and (vi) hydro. Therefore, $K = 6$. The following table illustrates the relationship between each of those explanatory variables and the resources. These variables are: (i) temperature, (ii) precipitation, (iii) snow level, (iv) cloud cover, (v) wind speed, (vi) solar radiation, (vii) gas price, (viii) imported coal price (in TL), and (ix) domestic coal price. Hence, in total we have a vector $z_t \in \mathbb{R}^9$ of 9 explanatory variables at each time. Note that we also have the calendar effect in our data; however, it does not affect the model. Moreover, we express the explicit model that captures the supply curve of each of the six resources. These explicit models show how the explanatory and latent variables from the supply curve.

Table 4.1 Relation matrix sources of supply and the explanatory variables.

Relation	Symbol	Solar	Wind	Dom. coal	Imp. coal	Gas	Hydro
Temperature	T					1	
Precipitation	PC						1
Snow level	SL						1
Cloud cover	CC	1					
Wind Speed	WS		1				
Solar Radiation	SR	1					
Gas Price	GP					1	
Imported Coal Price (TL)	ICP				1		
Domestic Coal Price	DCP			1			

Table 4.1 shows the relation matrix between the sources of supply and the explanatory variables. A 1 for a (factor, source) pair indicates that the corresponding factor is involved in the modelling of the supply curve for the corresponding source.

The specific forms of the functions φ_k, showing the explanatory variables and parameters with the critical points of the supply curves, are given below.

Table 4.2 Functions for the critical points of the supply curves for the sources (time index is dropped for brevity)

Source	q_k	Q_k	b_k	e_k
Solar	0	$q_k + \mathrm{SR}^2 x_{sr2}^{\mathrm{solar}} + \mathrm{SR} x_{sr}^{\mathrm{solar}} + (100 - \mathrm{CC}) x_{sr}^{\mathrm{solar}}$	0	0
Wind	0	$q_k + \mathrm{WS} x_{ws}^{wind}$	0	0
Domestic coal	x_{min}^{dc}	$q_k + x_{base}^{dc}$	$DCP x_{mult}^{dc}$	$b_k + x_{slope}^{dc}(Q_k - q_k)$
Imported coal	x_{min}^{ic}	$q_k + ICP^{-x_{icp}^{ic}} x_{base}^{ic}$	$ICP x_{mult}^{ic}$	$b_k + x_{slope}^{ic} s_k (Q_k - q_k)$
Gas	x_{min}^{gas}	$q_k + (x_{temp}^{gas})^{T<-4} GP^{-x_{gp}^{gas}} x_{base}^{gas}$	$GP x_{mult}^{gas}$	$b_k + x_{slope}^{gas}(Q_k - q_k)$
Hydro	x_{min}^{hyd}	$q_k + (PC x_{pc}^{hyd} + SL x_{sl}^{hyd})$	x_{mult}^{hyd}	$b_k + x_{slope}^{hyd}(Q_k - q_k)$

All parameters are non-negative except $x_{t,sr}^{solar}$. and $x_{t,sr2}^{solar}$. Furthermore, we also have

$$0 < x_{t,temp}^{gas} < 1, \qquad t = 1, \ldots, n.$$

4.2 Application of SMCMC in supply curve

As we discussed earlier, SMCMC does not weigh the particles. Instead, it generates N equally weighted particles using MCMC moves. Given N particles from time $t-1$, which is denoted by

$$X_{t-1,1:t-1}^{(i)} = \left(X_{t-1,1}^{(i)}, \ldots, X_{t-1,t-1}^{(i)} \right), \quad i = 1, \ldots, N,$$

the algorithm generates a total of $M+N$ samples $\tilde{X}_t^{(1)}, \ldots, \tilde{X}_t^{(N+M)}$ for the posterior distribution $\pi(x_{1:t-1}) = p(x_{1:t-1}|y_{1:t-1})$, using an MCMC kernel $\mathcal{K}_{\tilde{\pi}_t}$ that targets the distribution $\tilde{\pi}_t$ defined as

$$\tilde{\pi}_t(x_{1:t}) \propto \frac{1}{N} \sum_{j=1}^{N} \delta_{X_{t-1,1:t-1}^{(j)}}(\mathrm{d}x_{1:t-1}) f_t(x_{t-1}, \mathrm{d}x_t) g_t(x_t, y_t).$$

The distribution $\tilde{\pi}_t$ is simply the particle approximation of $\pi_t(x_{1:t}) = p(x_{1:t}|y_{1:t})$. Since the MCMC kernel needs some time to converge to its invariant distribution, particles during a certain initial phase are discarded, and the remaining particles are treated as samples from the posterior π_t. We let M and N be the numbers of particles discarded and kept, which explains the $M+N$ samples. Therefore, the particles that are kept are:

$$X_{t,1:t}^{(i)} = \tilde{X}_t^{(M+i)}, \quad i = 1, \ldots, N.$$

Algorithm 7: Sequential MCMC for supply

for $t = 1, \ldots, T$ **do**
  Initialize $X_t^{(0)}$ based on $X_{t-1,1:t-1}^{(N)}$.
  Set $\tilde{X}_{t,1:t}^{(0)} = (X_t^{(0)}, X_{t-1,1:t-1}^{(N)})$.
  for $i = 1, \ldots, M+N$ **do**
    Update $\tilde{X}_{t,1:t}^{(i)} \sim \mathcal{K}_{\tilde{\pi}_t}(\tilde{X}_{t,1:t}^{(i-1)}, \cdot)$
  for $i = 1, \ldots, N$ **do**
    Set $X_{t,1:t}^{(i)} = \tilde{X}_{t,1:t}^{(M+i)}$.

4.2.1 Choices for the MCMC kernel

One choice for the kernel $\mathcal{K}_{\tilde{\pi}_t}$ is a succession of two MCMC kernels or one MCMC kernel for updating x_t and a GIBBS move for updating the first $t-1$ components $x_{1:t-1}$

$$\mathcal{K}_{\tilde{\pi}_t} = \mathcal{K}_{\tilde{\pi}_t}^{(2)} \mathcal{K}_{\tilde{\pi}_t}^{(1)}$$

where $\mathcal{K}_{\tilde{\pi}_t}^{(1)}$ updates the first $t-1$ components $x_{1:t-1}$ while fixing x_t and the second once, $\mathcal{K}_{\tilde{\pi}_t}^{(2)}$, updates x_t while fixing $x_{t,1:t-1}$.

We describe a pair of choices for $\mathcal{K}_{\tilde{\pi}_t}^{(1)}$ and $\mathcal{K}_{\tilde{\pi}_t}^{(2)}$.

- $\mathcal{K}_{\tilde{\pi}_t}^{(1)}$: Suppose the current sample is $x_{1:t}$. Fixing x_t, propose $x_{1:t-1}$ from $\frac{1}{N}\sum_{i=1}^{N} \delta_{X_{t-1,1:t-1}^{(i)}}(\mathrm{d}x_{1:t-1})$, that is, propose among $X_{t-1,1:t-1}^{(1)}, \ldots, X_{t-1,1:t-1}^{(N)}$ uniformly. If the proposed value is $X_{t-1,1:t-1}^{(j)}$, accept it with probability

$$\min\left\{1, \frac{f_t(X_{t-1,t-1}^{(j)}, x_t)}{f(x_{t-1}, x_t)}\right\} \tag{4.6}$$

  Another approach for $\mathcal{K}_{\tilde{\pi}_t}^{(1)}$ is that instead of proposing $x_{1:t-1}$ uniformly, we choose $x_{1:t-1}$ from the particle cloud based on the transition probability. Unlike the previous approach we have to consider $f_t(X_{t-1,t-1}^{(j)}, x_t)$ for all $X_{t-1,1:t-1}^{(1)}, \ldots, X_{t-1,1:t-1}^{(N)}$ and choose the particles with respect to that probability. This approach is called Gibbs move.

- $\mathcal{K}_{\tilde{\pi}_t}^{(2)}$: Suppose the current sample is $x_{1:t}$. Fixing $x_{1:t-1}$, propose x_t with a ran-

dom walk proposal by $x'_t \sim q(\cdot|x_{1:t-1})$. Accept proposed value with probability

$$\min\left\{1, \frac{f_t(x_{t-1}, x'_t)g_t(x'_t, y_t)q(x_t|x_{1:t-1})}{f(x_{t-1}, x_t)g(x_t, y_t)q(x'_t|x_{1:t-1})}\right\}$$

4.3 EM algorithm for hyperparameter estimation

As we discussed in chapter 3, in real-life problems, we do not have the hyperparameter vector of the model. In this specific model, our hyperparameter vector is $\theta = (\Sigma, \sigma_y^2)$ where Σ is the covariance matrix of the latent variables.For simplicity, we assume this matrix is diagonal. When the hyperparameter vector has many elements having a method to approximate them seems necessary. In our model, we are using exponential families as the proposed distributions. Therefore one can use sufficient statistics for the EM algorithm. In the following, we will show sufficient statistics. To avoid confusion, we need to mention that these statistics are different from the supply functions, and we use a similar notation for both of them.

$$\begin{aligned}
S_{1,i}(x_{1:n}) &= \sum_{t=1}^{n} x_{t,i}^2, \quad i = 1, \ldots, d_x \\
S_{2,i}(x_{1:n}) &= \sum_{t=1}^{n} x_{t-1,i}^2, \quad i = 1, \ldots, d_x \\
S_{3,i}(x_{1:n}) &= \sum_{t=1}^{n} x_t x_{t-1,i}, \quad i = 1, \ldots, d_x \\
S_{4,i}(x_{1:n}, y_{1:n}) &= \sum_{t=1}^{n} \sum_{i=1}^{n_t} [y_{t,i} - \hat{y}_{t,i}]^2 .
\end{aligned} \tag{4.7}$$

note that $\hat{y}_{t,i} = \sum_{k=1}^{K} S_{t,k}(p_i; \beta_{t,k})$.

4.4 Data

As we mentioned earlier, forecasting in the electricity market context usually falls into the short-term category. It is the case in our study as well. There is an auction for the electricity prices in the electricity market where the retailers and suppliers submit a price-quantity pair schedule for each hour of the following day. These schedules are gathered into a supply and demand curve to establish the equilibrium point, which is the supply and demand curve intersection. In our model, we used the data provided by ***Technix GİP***. The Data is comprised of hour-by-hour supply-price curves from 21/01/2020 to 29/2/2020. In figure 4.1, we show some of the supply-price curves for 24 hours.

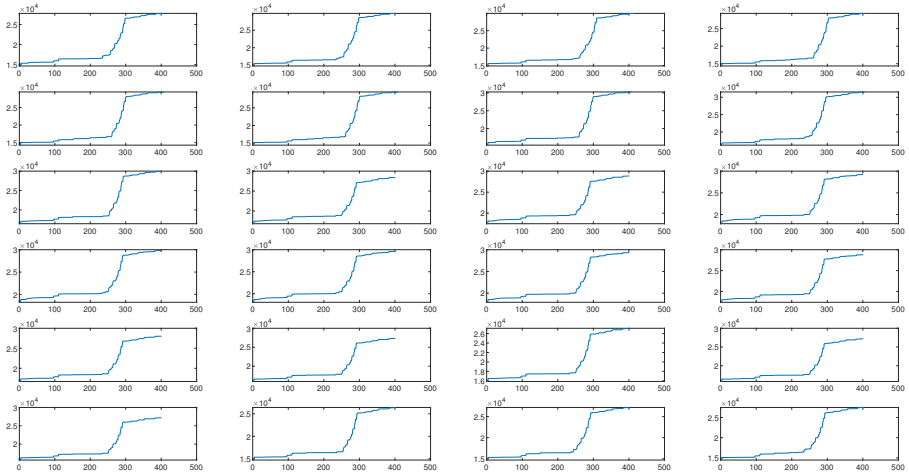

Figure 4.1 Supply-price curve for 24hours of a given day(21/1/2020)

Moreover, we need to point out that we had two different approaches to consider the calendar effect in our model. In one of them, we use a parameter $x^{\text{source}}_{t,\text{calendar effect}}$ to apply the calendar effect; another approach is to split the data into three different categories in terms of calendar effect:

- Weekdays

  Tuesday, Wednesday, Thursday

- Weekends and holidays

  Saturday, Sunday and any other public holiday

- Transition days which connect weekdays and weekends

Monday, Friday

Since we want to reduce the components of the latent variables vector, the latter approach makes more sense.

Moreover, In order to construct the functions φ_k we need to include the explanatory variables or $z_{t,k}$. In 4.2, you can see the explanatory variables for transition days.

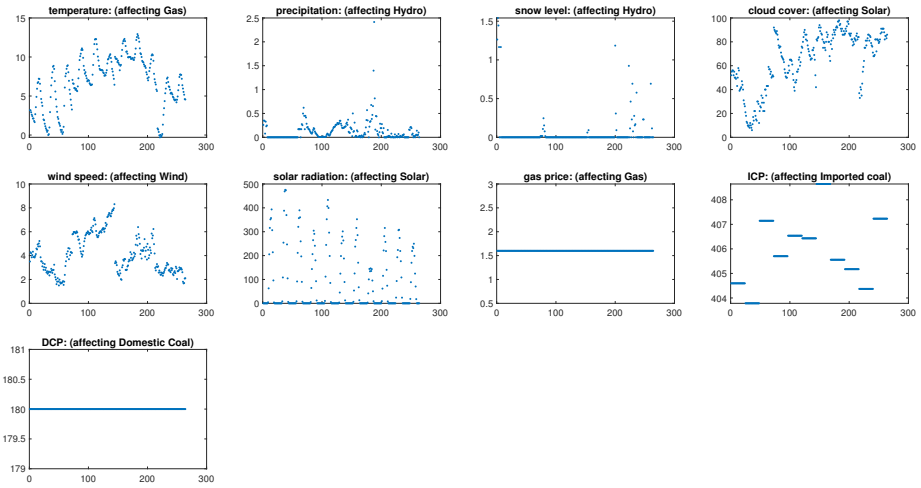

Figure 4.2 Explanatory variables for the transition days consist of 11 days or 264 hours

In the following section, we will state the results for each of those day categories. These predictions are going to the classified as two different kernels we chose for updating $t-1$ components $(x_{1:t-1})$. The first kernel for updating $x_{1:t-1}$ is an MH kernel where we choose one particle from the particle cloud and accept it with probability (4.6). Another kernel for updating the previous particles is to choose particles from the particle cloud proportional to the transition probability

$$X^{(j)}_{t-1,1:t-1} \sim f_t(X^{(j)}_{t-1,t-1}, x_t)$$

4.5 Results

4.5.1 Results based on MH kernel for updating $t-1$ components

Before running the expectation-maximization algorithm, we run a simple MCMC algorithm to have a good initial value used in our EM algorithm. The following section will illustrate our results and explain how we used the data to get these results.

4.5.1.1 Results for transition days

We used data of 2 days in the transition days category to make an initialization using a simple MCMC algorithm. After initialization, we used data of 10 days for the main algorithm that includes the EM step. In this stage, we run the EM algorithm 100 times with fewer particles (1000) to have a convergence in terms of our variances, and after this step, we run the algorithm with 1 EM iteration and more particles (10000). In 4.3, 4.4 ,4.5 and 4.6 we show the explanatory variables and predictions for the transition days.

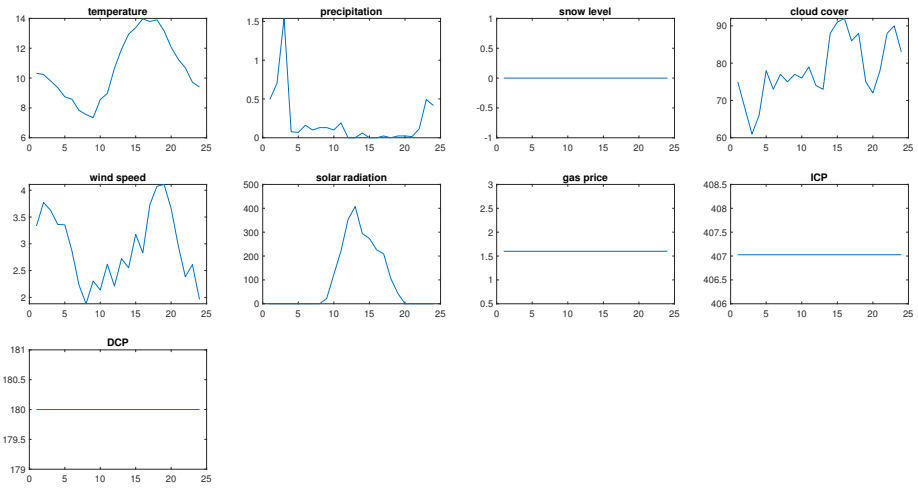

Figure 4.3 Explanatory variables of the last Friday

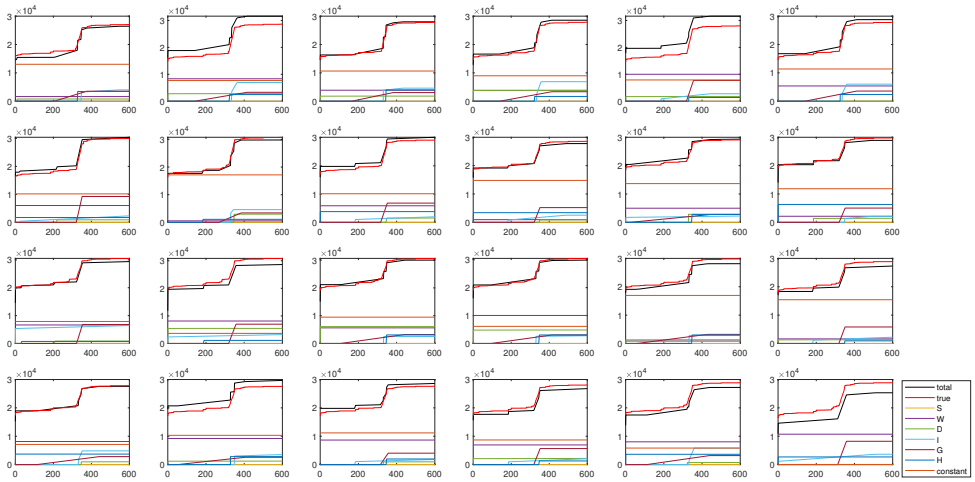

Figure 4.4 hour-by-hour prediction of the last Friday

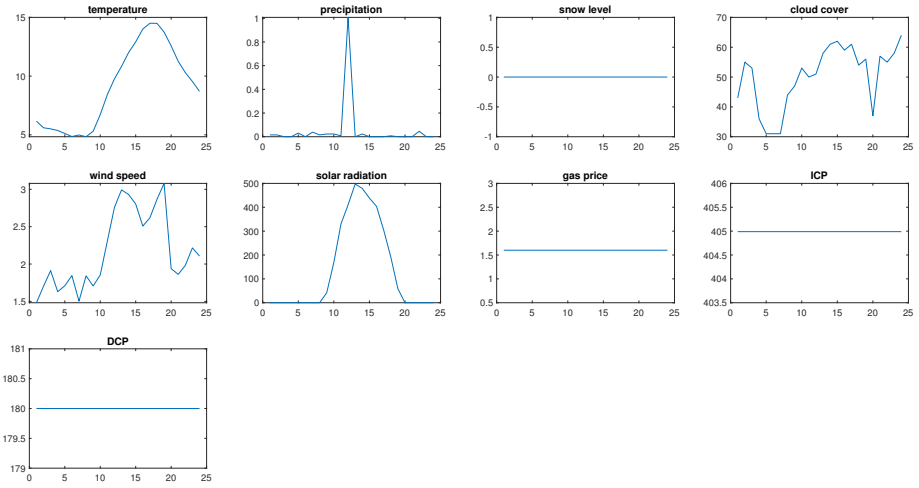

Figure 4.5 Explanatory variables of the last Monday

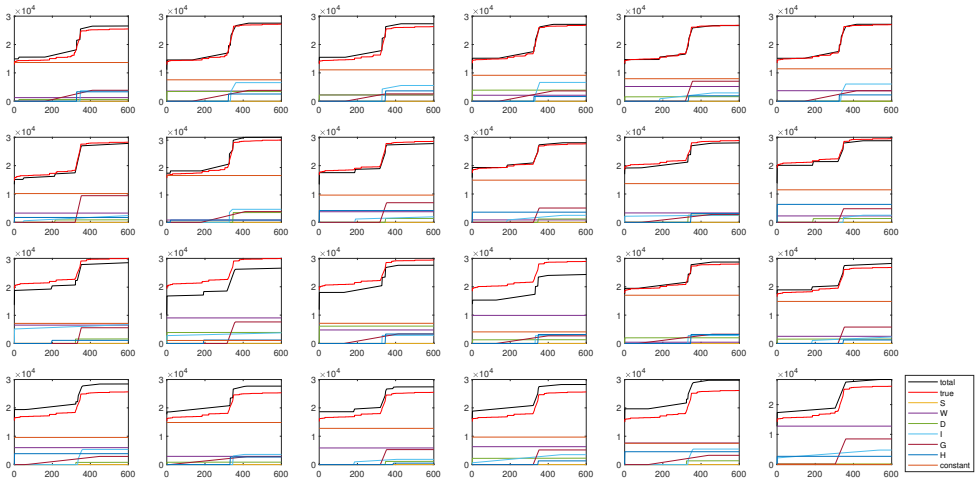

Figure 4.6 hour-by-hour prediction of the last Monday

4.5.1.2 Results for weekdays

We used three days in the weekdays category to make an initialization using a simple MCMC algorithm. After initialization, we used data of 15 days for the main algorithm that includes the EM step. In this stage, we run the EM algorithm 100 times with fewer particles to have a convergence in terms of our variances, and after this step, we run the algorithm with 1 EM iteration and more particles.

In the figures 4.7,4.8,4.9 and 4.10, we illustrate explanatory variables and predictions for three days of weekdays using the observed latent variable. As we discussed before, the prediction for each hour of a given day is based on the data of the same hour from the previous day. For example, to predict the supply curve at 10 pm on Tuesday, we are using the data and variables of Thursday. (Note that we use the data of Tuesday, Wednesday, and Thursday in the weekdays category).

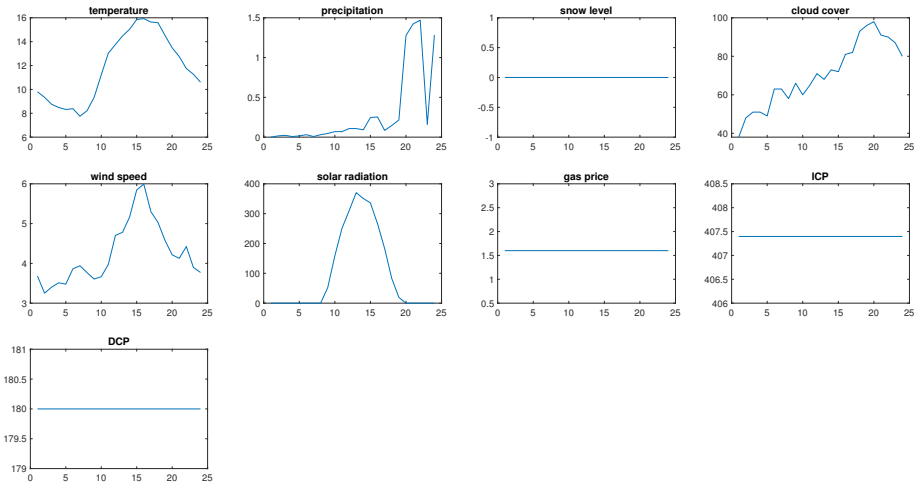

Figure 4.7 Explanatory variables of the last Tuesday

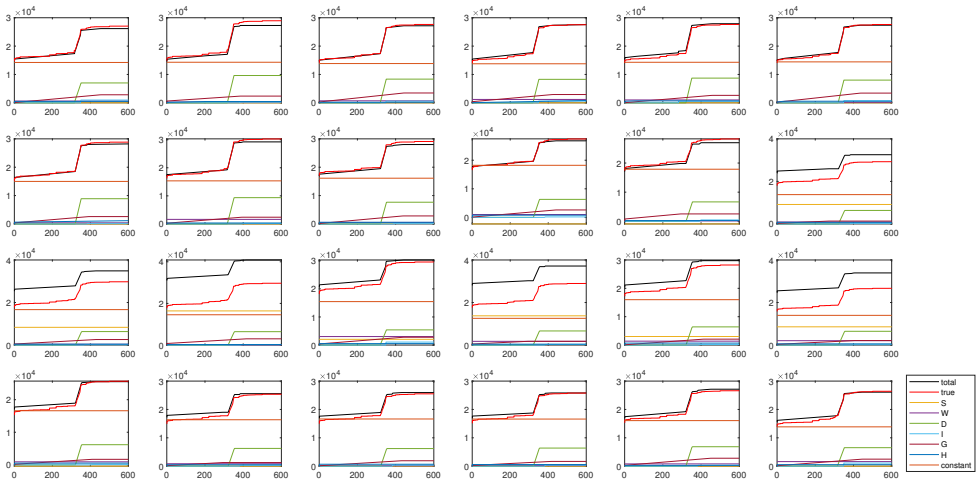

Figure 4.8 hour-by-hour prediction of the last Tuesday

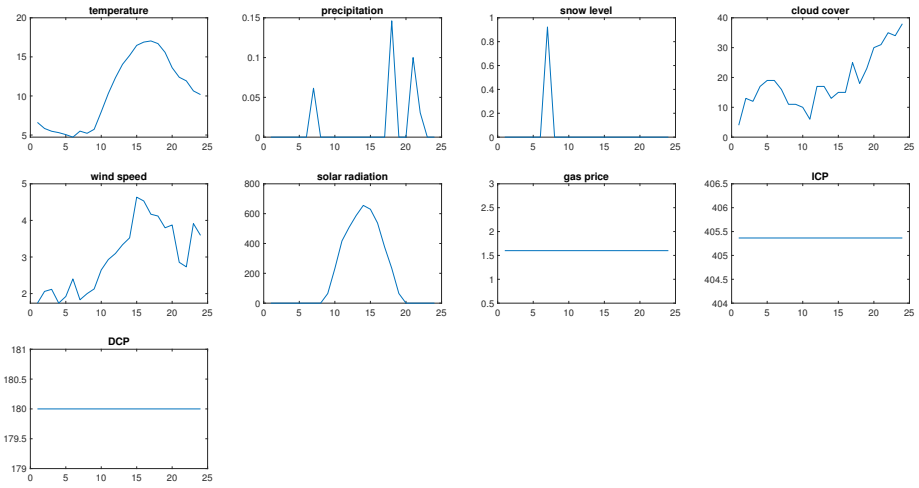

Figure 4.9 Explanatory variables of the last Wednesday

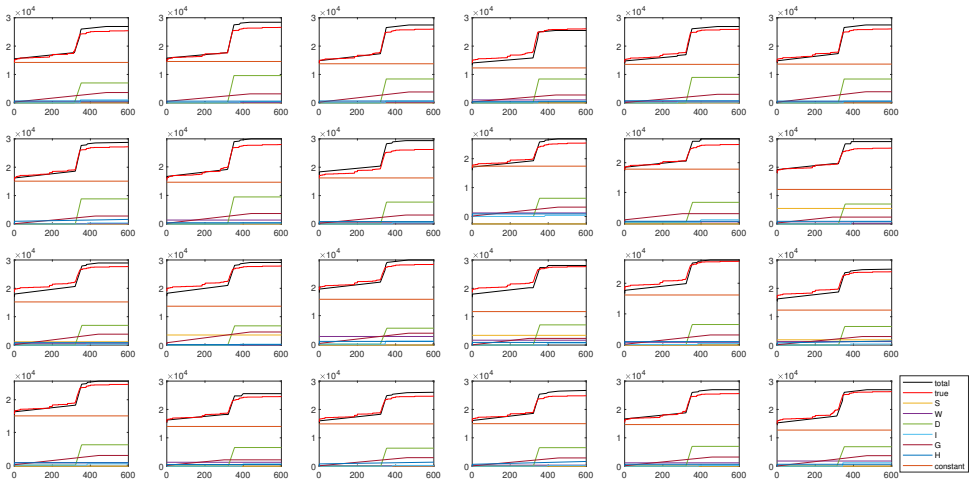

Figure 4.10 hour-by-hour prediction of the last Wednesday

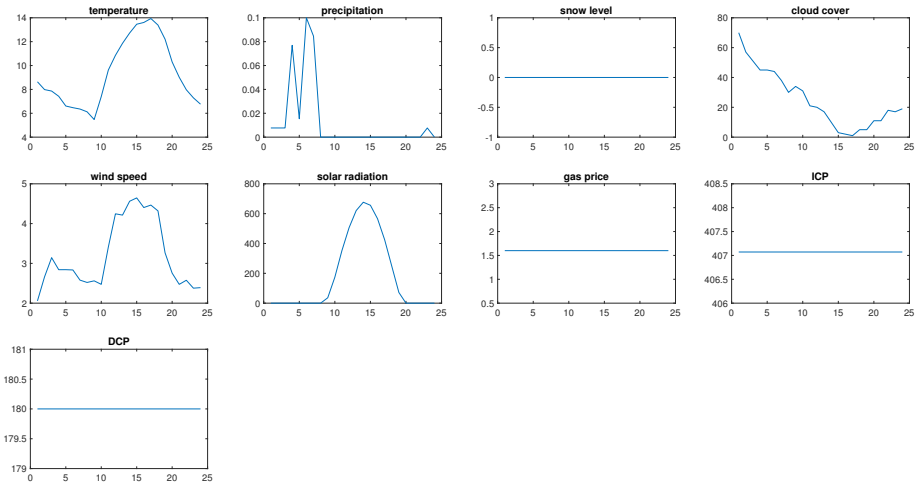

Figure 4.11 Explanatory variables of the last Thursday

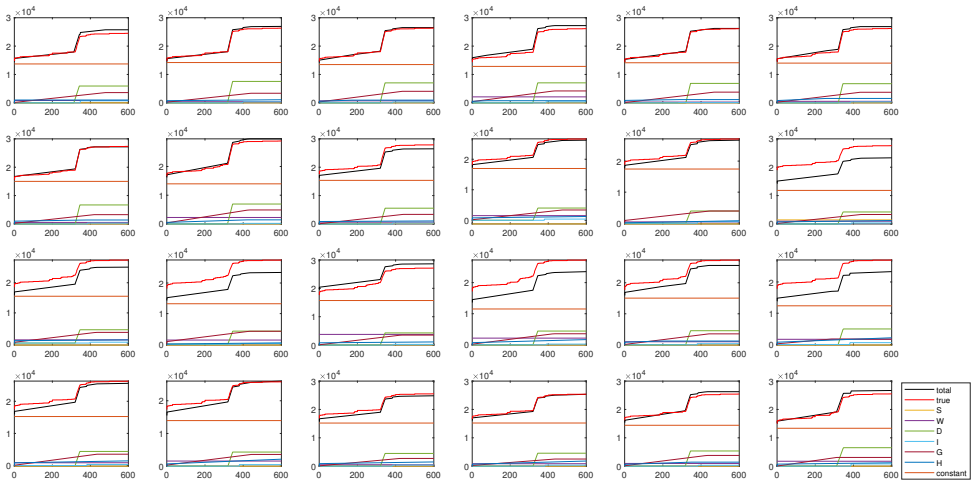

Figure 4.12 hour-by-hour prediction of the last Thursday

4.5.1.3 Results for weekends and holidays

As we mentioned before, we will predict the supply curve for weekends and holidays in this category. According to the calendar, we did not have a public holiday in that part of the year. We used two days to initialize the MCMC algorithm and ten days to run the EM algorithm with 100 iterations. In figures 4.13, 4.14, 4.15 and 4.16 we show the explanatory variables and predictions for the weekends and holidays.

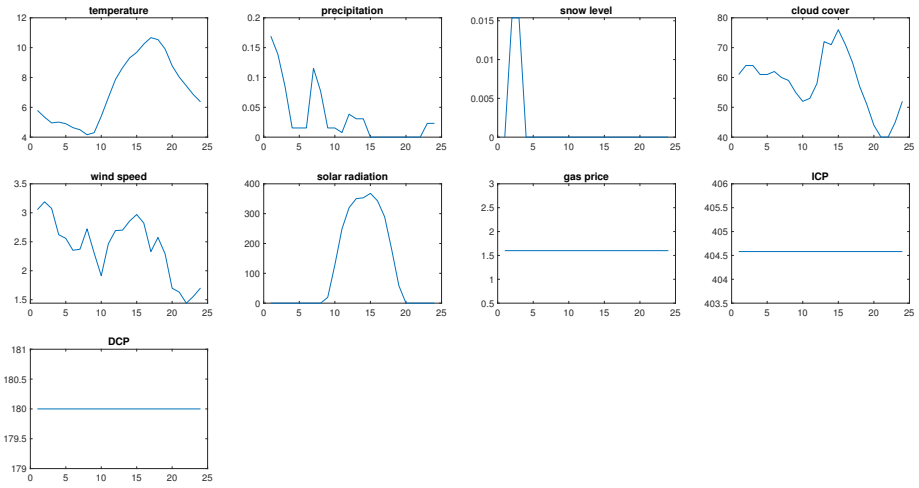

Figure 4.13 Explanatory variables of the last Sunday

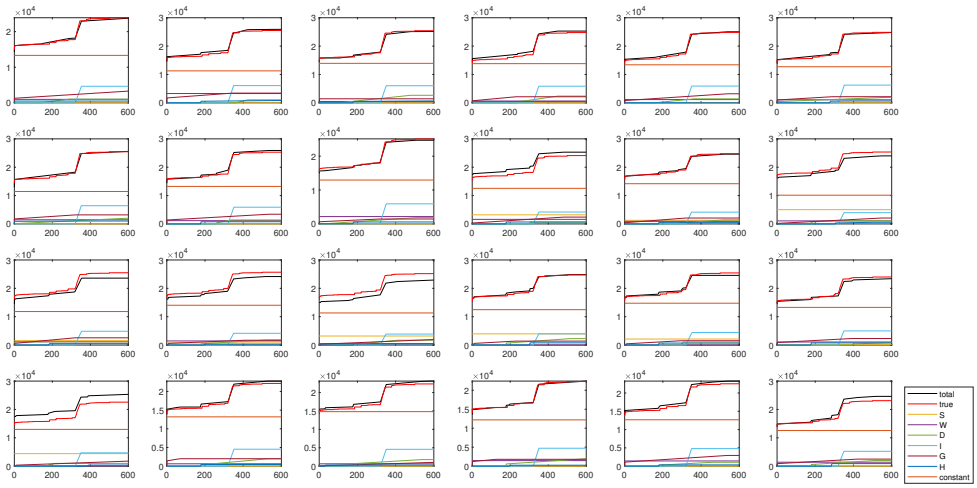

Figure 4.14 hour-by-hour prediction of the last Sunday

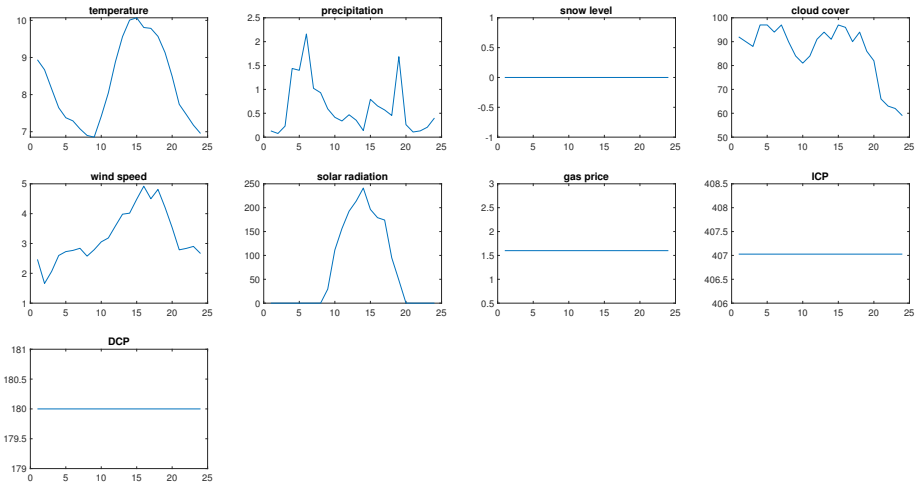

Figure 4.15 Explanatory variables of the last Saturday

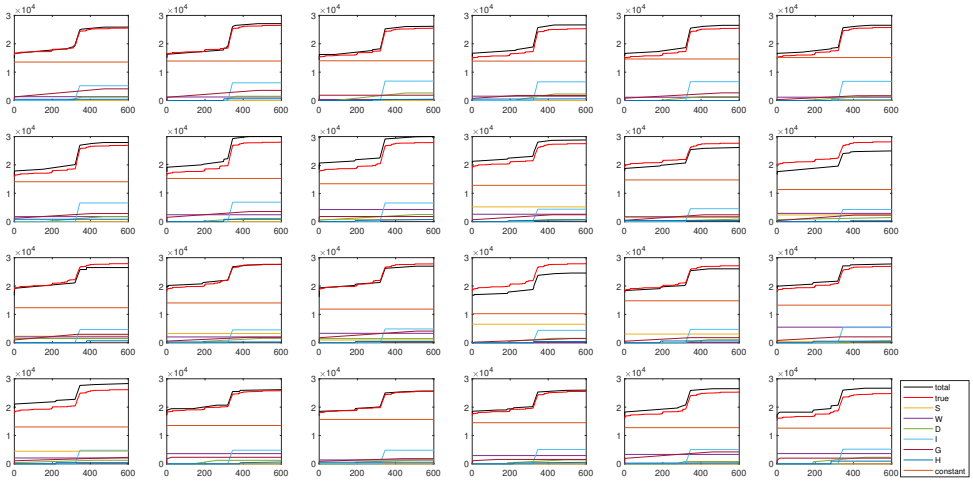

Figure 4.16 hour-by-hour prediction of the last Saturday

4.5.2 Results Based on Gibbs kernel for updating $t-1$ components

In the figures 4.17, 4.18, 4.19 and 4.20, we demonstrate the explanatory variables and predictions of last Friday and Monday of the Data using Gibbs move.

4.5.2.1 Results for the transition days

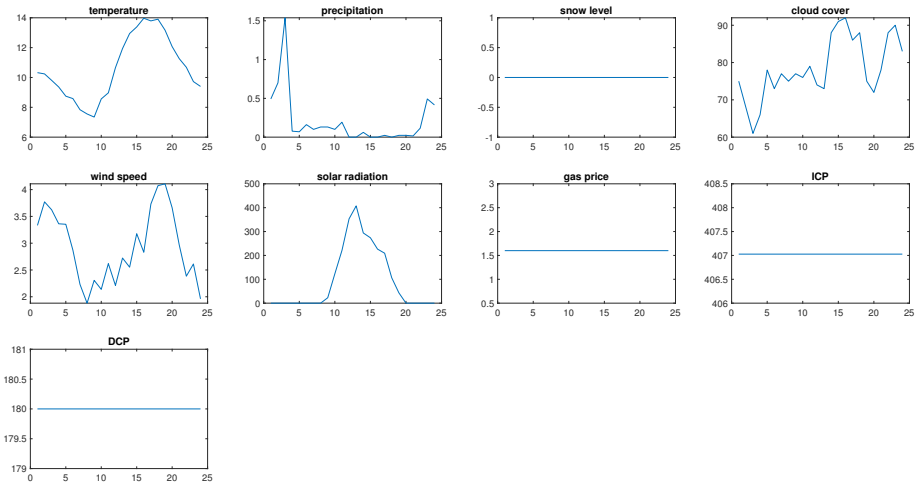

Figure 4.17 Explanatory variables of the last Friday

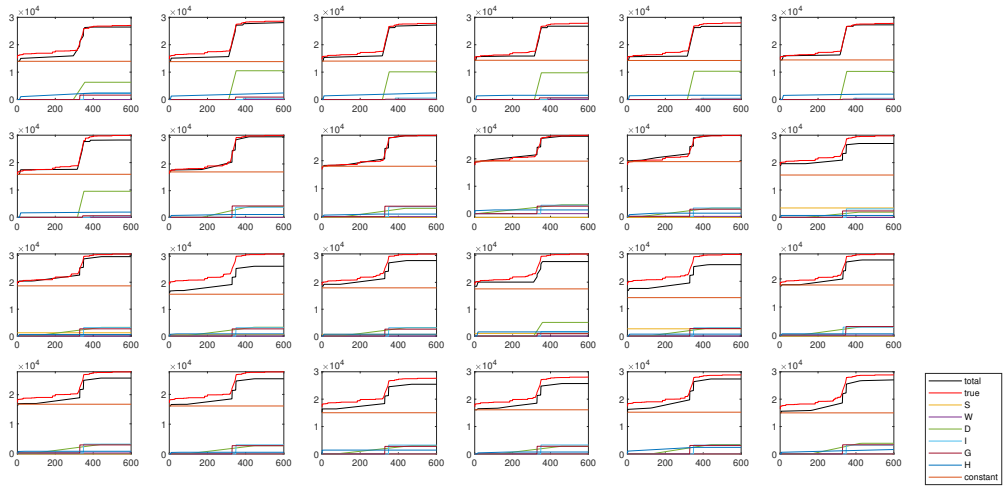

Figure 4.18 hour-by-hour prediction of the last Friday using Gibbs move

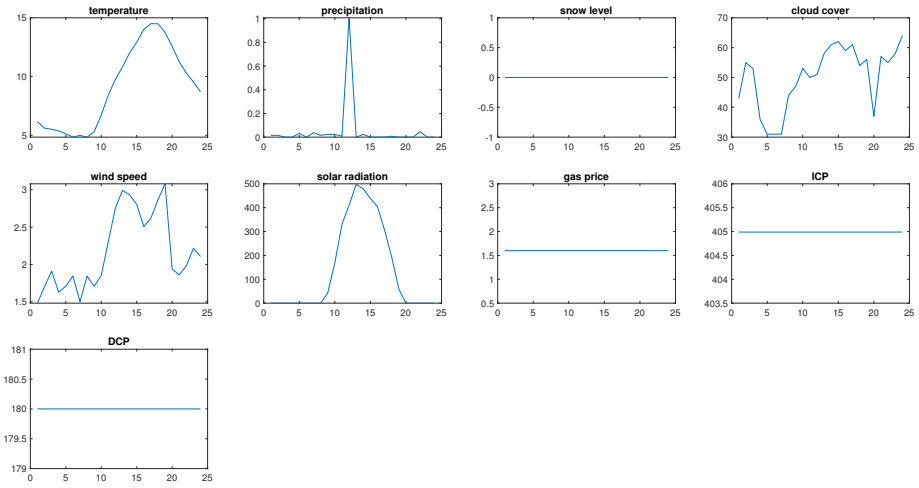

Figure 4.19 Explanatory variables of the last Monday

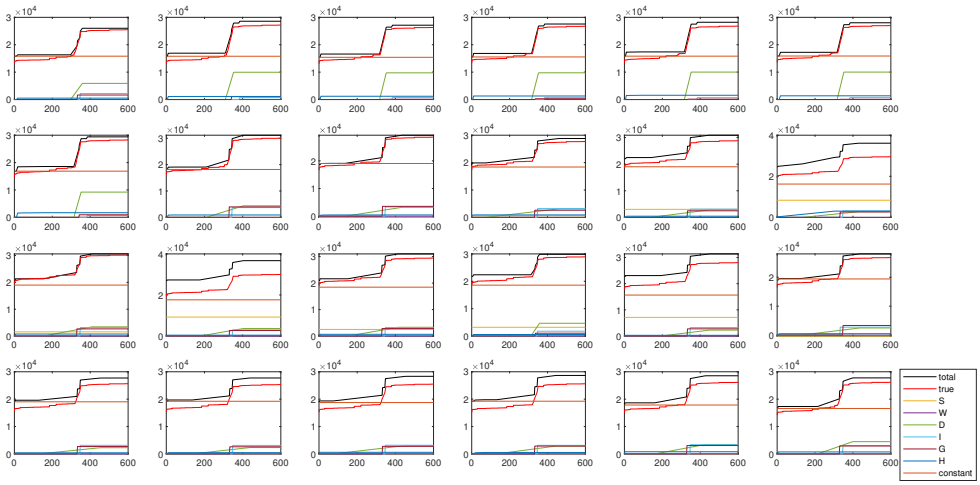

Figure 4.20 hour-by-hour prediction of the last Monday using Gibbs move

4.5.2.2 Results for weekdays

Figures 4.21, 4.22, 4.23, 4.24, 4.25 and 4.26 show the explanatory variables and predictions for the weekdays using Gibbs move.

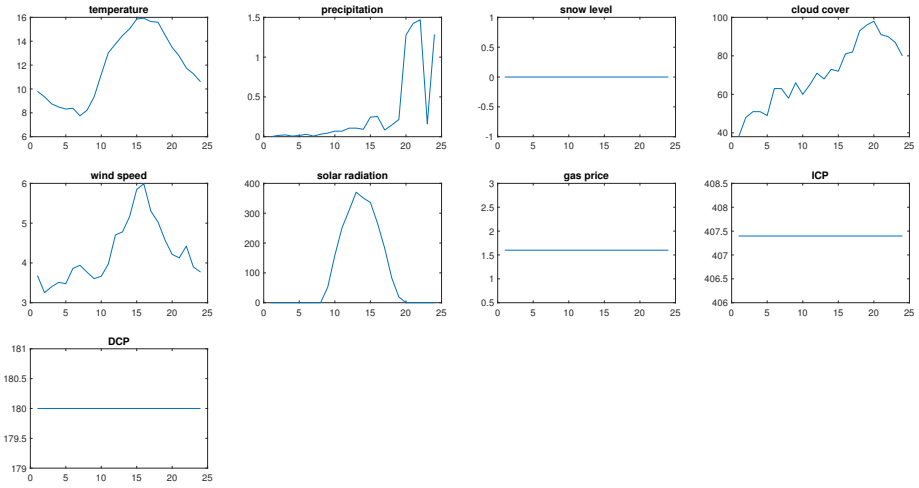

Figure 4.21 Explanatory variables of the last Tuesday

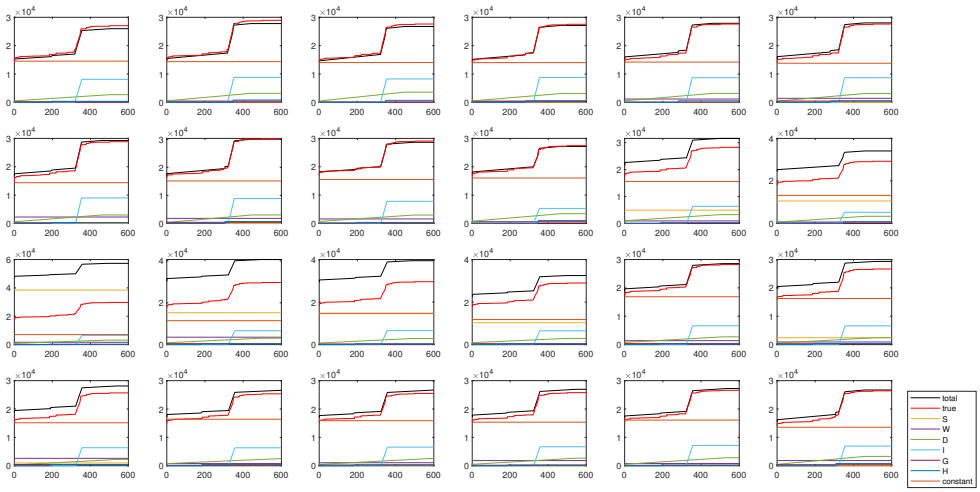

Figure 4.22 hour-by-hour prediction of the last Tuesday using Gibbs move

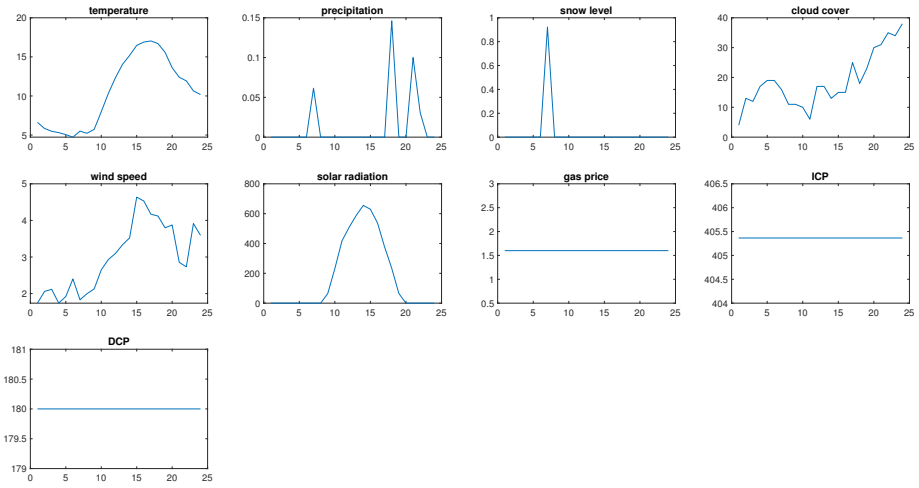

Figure 4.23 Explanatory variables of the last Wednesday

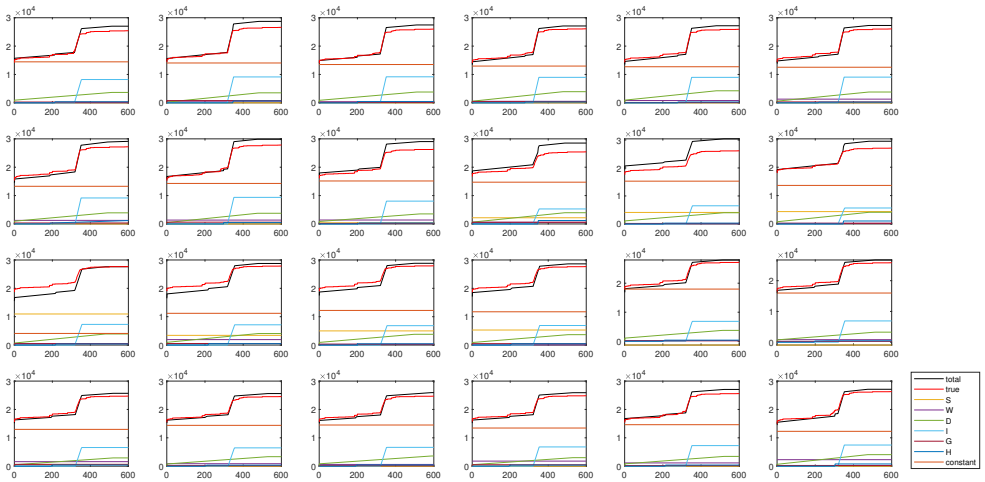

Figure 4.24 hour-by-hour prediction of the last Wednesday using Gibbs move

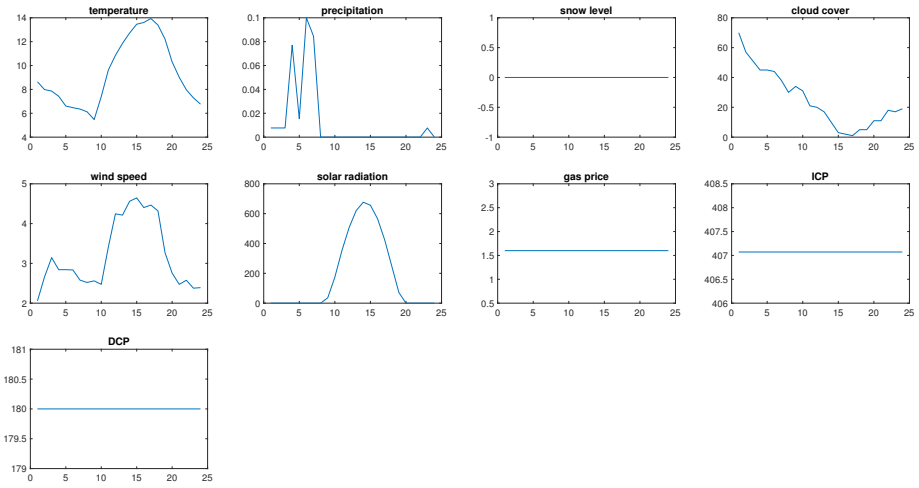

Figure 4.25 Explanatory variables of the last Thursday

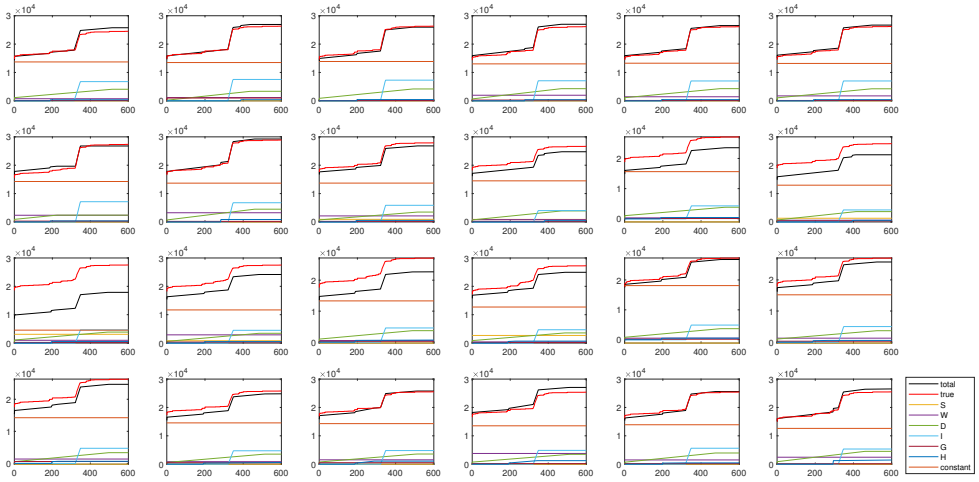

Figure 4.26 hour-by-hour prediction of the last Thursday using Gibbs move

4.5.2.3 Results for weekend and holidays

Figures 4.27, 4.28, 4.29 and 4.30, show the explanatory variables and predictions for the weekends and holidays using Gibbs move.

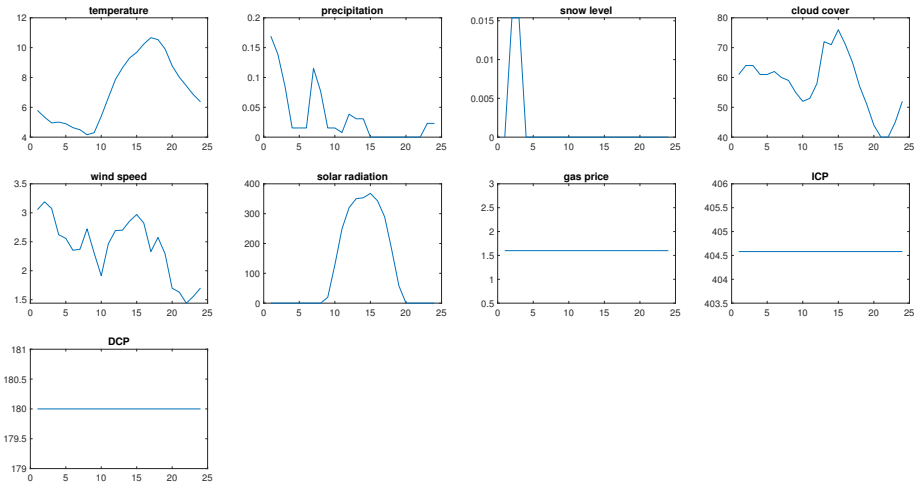

Figure 4.27 Explanatory variables of the last Sunday

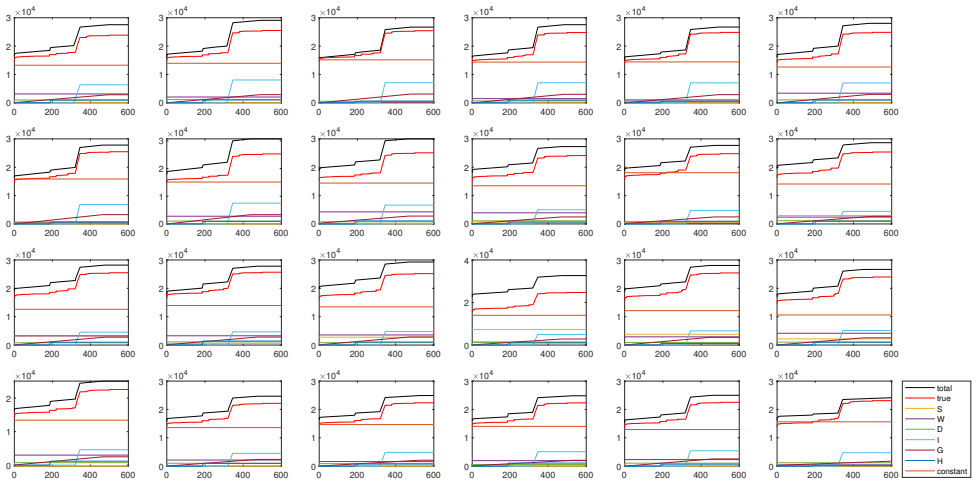

Figure 4.28 hour-by-hour prediction of the last Sunday using Gibbs move

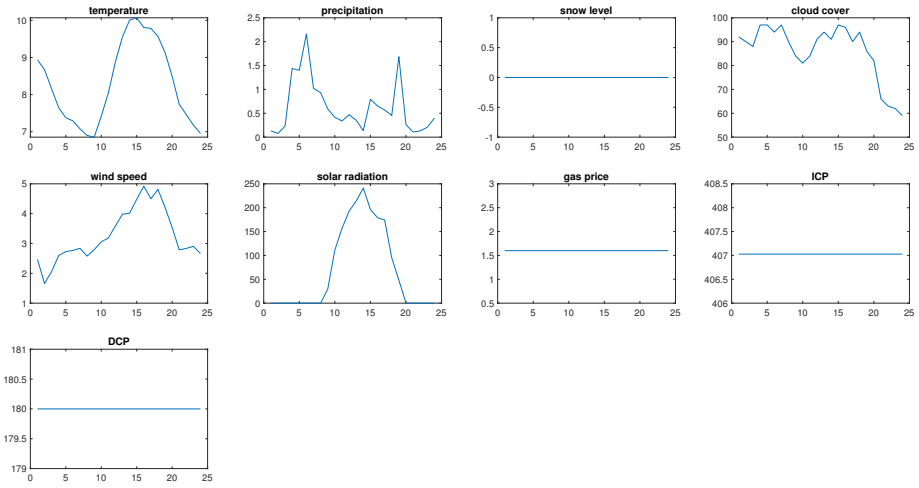

Figure 4.29 Explanatory variables of the last Saturday

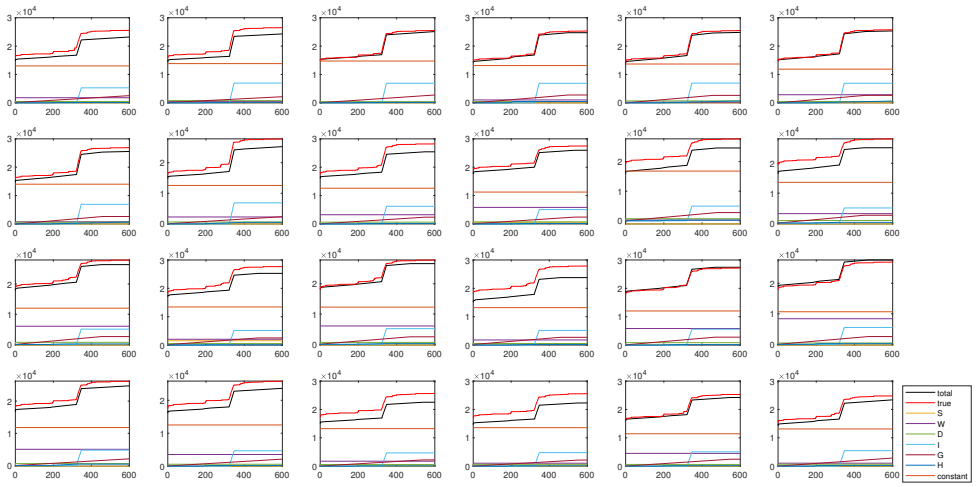

Figure 4.30 hour-by-hour prediction of the last Saturday using Gibbs move

As one can observe from the figures 4.8,4.22,4.26 the predictions are not good at some hours, such as 1 PM and 2 PM. We believe it is the case since some variances of the latent variables have not converged yet. Moreover, since these flawed estimations happen where the solar radiation increases, we suspect the model cannot predict the supply provided by solar energy. Another reason that we assume solar power is not predicted accurately because we can see an upward shift in the predictions (Note that solar source provides a constant supply value). Furthermore, we believe another reason that may cause some poor predictions for some hours is the high variances of some latent variables. To improve the predictions, we can correct the model, run the algorithm for more iterations (for example, 100 EM iterations) and provide more

data from the supply curves on the ground that we only use the data of 40 days of the year.

In the last chapter we will make a conclusion and discuss the results. Moreover we have provided some accuracy measures for each hours of the prediction in the appendix.

5. CONCLUSION AND DISCUSSION

5.1 Conclusion

Electricity price prediction has increasingly become one of the fundamental aspects for the companies to set their strategies. Despite some differences in electricity pricing, the nature of load forecasting stems from the relationship between supply and demand curves. These two curves are not necessarily dependent on each other, and multiple factors can affect each curve. Some factors that can influence the demand and supply of electricity can be temperature, prices of other energy resources, life quality of the population, solar radiation, wind speed, level of technology in the energy and electricity industry, and special days. The way these factors cause the demand and supply is not the same. For example, on holidays, electricity generation slows down, but the energy demand may increase.

In this study, we proposed a sequential Markov chain Monte Carlo algorithm to predict the hour-by-hour supply curve of the electricity market in Turkey. Moreover, we used many factors such as temperature, precipitation, and other factors to capture the supply curve. We discussed that due to the high dimensionality of the model, SMC approaches are inefficient for the prediction, and the SMCMC algorithm can be a remedy for this problem. In our thesis, we used two different methods in the SMCMC context as the following. In one approach, we used a kernel that uses MH updates, and in another one, we used a combination of Gibbs and MH updates to do SMCMC. The novel approach used in this thesis is not studied in the literature where we use particle filtering for a high-dimensional model in the electricity market. However, we need to use more powerful computational tools for the convergence of some latent variables. Since some of the latent variables had high variance, we had problems in predicting. However, One can observe that the algorithm has a good performance in most of the hours.

5.2 Future studies

Since electricity price prediction is the ultimate goal of the study, one should model the demand curve as well. In the future, we can use a similar approach to predict the demand curve too and predict the price. Moreover, to improve the performance, we can use other algorithms in the literature, such as SMC^2. As one can observe, we have poor performance in some hours, which stems from the model. For example, we should construct a better model for some resources like solar energy. Furthermore, some other resources provide electricity that can be mention in the study.

BIBLIOGRAPHY

Aggarwal, S. K., Saini, L. M., & Kumar, A. (2009). Electricity price forecasting in deregulated markets: A review and evaluation. *International Journal of Electrical Power & Energy Systems*, *31*(1), 13–22.

Barzola-Monteses, J., Mite-León, M., Espinoza-Andaluz, M., Gómez-Romero, J., & Fajardo, W. (2019). Time series analysis for predicting hydroelectric power production: The ecuador case. *Sustainability*, *11*(23), 6539.

Benth, F. E., Benth, J. S., & Koekebakker, S. (2008). *Stochastic modelling of electricity and related markets*, volume 11. World Scientific.

Beskos, A., Crisan, D., Jasra, A., Kamatani, K., & Zhou, Y. (2014). A stable particle filter in high-dimensions. *arXiv preprint arXiv:1412.3501*.

Burkhart, M. C., Brandman, D. M., Vargas-Irwin, C. E., & Harrison, M. T. (2016). The discriminative kalman filter for nonlinear and non-gaussian sequential bayesian filtering. *arXiv preprint arXiv:1608.06622*.

Buzoianu, M., Brockwell, A., & Seppi, D. (2005). A dynamic supply-demand model for electricity prices.

Costa, A., Crespo, A., Navarro, J., Lizcano, G., Madsen, H., & Feitosa, E. (2008). A review on the young history of the wind power short-term prediction. *Renewable and Sustainable Energy Reviews*, *12*(6), 1725–1744.

Del Moral, P., Doucet, A., & Jasra, A. (2006). Sequential monte carlo samplers. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *68*(3), 411–436.

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, *39*(1), 1–22.

Douc, R., Garivier, A., Moulines, E., Olsson, J., et al. (2011). Sequential monte carlo smoothing for general state space hidden markov models. *Annals of Applied Probability*, *21*(6), 2109–2145.

Doucet, A., Briers, M., & Sénécal, S. (2006). Efficient block sampling strategies for sequential monte carlo methods. *Journal of Computational and Graphical Statistics*, *15*(3), 693–711.

Eydeland, A. & Wolyniec, K. (2002). *Energy and power risk management: New developments in modeling, pricing, and hedging*, volume 97. John Wiley & Sons.

Fahrmeir, L. & Tutz, G. (2001). State space and hidden markov models. *Multivariate Statistical Modelling Based on Generalized Linear Models*, 331–383.

Fan, S., Mao, C., & Chen, L. (2007). Next-day electricity-price forecasting using a hybrid network. *IET generation, transmission & distribution*, *1*(1), 176–182.

Fearnhead, P., Taylor, B. M., et al. (2013). An adaptive sequential monte carlo sampler. *Bayesian analysis*, *8*(2), 411–438.

Fearnhead, P., Wyncoll, D., & Tawn, J. (2010). A sequential smoothing algorithm with linear computational cost. *Biometrika*, *97*(2), 447–464.

Filik, T., Filik, Ü. B., & Gerek, Ö. N. (2017). Solar radiation–to–power generation models for one-axis tracking pv system with on-site measurements from eskisehir, turkey. In *E3S Web of Conferences*, volume 22, (pp. 00046). EDP Sciences.

Filik, Ü. B. & Filik, T. (2017). Wind speed prediction using artificial neural networks based on multiple local measurements in eskisehir. *Energy Procedia*, *107*, 264–269.

Gneiting, T. (2011). Making and evaluating point forecasts. *Journal of the American Statistical Association*, *106*(494), 746–762.

Gonzalez, V., Contreras, J., & Bunn, D. W. (2011). Forecasting power prices using a hybrid fundamental-econometric model. *IEEE Transactions on Power Systems*, *27*(1), 363–372.

Gordon, N. J., Salmond, D. J., & Smith, A. F. (1993). Novel approach to nonlinear/non-gaussian bayesian state estimation. In *IEE proceedings F (radar and signal processing)*, volume 140, (pp. 107–113). IET.

Griffin, J. E. & Walker, S. G. (2013). On adaptive metropolis–hastings methods. *Statistics and Computing*, *23*(1), 123–134.

Guerci, E., Rastegar, M. A., & Cincotti, S. (2010). Agent-based modeling and simulation of competitive wholesale electricity markets. In *Handbook of power systems II* (pp. 241–286). Springer.

Haario, H., Saksman, E., Tamminen, J., et al. (2001). An adaptive metropolis algorithm. *Bernoulli*, *7*(2), 223–242.

Hammer, A., Heinemann, D., Lorenz, E., & Lückehe, B. (1999). Short-term forecasting of solar radiation: a statistical approach using satellite data. *Solar Energy*, *67*(1-3), 139–150.

Hastings, W. K. (1970). Monte carlo sampling methods using markov chains and their applications.

He, Y., Yan, Y., & Xu, Q. (2019). Wind and solar power probability density prediction via fuzzy information granulation and support vector quantile regression. *International Journal of Electrical Power & Energy Systems*, *113*, 515–527.

Hong, T. (2010). Short term electric load forecasting.

Hong, T. (2014). Energy forecasting: Past, present, and future. *Foresight: The International Journal of Applied Forecasting*, (32), 43–48.

Hong, T., Pinson, P., Wang, Y., Weron, R., Yang, D., & Zareipour, H. (2020). Energy forecasting: A review and outlook. *IEEE Open Access Journal of Power and Energy*.

Hong, T., Wilson, J., & Xie, J. (2013). Long term probabilistic load forecasting and normalization with hourly information. *IEEE Transactions on Smart Grid*, *5*(1), 456–462.

Inman, R. H., Pedro, H. T., & Coimbra, C. F. (2013). Solar forecasting methods for renewable energy integration. *Progress in energy and combustion science*, *39*(6), 535–576.

Jang, H. S., Bae, K. Y., Park, H.-S., & Sung, D. K. (2016). Solar power prediction based on satellite images and support vector machine. *IEEE Transactions on Sustainable Energy*, *7*(3), 1255–1263.

Johnsen, T. A. (2001). Demand, generation and price in the norwegian market for electric power. *Energy Economics*, *23*(3), 227–251.

Julier, S. J. & Uhlmann, J. K. (1997). New extension of the kalman filter to nonlinear systems. In *Signal processing, sensor fusion, and target recognition VI*, volume 3068, (pp. 182–193). International Society for Optics and Photonics.

Karakatsani, N. V. & Bunn, D. W. (2008). Forecasting electricity prices: The impact of fundamentals and time-varying coefficients. *International Journal of Forecasting*, *24*(4), 764–785.

Kavaklioglu, K. (2011). Modeling and prediction of turkey's electricity consumption using support vector regression. *Applied Energy*, *88*(1), 368–375.

Khiatani, D. & Ghose, U. (2017). Weather forecasting using hidden markov model. In *2017 International Conference on Computing and Communication Technologies for Smart Nation (IC3TSN)*, (pp. 220–225). IEEE.

Kopp, R. E. & Orford, R. J. (1963). Linear regression applied to system identification for adaptive control systems. *Aiaa Journal*, *1*(10), 2300–2306.

Kowalska-Pyzalska, A., Maciejowska, K., Suszczyński, K., Sznajd-Weron, K., & Weron, R. (2014). Turning green: Agent-based modeling of the adoption of dynamic electricity tariffs. *Energy Policy*, *72*, 164–174.

Kristiansen, T. (2012). Forecasting nord pool day-ahead prices with an autoregressive model. *Energy Policy*, *49*, 328–332.

Krogh, A., Brown, M., Mian, I. S., Sjölander, K., & Haussler, D. (1994). Hidden markov models in computational biology: Applications to protein modeling. *Journal of molecular biology*, *235*(5), 1501–1531.

Li, G., Liu, C.-C., Lawarree, J., Gallanti, M., & Venturini, A. (2005). State-of-the-art of electricity price forecasting. In *International Symposium CIGRE/IEEE PES, 2005.*, (pp. 110–119). IEEE.

Lin, M. T., Zhang, J. L., Cheng, Q., & Chen, R. (2005). Independent particle filters. *Journal of the American Statistical Association*, *100*(472), 1412–1421.

Liu, J. S. & Chen, R. (1998). Sequential monte carlo methods for dynamic systems. *Journal of the American statistical association*, *93*(443), 1032–1044.

Lorenz, E., Hammer, A., Heinemann, D., et al. (2004). Short term forecasting of solar radiation based on satellite data. In *EUROSUN2004 (ISES Europe Solar Congress)*, (pp. 841–848). Freiburg Germany.

Metaxiotis, K., Kagiannas, A., Askounis, D., & Psarras, J. (2003). Artificial intelligence in short term electric load forecasting: a state-of-the-art survey for the researcher. *Energy conversion and Management*, *44*(9), 1525–1534.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., & Teller, E. (1953). Equation of state calculations by fast computing machines. *The journal of chemical physics*, *21*(6), 1087–1092.

Monteiro, C., Ramirez-Rosado, I. J., & Fernandez-Jimenez, L. A. (2013). Short-term forecasting model for electric power production of small-hydro power plants. *Renewable Energy*, *50*, 387–394.

Mor, B., Garhwal, S., & Kumar, A. (2021). A systematic review of hidden markov models and their applications. *Archives of computational methods in engineering*, *28*, 1429–1448.

Nefian, A. V. & Hayes, M. H. (1998). Hidden markov models for face recognition. In *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP'98 (Cat. No. 98CH36181)*, volume 5, (pp. 2721–2724). IEEE.

Pan, D.-f., Liu, H., Li, Y.-f., et al. (2008). A wind speed forecasting optimization model for wind farms based on time series analysis and kalman filter algorithm. *Power System Technology*, *32*(7), 82–86.

Picone, J. (1990). Continuous speech recognition using hidden markov models. *IEEE Assp magazine*, *7*(3), 26–41.

Prema, V. & Rao, K. U. (2015). Development of statistical time series models for solar power prediction. *Renewable energy*, *83*, 100–109.

Sensfuß, F., Ragwitz, M., Genoese, M., & Möst, D. (2007). Agent-based simulation of electricity markets. a literature review.

Septier, F., Pang, S. K., Carmi, A., & Godsill, S. (2009). On mcmc-based particle methods for bayesian filtering: Application to multitarget tracking. In *2009 3rd IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, (pp. 360–363). IEEE.

Septier, F. & Peters, G. W. (2015). Langevin and hamiltonian based sequential mcmc for efficient bayesian filtering in high-dimensional spaces. *IEEE Journal of selected topics in signal processing*, *10*(2), 312–327.

Soman, S. S., Zareipour, H., Malik, O., & Mandal, P. (2010). A review of wind power and wind speed forecasting methods with different time horizons. In *North American Power Symposium 2010*, (pp. 1–8). IEEE.

Vehviläinen, I. & Pyykkönen, T. (2005). Stochastic factor model for electricity spot price—the case of the nordic market. *Energy Economics*, *27*(2), 351–367.

Wan, E. A. & Van Der Merwe, R. (2000). The unscented kalman filter for nonlinear estimation. In *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No. 00EX373)*, (pp. 153–158). Ieee.

Weron, R. (2014). Electricity price forecasting: A review of the state-of-the-art with a look into the future. *International journal of forecasting*, *30*(4), 1030–1081.

Yan, X. & Chowdhury, N. A. (2015). Midterm electricity market clearing price forecasting using two-stage multiple support vector machine. *Journal of Energy*, *2015*.

Yuksek, O., Komurcu, M. I., Yuksel, I., & Kaygusuz, K. (2006). The role of hydropower in meeting turkey's electric energy demand. *Energy Policy*, *34*(17), 3093–3103.

Zhang, Y. (2004). *Prediction of financial time series with Hidden Markov Models*. PhD thesis, Applied Sciences: School of Computing Science.

APPENDIX A

Performance measures of Model prediction Using MH kernel for updating $t-1$ components

Table A.1 Adjusted R^2 for each hour of the last week of data

	Monday	**Tuesday**	**Wednesday**	**Thursday**	**Friday**	**Saturday**	**Sunday**
0	0.7695	0.9866	0.9465	0.9280	0.7830	0.9881	0.9833
1	0.8699	0.9664	0.9404	0.9883	0.8316	0.9768	0.9714
2	0.8977	0.9979	0.9571	0.9937	0.8921	0.9664	0.9874
3	0.9460	0.9880	0.8848	0.9284	0.9238	0.8494	0.9512
4	0.9376	0.9700	0.9502	0.9966	0.8926	0.8942	0.9825
5	0.9640	0.9940	0.9455	0.9679	0.9232	0.9317	0.9842
6	0.9609	0.9962	0.9481	0.9945	0.9384	0.8908	0.9813
7	0.9844	0.9910	0.9284	0.9775	0.9250	0.7305	0.9738
8	0.9974	0.9780	0.6856	0.8484	0.9349	0.5883	0.9851
9	0.9887	0.9929	0.8648	0.9015	0.8758	0.6898	0.7117
10	-1.6548	0.9549	0.8373	0.8904	-23.6482	0.7601	0.9875
11	0.9803	-0.7001	0.7881	-1.8988	0.6088	-0.0479	0.8331
12	0.6270	-1.7187	0.7538	0.1260	-7.0377	0.9208	0.8119
13	0.5891	-9.0504	0.7523	-1.1184	-0.3892	0.9631	0.8136
14	-0.4368	0.8172	0.8688	0.7196	-8.8157	0.9696	0.4792
15	-2.3309	-5.5502	0.7279	-0.6206	-13.4933	0.2402	0.9760
16	0.9751	0.5661	0.8488	0.5190	0.9785	0.9532	0.9717
17	0.6023	-4.1092	0.8668	-1.0959	-0.3163	0.8603	0.9545
18	0.7319	0.9276	0.9325	0.8088	0.3191	0.2443	0.1521
19	0.6466	0.8473	0.9395	0.8409	0.1521	0.9373	0.9710
20	0.6188	0.8994	0.9151	0.8727	0.1905	0.9857	0.9513
21	0.5708	0.9351	0.8921	0.9765	0.1657	0.9404	0.9927
22	0.5455	0.9398	0.9441	0.9432	0.5001	0.7778	0.9587
23	0.6236	0.9607	0.9284	0.9337	0.7954	0.6160	0.9027

Performance measures of Model prediction Using Gibbs kernel for updating $t-1$ components

Table A.2 Adjusted R^2 for each hour of the last week of data

	Monday	**Tuesday**	**Wednesday**	**Thursday**	**Friday**	**Saturday**	**Sunday**
0	0.7878	0.9631	0.9437	0.9434	0.8029	0.6201	0.0481
1	0.8333	0.9799	0.9226	0.9898	0.8707	0.7136	0.4533
2	0.9136	0.9830	0.9613	0.9795	0.9023	0.9810	0.9314
3	0.9405	0.9963	0.9625	0.9629	0.9556	0.9834	0.5788
4	0.8914	0.9741	0.9456	0.9886	0.9411	0.9743	0.8155
5	0.9425	0.9674	0.9477	0.9835	0.9722	0.9900	0.4506
6	0.9321	0.9635	0.9252	0.9487	0.9690	0.9010	0.6997
7	0.9439	0.9923	0.9263	0.9904	0.9863	0.7052	-0.6983
8	0.8718	0.9959	0.7883	0.9051	0.9865	0.6310	-0.8407
9	0.5153	0.9903	0.4992	0.3345	0.9795	0.7353	-0.2989
10	-4.9817	-0.1464	-0.1816	-1.1928	0.9530	-0.3595	-0.1851
11	-2.2285	-1.4854	0.7779	-1.2111	0.9049	-0.0933	-0.4018
12	-6.2286	-51.8967	0.2873	-11.8171	0.8294	0.8351	0.0879
13	-4.3192	-8.3169	0.6666	-0.3514	0.9271	0.5455	0.4879
14	-1.7077	-7.2073	0.7947	-1.3311	0.9057	0.9194	-0.9082
15	-5.5480	-0.3389	0.8305	0.5192	0.6736	-0.2799	-5.8008
16	-1.2343	0.9393	0.9020	0.9175	0.8659	0.9859	0.0900
17	-0.5774	0.2216	0.9171	0.6791	0.6445	0.9414	0.1696
18	0.3182	0.3294	0.9329	0.5191	0.6717	0.7062	0.2080
19	0.2094	0.7715	0.9395	0.6309	0.6874	0.3717	0.1705
20	0.1932	0.8672	0.9250	0.9634	0.6238	-0.1227	0.1529
21	0.1620	0.8521	0.9217	0.8029	0.7119	-0.0998	0.2865
22	0.6986	0.9465	0.9438	0.9652	0.6166	0.9261	0.3789
23	0.7984	0.9391	0.9501	0.9615	0.6518	0.7575	0.5566