SCHOOL OF INFORMATION SCIENCES & TECHNOLOGY

DEPARTMENT OF STATISTICS

POSTGRADUATE PROGRAM

An econometric analysis of high-frequency financial data

By

Fiori Lamprinakou

A THESIS

Submitted to the Department of Statistics

of the Athens University of Economics and Business

in partial fulfilment of the requirements for

the degree of

Doctor of Philosophy in Statistics

December 2021
Athens, Greece

ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS

ΣΧΟΛΗ ΕΠΙΣΤΗΜΩΝ & ΤΕΧΝΟΛΟΓΙΑΣ ΤΗΣ ΠΛΗΡΟΦΟΡΙΑΣ

ΤΜΗΜΑ ΣΤΑΤΙΣΤΙΚΗΣ

ΔΙΔΑΚΤΟΡΙΚΟ ΠΡΟΓΡΑΜΜΑ

Οικονομετρική ανάλυση οικονομικών δεδομένων υψηλής συχνότητας

Φιόρη Λαμπρινάκου

ΔΙΑΤΡΙΒΗ

Που υποβλήθηκε στο Τμήμα Στατιστικής

του Οικονομικού Πανεπιστημίου Αθηνών

ως μέρος των απαιτήσεων για την απόκτηση

Διπλώματος Διδακτορικών Σπουδών στη Στατιστική

Αθήνα
Δεκέμβριος 2021

An econometric analysis of high-frequency financial data

Approved by
Dissertation Committee:

_____

_____

_____

_____

_____

_____

_____

Contents

List of Tables

List of Figures

Acknowledgments

I would like to express my great appreciation to my supervisor Prof. Petros Dellaportas. I am deeply grateful to him for offering me this opportunity. Under his insightful supervision I overcame successfully many obstacles and learned a lot. His patience, continuous support and encouragement make it possible for me to finish up this thesis. I owe a considerable debt to an outstanding committee member to whom I am thankful, Prof. Omiros Papaspiliopoulos, for his guidance and suggestions.

I would like to thank the third committee member Assistant Prof. Nikos Demiris, as well as Prof. Ioannis Ntzoufras for their helpful comments. Many thanks to Assistant Prof. Panagiotis Papastamoulis, member of the examination committee, for carefully reading the present thesis and for giving me such constructive comments. A special acknowledgment for Nikos Moraitis and Tatiana Mixou of the laboratory staff who helped me solve PC hardware problems.

This research has been co-financed by the European Union (European Social Fund – ESF) and Greek national funds through the Operational Program 'Education and Lifelong Learning' of the National Strategic Reference Framework (NSRF) - Research Funding Program: ARISTEIA-LIKEJUMPS-436.

Last but not least, I would like to thank my mother for her unlimited support throughout my life. *Mom, you are always by my side, with your endless and unconditional love, reminding me to believe in myself and never give up. I really cannot thank you enough.*

To my mother

Abstract

Fiori Lamprinakou

An econometric analysis of high-frequency financial data

December, 2021

We present and compare observation driven and parameter driven models for predicting integer price changes of high-frequency financial data. We explore Bayesian inference via Markov chain Monte Carlo (MCMC) and sequential Monte Carlo (SMC) for the observation driven model activity-direction-size (ADS), introduced by Rydberg and Shephard [1998a, 2003]. We extend the ADS model by proposing a parameter driven model and use a Bernoulli generalized linear model (GLM) with a latent process in the mean. We propose a new decomposition model that uses trade intervals and is applied on data that allow three possible tick movements: one tick up price change, one tick down price change, or no price change. We model each component sequentially using a Binomial generalized linear autoregressive moving average (GLARMA) model, as well as a GLM with a latent process in the mean. We perform a simulation study to investigate the effectiveness of the proposed parameter driven models using different algorithms within a Bayesian framework. We illustrate the analysis by modelling the transaction-by-transaction data of of E-mini Standard and Poor's (S&P) 500 index futures contract traded on the Chicago Mercantile Exchange's Globex platform between May 16th 2011 and May 24th 2011. In order to assess the predictive performance, we compare the mean square error (MSE) and mean absolute error (MAE) criterion, as well as four scalar performance measures, namely, accuracy, sensitivity, precision and specificity derived from the confusion matrix.

ΠΕΡΙΛΗΨΗ

Φιόρη Λαμπρινάκου

Οικονομετρική ανάλυση οικονομικών δεδομένων υψηλής συχνότητας

Δεκέμβριος, 2021

Παρουσιάζουμε και συγκρίνουμε μοντέλα βασισμένα στην παρατήρηση (observation driven) και στις παραμέτρους (parameter driven) για να προβλέψουμε τις διακριτές αλλαγές των τιμών οικονομικών δεδομένων υψηλής συχνότητας. Η ανάλυση γίνεται με Μπεϋζιανή προσέγγιση με Μαρκοβιανές αλυσίδες Monte Carlo (MC) και ακολουθιακές μεθόδους MC για το observation driven μοντέλο activity-direction-size (ADS) [Rydberg and Shephard, 1998a, 2003]. Επεκτείνουμε το ADS μοντέλο ορίζοντας ένα γενικευμένο γραμμικό μοντέλο (GLM) των οποίων τα δεδομένα απόκρισης προέρχονται από την Bernoulli κατανομή και διέπονται από μία μη παρατηρήσιμη στοχαστική διαδικασία. Προτείνουμε ένα νέο μοντέλο αποσύνθεσης που χρησιμοποιεί διαστήματα εμπορικών συναλλαγών και εφαρμόζεται σε δεδομένα που μεταξύ δύο συναλλαγών η τιμή μπορεί να κινηθεί: ένα tick (η μικρότερη μη μηδενική αλλαγή της τιμής) επάνω (ή κάτω) ή καθόλου. Μοντελοποιούμε κάθε παράγοντα της τιμής διαδοχικά χρησιμοποιώντας ένα διωνυμικό μοντέλο GLARMA, και ένα GLM μοντέλο με μία λανθάνουσα διαδικασία. Πραγματοποιούμε προσομοίωσεις για να διερευνήσουμε την αποτελεσματικότητα των προτεινόμενων parameter driven μοντέλων χρησιμοποιώντας διάφορους αλγόριθμους μέσα σε ένα Μπεϋζιανό πλαίσιο. Αναλύουμε τα δεδομένα ES από την πλατφόρμα Globex του Chicago Mercantile Exchange μεταξύ 16 και 24 Μαΐου 2011. Για την προβλεπτική ικανότητα του μοντέλου, συγκρίνουμε το μέσο τετραγωνικό και απόλυτο σφάλμα (MSE, MAE), καθώς και τέσσερα μέτρα εκτίμησης: accuracy, recall, precision και specificity του πίνακα σύγχυσης (confusion matrix).

Abbreviations

ACD	autoregressive conditional duration
ACM	autoregressive conditional multivariate
ACMD	autoregressive conditional multinomial duration
AD	activity-direction
ADS	activity-direction-size
aGrad-z	auxiliary sampler based on z
AIC	Akaike information criterion
AM	adaptive Metropolis
APF	auxiliary particle filter
AR(1)	autoregressive model of order one
AR(1)-AD	first-order autoregressive activity-direction
ASIS	ancillary-sufficiency interweaving strategy
BHHH	Berndt–Hall–Hall–Hausman
CLT	central limit theorem
CP	centered parameterization
ES	E-mini Standard and Poor's (S&P) 500 index futures contract
ESS	effective sample size
ESS/sec	effective sample size per second
EST	eastern standard time
FTSE	financial times stock exchange
GARCH	generalized autoregressive conditional heteroskedasticit
GAS	generalized autoregressive score

GIS	global interweaving strategy
GIS-C	GIS centered parameterization
GIS-NC	GIS non-centered parameterization
GLARMA	generalized linear autoregressive moving average
GLM	generalized linear model
Globex	Chicago Mercantile Exchange's Globex
GPD	generalized Poisson difference
HFD	high frequency data
i.i.d.	independent and identically distributed
IBIS	iterated batch importance sampling
ICH	integer count Hurdle
INAR	integer autoregressive
Intel®MKL	Intel Math Kernel Library
LIFFE	London international financial futures and option exchange
LOB	limit order book
MAE	mean absolute error
MC	Monte Carlo
MCMC	Markov chain Monte Carlo
MEM	multiplicative error model
MH	Metropolis–Hastings
mle	maximum likelihood estimator
MSE	mean square error
NCP	non-centered parameterization
NYSE	New York Stock Exchange
OSV	ordinal stochastic volatility
PD	Poisson difference
pdf	probability density function
RW(1)	first-order random walk
SMC	sequential Monte Carlo
SSM	state space model

SV	stochastic volatility
WN	white noise
ZIB	zero-inflated binomial

Chapter 1

Introduction

The unique characteristics of high frequency data (HFD) have introduced new theoretical and computational challenges. The objective of this thesis is to propose and compare observation driven and parameter driven models for predicting integer price changes of high-frequency financial data in a Bayesian framework. Section 1.1 presents a review of HFD. A description of the available data set for this study is described in Section 1.2. Finally, Section 1.3 presents an overview of the ensuing chapters is given.

1.1 Literature review

HFD or ultra-high frequency data [Engle, 2000], are called the financial data in which all single events are recorded whenever they arrive together with its characteristics (such as time, price, volume, etc.). These data are also known as trade by trade data or tick data; tick is called the smallest non-zero price change, whose size depends on the institutional setting. One distinguishing feature of trading data is that the price change in consecutive trades occurs in integer multiples of a tick size. In practice, most price changes take only a small number of values and a significant proportion of them are identically zero. Another characteristic of such data is that the time between two consecutive transactions (duration) is not fixed but random. For more details about HFD see Engle and Russell [2004] and Tsay [2005, Chapter 5].

In econometric applications considered prior to the advent of HFD these unique characteristics was not taking into consideration. For example, the sampling frequency was chosen to be a fixed interval, e.g. the daily closing price of a share, and not a random variable as well as the distribution of financial asset returns were assumed to be continuous. For these reasons,

researchers have developed new statistical models for the analysis of such data. This section identifies models applied to such data, and how they are interrelated, focused mainly on the literature that has cited the publication of Rydberg and Shephard [1998a, 2003]. We mention some related important papers, however our list does not pretend to be complete.

Two main classes of models can be identified in the literature: parameter driven and observation driven in the classification of Cox et al. [1981]. The main difference between the two approaches is the way the serial dependence is introduced to the models. In the observation driven models, the serial dependence relies on lagged observations as well as past and present exogenous variables. In parameter driven models, the parameters are functions of an unobserved (latent or hidden) stochastic process, and the observations are independent conditionally on the latent variable.

Concerning the modelling of time between two consecutive transactions, Engle and Russell [1998] consider the autoregressive conditional duration (ACD) model in which the durations are treated as random variables and the conditional mean function is parameterized in terms of past events. A drawback of the model is that it requires additional restrictions to ensure positivity of duration. **?** propose the logarithmicic version of ACD that prevents the non-negativity restrictions; details on the properties of the model can be found in Bauwens et al. [2003]. Further developments of ACD model have been concerned, among others, by Bauwens and Veredas [1999]; Zhang et al. [2001]; Drost and Werker [2004]; Czado and Haug [2010]. In the first paper, the conditional duration is modeled with a latent variable.

The ACD model belongs to a family, named multiplicative error models (MEMs) proposed by Engle [2002], who provide an observation driven approach for dynamic non-negative variables; see Russell and Engle [2010] for a recent survey. The key idea behind the uni-variate MEMs is that the dynamics of the variable of interest are expressed as the product of the expectation of the process conditionally on the available information which depends on past values of the process and past expected values of the process, and a second independent and identically distributed (i.i.d.) process with unit mean. The conditions to ensure stationarityohe model proposed by Bauwens et al. [2004] who investigate ACD models for price, volume and trade duration data, the models proposed by Manganelli [2000], Engle and Gallo [2002] or Brownlees et al. [2011] who introduce the component MEM for intra-daily volumes.

Concerning the modelling of price process, Russell and Engle [1998, 2005] propose

an observation driven model called autoregressive conditional multivariate (ACM) for price changes, in combination with an ACD model for the durations between trades. The distribution of price changes are viewed as realizations of a multinomial distribution, conditional on past information and the time interval between the transactions. The vector of log-odds ratios is specified as a multivariate generalized linear autoregressive moving average (GLARMA) model with a logistic link function. The study adopted the maximum likelihood in estimating the parameters using a year of the high frequently traded New York Stock Exchange (NYSE) stock Airgas. The authors conclude that the transaction price variance increases with the duration and that long durations are associated with falling prices. GLARMAs, proposed by Shephard [1995], are an extension of the generalized linear models (GLMs, McCullagh and Nelder [1983]) suitable for dependent observations. Dunsmuir et al. [2015] provide a `glarma` `R`-package for estimating GLARMA models supporting Poisson, binomial and negative binomial distributions.

Another observation driven model is the autoregressive conditional multinomial duration (ACMD) model, introduced by Tay et al. [2004], in which price changes and time durations are modeled simultaneously. The price movements are generated by three competing independent Poisson processes representing the three possible tick movements: positive price change, negative price change, or no price change. Every transaction is recorded as an event of the type given by the Poisson process with the shortest duration. These processes are related through their intensity functions and described by ACD structures. The study adopted the maximum likelihood in estimating the parameters using a year of five shares traded at the NYSE using some explanatory variables

Alternative discrete price change models are based on decomposition techniques in which the price movements is decomposed into components and model each component sequentially using conditional specifications for the components. Rydberg and Shephard [1998a, 2003] propose the activity-direction-size (ADS) model in which the price change is defined as the product of a binary (a binary variable only takes two values) process for a change in price or not, a second binary process for a positive or negative change when one occurs and a count process for the size of price change in ticks. The dynamics of price direction are modeled by an autologistic model (logistic model including past returns in a binary model), while a binary and a negative binomial based GLARMA is used for the activity and size of price movements, respectively. They estimate the model via maximum likelihood using a year of IBM stock traded at the NYSE. The

variables considered include current and lagged values of the logarithmic of trade volume and the logarithmic of time duration between trades, a monthly trend, a long-term trend, and day of week dummy variables. The authors did not initially model duration. Rydberg and Shephard [2000] expand on the model by proposing the use of a Cox (doubly stochastic) process for the durations. An ACD model would be a specific case of a Cox process. Rydberg and Shephard [1999] extend the ADS model to the multivariate case. They apply it to a bivariate time series of the trade in the Ford and the GM shares, traded at the NYSE. Both models belong to the class of observation driven models.

Shahtahmassebi [2011] implement a Bayesian analysis of the ADS model applied to the financial times stock exchange (FTSE) 100 index futures for two days traded at the London international financial futures and option exchange (LIFFE), using Markov chain Monte Carlo (MCMC) methods. The model include one lagged value of the logarithmic trade volume as significant covariate for the size process, and one lagged value of the logarithmic duration for the activity and size process as significant covariates. Müller and Czado [2005], following Rydberg and Shephard [1998a, 2003], split the price process into components and model the absolute price changes with an autoregressive extension of the ordered probit model applied to data of the IBM stock traded on one day at the NYSE, using MCMC methods. The current values of the logarithmic duration and of the logarithmic traded volume are significant. Kent [2015], in an unpublished article, propose a model for price changes and time durations. The author extends the ADS model by defining the size of price change process to follow the ACM model and suggest a maximum likelihood estimator. Besides, the trade arrival process is defined to be a doubly-stochastic Poisson process (or Cox process) and suggest estimating its random intensity through kernel density estimation. The author provides details of the dataset used in this research, however results of the empirical study are not presented .

Following the idea of decomposition, Liesenfeld et al. [2006] propose the observation driven integer count Hurdle (ICH) model. The price is decomposed into a trinomial process for the occurrence of a negative change, no change or a positive change and a component for the size of change. For both components of the price process, the dynamics are modeled using a GLARMA specification, and the parametric model for the trinomial process is taken from the class of logistic ACM. To model the absolute size of the price change, the authors develop an extension of the hurdle model for counts [Mullahy, 1986] to work on negative as well as

positive integers. The inference is based on maximum likelihood estimators applied to the HAL and JBX stocks traded over a period of one trading month at the NYSE. Both models include a daily diurnal trend and current and one lagged values of price change direction, of logarithmic trade volume, and of the logarithmic duration as significant covariates.

McCulloch and Tsay [2001] propose a model for non-zero price changes and non-zero time durations. Similar to the ICH model, they decompose the price change into the direction and the size of price change. They use a hierarchical model that consists of six conditional generalized linear models, and estimate each of these components via MCMC methods. The analysis runs with different parameters each day, which are assumed to be i.i.d. following a normal distribution. The variables considered include only time durations between trades and price changes and not other explanatory variables, applied to data of the IBM stock taken from the NYSE for a period of three months. Dionne and Zhou [2016] consider the ADS model for investigating the impact of trade duration, quote duration and other exogenous variables on ex-ante liquidity embedded in an open limit order book (LOB). Hautsch et al. [2014] extent the literature of decomposition models by proposing a MEM a dynamic binary-choice part to model serially dependent positive-valued variables which realize a non-trivial proportion of zero outcome.

Models for high-frequency price changes that capture both the discreteness and the heteroscedasticity in the data have been developed. Hausman et al. [1992] use an ordered probit model to analyze conditional distribution of intraday price movements. They investigate the impact of several explanatory variables in capturing the transaction price changes, such as the time between trades, bid-ask spread, trade size and market-wide. They estimate the model via maximum likelihood using transactions data for over 100 randomly chosen U.S. stocks. The ordered probit model has its origins in bio-statistics; see Aitchison and Silvey [1957]. It is a generalization of the widely used probit analysis to the case of more than two outcomes of an ordinal dependent variable; probit analysis is a type of regression used to model dichotomous or binary outcome variables, in which the inverse standard normal distribution of the probability is modeled as a linear combination of explanatory variables.

Under the ordered probit framework, Müller and Czado [2009] consider a stochastic volatility (SV) model (see Shephard [2005]), while Yang and Parwada [2012] and Dimitrakopoulos and Tsionas [2019] adopt a generalized autoregressive conditional heteroskedasticit (GARCH)

model of Bollerslev [1986]. Müller and Czado [2009] introduce the class of ordinal stochastic volatility (OSV) to account for the discreteness of financial price changes with Gaussian and Student-t distributed errors. The model allows for exogenous factors both on the mean and volatility level. A Bayesian approach using MCMC is followed to facilitate estimation in the model applied to the IBM stock traded at the NYSE. In the paper of Czado et al. [2010] the applicability of the OSV model to financial stocks with different levels of trading activity is investigated. They conclude that a higher number of quotes between trades increases the volatility for less frequently traded stocks, whereas the opposite pattern is observed for stocks which are more frequently traded, while both volume and time elapsed between trades increase the volatility, thus making more extreme price changes more likely. Czado et al. [2010] also show that the OSV model outperforms the standard SV model when they are both applied to discrete financial price changes. Koopman et al. [2017] assume that the price changes are conditionally distributed according to a Skellam distribution with stochastic volatility. The estimation is based on simulated maximum likelihood and importance sampling methods applied on the assets traded on NYSE. The dynamic specification of the Skellam distribution have been explored by Koopman et al. [2014]. Barra et al. [2018] introduce a dynamic negative binomial difference model that measures stochastic volatility of discrete price changes. They provide the properties of this distribution, and emphasize that the distribution has heavy tails to address occurrences of jumps in prices changes. The model is applied on six stocks from the NYSE in two different periods, using Bayesian estimation procedures. GARCH models belong to the observation driven class, while SV to the parameter driven class.

Score driven models as introduced by Koopman et al. [2008]; Creal et al. [2011, 2013] and Harvey [2013] provide an observation driven approach for dynamic variables. The idea of these models is that time-varying parameter is updated on time via an autoregressive function that depends on a time-varying parameter by the conditional score, that is the first derivative of the logarithmic of the conditional probability density function (of the last observation) with respect to the same time-varying parameter. Known models are special cases of score driven models, for instance, the ACM and ACD models, the MEM, and many more. A recent review of these models is provided in Blazsek and Licht [2020]. The unknown coefficients of these models are estimated via a maximum likelihood. In a more general context, the forecasting performance of such models is investigated in detail by Koopman et al. [2016].

An alternative approach for modelling high frequency price changes is based on the Poisson difference (PD); Irwin [1937] discusses the difference for equal parameters and Skellam [1946] for unequal parameters. It is also called Skellam density; further details can be found in e.g. Karlis and Ntzoufras [2006] and Alzaid and Omair [2010]. In the latter paper, the authors fitted this density to model a set of high frequency data from the Saudi stock exchange using the maximum likelihood and the method of moments to estimate the parameters. Shahtahmassebi [2011] apply the PD density and its zero inflated version (to deal with the excess of zeros in the data) for modelling the price changes of high-frequency data. The zero version was first introduced by Karlis and Ntzoufras [2006] in order to model dental epidemiology. Shahtahmassebi and Moyeed [2014] propose a generalized Poisson difference (GPD) distribution, obtained as the difference of two underling generalized Poisson distributions with different intensity parameters. They derive the properties of the suggested density, and provide a zero inflated version of the distribution. In the last two works, the logarithmic of the expected value is modeled by a linear combination of the previous index change, the logarithmic of the traded volume and the logarithmic of the time between consecutive transactions, applied to the FTSE100 index changes using MCMC methods. All models belong to the observation driven class.

Contrary to the observation driven models, Czado and Kolbe [2004] study absolute price movements of option using a parameter driven Poisson GLM with an autoregressive model of order one (AR(1)) latent process in the mean. The model includes the price change of the underlying asset, the intrinsic value of the option at the time of the trade, the bid-ask spread, number of new quotes between trades, and the time duration between trades as significant covariates. Remaining time to maturity of the option is not significant for predicting option price changes. They have applied MCMC methods to perform Bayesian analysis with the S-Plus and WinBUGS software packages applied to the XETRA DAX index based on quote-by-quote data from the EUREX exchange. Shahtahmassebi [2011] propose a dynamic framework for modelling ultra high-frequency financial data using a zero inflated Poisson difference model to deal with the excess of zeros in the data. The analysis is carried out using sequential Monte Carlo (SMC) methods applied on FTSE100 index futures. A recent model in this class is proposed by Catania et al. [2019] based on the Skellam distribution.

Applications of integer autoregressive (INAR) models to high frequency data have been reported in, for instance, Alzaid and Omair [2014] and Andersson and Karlis [2014], amongst

others, who present extensions to $\mathbb{Z}$ of the INAR model with Skellam innovations. Al-Osh and Alzaid [1987] and Alzaid and Al-Osh [1990] propose INAR model of order one and p, respectively, to model non-negative integer-valued time series.

The idea to use copulas on high frequency discrete valued time series has been explored. Copulas are multivariate distribution functions with uniform margins which allow to represent a joint distribution function as a function of marginal distributions and a copula [Sklar, 1959]. **?** uses a four dimensional Gaussian copula to model price changes, transaction volumes, bid-ask spreads and intertrade durations jointly. For the price change is used the ICH model, whereas for the rest components are used ACD-type models. The model is applied to three stocks traded at the NYSE for one month, and all estimation results are obtained by jointly maximizing the log-likelihood. Bien et al. [2011] extend the ICH model to the multivariate case whereas the dependency between the marginals is modeled with a copula function. The authors model the conditional bivariate density of bid and ask quote changes in a high frequency setup, applied on the Citicorp stock for four days traded at the NYSE. Koopman et al. [2018] model price changes and investigate their dependence structures using dynamic copula models and a GAS specification with Skellam distribution, applied on price changes of stocks traded on the NYSE.

1.2 The data

The dataset used in this study is the E-mini Standard and Poor's (S&P) 500 index futures contract (ES) traded on the Chicago Mercantile Exchange's Globex (Globex) platform between May 16th 2011 and May 24th 2011. Globex is active from Monday to Friday, thus the dataset consists of seven days in total. The data comprises of trades and limit order activities, and contains information on the two best quotes on both bid and ask sides. We extract the data from 9:00 a.m. to 5:00 p.m. eastern time and we remove observations with spread (the difference between ask and bid price) zero or negative. We segment the data by day, and for data of each day, we again divide the data into morning and afternoon segments. The morning segment commences at 9:00 a.m, and concludes just before 1:00 p.m, while the afternoons begin at 1:00 p.m. and close at 5:00 p.m. This partitioning scheme allows us to explore the variations of trade data of different time segments. The first five days in the study are used for estimating the unknown parameters, and the rest for predicting price changes.

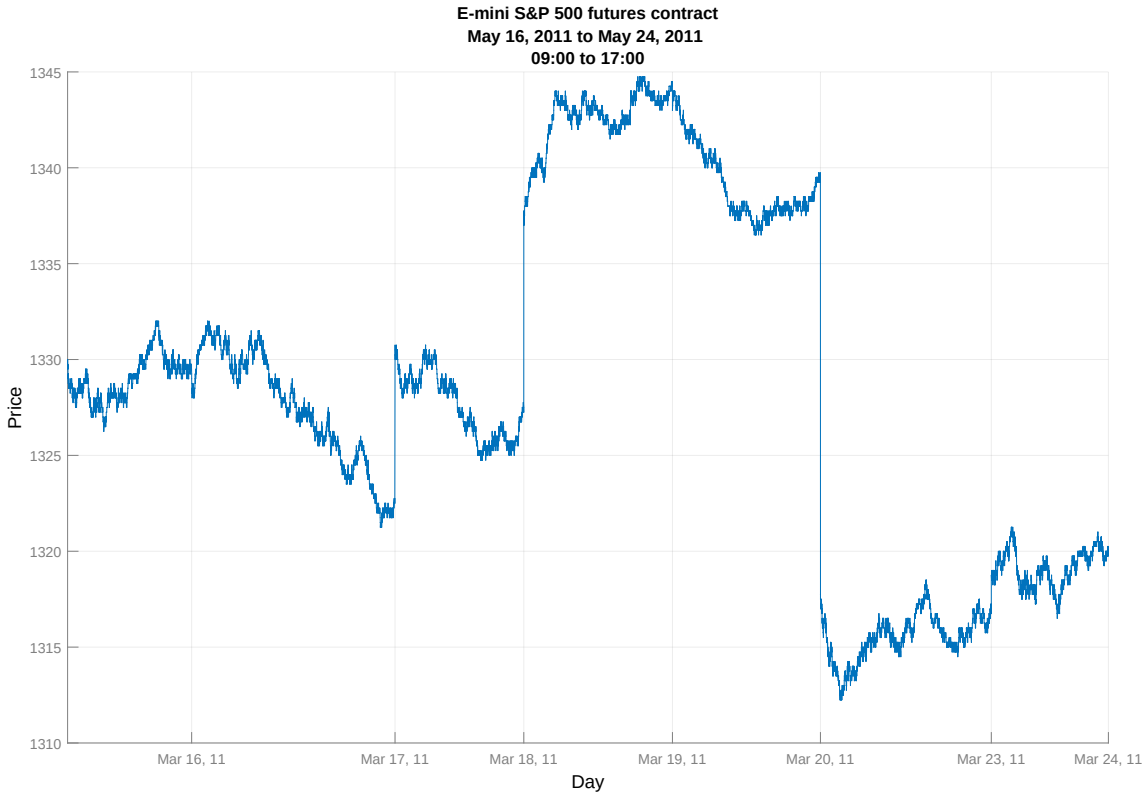


Figure 1.1: Time series plot of ES prices. The sample period covers Monday to Friday from May 16th 2011 to May 24th 2011, 9:00 a.m. to 5:00 p.m. eastern time. The labels along the x-axis depict the last transaction of the corresponding day. Tick size equals \$0.25.

Matching of trade and LOB data is achieved by an algorithm which is described in detail in Appendix A. Each final observation contains information on each trade including the date, the time, the transaction price, the number of futures contract traded, the type of market order (buy or sell), the bid/offer prices and volumes at levels 1, 2 of the LOB. Bid and ask quotes are the quotes immediately after the trade.

The total number of observations is $524,746$ from which $79,894$ and $444,852$ are reported on the morning and afternoon time period, respectively. The tick size is considered to be 1/4th of a dollar for each segment. In order to determine it we calculate the differences between adjacent elements of trade price within each day, and divide each element of the new vector with its minimum value verifying that the updated values are positive integers; the amount by which the vector is divided is the tick size. The above calculation is applied for each partition.

Table 1.2 provides daily summary statistics for variables within each day. The table consists of four main columns; the first column gives the variables names, while the last three main columns contain the line headings of the statistics being reported and they are splitted into two

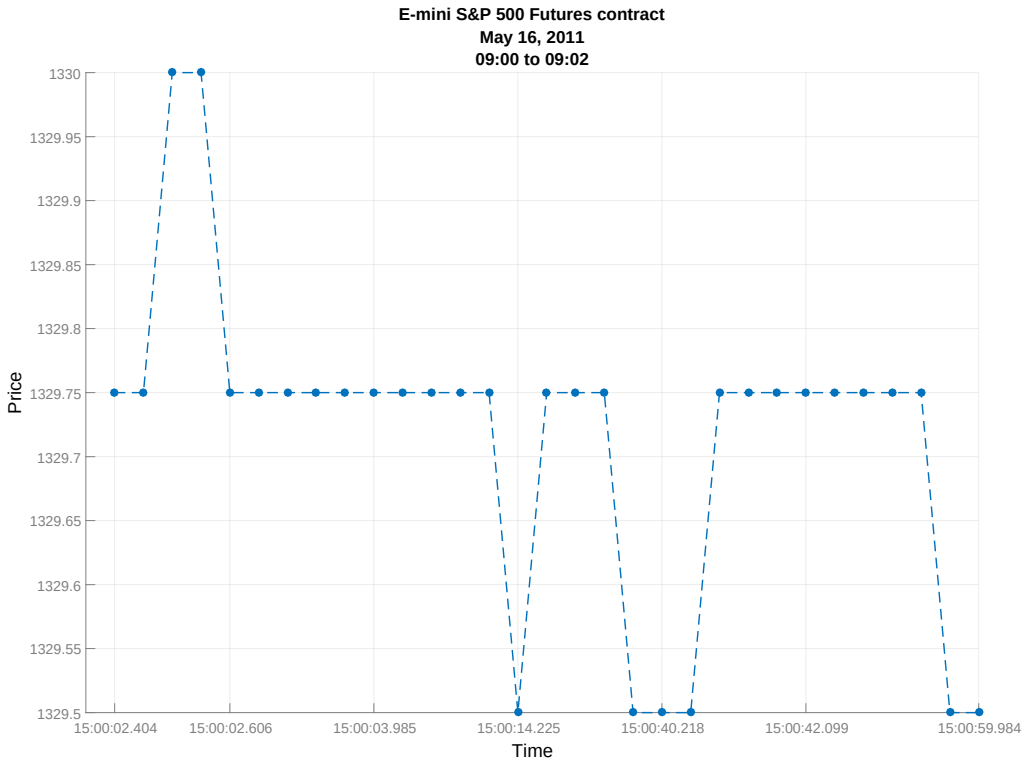


Figure 1.2: Time series plot of ES prices on 16 May, 9:00 to 9:02 am. Tick size equals $0.25.

sub-columns with names corresponding to the morning and afternoon time period. To compute all the statistics shown in the table, first for each variable we calculate the mean and the standard deviation within each day. Then we calculate the mean and the standard deviation across the seven days, to obtain daily statistics. According to the table, the daily average number of trades is substantially smaller in the morning than it is in the afternoon. The daily mean proportion of intraday active orders (ACT) in the morning is 18.8%, compared to 21.5% in the afternoon. On a daily average basis the half market orders and the half active market order are buyer-initiated (BUY/ACTBUY) within each day, and intraday upward (UP) or downward (DOWN) moves occur with almost equal probability. The table indicates that there is almost no spread at all for the average of ACT, UP, BUY and ACTBUY variables during the days, and a relatively stable statistics for these variables within each day. Regarding the traded volume (VOL) and the traded active volume (ACTVOL), the daily average of its intraday mean has almost multiplied 1.5-fold from 11.24 in the afternoon to 16.17 in the morning. Furthermore, the average of the traded volume is relatively constant across the days, while the traded volume changes considerably within each day. Moreover, the daily mean of the average intraday bid/ask volume on the second level

Table 1.1: Relative frequencies of price change of ES in multiple of tick size, from May 16th 2011 to May 24th 2011. Trading hours are Monday to Friday 9:00 a.m. to 5:00 p.m. eastern time. tick =$0.25

Number (tick)	≤ -2	-1	0	1	≥ 2
Percentage	0.01	10.53	78.94	10.51	0.01

(BIDVOLL2/ASKVOLL2) is almost twice than on the best level (BIDVOLL1/ASKVOLL1), while their values on each observed quote level on both sides of the market is quite the same. The table shows a relative wide-spread size of the volume (on both sides of the market) across different days and a wider-spread size of the volume within each day. Similar holds true for the active bid/ask volume (ACTBIDVOLL1,ACTBIDVOLL2,ACTASKVOLL1,ACTASKVOLL2). Additionally, the daily mean of the intraday time duration between trades (DUR) average as well as the daily mean of the intraday time duration between active trades (ACTDUR) average seem to be shortest after 1 p.m. than it is before, while the second value is seven times the value of the first. In the morning, time duration between active trades within each day varies more than in the afternoon and more than the corresponding values of time duration between trades. For the spread (SPREAD) and the active spread (ACTSPREAD), the daily average of its intraday value equals approximately one tick and the table indicates that there is almost no spread at all for the average of the variable during the days, while the spread is peaked at number one within each day.

In Figure 1.1, the ES is plotted from May 16 through May 24, 9 a.m. to 5 p.m. The labels along the x-axis depict the last transaction of the corresponding day. May 16 and 23 correspond to the beginning of two consecutive working weeks starting on Monday. The intervals do not have the same length because the number of transaction varies daily. Note that there is a substantial drop in the price of 18.5 units just on the opening of the second week, ie. on May 23. Furthermore, on May 18 and 24 the price of the first transaction increases significantly 10.25 and 4.25 units, respectively. Figure 1.2 displays the prices for the day of May 16, 2011 at 9:00-9:02 in which it can be noted the price discreteness and the unequal spaced time intervals. Besides, Table 1.1 shows the percentage of the transaction price changes between 09:00 and 17:00. It seems that most of the transactions (78.94%) are the same as their previous values. The distribution of transaction price changes is roughly symmetric as down and up one ticks occurs with 10.53% and 10.51% frequency. Furthermore, both downwards and upwards moves

greater than two ticks occurs too rarely with frequency 0.01%.

Table 1.2: Daily statistics for the front month E-mini S&P 500 futures contract. The sample period covers from May 16th 2011 to May 24th 2011, including 7 trading days. The statistics are reported separately for two sub-periods: morning (9 a.m. to 1 p.m.) and afternoon (1 p.m. to 5 p.m.) with 79,894 and 444,852 observations. The daily average number of trades is 11,420.43 and 63,557.29, respectively. The variable names are listed in the first column in the table. ACT, UP and BUY denote the percentage of the active trades, upward trades and buy market orders, respectively. VOL denotes the traded volume. ASKVOLL1 and BIDVOLL1 indicate the ask and bid volume on the best quote level. ASKVOLL2 and BIDVOLL2 denote the ask and bid volume on the second level. DUR denotes the duration time between two successive trades in seconds. SPREAD (in ticks) is the bid-ask spread. ACT'X' denotes the variable X restricted on active trades. The tick size equals \$0.25. $\mathbb{E}(\mathbb{E}(X))$ and $\text{sd}(\mathbb{E}(X))$ return the daily average and the daily standard deviation of the mean of the variable X within each day, respectively. $\mathbb{E}(\text{sd}(X))$ returns the daily average of the standard deviation of the variable X intraday.

Variable X	$\mathbb{E}(\mathbb{E}(X))$		$\text{sd}(\mathbb{E}(X))$		$\mathbb{E}(\text{sd}(X))$	
	Morning	Afternoon	Morning	Afternoon	Morning	Afternoon
ACT	18.8%	21.5%	0.02	0.01	0.39	0.41
UP	49.9%	50.0%	0.00	0.00	1.00	1.00
BUY	48.0%	49.9%	0.02	0.01	0.49	0.50
VOL	11.24	16.17	1.00	0.95	35.60	55.41
ASKVOLL1	254.24	609.86	32.36	56.57	244.11	517.66
BIDVOLL1	234.36	583.02	66.50	57.65	215.45	494.73
ASKVOLL2	531.47	1390.20	106.85	91.21	254.18	531.87
BIDVOLL2	528.75	1351.54	141.85	121.72	232.84	519.47
DUR	1.30	0.23	0.24	0.04	3.10	0.57
SPREAD	1.10	1.04	0.01	0.00	0.31	0.20
ACTBUY	48.0%	49.9	0.00	0.00	0.50	0.50
ACTVOL	17.67	17.86	2.17	1.51	55.55	68.53
ACTASKVOLL1	238.12	615.27	34.42	54.94	220.11	478.97
ACTBIDVOLL1	229.92	601.02	67.81	64.19	189.88	468.38
ACTASKVOLL2	527.60	1396.35	106.39	92.32	245.38	519.37
ACTBIDVOLL2	528.44	1376.29	142.58	125.80	229.38	524.41
ACTDUR	6.98	1.08	1.44	0.18	11.42	1.97
ACTSPREAD	1.06	1.02	0.01	0.01	0.26	0.15

1.3 Outline of the thesis

In Chapter 2, a review of MCMC and state space models is given. These computational techniques constitute an essential ingredient of Bayesian estimation and will be used throughout the thesis.

In Chapter 3, both maximum likelihood and Bayesian estimation, via MCMC and SMC methods, for the observation driven model activity-direction (AD) is conducted applied on ES. Since only a few transactions result in a price change of more than one tick we convert all price movements in our data to indicators 1, 0 and –1, corresponding to price increases, no change, and price decreases respectively. The time gaps between trading days are ignored. The price movement is specified as a product of a binary variable A_i on $\{0, 1\}$ defining the market activity (the price moves or not) and a binary variable D_i on $\{-1, 1\}$ defining a negative or positive price move (if a change occurs). Hence, the transaction price evolves over time by

$$P_{t_i} = P_{t_i} + A_i D_i,$$

where P_{t_i} denotes the transaction price at time t_i. We model each component sequentially using a first order GLARMA model to investigate serial dependence in the data. In MCMC, samples are drawn from the posterior by the adaptive Metropolis (AM) algorithm, proposed by Haario et al. 2001. In SMC, we use the iterated batch importance sampling (IBIS) algorithm, proposed by Chopin [2002], where the algorithm runs three times over the whole data sequence, with different number of particles and discuss how the estimated parameters are affected. The resampling threshold is set to 80%, and the resampling method is the stratified resampling scheme. Simulation results are conducted to evaluate the performance of the proposed methods.

In Chapter 4, we extend the AD model by proposing a parameter driven model for each price component based on the Bernoulli distribution. Specifically, we model the logit of the probability of success at each time point of each price factor as a linear function of regression variables, and a latent process; We examine the autoregressive model of order one (AR(1)) and white noise (WN) process. The estimation is performed using MCMC algorithms where we want to estimate both the latent state and the model parameter of each component, conditional on the observed data, in its centered and non-centered parametrization form. We investigate two methods for estimating the latent process: Firstly, we sample the components of the latent state

one at time in separate Metropolis–Hastings (MH) steps [Metropolis et al., 1953] conditional on all others values of the state process and on the parameter vector, and secondly we update the whole process in one move focused on the method proposed by Titsias [2011]; Titsias and Papaspiliopoulos [2018]. Besides, the autoregressive coefficient and the conditional precision are sampled in one move, and in different MH steps. In both samplers, the regression parameters are drawn jointly from their posterior density. Besides, we apply the interweaving sampler [Yu and Meng, 2011], which combines both parametrizations by interweaving the two strategies in order to improve MCMC mixing. We perform a simulation study to investigate the effectiveness of the binomial parameter driven model using different algorithms by evaluating the effective sample size per second (ESS/sec).

The observation and parameter driven AD models are applied on the ES data. We include wo separate conditioning information sets; that is, different set of explanatory variables. The first set includes only past values of the processes. The second set includes these variables plus several market microstructure variables, including two lags of the logarithmic duration, the logarithmic traded volume, the spread, the logarithmic bid and ask volume on the two best observed quote levels. Additionally, we include two lags of a dummy variable indicating that a trade is a buy market order, and two lags of a dummy variable denoting that a trade moves up the price. We apply the Brier score [Brier, 1950; Blattenberger and Lad, 1985] for model comparison of each price component. In order to assess the predictive performance, we compare the mean square error (MSE) and mean absolute error (MAE) criterion, as well as four scalar performance measures, namely, accuracy, sensitivity, precision and specificity derived from the confusion matrix.

In Chapter 5, we propose a new decomposition model that uses trade intervals and is applied on data that allow three possible tick movements: one tick up price change, one tick down price change, or no price change. We aggregate observations over a small number of trade intervals (denoted trade bars), assign each interval the last price included in it, and then use the price change between the two consecutive intervals for analyzing the process of price changes. Following the idea of decomposition, we define the price change as the subtraction of two binomial processes. Specifically, the transaction price evolves over time by

$$P_{t_i} = P_{t_i} + 2D_i - A_i,$$

where P_{t_i} denotes the transaction price of the last price included in the ith trade bar which is recorded at time t_i. A_i and D_i denote the number of active trades and the number of upward moves (if at least one change occurs) during the ith trade bar. We assume that the activity (direction) binary variables that belong to the ith trade bar follow a Bernoulli distribution with the same probability. Based on this decomposition, we model each component sequentially with binomial response data. Firstly, a generalized linear autoregressive moving average (GLARMA) model is examined, and secondly, it is assumed that the data are governed by a latent process: an AR(1), a WN and a random walk of order, and use MCMC techniques for estimation. We describe the binomial AD component model with Student's t errors. The estimation methods and the algorithms that are used with the corresponding Bernoulli models are applied here.

Table 1.3: Relative frequencies (in percentage) of price change of ES in multiple of tick size, from May 16th 2011 to May 24th 2011. Trading hours are Monday to Friday 9:00 a.m. to 5:00 p.m. eastern time. The time gap between trades is ignored. N denotes the size of trade interval. tick = \$0.25.

Number (tick) / N	-2	-1	0	1	2
2	0.05	12.14	75.66	12.10	0.06
5	0.16	14.40	70.95	14.30	0.18

The ES data are compressed into trade bars of size two and five. Table 1.3 shows the percentage of the transaction price changes between 09:00 and 17:00 after the aggregation. It seems that most of the transactions are the same as their previous values, while the distribution of transaction price changes is roughly symmetric as down and up one ticks occur with almost equal frequency. Furthermore, both downward and upward two tick moves occur too rarely. The time gaps between trading days are ignored. We adopt a conditioning information set which includes two lags of the number of active trades and the number upward moves, as well as two lags of a dummy variable denoting that a trade moves up the price. Additionally, we include two lags of the logarithmic duration, the logarithmic traded volume, the spread, the logarithmic bid and ask volume on the two best observed quote levels, the type of market order update action, as well as a dummy variable indicating that a trade is a buy market order. In order to assess the predictive performance, we compare the MSE and MAE criterion, as well as four scalar performance measures, namely, accuracy, sensitivity, precision and specificity derived

from the confusion matrix. We compare the models with an alternative model that assume the best prediction for tomorrow's market price is simply today's price.

Finally, Chapter 6 summarizes the main conclusions of the thesis and provides future work suggestions.

All experiments are conducted on a desktop PC with an Intel Core i7-3770 processor, running at 3.40GHz with 16.0 GB of RAM available for the tools, and estimations are implemented in MATLAB 2013b and C with the Intel®MKL and compiled using Intel C++ 13.0 (with Microsoft Visual C++ 2012 linker).

Chapter 2

Bayesian inference

2.1 Introduction

The object of this chapter is to present the theory that will be used throughout the thesis. This chapter is organized as follows. Section 2.2 gives a brief overview of Bayesian inference. Section 2.3 summarizes posterior sampling methods for time-invariant parameters, such as MCMC and SMC. Section 2.4 introduces the main concepts of state space models and gives details of one important state equation: the autoregressive model of order one. Section 2.5 introduces Bayesian inference for non-Gaussian state space models using MCMC methods for estimating the latent path and the model parameter. Besides, it reviews sequential estimation of the filtering density using the auxiliary particle filter (APF).

2.2 Brief overview of Bayesian inference

Let $\boldsymbol{y} \triangleq (y_1, \cdots, y_n)^\top$ be a $(n \times 1)$ vector of observations, where the symbol $\triangleq$ means 'is equal by definition to'. It is assumed that the data are generated from a probability density, $\pi(\boldsymbol{y}|\boldsymbol{\theta})$, where $\boldsymbol{\theta}$ is an unknown parameter vector in the parameter space Θ. We want to infer some properties of the unknown parameter based on the data as well as to predict future data.

There are two methods to estimate these parameters: classical analysis and Bayesian analysis. In classical analysis, parameters are assumed to be fixed contrary to Bayesian analysis in which parameters are assumed to be random variables and assigned a suitable prior distribution $\pi(\boldsymbol{\theta})$. The prior distribution captures our beliefs about the parameters before looking at the data. The basic idea of Bayesian inference is to combine prior beliefs about the unknown

parameters and the information about the parameter that is available in the observations to produce the posterior distribution (or else the target distribution) denoted as $\pi(\boldsymbol{\theta}|\boldsymbol{y})$. According to Bayes' theorem (or Bayes' rule, Bayes, 1763) it is given by

$$\pi(\boldsymbol{\theta}|\boldsymbol{y}) = \frac{\pi(\boldsymbol{y}, \boldsymbol{\theta})}{\pi(\boldsymbol{y})} = \frac{\pi(\boldsymbol{\theta})\pi(\boldsymbol{y}|\boldsymbol{\theta})}{\pi(\boldsymbol{y})}. \tag{2.1}$$

The quantity $\pi(\boldsymbol{y}|\boldsymbol{\theta})$ viewed as a function of the parameter vector $\boldsymbol{\theta}$, is called the likelihood function. It quantifies the probability that the observed data would have been observed as a function of the unknown model parameters. The likelihood is not a probability distribution in $\boldsymbol{\theta}$ (and it does not integrate necessarily to 1), but integrate to 1 over the possible values of $\boldsymbol{y}$.

The quantity $\pi(\boldsymbol{y})$ is called the marginal likelihood and it can be seen as a normalizing constant to make the integral of posterior density to be one. Integrating both sides of (2.1) with respect to $\boldsymbol{\theta}$, we have

$$\pi(\boldsymbol{y}) \triangleq \int_{\Theta} \pi(\boldsymbol{\theta})\pi(\boldsymbol{y}|\boldsymbol{\theta})\, d\boldsymbol{\theta}. \tag{2.2}$$

The marginal likelihood can be used to compare models by computing Bayes factors [Kass, 1993; Kass and Raftery, 1995]. Since it is independent of unknown parameters, the posterior density can be written in a more compact form

$$\pi(\boldsymbol{\theta}|\boldsymbol{y}) \propto \pi(\boldsymbol{y}|\boldsymbol{\theta})\pi(\boldsymbol{\theta}), \tag{2.3}$$

where the symbol $\propto$ means 'is proportional to'. Equation (2.3) is fundamental in Bayesian analysis and states that the posterior distribution of model parameters is proportional to the likelihood and prior probability distribution.

Forecasting a future value of an observation y' given the data $\boldsymbol{y}$ is solved by computing the conditional distribution

$$\pi(y'|\boldsymbol{y}) \triangleq \int_{\Theta} \pi(y', \boldsymbol{\theta}|\boldsymbol{y})\, d\boldsymbol{\theta} = \int_{\Theta} \pi(y'|\boldsymbol{y}, \boldsymbol{\theta})\pi(\boldsymbol{\theta}|\boldsymbol{y})\, d\boldsymbol{\theta}, \tag{2.4}$$

which is called (posterior) predictive distribution. In the common case of independent sampling, that is when the future observation y' is independent of $\boldsymbol{y}$, (2.4) becomes

$$\pi(y'|\boldsymbol{y}) = \int_{\Theta} \pi(y'|\boldsymbol{\theta})\pi(\boldsymbol{\theta}|\boldsymbol{y})\, d\boldsymbol{\theta}.$$

Often we are not interested in the entire posterior distribution $\pi(\boldsymbol{\theta}|\boldsymbol{y})$ but, rather, in some

feature of it, for instance, the posterior expectation of some integrable function $f_0(\cdot)$,

$$\mathbb{I}(f_0) \triangleq \mathbb{E}_{\pi(\boldsymbol{\theta}|\boldsymbol{y})}[f_0(\boldsymbol{\theta})] = \int_{\Theta} f_0(\boldsymbol{\theta})\pi(\boldsymbol{\theta}|\boldsymbol{y})\, d\boldsymbol{\theta}, \tag{2.5}$$

where $\mathbb{E}_\pi$ denotes the expectation of a random variable with distribution π. A common obstacle is that such integrals may be analytically intractable (or the computation is highly expensive). However, given a sample from the posterior distribution, we can use Monte Carlo (MC, Metropolis and Ulam, 1949; von Neumann, 1963) integration and approximate them. Let $\{\boldsymbol{\theta}^{(l)}\}_{l=1}^N$ be an i.i.d. sample generated from $\pi(\boldsymbol{\theta}|\boldsymbol{y})$. The target density (which may be continuous) is approximated by a discrete probability given by

$$\hat{\pi}(\boldsymbol{\theta}|\boldsymbol{y}) \triangleq \frac{1}{N}\sum_{l=1}^{N} \delta(\boldsymbol{\theta} - \boldsymbol{\theta}^{(l)}), \tag{2.6}$$

where $\delta(\boldsymbol{\theta} - \boldsymbol{\theta}')$ denotes a Dirac point-mass located at the point $\boldsymbol{\theta}' \in \Theta$. Substitution this approximation to (2.5) results in

$$\hat{I}(f_0) \triangleq \int f_0(\boldsymbol{\theta}) \frac{1}{N}\sum_{l=1}^{N} \delta(\boldsymbol{\theta} - \boldsymbol{\theta}^{(l)})\, d\boldsymbol{\theta} = \frac{1}{N}\sum_{l=1}^{N} f_0(\boldsymbol{\theta}^{(l)}), \tag{2.7}$$

which is called the MC estimator. This estimator is unbiased, and the accuracy of it increases with N. Besides, if the variance of $f_0(\boldsymbol{\theta})$ is finite, a central limit theorem (CLT) holds; see Robert and Casella [2013, Chapter 3]. The key idea behind MC is that an integral which cannot be computed exactly, is transformed into a tractable finite sum.

For most inference problems, generating samples from the posterior density is not usually feasible, because the (high-dimensional) integral in the denominator (2.1) may be not possible to be solved hence the posterior distribution will only be known up to a normalizing constant; that makes $\pi(\boldsymbol{\theta}|\boldsymbol{y})$ analytically intractable.

2.3 Bayesian inference for θ

Suppose that we are interested in sampling the target distribution $\pi(\boldsymbol{\theta}|\boldsymbol{y})$ over a parameter vector $\boldsymbol{\theta} \in \Theta$, and to estimate functions depending on the desired density. Assume that the posterior density admits no closed form expression, and it is known up to a normalizing constant. To address this problem we have to resort to methods, such as (adaptive) MCMC and SMC for

time-invariant parameters.

2.3.1 Markov chain Monte Carlo

Here we describe the basic idea of MCMC and the widely known Metropolis–Hastings (MH) algorithm.

2.3.1.1 Basic idea of MCMC

A first order Markov chain is a sequence of random variables $\{\boldsymbol{\theta}^{(l)}\}_{l\geq 0}$ with the property that the next state depends only on the current state and not on the entire past states of the process; this property is known as the Markov property. We call $\boldsymbol{\theta}^{(l)}$ the state of the process and the set of possible values of the random variables is called the state space. The distribution of $\boldsymbol{\theta}^{(0)}$ is the initial distribution of the chain. The joint distribution of the Markov chain is determined fully by its initial distribution and its transition kernel. A detailed survey for the Markov chain theory is provided in Grimmett et al. [2001], Robert and Casella [2013, Chapter 6].

While some Markov chains forget their starting conditions in the long term, others do not. The key idea in MCMC is to create a Markov chain that will gradually forget its initial state and converge to the unique stationary (or invariant) distribution $\pi(\boldsymbol{\theta}|\boldsymbol{y})$, which does not depend on $\boldsymbol{\theta}^{(0)}$. If a chain has a proper invariant distribution $\pi(\boldsymbol{\theta}|\boldsymbol{y})$, it is irreducible (Markov chain that eventually reaches every set that has a positive probability, no matter where it starts) and aperiodic (it should not oscillate between any two point in a periodic manner), then $\pi(\boldsymbol{\theta}|\boldsymbol{y})$ is the unique invariant distribution and is also the equilibrium distribution of the chain [Tierney, 1994, Theorem 1].

When $\{\boldsymbol{\theta}^{(l)}\}_{l=1}^{N}$ is a simulated path of the constructed chain, the estimator (2.7) holds and it is called ergodic average. Provided that the Markov chain is ergodic with stationary distribution $\pi(\boldsymbol{\theta}|\boldsymbol{y})$, it converges to (2.5) almost surely as N tends to infinity [Tierney, 1994, Theorem 3]. Besides, under certain assumptions on the transition kernel, a CLT holds; see Tierney [1994, Theorem 4 and 5], Robert and Casella [2013, Theorem 6.65 and 6.67].

In practice, the chain must be initialized with some initial value and runs for a sufficiently long time, N. Then, an initial number of iterations, m, is discarded; this number is called burn-in and denotes the number of iterations towards the beginning when the chain has not converged yet. The remaining values, $\{\boldsymbol{\theta}^{(l)}\}_{l=m+1}^{N}$, will be the dependent sample approximately from the

Algorithm 1 Draw N samples from $\pi(\boldsymbol{\theta}|\boldsymbol{y})$ using the MH algorithm [Metropolis et al., 1953; Hastings, 1970].

1: Start from an initial sample $\boldsymbol{\theta}^{(0)}$, drawn from any density π_0, with the requirement $\pi_0(\boldsymbol{\theta}^{(0)}|\boldsymbol{y}) > 0$.
2: **for** $l = 1, 2, \cdots, N$ **do**
3: Draw a candidate value $\boldsymbol{\theta}'$ from the proposal distribution $q(\cdot|\boldsymbol{\theta}^{(l-1)})$.
4: Calculate the ratio $r = \dfrac{\pi(\boldsymbol{\theta}'|y)}{\pi(\boldsymbol{\theta}^{(l-1)}|y)} \dfrac{q(\boldsymbol{\theta}^{(l-1)}|\boldsymbol{\theta}')}{q(\boldsymbol{\theta}'|\boldsymbol{\theta}^{(l-1)})}$. ▷ Any constants in the expression cancel out.
5: Set $\boldsymbol{\theta}^{(l)} = \boldsymbol{\theta}'$ with probability $\min(1, r)$, otherwise set $\boldsymbol{\theta}^{(l)} = \boldsymbol{\theta}^{(l-1)}$.
6: **end for**
7: **return** $\boldsymbol{\theta}^{(1)}, \cdots, \boldsymbol{\theta}^{(N)}$.

desired density and can be used to estimate the required quantities.

2.3.1.2 The MH algorithm

A method for constructing a Markov chain for obtaining approximate draws from the target density is the Metropolis algorithm which was first introduced by Metropolis et al. [1953], and generalized by Hastings [1970]. In a nutshell the MH algorithm works as follows: Let $\boldsymbol{\theta}^{(0)}$ be an initial value within the domain of the desired density. At iteration l, given that the current value of the chain is $\boldsymbol{\theta}^{(l-1)}$, a new state $\boldsymbol{\theta}'$ is drawn from a proposal kernel Q on Θ, with density $q(\cdot|\boldsymbol{\theta}^{(l-1)})$, which is termed proposal (or a candidate) distribution. The proposal value is accepted with probability $\alpha(\boldsymbol{\theta}^{(l-1)}, \boldsymbol{\theta}') \triangleq \min(1, r)$ where r is called the acceptance ratio given by

$$r \triangleq \frac{\pi(\boldsymbol{\theta}'|y)}{\pi(\boldsymbol{\theta}^{(l-1)}|y)} \frac{q(\boldsymbol{\theta}^{(l-1)}|\boldsymbol{\theta}')}{q(\boldsymbol{\theta}'|\boldsymbol{\theta}^{(l-1)})}, \tag{2.8}$$

otherwise it stays $\boldsymbol{\theta}^{(l-1)}$.

In the original paper of Metropolis et al. [1953] the proposal distribution must be only symmetric around the current value $\boldsymbol{\theta}^{(l-1)}$, such that $q(\boldsymbol{\theta}'|\boldsymbol{\theta}^{(l-1)}) = q(\boldsymbol{\theta}^{(l-1)}|\boldsymbol{\theta}')$. The random walk Metropolis algorithm arises when the sampler proposes $\boldsymbol{\theta}' = \boldsymbol{\theta}^{(l-1)} + \boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon}$ is drawn from a zero mean distribution with variance σ^2 (such as a standard normal or t-distribution). Algorithm 1 presents the general MH method. The samples generated using the MH eventually converges to the desired distribution; see Gilks et al. [1995]; Robert and Casella [2013].

Gibbs sampler [Geman and Geman, 1984]is a special case of MH algorithm with acceptance rate equals to one. It can only be used when the full conditional distribution (the

conditional distribution of one component or block, given all of the other ones) is known in closed form.

2.3.2 Adaptive MCMC methods

The MH algorithm can be used to sample from any target distribution, but at the expense of having to choose a proposal distribution which plays an important role for the success of the algorithm. Often this problem is done manually, though this can be difficult especially in high dimensions. An alternative approach is adaptive MCMC, which automatically changes the proposal distribution. We focus on the adaptive Metropolis (AM) and adaptive random walk Metropolis-within-Gibbs algorithm. For broader perspectives on this topic see Givens and Hoeting [2012, Chapter 8] and Damien et al. [2013, Chapter 7].

2.3.2.1 Adaptive Metropolis

The first adaptive MCMC algorithm, which adapts continuously to the target distribution, is the AM [Haario et al., 2001]. The key idea behind this approach is that the algorithm will be optimal if the proposal covariance matrix is chosen to be $2.38^2/d$ times the covariance matrix of the target, where d is the dimension of the target [Gelman et al., 1996; Roberts et al., 1997, 2001]. Since the true covariance matrix is rarely known, Haario et al. [2001] estimate it with the empirical covariance, and the candidate covariance is updated at each iteration using the past values of the simulations. In a nutshell the algorithm works as follows: It starts from an initial index l_0 and initial covariance $\boldsymbol{\Sigma}_0$ which can be chosen as the identity if we have no prior knowledge about. At the lth iteration, define the candidate covariance matrix

$$\boldsymbol{\Sigma}_{(l)} \triangleq \begin{cases} \boldsymbol{\Sigma}_0 & \text{if } l \leq l_0 \\ (2.38^2/d)\boldsymbol{\Sigma}_{(l-1)} + \epsilon \mathbf{I}_d & \text{otherwise,} \end{cases}$$

for some small $\epsilon > 0$ that prevents the covariance matrix from becoming singular. Then, propose $\boldsymbol{\theta}'$ from a Gaussian distribution centered at the current value $\mathcal{N}_d(\boldsymbol{\theta}^{(l-1)}, \boldsymbol{\Sigma}_{(l)})$, and accept or reject the same way as in MH algorithm with acceptance probability $\min(1, \pi(\boldsymbol{\theta}'|\boldsymbol{y})/\pi(\boldsymbol{\theta}^{(l-1)}|\boldsymbol{y}))$. The covariance matrix is calculated iteratively to avoid the method becoming computationally expensive. Haario et al. [2001] prove that the samples come from the target distribution and the algorithm is ergodic for bounded functions.

Algorithm 2 Draw N samples from $\pi(\boldsymbol{\theta}|\boldsymbol{y})$ using the AM algorithm [Haario et al., 2001; Roberts and Rosenthal, 2009].

1: Start from an initial sample $\boldsymbol{\theta}^{(0)}$, drawn from any density π_0, with the requirement $\pi_0(\boldsymbol{\theta}^{(0)}|\boldsymbol{y}) > 0$. ▷ $\boldsymbol{\theta}$ contains real-valued parameters
2: **for** $l = 1, 2, \cdots, 2d$ **do**
3: Draw a candidate value $\boldsymbol{\theta}'$ from the proposal distribution $\mathcal{N}_d(\boldsymbol{\theta}^{(l-1)}, \boldsymbol{\Sigma}_0)$.
4: Calculate the ratio $r = \pi(\boldsymbol{\theta}'|y)/\pi(\boldsymbol{\theta}^{(l-1)}|y)$.
5: Set $\boldsymbol{\theta}^{(l)} = \boldsymbol{\theta}'$ with probability $\min(1, r)$, otherwise set $\boldsymbol{\theta}^{(l)} = \boldsymbol{\theta}^{(l-1)}$.
6: **end for**
7: Evaluate $\boldsymbol{\mu}_{(2d)} \triangleq [\mu_{2d,1}, \cdots, \mu_{2d,d}]^\top$, where $\mu_{2d,j} \triangleq (1/2d) \sum_{l=1}^{2d} \theta_j^{(l)}$ for $j \leq d$; $\mu_{2d,j}$ and $\theta_j^{(l)}$ denote the jth element of vector $\boldsymbol{\mu}_{(2d)}$ and $\boldsymbol{\theta}^{(l)}$, respectively. ▷ $d \times 1$ vector
8: Evaluate $\boldsymbol{\Sigma}_{(2d)} \triangleq \sum_{l=1}^{2d} (\boldsymbol{\theta}^{(l)} - \boldsymbol{\mu}_{(l)})(\boldsymbol{\theta}^{(l)} - \boldsymbol{\mu}_{(l)})^\top$. ▷ $d \times d$ matrix
9: **for** $l = 2d + 1, 2d + 2, \cdots, N$ **do**
10: Draw a candidate value $\boldsymbol{\theta}'$ from $(1 - \beta)\mathcal{N}_d(\boldsymbol{\theta}^{(l-1)}, c\boldsymbol{\Sigma}_{(l-1)}) + \beta\mathcal{N}_d(\boldsymbol{\theta}^{(l-1)}, \boldsymbol{\Sigma}_0)$.
11: Repeat steps 4 and 5.
12: Update the covariance matrix of the proposal density: $\boldsymbol{\mu}_{(l)} = \boldsymbol{\mu}_{(l-1)} + (1/l)(\boldsymbol{\theta}^{(l)} - \boldsymbol{\mu}_{(l-1)})$ and $\boldsymbol{\Sigma}_{(l)} = \boldsymbol{\Sigma}_{(l-1)} + (1/l)\left((\boldsymbol{\theta}^{(l)} - \boldsymbol{\mu}_{(l-1)})(\boldsymbol{\theta}^{(l)} - \boldsymbol{\mu}_{(l-1)})^\top - \boldsymbol{\Sigma}_{(l-1)}\right)$.
13: **end for**
14: **return** $\boldsymbol{\theta}^{(1)}, \cdots, \boldsymbol{\theta}^{(N)}$.

Roberts and Rosenthal [2009] propose a new version of the above algorithm in which the lth proposal covariance matrix is a mixture of the form

$$\boldsymbol{\Sigma}_{(l)} \triangleq \begin{cases} \boldsymbol{\Sigma}_0 & \text{if } l \leq 2d \\ (2.38^2/d)\boldsymbol{\Sigma}_{(l-1)} + \beta\boldsymbol{\Sigma}_0 & \text{otherwise,} \end{cases}$$

where $\boldsymbol{\Sigma}_0 \triangleq (0.1^2/d)\mathbf{I}_d$ and $\beta \in (0, 1)$ to avoid the algorithm being stuck with a singular covariance matrix. Algorithm 2 presents this version of AM algorithm.

An important advantage of the AM algorithm is that it starts using the cumulating information from the beginning of the run, ensuring that the search becomes more effective at an early stage [Haario et al., 2001].

2.3.2.2 Adaptive random walk Metropolis-within-Gibbs

The key idea behind the adaptive random walk Metropolis-within-Gibbs [Roberts and Rosenthal, 2009, Section 3] is that a unique proposal variance for each component with intractable full conditional distribution is used, and it is tuned automatically on the fly with the goal of

obtaining a proposal acceptance rate of about 44%. This rate has been shown to be optimal for univariate normally distributed target and proposal distributions [Gelman et al., 1996; Roberts et al., 2001].

Algorithm 3 Draw Nb samples from $\pi(\boldsymbol{\theta}|\boldsymbol{y})$ using the adaptive random walk Metropolis-within-Gibbs algorithm [Roberts and Rosenthal, 2009]. $\pi(\theta_j|\boldsymbol{\theta}_{-j}, \boldsymbol{y})$ is not easy to directly sample for $j \leq k$ so it is approximated via MH. All operations involving j and g must be performed for all $j \in 1:k$ and $g \in k+1:d$, respectively.

1: Start from an initial sample $\boldsymbol{\theta}^{(0)}$, drawn from any density π_0, with the requirement $\pi_0(\boldsymbol{\theta}^{(0)}|\boldsymbol{y}) > 0$. Let $ls_j = 0$.
2: **for** $i = 1, 2, \cdots, N$ **do**
3: **for** $l = 1, 2, \cdots, b$ **do** ▷ batch loop
4: Draw a candidate value θ'_j from from $\mathcal{N}(\theta_j^{((i-1)b+(l-1))}, \exp(ls_j))$.
5: Set $\boldsymbol{\theta}' = \boldsymbol{\theta}^{((i-1)b+(l-1))}$ and $\theta'_j = \theta_j$.
6: Calculate the ratio $r_j = \pi(\boldsymbol{\theta}'|\boldsymbol{y})/\pi(\boldsymbol{\theta}^{((i-1)b+(l-1))}|\boldsymbol{y})$.
7: Set $\theta_j^{((i-1)b+l)} = \theta'_j$ with probability $\min(1, r_j)$ and update ratio $r_j = r_j + 1$, otherwise set $\theta_j^{((i-1)b+l)} = \theta_j^{((i-1)b+(l-1))}$. ▷ MH update
8: Draw $\theta_g^{((i-1)b+l)}$ from its full conditional posterior distribution, conditioning on the most recent updates to all other elements of $\boldsymbol{\theta}$. ▷ Gibbs update
9: **end for** ▷ end of l-loop
10: **if** $r_j < 0.44$ **then** $ls_j = ls_j + \delta(i)$ **else** $ls_j = ls_j - \delta(i)$. ▷ adaptation step
11: **end for** ▷ end of i-loop
12: **return** $\boldsymbol{\theta}^{(1)}, \cdots, \boldsymbol{\theta}^{(Nb)}$.

The algorithm (is summarized in)proceeds as follows: For each element j of the vector $\boldsymbol{\theta}$, θ_j, with posterior density not available in closed form, the proposal distribution is normal centered at its current value and variance ${\sigma_j}^2 \triangleq \exp(ls_j)$; the initial value of ls_j is zero. Then the algorithm runs for b times and during this process the variables with unknown posterior form are updated via MH, while the remaining variables with Gibbs. After each ith batch of b iterations, the acceptance ratio for each variable is computed. If the acceptance rate of element j is lower (or higher) than 0.44, then the term ls_j is decreased (increased) by $\delta(i)$. The term $\delta(i)$ equals $\min(0.01, i^{-1/2})$, and it converges to zero as i tends to infinity. ls_j is restricted within $[-M, M]$ for a constant $M < \infty$. Under these choices, Roberts and Rosenthal [2009] prove that the samples come from the target distribution and the algorithm is ergodic for bounded

functions.

Contrary to the AM, the correlations between the parameters are ignored, and the adaptation step is not performed at each iteration of the algorithm but only at the specific iterations $\{b, 2b, \cdots\}$, where b denotes the size of each batch. Algorithm 3 presents the adaptive random walk Metropolis-within-Gibbs with normal proposal. The parameters are arranged so that the first k variables are updated with the MH algorithm, as opposed to the rest variables that permit Gibbs updates.

2.3.2.3 Convergence of adaptive MCMC

The idea of adaptive MCMC algorithms is to adapt the proposal distribution from the sample history. One problem in adapting is that the chain is not Markovian and standard convergence results do not apply. One solution is to use adaptation only for a prespecified period, and then run a conventional non-adaptive algorithm with this scaling. These algorithms are called finite adaptive algorithms. However, we have to decide when to stop the adaption.

Ergodicity of adaptive MCMC has been studied by Andrieu et al. [2006], and their results have been extended by Roberts and Rosenthal [2007], Atchadé et al. [2010] among others. Roberts and Rosenthal [2007] prove that an adaptive MCMC algorithm is ergodic with respect to the target stationary distribution if it satisfies diminishing adaptation and bounded convergence. The diminishing adaptation condition intuitively means that we adapt less and less as time increases. The bounded convergence (containment) condition considers the time until near convergence. Besides, they prove a weak law of large numbers for bounded functions. For more details see Roberts and Rosenthal [2007], Roberts and Rosenthal [2009] and references therein.

2.3.3 Sequential Monte Carlo for time-invariant parameters

SMC methods sample sequentially a sequence of target probability densities through importance sampling and resampling steps [Doucet et al., 2001; Doucet and Johansen, 2009]. Initially, SMC has applied to the analysis of dynamic systems, and later they have been extended to time-invariant parameters; see Neal [2001]; Chopin [2002]; Ridgeway and Madigan [2003]; Del Moral et al. [2006]; Fearnhead et al. [2013]; Gunawan et al. [2018].

In contrast to MCMC methods, SMC are useful for online estimation problems, or in other words, if we want to infer unknown parameters of a statistical model sequentially after

Algorithm 4 Draw N samples from $\pi(\boldsymbol{\theta}|\boldsymbol{y})$ using IBIS algorithm [Chopin, 2002]. *Note*: The observations $\boldsymbol{y}$ are independent given $\boldsymbol{\theta}$. n_0 denotes a small number of the first observations, $\gamma \in (0, 1)$. All operations involving l must be performed for all $l \in 1 : N$.

Start from an initial sample $\boldsymbol{\theta}^{(l)}$ from $\pi_{n_0}(\boldsymbol{\theta})$, set $w_0^{(l)} = 1$ and calculate $\pi(y_{1:n_0}|\boldsymbol{\theta}^{(l)})$.
for $t = n_0 + 1, 2, \cdots, n$ **do**
    Calculate the importance weights $w_t^{(l)} = w_{t-1}^{(l)} \pi(y_t|\boldsymbol{\theta}^{(l)})$.
    Set $\pi(y_{1:t}|\boldsymbol{\theta}^{(l)}) = \pi(y_{1:t-1}|\boldsymbol{\theta}^{(l)})\pi(y_t|\boldsymbol{\theta}^{(l)})$.
    Obtain ESS $= (\sum_{l=1}^N w_t^{(l)})^2 / \sum_{l=1}^N (w_t^{(l)})^2$.
    if ESS$_t < \gamma N$ **then** ▷ resample - move step
        Evaluate $\hat{\boldsymbol{\mu}}_t = \sum_{l=1}^N w_t^{(l)} \boldsymbol{\theta}^{(l)} / \sum_{l=1}^N w_t^{(l)}$.
        Evaluate $\hat{\boldsymbol{\Sigma}}_t = \sum_{l=1}^N w_t^{(l)} (\boldsymbol{\theta}^{(l)} - \hat{\boldsymbol{\mu}}_t)(\boldsymbol{\theta}^{(l)} - \hat{\boldsymbol{\mu}}_t)^\top / \sum_{l=1}^N w_t^{(l)}$.
        Resample particles $\boldsymbol{\theta}^{(l)}$ with probabilities proportional to $w_t^{(l)}$, and set $w_t^{(l)} = 1$.
        Draw a candidate value $\boldsymbol{\theta}^{l'}$ from the proposal distribution $\mathcal{N}(\hat{\boldsymbol{\mu}}_t, \hat{\boldsymbol{\Sigma}}_t)$.
        Calculate the ratio $r_l = \pi(\boldsymbol{\theta}^{l'}|y_{1:t}) / \pi(\boldsymbol{\theta}^{(l)}|y_{1:t})$.
        Set $\boldsymbol{\theta}^{(l)} = \boldsymbol{\theta}^{l'}$ with probability $\min(1, r_l)$.
    end if
end for
return $\boldsymbol{\theta}^{(1)}, \cdots, \boldsymbol{\theta}^{(N)}$.

each new observation comes in. Each time a new observation is arrived the already sampled values (or particles) are reused and the posterior has not need to be recomputed.

2.3.3.1 Iterated batch importance sampling

We focus here on the iterated batch importance sampling (IBIS, Chopin [2002]) algorithm to estimate recursively in time the posterior distribution $\pi_t(\boldsymbol{\theta}) \triangleq \pi(\boldsymbol{\theta}|y_{1:t})$. The method is applied to data which are either conditionally independent given the parameter $\boldsymbol{\theta}$, i.e. $\pi(y_{1:t}|\boldsymbol{\theta}) = \prod_{i=1}^t \pi(y_i|\boldsymbol{\theta})$ or Markov, i.e. $\pi(y_{1:t}|\boldsymbol{\theta}) = \pi(y_1|\boldsymbol{\theta}) \prod_{i=2}^t \pi(y_i|y_{1:i-1}, \boldsymbol{\theta})$.

IBIS algorithm approximate $\pi_{t-1}(\boldsymbol{\theta})$ with particles $\{\boldsymbol{\theta}^{(l)}\}_{l=1}^N$, and associated probability weights $\{W_{t-1}^{(l)}\}_{l=1}^N$ which sum to one, constituting the discrete probability distribution $\hat{\pi}_{t-1}(\boldsymbol{\theta}) \triangleq \sum_{l=1}^N W_{t-1}^{(l)} \delta(\boldsymbol{\theta} - \boldsymbol{\theta}^{(l)})$. The aim of the tth iteration of the algorithm, is to sample from the target density $\pi_t(\boldsymbol{\theta})$ and construct a (discrete distribution) approximation via importance sampling and resampling techniques.

Since it is generally impossible to sample directly from $\pi_t(\boldsymbol{\theta})$, IBIS approximates it using an importance density $q_t(\boldsymbol{\theta}) = q(\boldsymbol{\theta})$ whose support includes that of the target. The weight

assigned to the lth sample, $\boldsymbol{\theta}^{(l)}$, is given by

$$w_t^{(l)} \triangleq \frac{\pi_t(\boldsymbol{\theta}^{(l)})}{q(\boldsymbol{\theta}^{(l)})} = \frac{\pi_t(\boldsymbol{\theta}^{(l)})}{\pi_{t-1}(\boldsymbol{\theta}^{(l)})} \times \frac{\pi_{t-1}(\boldsymbol{\theta}^{(l)})}{q(\boldsymbol{\theta}^{(l)})} = u_t(\boldsymbol{\theta}^{(l)}) w_{t-1}^{(l)}, \tag{2.9}$$

where $u_t(\boldsymbol{\theta}^{(l)})$ equals to $\pi(y_t|\boldsymbol{\theta}^{(l)})$ when the observations are independent, and to $\pi(y_t|y_{t-1}, \boldsymbol{\theta}^{(l)})$ when the observations are Markov of order one. This is repeated N times to produce the set of weighted particles at time t, $\{(\boldsymbol{\theta}^{(l)}, w_t^{(l)})\}_{l=1}^N$, which gives an importance sampling approximation to $\pi_t(\boldsymbol{\theta})$.

At each iteration of the algorithm if some degeneracy criterion is fulfilled, a resample-move step is applied. Firstly, each particle $\boldsymbol{\theta}^{(l)}$ is selected with replacement with a probability proportional to $w_t^{(l)}$, and the weights are reset to one; this statistical procedure is called resampling. The intuition behind this step is that weak particles with low importance weights (relative to $1/N$) will be discarded and replace them with stronger particles. Secondly, a single MH step is applied to each particle [Gilks and Berzuini, 2001]. It is not required a burn-in period, because the particles are approximately distributed according to $\pi_t(\boldsymbol{\theta})$ and the Markov kernel is $\pi_t(\boldsymbol{\theta})$-invariant. The intuition behind this step is that many of the resampled particles may have identical values. By applying the move step, the final sample represents a more diverse set of parameter values. New particles can be proposed according to a Gaussian random walk $\mathcal{N}(\boldsymbol{\theta}^{(l)}, c\hat{\boldsymbol{\Sigma}}_t)$, where c is a tuning constant, or an independent MH algorithm with proposal distribution $\mathcal{N}(\hat{\boldsymbol{\mu}}_t, \hat{\boldsymbol{\Sigma}}_t)$, where

$$\hat{\boldsymbol{\mu}}_t = \frac{\sum_{l=1}^N w_t^{(l)} \boldsymbol{\theta}^{(l)}}{\sum_{l=1}^N w_t^{(l)}}, \quad \hat{\boldsymbol{\Sigma}}_t = \frac{\sum_{l=1}^N w_t^{(l)} (\boldsymbol{\theta}^{(l)} - \hat{\boldsymbol{\mu}}_t)(\boldsymbol{\theta}^{(l)} - \hat{\boldsymbol{\mu}}_t)^\top}{\sum_{l=1}^N w_t^{(l)}},$$

and $\hat{\boldsymbol{\mu}}_t \triangleq \hat{\mathbb{E}}_{\pi_t(\boldsymbol{\theta})}, \hat{\boldsymbol{\Sigma}}_t \triangleq \widehat{\mathbb{V}}_{\pi_t(\boldsymbol{\theta})}$ [Chopin, 2002]; $\mathbb{V}$ denotes the variance of a random variable.

The decision of whether to apply a resample step is based on the effective sample size (ESS). It is theoretically defined as the equivalent number of independent samples generated directly form the target distribution, which yields the same efficiency in the estimation obtained with the N weighted draws. Kong et al. [1994] shows that it can be approximated as

$$\text{ESS}_t \triangleq \frac{(\sum_{l=1}^N w_t^{(l)})^2}{\sum_{l=1}^N (w_t^{(l)})^2}, \tag{2.10}$$

where ESS_t varies between 1 (all but one particle have weight zero) and N (all particles have equal weight). Thus, a low value indicates that the weights are concentrated only on a few

particles. During each iteration of the algorithm, if that $\text{ESS}_t < \gamma N$, e.g. $\gamma = 2/3$ or 1/3, then resampling is performed. Resampling is therefore performed at random times. For a review of resampling techniques see Douc and Cappé, 2005.

IBIS algorithm is usually initialized from the prior density. However, if the priors are flat the particle system may be degenerate in the first iterations. Thus, Chopin [2002] proposes to initialize the algorithm from the posterior $\pi_{n_0}(\boldsymbol{\theta})$, where n_0 denotes a small number of the first observations. IBIS with independent observations is presented in Algorithm 4. A consistent and asymptotically normal estimator of $\mathbb{E}_{\pi_t(\boldsymbol{\theta})}[f_0(\boldsymbol{\theta})]$ for an appropriately integrable function $f_0(\cdot)$, is given by $\sum_{l=1}^{N} w_t^{(l)} f_0(\boldsymbol{\theta}^{(l)}) / \sum_{l=1}^{N} w_t^{(l)}$; see Chopin et al. [2004].

2.4 State space models

State space models (SSMs) are a popular class of time series models that have been widely used in many different fields as statistics, econometrics and engineering; for more information see Doucet et al. [2001]; Campagnoli et al. [2009]; Durbin and Koopman [2012]; Zucchini et al. [2017]. In state space analysis, it is assumed that observations are associated with an underlying latent process which is not directly observed. The relationship between the two processes is specified by the state space model.

Let $\boldsymbol{h} \triangleq (h_1, \cdots, h_n)^\top$ be a (first order) Markov process taking values on some measurable space $\mathcal{H}$; it is a $n \times 1$ vector. This process is called latent (or hidden) since it is not directly observed, and it is fully specified by its initial and transition density which are given respectively by

$$\pi_{\boldsymbol{\theta}}(h_1) \triangleq \pi(h_1|\boldsymbol{\theta}), \quad \pi_{\boldsymbol{\theta}}(h_t|h_{t-1}) \triangleq \pi(h_t|h_{t-1}, \boldsymbol{\theta}), \tag{2.11}$$

where $\boldsymbol{\theta} \in \Theta$ denotes the (unknown) parameter of the model. Conditional on both the latent process and the model parameter, the observations y_t are independent, with distribution determined by the current state of the parameter process satisfying

$$\pi(y_t|y_{1:t-1}, h_{1:t}, \boldsymbol{\theta}) = \pi(y_t|h_t, \boldsymbol{\theta}) \triangleq \pi_{\boldsymbol{\theta}}(y_t|h_t). \tag{2.12}$$

Models compatible with (2.11)-(2.12) are known as SSMs. When the states are discrete they are usually referred to as hidden Markov models. Equation (2.11) is called the state equation, which describes how the system evolves from one time point to the next, while (2.12) is the

observation equation, which describes how the underlying state is transformed into something that is measured directly.

A key ingredient for classical and Bayesian inference is the likelihood function of $\boldsymbol{\theta}$, $\pi_{\boldsymbol{\theta}}(y_{1:t})$, which satisfies

$$\pi_{\boldsymbol{\theta}}(y_{1:t}) \triangleq \int_{\mathcal{H}} \pi_{\boldsymbol{\theta}}(y_{1:t}, \boldsymbol{h})\, d\boldsymbol{h}, \tag{2.13}$$

where $\pi_{\boldsymbol{\theta}}(y_{1:t}, \boldsymbol{h})$ denotes the joint density of $(y_{1:t}, \boldsymbol{h})$ which is given from equations (2.11)-(2.12) by

$$\pi_{\boldsymbol{\theta}}(\boldsymbol{h}, y_{1:t}) = \pi_{\boldsymbol{\theta}}(h_1) \prod_{t=2}^{n} \pi_{\boldsymbol{\theta}}(h_t|h_{t-1}) \prod_{t=1}^{n} \pi_{\boldsymbol{\theta}}(y_t|h_t).$$

The computation of the likelihood is not an easy task since it requires the evaluation of complex high dimensional integrals. An exception is the linear Gaussian models (the latent state and the observed variables, have Gaussian distributions with a linear dependency between variables), for which it can be calculated by using Kalman techniques [Kalman, 1960]. In the case of nonlinear or non-Gaussian state space models, the Kalman filter cannot be used. Popular approximation methods are the extended Kalman filter [Jazwinski, 2007] and the unscented Kalman filter [Julier and Uhlmann, 2004]; see Julier et al. 1995; Van Der Merwe et al. 2001. However, depending on the degree of non-linearity and non-Gaussianity these alternatives may perform poorly.

2.4.1 The AR(1) process

A popular case of SSMs is when the latent state is an AR(1) process, that is

$$h_t = \phi_0 + \phi_1(h_{t-1} - \phi_0) + \varepsilon_t, \quad \varepsilon_t \overset{i.i.d.}{\sim} \mathcal{N}(0, \sigma^2), \tag{2.14}$$

where the symbol $\sim$ means 'is distributed according to'. Conditional on h_{t-1}, h_t follows a Gaussian distribution with mean and variance given by

$$\mathbb{E}(h_t|h_{t-1}) = \phi_0 + \phi_1(h_{t-1} - \phi_0), \quad \mathbb{V}(h_t|h_{t-1}) = \sigma^2.$$

In general, a process is called weakly stationary if its mean is constant and its lag k autocovariance depends only on the lag. The AR(1) latent process is weakly stationary when $|\phi_1| < 1$ and it can be written as a weighted sum of past error terms (see Tsay 2005, Section

2.4.1)

$$h_t = \phi_0 + \sum_{t=0}^{\infty} \phi_1^i \varepsilon_{t-i}, \quad t \geq 2.$$

$$\mathbb{E}(h_t) = \phi_0, \quad \text{cov}(h_t, h_{t+k}) = \frac{\sigma^2}{1-\phi_1^2}\phi_1^k.$$

When $\phi_1 = 0$, the process is called white noise (WN), while when $\phi_1 = 1$ it is called first-order random walk (RW(1)).

2.5 Bayesian inference for state space models

The joint probability density function (pdf) of the latent state, $\pi_{\boldsymbol{\theta}}(\boldsymbol{h})$, is assumed to be $\mathcal{N}_n(\boldsymbol{m}, \boldsymbol{C})$ at $\boldsymbol{h}$. Besides, $\pi(\boldsymbol{\theta})$ denotes a suitable prior for the model parameter. Instead of simulating directly from the target distribution, this can be achieved by constructing a Markov chain which mimics the two-component Gibbs sampler sampling iteratively from $\pi(\boldsymbol{h}|\boldsymbol{\theta}, \boldsymbol{y})$ and $\pi(\boldsymbol{\theta}|\boldsymbol{h}, \boldsymbol{y})$; see Sections 2.5.1 and 2.5.2, respectively.

2.5.1 Updating the state

In this section we sample from $\pi(\boldsymbol{h}|\boldsymbol{\theta}, \boldsymbol{y}) \overset{(2.3)}{\propto} \pi_{\boldsymbol{\theta}}(\boldsymbol{y}|\boldsymbol{h})\pi_{\boldsymbol{\theta}}(\boldsymbol{h})$, where $\pi_{\boldsymbol{\theta}}(\boldsymbol{y}|\boldsymbol{h}) \triangleq \pi(\boldsymbol{y}|\boldsymbol{h}, \boldsymbol{\theta})$.

Updating the state with MCMC A detailed review on MCMC in the context of SSMs can be found in Fearnhead [2011]. Firstly, we sample the components of the latent state one at time in separate MH steps conditional on all others values of the state process and on the parameter vector by employing the random walk Metropolis algorithm.The full conditional posterior distribution of the tth element of the vector $\boldsymbol{h}$ is given by

$$\pi(h_t|\boldsymbol{h}_{-t}, \boldsymbol{\theta}, y_{1:t}) \propto \pi_{\boldsymbol{\theta}}(y_t|h_t)\pi_{\boldsymbol{\theta}}(h_t|h_{-t}),$$

where

$$\pi_{\boldsymbol{\theta}}(h_t|h_{-t}) \triangleq \begin{cases} \pi_{\boldsymbol{\theta}}(h_1)\pi_{\boldsymbol{\theta}}(h_2|h_1), & t = 1 \\ \pi_{\boldsymbol{\theta}}(h_t|h_{t-1})\pi_{\boldsymbol{\theta}}(h_{t+1}|h_t), & t = 2, \cdots, n-1 \\ \pi_{\boldsymbol{\theta}}(h_n|h_{n-1}), & t = n. \end{cases}$$

If some target distribution is known in closed form, the sampling may be done with the Gibbs sampler.

The previous method may lead to slow mixing if there is strong temporal dependence in the state process, while increasing the number of states, the run time increases too. Titsias [2011]; Titsias and Papaspiliopoulos [2018] propose the auxiliary sampler based on $\boldsymbol{z}$ (aGrad-z) algorithm which updates the whole process in one move. It belongs to a new family of MCMC samplers that combine auxiliary variables and approximate the intractable likelihood function. Let $\boldsymbol{z}$ ba an auxiliary vector of variables with conditional pdf given $\boldsymbol{h}$, $\pi(\boldsymbol{z}|\boldsymbol{h})$, the $\mathcal{N}_n\left(\boldsymbol{h} + (\delta/2)\nabla_{\boldsymbol{h}} f(\boldsymbol{h}), (\delta/2)\mathbb{I}_n\right)$ at $\boldsymbol{z}$, where $f(\boldsymbol{h}) \triangleq \log \pi_{\boldsymbol{\theta}}(\boldsymbol{y}|\boldsymbol{h})$, $\nabla_{\boldsymbol{h}} f(\boldsymbol{h})$ denotes partial derivative of $f(\boldsymbol{h})$ with respect to $\boldsymbol{h}$ such as $\nabla_{\boldsymbol{h}} f(\boldsymbol{h}) = \boldsymbol{h} - \exp(\boldsymbol{h})/(1+\exp(\boldsymbol{h}))$ and $\delta > 0$ is a parameter that has to be tuned. The auxiliary variable $\boldsymbol{z}$ tends to move the chain in the direction of $\nabla_{\boldsymbol{h}} f(\boldsymbol{h})$. The modified target density is given by

$$\pi(\boldsymbol{h}, \boldsymbol{z}|\boldsymbol{\theta}, \boldsymbol{y}) \propto \pi_{\boldsymbol{\theta}}(\boldsymbol{y}|\boldsymbol{h})\pi(\boldsymbol{h}|\boldsymbol{\theta})\pi(\boldsymbol{z}|\boldsymbol{h}).$$

This augmentation is consistent since by marginalizing out $\boldsymbol{z}$ it is recovered the initial target density. Thus, one can sample to produce one draw $(\boldsymbol{h}^{(l)}, \boldsymbol{z}^{(l)})$ from the modified target distribution, and by construction the sample $\boldsymbol{h}^{(l)}$ is from the desired density.

Algorithm 5 presents the steps to sample $\boldsymbol{h}^{(l)}$ from $\pi(\boldsymbol{h}|\boldsymbol{\theta}, \boldsymbol{y})$, given that the current value of the chain is $(\boldsymbol{h}^{(l-1)}, \boldsymbol{\theta}^{(l-1)})$. In step 2, the proposal density is the product of the pdf $\mathcal{N}_n(\boldsymbol{m}, \boldsymbol{C})$ at $\boldsymbol{h}'$ and the pdf $\mathcal{N}_n(\boldsymbol{h}', (\delta/2)\mathbb{I}_n)$ at $\boldsymbol{z}$. Hence, the proposal distribution is invariant to the Gaussian prior, and defines a non-independent MH algorithm. The parameter δ is tuned to achieve an acceptance rate between 50% and 60% which empirically the authors have observed to maximize sampling efficiency (the efficiency is measured as ESS per unit of computing time). Titsias and Papaspiliopoulos [2018] compare their proposed algorithms to other methods such as preconditioned Crank-Nicolson Langevin (pCNL); see Beskos et al. [2008] and Cotter et al. [2013] and conclude that the new samplers are more efficient.

Updating the state with SMC In this section the parameter $\boldsymbol{\theta}$ is assumed known and we want to recursively calculate the filtering densities $\pi_{\boldsymbol{\theta}}(h_t|y_{1:t})$ by employing the APF [Pitt and Shephard, 1999, 2001].

Algorithm 5 Sample $\boldsymbol{h}^{(l)}$ of $\pi(\boldsymbol{h}|\boldsymbol{\theta}, \boldsymbol{y})$, given that the current value of the chain is $(\boldsymbol{h}^{(l-1)}, \boldsymbol{\theta}^{(l-1)})$, using the aGrad-z sampler [Titsias, 2011; Titsias and Papaspiliopoulos, 2018].

1: Draw an auxiliary variable $\boldsymbol{z}$ from $\mathcal{N}_n(\boldsymbol{h}^{(l-1)} + (\delta/2)\nabla_{\boldsymbol{h}^{(l-1)}} f(\boldsymbol{h}^{(l-1)}), (\delta/2)\mathbf{I}_n)$.
2: Draw a candidate value $\boldsymbol{h}'$ from the proposal $\mathcal{N}_n((2/\delta)\boldsymbol{\Sigma}\left(\boldsymbol{z} + (\delta/2)\boldsymbol{C}^{-1}\boldsymbol{m}\right), \boldsymbol{\Sigma})$, where $\boldsymbol{\Sigma} \triangleq \left(\boldsymbol{C}^{-1} + (2/\delta)\mathbb{I}_n\right)^{-1}$.
3: Calculate the acceptance ratio $r = \exp\{f(\boldsymbol{h}') - f(\boldsymbol{h}^{(l-1)}) + g(\boldsymbol{z}, \boldsymbol{h}') - g(\boldsymbol{z}, \boldsymbol{h}^{(l-1)})\}$, where $g(\boldsymbol{z}, \boldsymbol{h}) = (\boldsymbol{z} - \boldsymbol{h} - (\delta/4)\nabla_{\boldsymbol{h}} f(\boldsymbol{h}))^\top \nabla_{\boldsymbol{h}} f(\boldsymbol{h})$.
4: Set $\boldsymbol{h}^{(l)} = \boldsymbol{h}'$ with probability $\min(1, r)$, otherwise set $\boldsymbol{h}^{(l)} = \boldsymbol{h}^{(l-1)}$.

The filtered density, $\pi_{\boldsymbol{\theta}}(h_t|y_{1:t})$, satisfies the following recursion

$$\pi_{\boldsymbol{\theta}}(h_t|y_{1:t}) = \frac{\pi_{\boldsymbol{\theta}}(y_t|h_t)\pi_{\boldsymbol{\theta}}(h_t|y_{1:t-1})}{\pi_{\boldsymbol{\theta}}(y_t|y_{1:t-1})}, \tag{2.15}$$

where $\pi_{\boldsymbol{\theta}}(h_t|y_{1:t-1}) = \int_{\mathcal{H}} \pi_{\boldsymbol{\theta}}(h_t|h_{t-1})\pi_{\boldsymbol{\theta}}(h_{t-1}|y_{1:t-1})\, dh_{t-1}$. Particle filters (SMC methods applied to SSMs) approximate $\pi_{\boldsymbol{\theta}}(h_{t-1}|y_{1:t-1})$ (which may be continuous) with a set of N particles, $\{h_{t-1}^{(l)}\}_{l=1}^N$, and associated probability weight $\{W_{t-1}^{(l)}\}_{l=1}^N$ which sum to one, constituting a discrete density $\hat{\pi}_{\boldsymbol{\theta}}(h_{t-1}|y_{1:t-1}) \triangleq \sum_{l=1}^N W_{t-1}^{(l)} \delta(h_{t-1} - h_{t-1}^{(l)})$. Substituting $\hat{\pi}_{\boldsymbol{\theta}}(h_{t-1}|y_{1:t-1})$ for $\pi_{\boldsymbol{\theta}}(h_t|y_{1:t-1})$, yields a (continuous density) approximation to $\pi_{\boldsymbol{\theta}}(h_t|y_{1:t})$ given by

$$\tilde{\pi}_{\boldsymbol{\theta}}(h_t|y_{1:t}) \propto \sum_{l=1}^N W_{t-1}^{(l)} \pi_{\boldsymbol{\theta}}(y_t|h_t)\pi_{\boldsymbol{\theta}}(h_t|h_{t-1}^{(l)}). \tag{2.16}$$

The aim of the tth iteration of the particle filter, is to construct a new (discrete distribution) approximation to $\tilde{\pi}_{\boldsymbol{\theta}}(h_t|y_{1:t})$ via importance sampling and resampling techniques.

Let J be an auxiliary random variable taking values on the set of integers $\{1, \cdots, N\}$, which can be thought of as a variable selecting one of the components from the mixture in (2.16). The modified target density is given by

$$\pi_{\boldsymbol{\theta}}(j, h_t|y_{1:t}) \propto W_{t-1}^j \pi_{\boldsymbol{\theta}}(y_t|h_t)\pi_{\boldsymbol{\theta}}(h_t|h_{t-1}^j).$$

This augmentation is consistent since by marginalizing out j it is recovered the initial target density. The proposal distribution is chosen as a factorized form

$$q(j, h_t|y_{1:t}) \propto W_{t-1}^j \nu(h_{t-1}^j, y_t) q(h_t|h_{t-1}^j, y_t). \tag{2.17}$$

Drawing one realization from the proposal distribution 2.17 is equivalent to (a) draw the particle

h_{t-1}^j where j is selected from $\{1, \cdots, N\}$ with probability proportional to $W_{t-1}^j \nu(h_{t-1}^j, y_t)$; (b) draw $h_t \sim q(h_t|h_{t-1}^j, y_t)$. The weight assigned to the sample (j, h_t) is given by

$$w_t \propto \frac{\pi_{\boldsymbol{\theta}}(y_t|h_t)\pi_{\boldsymbol{\theta}}(h_t|h_{t-1}^j)}{\nu(h_{t-1}^j, y_t)q(h_t|h_{t-1}^j, y_t)}. \tag{2.18}$$

This is repeated N times to produce the set of weighted particles at time t, $\{(h_t^{(l)}, w_t^{(l)})\}_{l=1}^N$, which gives an importance sampling approximation to $\pi_{\boldsymbol{\theta}}(h_t|y_{1:t})$.

The key idea of APF is to increase the probability of resampling particles at time $t-1$ that are in agreement with the observation y_t using an algorithm with computational complexity of $\mathcal{O}(N)$ per time step. In other words, $\nu(h_{t-1}, y_t)$ should take a large value if it is likely to observe the observation y_t at time t, given the system state is h_{t-1} at time t. This should produce more even weights and thus a better approximation of the filtering density. Pitt and Shephard [1999] recommend selecting $\nu(h_{t-1}, y_t) = \pi_{\boldsymbol{\theta}}(y_t|h_{t-1})$, which is called the predictive likelihood. If it is not tractable, the authors suggest to be approximated by $\pi_{\boldsymbol{\theta}}(y_t|\mu_t)$ where μ_t is a value (e.g. mean, mode, or sample value) associated with the distribution of $h_t|h_{t-1}$.

Algorithm 6 Draw $h_t^{(l)}$ from $\pi(h_t|y_{1:t}, \boldsymbol{\theta})$ given the existing particle system $(h_{t-1}^{(l)}, w_{t-1}^{(l)})_{l=1}^N$ using the APF [Pitt and Shephard, 1999, 2001]. All operations involving l must be performed for all $l \in 1:N$.

1: Calculate $\lambda^{(l)} \propto w_{t-1}^{(l)} \pi_{\boldsymbol{\theta}}(y_t|\mu_t^{(l)})$, where $\mu_t^{(l)}$ denotes the mean associated with the distribution of $h_t|h_{t-1}^{(l)}$.

2: Sample the indices $j^{(l)}$ from $\{1, \cdots, N\}$ with probabilities proportional to $\{\lambda^1, \cdots, \lambda^N\}$, and set $\bar{h}_{t-1}^{(l)} \triangleq h_{t-1}^{j^{(l)}}$, and $\bar{\mu}_t^{(l)} \triangleq \mu_t^{j^{(l)}}$. ▷ resample step

3: Propagate particles $h_t^{(l)} \sim \pi_{\boldsymbol{\theta}}(h_t|\bar{h}_{t-1}^{(l)})$.

4: Calculate the weights $w_t^{(l)} = \pi_{\boldsymbol{\theta}}(y_t|h_t^{(l)})/\pi_{\boldsymbol{\theta}}(y_t|\bar{\mu}_t^{(l)})$.

5: Resample the values $h_t^{(l)}$ with replacement from $\{h_t^{(1)}, \cdots, h_t^{(N)}\}$ with probabilities proportional to $\{w_t^{(1)}, \cdots, w_t^{(N)}\}$, and set $w_t^{(l)} = 1$.

6: **return** $h_n^{(1)}, \cdots, h_n^{(N)}$.

The variance of the importance weights is affected by the choice of proposal distributions in the importance sampling stage. By choosing a proposal which closely resembles the target, the importance weights will be approximately equal, and thus the variance of the weights will be close to zero; for more details see Doucet et al. [2000]. Drawing $h_t^{(l)}$ from $\pi(h_t|y_{1:t}, \boldsymbol{\theta})$ given

the existing particle system $\{(h_{t-1}^{(l)}, w_{t-1}^{(l)})\}_{l=1}^{N}$ is summarized in Algorithm 6.

2.5.2 Updating the parameters

We want to update the posterior conditional density of $\boldsymbol{\theta}$, $\pi(\boldsymbol{\theta}|\boldsymbol{h}, \boldsymbol{y})$. Often, it is useful to update the parameter vector in blocks, or its individual components one by one sequentially in time, or updating the whole parameter in one step. However, when there is strong dependence between $\boldsymbol{\theta}$ and $\boldsymbol{h}$, the overall efficiency of the MCMC algorithm may be poor. This problem can often be improved through model parametrizations.

Papaspiliopoulos [2003]; Papaspiliopoulos et al. [2003] present two parametrizations that can be used for SSMs: centered parameterization (CP) and non-centered parameterization (NCP). This terminology was first introduced by Gelfand et al. [1995], who discussed parametrization for normal linear mixed model, and was generalized by Papaspiliopoulos [2003] in the context of hierarchical models. More specifically, the CP is defined as the parametrization under which the data $\boldsymbol{y}$ are independent of the parameters $\boldsymbol{\theta}$ conditionally on the state $\boldsymbol{h}$. More specifically, the CP is defined by a model where $\pi(\boldsymbol{\theta}|\boldsymbol{h}, \boldsymbol{y}) = \pi(\boldsymbol{\theta}|\boldsymbol{h})$. Thus, the posterior density of $\boldsymbol{\theta}$ is independent of the observations. Since, in the general state space models the likelihood has an intractable form, the centered parametrization implies that it is likely to update the component $\boldsymbol{\theta}$ via Gibbs sampling. On the other hand, the NCP is based on the assumption that there is an alternative parametrization $(\boldsymbol{h}, \boldsymbol{\theta}) \rightarrow (\tilde{\boldsymbol{h}}, \boldsymbol{\theta})$ such that: $\tilde{\boldsymbol{h}}$ is a-prior independent of $\boldsymbol{\theta}$ and there is a function $\psi(\cdot)$ where $\boldsymbol{h} = \psi(\tilde{\boldsymbol{h}}, \boldsymbol{\theta})$. Under this parametrization the posterior of $\boldsymbol{\theta}$ is not independent of the data, which makes updating $\boldsymbol{\theta}$ computationally more challenging in general.

These parametrizations are complementary, in the sense that the one is likely to perform well when the other do not and conversely [Papaspiliopoulos, 2003]. Further, Papaspiliopoulos et al. [2003] provide insights into the conditions under which centered and non-centered parametrizations outperform each other. Similar to Gelfand et al. [1995], they conclude that CP is better when the likelihood is informative and NCP is better when the likelihood is relatively uninformative. For spatial models, they observe that CP outperforms NCP in MCMC as the correlation of the spatial process increases. These ideas have been used extensively within continuous time stochastic volatility models; see, for instance, Shephard and Kim [1994], Kim et al. [1998], Frühwirth-Schnatter [2004].

Yu and Meng [2011] present ancillary-sufficiency interweaving strategy (ASIS) which is a method of combining both parametrizations by interweaving (and not alternating) the two strategies at each iteration in order to improve MCMC mixing. The proposed algorithm converges geometrically even when CP and NCP fail to do so; for more details see the original paper. In a nutshell, the ASIS works as follows: At the lth iteration of the algorithm, let the current value of the chain be $\{\boldsymbol{\theta}^{(l-1)},\ \boldsymbol{h}^{(l-1)},\ \tilde{\boldsymbol{h}}^{(l-1)}\}$, where $\boldsymbol{h}^{(l-1)} = \psi(\tilde{\boldsymbol{h}}^{(l-1)}, \boldsymbol{\theta}^{(l-1)})$. The interweaving says that given the model parameter $\boldsymbol{\theta}^{(l-1)}$, sample the latent state $\boldsymbol{h}^{(l)}$ ($\tilde{\boldsymbol{h}}^{(l)}$) utilizing CP (NCP). Then, given the updated latent vector sample $\boldsymbol{\theta}^{(l)}$ in CP (NCP). Next, move to NCP (CP) and redraw $\boldsymbol{\theta}^{(l)}$ in NCP (CP). At the end, move back to CP (NCP) by the transformation $\boldsymbol{h}^{(l)} = \psi(\tilde{\boldsymbol{h}}^{(l)}, \boldsymbol{\theta}^{(l)})$ ($\tilde{\boldsymbol{h}}^{(l)} = \psi^{-1}(\boldsymbol{h}^{(l)}, \boldsymbol{\theta}^{(l)})$). Thus, the parameters are sampled twice, while the latent state only once; either in CP or NCP. The model parameters may be updated in blocks of its components, or the whole vector in a single move. Following Kastner and Frühwirth-Schnatter [2014] if the state is updated in CP it is termed GIS-C, otherwise it is termed GIS-NC, where the terminology GIS stands for global interweaving strategy and was introduced by Yu and Meng [2011]. Algorithm 7 presents the GIS-C algorithm where the latent states are updated in CP.

Algorithm 7 Draw N samples from $\pi(\boldsymbol{h}, \boldsymbol{\theta}|\boldsymbol{y})$ using the GIS-C algorithm [Yu and Meng, 2011].

1: Start from an initial sample $\{\boldsymbol{\theta}^{(0)}, \boldsymbol{h}^{0}\}$, drawn from any density π_0, with the requirement $\pi_0(\boldsymbol{\theta}^{(0)}, \boldsymbol{h}^{0}|\boldsymbol{y}) > 0$.
2: **for** $l = 1, 2, \cdots, N$ **do**
3: Draw $\boldsymbol{h}^{(l)}$ in CP from $\pi(\boldsymbol{h}|\boldsymbol{\theta}^{(l-1)}, \boldsymbol{y})$.
4: Draw $\boldsymbol{\theta}^{(l)}$ in CP from $\pi(\boldsymbol{\theta}|\boldsymbol{h}^{(l)}, \boldsymbol{y})$.
5: Move to NCP by calculating $\tilde{\boldsymbol{h}}^{(l)} = h^{-1}(\boldsymbol{h}^{(l)}, \boldsymbol{\theta}^{(i)})$.
6: Redraw $\boldsymbol{\theta}^{(l)}$ in NCP $\pi(\tilde{\boldsymbol{h}}^{(l)}|\boldsymbol{\theta}^{(l)}, \boldsymbol{y})$.
7: Move to CP by calculating $\boldsymbol{h}^{(l)} = \psi(\tilde{\boldsymbol{h}}^{(l)}, \boldsymbol{\theta}^{(l)})$.
8: **end for**
9: **return** $(\boldsymbol{\theta}^{(1)}, \boldsymbol{h}^{(1)}), \cdots, (\boldsymbol{\theta}^{(N)}, \boldsymbol{h}^{(N)})$.

Chapter 3

Bayesian inference of the AD model

3.1 Introduction

In this chapter, we explore Bayesian inference via MCMC and SMC for the ADS model to analyze high-frequency integer price changes. Models that are similar in spirit to the ADS, but not necessarily applied on tick data, have been studied among others, by Anatolyev and Gospodinov [2010] and Kauppi and Saikkonen [2008] in the context of forecasting returns and recessions, respectively. In the former paper, the return is expressed into a sign component and absolute value component which are modeled separately as a copula before the joint forecast is constructed. The authors use a logistic link function and no lagged logit term. In the latter paper, the authors construct four different model specifications, including the static, dynamic, autoregressive, and dynamic autoregressive probit models and find that dynamic models have better predictive power than static models. The last paper is used by Nyberg [2011]; Pönkä [2017] to investigate the sign predictability of U.S. stock returns. The models discussed in this paragraph are estimated by maximum likelihood based on the Bernoulli density.

This chapter is organized as follows. Section 3.2 presents the AD model and Section 3.3 analyses the proposed model within a Bayesian framework. Section 3.4 presents the results of a simulation study. Section 3.5 presents the empirical results.

3.2 The AD model for price movements

Let P_{t_i} be the transaction price at time t_i and $Y_i \triangleq P_{t_i} - P_{t_{i-1}}$ be the price difference of the ith transaction which is an integer multiple of a fixed tick. The interest lies in estimating the condi-

tional distribution of the discrete price changes, $Y_i|\mathcal{F}_{i-1}$, where $\mathcal{F}_i$ denotes the information set available at the time transaction i takes place generated by past values of the observed series, as well as past values of a covariate vector.

The price decomposition model of Rydberg and Shephard [1998a, 2003] specifies the price movement as a product of a binary (takes only two values) variable A_i on $\{0, 1\}$ defining the market activity (the price moves or not), a binary variable D_i on $\{-1, 1\}$ defining a negative or positive price move (if a change occurs) and the size of a price change in ticks on the strictly positive integers. In our dataset, most of the transactions prices are the same as their previous values (78.94%), one tick down and up occur with 10.53% and 10.51% frequency, respectively, and movements of more than one tick occur about 0.01% of the time. Since only a few transactions result in a price change of more than one tick we convert all price movements in our data to indicators 1, 0 and –1, corresponding to price increases, no change, and price decreases respectively. For this reason Y_i can be written as

$$Y_i = A_i D_i, \tag{3.1}$$

where $D_i = 0$ if $A_i = 0$. The conditional joint distribution of (A_i, D_i) is decomposed as

$$\pi(A_i, D_i|\mathcal{F}_{i-1}) = \pi(A_i = 1|\mathcal{F}_{i-1})\pi(D_i|A_i = 1, \mathcal{F}_{i-1}). \tag{3.2}$$

3.2.1 Models for the activity and direction process

Activity is a bivariate variable, denoted as A_i, that takes the values of 0 or 1 to indicate whether there is a price change in the ith trade. Conditional on the information set available at time t_i, $\mathcal{F}^{\mathrm{A}}_{i-1}$, we assume that A_i has a Bernoulli distribution with probability $\pi_{A_i} \triangleq \pi(A_i = 1|\mathcal{F}^{\mathrm{A}}_{i-1})$ or, in symbols,

$$A_i|\mathcal{F}^{\mathrm{A}}_{i-1} \sim \text{Bernoulli}(\pi_{A_i}).$$

Following Rydberg and Shephard [1998a, 2003], we parametrize π_{A_i} as a GLARMA binary model of the form

$$\text{logit}(\pi_{A_i}) = \boldsymbol{x}^{\mathrm{A},\top}_{i-1}\boldsymbol{\beta}_{\mathrm{A}} + Z^{\mathrm{A}}_i \tag{3.3}$$

$$Z^{\mathrm{A}}_i = \phi_{\mathrm{A}} Z^{\mathrm{A}}_{i-1} + \delta_{\mathrm{A}}\varepsilon^{\mathrm{A}}_{i-1}, \tag{3.4}$$

where $\text{logit}(\pi_{A_i}) \triangleq \log(\pi_{A_i}) - \log(1-\pi_{A_i})$, $\varepsilon_i^{\text{A}} \triangleq (A_i - \pi_{A_i})/\sqrt{\pi_{A_i}(1-\pi_{A_i})}$, $\boldsymbol{x}_{i-1}^{\text{A}}$ denotes the $d_{\text{A}} \times 1$ dimensional vector of explanatory variables (such as volume or time of transaction) for the activity component known at t_{i-1} in which the first element is always one and $\boldsymbol{\beta}_{\text{A}}$ is a $d_{\text{A}} \times 1$ parameter vector. Moreover, $|\phi_{\text{A}}| < 1$ and $\delta_{\text{A}} > 0$.

Consequently,

$$\pi_{A_i} = \frac{\exp(\text{logit}(\pi_{A_i}))}{1+\exp(\text{logit}(\pi_{A_i}))} \text{ and } 1-\pi_{A_i} = \frac{1}{1+\exp(\text{logit}(\pi_{A_i}))}.$$

The conditional mean of A_i is $\mathbb{E}(A_i|\mathcal{F}_{i-1}^{\text{A}}) = \pi_{A_i}$ and the conditional variance of the price activity can be defined as $\mathbb{V}(A_i|\mathcal{F}_{i-1}^{\text{A}}) = \pi_{A_i}(1-\pi_{A_i})$. Furthermore, the term ε_i form a Martingale difference sequence with unit conditional and unconditional variance characterizing the new information associated with the ith transaction. The log-likelihood function is given by

$$\begin{aligned}\ell(\boldsymbol{\theta}_{\text{A}}) &\triangleq \sum_{i\in 1:n} \log \pi(A_i = \alpha_i|\mathcal{F}_{i-1}^{\text{A}}, \boldsymbol{\theta}_{\text{A}}) \\ &= \sum_{i\in 1:n} \left(\alpha_i(\boldsymbol{x}_{i-1}^{\top}\boldsymbol{\beta}_{\text{A}} + Z_i^{\text{A}}) - \log(1+\exp(\boldsymbol{x}_{i-1}^{\text{A},\top}\boldsymbol{\beta}_{\text{A}} + Z_i^{\text{A}})\right), \end{aligned} \tag{3.5}$$

where $\alpha_i \in (0,1)$, $\boldsymbol{\theta}_{\text{A}} = (\phi_{\text{A}}, \delta_{\text{A}}, \boldsymbol{\beta}_{\text{A}})^{\top}$ collects the parameters of the price activity process.

The direction component, conditional on the activity component, is a bivariate variable, denoted as D_i, that takes the values of -1 or 1 to indicate a negative or positive price move of the ith trade. We re-parametrize D_i as $D_i' = 1$ if $D_i = 1$, and $D_i' = 0$ if $D_i = -1$. In a similar way, we assign a Bernoulli distribution D_i' with probability $\pi_{D_i'} \triangleq \pi(D_i' = 1|A_i = 1, \mathcal{F}_{i-1}^{\text{D}}) = \pi(D_i = 1|A_i = 1, \mathcal{F}_{i-1}^{\text{D}}) \triangleq \pi_{D_i}$, conditional on the past information $\mathcal{F}_{i-1}^{\text{D}}$ and $A_i = 1$ or, in symbols,

$$D_i'|A_i = 1, \mathcal{F}_{i-1}^{\text{D}} \sim \text{Bernoulli}(\pi_{D_i}).$$

We parametrize π_{D_i} as an autologistic model of the form

$$\text{logit}(\pi_{D_i}) = \boldsymbol{x}_{i-1}^{\text{D},\top}\boldsymbol{\beta}_{\text{D}} + \zeta_1 D_{i-1}' + \zeta_2 D_{i-2}' \tag{3.6}$$

and a GLARMA binary model similarly to the activity factor; $\boldsymbol{x}_{i-1}^{\text{D}}$ denotes the $d_{\text{D}} \times 1$ dimensional vector of explanatory variables or the direction component known at t_{i-1} in which the first element is always one and $\boldsymbol{\beta}_{\text{D}}$ is a $d_{\text{D}} \times 1$ parameter vector. The log-likelihood function is

given by

$$\begin{aligned}\ell(\boldsymbol{\theta}_{\mathrm{D}}) &\triangleq \sum_{\substack{i\in 1:n\\ \forall i:A_i=1}} \log \pi(D_i = d_i' | A_i = 1, \mathcal{F}_{i-1}^{\mathrm{D}}, \boldsymbol{\theta}_{\mathrm{D}}) \\ &= \sum_{\substack{i\in 1:n\\ \forall i:A_i=1}} \left(d_i'(\boldsymbol{x}_{i-1}^{\mathrm{D},\top}\boldsymbol{\beta}_{\mathrm{D}} + Z_i^{\mathrm{D}}) - \log(1+\exp(\boldsymbol{x}_{i-1}^{\mathrm{D},\top}\boldsymbol{\beta}_{\mathrm{D}} + Z_i^{\mathrm{D}})\right), \end{aligned} \tag{3.7}$$

for the GLARMA model, replacing Z_i by $\zeta_1 D_{i-1}' + \zeta_2 D_{i-2}'$ in (3.7) for the autologistic model. Moreover, $d_i' \in (0,1)$, $\boldsymbol{\theta}_{\mathrm{D}} = (\phi_{\mathrm{D}}, \delta_{\mathrm{D}}, \boldsymbol{\beta}_{\mathrm{D}})^\top$ ($\boldsymbol{\theta}_{\mathrm{D}} = (\zeta_1, \zeta_2, \boldsymbol{\beta}_{\mathrm{D}})^\top$) collects the parameters of the direction process.

The next transaction may fall into one of the following categories: no price change with probability $1-\pi_{A_i}$, a price increase with probability $\pi_{A_i}\pi_{D_i}$ or a price decrease with probability $\pi_{A_i}(1-\pi_{D_i})$. The overall log-likelihood of the AD model is given by

$$\ell(\boldsymbol{\theta}_{\mathrm{A}}, \boldsymbol{\theta}_{\mathrm{D}}) \triangleq \log \pi(\boldsymbol{y}|\mathcal{F}_0) = \ell(\boldsymbol{\theta}_{\mathrm{A}}) + \ell(\boldsymbol{\theta}_{\mathrm{D}}). \tag{3.8}$$

Rydberg and Shephard [1998a, 2003] estimate the unknown parameters with maximum likelihood using the Berndt–Hall–Hall–Hausman (BHHH, Berndt et al. 1974) algorithm with analytic first derivatives. The authors include higher number of past values of Z_i and ε_i and the selection of the lag order is determined based on Akaike information criterion (AIC, Akaike [1974]) initializing them to zero when needed. Due to the decomposition of the log-likelihood, the estimation of the AD parameters can be achieved for each component separately and then combine the output to make inference for the price change. The authors include two separate conditioning information sets; that is, different set of explanatory variables. The first conditioning information set includes only past values of the processes. The second set includes these variables plus several market microstructure variables, such as current and lagged values of the logarithmic of trade volume and the logarithmic of time duration between trades, dummy variables to denote the hour, day of the week and month of the year in which the trade takes place, two trending variables, as well as the log of the actual price level of the stock price, but this always tested out in the empirical work.

3.3 Bayesian analysis of AD

Assuming independence among the parameters, the joint prior is $\pi(\boldsymbol{\theta}_{\mathrm{A}}, \boldsymbol{\theta}_{\mathrm{D}}) = \pi(\boldsymbol{\theta}_{\mathrm{A}})\pi(\boldsymbol{\theta}_{\mathrm{D}})$. Due to 2.3 and (3.8), the joint posterior distribution, $\pi(\boldsymbol{\theta}_{\mathrm{A}}, \boldsymbol{\theta}_{\mathrm{D}}|\boldsymbol{y})$, can be written as

$$\pi(\boldsymbol{\theta}_{\mathrm{A}}, \boldsymbol{\theta}_{\mathrm{D}}|\boldsymbol{y}) \propto \pi(\boldsymbol{\theta}_{\mathrm{A}}|\boldsymbol{\alpha})\pi(\boldsymbol{\theta}_{\mathrm{D}}|\boldsymbol{d}, \boldsymbol{\alpha} = \mathbf{1}_n), \tag{3.9}$$

where $\pi(\boldsymbol{\theta}_{\mathrm{A}}|\boldsymbol{\alpha}) \propto \pi(\boldsymbol{\alpha}|\boldsymbol{\theta}_{\mathrm{A}})\pi(\boldsymbol{\theta}_{\mathrm{A}})$ and $\pi(\boldsymbol{\theta}_{\mathrm{D}}|\boldsymbol{d}, \boldsymbol{\alpha} = \mathbf{1}_n) \propto \pi(\boldsymbol{d}|\boldsymbol{\alpha} = \mathbf{1}_n, \boldsymbol{\theta}_{\mathrm{D}})\pi(\boldsymbol{\theta}_{\mathrm{D}})$ denote the posterior distribution of the activity and the direction component, respectively. Hence, by obtaining the posterior distribution for each component of price change and by combining the outcome, the posterior distribution of the AD model is evaluated.

We now turn to specifying the prior of the parameters that we want to estimate. Note that for both factors we choose the same prior distributions and all the samplers run on the unrestricted parameter space. For the parameter $\phi = \phi_{\mathrm{A}}$ or $\phi_{\mathrm{D}} \in (-1, 1)$, which denotes the persistence parameter in the model, following Kim et al. [1998] we re-parametrize ϕ by $\psi \triangleq 0.5(\phi + 1)$, where ψ is distributed as Beta distribution with parameters (a_0, b_0) and takes values on $(0, 1)$. If not specified otherwise, $a_0 = b_0 = 0.5$. We transform the variable ψ to the unrestricted variable θ given by

$$\theta \triangleq \log(\psi) - \log(1 - \psi),$$

which takes values on $\mathcal{R}$. Its prior density is proportional to

$$\pi(\theta) \propto \theta\alpha_0 - (\alpha_0 + \beta_0)\log(1 + \exp(\theta)).$$

For the regression parameters, $\boldsymbol{\beta} = \boldsymbol{\beta}_{\mathrm{A}}$ or $\boldsymbol{\beta}_{\mathrm{D}}$, we choose a normal prior with zero mean and a prior variance equal to 10^3. The same prior is used for the $\log(\delta)$ parameter.

Independent of the prior choice, the partial posterior distributions of the parameters of the components do not have a closed form, and it is known up to a normalizing constant. Therefore, in order to generate samples from the posterior densities $\pi(\boldsymbol{\theta}_{\mathrm{A}}|\boldsymbol{\alpha})$ and $\pi(\boldsymbol{\theta}_{\mathrm{D}}|\boldsymbol{d}, \boldsymbol{\alpha} = \mathbf{1}_n)$ we apply MCMC and SMC for time-invariant parameters. We focus on the adaptive Metropolis (Haario et al. [2001]; Roberts and Rosenthal [2009]) and IBIS algorithm [Chopin, 2002] as described in Sections 2.3.2 and 2.3.3, respectively.

3.3.1 Predictive performance

To evaluate the performance of the proposed model we conduct a recursive out-of-sample forecasting procedure, using predictive likelihoods.

In our empirical study the total number of trading days are seven; the first five days are used for estimating the unknown parameters (in which the total number of trades is n), and the remaining observations (of size n') are used as a verification sample to assess the predictive accuracy of our modelling approach. The (one-step ahead) conditional predictive density for the y_{n+1}, $\pi(y_{n+1}|y_{1:n}, \boldsymbol{\theta}_{\mathrm{A}}, \boldsymbol{\theta}_{\mathrm{D}})$, can be expressed as

$$\pi(y_{n+1}|y^o_{1:n}, \boldsymbol{\theta}_{\mathrm{A}}, \boldsymbol{\theta}_{\mathrm{D}}) \propto \pi(\alpha_{n+1}|\alpha^o_{1:n}, \boldsymbol{\theta}_{\mathrm{A}})\pi(d_{n+1}|d^o_{1:n}, \alpha^o_{1:n}, \boldsymbol{\theta}_{\mathrm{D}}) \tag{3.10}$$

due to the decomposition of the price change and the decomposition of the posterior density; the superscript denotes the observed values. The terms on the left side of the equation are called the predictive distribution for the activity and direction component. Therefore, by obtaining the predictive distribution for each component of price change we may be able to evaluate the predictive distribution of price movements. Here we only present the procedure on the activity component.

At the end of time t_n, the existing samples $\{\boldsymbol{\theta}^{(l)}_{\mathrm{A}}\}^N_{l=1}$ approximate the posterior density $\pi(\boldsymbol{\theta}_{\mathrm{A}}|\alpha^o_{1:n})$, respectively. At time t_{n+1}, calculate $\mathrm{logit}(\pi^{(l)}_{A_{n+1}}) = \boldsymbol{x}^{\mathrm{A},\top}_n \boldsymbol{\beta}^{(l)}_{\mathrm{A}} + Z^{\mathrm{A},(l)}_{n+1}$, where $Z^{\mathrm{A},(l)}_{n+1} = \phi^{(l)}_{\mathrm{A}} Z^{\mathrm{A},(l)}_n + \delta^{(l)}_{\mathrm{A}} \varepsilon^{\mathrm{A},(l)}_n$ and $\varepsilon^{\mathrm{A},(l)}_n = (A_n - \pi^{(l)}_{A_n})/\sqrt{\pi^{(l)}_{A_n}(1 - \pi^{(l)}_{A_n})}$. Next, the predictive distribution of α_{n+1} can be evaluated numerically by

$$\pi(\alpha_{n+1}|\alpha^o_{1:n}) = \int \pi(\alpha_{n+1}|\boldsymbol{\theta}_{\mathrm{A}})\pi(\boldsymbol{\theta}_{\mathrm{A}}|\alpha^o_{1:n})\, d\boldsymbol{\theta}_{\mathrm{A}} = \frac{1}{N}\sum_{l=1}^{N} \pi(\alpha_{n+1}|\boldsymbol{\theta}^{(l)}_{\mathrm{A}}).$$

By replacing α_{n+1} by the observed value α^o_{n+1}, we observe the value $\pi(\alpha^o_{n+1}|\alpha^o_{1:n})$ which is called the predictive likelihood of α^o_{n+1}. Next we move one period ahead and repeat the same forecasting procedure with the information set available at time t_{n+2}. The log predictive score of the model for the evaluation period $n+1, \cdots, n'$ is the sum of the log predictive likelihoods $\sum_{l=n}^{n'-1} \log \pi(\alpha^o_{l+1}|\alpha^o_{1:l})$. Higher values indicate better (out-of-sample) forecasting ability of the model. Moreover, draws from the posterior predictive distribution can be obtained by simulating values $\alpha^{(l)}_{n+1}$ from the Bernoulli distribution with probability of success $\exp(\mathrm{logit}(\pi^{(l)}_{A_{n+1}}))/1 + \exp(\mathrm{logit}(\pi^{(l)}_{A_{n+1}}))$.

We also consider the Brier score [Brier, 1950; Blattenberger and Lad, 1985]

$$BS = \frac{1}{n'} \sum_{t=n+1}^{n'} (\hat{\pi}_{A_t} - \alpha_t^o)^2, \tag{3.11}$$

which is always between 0 and 1, with a value closer to 0 being more preferable and n' is the number of forecasting instances. The quantity $\hat{\pi}_{A_t}$ denotes the Monte Carlo [Metropolis and Ulam, 1949; von Neumann, 1963] average of the probability π_{A_t} given by

$$\hat{\pi}_{A_t} = \frac{1}{N} \sum_{l=1}^{N} \frac{\exp(\text{logit}(\pi_{A_t}^{(l)}))}{1 + \exp(\text{logit}(\pi_{A_t}^{(l)}))},$$

where $\text{logit}(\pi_{A_t}^{(l)})$ is the sampled value of $\text{logit}(\pi_{A_t})$ in iteration l, after burn in.

3.4 Simulation study

In this section we perform a simulation study to investigate the performance of IBIS, AM and maximum likelihood for the estimation of the models described in section 3.2.

Suppose that the random variables $A_1, \cdots, A_n$ are independent with conditional distribution $A_i | \mathcal{F}_i \sim \text{Bernoulli}(\pi_i)$, where

$$\text{logit}(\pi_i) = \boldsymbol{x}_i^\top \boldsymbol{\beta} + Z_i,$$
$$Z_i = \phi Z_{i-1} + \delta \varepsilon_{i-1},$$

$\varepsilon_i = (A_i - \pi_i)/\sqrt{\pi_i(1-\pi_i)}$, $Z_0 = \varepsilon_0 = 0$ and $i = 1, 2, \cdots, n$. We consider two experiments with sample size $n = 58,000$, which corresponds to the average number of our data during the trading period between 9:00 and 13:00 eastern standard time (EST).

In the first experiment, it is not included any covariate, i.e. $\boldsymbol{x}_i = 1$ and set $\phi = 0.65$, $\delta = 0.5$, $\boldsymbol{\beta} = -1.7$. For the second dataset the covariate vector is defined by $\boldsymbol{x}_i^\top = (1, i/n, \cos(2\pi i/n), \sin(2\pi i/n))$ is used. This sequence is selected in the study of Wu and Cui [2014, Section 4.1] where the process $\{Z_i\}$ is specified by an AR(1) model. The true value of the parameters are taken to be $\phi = 0.8$, $\delta = 0.1$ and $\boldsymbol{\beta}^\top = (1, -2, 2, -2)$.

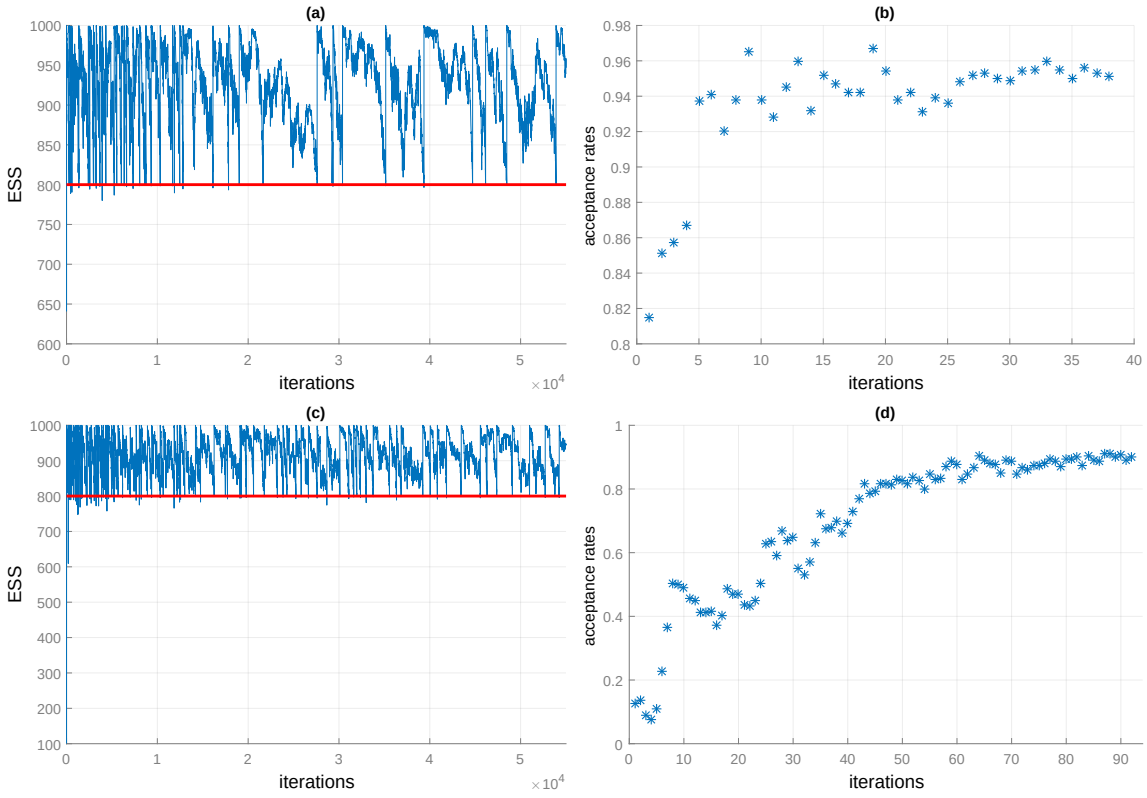
About the maximum likelihood estimator, the solver runs from the true value of the parameters and finds one local maximum. Regarding the AM, the algorithm runs 60,000 (160,000) iterations for the first (second) dataset after a burn-in of of 10,000 (20,000) iterations and the

Table 3.1: Maximum likelihood, AM and IBIS estimates of parameters for the simulated datasets. 'Run-time' returns the execution time in minutes. The Bayesian estimates represent posterior means and standard deviations in parenthesis. For the maximum likelihood in parenthesis is reported the asymptotic standard errors and the solver runs from the true parameter vector. The AM algorithm runs 70,000 (180,000) iterations with a burn-in period of 10,000 (20,000) draws, thinning every 30th (80th) iteration over the first (second) simulation, yielding 2000 draws; the algorithm is initialized with the true parameter vector. For IBIS, the initial particles are sampled from the posterior density based on the first 3000 observations. Model: A_i takes on value 1 (0) with probability π_i $(1-\pi_i)$, where $\text{logit}(\pi_i) = \boldsymbol{x}_{i-1}^\top \boldsymbol{\beta} + Z_i$, $Z_i = \phi Z_{i-1} + \delta\varepsilon_{i-1}$, $\varepsilon_i = (A_i - \pi_i)/\sqrt{\pi_i(1-\pi_i)}$, $Z_0 = \varepsilon_0 = 0$, $i = 1, \cdots, n$, $n = 58,000$. Simulation 1: $\boldsymbol{x}_i = 1$, $\boldsymbol{\beta} = \beta_0$, Simulation 2: $\boldsymbol{x}_i^\top = (1, i/n, \cos(2\pi i/n), \sin(2\pi i/n))$. Priors: $(\phi+1)/2 \sim \mathcal{B}e(0.5, 0.5)$, $\log(\delta) \sim \mathcal{N}(0, 10^3)$, $d \in \{1, 4\}$.

				IBIS Number of Particles		
Parameter	True	MLE	AM	1000	2000	5000
Simulation 1						
ϕ	0.65	0.641	0.641	0.641	0.641	0.641
		(0.011)	(0.012)	(0.012)	(0.012)	(0.012)
δ	0.5	0.497	0.494	0.494	0.495	0.495
		(0.008)	(0.008)	(0.008)	(0.008)	(0.008)
β_0	-1.7	-1.6901	-1.689	-1.689	-1.689	-1.690
		(0.018)	(0.018)	(0.018)	(0.018)	(0.018)
Run-time		0.008	4.76	11.68	22.84	54.52
Simulation 2						
ϕ	-0.8	-0.797	-0.790	-0.790	-0.788	-0.790
		(0.027)	(0.033)	(0.032)	(0.033)	(0.033)
δ	0.1	0.087	0.088	0.088	0.088	0.088
		(0.008)	(0.009)	(0.009)	(0.009)	(0.009)
β_0	1.0	1.023	1.024	1.023	1.023	1.024
		(0.025)	(0.024)	(0.024)	(0.025)	(0.025)
β_1	-2.0	-2.052	-2.052	-2.052	-2.051	-2.052
		(0.043)	(0.043)	(0.042)	(0.042)	(0.043)
β_2	2.0	1.991	1.9913	1.992	1.991	1.991
		(0.019)	(0.019)	(0.019)	(0.019)	(0.019)
β_3	-2.0	-1.994	-1.995	-1.994	-1.994	-1.994
		(0.018)	(0.018)	(0.018)	(0.017)	(0.018)
Run-time		0.0225	12.22	13.58	25.64	62.11

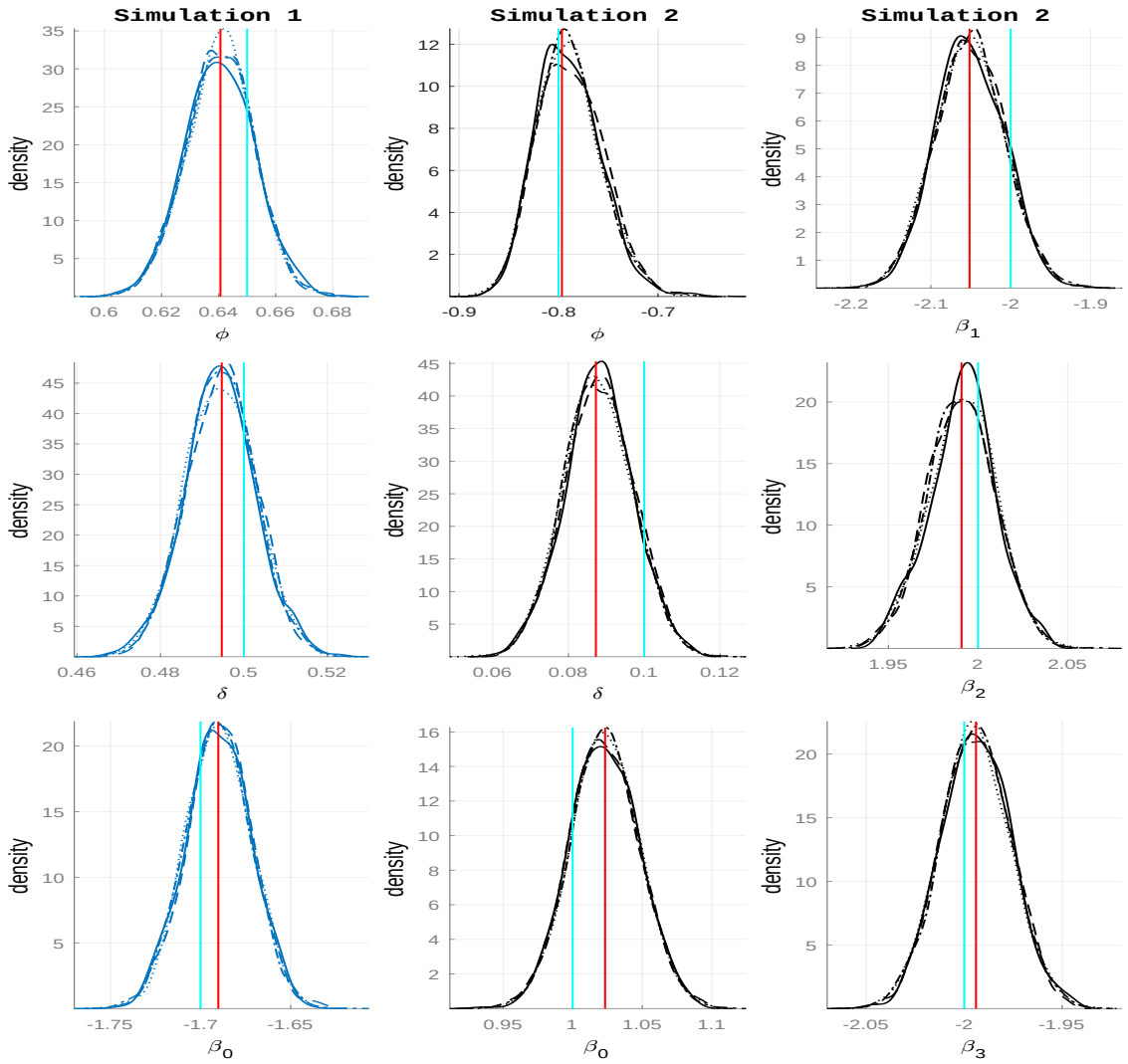
autocorrelation is reduced by retaining only every 30th (80th) iteration of the chain. Finally, the resulting sample consists of 2000 draws and the components of the parameter vector are updated in a single block. The acceptance ratio is 31.7% (27.7%). As a starting point of the algorithm is used the true parameter value. IBIS algorithm is initialized with MCMC draws based on the first 3000 observations. The algorithm runs three times over the rest data sequence, with number of particles $N \in \{1000, 2000, 5000\}$, and discuss how the estimated parameters are affected. The resampling threshold is set to 80% and the resampling method is the stratified resampling scheme [Carpenter et al., 1999].

The simulation results for all datasets and algorithms considered are presented in Table 4.1 along with the running time (in minutes) that describes the amount of time it takes to run an algorithm. The Bayesian estimates reported in the table represent posterior means and standard



Model: A_i takes on value 1 (0) with probability π_i $(1-\pi_i)$, where $\text{logit}(\pi_i) = \boldsymbol{x}_i^\top \boldsymbol{\beta} + Z_i$, $Z_i = \phi Z_{i-1} + \delta\varepsilon_{i-1}$, $\varepsilon_i = (A_i - \pi_i)/\sqrt{\pi_i(1-\pi_i)}$, $Z_0 = \varepsilon_0 = 0$, $i = 1, \cdots, n$, $n = 58,000$. Simulation 1: $\boldsymbol{x}_{i-1} = 1$. Simulation 2: $\boldsymbol{x}_{i-1}^\top = (1, i/n, \cos(2\pi i/n), \sin(2\pi i/n))$.

Figure 3.1: Results of IBIS estimation for the simulated dataset. ESS along the iterations (first column), and acceptance rate at each move step (second column). The number of particles is 1000 and a resample-move step is triggered when ESS drops below 800 (red line). The ith row illustrates the results of the jth simulation study, $j = 1, 2$.



Model: A_i takes on value 1 (0) with probability π_i $(1-\pi_i)$, where $\text{logit}(\pi_i) = \boldsymbol{x}_i^\top \boldsymbol{\beta} + Z_i$, $Z_i = \phi Z_{i-1} + \delta \varepsilon_{i-1}$, $\varepsilon_i = (A_i - \pi_i)/\sqrt{\pi_i(1-\pi_i)}$, $Z_0 = \varepsilon_0 = 0$, $i = 1, \cdots, n$, $n = 58,000$. Simulation 1: $\boldsymbol{x}_i = 1$. Simulation 2: $\boldsymbol{x}_i^\top = (1, i/n, \cos(2\pi i/n), \sin(2\pi i/n))$. Priors: $(\phi+1)/2 \sim \mathcal{B}e(0.5, 0.5)$, $\log(\delta) \sim \mathcal{N}(0, 10^3)$, $\boldsymbol{\beta} \sim \mathcal{N}_d(\mathbf{0}_d, 10^3 \mathbf{I}_d)$, $d \in \{1, 4\}$.

Figure 3.2: Marginal posterior densities of the estimated parameters for the simulated dataset, estimated from MCMC and IBIS samples using a kernel density (blue and black curve). Red and cyan lines represent the mle and the true parameter vector, respectively.

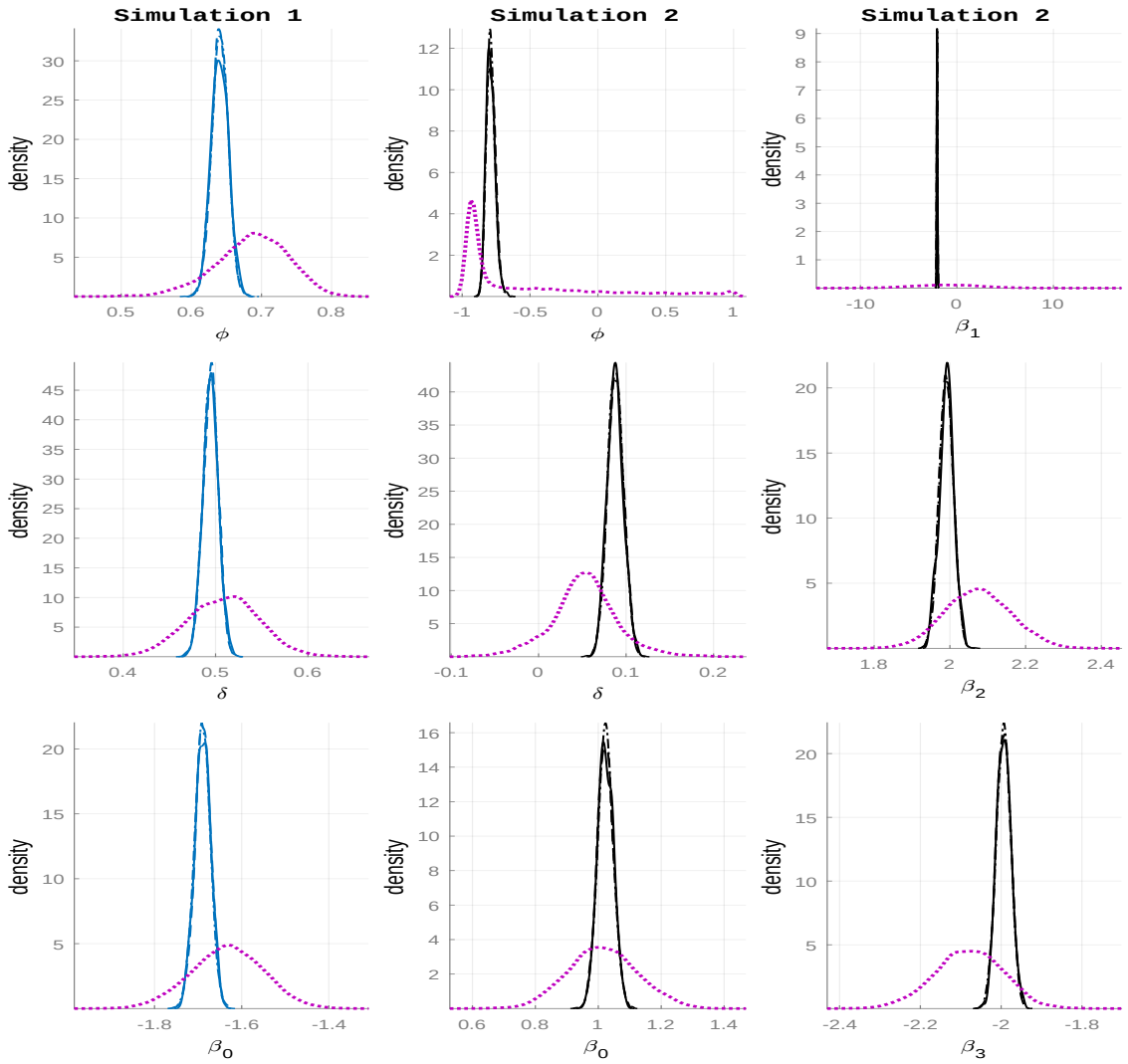
deviations in parenthesis. For the maximum likelihood in parenthesis is reported the asymptotic standard errors. Regarding the running time, for the MCMC it shows the time requirements of the algorithm using the whole dataset to draw a thinned chain with 2000 samples including the burn-in length, for IBIS it determines the necessary time to draw N particles without including the time needed to generate the initial set of particles and for the maximum likelihood it gives the time needed by the optimization algorithm to end using only one starting value. As shown

in the table, all three methods give almost identical results. In particularly, as the number of particles increases, the computational time increases linearly, but the estimated parameters are not significantly affected. Moreover, by comparing the ratio of execution time between AM and IBIS, adjusted to generate 2000 samples, the first method is almost five (two) times faster over the first (second) simulation, thus it is reduced by increasing the number of estimated parameters.

Figure 3.1 shows the ESS (left plots) along the iterations as well as the acceptance rates (right plots) of the move steps, with 1000 particles. Panels (a, b) of the figure show the results over for the first simulation, while panels (c, d) depict the results of the second simulation. Since the resampling threshold is set to 80%, a resample-move is triggered when ESS drops below 800. As expected, the frequency of the resample-moves steps seems to decrease over time. On the other hand, the acceptance rate in the first simulation is quite high along the iterations, while in the second simulation is low in the beginning, increases steeply to 50.2%, and continues to increase gradually to approximately 90%. Similar results are obtained for the other particles of the corresponding plots; these figures are not presented here. Figure 3.2 illustrates the marginal posterior densities of the estimated parameters, estimated from AM and IBIS samples using a kernel density; the two methods produce nearly identical posterior distributions. The figure also marks the maximum likelihood estimators; it seems that they coincide with the posterior mode. In Figure 3.3, the wider density represents the sampling density that generates the initial particles for IBIS, and the shifted and tighter curve shows the target posterior density based on the whole dataset. The number of samples is 1000 and this tightening is expected as additional points become available.

3.5 Real data results

In this section we report and evaluate the results of the analysis of ES price changes. The estimation is based on the AD model with two different set of explanatory variables. The total number of trading days are seven; the first five days in the study are used for estimating the unknown parameters, and the rest for predicting price changes. The prediction is discussed in the next chapter.



Model: A_i takes on value 1 (0) with probability π_i $(1-\pi_i)$, where $\text{logit}(\pi_i) = \boldsymbol{x}_{i-1}^\top \boldsymbol{\beta} + Z_i$, $Z_i = \phi Z_{i-1} + \delta\varepsilon_{i-1}$, $\varepsilon_i = (A_i - \pi_i)/\sqrt{\pi_i(1-\pi_i)}$, $Z_0 = \varepsilon_0 = 0$, $i = 1, \cdots, n$, $n = 58,000$. Simulation 1: $\boldsymbol{x}_{i-1} = 1$. Simulation 2: $\boldsymbol{x}_{i-1}^\top = (1, i/n, \cos(2\pi i/n), \sin(2\pi i/n))$. Priors: $(\phi+1)/2 \sim \mathcal{B}e(0.5, 0.5)$, $\log(\delta) \sim \mathcal{N}(0, 10^3)$, $\boldsymbol{\beta} \sim \mathcal{N}_d(\mathbf{0}_d, 10^3\mathbf{I}_d)$, $d \in \{1, 4\}$.

Figure 3.3: Marginal posterior density of the parameters for the simulated dataset model, estimated from IBIS samples using a kernel density (blue and black curve). Kernel density estimation of $\pi(\phi, \delta, \boldsymbol{\beta}|A_{1:n_0})$ using MCMC samples, where $n_0 = 3000$ (purple curve).

3.5.1 Preliminaries

Table 3.2 presents the variable definitions. Both maximum likelihood and Bayesian estimation is conducted. The `fmincon` function in MATLAB's optimization toolbox is used for the maximum likelihood estimation with the interior-point optimization algorithm. Furthermore, analytic first derivatives are provided and MATLAB approximates the Hessian internally. We also have let `fmincon` to approximate the first gradients numerically, leading to similar results. In MCMC,

Table 3.2: Variable Definitions.

Variable Name	Description
	Continuous Variables
t_i	the time at which the ith transaction occurs
Δt_i	$\triangleq t_i - t_{i-1}$
τ_i	$\triangleq \log(\Delta t_i + 1)$
$P_i^{\text{b,j}}$	the jth best bid price just after the ith trade
$P_i^{\text{a,j}}$	the jth best ask price just after the ith trade
P_i^{mo}	the ith trade price
$V_i^{\text{b,j}}$	the log total volume on the jth best bid quote right after the ith trade
$V_i^{\text{a,j}}$	the log total volume on the jth best ask quote right after the ith trade
V_i^{mo}	the log volume of the ith trade
	Discrete Variables
SP_i	the spread (in tick) instantaneously after the ith trade
G_i^{b}	$\triangleq P_i^{\text{b,1}} - P_i^{\text{b,2}}$ (in tick)
G_i^{a}	$\triangleq P_i^{\text{a,2}} - P_i^{\text{a,1}}$ (in tick)
	Dummy Variables
BMO_i	1: if the ith trade is a buy market order, 0: otherwise
D_i'	1: if the ith active trade moves the price up, 0: moves down

Index j corresponds to the jth limit order level, $j = 1, 2$. Index i corresponds to the ith trade.

samples are drawn from the posterior by AM algorithm with a burn-in period during which the samples are discarded. Correlation between successive chain draws is reduced by retaining only every kth iteration; for this task the autocorrelation plots are examined. Finally, the resulting sample consists of 2000 draws and the components of the parameter vector are updated in a single block. In SMC, one key parameter of IBIS is the number of particles. In this study, the algorithm runs three times over the whole data sequence, with number of particles $N \in \{1000, 2000, 5000\}$ and discuss how the estimated parameters are affected. Another important issue is the generation of the initial set of particles from the posterior distribution of the unknown parameters. When the algorithm is initialized from the prior density, it degenerates rapidly due to the flat priors [Chopin, 2002]. Therefore, MCMC is applied to initialize the algorithm based on a few thousand observations. Besides, the resampling threshold is set to 80%, and the resampling method is the stratified resampling scheme.

MCMC and maximum likelihood estimation require the use of starting values for the parameters. For the mle, the algorithm runs from different initialization points, the corresponding log-likelihood is recorded and pick the one that is largest. More specifically, the initial value for $\boldsymbol{\beta}$ is set to the returned coefficient estimate for the standard logistic regression to predict the probability of a price movement or an upward price change, while a list of candidate values for ϕ and δ is defined. The MCMC algorithm is initialized with the maximum likelihood solution instead of using random initial values.

The execution time, that is the amount of time it takes to run an algorithm, is denoted by the variable name 'Run-time'. It is calculated using `tic` and `toc` MATLAB commands; the elapsed time is recorded in seconds. For AM, the variable shows the time requirements of the algorithm using the whole dataset to draw a thinned chain with 2000 samples; this time includes the burn-in length. For IBIS, it determines the necessary time to draw N particles without including the time needed to generate the initial set of particles. For the maximum likelihood estimation, the variable gives the average time needed by the optimization algorithm to end using only one starting value; it equals the time required to find the global maxima using all different starting points divided by the number of the initialization points.

The estimated model parameters are displayed in tables, using the three estimation methods, along with the running time. The Bayesian estimates represent posterior means and standard deviations in parenthesis. For the maximum likelihood in parenthesis is reported the asymptotic standard errors. The marginal posterior densities of the estimated parameters with AM and IBIS samples using a kernel density are displayed.

3.5.2 Part A

This section provides results for modelling the activity of ES. The empirical analysis is done on two separated datasets: morning and afternoon, between 16 May (Monday) and 20 May (Friday) with 57,801 and 322,182 observations, respectively. For the process of regressors we consider two cases:

- *Case I*: $\boldsymbol{x}_{i-1}^{\mathrm{A},\top} = (1, \mathrm{DAct}_{ii}, A_{ii})$.
- *Case II:* $\boldsymbol{x}_{i-1}^{\mathrm{A},\top} = (1, \mathrm{DAct}_{ii}, A_{ii}, \tau_{ii}, \mathrm{BMO}_{ii}, V_{ii}^{\mathrm{mo}}, SP_{ii}, V_{ii}^{\mathrm{b},1}, V_{ii}^{\mathrm{b},2}, V_{ii}^{\mathrm{a},1}, V_{ii}^{\mathrm{a},2}, G_{ii}^{\mathrm{b}}, G_{ii}^{\mathrm{a}})$,

where $ii \in \{i-2, i-1\}$. Thus, the total number of possible covariates are 4 and 24 in the

first and second case, respectively. The definition of the variables is detailed in Table 3.2. In the following sections, the estimation results for the two cases are presented. If we do not make a specific reference to which data segment we are referring to, it means that we imply both examined time periods.

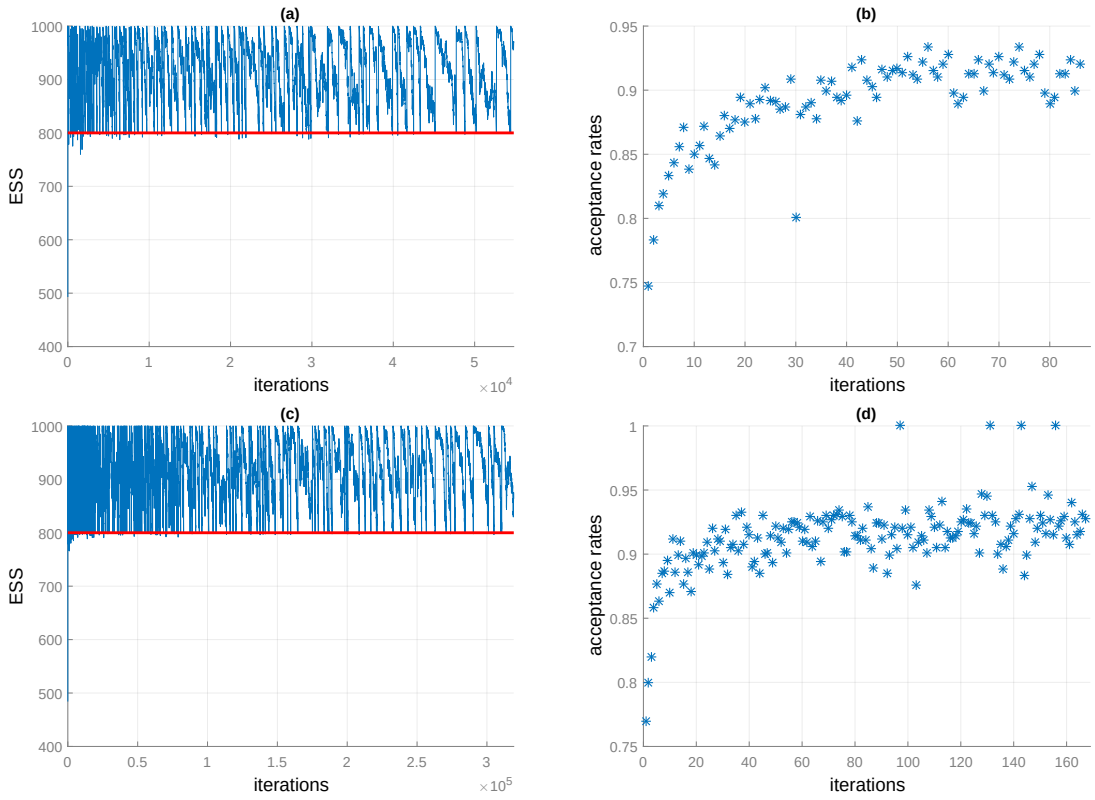
Table 3.3: Maximum likelihood, AM and IBIS estimates of parameters for the activity model for ES, May 16th to May 20th, case (I). 'Run-time' returns the execution time in minutes. The Bayesian estimates represent posterior means and standard deviations in parenthesis. For the maximum likelihood in parenthesis is reported the asymptotic standard errors and the solver runs from multiple initialization points. The AM algorithm runs 140,000 (190,000) iterations with a burn-in period of 50,000 draws, thinning every 45th (70th) iteration over the morning (afternoon) period, yielding 2000 draws; the algorithm is initialized with the mle. For IBIS, the initial particles are sampled from the posterior density based on the first 3000 observations. Model: A_i takes on value 1 (0) with probability π_i $(1-\pi_i)$, where $\text{logit}(\pi_i) = \boldsymbol{x}_{i-1}^\top \boldsymbol{\beta} + Z_i$, $Z_i = \phi Z_{i-1} + \delta\varepsilon_{i-1}$, $\varepsilon_i = (A_i - \pi_i)/\sqrt{\pi_i(1-\pi_i)}$, $Z_0 = \varepsilon_0 = 0$, $\boldsymbol{x}_{i-1}^\top = (1, A_{i-1}, A_{i-2})$. Priors: $(\phi+1)/2 \sim \mathcal{B}e(0.5, 0.5)$, $\log(\delta) \sim \mathcal{N}(0, 10^3)$, $\boldsymbol{\beta} \sim \mathcal{N}_3(\mathbf{0}_3, 10^3\mathbf{I}_3)$.

Parameter	MLE	AM	IBIS Number of Particles		
			1000	2000	5000
Morning period					
ϕ	0.571	0.571	0.572	0.571	0.572
	(0.009)	(0.011)	(0.011)	(0.011)	(0.011)
δ	1.086	1.085	1.083	1.084	1.084
	(0.034)	(0.033)	(0.033)	(0.033)	(0.033)
β_0	-1.292	-1.292	-1.294	-1.295	-1.294
	(0.033)	(0.031)	(0.033)	(0.031)	(0.031)
β_1	-1.592	-1.586	-1.583	-1.587	-1.586
	(0.095)	(0.091)	(0.090)	(0.089)	(0.090)
β_2	-0.526	-0.524	-0.524	-0.525	-0.524
	(0.036)	(0.035)	(0.034)	(0.034)	(0.034)
Run-time	0.8482	9.618	12.851	24.998	60.278
Afternoon period					
ϕ	0.691	0.691	0.692	0.692	0.692
	(0.003)	(0.004)	(0.004)	(0.004)	(0.004)
δ	1.083	1.083	1.082	1.082	1.082
	(0.013)	(0.015)	(0.015)	(0.014)	(0.016)
β_0	-1.123	-1.123	-1.126	-1.126	-1.126
	(0.016)	(0.016)	(0.016)	(0.016)	(0.016)
β_1	-1.477	-1.476	-1.476	-1.476	-1.476
	(0.035)	(0.038)	(0.038)	(0.037)	(0.038)
β_2	-0.608	-0.607	-0.608	-0.608	-0.608
	(0.015)	(0.013)	(0.013)	(0.013)	(0.013)
Run-time	4.826	71.773	106.783	180.386	404.429

3.5.2.1 Case (I)

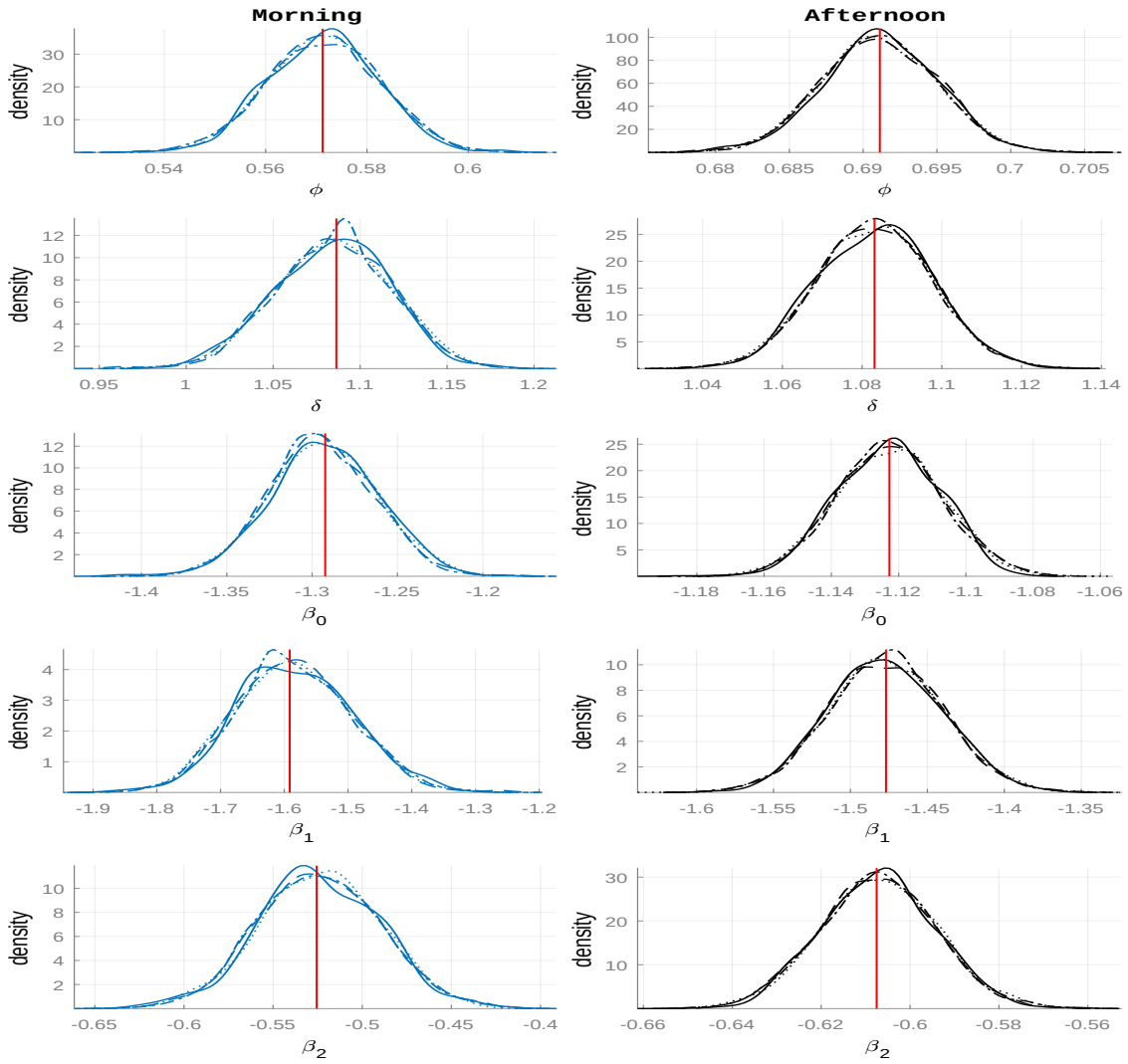
As a starting point, we examine which of the variables do have a significant effect on the price activity. There seems that the direction of a price movement does not affect the probability of having a future price change different from zero. Therefore, the final covariate vector is $\boldsymbol{x}_{i-1}^{\mathrm{A},\top} = (1, A_{i-1}, A_{i-2})$.

The AM algorithm runs 140,000 (190,000) iterations for the morning (afternoon) dataset with burn-in of 50,000 iterations and the autocorrelation is reduced by retaining only every 45th (70th) iteration of the chain. For IBIS, we use the first 3000 observations to initialize the algorithm. Table 3.3 lists the parameter estimates of the price activity model, using the three estimation methods, along with the running time (in minutes). The overall impression is that AM and IBIS sample estimates agree well with the maximum likelihood estimates. A closer look



Model: A_i takes on value 1 (0) with probability π_i $(1-\pi_i)$, where $\text{logit}(\pi_i) = \boldsymbol{x}_{i-1}^{\top}\boldsymbol{\beta} + Z_i$, $Z_i = \phi Z_{i-1} + \delta\varepsilon_{i-1}$, $\varepsilon_i = (A_i - \pi_i)/\sqrt{\pi_i(1-\pi_i)}$, $i = 1, 2, \cdots, n$, $Z_0 = \varepsilon_0 = 0$. $\boldsymbol{x}_{i-1}^{\top} = (1, A_{i-1}, A_{i-2})$.

Figure 3.4: Results of IBIS estimation for ES, May 16th to May 20th, during the morning (panels a, b) and afternoon period (panels c, d), case (I). ESS along the iterations (first column), and acceptance rate at each move step (second column). The number of particles is 1000 and a resample-move step is triggered when ESS drops below 800 (red line).



Model: A_i takes on value 1 (0) with probability π_i $(1-\pi_i)$, where $\text{logit}(\pi_i) = \boldsymbol{x}_{i-1}^\top \boldsymbol{\beta} + Z_i$, $Z_i = \phi Z_{i-1} + \delta\varepsilon_{i-1}$, $\varepsilon_i = (A_i - \pi_i)/\sqrt{\pi_i(1-\pi_i)}$, $i = 1, 2, \cdots, n$, $Z_0 = \varepsilon_0 = 0$. $\boldsymbol{x}_{i-1}^\top = (1, A_{i-1}, A_{i-2})$. Priors: $(\phi+1)/2 \sim \mathcal{B}e(0.5, 0.5)$, $\log(\delta) \sim \mathcal{N}(0, 10^3)$, $\boldsymbol{\beta} \sim \mathcal{N}_3(\mathbf{0}_3, 10^3\mathbf{I}_3)$.

Figure 3.5: Marginal posterior densities of the estimated parameters for the activity model, estimated from AM (dotted curve) and IBIS (solid, dashed and dash-dotted curve for 1000, 2000 and 5000 particles, respectively) samples using a kernel density. Vertical line represents the mle. The data set we are analyzing is ES, May 16th to May 20th, confined to the morning partition (blue curve) and afternoon partition (black curve), case (I).

at the table shows that all estimated coefficients (except β_2) are slightly larger for the afternoon dataset compared to the morning dataset. Besides, during the afternoon the standard error of estimation is about one-third than those during the morning. Furthermore, the coefficient on the lagged price activity is negative for all past period values, indicating that past active trades tend to decrease the probability of subsequent movements in the price, while this reduction

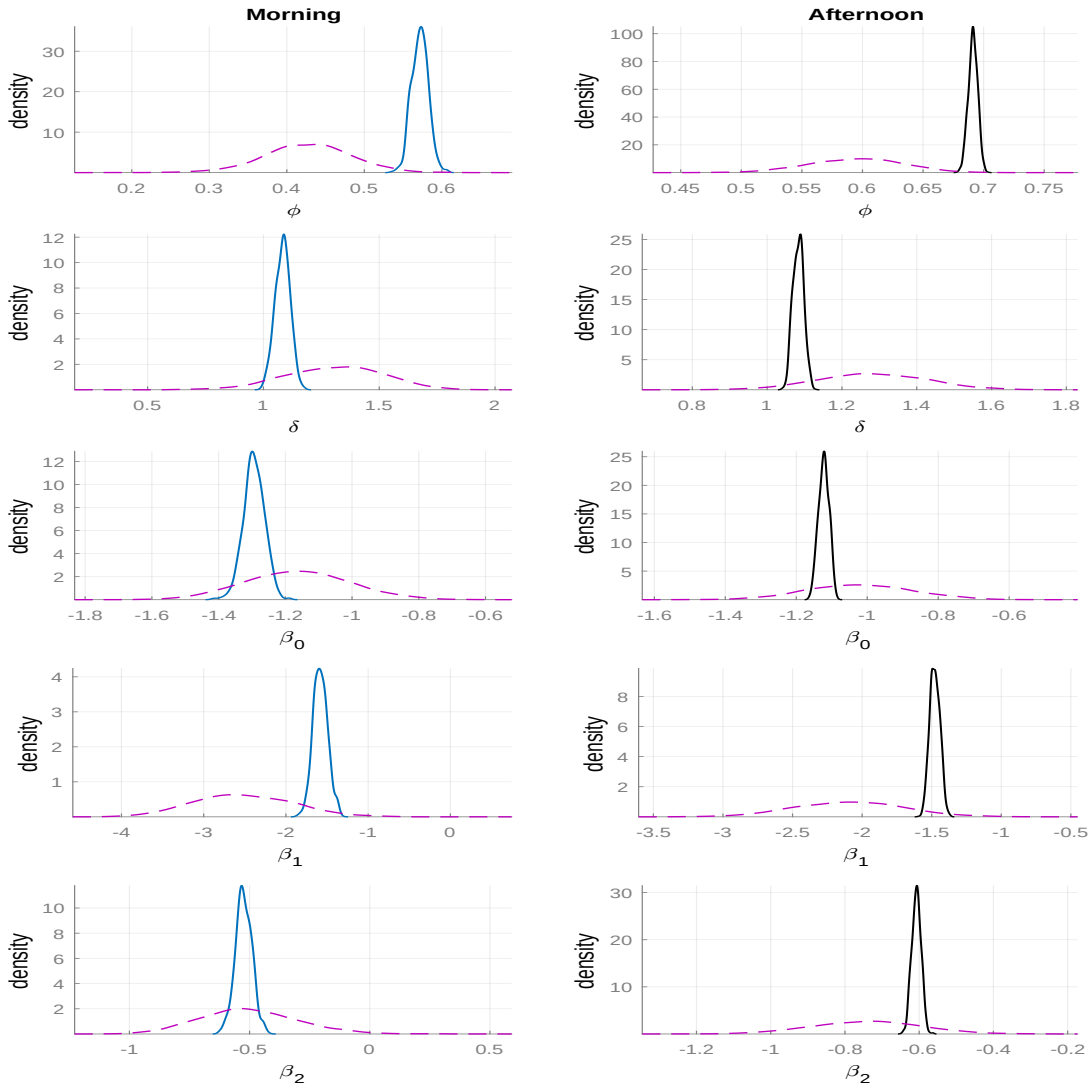
decays down at lag two. It can be observed, that by increasing the number of particles does not significantly affect the estimated parameters. However, since the computational cost grows with the number of particles, it is suggested to use 1000 particles in the filter. Moreover, comparing the run time of AM and IBIS to generate 2000 draws, the first method is four times faster.

Figure 3.4 shows the ESS along the iterations (left plots) as well as the acceptance rates of the move steps (right plots) with 1000 particles. Panels (a, b) of the figure show the results over for the morning period, while panels (c, d) for the afternoon. Since the resampling threshold is set to 80%, a resample-move is triggered when ESS drops below 800 (red line). As expected, the frequency of the resample-move steps seems to decrease over time. On the other hand, the acceptance rate is quite high along the iterations. Similar results are obtained for the other particles of the corresponding plots; these figures are not presented here. Figure 3.5 illustrates the marginal posterior densities and it seems that AM and IBIS produce nearly identical posterior distributions. Note that by increasing the number of particles we obtain relatively similar estimates. The figure also marks the maximum likelihood estimators; it seems that they coincide with the posterior mode. In Figure 3.6, the wider density represents the sampling density that generates the initial particles for IBIS, and the shifted and tighter curve shows the target posterior density based on the whole dataset. The number of samples is 1000 and this tightening is expected as additional points become available.

3.5.2.2 Case (II)

It turns out that the inclusion of past information of both the direction of a price movement and the spread do not affect the probability of having a future price change different from zero. Besides, the type of lagged market order (buy or sell) do not reveals traders' expectations with respect to future price movements. Concerning the log bid and ask volume, only the lag-1 volumes of the first level are statistically significant and become insignificant for higher lags or levels. The best bid and ask limit gap price is not significant. Finally, the log trading volume at the previous transaction is significant for both time periods, while its penultimate value is significant only for the afternoon subset. After testing out insignificant explanatory variables we end up with $\boldsymbol{x}_{i-1}^{\mathrm{A},\top} = (1, A_{i-1}, \tau_{i-1}, V_{i-1}^{\mathrm{mo}}, V_{i-1}^{\mathrm{b,l}}, V_{i-1}^{\mathrm{a,l}}, A_{i-2}, \tau_{i-2})$ for the morning subset, and $\boldsymbol{x}_{i-1}^{\top} = (1, A_{i-1}, \tau_{i-1}, V_{i-1}^{\mathrm{mo}}, V_{i-1}^{\mathrm{b,l}}, V_{i-1}^{\mathrm{a,l}}, A_{i-2}, \tau_{i-2}, V_{i-2}^{\mathrm{mo}})$ for the afternoon subset.

The AM algorithm runs 750,000 (1,150,000) iterations with burn-in of 150,000 (400,000)



Model: A_i takes on value 1 (0) with probability π_i $(1-\pi_i)$, where $\text{logit}(\pi_i) = \boldsymbol{x}_{i-1}^\top \boldsymbol{\beta} + Z_i$, $Z_i = \phi Z_{i-1} + \delta\varepsilon_{i-1}$, $\varepsilon_i = (A_i - \pi_i)/\sqrt{\pi_i(1-\pi_i)}$, $i = 1, 2, \cdots, n$, $Z_0 = \varepsilon_0 = 0$. $\boldsymbol{x}_{i-1}^\top = (1, A_{i-1}, A_{i-2})$. Priors: $(\phi+1)/2 \sim \mathcal{B}e(0.5, 0.5)$, $\log(\delta) \sim \mathcal{N}(0, 10^3)$, $\boldsymbol{\beta} \sim \mathcal{N}_3(\mathbf{0}_3, 10^3\mathbf{I}_3)$.

Figure 3.6: Comparison of kernel density estimation of marginal posterior densities based on the whole dataset (solid line) and the first 3000 observations (purple curve) with 1000 IBIS and MCMC samples, respectively. For inference is used ES, May 16th to May 20th, during the morning (blue curve) and afternoon period (black curve), case (I).

iterations and the autocorrelation is reduced by retaining only every 300th (350th) iteration of the chain. The acceptance ratio is 30.24%. For IBIS, we use the first 3000 observations to initialize the algorithm.

Tables 3.4 and 3.5 list the parameter estimates of the price activity model for the morning and afternoon segment, respectively, using the three estimation methods, along with the running time (in minutes). A closer look at the table shows that the majority of the estimated coefficients

Table 3.4: Maximum likelihood, AM and IBIS estimates of parameters for the activity model for ES, May 16th to May 20th, 9 a.m. to 1 p.m., case (II). 'Run-time' returns the execution time in minutes. The Bayesian estimates represent posterior means and standard deviations in parenthesis. For the maximum likelihood in parenthesis is reported the asymptotic standard errors and the solver runs from multiple initialization points. The AM algorithm runs 750,000 iterations with a burn-in period of 150,000 draws, thinning every 300th iteration, yielding 2000 draws; the algorithm is initialized with the mle. For IBIS, the initial particles are sampled from the posterior density based on the first 3000 observations. The tick size is \$0.25. Model: A_i takes on value 1 (0) with probability π_i $(1-\pi_i)$, where $\text{logit}(\pi_i) = \boldsymbol{x}_{i-1}^\top \boldsymbol{\beta} + Z_i$, $Z_i = \phi Z_{i-1} + \delta\varepsilon_{i-1}$, $\varepsilon_i = (A_i - \pi_i)/\sqrt{\pi_i(1-\pi_i)}$, $Z_0 = \varepsilon_0 = 0$, $\boldsymbol{x}_{i-1}^\top = (1, A_{i-1}, \tau_{i-1}, V_{i-1}^{\text{mo}}, V_{i-1}^{\text{b,1}}, V_{i-1}^{\text{a,1}}, A_{i-2}, \tau_{i-2})$. Priors: $(\phi+1)/2 \sim \mathcal{B}e(0.5, 0.5)$, $\log(\delta) \sim \mathcal{N}(0, 10^3)$, $\boldsymbol{\beta} \sim \mathcal{N}_8(\mathbf{0}_8, 10^3\mathbf{I}_8)$.

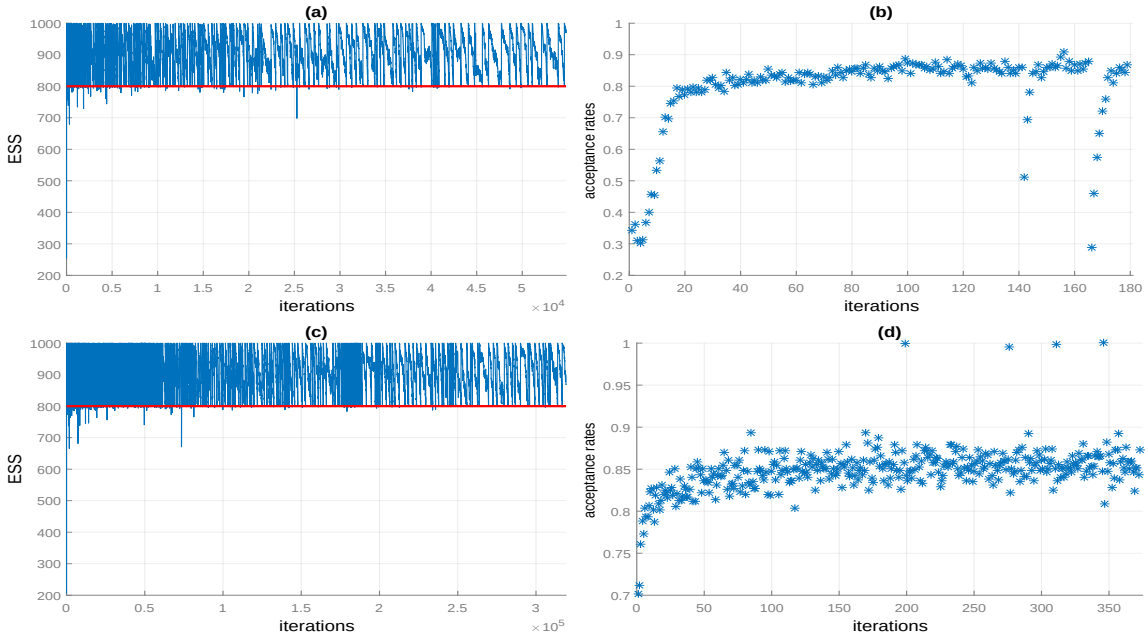
Parameter	MLE	AM	IBIS		
			Number of Particles		
			1000	2000	5000
ϕ	0.571	0.573	0.572	0.572	0.572
	(0.011)	(0.014)	(0.013)	(0.013)	(0.013)
δ	0.912	0.908	0.909	0.910	0.909
	(0.029)	(0.031)	(0.032)	(0.032)	(0.032)
β_0	-3.243	-3.248	-3.243	-3.243	-3.246
	(0.096)	(0.096)	(0.091)	(0.097)	(0.095)
β_1	-1.201	-1.192	-1.195	-1.199	-1.195
	(0.082)	(0.084)	(0.085)	(0.085)	(0.087)
β_2	0.369	0.371	0.369	0.369	0.369
	(0.023)	(0.022)	(0.021)	(0.021)	(0.022)
β_3	-0.055	-0.055	-0.055	-0.055	-0.055
	(0.008)	(0.008)	(0.008)	(0.008)	(0.008)
β_4	0.124	0.124	0.124	0.124	0.124
	(0.010)	(0.010)	(0.010)	(0.010)	(0.010)
β_5	0.142	0.1423	0.142	0.142	0.142
	(0.013)	(0.012)	(0.012)	(0.012)	(0.012)
β_6	-0.461	-0.4610	-0.459	-0.461	-0.459
	(0.035)	(0.035)	(0.034)	(0.035)	(0.034)
β_7	0.490	0.489	0.488	0.488	0.489
	(0.024)	(0.023)	(0.022)	(0.022)	(0.023)
Run-time	1.518	24.783	16.004	31.903	78.379

are slightly larger during the afternoon. The influence of lagged price activity, is negative for all past period values, indicating that past active trades tend to decrease the probability of subsequent movements in the price, while this reduction decays down at lag two. The table shows

Table 3.5: Maximum likelihood, AM and IBIS estimates of parameters for the activity model for ES, May 16th to May 20th, 1 a.m. to 5 p.m., case (II). 'Run-time' returns the execution time in minutes. The Bayesian estimates represent posterior means and standard deviations in parenthesis. For the maximum likelihood in parenthesis is reported the asymptotic standard errors. The AM algorithm runs 750,000 iterations with a burn-in period of 150,000 draws, thinning every 300th iteration, yielding 2000 draws; the algorithm is initialized with the mle. For IBIS, the initial particles are sampled from the posterior density based on the first 3000 observations. The tick size is \$0.25. Model: A_i takes on value 1 (0) with probability π_i $(1-\pi_i)$, where $\text{logit}(\pi_i) = \boldsymbol{x}_{i-1}^\top \boldsymbol{\beta} + Z_i$, $Z_i = \phi Z_{i-1} + \delta\varepsilon_{i-1}$, $\varepsilon_i = (A_i - \pi_i)/\sqrt{\pi_i(1-\pi_i)}$, $Z_0 = \varepsilon_0 = 0$, $\boldsymbol{x}_{i-1}^\top = (1, A_{i-1}, \tau_{i-1}, V_{i-1}^{\text{mo}}, V_{i-1}^{\text{b,l}}, V_{i-1}^{\text{a,l}}, A_{i-2}, \tau_{i-2}, V_{i-2}^{\text{mo}})$. Priors: $(\phi+1)/2 \sim \mathcal{B}e(0.5, 0.5)$, $\log(\delta) \sim \mathcal{N}(0, 10^3)$, $\boldsymbol{\beta} \sim \mathcal{N}_9(\mathbf{0}_9, 10^3\mathbf{I}_9)$.

Parameter	MLE	AM	IBIS		
			Number of Particles		
			1000	2000	5000
ϕ	0.629	0.629	0.629	0.629	0.629
	(0.004)	(0.004)	(0.005)	(0.005)	(0.005)
δ	0.953	0.961	0.958	0.959	0.958
	(0.013)	(0.008)	(0.015)	(0.014)	(0.014)
β_0	-4.045	-4.033	-4.048	-4.047	-4.045
	(0.0636)	(0.049)	(0.066)	(0.064)	(0.065)
β_1	-1.253	-1.258	-1.251	-1.253	-1.252
	(0.033)	(0.025)	(0.036)	(0.036)	(0.036)
β_2	0.523	0.529	0.523	0.523	0.523
	(0.020)	(0.016)	(0.020)	(0.021)	(0.021)
β_3	-0.108	-0.109	-0.108	-0.108	-0.108
	(0.003)	(0.001)	(0.003)	(0.003)	(0.002)
β_4	0.212	0.210	0.212	0.212	0.212
	(0.006)	(0.004)	(0.006)	(0.006)	(0.006)
β_5	0.239	0.236	0.240	0.240	0.240
	(0.007)	(0.006)	(0.007)	(0.006)	(0.007)
β_6	-0.449	-0.450	-0.449	-0.449	-0.449
	(0.014)	(0.009)	(0.013)	(0.013)	(0.013)
β_7	1.011	1.011	1.010	1.010	1.010
	(0.021)	(0.017)	(0.021)	(0.021)	(0.021)
β_8	-0.075	-0.075	-0.075	-0.075	-0.075
	(0.003)	(0.003)	(0.003)	(0.003)	(0.003)
Run-time	9.061	235.135	151.481	249.542	554.504

that lagged log-durations have a very dramatic positive impact on the chance that a trade moves the transaction price, with their influence being larger at the second lag. Besides, a smaller but negative impact is made by the log trading volume, with its impact being reduced at the second lag. For the quoted volumes on the previous best level, we find a positive impact on the activity process. More specifically, the effect of the buying volume is slightly larger than the impact of the selling volume. Concerning the number of particles, it can be observed that increasing them over 1000 does not significantly affect the estimated parameters, while it dramatically affects the computational time linearly. Moreover, the execution time of drawing a sample of size 2000 with IBIS is almost equal than the corresponding time of drawing the double size with AM.

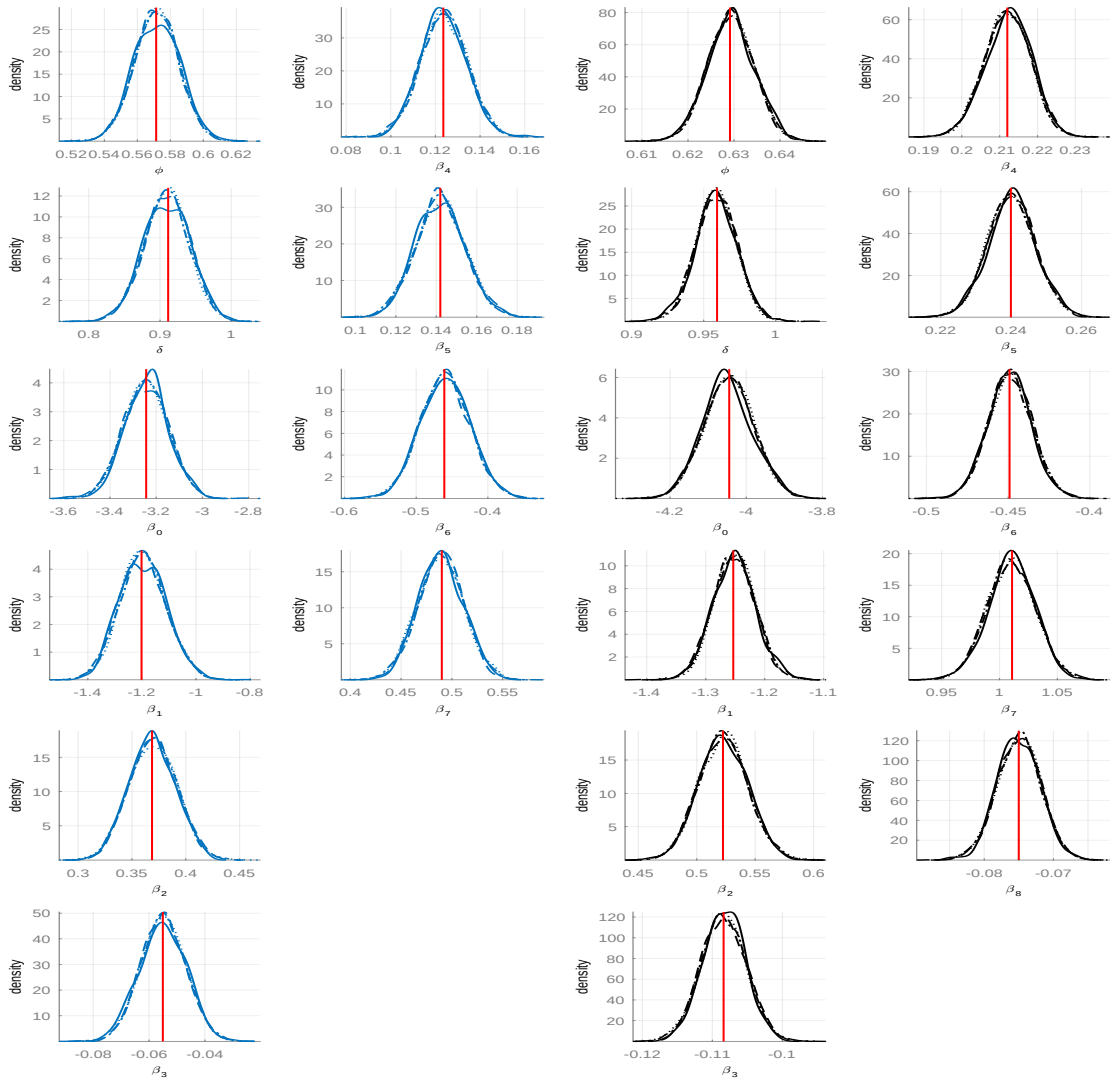


Model: A_i takes on value 1 (0) with probability π_i $(1-\pi_i)$, where $\text{logit}(\pi_i) = \boldsymbol{x}_{i-1}^\top \boldsymbol{\beta} + Z_i$, $Z_i = \phi Z_{i-1} + \delta\varepsilon_{i-1}$, $\varepsilon_i = (A_i - \pi_i)/\sqrt{\pi_i(1-\pi_i)}$, $Z_0 = \varepsilon_0 = 0$. Morning: $\boldsymbol{x}_{i-1}^\top = (1, A_{i-1}, \tau_{i-1}, V_{i-1}^{\text{mo}}, V_{i-1}^{\text{b,1}}, V_{i-1}^{\text{a,1}}, A_{i-2}, \tau_{i-2})$. Afternoon: $\boldsymbol{x}_{i-1}^\top = (1, A_{i-1}, \tau_{i-1}, V_{i-1}^{\text{mo}}, V_{i-1}^{\text{b,1}}, V_{i-1}^{\text{a,1}}, A_{i-2}, \tau_{i-2}, V_{i-2}^{\text{mo}})$.

Figure 3.7: Results of IBIS estimation for ES, May 16th to May 20th, during the morning (panels a, b) and afternoon period (panels c, d), case (II). ESS along the iterations (first column), and acceptance rate at each move step (second column). The number of particles is 1000 and a resample-move step is triggered when ESS drops below 800 (red line).

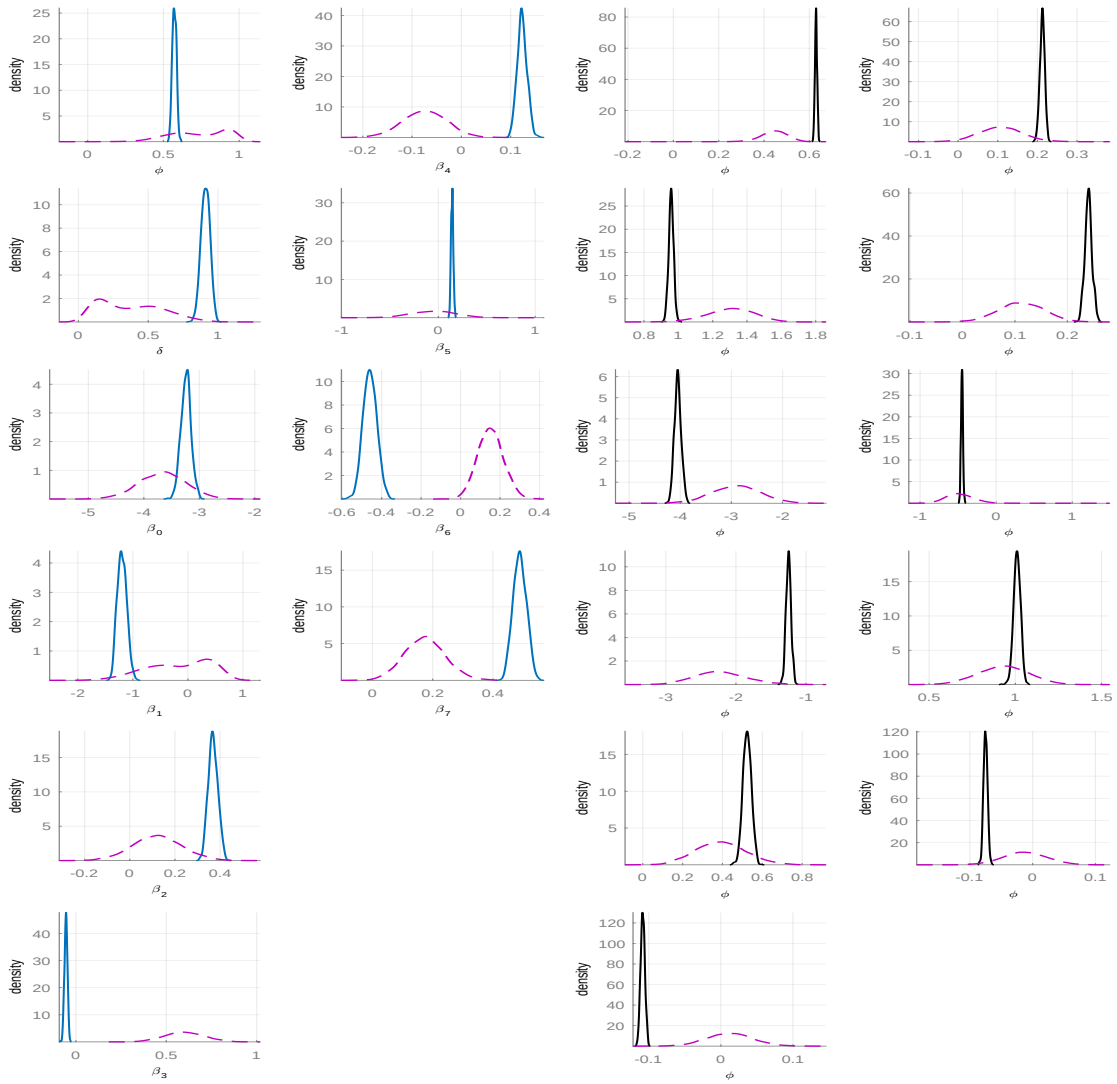
Figure 3.7 shows the ESS along the iterations (left panels) as well as the acceptance rates of the move steps (right panels) with 1000 particles. Panels (a, b) of the figure show the results over for the morning period, while panels (c, d) for the afternoon. Since the ESS threshold is set at 80%, a resample-move is triggered when ESS goes below 800 (red line). The

frequency of the resample-move steps seems to decrease over time. Besides, the acceptance rate during the afternoon is quite high along the iterations. On the other hand, during the morning the acceptance rate is relative low in the beginning, increases steeply to 80%, and continues to increase gradually to approximately 90% with some sudden drops with the lowest being just below 30%. Similar conclusions are drawn from the corresponding plots with 2000 and



Model: A_i takes on value 1 (0) with probability π_i $(1-\pi_i)$, where $\text{logit}(\pi_i) = \boldsymbol{x}_{i-1}^\top \boldsymbol{\beta} + Z_i$, $Z_i = \phi Z_{i-1} + \delta\varepsilon_{i-1}$, $\varepsilon_i = (A_i - \pi_i)/\sqrt{\pi_i(1-\pi_i)}$, $Z_0 = \varepsilon_0 = 0$. Morning: $\boldsymbol{x}_{i-1}^\top = (1, A_{i-1}, \tau_{i-1}, V_{i-1}^{\text{mo}}, V_{i-1}^{\text{b,l}}, V_{i-1}^{\text{a,l}}, A_{i-2}, \tau_{i-2})$. Afternoon: $\boldsymbol{x}_{i-1}^\top = (1, A_{i-1}, \tau_{i-1}, V_{i-1}^{\text{mo}}, V_{i-1}^{\text{b,l}}, V_{i-1}^{\text{a,l}}, A_{i-2}, \tau_{i-2}, V_{i-2}^{\text{mo}})$. Priors: $(\phi+1)/2 \sim \mathcal{B}e(0.5, 0.5)$, $\log(\delta) \sim \mathcal{N}(0, 10^3)$, $\boldsymbol{\beta} \sim \mathcal{N}_d(\mathbf{0}_d, 10^3\mathbf{I}_d)$.

Figure 3.8: Marginal posterior densities of the estimated parameters for the activity model, estimated from AM (dotted curve) and IBIS (solid, dashed and dash-dotted curve for 1000, 2000 and 5000 particles, respectively) samples using a kernel density. Vertical lines represent the mle. The data set we are analyzing is ES, May 16th to May 20th during the morning (blue curve) and afternoon (black curve) period, case (II).



Model: A_i takes on value 1 (0) with probability π_i $(1-\pi_i)$, where $\text{logit}(\pi_i) = \boldsymbol{x}_{i-1}^\top \boldsymbol{\beta} + Z_i$, $Z_i = \phi Z_{i-1} + \delta\varepsilon_{i-1}$, $\varepsilon_i = (A_i - \pi_i)/\sqrt{\pi_i(1-\pi_i)}$, $Z_0 = \varepsilon_0 = 0$. Morning: $\boldsymbol{x}_{i-1}^\top = (1, A_{i-1}, \tau_{i-1}, V_{i-1}^{\text{mo}}, V_{i-1}^{\text{b,l}}, V_{i-1}^{\text{a,l}}, A_{i-2}, \tau_{i-2})$. Afternoon: $\boldsymbol{x}_{i-1}^\top = (1, A_{i-1}, \tau_{i-1}, V_{i-1}^{\text{mo}}, V_{i-1}^{\text{b,l}}, V_{i-1}^{\text{a,l}}, A_{i-2}, \tau_{i-2}, V_{i-2}^{\text{mo}})$. Priors: $(\phi+1)/2 \sim \mathcal{B}e(0.5, 0.5)$, $\log(\delta) \sim \mathcal{N}(0, 10^3)$, $\boldsymbol{\beta} \sim \mathcal{N}_d(\mathbf{0}_d, 10^3\mathbf{I}_d)$.

Figure 3.9: Comparison of kernel density estimation of marginal posterior densities based on the whole dataset (solid line) and the first 3000 observations (purple curve) with 1000 IBIS and MCMC samples, respectively. For inference is used ES, May 16th to May 20th, during the morning (blue curve) and afternoon (black curve) partition, case (II).

5000 particles, although the graphs are not presented here. Figures 3.8 illustrates the marginal posterior densities; the vertical lines correspond to maximum likelihood estimators. It can be seen that the two methods produce nearly identical posterior distributions as well as the posterior mode corresponds to the maximum likelihood estimate. Furthermore, the impact of increasing the number of particles is dramatically insignificant. Figures 3.9 compares the sampling density

that generates the initial particles for IBIS (purple curve) with the target posterior density based on the whole dataset using 1000 IBIS samples. Not surprisingly, IBIS initial densities are wider than that of the target. This tightening is expected as additional points become available.

3.5.3 Part B

This section provides results for modelling the price direction of ES. The empirical analysis is done on two separated datasets: morning and afternoon, between 16 May and 20 May with 10,524 and 69,324 observations, respectively. For the process of regressors we consider two cases:

- *Case I*: $\boldsymbol{x}_{i-1}^{\mathrm{D},\top} = (1, D'_{ii}, A_{ii})$.
- *Case II:* $\boldsymbol{x}_{i-1}^{\mathrm{D},\top} = (1, D'_{ii}, A_{ii}, \tau_{ii}, \mathrm{BMO}_{ii}, V_{ii}^{\mathrm{mo}}, SP_{ii}, V_{ii}^{\mathrm{b},1}, V_{ii}^{\mathrm{b},2}, V_{ii}^{\mathrm{a},1}, V_{ii}^{\mathrm{a},2}, G_{ii}^{\mathrm{b}}, G_{ii}^{\mathrm{a}})$,

where $ii \in \{i-2, i-1\}$ refers to trading time. For example, V_{i-1}^{mo} denotes the lag-1 log volume of the previous active transaction. We have considered the same covariate vector restricted only in trading time. In this case, for example, V_{i-1}^{mo} denotes the lag-1 log volume of the previous active or not transaction. However, the predictive log likelihood was considerably lower in the trading time than in the activity time. The results for the covariate vector restricted on trading time is not presented here.

3.5.3.1 Case I

It turns out that the inclusion of past information of the activity process do not affect the probability of an upward price movement. Concerning the direction of a non-zero price movement, only its lag-1 value is statistically significant and become insignificant for higher lags. Therefore, the final covariate vector is $\boldsymbol{x}_{i-1}^{\mathrm{D},\top} = (1, D'_{i-1})$. Concerning the binary GLARMA, the parameters ϕ and δ do not affect the probability of an upward price movement. Concerning the autologistic model, theAM algorithm runs 50,000 iterations with burn-in of 10,000 iterations and the autocorrelation is reduced by retaining only every 20th iteration of the chain. The acceptance ratio is 22.10% (18.72%). For IBIS, we use the first 1000 (3000) observations to initialize the algorithm for both models.

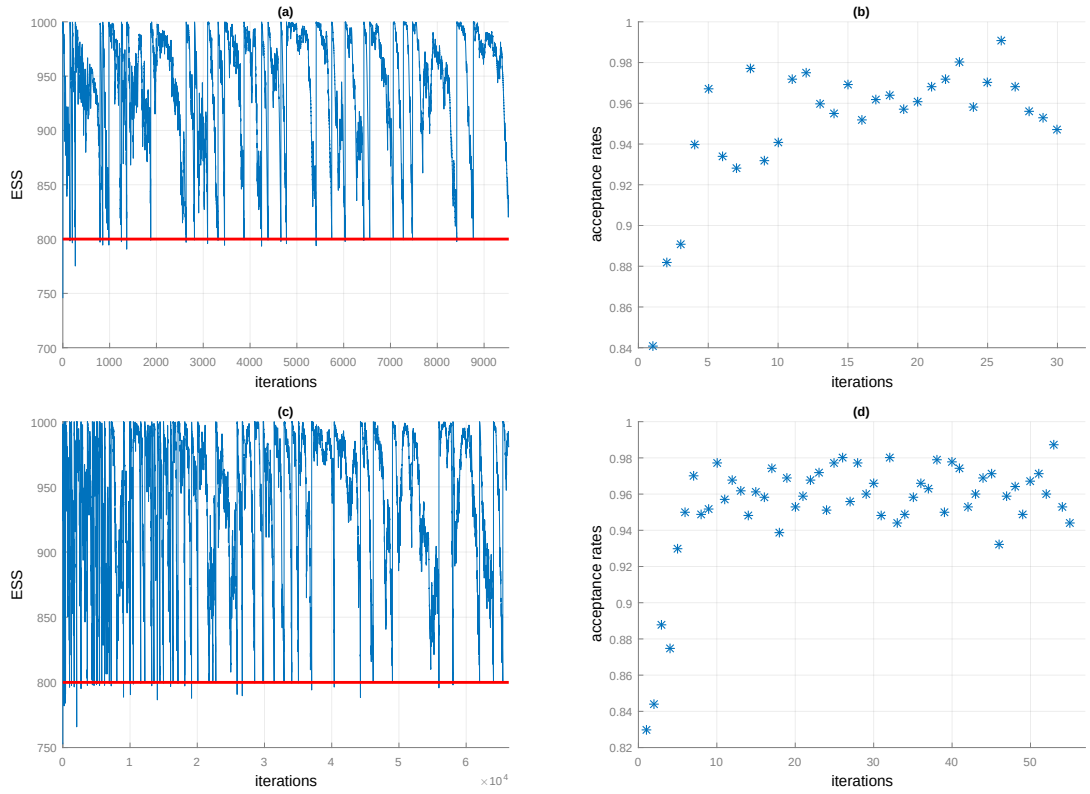
Table 3.6 lists the parameter estimates for the direction of active trade using an autologistic model, with the three estimation methods along with the running time (in minutes).

Table 3.6: Maximum likelihood, AM and IBIS estimates of parameters for the direction of active trade using an autologistic model, case (I). The results are reported separately for two sub-periods: morning and afternoon. The Bayesian estimates represent posterior means and standard deviations in parenthesis. For the maximum likelihood in parenthesis is reported the asymptotic standard errors. 'Run-time' returns the execution time in minutes. Model: D_i' takes on value 1 (0) with probability π_i $(1-\pi_i)$, where $\text{logit}(\pi_i) = \beta_0 + \beta_1 D_{i-1}'$. Prior: $\boldsymbol{\beta} \sim \mathcal{N}_2(\mathbf{0}_2, 10^3\mathbf{I}_2)$.

Parameter	MLE	AM	IBIS		
			Number of Particles		
			1000	2000	5000
Morning period					
β_0	1.578	1.578	1.576	1.579	1.578
	(0.037)	(0.035)	(0.035)	(0.038)	(0.036)
β_1	-3.191	-3.191	-3.191	-3.191	-3.190
	(0.052)	(0.052)	(0.048)	(0.052)	(0.052)
Run-time	0.000	0.286	1.057	2.035	5.112
Afternoon period					
β_0	2.383	2.383	2.382	2.382	2.382
	(0.019)	(0.019)	(0.020)	(0.019)	(0.019)
β_1	-4.769	-4.770	-4.768	-4.768	-4.769
	(0.027)	(0.027)	(0.028)	(0.027)	(0.028)
Run-time	0.001	1.178	7.861	14.518	34.433

The overall impression is that AM and IBIS sample estimates agree well with the maximum likelihood estimates. The table shows that all estimated coefficients are slightly larger during the afternoon compared to the morning period, while the corresponding standard errors are smaller. The direction variables are negative, which suggests that the next active trade is less likely to move upward if the previous trade moved upward. It can be observed, that the number of particles does not significantly affect the estimated parameters. However, since the computational cost grows with the number of particles, it is suggested to use 1000 particles in the filter. Moreover, comparing the run time of AM and IBIS to generate 2000 draws, the first method is approximately two times faster.

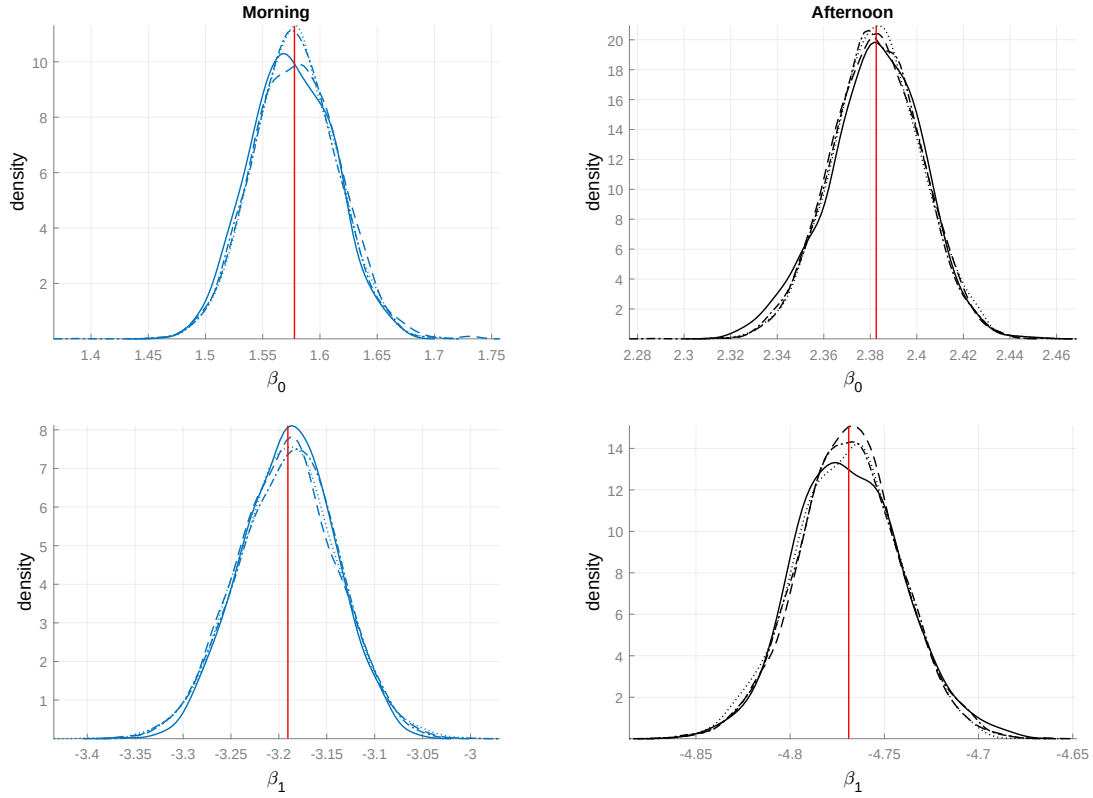
Figure 3.10 shows the ESS along the iterations (left plots) as well as the acceptance rates of the move steps (right plots) with 1000 particles. Panels (a, b) of the figure show the results over for the morning period, while panels (c, d) for the afternoon. Since the resampling



Model: D'_i takes on value 1 (0) with probability π_i $(1 - \pi_i)$, where $\text{logit}(\pi_i) = \beta_0 + \beta_1 D'_{i-1}$. Prior: $\boldsymbol{\beta} \sim \mathcal{N}_2(\mathbf{0}_2, 10^3 \mathbf{I}_2)$.

Figure 3.10: Results of IBIS estimation for the direction of active trade using an autologistic model during the morning (panels a, c) and afternoon period (panels d, dh), case (I). ESS along the iterations (first column), and acceptance rate at each move step (second column). The number of particles is 1000 and a resample-move step is triggered when ESS drops below 800 (red line).

threshold is set to 80%, a resample-move is triggered when ESS drops below 800 (red line). As expected, the frequency of the resample-moves steps seems to decrease over time and the acceptance rate is quite high along the iterations. Similar results are obtained for the other particles of the corresponding plots; these figures are not presented here. Figure 3.11 illustrates the marginal posterior densities; the two methods produce nearly identical posterior distributions. Note that by increasing the number of particles we obtain relatively similar estimates. The figure also marks the maximum likelihood estimators; it seems that they coincide with the posterior mode. In Figure 3.12, the wider density represents the sampling density that generates the initial particles for IBIS, and the shifted and tighter curve shows the target posterior density based on the whole dataset.

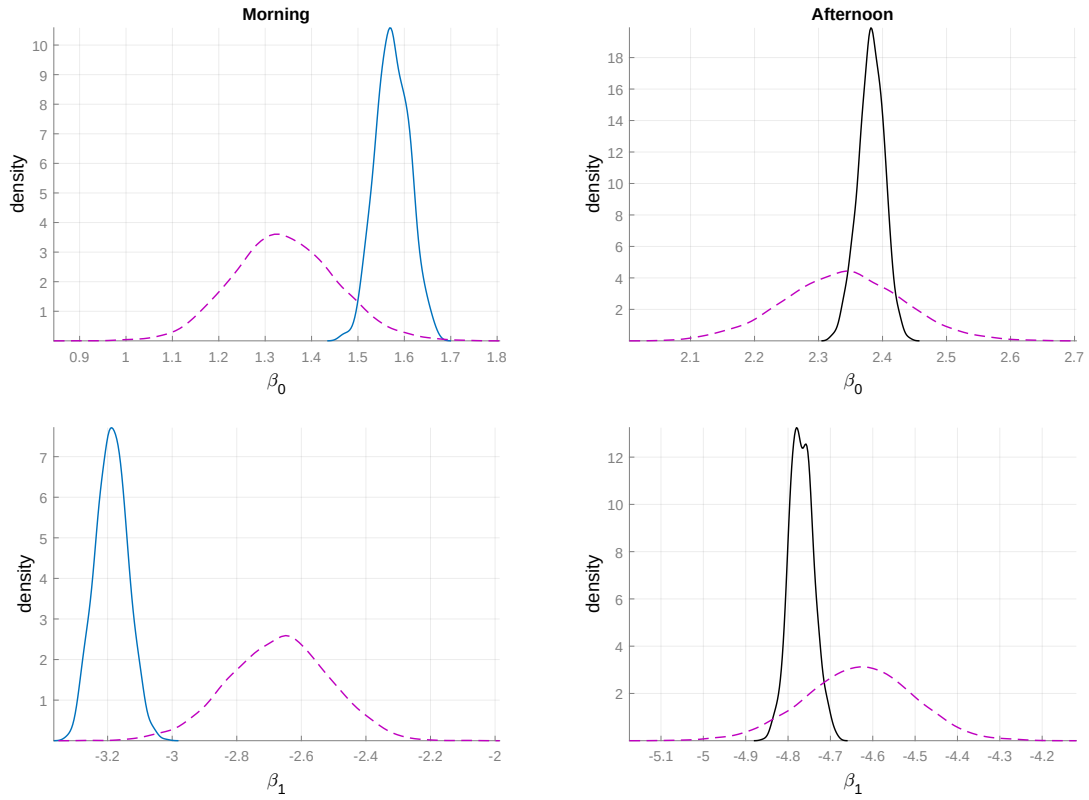


Autologistic model: D_i' takes on value 1 (0) with probability π_i $(1-\pi_i)$. $\text{logit}(\pi_i) = \beta_0 + \beta_1 D_{i-1}'$, Prior: $\boldsymbol{\beta} \sim \mathcal{N}_2(\mathbf{0}_2, 10^3\mathbf{I}_2)$.

Figure 3.11: Marginal posterior densities of the estimated parameters for the direction of active trade using an autologistic model, estimated from AM (dotted curve) and IBIS (solid, dashed and dash-dotted curve for 1000, 2000 and 5000 particles, respectively) samples using a kernel density. Vertical lines represent the mle, confined to the morning partition (blue curve) and afternoon partition (black curve), case (I).

3.5.3.2 Case II

It turns out that the inclusion of past information of the activity process as well as the lagged duration do not affect the probability of an upward price movement. Concerning the direction of a non-zero price movement, only its lag-1 value is statistically significant and become insignificant for higher lags. The best bid and ask limit gap price, the past log trading volume and spread do not affect the probability of having a upward price change. Besides, the type of lagged market order (buy or sell), only the lag-1 volumes of the first level are is significant and become insignificant for higher lags or levels. Finally, the lag-2 log bid and ask volume of the first level is significant only for the afternoon subset, while the lag-1 bid (ask) volume of the second (first) level is only statistically significant for the morning (afternoon) subset. After testing out insignificant explanatory variables we end up with the final covariate



Model: D'_i takes on value 1 (0) with probability π_i $(1-\pi_i)$, where $\text{logit}(\pi_i) = \beta_0 + \beta_1 D'_{i-1}$, Prior: $\boldsymbol{\beta} \sim \mathcal{N}_2(\mathbf{0}_2, 10^3\mathbf{I}_2)$.

Figure 3.12: Comparison of kernel density estimation of marginal posterior densities for the direction of active trade using an autologistic model based on the whole dataset (solid line) and the first 3000 observations (purple curve) with 1000 IBIS and MCMC samples, respectively, during the morning (blue curve) and afternoon period (black curve), case (I).

vector $\boldsymbol{x}_{i-1}^\top = (1, D'_{i-1}, \text{BMO}_{i-1}, V_{i-1}^{\text{b},1}, V_{i-1}^{\text{b},2}, V_{i-1}^{\text{a},2}, V_{i-2}^{\text{b},1}, V_{i-2}^{\text{b},2}, V_{i-2}^{\text{a},1}, V_{i-2}^{\text{a},2})$ for the morning and $\boldsymbol{x}_{i-1}^\top = (1, D'_{i-1}, \text{BMO}_{i-1}, V_{i-1}^{\text{b},1}, V_{i-1}^{\text{a},1}, V_{i-2}^{\text{b},1}, V_{i-2}^{\text{a},1})$ for the afternoon.

Concerning the binary GLARMA, the parameters ϕ and δ do not affect the probability of an upward price movement. Concerning the autologistic model, the AM algorithm runs 300,000 (200,000) iterations with burn-in of 20,000 iterations and the autocorrelation is reduced by retaining only every 150th (100th) iteration of the chain. The acceptance ratio is 21.95% (25.94%) Concerning the autologistic model, theAM algorithm runs 490,000 iterations with burn-in of 50,000 iterations and the autocorrelation is reduced by retaining only every 220th iteration of the chain. The acceptance ratio is 26.16% (28.68%). For IBIS, we use the first 1000 observations to initialize the algorithm for both models.

Tables 3.7 - 3.8 list the parameter estimates for the direction of active trade using the autologistic model, with the three estimation methods along with the running time (in minutes).

Table 3.7: Maximum likelihood, AM and IBIS estimates of parameters for the direction of active trade using an autologistic model, during the morning period, case (II). The Bayesian estimates represent posterior means and standard deviations in parenthesis. For the maximum likelihood in parenthesis is reported the asymptotic standard errors. 'Run-time' returns the execution time in minutes. The AM algorithm runs 300,000 iterations with a burn-in period of 20,000 draws, thinning every 150th iteration, yielding 2000 draws; the algorithm is initialized with the mle. For IBIS, the initial particles are sampled from the posterior density based on the first 1000 observations. The tick size is \$0.25. Model: D_i' takes on value 1 (0) with probability π_i $(1-\pi_i)$, where $\text{logit}(\pi_i) = \boldsymbol{x}_{i-1}^\top \boldsymbol{\beta}$, $\boldsymbol{x}_{i-1}^\top = (1, D_{i-1}', \text{BMO}_{i-1}, V_{i-1}^{\text{b},1}, V_{i-1}^{\text{b},2}, V_{i-1}^{\text{a},2}, V_{i-2}^{\text{b},1}, V_{i-2}^{\text{b},2}, V_{i-2}^{\text{a},1}, V_{i-2}^{\text{a},2})$.

Parameter	MLE	AM	IBIS		
			Number of Particles		
			1000	2000	5000
β_0	4.088	4.108	4.094	4.075	4.084
	(0.624)	(0.609)	(0.634)	(0.644)	(0.621)
β_1	-1.184	-1.182	-1.189	-1.184	-1.183
	(0.111)	(0.112)	(0.110)	(0.113)	(0.114)
β_2	-6.069	-6.083	-6.074	-6.083	-6.080
	(0.132)	(0.133)	(0.127)	(0.129)	(0.131)
β_3	-0.861	-0.864	-0.862	-0.862	-0.863
	(0.045)	(0.045)	(0.043)	(0.043)	(0.045)
β_4	-3.279	-3.292	-3.290	-3.290	-3.294
	(0.183)	(0.181)	(0.181)	(0.178)	(0.182)
β_5	3.309	3.316	3.302	3.316	3.319
	(0.196)	(0.189)	(0.188)	(0.195)	(0.197)
β_6	0.704	0.705	0.703	0.706	0.706
	(0.036)	(0.036)	(0.036)	(0.036)	(0.037)
β_7	2.651	2.666	2.668	2.662	2.668
	(0.173)	(0.172)	(0.174)	(0.174)	(0.174)
β_8	-0.443	-0.444	-0.445	-0.444	-0.443
	(0.041)	(0.040)	(0.041)	(0.041)	(0.041)
β_9	-2.263	-2.273	-2.260	-2.269	-2.273
	(0.193)	(0.189)	(0.190)	(0.193)	(0.192)
Run-time	0.028	117.665	90.234	172.297	418.103

The overall impression is that AM and IBIS sample estimates agree well with the maximum likelihood estimates. Based on the tables the following findings can be summarized: The influence of lagged price direction, is negative indicating that if the price moved on the last trade then there

is a large chance that this movement will be reversed if there is an active trade. Concerning the impact of buy market orders, we observe a significant negative influence at lag one, hence the odds of an up movement are larger for sell market orders than buy market order at lag one. Bid (ask) volume at lag one reduces (increases) the chance that the price movement will be upward, while the reverse is true at the second lag. It can be observed, that the number of particles does not significantly affect the estimated parameters. However, since the computational cost grows with the number of particles, it is suggested to use 1000 particles in the filter. Moreover, comparing the run time of AM and IBIS to generate 2000 draws, the first method is approximately faster. Figure 3.13 shows the ESS along the iterations (left panels) as well as the acceptance rates of the move steps (right panels) with 1000 particles using an autologistic model. Panels (a, c) of the figure show the results over for the morning period, while panels (c, d) for the afternoon. Since the ESS threshold is set at 80%, a resample-move is triggered when ESS goes below

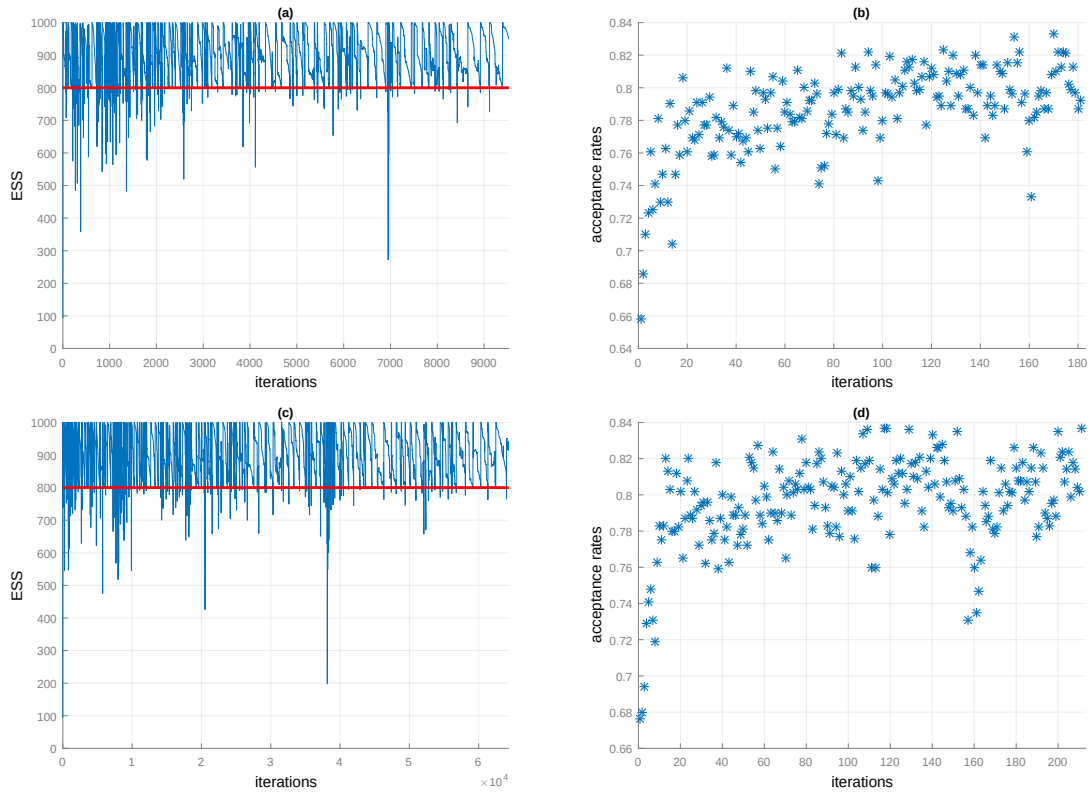


Figure 3.13: Results of IBIS estimation for the direction of active trade using an autologistic model during the morning (panels a, c) and afternoon period (panels c, d), case (II). ESS along the iterations (first column), and acceptance rate at each move step (second column). The number of particles is 1000 and a resample-move step is triggered when ESS drops below 800 (red line).

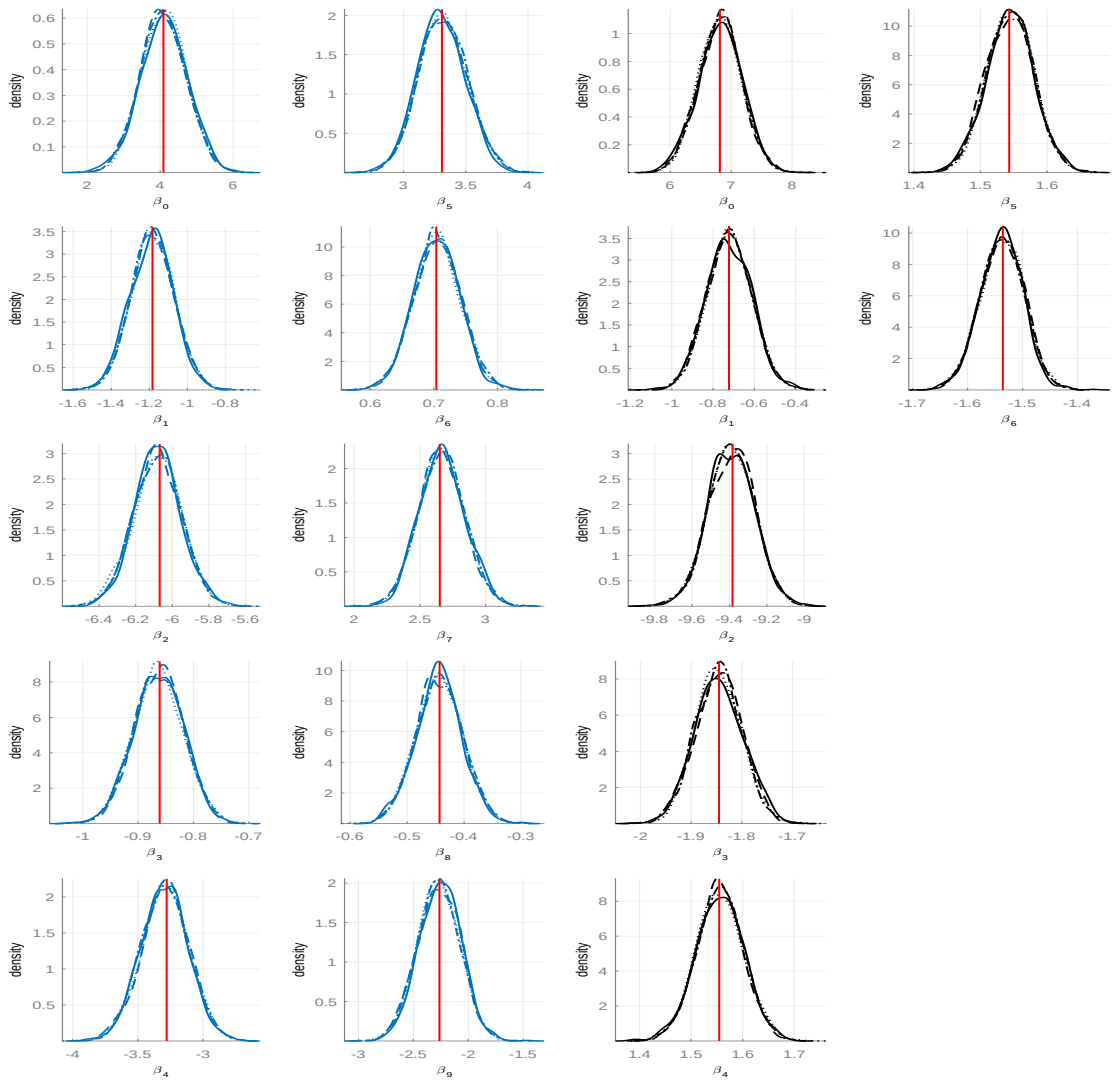
800 (red line). The frequency of the resample-move steps seems to decrease over time. Similar

Table 3.8: Maximum likelihood, AM and IBIS estimates of parameters for the direction of active trade using an autologistic model, during the afternoon period, case (II). The Bayesian estimates represent posterior means and standard deviations in parenthesis. For the maximum likelihood in parenthesis is reported the asymptotic standard errors. 'Run-time' returns the execution time in minutes. The AM algorithm runs 200,000 iterations with a burn-in period of 20,000 draws, thinning every 100th iteration, yielding 2000 draws; the algorithm is initialized with the mle. For IBIS, the initial particles are sampled from the posterior density based on the first 1000 observations. The tick size is \$0.25. Model: D_i' takes on value 1 (0) with probability π_i $(1-\pi_i)$, where $\text{logit}(\pi_i) = \boldsymbol{x}_{i-1}^\top \boldsymbol{\beta}$, $\boldsymbol{x}_{i-1}^\top = (1, D_{i-1}', \text{BMO}_{i-1}, V_{i-1}^{\text{b,l}}, V_{i-1}^{\text{a,l}}, V_{i-2}^{\text{b,l}}, V_{i-2}^{\text{a,l}})$.

Parameter	MLE	AM	IBIS		
			Number of Particles		
			1000	2000	5000
β_0	6.813	6.812	6.815	6.820	6.819
	(0.353)	(0.346)	(0.362)	(0.351)	(0.354)
β_1	-0.723	-0.720	-0.720	-0.724	-0.722
	(0.107)	(0.107)	(0.107)	(0.107)	(0.108)
β_2	-9.384	-9.391	-9.389	-9.390	-9.391
	(0.122)	(0.122)	(0.120)	(0.124)	(0.123)
β_3	-1.845	-1.846	-1.844	-1.846	-1.846
	(0.046)	(0.045)	(0.048)	(0.046)	(0.046)
β_4	1.555	1.556	1.554	1.555	1.557
	(0.045)	(0.046)	(0.046)	(0.043)	(0.044)
β_5	1.543	1.545	1.545	1.544	1.544
	(0.035)	(0.034)	(0.035)	(0.035)	(0.035)
β_6	-1.536	-1.537	-1.537	-1.537	-1.538
	(0.040)	(0.040)	(0.039)	(0.039)	(0.039)
Run-time	0.002	5.726	9.950	17.891	41.280

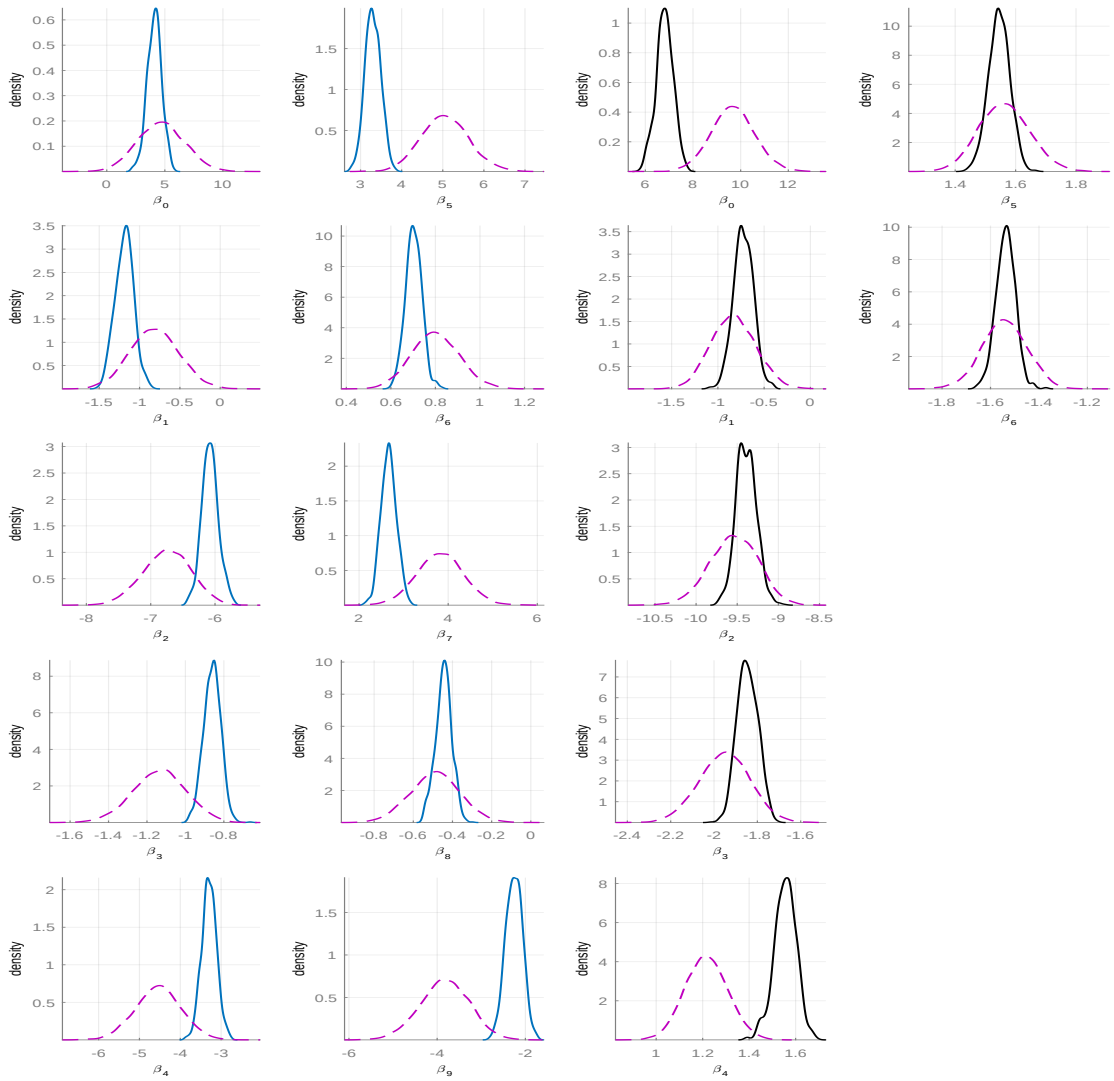
conclusions are drawn from the corresponding plots with 2000 and 5000 particles, although the graphs are not presented here. Figure 3.14 illustrates the marginal posterior densities of the parameters using the autologistic model; the vertical lines correspond to maximum likelihood estimators. It can be seen that the two methods produce nearly identical posterior distributions as well as the posterior mode corresponds to the maximum likelihood estimate. Furthermore, the impact of increasing the number of particles is dramatically insignificant. Figure 3.15 compares the sampling density that generates the initial particles for IBIS (purple curve) with the target posterior density based on the whole dataset using 1000 (3000) IBIS samples during the

morning (afternoon) period. Not surprisingly, IBIS initial densities are wider than that of the target. This tightening is expected as additional points become available.



Model: D_i' takes on value 1 (0) with probability π_i $(1-\pi_i)$, where $\text{logit}(\pi_i) = \boldsymbol{x}_{i-1}^\top \boldsymbol{\beta}$. Morning: $\boldsymbol{x}_{i-1}^\top = (1, D_{i-1}', \text{BMO}_{i-1}, V_{i-1}^{\text{b},1}, V_{i-1}^{\text{b},2}, V_{i-1}^{\text{a},2}, V_{i-2}^{\text{b},1}, V_{i-2}^{\text{b},2}, V_{i-2}^{\text{a},1}, V_{i-2}^{\text{a},2})$. Afternoon: $\boldsymbol{x}_{i-1}^\top = (1, D_{i-1}', \text{BMO}_{i-1}, V_{i-1}^{\text{b},1}, V_{i-1}^{\text{a},1}, V_{i-2}^{\text{b},1}, V_{i-2}^{\text{a},1})$. Prior: $\boldsymbol{\beta} \sim \mathcal{N}_d(\mathbf{0}_d, 10^3 \mathbf{I}_d)$.

Figure 3.14: Marginal posterior densities of the estimated parameters for the direction of active trade using an autologistic model, estimated from AM (dotted curve) and IBIS (solid, dashed and dash-dotted curve for 1000, 2000 and 5000 particles, respectively) samples using a kernel density. Vertical lines represent the mle, confined to the morning (blue curve) and afternoon (black curve) partition, case (II).



Model: D_i' takes on value 1 (0) with probability π_i $(1-\pi_i)$, where $\text{logit}(\pi_i) = \boldsymbol{x}_{i-1}^\top \boldsymbol{\beta}$. Morning: $\boldsymbol{x}_{i-1}^\top = (1, D_{i-1}', \text{BMO}_{i-1}, V_{i-1}^{\text{b},1}, V_{i-1}^{\text{b},2}, V_{i-1}^{\text{a},2}, V_{i-2}^{\text{b},1}, V_{i-2}^{\text{b},2}, V_{i-2}^{\text{a},1}, V_{i-2}^{\text{a},2})$. Afternoon: $\boldsymbol{x}_{i-1}^\top = (1, D_{i-1}', \text{BMO}_{i-1}, V_{i-1}^{\text{b},1}, V_{i-1}^{\text{a},1}, V_{i-2}^{\text{b},1}, V_{i-2}^{\text{a},1})$. Prior: $\boldsymbol{\beta} \sim \mathcal{N}_d(\mathbf{0}_d, 10^3 \mathbf{I}_d)$.

Figure 3.15: Comparison of kernel density estimation of marginal posterior densities for the direction of active trade using an autologistic model based on the whole dataset (solid line) and the first 3000 observations (purple curve) with 1000 IBIS and MCMC samples, respectively, during the morning (blue curve) and afternoon period (black curve), case (I).

Chapter 4

The Bernoulli parameter driven AD model

4.1 Introduction

In this chapter, we extend the AD model, which is an observation driven model for analyzing high-frequency integer price changes, and propose a state space model within a Bayesian framework. Given the use of logistic regression with a latent process for each component, we review some similar models applied on binary time series, but not necessarily applied on tick data. Wu and Cui [2014] consider an AR(1) latent process into the logit link function and propose a modified GLM estimation procedure and show the resulting estimator is consistent and asymptotically normal; they apply the model to boat race data. Dunsmuir and He [2016] focus on the development of methods for detecting serial dependence in time series of binomial counts in which the logit of the probability of success at each time point is a linear function of regression variables and a latent autocorrelated process. Dunsmuir and He [2017] consider a parameter driven model for binomial responses time series. The binary state space model with probit link, in a Bayesian framework, has been studied by Carlin et al. [1992]; Fahrmeir [1992]; Song [2000]; Abanto-Valle et al. [2015]; Czado and Song [2008], among others. In the last paper, the authors allow covariates (and binomial responses) and apply MCMC methods, while Abanto-Valle and Dey [2014] extend it using the usual links as well as an extension of them called power links. Fasano et al. [2019] prove that the filtering, predictive and smoothing distributions in dynamic probit models with Gaussian state variables are available and belong to a class of unified skew-normals whose parameters can be updated recursively in time via analytical expressions.

This chapter is organized as follows. Section 4.2 presents the parameter driven AD model

and Section 4.3 analyses the proposed model within a Bayesian framework via MCMC. Section 4.4 presents the results of a simulation study. Section 4.5 discusses the empirical results and evaluates the predictive performance of the proposed model.

4.2 The AR(1)-AD model

As explained in Section 3.2, Y_i can be written as in equation (3.1) with conditional joint distribution given in (3.2). Here we model the logit of the probability of success at each time point of each price factor as a linear function of regression variables and a latent process. In the following section we detail the modelling of the activity and direction process.

4.2.1 Models for the activity and direction process

Conditional on a latent process $\{\alpha_i^{\mathrm{A}}\}$, we assume that the observation A_i only depends on the current state and not on previous states or observations and is specified by a Bernoulli distribution with probability π_{A_i}. We parametrize π_{A_i} as a dynamic model of the form

$$\begin{aligned}\text{logit}(\pi_{A_i}) &= \boldsymbol{x}_{i-1}^{\mathrm{A},\top}\boldsymbol{\beta}_{\mathrm{A}} + \alpha_i^{\mathrm{A}} \\ \alpha_i^{\mathrm{A}} &= \phi_{\mathrm{A}}\alpha_{i-1}^{\mathrm{A}} + \varepsilon_i^{\mathrm{A}}, \;\; \varepsilon_i^{\mathrm{A}} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, 1/\tau_{\mathrm{A}})\end{aligned}$$

where $\alpha_1^{\mathrm{A}}|\phi_{\mathrm{A}}, \tau_{\mathrm{A}} \sim \mathcal{N}(0, 1/\tau_{\mathrm{A}}(1-\phi_{\mathrm{A}}^2))$, $|\phi_{\mathrm{A}}| < 1$ and $\tau_{\mathrm{A}} > 0$. The model can be written equivalently as

$$A_i|h_i^{\mathrm{A}} \sim \text{Bernoulli}(\pi_{A_i}) \tag{4.1}$$

$$\text{logit}(\pi_{A_i}) = h_i^{\mathrm{A}} \tag{4.2}$$

$$h_i^{\mathrm{A}} = \boldsymbol{x}_{i-1}^{\mathrm{A},\top}\boldsymbol{\beta}_{\mathrm{A}} + \phi_{\mathrm{A}}(h_{i-1}^{\mathrm{A}} - \boldsymbol{x}_{i-2}^{\mathrm{A},\top}\boldsymbol{\beta}_{\mathrm{A}}) + \varepsilon_i^{\mathrm{A}}, \tag{4.3}$$

where $\pi_{A_i} \triangleq \pi(A_i = 1|h_i^{\mathrm{A}})$ and $h_1^{\mathrm{A}}|\phi_{\mathrm{A}}, \tau_{\mathrm{A}}, \boldsymbol{\beta}_{\mathrm{A}} \sim \mathcal{N}(\boldsymbol{x}_0^{\mathrm{A},\top}\boldsymbol{\beta}_{\mathrm{A}}, 1/\tau_{\mathrm{A}}(1-\phi_{\mathrm{A}}^2))$. Equations 4.1 to 4.3 is called the AR(1) activity model in its centered form. If $\phi_{\mathrm{A}} = 0$, the model is called a WN activity model, while when $\phi_{\mathrm{A}} = 1$ it is called a RW(1) activity model where $h_1^{\mathrm{A}}|\tau_{\mathrm{A}}, \boldsymbol{\beta}_{\mathrm{A}} \sim \mathcal{N}(\boldsymbol{x}_0^{\mathrm{A},\top}\boldsymbol{\beta}_{\mathrm{A}}, 1/\tau_{\mathrm{A}})$. In a similar way, we parametrize π_{D_i} for the direction component. Figure 4.1 shows a graphical representation of the activity model. The shaded nodes indicate that the corresponding variable is observed, while all variables within the plate are repeated for all

values of the index i.

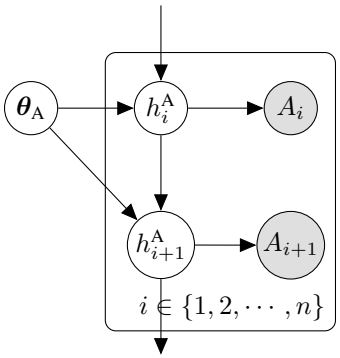


Figure 4.1: Directed acyclic graph corresponding to the AR(1) activity model. The shaded nodes indicate that the corresponding variable is observed. All variables within the plate are repeated for all values of the index i.

The conditional mean and variance of A_i is defined as $\mathbb{E}(A_i|h_i^{\mathrm{A}}) = \pi_{A_i} = \dfrac{\exp(h_i^{\mathrm{A}})}{1+\exp(h_i^{\mathrm{A}})}$ and $\mathbb{V}(A_i|h_i^{\mathrm{A}}) = \pi_{A_i}(1-\pi_{A_i})$. The likelihood function involves n intractable integrals and is given by

$$\mathcal{L}_C(\boldsymbol{\theta}_{\mathrm{A}}) \triangleq \int \prod_{i=1}^{n} \pi(\alpha_i|h_i^{\mathrm{A}})\pi(\boldsymbol{h}_{\mathrm{A}}|\boldsymbol{\theta}_{\mathrm{A}})\, d\boldsymbol{h}_{\mathrm{A}}, \tag{4.4}$$

where $\pi(\alpha_i|h_i^{\mathrm{A}}) \propto \exp(h_i^{\mathrm{A}}\alpha_i)(1+\exp(h_i^{\mathrm{A}}))^{-1}$ and the quantity $\pi(\boldsymbol{h}_{\mathrm{A}}|\boldsymbol{\theta}_{\mathrm{A}})$ is a multivariate normal density; $\boldsymbol{\theta}_{\mathrm{A}} \triangleq (\phi_{\mathrm{A}}, \tau_{\mathrm{A}}, \boldsymbol{\beta}_{\mathrm{A}})^\top$ collects the parameters of the price activity process, and $\boldsymbol{h}_{\mathrm{A}} \triangleq (h_1^{\mathrm{A}}, \cdots, h_n^{\mathrm{A}})^\top$.

As discussed in Section 2.5.2, different model parametrizations can perform better than others under certain conditions. The AR(1) activity model in its non-centered form satisfies

$$A_i|\tilde{h}_i^{\mathrm{A}}, \tau_{\mathrm{A}}, \boldsymbol{\beta}_{\mathrm{A}} \sim \mathrm{Bernoulli}(\pi_{A_i}) \tag{4.5}$$

$$\mathrm{logit}(\pi_{A_i}) = \boldsymbol{x}_{i-1}^{\mathrm{A},\top}\boldsymbol{\beta}_{\mathrm{A}} + \tilde{h}_i^{\mathrm{A}}/\sqrt{\tau_{\mathrm{A}}} \tag{4.6}$$

$$\tilde{h}_i^{\mathrm{A}} = \phi_{\mathrm{A}}\tilde{h}_{i-1}^{\mathrm{A}} + \varepsilon_i^{\mathrm{A}}, \;\; \varepsilon_i^{\mathrm{A}} \sim \mathcal{N}(0,1), \tag{4.7}$$

where $\pi_{A_i} \triangleq \pi(A_i = 1|\tilde{h}_i^{\mathrm{A}}, \tau_{\mathrm{A}}, \boldsymbol{\beta}_{\mathrm{A}})$ and $\tilde{h}_1^{\mathrm{A}}|\phi_{\mathrm{A}} \sim \mathcal{N}(0, 1/(1-\phi_{\mathrm{A}}^2))$. Note that $\tilde{h}_i^{\mathrm{A}} \triangleq \sqrt{\tau_{\mathrm{A}}}(h_i^{\mathrm{A}} - \boldsymbol{x}_{i-1}^{\mathrm{A},\top}\boldsymbol{\beta}_{\mathrm{A}})$, hence the two parametrizations are equivalent probabilistically. The conditional mean of A_i can be defined as $\mathbb{E}(A_i|\tilde{h}_i^{\mathrm{A}}, \tau_{\mathrm{A}}, \boldsymbol{\beta}_{\mathrm{A}}) = \pi_{A_i} = \dfrac{\exp(\boldsymbol{x}_{i-1}^{\mathrm{A},\top}\boldsymbol{\beta}_{\mathrm{A}} + \tilde{h}_i^{\mathrm{A}}/\sqrt{\tau_{\mathrm{A}}})}{1+\exp(\boldsymbol{x}_{i-1}^{\mathrm{A},\top}\boldsymbol{\beta}_{\mathrm{A}} + \tilde{h}_i^{\mathrm{A}}/\sqrt{\tau_{\mathrm{A}}})}$. The likelihood function is given by

$$\mathcal{L}_{NC}(\boldsymbol{\theta}_{\mathrm{A}}) \triangleq \int \prod_{i=1}^{n} \pi(\alpha_i|\tilde{h}_i^{\mathrm{A}}, \tau_{\mathrm{A}}, \boldsymbol{\beta}_{\mathrm{A}})\pi(\tilde{\boldsymbol{h}}_{\mathrm{A}}|\phi_{\mathrm{A}})\, d\tilde{\boldsymbol{h}}_{\mathrm{A}}, \tag{4.8}$$

where $\pi(\alpha_i|\tilde{h}_i^{\mathrm{A}}, \tau_{\mathrm{A}}, \boldsymbol{\beta}_{\mathrm{A}}) \propto \exp((\boldsymbol{x}_{i-1}^{\mathrm{A},\top}\boldsymbol{\beta}_{\mathrm{A}} + \tilde{h}_i^{\mathrm{A}}/\sqrt{\tau_{\mathrm{A}}})\alpha_i)(1+\exp(\boldsymbol{x}_{i-1}^{\mathrm{A},\top}\boldsymbol{\beta}_{\mathrm{A}} + \tilde{h}_i^{\mathrm{A}}/\sqrt{\tau_{\mathrm{A}}}))^{-1}$ and the quantity $\pi(\tilde{\boldsymbol{h}}_{\mathrm{A}}|\phi_{\mathrm{A}})$ is a multivariate normal density, where $\tilde{\boldsymbol{h}}_{\mathrm{A}} \triangleq (\tilde{h}_1^{\mathrm{A}}, \cdots, \tilde{h}_n^{\mathrm{A}})$. In a similar way, we parameterize π_{D_i} for the direction component.

4.3 Bayesian analysis of AR(1)-AD

Assuming independence among the parameters, the joint prior is $\pi(\boldsymbol{\theta}_{\mathrm{A}}, \boldsymbol{\theta}_{\mathrm{D}}) = \pi(\boldsymbol{\theta}_{\mathrm{A}})\pi(\boldsymbol{\theta}_{\mathrm{D}})$. The joint posterior distribution, $\pi(\boldsymbol{\theta}_{\mathrm{A}}, \boldsymbol{h}_{\mathrm{A}}, \boldsymbol{\theta}_{\mathrm{D}}, \boldsymbol{h}_{\mathrm{D}}|\boldsymbol{y})$ takes the form

$$\pi(\boldsymbol{\theta}_{\mathrm{A}}, \boldsymbol{h}_{\mathrm{A}}, \boldsymbol{\theta}_{\mathrm{D}}, \boldsymbol{h}_{\mathrm{D}}|\boldsymbol{y}) \propto \pi(\boldsymbol{\theta}_{\mathrm{A}}, \boldsymbol{h}_{\mathrm{A}}|\boldsymbol{\alpha})\pi(\boldsymbol{\theta}_{\mathrm{D}}, \boldsymbol{h}_{\mathrm{D}}|\boldsymbol{d}, \boldsymbol{\alpha} = \mathbf{1}_n), \tag{4.9}$$

where $\pi(\boldsymbol{\theta}_{\mathrm{A}}, \boldsymbol{h}_{\mathrm{A}}|\boldsymbol{\alpha})$ and $\pi(\boldsymbol{\theta}_{\mathrm{D}}, \boldsymbol{h}_{\mathrm{D}}|\boldsymbol{d}, \boldsymbol{\alpha} = \mathbf{1}_n)$ denote the posterior distribution of each component satisfying

$$\pi(\boldsymbol{\theta}_{\mathrm{A}}, \boldsymbol{h}_{\mathrm{A}}|\boldsymbol{\alpha}) \propto \pi(\boldsymbol{\alpha}|\boldsymbol{h}_{\mathrm{A}})\pi(\boldsymbol{h}_{\mathrm{A}}|\boldsymbol{\theta}_{\mathrm{A}})\pi(\boldsymbol{\theta}_{\mathrm{A}})$$
$$\pi(\boldsymbol{\theta}_{\mathrm{D}}, \boldsymbol{h}_{\mathrm{D}}|\boldsymbol{d}, \boldsymbol{\alpha} = \mathbf{1}_n) \propto \pi(\boldsymbol{d}|\boldsymbol{\alpha} = \mathbf{1}_n, \boldsymbol{h}_{\mathrm{D}})\pi(\boldsymbol{h}_{\mathrm{D}}|\boldsymbol{\theta}_{\mathrm{D}})\pi(\boldsymbol{\theta}_{\mathrm{D}}).$$

Furthermore, $\pi(\boldsymbol{h}_{\mathrm{A}}|\boldsymbol{\theta}_{\mathrm{A}}) \triangleq \mathcal{N}_n(\boldsymbol{m}_{\mathrm{A}}, \boldsymbol{Q}_{\mathrm{A}}^{-1})$, where $\boldsymbol{m}_{\mathrm{A}} \triangleq (\boldsymbol{x}_0^{\mathrm{A},\top}\boldsymbol{\beta}_{\mathrm{A}}, \cdots, \boldsymbol{x}_{n-1}^{\mathrm{A},\top}\boldsymbol{\beta}_{\mathrm{A}})\top$ $(n \times 1)$ and the precision matrix $\boldsymbol{Q}_{\mathrm{A}}$ $(n \times n)$ is a tridiagonal matrix in which the primary diagonal is formed by the elements $\{\tau_{\mathrm{A}}, (1+\phi_{\mathrm{A}})^2\tau_{\mathrm{A}}, \cdots, (1+\phi_{\mathrm{A}})^2\tau_{\mathrm{A}}, \tau_{\mathrm{A}}\}$, while the diagonal above the principal diagonal is a vector whose every element is equal to $-\phi_{\mathrm{A}}\tau_{\mathrm{A}}$. If $\phi_{\mathrm{A}} = 1$, replace the elements of the primary diagonal by the elements $\{2\tau_{\mathrm{A}}, \cdots, 2\tau_{\mathrm{A}}, \tau_{\mathrm{A}}\}$. If $\phi_{\mathrm{A}} = 0$, the precision matrix is a diagonal matrix with all diagonal elements equal to τ_{A}. In NCP, $\pi(\tilde{\boldsymbol{h}}_{\mathrm{A}}|\boldsymbol{\theta}_{\mathrm{A}}) \triangleq \mathcal{N}_n(\mathbf{0}_n, \tilde{\boldsymbol{Q}}_{\mathrm{A}}^{-1})$, where $\tilde{\boldsymbol{Q}}_{\mathrm{A}} \triangleq (1/\tau_{\mathrm{A}})\boldsymbol{Q}_{\mathrm{A}}$.

Equation 4.9 implies that instead of simulating directly from the target density, this can be achieved by simulating from the posterior distribution for each component of price change and by combining the outcome the final samples are from the desired density.

For both factors we choose the same prior distributions and all the samplers run on the unrestricted parameter space. For the parameter $\tau = \tau_{\mathrm{A}}$ or $\tau_{\mathrm{D}} > 0$, which denotes the conditional precision in the autoregressive sequence, we choose a gamma distribution with parameters γ_0 and δ_0 or, in symbols $\tau \sim \mathrm{Gamma}(\gamma_0, \delta_0)$ with probability density function

$$\pi(\tau) = \frac{1}{\delta_0^{\gamma_0}\Gamma(\gamma_0)}\tau^{\gamma_0-1}(-\tau/\delta_0), \quad \text{for } \tau > 0, \ \ \gamma_0, \ \ \delta_0 > 0,$$

where $\Gamma(\alpha)$ is the gamma function satisfying $\Gamma(\alpha) \triangleq (\alpha - 1)!$ for all positive integers. If not specified otherwise, $\gamma_0 = 1\mathrm{e}{-3}$ and $\delta_0 = 1\mathrm{e}{+3}$. The variable is transformed to the unrestricted variable $\eta \triangleq \log(\tau)$, with prior proportional to $\log \pi(\eta) \propto \eta\gamma_0 + (-1/\delta_0)(\eta)$. The prior of the parameters ϕ and $\boldsymbol{\beta}$ is discussed in Section 3.3. Independent of the prior choice, the partial posterior distributions of the parameters of the components do not have a closed form, and it is known up to a normalizing constant.

4.3.1 MCMC sampling

We construct a Markov chain which mimics the two-component Gibbs sampler to estimate the posterior distribution $\pi(\boldsymbol{h}_{\mathrm{A}}, \boldsymbol{\theta}_{\mathrm{A}}|\boldsymbol{\alpha}) \propto \pi(\boldsymbol{\alpha}|\boldsymbol{h}_{\mathrm{A}})\pi(\boldsymbol{h}_{\mathrm{A}}|\boldsymbol{\theta}_{\mathrm{A}})$; the procedure is summarized in Algorithm 8. For more details Section 2.5.

Algorithm 8 Draw N samples from $\pi(\boldsymbol{h}_{\mathrm{A}}, \boldsymbol{\theta}_{\mathrm{A}}|\boldsymbol{\alpha})$ using MCMC methods. varargin is a variable that enables the function to accept any number of input arguments

1: Initialize $\boldsymbol{h}_{\mathrm{A}}^{(0)}$, $\boldsymbol{\theta}_{\mathrm{A}}^{(0)} = (\phi_{\mathrm{A}}^{(0)}, \tau_{\mathrm{A}}^{(0)}, \boldsymbol{\beta}_{\mathrm{A}}^{(0)})$.
2: **for** $l = 1, 2, \cdots, N$ **do**
3: $\quad \boldsymbol{h}_{\mathrm{A}}^{(l)} \leftarrow$ UpdatePathC($\boldsymbol{h}_{\mathrm{A}}^{(l-1)}, \boldsymbol{\theta}_{\mathrm{A}}^{(l-1)}$, varargin)
4: $\quad \boldsymbol{\theta}_{\mathrm{A}}^{(l)} \leftarrow$ UpdateThetaC($\boldsymbol{h}_{\mathrm{A}}^{(l)}, \boldsymbol{\theta}_{\mathrm{A}}^{(l-1)}$, varargin)
5: **end for**
6: **return** $(\boldsymbol{h}_{\mathrm{A}}^{(1)}, \boldsymbol{\theta}_{\mathrm{A}}^{(1)}), \cdots, (\boldsymbol{h}_{\mathrm{A}}^{(N)}, \boldsymbol{\theta}_{\mathrm{A}}^{(N)})$

UpdatePathC and UpdatePathNC: We investigate two methods (see Section 2.5.1): Firstly, we sample the components of the latent state one at time in separate MH steps conditional on all others values of the state process and on the parameter vector ('1-1'), and secondly we update the whole process in one move ('all') focused on the aGrad-z (Titsias [2011]; Titsias and Papaspiliopoulos [2018]) algorithm. After some calculations, the following equations hold:

$$\log \pi(\alpha_i|h_i^{\mathrm{A}}) \propto h_i^{\mathrm{A}}\alpha_i - \log(1 + (h_i^{\mathrm{A}})),$$
$$\log \pi(\alpha_i|\tilde{h}_i^{\mathrm{A}}, \tau_{\mathrm{A}}, \boldsymbol{\beta}_{\mathrm{A}}) \propto (\boldsymbol{x}_{i-1}^{\mathrm{A},\top}\boldsymbol{\beta}_{\mathrm{A}} + \tilde{h}_i^{\mathrm{A}}/\sqrt{\tau_{\mathrm{A}}}) - \log(1 + (\boldsymbol{x}_{i-1}^{\mathrm{A},\top}\boldsymbol{\beta}_{\mathrm{A}} + \tilde{h}_i^{\mathrm{A}}/\sqrt{\tau_{\mathrm{A}}}))$$

and

$$\log \pi(h_1^{\mathrm{A}}|\boldsymbol{\theta}_{\mathrm{A}}) \propto -(\tau_{\mathrm{A}}/2)(1-\phi_{\mathrm{A}}^2)(h_1^{\mathrm{A}} - \boldsymbol{x}_0^{\mathrm{A},\top}\boldsymbol{\beta}_{\mathrm{A}})^2,$$
$$\log \pi(h_i^{\mathrm{A}}|h_{i-1}^{\mathrm{A}}, \boldsymbol{\theta}_{\mathrm{A}}) \propto -(\tau_{\mathrm{A}}/2)(h_i^{\mathrm{A}} - \boldsymbol{x}_{i-1}^{\mathrm{A},\top}\boldsymbol{\beta}_{\mathrm{A}} - \phi_{\mathrm{A}}(h_{i-1}^{\mathrm{A}} - \boldsymbol{x}_{i-2}^{\mathrm{A},\top}\boldsymbol{\beta}_{\mathrm{A}}))^2.$$

UpdateThetaC and UpdateThetaNC: We sample from $\pi(\boldsymbol{\theta}_{\mathrm{A}}|\boldsymbol{h}_{\mathrm{A}}, \boldsymbol{\alpha}) \propto \pi(\boldsymbol{\alpha}|\boldsymbol{h}_{\mathrm{A}})\,\pi(\boldsymbol{h}_{\mathrm{A}}|\boldsymbol{\theta}_{\mathrm{A}})\,\pi(\boldsymbol{\theta}_{\mathrm{A}})$ using a 3-block sampler, a 2-block sampler and the interweaving sampler (Yu and Meng [2011]; see Section 2.5.2). After some manipulations, the density $\pi(\boldsymbol{h}_{\mathrm{A}}|\boldsymbol{\theta}_{\mathrm{A}})$, by taking the logarithm, can be written as

$$\log \pi(\boldsymbol{h}_{\mathrm{A}}|\boldsymbol{\theta}_{\mathrm{A}}) \propto \frac{n}{2}\log(\tau_{\mathrm{A}}) + \frac{1}{2}\log(1-\phi_{\mathrm{A}}^2) - \frac{\tau_{\mathrm{A}}}{2}\sum_{i=1}^{n}(h_{*i}^{\mathrm{A}} - \boldsymbol{x}_{*i}^{\mathrm{A},\top}\boldsymbol{\beta}_{\mathrm{A}})^2,$$

where $\boldsymbol{h}_*^{\mathrm{A}} \triangleq (h_{*1}^{\mathrm{A}}, \cdots, h_{*n}^{\mathrm{A}})^\top (n \times 1)$ and $\boldsymbol{x}_*^{\mathrm{A}} \triangleq (\boldsymbol{x}_{*1}^{\mathrm{A}}, \cdots, \boldsymbol{x}_{*n}^{\mathrm{A}})(d_{\mathrm{A}} \times n)$ satisfying

$$h_{*i}^{\mathrm{A}} = \begin{cases} \sqrt{1-\phi_{\mathrm{A}}^2}h_i^{\mathrm{A}}, & i=1 \\ h_i^{\mathrm{A}} - \phi_{\mathrm{A}}h_{i-1}^{\mathrm{A}}, & i \geq 2 \end{cases} \text{ and } \boldsymbol{x}_{*i}^{\mathrm{A}} = \begin{cases} \sqrt{1-\phi_{\mathrm{A}}^2}\boldsymbol{x}_0^{\mathrm{A}}, & i=1 \\ \boldsymbol{x}_{i-1}^{\mathrm{A}} - \phi_{\mathrm{A}}\boldsymbol{x}_{i-2}^{\mathrm{A}}, & i \geq 2. \end{cases}$$

In the 3-block sampler, ϕ_{A}, τ_{A} and the block of $\boldsymbol{\beta}_{\mathrm{A}}$ are sampled separately. The parameters τ_{A} and $\boldsymbol{\beta}_{\mathrm{A}}$ are drawn with Gibbs from their corresponding conditional posterior given by

$$\tau_{\mathrm{A}}|\boldsymbol{h}_{\mathrm{A}}, \boldsymbol{\alpha}, \phi_{\mathrm{A}}, \boldsymbol{\beta}_{\mathrm{A}} \sim \mathrm{Gamma}\left(\gamma_0 + \frac{n}{2}, \frac{1}{G}\right), \;\; \boldsymbol{\beta}_{\mathrm{A}}|\boldsymbol{h}_{\mathrm{A}}, \boldsymbol{\alpha}, \phi_{\mathrm{A}}, \tau_{\mathrm{A}} \sim \mathcal{N}_{d_{\mathrm{A}}}\left(\boldsymbol{\mu}, \boldsymbol{\Sigma}^{-1}\right),$$

where $G \triangleq \dfrac{1}{\delta_0} + \dfrac{1}{2}\displaystyle\sum_{i=1}^{n}(h_{*i}^{\mathrm{A}} - \boldsymbol{x}_{*i}^{\mathrm{A},\top}\boldsymbol{\beta}_{\mathrm{A}})^2$, $\boldsymbol{\mu} \triangleq \tau_{\mathrm{A}}\boldsymbol{\Sigma}^{-1}\boldsymbol{x}_*^{\mathrm{A}}\boldsymbol{h}_*^{\mathrm{A}}$ and $\boldsymbol{\Sigma} \triangleq \dfrac{1}{\sigma_0^2}\mathbb{I}_{d_{\mathrm{A}}} + \tau_{\mathrm{A}}\boldsymbol{x}_*^{\mathrm{A}}\boldsymbol{x}_*^{\mathrm{A},\top}$. The parameter ϕ_{A} is drawn from its conditional posterior density by utilizing the random walk Metropolis algorithm with a Gaussian proposal with variance δ, where δ is a tuning constant. In NCP, due to non-conjugacy of the chosen priors, all parameters are updated with the Gaussian random walk Metropolis algorithm. In both parametrizations, $\boldsymbol{\beta}_{\mathrm{A}}$ is drawn jointly in one block.

In the 2-block sampler, ϕ_{A} and τ_{A} are sampled jointly, and then the elements within $\boldsymbol{\beta}_{\mathrm{A}}$ are updated simultaneously in a different step. The conditional posterior density of ϕ_{A} and τ_{A} is not known in closed form, thus it is approximated with a Gaussian random walk Metropolis algorithm. Each tuning parameter has to be adjusted to reach an acceptance rate around to 44% for univariate parameters and around to 23.4% for multivariate parameters. We follow the

idea of the adaptive random walk Metropolis-within-Gibbs (Roberts and Rosenthal [2009], see Section 2.3.2.2) which automatically tune the Markov chain parameters during the run.

4.3.2 Predictive Performance

We conduct a recursive out-of-sample forecasting procedure, using predictive likelihoods. The general estimation procedure is similar with its corresponding observation driven model that described in Section 3.3.1.

At the end of time t_n, the existing samples $\{h_n^{\mathrm{A},(l)}\}_{l=1}^N$ and $\{\boldsymbol{\theta}_{\mathrm{A}}^{(l)}\}_{l=1}^N$ approximate the posterior densities $\pi(h_n^{\mathrm{A}}|\alpha_{1:n}^o, \boldsymbol{\theta}_{\mathrm{A}})$ and $\pi(\boldsymbol{\theta}_{\mathrm{A}}|\alpha_{1:n}^o, h_{1:n}^{\mathrm{A}})$, respectively. Set $\boldsymbol{\theta}_{\mathrm{A}}$ to the mean of its posterior density, and assume hereafter that it is fixed. At time t_{n+1}, we sample $h_{n+1}^{\mathrm{A},(l)} \sim \mathcal{N}(\boldsymbol{x}_n^{\mathrm{A},\top}\boldsymbol{\beta}_{\mathrm{A}} + \phi_{\mathrm{A}}(h_n^{\mathrm{A},(l)} - \boldsymbol{x}_{n-1}^{\mathrm{A},\top}\boldsymbol{\beta}_{\mathrm{A}}), 1/\tau_{\mathrm{A}}), \forall l$; if α_{n+1} denotes the first observation of the next day, then sample from $\mathcal{N}(\boldsymbol{x}_n^{\mathrm{A},\top}\boldsymbol{\beta}_{\mathrm{A}}, 1/\tau_{\mathrm{A}}(1-\phi_{\mathrm{A}}^2))$. Next, the predictive distribution of α_{n+1} can be evaluated numerically by

$$\pi(\alpha_{n+1}|\alpha_{1:n}^o, \boldsymbol{\theta}_{\mathrm{A}}) = \frac{1}{N}\sum_{l=1}^{N} \pi(\alpha_{n+1}|h_{n+1}^{\mathrm{A},(l)}, \boldsymbol{\theta}_{\mathrm{A}}).$$

The log predictive score of the model for the evaluation period $n+1, \cdots, n'$ is the sum of the log predictive likelihoods $\sum_{l=n}^{n'-1} \log \pi(\alpha_{l+1}^o|\alpha_{1:l}^o, \boldsymbol{\theta}_{\mathrm{A}})$. Before the calculation of the predictive likelihood $\pi(\alpha_{n+2}^o|\alpha_{1:n+1}^o, \boldsymbol{\theta}_{\mathrm{A}})$, we employ the APF (Pitt and Shephard [1999], see Section 2.5.1), to sample again $\{h_{n+1}^{\mathrm{A},(l)}\}_{l=1}^N$ with a proposal distribution which depends on the information available in α_{n+1}^o, as opposed to the previous step in which the proposal density does not depend on this value in order to increase the probability of producing particles that are in agreement with α_{n+1}^o.

4.4 Simulation study

Suppose that the random variables $A_1, \cdots, A_n$ are independent with conditional distribution $A_i|h_i \sim \text{Bernoulli}(\pi_i)$, where

$$\text{logit}(\pi_i) = h_i,$$
$$h_i = \beta_0 + \phi(h_{i-1} - \beta_0) + \varepsilon_i, \;\; \varepsilon_i \sim \mathcal{N}(0, 1/\tau)$$

$h_1 \sim \mathcal{N}(\beta_0, 1/\tau(1-\phi^2))$ and $i \in \{1, 2, \cdots, n\}$. The parameter β_0 is equal to 3 and the parameters ϕ and τ vary on $\{0, 0.95, 0.85, 0.65, 0.55\} \times \{0.05, 0.5, 1, 2, 10, 50\}$. The sample size, n, is fixed to $10,000$ which corresponds to the average number of our data during the trading period between 9:00 and 13:00 EST for the direction model. We also consider the negative values of ϕ, different values of β_0 and n, as well as larger values of τ, and we report if the results are influenced.

Initially, we use the prior distributions specified in Section 4.3. However, we notice that the convergence rate is too slow and gets worse when the parameter ϕ decreases to an absolute value or the conditional precision increases. This behaviour also deteriorates when we increase the number of observations or add more explanatory variables. To speed up the convergence, we use priors with means equaling the true values, more specifically, following Kastner and Frühwirth-Schnatter [2014], the transformed parameter ψ is assumed to follow a Beta distribution with parameters (a_0, b_0), where $a_0 = 40$ and $b_0 = 80/(1 + \phi_{\text{true}}) - 40$. For the parameter τ, we use a gamma distribution with parameters (γ_0, δ_0), where $\gamma_0 = 10$ and $\delta_0 \in \{0.005, 0.05, 0.1, 0.2, 1, 5\}$ for $\tau \in \{0.05, 0.5, 1, 2, 10, 50\}$, respectively. Finally, for the constant parameter, we use a normal distribution with mean β_{true} and variance 10.

We use $N = 100,000$ MCMC draws after a burn-in of $30,000$ for each data set. Starting values are set to the true values. We apply four sampling schemes: C, NC, CNC, NCC. In C (NC), both the latent state and the parameter vector are updated in the centered (non-centered) parametrization. In CNC (NCC), the latent state is only updated in the centered (non-centered) parametrization, while the parameter vector firstly is updated in the centered (non-centered) setting and then in the non-centered (centered) setting. Besides, The parameter vector is updated by applying the '2-bl' and '3-bl' sampler. The latent state is updated using the '1-1' and 'all' method.

In comparing the performance of the MCMC sampling methods we evaluate the ES-S/sec. ESS is theoretically defined for each parameter as the total number of samples generated divided by the autocorrelation time, which is defined to be $1 + \sum_{i=1}^{N} \rho(k)$, where $\rho(k)$ is the autocorrelation at lag k [Kass et al., 1998; Chib and Carlin, 1999]. The sum over the autocorrelation is usually truncated when the autocorrelation drops below 0.1. ESS is a rough estimate of the number of i.i.d. draws that are equivalent to the samples drawn, hence the higher ESS the better.

Table 4.1: Performance of MCMC samplers as measured by ESS/sec for $\pi(\phi|\boldsymbol{h}, \boldsymbol{\alpha}, \tau, \beta_0)$ for the simulated datasets. Higher is better. The quantities are averaged over twenty runs under different seeds initializations. All algorithms run 130,000 iterations after a burn-in of $30,000$ for each data set. The latent state is updated using '1-1' and 'all' method. The parameter vector is updated by applying the '2-bl' and '3-bl' sampler.

Alg.	τ											
	0.05		0.5		1		2		10		50	
	3-bl	2-bl	3-bl	2-bl	3-bl	2-bl	3-bl	2-bl	3-bl	2-bl	3-bl	2-bl
$\phi = 0.95$												
C(1-1)	1.14	1.14	0.91	0.73	0.80	0.65	0.49	0.43	0.18	0.10	0.05	0.04
NC(1-1)	0.46	0.35	0.43	0.30	0.37	0.23	0.23	0.19	0.06	0.07	0.04	0.03
CNC(1-1)	1.09	1.11	0.97	0.85	0.77	0.56	0.41	0.41	0.14	0.12	0.04	0.03
NCC(1-1)	1.09	0.67	0.90	0.79	0.52	0.52	0.44	0.37	0.13	0.09	0.04	0.06
C(all)	0.27	0.36	0.22	0.41	0.46	0.32	0.34	0.39	0.44	0.16	0.33	0.14
NC(all)	0.03	0.05	0.09	0.11	0.17	0.09	0.16	0.10	0.32	0.15	0.56	0.31
CNC(all)	0.45	0.40	0.22	0.54	0.28	0.53	0.27	0.39	0.42	0.43	0.60	0.43
NCC(all)	0.18	0.11	0.22	0.39	0.26	0.35	0.27	0.38	0.50	0.36	0.45	0.38
$\phi = 0.85$												
C(1-1)	1.19	0.95	0.71	0.41	0.40	0.26	0.28	0.15	0.08	0.07	0.05	0.04
NC(1-1)	1.42	1.24	0.58	0.54	0.29	0.33	0.21	0.17	0.05	0.05	0.05	0.03
CNC(1-1)	2.23	1.87	1.03	0.84	0.59	0.49	0.25	0.25	0.08	0.09	0.05	0.04
NCC(1-1)	2.08	1.76	0.93	0.79	0.57	0.46	0.25	0.26	0.08	0.06	0.05	0.04
C(all)	0.33	0.27	0.23	0.29	0.20	0.20	0.24	0.22	0.32	0.17	0.24	0.13
NC(all)	0.17	0.09	0.18	0.12	0.15	0.10	0.19	0.14	0.30	0.24	0.30	0.33
CNC(all)	0.19	0.18	0.28	0.26	0.27	0.23	0.27	0.31	0.36	0.23	0.26	0.23
NCC(all)	0.17	0.19	0.26	0.23	0.25	0.22	0.30	0.22	0.29	0.23	0.25	0.22
$\phi = 0.65$												
C(1-1)	0.54	0.30	0.28	0.15	0.26	0.14	0.20	0.09	0.11	0.08	0.08	0.05
NC(1-1)	0.84	0.52	0.40	0.31	0.28	0.15	0.16	0.15	0.10	0.07	0.08	0.05
CNC(1-1)	1.11	0.87	0.60	0.44	0.37	0.30	0.19	0.20	0.08	0.08	0.06	0.07
NCC(1-1)	1.05	0.69	0.60	0.44	0.36	0.31	0.16	0.20	0.10	0.10	0.08	0.08
C(all)	0.32	0.32	0.14	0.14	0.15	0.18	0.22	0.13	0.25	0.15	0.24	0.12
NC(all)	0.14	0.12	0.16	0.12	0.16	0.15	0.15	0.21	0.22	0.24	0.22	0.35
CNC(all)	0.16	0.18	0.16	0.16	0.18	0.18	0.25	0.19	0.26	0.20	0.30	0.23
NCC(all)	0.16	0.13	0.18	0.15	0.13	0.15	0.23	0.14	0.18	0.22	0.22	0.23
$\phi = 0.55$												
C(1-1)	0.36	0.34	0.22	0.14	0.27	0.12	0.23	0.12	0.12	0.08	0.11	0.07
NC(1-1)	0.79	0.44	0.45	0.27	0.34	0.19	0.22	0.15	0.10	0.08	0.12	0.08
CNC(1-1)	0.87	0.65	0.55	0.42	0.35	0.33	0.19	0.17	0.12	0.10	0.09	0.11
NCC(1-1)	0.74	0.59	0.48	0.34	0.35	0.30	0.19	0.19	0.09	0.12	0.10	0.11
C(all)	0.29	0.26	0.11	0.13	0.17	0.16	0.18	0.14	0.26	0.14	0.29	0.14
NC(all)	0.15	0.14	0.12	0.12	0.17	0.16	0.22	0.22	0.29	0.31	0.23	0.36
CNC(all)	0.16	0.16	0.16	0.14	0.20	0.17	0.23	0.21	0.26	0.22	0.33	0.26
NCC(all)	0.18	0.11	0.10	0.14	0.15	0.13	0.21	0.19	0.19	0.23	0.24	0.26
$\phi = 0.0$												
C(1-1)	8.04	4.65	0.79	0.44	0.41	0.29	0.36	0.21	0.30	0.16	0.32	0.19
NC(1-1)	7.81	5.68	0.77	0.61	0.44	0.37	0.37	0.25	0.34	0.23	0.33	0.21
CNC(1-1)	7.31	7.52	0.73	0.74	0.47	0.43	0.34	0.33	0.31	0.32	0.28	0.31
NCC(1-1)	7.04	6.81	0.79	0.76	0.40	0.45	0.35	0.32	0.29	0.34	0.36	0.33
C(all)	0.34	0.40	0.20	0.26	0.25	0.20	0.27	0.20	0.44	0.23	0.41	0.26
NC(all)	0.37	0.28	0.30	0.24	0.27	0.20	0.27	0.20	0.41	0.49	0.40	0.50
CNC(all)	0.32	0.36	0.20	0.19	0.19	0.24	0.29	0.30	0.38	0.35	0.48	0.43
NCC(all)	0.32	0.31	0.19	0.21	0.20	0.23	0.24	0.27	0.26	0.38	0.24	0.42

Table 4.2: Performance of MCMC samplers as measured by ESS/sec for $\pi(\tau|\boldsymbol{h}, \boldsymbol{\alpha}, \phi, \beta_0)$ for the simulated datasets. Higher is better. The quantities are averaged over twenty runs under different seeds initializations. All algorithms run 130,000 iterations after a burn-in of $30,000$ for each data set. The latent state is updated using '1-1' and 'all' method. The parameter vector is updated by applying the '2-bl' and '3-bl' sampler.

Alg.	τ											
	0.05		0.5		1		2		10		50	
	3-bl	2-bl	3-bl	2-bl	3-bl	2-bl	3-bl	2-bl	3-bl	2-bl	3-bl	2-bl
$\phi = 0.95$												
C(1-1)	0.06	0.05	0.15	0.17	0.22	0.21	0.17	0.18	0.10	0.06	0.05	0.05
NC(1-1)	0.72	0.47	0.59	0.44	0.45	0.30	0.26	0.26	0.07	0.09	0.27	0.23
CNC(1-1)	0.73	0.73	1.24	1.04	0.95	0.74	0.68	0.53	0.16	0.13	0.23	0.13
NCC(1-1)	0.73	0.74	1.01	0.87	0.80	0.67	0.51	0.48	0.15	0.14	0.23	0.18
C(all)	0.03	0.03	0.05	0.08	0.08	0.11	0.16	0.16	0.25	0.13	0.14	0.07
NC(all)	0.10	0.06	0.11	0.11	0.14	0.10	0.18	0.13	0.40	0.23	2.45	1.51
CNC(all)	0.26	0.21	0.47	0.45	0.46	0.59	0.37	0.60	0.68	0.54	2.44	1.85
NCC(all)	0.23	0.20	0.28	0.39	0.31	0.41	0.35	0.44	0.52	0.46	2.19	1.83
$\phi = 0.85$												
C(1-1)	0.08	0.07	0.20	0.16	0.21	0.15	0.16	0.09	0.10	0.07	0.08	0.05
NC(1-1)	0.54	0.35	0.59	0.50	0.34	0.36	0.28	0.22	0.21	0.15	2.09	0.67
CNC(1-1)	0.54	0.55	0.91	0.75	0.69	0.54	0.31	0.33	0.33	0.30	2.21	0.89
NCC(1-1)	0.56	0.45	0.91	0.74	0.65	0.51	0.34	0.33	0.33	0.27	1.84	0.71
C(all)	0.04	0.04	0.09	0.07	0.09	0.08	0.14	0.12	0.19	0.12	0.14	0.10
NC(all)	0.13	0.08	0.24	0.15	0.18	0.14	0.26	0.22	1.26	1.10	17.82	3.44
CNC(all)	0.16	0.14	0.32	0.28	0.36	0.27	0.40	0.38	1.82	0.85	17.95	3.66
NCC(all)	0.18	0.14	0.30	0.29	0.30	0.29	0.38	0.29	1.45	0.74	15.59	3.52
$\phi = 0.65$												
C(1-1)	0.12	0.08	0.14	0.09	0.14	0.08	0.11	0.08	0.09	0.06	0.12	0.06
NC(1-1)	0.31	0.20	0.41	0.29	0.37	0.21	0.39	0.29	3.00	0.96	11.22	1.50
CNC(1-1)	0.37	0.30	0.64	0.41	0.49	0.38	0.46	0.44	2.71	0.90	9.11	1.35
NCC(1-1)	0.39	0.26	0.58	0.41	0.53	0.38	0.47	0.39	2.90	0.81	9.82	0.91
C(all)	0.03	0.03	0.06	0.05	0.08	0.08	0.14	0.09	0.18	0.11	0.14	0.10
NC(all)	0.09	0.10	0.21	0.14	0.26	0.25	0.53	0.59	9.70	1.90	41.28	2.92
CNC(all)	0.12	0.11	0.22	0.19	0.34	0.25	0.80	0.50	10.29	1.95	37.74	2.75
NCC(all)	0.11	0.09	0.20	0.18	0.23	0.25	0.62	0.40	7.11	1.83	34.76	2.50
$\phi = 0.55$												
C(1-1)	0.08	0.07	0.11	0.08	0.14	0.07	0.11	0.06	0.10	0.06	0.13	0.07
NC(1-1)	0.26	0.15	0.44	0.25	0.53	0.26	0.99	0.38	5.08	1.26	18.86	1.52
CNC(1-1)	0.29	0.23	0.56	0.39	0.64	0.37	0.93	0.54	4.62	0.97	15.46	1.40
NCC(1-1)	0.26	0.22	0.47	0.30	0.60	0.34	0.82	0.50	4.24	0.78	15.57	0.78
C(all)	0.02	0.04	0.05	0.05	0.08	0.08	0.16	0.08	0.18	0.11	0.16	0.09
NC(all)	0.10	0.08	0.16	0.14	0.33	0.32	1.30	0.72	12.66	1.84	45.53	2.73
CNC(all)	0.10	0.07	0.19	0.17	0.42	0.25	1.40	0.67	13.15	1.75	42.84	2.36
NCC(all)	0.09	0.07	0.16	0.15	0.26	0.24	0.98	0.66	9.76	1.73	40.38	2.08
$\phi = 0.0$												
C(1-1)	0.11	0.06	0.10	0.05	0.10	0.06	0.10	0.06	0.12	0.07	0.13	0.08
NC(1-1)	0.20	0.13	0.65	0.31	1.36	0.44	3.02	0.66	14.62	1.10	39.05	1.25
CNC(1-1)	0.26	0.18	0.52	0.34	1.04	0.51	2.55	0.66	13.61	0.89	36.03	0.98
NCC(1-1)	0.27	0.14	0.59	0.32	1.13	0.46	2.21	0.51	13.05	0.61	35.40	0.61
C(all)	0.04	0.04	0.05	0.05	0.08	0.06	0.14	0.07	0.15	0.10	0.14	0.10
NC(all)	0.10	0.08	0.32	0.17	0.82	0.30	2.29	0.73	21.18	1.51	54.90	1.85
CNC(all)	0.11	0.07	0.31	0.19	0.75	0.43	2.31	0.81	19.47	1.50	50.13	1.88
NCC(all)	0.09	0.06	0.24	0.19	0.66	0.36	1.94	0.75	17.60	1.48	48.77	1.43

Table 4.3: Performance of MCMC samplers as measured by ESS/sec for $\pi(\beta_0|\boldsymbol{h}, \boldsymbol{\alpha}, \phi, \tau)$ for the simulated datasets. Higher is better. The quantities are averaged over twenty runs under different seeds initializations. All algorithms run 130,000 iterations after a burn-in of $30,000$ for each data set. The latent state is updated using '1-1' and 'all' method. The parameter vector is updated by applying the '2-bl' and '3-bl' sampler.

Alg.	τ											
	0.05		0.5		1		2		10		50	
	3-bl	2-bl	3-bl	2-bl	3-bl	2-bl	3-bl	2-bl	3-bl	2-bl	3-bl	2-bl
$\phi = 0.95$												
C(1-1)	1.79	0.95	0.74	0.85	0.78	0.87	0.82	0.65	0.23	0.19	0.05	0.05
NC(1-1)	0.09	0.06	0.07	0.09	0.09	0.09	0.07	0.09	0.15	0.16	0.73	0.34
CNC(1-1)	5.43	4.89	7.54	6.75	7.22	6.97	6.21	5.71	3.50	3.83	1.46	1.67
NCC(1-1)	5.19	5.02	7.46	6.03	6.39	5.91	6.57	5.36	2.95	2.37	1.79	0.94
C(all)	1.61	1.59	0.71	0.45	0.41	0.47	0.40	0.54	0.61	0.71	1.36	1.06
NC(all)	0.06	0.04	0.05	0.04	0.05	0.04	0.07	0.04	0.73	0.54	7.54	5.15
CNC(all)	1.38	2.50	2.51	3.12	4.54	4.17	6.08	6.27	12.72	11.77	9.96	6.53
NCC(all)	2.00	1.11	3.38	2.53	5.35	4.10	6.15	4.89	16.28	10.88	14.20	8.72
$\phi = 0.85$												
C(1-1)	0.17	0.18	0.34	0.31	0.34	0.26	0.25	0.16	0.11	0.11	0.06	0.05
NC(1-1)	0.35	0.29	0.75	0.69	0.86	0.74	0.99	0.84	0.99	0.83	5.14	1.51
CNC(1-1)	0.97	0.89	2.42	2.34	2.25	1.95	1.92	1.93	1.48	1.68	8.08	3.16
NCC(1-1)	1.00	1.03	2.64	2.27	2.25	1.90	2.13	1.96	1.43	1.09	5.65	2.08
C(all)	0.53	0.32	0.16	0.10	0.14	0.12	0.17	0.18	0.48	0.38	0.99	0.94
NC(all)	0.08	0.06	0.22	0.09	0.39	0.20	1.03	0.67	5.01	3.53	9.94	9.25
CNC(all)	0.28	0.37	1.08	1.01	1.42	1.34	2.42	1.98	5.71	4.03	8.67	9.14
NCC(all)	0.36	0.27	1.10	1.02	1.63	1.20	2.48	2.04	6.41	4.09	6.15	7.44
$\phi = 0.65$												
C(1-1)	0.13	0.10	0.16	0.11	0.16	0.11	0.13	0.15	0.14	0.12	0.07	0.06
NC(1-1)	0.31	0.19	0.59	0.39	0.94	0.51	2.40	1.16	2.99	1.23	39.15	17.29
CNC(1-1)	0.38	0.33	0.77	0.53	0.94	0.66	2.22	1.16	2.63	1.93	32.09	19.23
NCC(1-1)	0.42	0.27	0.84	0.55	1.20	0.67	2.13	1.00	2.20	1.13	33.48	22.82
C(all)	0.04	0.06	0.07	0.05	0.09	0.09	0.21	0.12	0.41	0.41	0.57	0.62
NC(all)	0.12	0.08	0.31	0.20	0.75	0.52	2.05	1.28	6.77	3.46	55.78	53.10
CNC(all)	0.16	0.13	0.30	0.26	0.80	0.56	2.29	1.31	6.26	3.44	52.72	38.29
NCC(all)	0.13	0.11	0.32	0.33	0.59	0.53	1.86	1.33	5.95	3.13	54.29	36.26
$\phi = 0.55$												
C(1-1)	0.09	0.08	0.12	0.09	0.15	0.11	0.14	0.10	0.15	0.13	0.07	0.07
NC(1-1)	0.23	0.15	0.54	0.30	1.07	0.44	2.84	1.06	4.81	1.70	51.59	43.07
CNC(1-1)	0.29	0.24	0.55	0.42	0.93	0.59	2.32	1.16	3.92	1.74	41.78	36.66
NCC(1-1)	0.25	0.23	0.65	0.35	1.23	0.56	2.40	0.84	4.75	1.78	43.81	42.64
C(all)	0.03	0.06	0.05	0.05	0.08	0.09	0.24	0.09	0.41	0.36	0.41	0.45
NC(all)	0.11	0.08	0.21	0.18	0.70	0.57	2.38	1.15	11.18	5.10	72.10	66.36
CNC(all)	0.09	0.08	0.24	0.17	0.74	0.40	2.38	1.26	11.50	4.21	67.95	56.91
NCC(all)	0.12	0.07	0.21	0.21	0.54	0.49	1.85	1.20	8.68	3.79	66.95	59.01
$\phi = 0.0$												
C(1-1)	0.10	0.07	0.10	0.05	0.10	0.06	0.12	0.07	0.20	0.19	0.07	0.06
NC(1-1)	0.20	0.12	0.56	0.27	1.20	0.40	2.86	0.72	33.51	9.43	55.74	55.70
CNC(1-1)	0.21	0.17	0.42	0.30	0.87	0.50	2.36	0.71	30.11	6.37	50.57	47.95
NCC(1-1)	0.27	0.14	0.47	0.27	0.85	0.43	1.68	0.52	30.22	2.90	50.43	49.75
C(all)	0.04	0.04	0.04	0.05	0.07	0.06	0.15	0.08	0.23	0.22	0.07	0.11
NC(all)	0.09	0.07	0.26	0.14	0.59	0.24	1.79	0.72	46.60	9.54	77.84	75.03
CNC(all)	0.08	0.06	0.28	0.18	0.63	0.41	2.11	0.86	41.72	9.85	70.89	65.30
NCC(all)	0.10	0.07	0.22	0.15	0.50	0.31	1.58	0.83	38.13	10.31	70.24	67.47

The simulation results for all datasets and algorithms considered are presented in Tables 4.1 - Table 4.3. The quarantines in the tables are averaged over twenty runs under different seeds initializations. The mean running time (in minutes) that describes the amount of time it takes for the C(1-1), NC(1-1), CNC(1-1), NCC(1-1), C(all), NC(all), CNC(all), NCC(all) algorithm to complete 130,000 iterations is 5.72, 5.86, 6.49, 6.36, 4.10, 4.19, 4.87, 4.70 for the 3-block sampler and 5.67, 5.85, 6.53, 6.46, 4.02 4.19, 4.87, 4.81 for the 2-block sampler, respectively.

Concerning the simulation efficiency of ϕ (Table 4.1), the results show that keeping ϕ fixed, it seems that as the value of τ increases, the sampling efficiency of ϕ deteriorates. Keeping τ fixed, as the value of $|\phi|$ decreases, the sampling efficiency of ϕ gradually decline, and rise when $\phi = 0$. Note that, when $\tau < 1$, as $|\phi|$ drops form 0.95 to 0.85, the sampling efficiency increases. Comparing the sampling schemes, it seems that the interweaving samplers exhibit lower autocorrelation (and therefore a higher ESS/sec) and show similar values for both parametrizations with the centered parametrization has larger (even slightly) values for most underlying parameter values. In most cases, for $|\phi| \geq 0.85$, the centered parametrization exhibits lower autocorrelation than the non-centered, however for the rest values of $|\phi|$ this behaviour is reversed. Comparing the latent path strategy, the results show that as the value of the conditional precision increases it is preferable to update the components of the latent state in one step. Finally, by comparing the blocking strategy the 3-block sampler shows higher ESS/sec most of the time. As the sample size increases, the sampling efficiency deteriorates, for example, when $n = 5,000$ the ESS/sec is on average four times larger than its corresponding value when the sample size is doubled.

Concerning the simulation efficiency of τ (Table 4.2), it seems that for the two smallest (largest) values of τ, as $|\phi|$ decreases, the sampling efficiency deteriorates (increases), while its moderate values witness a gradual decline, followed by a rise. Comparing the sampling schemes, the result show that the centered parametrization performs poorly compared to the non-centered which exhibits, for some underlying parameter values, ESS/sec similar to the interweaving samplers which have the smallest autocorrelation for both parametrizations for most of the parameters values. Comparing the method for updating the latent state, the table reports that for $\tau < 2$ it is preferable to update the components of the latent state one at time, while for the rest values of τ it is preferable to update the latent state in one step. Besides, by comparing the blocking strategy the 3-block sampler shows lower autocorrelation most of the time. The

above results hold irrespective of the sample size or the choice of β_0. Finally, as the sample size increases, the sampling efficiency deteriorates, for example, when $n = 5,000$ the ESS/sec is on average three times larger than its corresponding value when the sample size is doubled.

Concerning the simulation efficiency of β_0 (Table 4.3), the results show that keeping $|\phi|$ fixed, it seems that as the value of τ increases, the sampling efficiency increases. On the other hand, for $\tau < 10$ ($\tau \geq 10$), as $|\phi|$ decreases, the ESS/sec declines (initially declines and then rises). However, the last conduction depends on the true value of β_0, for example, when $\beta_0 = 0$ and $\tau = 0.05$ (10), the sampling efficiency increases (deteriorates). Comparing the latent path strategy, the results show that when $\tau \geq 10$, it is preferable to update the whole latent state in one move, while for the rest values of τ it is preferable to update its components one at a time. Comparing the sampling schemes, it seems that the interweaving samplers have higher ESS/sec and show similar values for both parametrizations for most underlying parameter values. In most cases, the non-centered parametrization exhibits lower autocorrelation than the centered, but this conclusion does not hold in general, as it depends on the sample size and the choice of β_0. Comparing the blocking strategy, the 3-block sampler show higher ESS/sec. Finally, as the sample size increases, the sampling efficiency deteriorates, for example, when $n = 5,000$ the ESS/sec is on average three times larger than its corresponding value when the sample size is doubled.

4.5 Real data results

The general estimation procedure is described in Section 3.5.1. If we do not make a specific reference to which data segment we are referring to, it means that we imply both examined time periods. The curves with blue (black) color refer to the morning (afternoon) period. The regressors are explained in Sections 3.5.2 and 3.5.3.

4.5.1 Part A

We use the CNC algorithm combined with the 3-block sampler in which the latent path is updated using the '1-1' method. Red curves indicate that the latent path is drawn in one move.

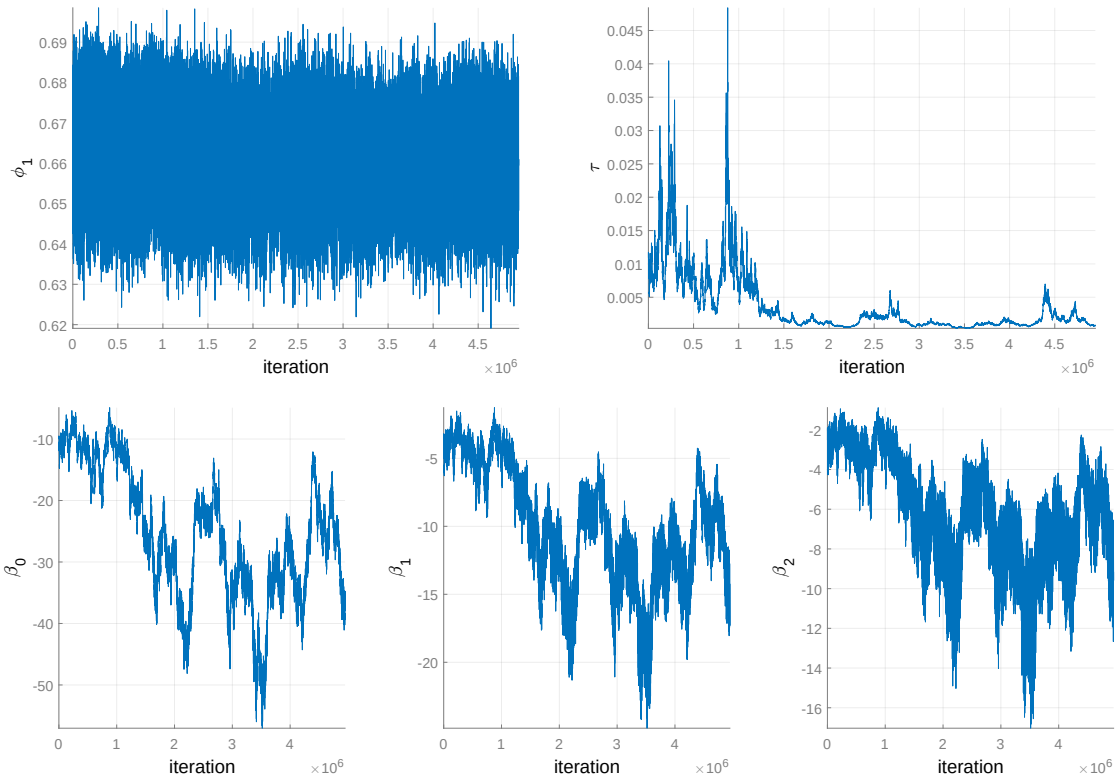


Figure 4.2: Trace plot of the 5,000,000 iterations of the Markov chain targeting the posterior distribution of the estimated parameters for the activity model, May 16th to May 20th, 9 a.m. - 1 p.m., case (I), generated by the algorithm CNC combined with the 3-block sampler. The latent path is updated using the '1-1' method.

4.5.1.1 Case (I)

Using the default priors the algorithms result in very slow convergence, for example, Figure 4.2 displays the trace plot of the MCMC chains (i.e. a plot of iterations vs. sampled values) during the morning period, indicating that the series of iterations has not converged even after 5,000,000 iterations. However, if we pre-specify the conditional precision, τ_{A}, as a known constant (despite a parameter in the model) this behaviour of the model improves, for example, Figure 4.3 shows the trace plot of the 100,000 iterations of the Markov chain targeting the posterior distribution of the estimated parameters, during the morning (afternoon) period with $\tau_{\mathrm{A}} = 0.07$ (0.1).

Figure 4.4 displays the effect of the choice of τ_{A}, based on the (one-step ahead) log predictive score of the model for the last two days of the dataset. The figure suggests that it is maximized for values of τ near 0.07 (0.1) for the morning (afternoon) period. A suitable informative prior for ψ_{A} is $\mathcal{B}e(23, 4)$ ($\mathcal{B}e(41, 5)$) with a prior mean 0.7 (0.78) and variance

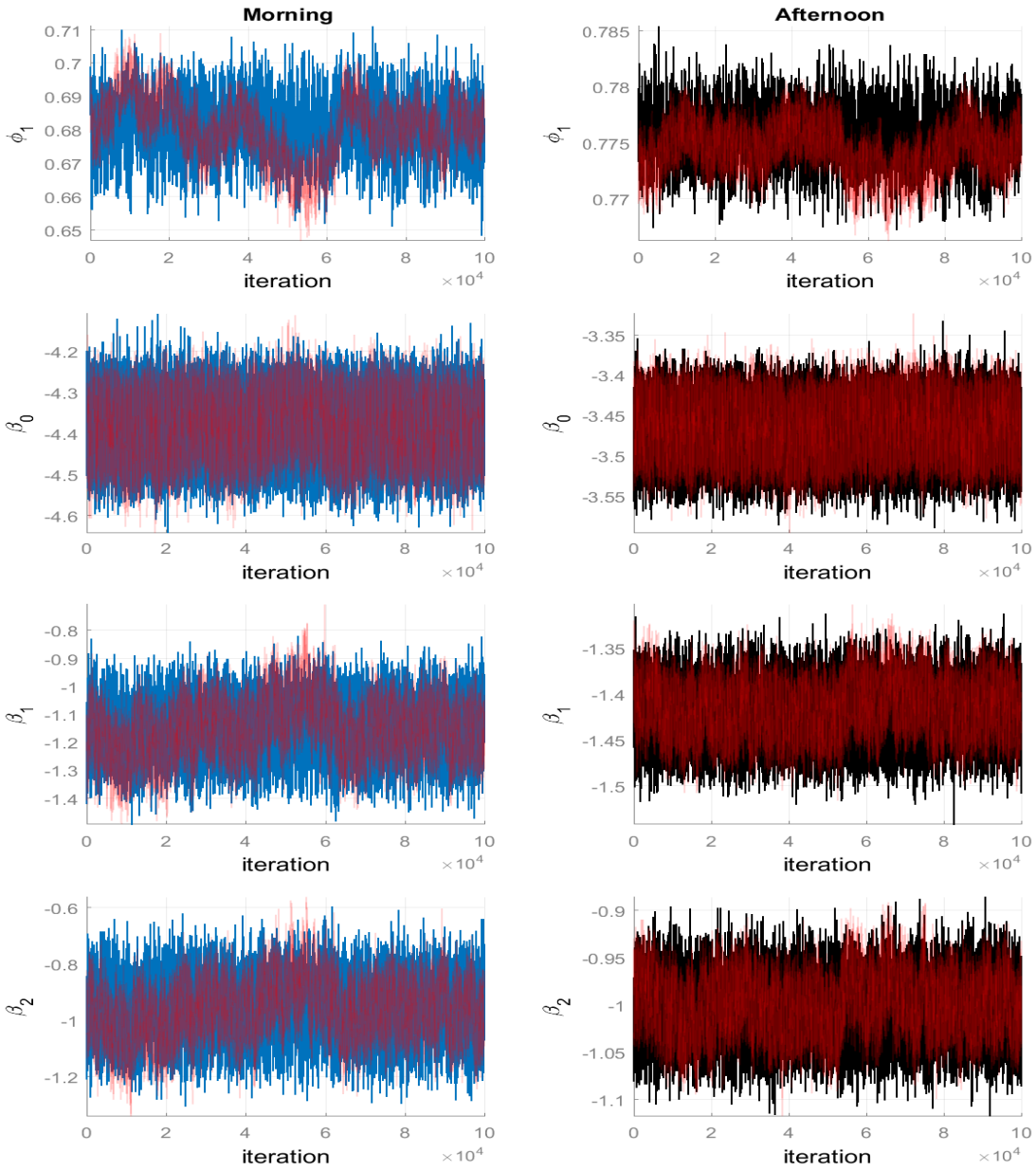


Figure 4.3: Trace plots of the 100,000 iterations post burn-in of Markov chains targeting the posterior distribution of the estimated parameters for the activity model, May 16th to May 20th, 9 a.m. - 1 p.m. (blue curve) and 1 a.m. to 5 p.m. (black curve), case (I), with $\tau = 0.07$ (0.1). The samples are generated by the algorithm CNC combined with the 3-block sampler in which the latent path is updated using the '1-1' and the 'all' (red curve) method.

0.02 (0.01) of ϕ_{A} during the morning (afternoon) period. For τ_{A}, we choose a $\mathrm{Gamma}(70, 1\mathrm{e}{-3})$ ($\mathrm{Gamma}(1\mathrm{e}{+3}, 1\mathrm{e}{-4})$) with prior mean 0.07 (0.1) and variance $6.98\mathrm{e}{-5}$ ($9.98\mathrm{e}{-6}$). We have tried priors with larger prior variance, for example $\mathrm{Gamma}(35, 2\mathrm{e}{-3})$ ($\mathrm{Gamma}(100, 1\mathrm{e}{-3})$) with variance $1.40\mathrm{e}{-4}$ ($1.01\mathrm{e}{-4}$), however this choice results in high autocorrelation samples. The algorithm runs 3,800,000 (2,880,000) iterations for the morning (afternoon) dataset with

burn-in of 50,000 iterations and the autocorrelation is reduced by retaining only every 1900th (1440th) iteration of the chain. Figure 4.5 illustrates the marginal posterior densities of the estimated parameters and prior densities (dashed line).

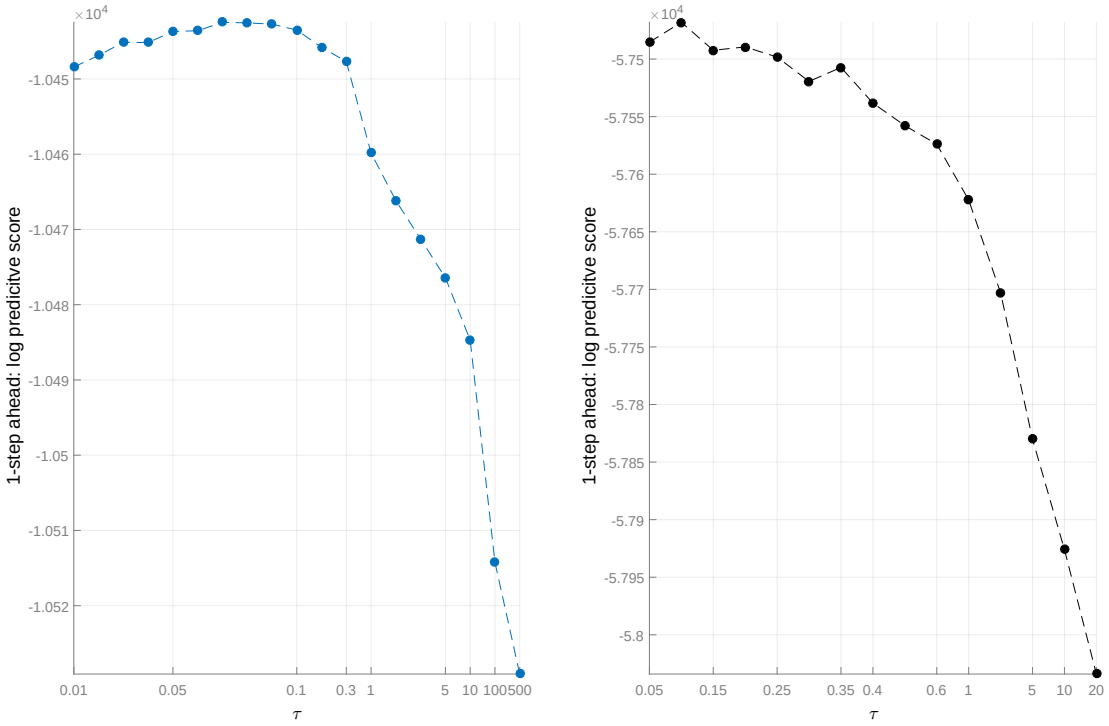
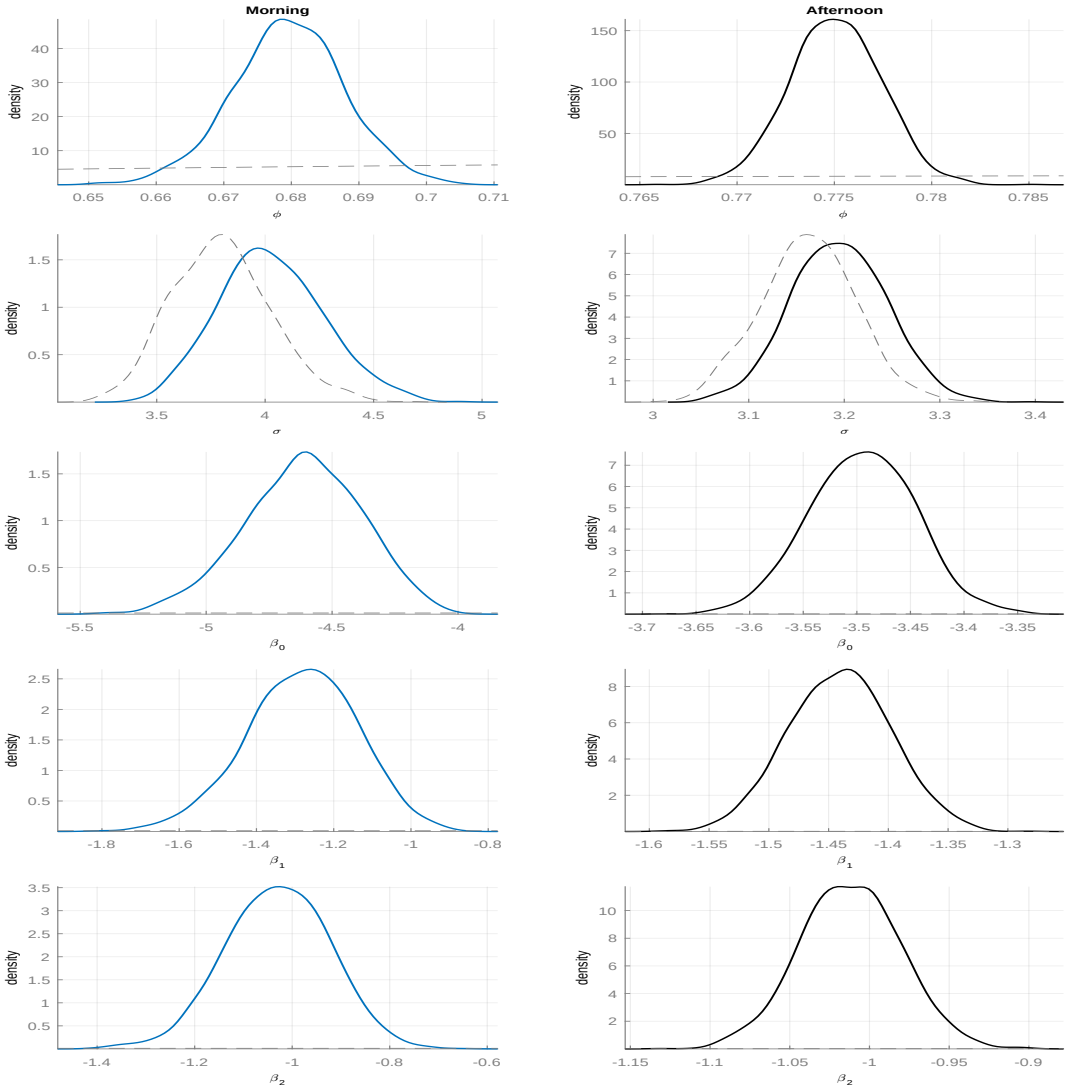


Figure 4.4: Forecasting performance: (one-step ahead) log predictive score versus τ for the activity model for ES, May 23rd & 24th 2011, 9 a.m. - 1 p.m. (left panel) and 1 p.m. - 5 p.m. (right panel), case (I).

Table 4.4 lists the parameter estimates along with the running time (in hours). The posterior mean of ϕ_{A} is 0.68 (0.78), and the posterior mean of $\sigma_{\mathrm{A}} \triangleq \sqrt{1/\tau_{\mathrm{A}}}$ is 4.03 (3.20) for the morning (afternoon) dataset. The negative value of the posterior mean of the lagged price activity shows that past active trades tend to decrease the probability of subsequent movements in the price. The standard deviations are smaller for the afternoon dataset compared to the morning dataset.



Model: A_i takes on value 1 (0) with probability π_i $(1-\pi_i)$, with $\text{logit}(\pi_i) = h_i$, $h_i = \boldsymbol{x}_{i-1}^\top \boldsymbol{\beta} + \phi(h_{i-1} - \boldsymbol{x}_{i-2}^\top \boldsymbol{\beta}) + \varepsilon_i$, $\varepsilon_i \sim \mathcal{N}(0, 1/\tau)$, $\boldsymbol{x}_{i-1}^\top = (1, A_{i-1}, A_{i-2})$. Priors: $(\phi + 1)/2 \sim \mathcal{B}e(23, 4)$ $(\mathcal{B}e(41, 5))$, $\tau \sim \text{Gamma}(70, 1\text{e}{-3})$ $(\text{Gamma}(1\text{e}{+3}, 1\text{e}{-4}))$, $\boldsymbol{\beta} \sim \mathcal{N}_3(\mathbf{0}_3, 10^3 \mathbf{I}_3)$.

Figure 4.5: Marginal posterior densities (using a kernel density with 2000 MCMC thinned samples) of the estimated parameters for the activity model, estimated with the algorithm CNC combined with the 3-block sampler in which the latent path is updated using the '1-1' method; prior (dashed line). The data set we are analyzing is ES, May 16th to May 20th, confined to the morning partition (blue curve) and afternoon partition (black curve), case (I).

Table 4.4: MCMC estimates of parameters for the activity AR(1) model for ES, May 16th to May 20th, case (I), estimated by the algorithm CNC combined with the 3-block sampler in which the latent path is updated using the '1-1' ('all') method, for the AR1 (WN) model. 'Run-time' returns the execution time in hours. Model: A_i takes on value 1 (0) with probability π_i $(1-\pi_i)$, with logit$(\pi_i) = h_i$, $h_i = \boldsymbol{x}_{i-1}^\top \boldsymbol{\beta} + \phi(h_{i-1} - \boldsymbol{x}_{i-2}^\top \boldsymbol{\beta}) + \varepsilon_i$, $\varepsilon_i \sim \mathcal{N}(0, 1/\tau)$, $\boldsymbol{x}_{i-1}^\top = (1, A_{i-1}, A_{i-2})$.

	MCMC	
Parameter	Morning	Afternoon
ϕ	0.680	0.775
	(0.008)	(0.002)
σ	4.028	3.195
	(0.2413)	(0.049)
β_0	-4.614	-3.495
	(0.231)	(0.049)
β_1	-1.283	-1.441
	(0.144)	(0.043)
β_2	-1.030	-1.012
	(0.107)	(0.032)
Run-time	18h, *0.75 days*	78h, *3.25 days*

4.5.1.2 Case (II)

As in the previous section, the algorithms result in very slow convergence, for example, Figure 4.6 displays the trace plot of the MCMC chains during the morning period, indicating that the series of iterations has not converged even after 3,000,000 iterations. However, keeping τ_{A} fixed result in better mixing; see Figure 4.7. Figure 4.8 suggests that the log predictive score of the model is maximized for τ near 0.08 (0.25) for the morning (afternoon) period. However, for largest values of the conditional precision the mixing of the autocorrelation parameter is slow. For this reason, we keep constant ϕ_{A} and τ_{A} and compare the effect of these choices leading to the same conduction; the results are not presented here.

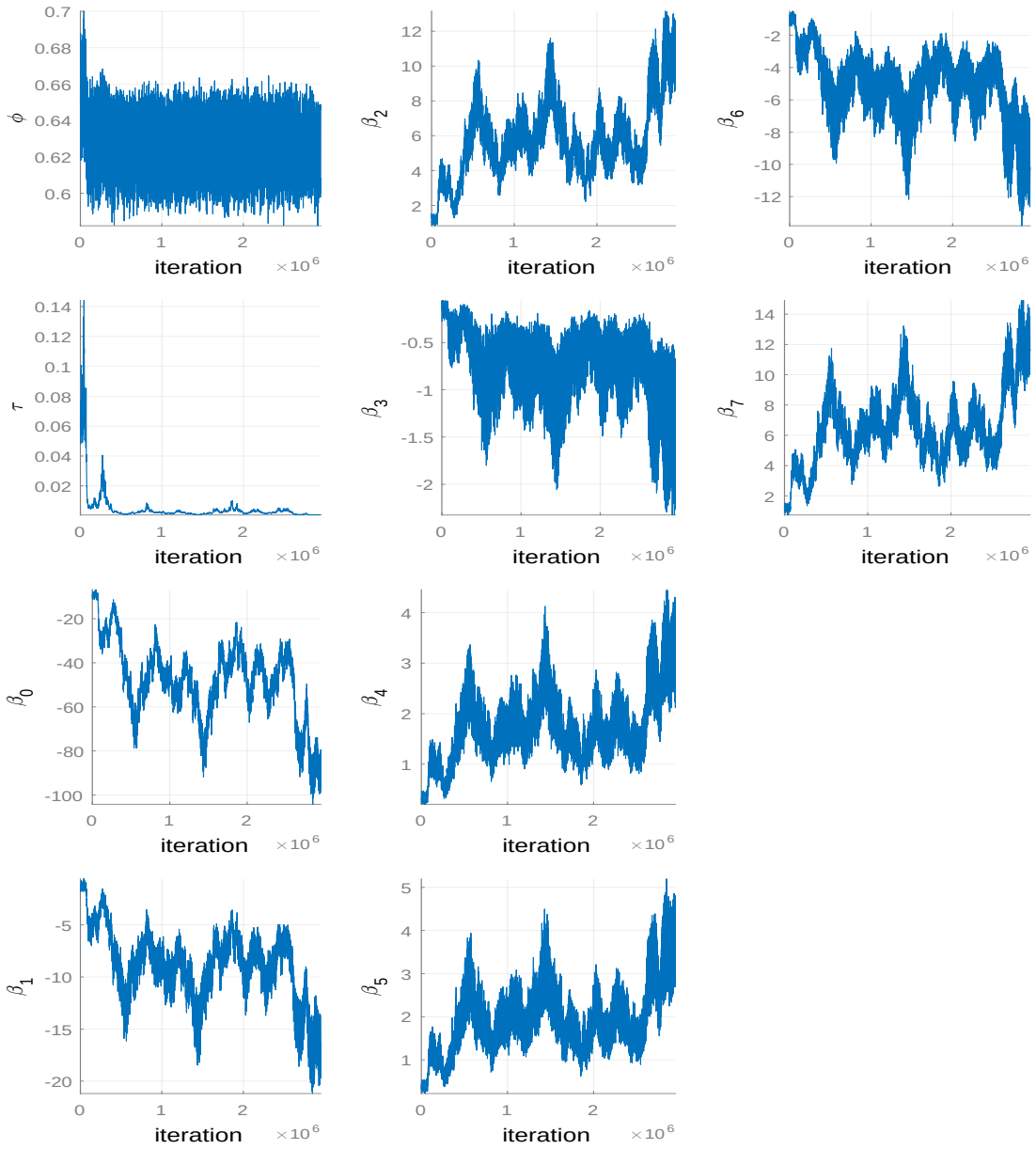


Figure 4.6: Trace plot of the 3,000,000 iterations of the Markov chain targeting the posterior distribution of the estimated parameters for the activity model, May 16th to May 20th, 1 p.m. - 5 p.m., case (II), generated by the algorithm CNC combined with the 3-block sampler. The latent path is updated one at time.

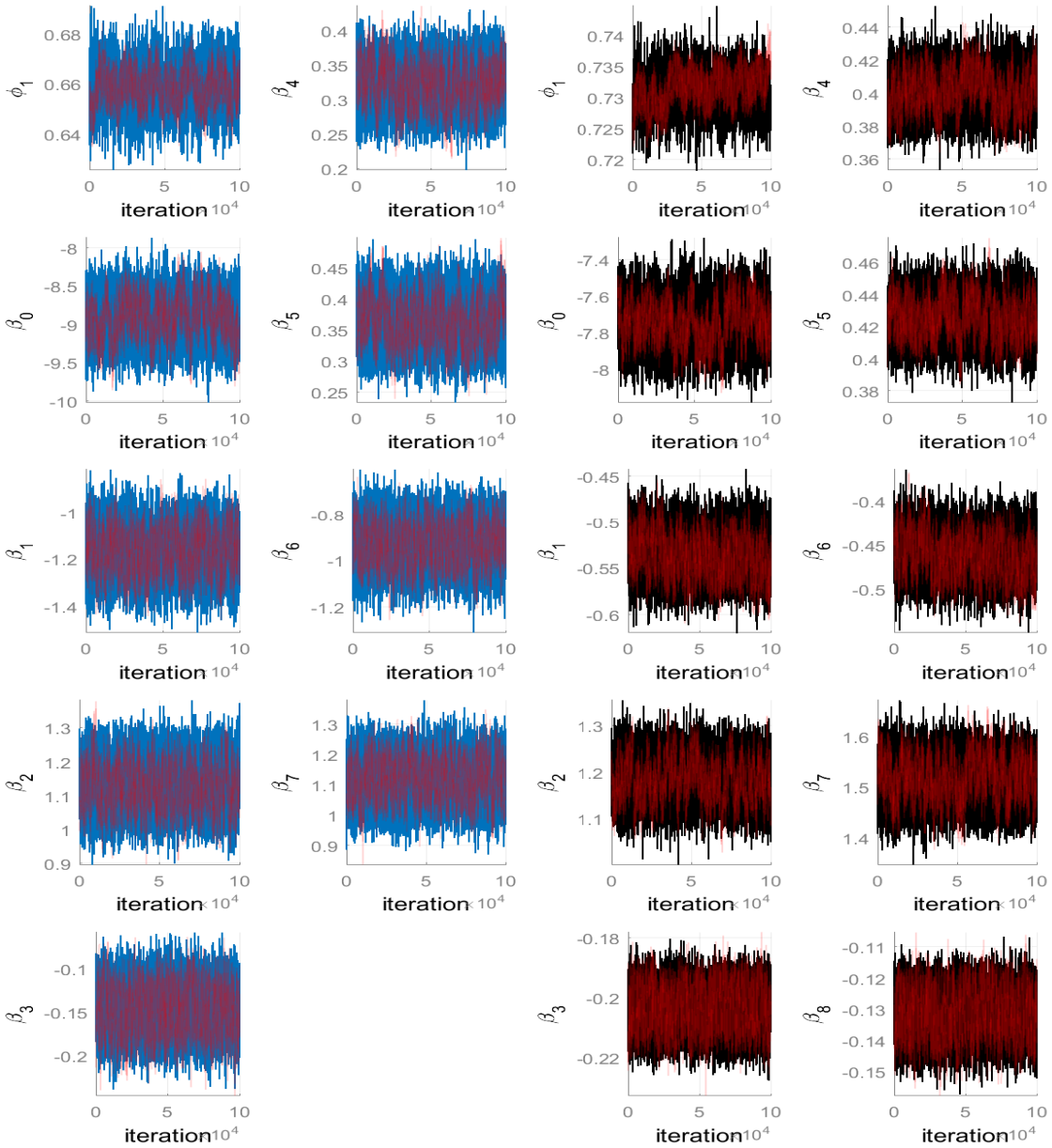


Figure 4.7: Trace plots of the 100,000 iterations post burn-in of Markov chains targeting the posterior distribution of the estimated parameters for the activity model, May 16th to May 20th, 9 a.m. - 1 p.m. (blue curve) and 1 a.m. to 5 p.m. (black curve), case (II), with $\tau = 0.08$ (0.25). The samples are generated by the algorithm CNC combined with the 3-block sampler in which the latent path is updated using the '1-1' and the 'all' (red curve) method.

For ψ_{A}, we choose a $\mathcal{B}e(25, 5)$ ($\mathcal{B}e(23, 4)$) with a prior mean 0.67 (0.70) and variance 0.02 of ϕ_{A} during the morning (afternoon) period. For τ_{A}, we choose a $\mathrm{Gamma}(80, 1\mathrm{e}{-3})$ ($\mathrm{Gamma}(250, 1\mathrm{e}{-3})$) with prior mean 0.08 (0.25) and variance $7.99\mathrm{e}{-5}$ ($2.49\mathrm{e}{-4}$). We have tried priors with larger prior variance, for example $\mathrm{Gamma}(41, 2\mathrm{e}{-3})$ ($\mathrm{Gamma}(25, 0.01)$) with variance $1.64\mathrm{e}{-4}$ ($3\mathrm{e}{-3}$), however this choice results in high autocorrelation samples.

The algorithm runs 3,200,000 (3,400,000) iterations for the morning (afternoon) dataset with burn-in of 50,000 (100,000) iterations and the autocorrelation is reduced by retaining only every 1600th (1700th) iteration of the chain.

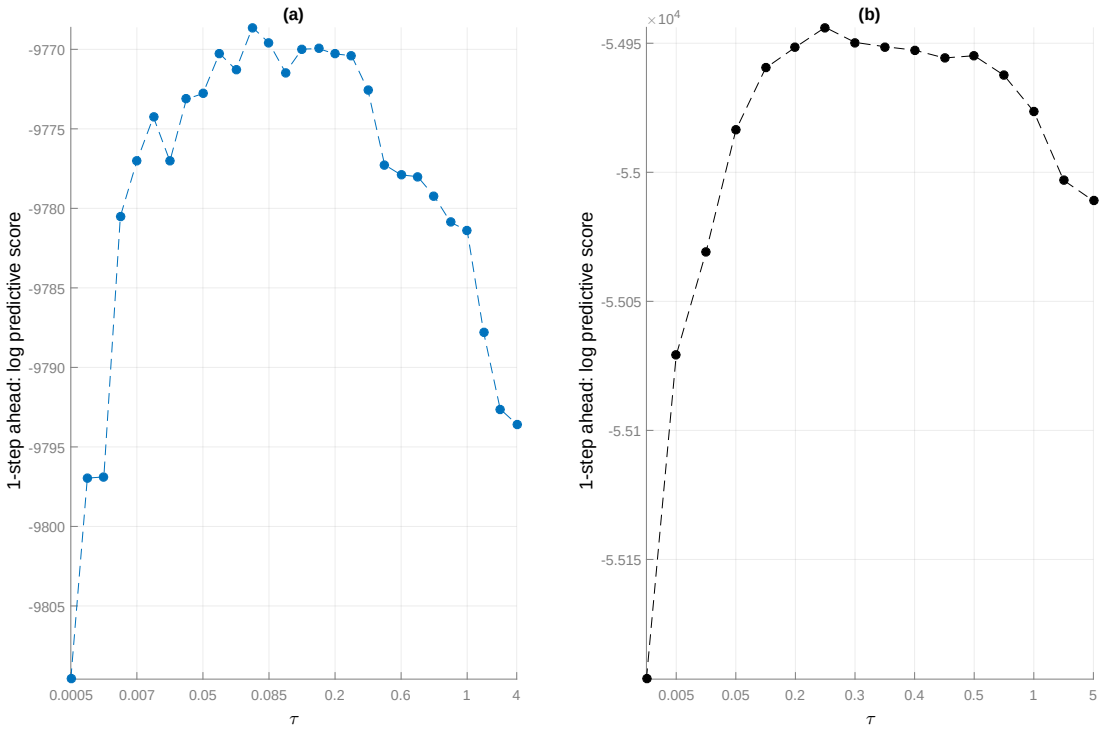
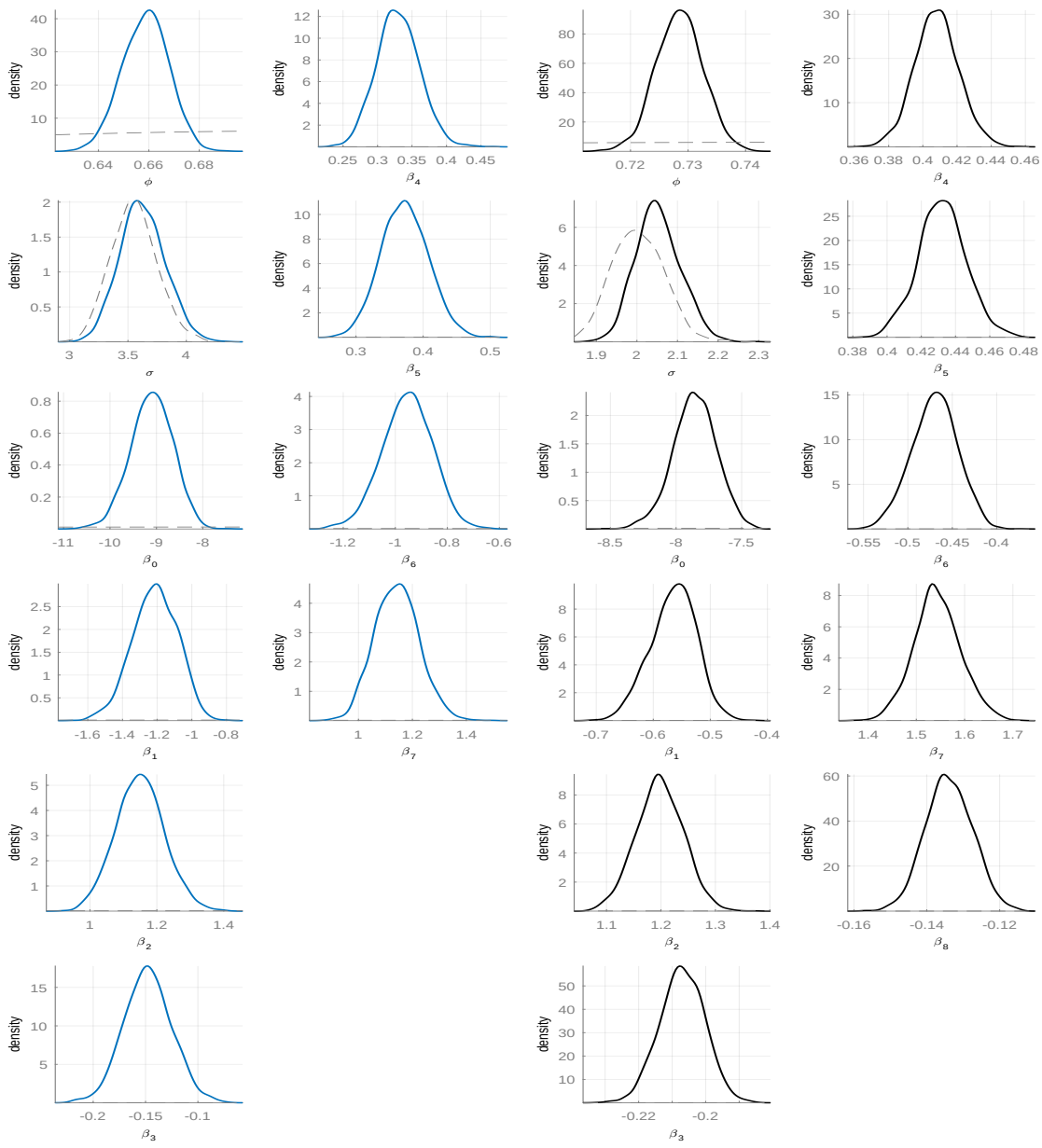


Figure 4.8: Forecasting performance: (one-step ahead) log predictive score versus τ for the activity model for ES, May 23rd & 24th 2011, 9 a.m. - 1 p.m. (left panel) and 1 p.m. - 5 p.m. (right panel), case (II).

Table 4.5 lists the parameter estimates along with the running time (in hours). The posterior mean of ϕ_{A} is 0.66 (0.73), and the posterior mean of $\sigma_{\mathrm{A}} \triangleq \sqrt{1/\tau_{\mathrm{A}}}$ is 3.62 (2.05) for the morning (afternoon) dataset. The influence of lagged price activity, is negative for all past period values, indicating that past active trades tend to decrease the probability of subsequent movements in the price, while this reduction decays down at lag two. The table shows that lagged log-durations have a very dramatic positive impact on the chance that a trade moves the transaction price. Besides, a smaller but negative impact is made by the log trading volume. For the quoted volumes on the previous best level, we find a positive impact on the activity process. More specifically, the effect of the buying volume is slightly larger than the impact of the selling volume. The standard deviations are smaller for the afternoon dataset compared to the morning dataset. Figure 4.9 illustrates the marginal posterior densities of the estimated parameters and

prior densities (dashed line).



Model: A_i takes on value 1 (0) with probability π_i $(1-\pi_i)$, with $\text{logit}(\pi_i) = h_i$, $h_i = \boldsymbol{x}_{i-1}^\top \boldsymbol{\beta} + \phi(h_{i-1} - \boldsymbol{x}_{i-2}^\top \boldsymbol{\beta}) + \varepsilon_i$, $\varepsilon_i \sim \mathcal{N}(0, 1/\tau)$, $\boldsymbol{x}_{i-1}^\top = (1, A_{i-1}, \tau_{i-1}, V_{i-1}^{\text{mo}}, V_{i-1}^{\text{b,l}}, V_{i-1}^{\text{a,l}}, A_{i-2}, \tau_{i-2})$. Priors: $(\phi+1)/2 \sim \mathcal{B}e(25,5)$, $\tau \sim \text{Gamma}(80, 1\text{e}{-3})$, $\boldsymbol{\beta} \sim \mathcal{N}_d(\mathbf{0}_d, 10^3 \mathbf{I}_d)$.

Figure 4.9: Marginal posterior densities (using a kernel density with 2000 MCMC thinned samples) of the estimated parameters for the activity model, estimated with the algorithm CNC combined with the 3-block sampler in which the latent path is updated using the '1-1' method; prior (dashed line). The data set we are analyzing is ES, May 16th to May 20th, confined to the morning partition, case (II).

Table 4.5: MCMC estimates of parameters for the activity model for ES, May 16th to May 20th, case (II), estimated by the algorithm CNC combined with the 3-block sampler in which the latent path is updated using the '1-1' method. 'Run-time' returns the execution time in hours. The Bayesian estimates represent posterior means and standard deviations in parenthesis. The algorithm runs 3,200,000 (3,400,000) iterations with a burn-in period of 50,000 (100,000) draws, thinning every 1600th (1700th) iteration over the morning (afternoon) period, yielding 2000 draws. Model: A_i takes on value 1 (0) with probability π_i $(1-\pi_i)$, with $\text{logit}(\pi_i) = h_i$, $h_i = \boldsymbol{x}_{i-1}^\top \boldsymbol{\beta} + \phi(h_{i-1} - \boldsymbol{x}_{i-2}^\top \boldsymbol{\beta}) + \varepsilon_i$, $\varepsilon_i \sim \mathcal{N}(0, 1/\tau)$, $\boldsymbol{x}_{i-1}^\top = (1, A_{i-1}, \tau_{i-1}, V_{i-1}^{\text{mo}}, V_{i-1}^{\text{b,1}}, V_{i-1}^{\text{a,1}}, A_{i-2}, \tau_{i-2})$ ($\boldsymbol{x}_{i-1}^\top = (1, A_{i-1}, \tau_{i-1}, V_{i-1}^{\text{mo}}, V_{i-1}^{\text{b,1}}, V_{i-1}^{\text{a,1}}, A_{i-2}, \tau_{i-2}, V_{i-2}^{\text{mo}})$). Priors: $(\phi+1)/2 \sim \mathcal{B}e(25, 5)$, $(\mathcal{B}e(23, 4))$, $\tau \sim \text{Gamma}(80, 1\text{e}{-3})$ $(\text{Gamma}(250, 1\text{e}{-3}))$, $\boldsymbol{\beta} \sim \mathcal{N}_d(\mathbf{0}_d, 10^3 \mathbf{I}_d)$, $d \in \{8, 9\}$. The acceptance ratio is 0.38, 0.50, 0.46, 0.56, 0.45, 0.52, 0.39. 0.56, 0.46 0.27 (0.44 0.46, 0.45, 0.44, 0.44, 0.42, 0.46, 0.39, 0.45, 0.54, 0.27) for ϕ, τ, $\boldsymbol{\beta}$, respectively.

	MCMC	
Parameter	Morning	Afternoon
ϕ	0.659	0.729
	(0.009)	(0.004)
σ	3.624	2.053
	(0.194)	(0.056)
β_0	-9.102	-7.859
	(0.453)	(0.166)
β_1	-1.213	-0.565
	(0.129)	(0.040)
β_2	1.150	1.198
	(0.072)	(0.043)
β_3	-0.148	-0.207
	(0.023)	(0.007)
β_4	0.330	0.408
	(0.031)	(0.013)
β_5	0.373	0.432
	(0.035)	(0.014)
β_6	-0.954	-0.470
	(0.096)	(0.026)
β_7	1.143	1.544
	(0.082)	(0.048)
β_8	-	-0.134
	-	(0.006)
Run-time	18h, *0.75 days*	126h, *5.25 days*

4.5.2 Part B

The results obtained with the AR(1) direction model still lead to highly autocorrelated parameter values, even after keeping the conditional precision as a known constant. For this reason, we consider only the WN and RW(1) model.

4.5.2.1 Case I

Concerning the WN model, the algorithm runs 180,000 (300,000) iterations for the morning (afternoon) dataset with burn-in of 20,000 iterations and the autocorrelation is reduced by retaining only every 90th (150th) iteration of the chain. Concerning the RW(1) model, the algorithm runs $2\text{e}{+6}$ iterations with burn-in of 20,000 iterations and the autocorrelation is reduced by retaining only every 1000th iteration of the chain. We use the CNC algorithm combined with the 3-block sampler in which the latent path is updated using the 'all' method.

Table 4.6: MCMC estimates of parameters of the WN and RW(1) direction model for ES, May 16th to May 20th, case (I), estimated by the algorithm CNC combined with the 3-block sampler in which the latent path is updated using the 'all' method. 'Run-time' returns the execution time in minutes. The Bayesian estimates represent posterior means and standard deviations in parenthesis. WN model: the algorithm runs 180,000 (300,000) iterations for the morning (afternoon) dataset with burn-in of 20,000 iterations and the autocorrelation is reduced by retaining only every 90th (150th) iteration of the chain. RW(1) model: the algorithm runs $2\text{e}{+6}$ iterations with burn-in of 20,000 iterations and the autocorrelation is reduced by retaining only every 1000th iteration of the chain. Model: D_i' takes on value 1 (0) with probability π_i $(1-\pi_i)$ in which $\text{logit}(\pi_i) = h_i$, $h_i = \boldsymbol{x}_{i-1}^\top \boldsymbol{\beta} + \phi(h_{i-1} - \boldsymbol{x}_{i-2}^\top \boldsymbol{\beta}) + \varepsilon_i$, $\varepsilon_i \sim \mathcal{N}(0, 1/\tau)$ and $\phi \in (0,1)$. Priors: $\tau \sim \text{Gamma}(1\text{e}{+3}, 1\text{e}{-3})$, $\boldsymbol{\beta} \sim \mathcal{N}_2(\mathbf{0}_2, 10^3\mathbf{I}_2)$.

	WN ($\phi = 0$)		RW(1) ($\phi = 1$)	
Parameter	Morning	Afternoon	Morning	Afternoon
σ	0.131	0.092	0.012	0.007
	(0.082)	(0.050)	(0.002)	(0.001)
β_0	1.581	2.387	1.549	2.430
	(0.038)	(0.020)	(0.114)	(0.117)
β_1	-3.198	-4.779	-3.207	-4.785
	(0.053)	(0.029)	(0.054)	(0.028)
Run-time	6.53	7319	29.16	514.95

Table 4.6 lists the parameter estimates of the price direction model, along with the running time (in hours). According to the estimation results, the estimates of the conditional standard deviation through RW(1) are much smaller than those from the white noise. The direction variables are negative, which suggests that the next active trade is less likely to move upwards if the previous trade moved upwards. Figure 4.10 illustrates the marginal posterior densities of the estimated parameters for the WN and RW(1) direction model using a kernel density, confined to the two time periods, along with the autologistic model (red curve).

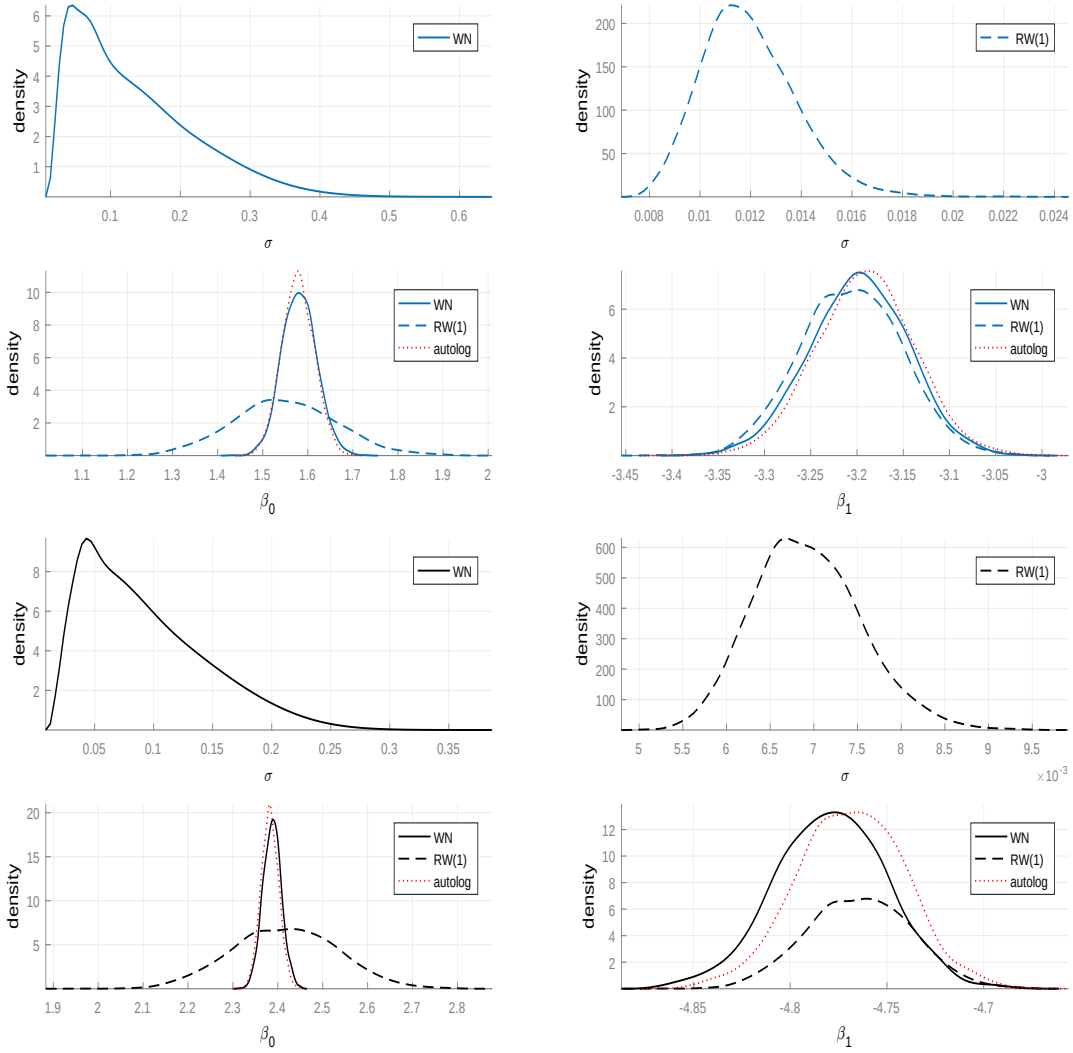


Figure 4.10: Marginal posterior densities (using a kernel density) of the estimated parameters for the WN (solid curve) and RW(1) (dashed curve) direction model, estimated with the algorithm CNC combined with the 3-block sampler in which the latent path is updated using the 'all' method, along with the autologistic model (red curve). The data set we are analyzing is ES, May 16th to May 20th, confined to the morning partition (blue curve) and afternoon partition (black curve), case (I).

4.5.2.2 Case II

In order to reduce the correlations, the regression parameters are updated in groups; namely, for the afternoon and morning period the groups are β_0, (β_1, β_2), (β_3, β_5), (β_4, β_6), and β_0, $(\beta_1, \beta_2, \beta_3, \beta_6, \beta_8)$, (β_4, β_7),(β_5, β_9), respectively. However, the autocorrelation of the estimated regression parameters is still high: during the morning the maximum value of the autocorrelation is 8000 (> 9000) for the WN (RW(1)), while during the afternoon it is 3000 (2000). Since the correlations involved in the proposal covariance matrix of the regression parameters influence the direction of the proposed movement, for each value of the conditional precision, we roughly estimate the correlation structure of this marginal posterior distribution and then run again the algorithm using this estimated correlation matrix. Concerning the WN model, the algorithm runs $4\text{e}{+6}$ ($3.2\text{e}{+6}$) iterations for the morning (afternoon) dataset with burn-in of 20,000 iterations and the autocorrelation is reduced by retaining only every 2000th (1600th) iteration of the chain. Concerning the RW(1) model, the algorithm runs $4\text{e}{+6}$ ($2.2\text{e}{+6}$) iterations with burn-in of 20,000 iterations and the autocorrelation is reduced by retaining only every 2000th (1100)th iteration of the chain. We use the CNC algorithm combined with the 3-block sampler in which the latent path is updated using the 'all' method, except for the white noise model during the afternoon in which the latent path is updated with the '1-1' method.

Table 4.7 lists the parameter estimates of the price direction model, along with the running time (in minutes). According to the estimation results, the estimates of the conditional standard deviation through RW(1) are much smaller than those from the WN. The influence of lagged price direction, is negative indicating that if the price moved on the last trade then there is a large chance that this movement will be reversed if there is an active trade. Concerning the impact of buy market orders, we observe a significant negative influence at lag one, hence the odds of an up movement are larger for sell market orders than buy market order at lag one. Bid (ask) volume at lag one reduces (increases) the chance that the price movement will be upwards, while the reverse is true at the second lag.

Table 4.7: MCMC estimates of parameters of the WN and RW(1) direction model for ES, May 16th to May 20th, case (II). 'Run-time' returns the execution time in minutes. The Bayesian estimates represent posterior means and standard deviations in parenthesis. WN model: the algorithm runs $4\text{e}{+6}$ ($3.2\text{e}{+6}$) iterations for the morning (afternoon) dataset with burn-in of 20,000 iterations and the autocorrelation is reduced by retaining only every 2000th (1600th) iteration of the chain. RW(1) model: the algorithm runs $4\text{e}{+6}$ ($2.2\text{e}{+6}$) iterations with burn-in of 20,000 iterations and the autocorrelation is reduced by retaining only every 2000 (1100)th iteration of the chain. Model: D_i' takes on value 1 (0) with probability π_i ($1-\pi_i$) in which $\text{logit}(\pi_i) = h_i$, $h_i = \boldsymbol{x}_{i-1}^\top \boldsymbol{\beta} + \phi(h_{i-1} - \boldsymbol{x}_{i-2}^\top \boldsymbol{\beta}) + \varepsilon_i$, $\varepsilon_i \sim \mathcal{N}(0, 1/\tau)$ and $\phi \in (0,1)$. Morning: $\boldsymbol{x}_{i-1}^\top = (1, D_{i-1}', \text{BMO}_{i-1}, V_{i-1}^{\text{b,1}}, V_{i-1}^{\text{b,2}}, V_{i-1}^{\text{a,2}}, V_{i-2}^{\text{b,1}}, V_{i-2}^{\text{b,2}}, V_{i-2}^{\text{a,1}}, V_{i-2}^{\text{a,2}})$ for the morning and Afternoon: $\boldsymbol{x}_{i-1}^\top = (1, D_{i-1}', \text{BMO}_{i-1}, V_{i-1}^{\text{b,1}}, V_{i-1}^{\text{a,1}}, V_{i-2}^{\text{b,1}}, V_{i-2}^{\text{a,1}})$. Priors: $\tau \sim \text{Gamma}(1\text{e}{+3}, 1\text{e}{-3})$, $\boldsymbol{\beta} \sim \mathcal{N}_d(\mathbf{0}_d, 10^3 \mathbf{I}_d)$.

	Morning		Afternoon	
Parameter	WN	RW(1)	WN	RW(1)
σ	0.225	0.016	1.680	0.017
	(0.198)	(0.003)	(0.142)	(0.003)
β_0	4.137	5.061	8.303	7.992
	(0.626)	(0.603)	(0.488)	(0.487)
β_1	-1.200	-1.183	-0.996	-0.733
	(0.118)	(0.113)	(0.137)	(0.108)
β_2	-6.166	-6.141	-12.056	-9.564
	(0.188)	(0.132)	(0.429)	(0.128)
β_3	-0.875	-0.863	-2.236	-1.899
	(0.048)	(0.045)	(0.082)	(0.048)
β_4	-3.317	-3.398	1.930	1.532
	(0.193)	(0.175)	(0.080)	(0.048)
β_5	3.358	3.309	1.913	1.536
	(0.213)	(0.196)	(0.071)	(0.036)
β_6	0.715	0.705	-1.894	-1.599
	(0.040)	(0.037)	(0.074)	(0.042)
β_7	2.687	2.643	-	-
	(0.180)	(0.165)	-	-
β_8	-0.447	-0.450	-	-
	(0.041)	(0.040)	-	-
β_9	-2.303	-2.310	-	-
	(0.207)	(0.194)	-	-
Run-time	159.69	259.63	589.77	961.46

Figures 4.11 - 4.12 illustrate the marginal posterior densities of the estimated parameters

for the WN and RW(1) direction model using a kernel density, along with the autologistic model (red curve). It is observed that the posterior distributions of the regression parameters, through the three models, are almost identical; an exception arises during the afternoon from the white noise model.

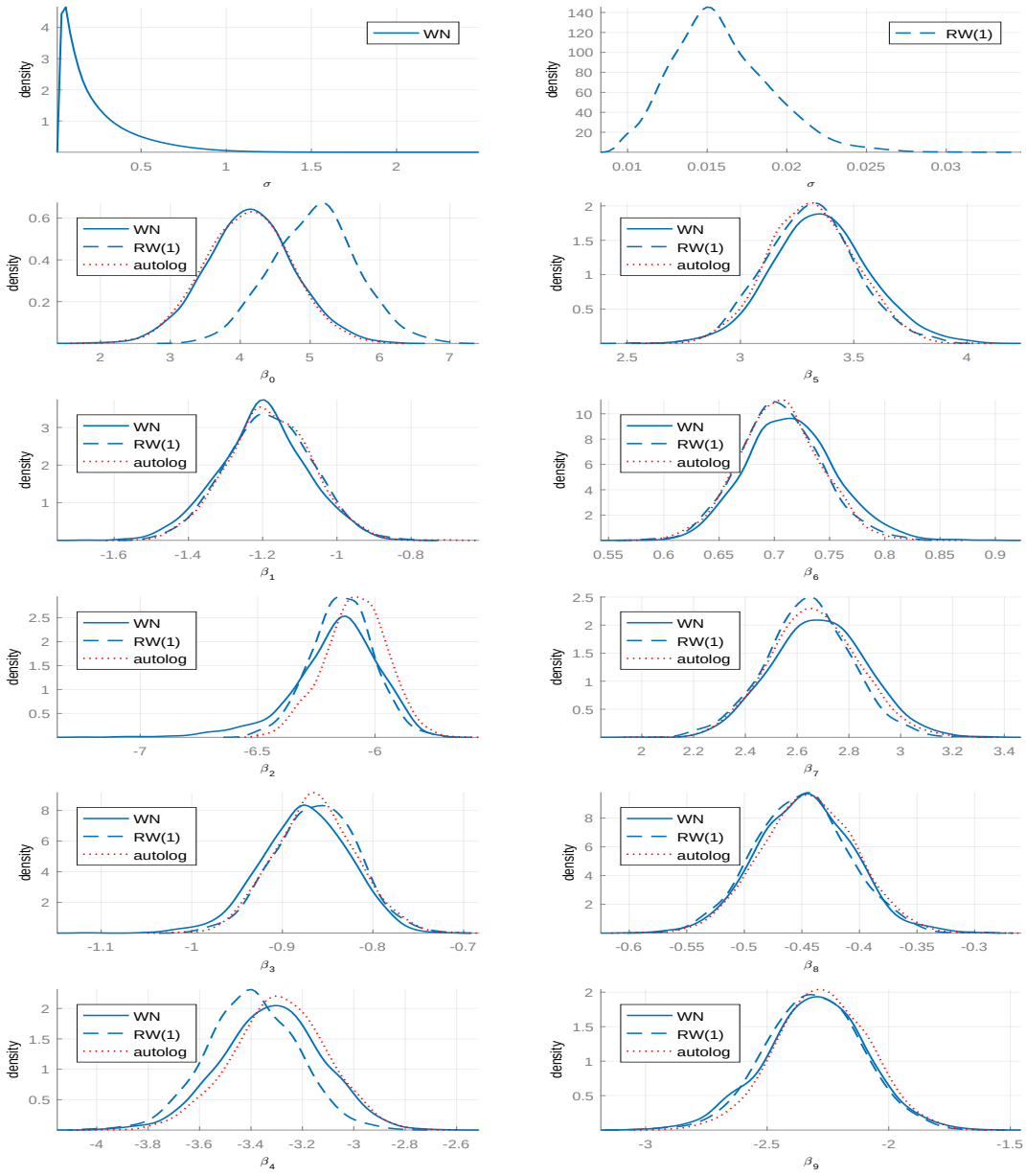


Figure 4.11: Marginal posterior densities (using a kernel density) of the estimated parameters for the WN (solid curve) and RW(1) (dashed curve) direction model, estimated with the algorithm CNC combined with the 3-block sampler in which the latent path is updated using the 'all' method, along with the autologistic model (red curve). The data set we are analyzing is ES, May 16th to May 20th, confined to the morning partition, case (II).

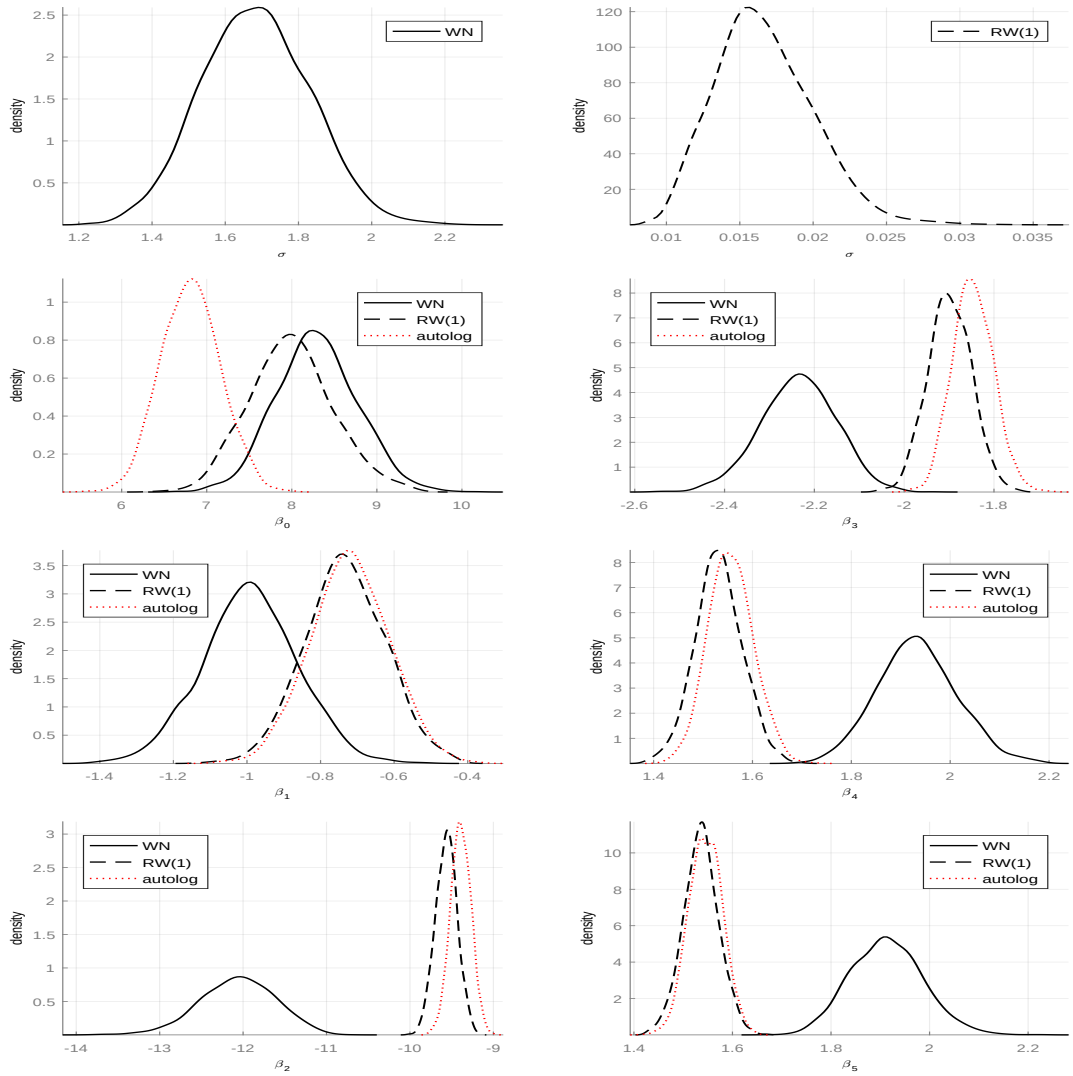


Figure 4.12: Marginal posterior densities (using a kernel density) of the estimated parameters for the WN (solid curve) and RW(1) (dashed curve) direction model, estimated with the algorithm CNC combined with the 3-block sampler in which the latent path is updated using the 'all' method, except for the white noise model in which the latent path is updated with the '1-1' method, along with the autologistic model (red curve). The data set we are analyzing is ES, May 16th to May 20th, confined to the afternoon partition case (II).

4.5.3 Predicting price movements

In the section we analyze the predictive performance of each price component and predict price changes from May 23 to May 24 during the morning and afternoon period. We report each day separately allowing us to assess the predicted accuracy over the two days separately.

Table 4.8: Brier score (BS) and log predictive density (LPS) of the active model of ES for the evaluation period May 23 to May 24.

	Case (I)				Case (II)			
	Morning		Afternoon		Morning		Afternoon	
Model	May 23	May 24	May 23	May 24	May 23	May 24	May 23	May 24
Panel A: BS								
GLM	0.153	0.145	0.151	0.148	0.145	0.137	0.145	0.144
GLARMA	0.156	0.148	0.156	0.156	0.148	0.139	0.149	0.147
WN	0.153	0.145	0.151	0.148	0.145	0.137	0.145	0.144
Panel B: LPS								
GLM	-6260.0	-4111.0	-30119.0	-26775.0	-5877.6	-3812.0	-28752.0	-25634.0
GLARMA	-6375.8	-4185.3	-31139.0	-27637.0	-5999.1	-3904.8	-29440.0	-26266.0
WN	-6260.0	-4111.0	-30119.0	-26775.0	-5876.6	-3812.0	-28752.0	-25635.0

We apply the MSE, the MAE and the Brier score [Brier, 1950; Blattenberger and Lad, 1985] for model comparison which are specified as

$$BS = \frac{\sum_{i=n+1}^{n'}(\hat{\pi}_i - o_i)^2}{n'}, \text{ MSE} = \frac{\sum_{i=1}^{n'}(Z_i^o - \hat{Z}_i)^2}{n'}, \text{ MAE} = \frac{\sum_{i=1}^{n'}|Z_i^o - \hat{Z}_i|}{n'},$$

where $\hat{\pi}_i$ denoted the predicted probability and o_i is the true value associated with the ith observation, Z_i^o and $\hat{Z}_i$ denote the true and predicted value associated with the ith transaction as well as n' is the number of forecasting instances. The quantity $\hat{\pi}_i$ denotes the Monte Carlo [Metropolis and Ulam, 1949; von Neumann, 1963] average of the probability π_{A_t} given by

$$\hat{\pi}_i = \frac{1}{N}\sum_{l=1}^{N}\frac{\exp(\text{logit}(\pi_i^{(l)}))}{1+\exp(\text{logit}(\pi_i^{(l)}))},$$

where $\text{logit}(\pi_i^{(l)})$ is the sampled value of $\text{logit}(\pi_i)$ in iteration l, after burn in. The lower the BS/MSE/MAE value the better and 0 means the model is perfect. Besides, we compare the predicted observations with their values changes in the original sample data by using scalar performance measures, namely, accuracy, sensitivity (or recall), precision and specificity derived from the confusion matrix [Hajmeer and Basheer, 2003]. Accuracy is a ratio of correctly

predicted observations to the total observations. Sensitivity (of the jth event) is the ratio of j-events correctly predicted. In fact, out of all the j-events, how many of them have been predicted by the algorithm. Specificity (of the jth event) is the ratio of not j-events correctly predicted. Precision (of the jth event) is the percentage of correct j-predictions out of all j-predictions, which are used to calculate the model's ability to classify j-predictions correctly.

Tables 4.8-4.9 provide the BS (panel A) and the log predictive likelihood (panel B) for active trades and the direction of them, respectively. The tables consist of three main columns; the first column gives the process that is used for each price component, while the last two main columns contain the line headings of the information set that is used and they are splitted into two sub-columns with names corresponding to the morning and afternoon time period. Regarding the direction model, the results show that the predictive ability of the proposed models is the same, indicating that there is not much difference between the models. On the other hand, the performance of the GLM and WN model for the activity process is the same. The BS of the best performance model (for each component) shows it is almost two times smaller than a BS of value 0.25 which indicates a random guess. Judging by the information set, the BS of the direction model using only past values of the processes is 3.5 (7) times smaller than the corresponding wider information set during the morning (afternoon) period. However, the improvement of the activity model is significantly lower.

Table 4.9: Brier score (BS) and log predictive density (LPS) of the direction model of ES for the evaluation period May 23 to May 24.

	Case (I)				Case (II)			
	Morning		Afternoon		Morning		Afternoon	
Model	May 23	May 24	May 23	May 24	May 23	May 24	May 23	May 24
Panel A: BS								
GLM	0.128	0.156	0.084	0.079	0.036	0.050	0.012	0.012
WN	0.128	0.156	0.084	0.079	0.036	0.050	0.012	0.012
Panel B: LPS								
GLM	-1166.3	-854.9	-4282.6	-3631.8	-401.3	-308.3	-689.7	-611.4
WN	-1165.9	-854.8	-4282.8	-3631.8	-399.4	-307.6	-688.9	-610.7

Table 4.10: Prediction accuracy of the activity and direction model for the evaluation period May 23 to May 24. The results are presented as percentages. Notes: Inactive (no active trade), Sensitivity (sens), specificity (spec), precision (prec) and accuracy (Acc).

	May 23						May 24					
	Morning			Afternoon			Morning			Afternoon		
Class	Sens	Prec	Spec	Sens	Prec	Spec	Sens	Prec	Spec	Sens	Prec	Spec
Panel A: Plain activity model												
inact	83.34	84.96	43.59	84.11	84.28	40.94	84.40	85.87	40.97	84.99	84.76	41.76
active	43.59	40.62	83.34	40.94	41.55	84.61	40.97	38.20	84.40	41.76	42.20	85.00
Acc	75.10			75.40			76.13			76.01		
Panel B: Activity model with covariates												
inact	87.48	86.06	42.40	87.04	84.85	41.36	86.90	87.42	44.65	86.42	85.31	42.92
active	42.40	45.45	87.48	41.36	45.82	87.24	44.65	43.50	86.90	42.92	45.16	86.42
Acc	78.58			77.47			79.11			77.43		
Panel C: Direction model with covariates												
down	95.44	94.82	94.76	98.60	98.43	98.42	94.59	91.83	91.65	98.64	98.49	98.48
up	94.76	95.38	95.44	98.42	98.60	98.60	91.65	94.46	94.59	98.48	98.63	98.64
Acc	95.10			98.51			93.11			98.56		

Accuracy, sensitivity, specificity, and precision measures for all the price components are summarized in Table 4.9 using the two different information sets. In the following we only present the results for May 23 which are similar to May 24. We use the activity GLM and the direction GLM. If we do not make a specific reference to which data segment we are referring to, it means that we imply both examined time periods; the same is true for the different trade bar sizes. About the activity model, it can be seen that the ability of the model to classify correctly the trade activity is over 75% for the plain model, and it improves 3% with the wider set. Regarding the sensitivity, the results show that the active trades are harder to classify than the inactive. More specifically, 6 of every 10 active trades, in reality, are missed by our model and 4 are correctly identified as active. On the other hand, less than 20% of non-active trades are incorrectly classified as active. Regarding the precision, out of the total observations that the

model predicts as active (inactive) trades over 40% (85%) are correct. Regarding the specificity, the model allows to identify active moves as inactive at a rate less than 60%, and non active trades as active at a rate less than 20%. The corresponding measures using the wider set is a little better. About the direction model, the model has a good accuracy, sensitivity and specificity which are over 90%, and some of them even reach 99%.

Table 4.11: Prediction accuracy of the Bernoulli AD model for the evaluation period May 23 to May 24. The results are presented as percentages. Notes: Down (one tick down), Zero (zero move), Up (one tick up), Sensitivity (sens), specificity (spec), precision (prec) and accuracy (Acc). Lag1 reports the MSE (MAE) values using a model that assume that the best prediction for tomorrow's market price is simply today's.

	May 23						May 24					
	Morning			Afternoon			Morning			Afternoon		
Class	Sens	Prec	Spec	Sens	Prec	Spec	Sens	Prec	Spec	Sens	Prec	Spec
down	40.00	34.78	93.18	43.80	43.92	93.28	37.90	38.71	93.57	43.44	43.59	93.34
zero	84.21	84.22	40.65	84.32	84.32	42.57	85.11	85.11	38.17	84.73	84.73	43.06
up	39.45	38.58	92.65	40.65	40.54	92.84	35.70	34.98	92.84	42.00	41.85	93.12
Acc	74.86			75.29			75.73			75.85		
MSE	0.259			0.243			0.253			0.243		
MSE-lag1	0.581			0.609			0.523			0.603		
MAE	0.254			0.248			0.247			0.242		
MAE-lag1	0.415			0.428			0.381			0.422		

Considering the execution time, since the BSs of the best parameter and observation driven models are almost equal we conclude that it is better to use a Bernoulli GLM for both price factors. About the information set, the wider set is clearly superior than the simpler set for the direction model, meanwhile, for the activity model the improvement is slightly better. Table 4.11 summarizes the accuracy, sensitivity, specificity, and precision measures for all the price changes using the activity GLM and direction GLM. The results show that the price direction moves are harder to classify than the zero moves, while on the afternoon the predictive measures are slightly better. More specifically, the ability of the model to classify correctly the price changes is 75%. Less than 20% of non-zero moves are incorrectly classified as zero moves,

while 6 of every 10 up (down) moves, in reality, are missed by our model and 4 are correctly identified as up (down) moves. Finally, more than 85% out of zero predictive moves are correctly classified, while this value drops to over 40% in identifying the non-zero price moves.

Chapter 5

The Binomial AD model

5.1 Introduction

Following the idea of decomposition, we propose a new parameter and observation driven model for analyzing high-frequency integer price changes. The model is applied on data where a transaction can move the stock price either one tick up or one tick down or not at all. We aggregate observations over a small number of trade intervals (denoted trade bars), assign each interval the last price included in it, and then use the price change between the two consecutive intervals for analyzing the process of price changes. This chapter is organized as follows. Section 5.2 presents the binomial AD model and Section 5.3 analyses the proposed model within a Bayesian framework via MCMC. Section 5.4 presents the results of a simulation study. Section 5.5 provides the data, introduces variables for modelling price changes, discusses the empirical results and evaluates the predictive performance of the proposed model.

5.2 The binomial AD model

In our dataset, price movements of more than one tick occur about 0.01% of the time. If we observe a transaction that moves the price greater than one tick, for example two ticks, it is treated as two transactions where each transaction moves the price by one tick. Let A^j be a binary (takes only two values) variable on $\{0, 1\}$ defining the market activity (the price does moves or not) and a binary variable D^j on $\{0, 1\}$ defining a negative or positive price move (if a change occurs) of the jth transaction. We split the observations into trade bars of size N_i, and during each bar we aggregate the binary observations. We assume that the variables A^j and D^j

that belong to the ith trade bar follows a Bernoulli distribution with the same probability π_{A_i} and π_{D_i}, respectively. Let A_i and D_i denote the number of active trades and the number of up moves (if at least one change occurs) during the ith trade bar, respectively. Besides, P_{t_i} denotes the transaction price of the last price included in the i trade bar which is recorded at time t_i. Under these definitions, the transaction price evolves over time by

$$P_{t_i} = P_{t_i} + 2D_i - A_i,$$

where $D_i = 0$ if $A_i = 0$. Here we choose $N_i = N, \forall i.\ A_i \in \{0, \cdots, N\}$ and $D_i|A_i > 0 \in \{1, \cdots, A_i\}$.

The conditional joint distribution of (A_i, D_i) is decomposed as

$$\pi(A_i, D_i|\mathcal{F}_{i-1}) = \pi(A_i|\mathcal{F}_{i-1})\pi(D_i|A_i > 0, \mathcal{F}_{i-1}),$$

where $\mathcal{F}_{i-1}$ is a σ-field generated either by past values of the observed series (observation driven models) or by a latent process (parameter driven models), as well as past or present values of the covariates.

5.2.1 Models for the component factors

We model the logit of the probability of success at each time point of each price factor as a linear function of regression variables, and a process $\{g_i\}$ based on the binomial distribution. Firstly, we consider a GLARMA process as described in Section 3.2.1. Secondly, we examine an AR(1), WN and a RW(1) latent process with Gaussian (see Section 4.2) and Student's t errors.

All the price changes, Y_i, can be classified into one of the following categories: no price change with probability $\pi(Y_i = 0|\mathcal{F}_{i-1}) = \pi(A_i = 0|\mathcal{F}_{i-1}) + \sum_{l=1}^{[\frac{N}{2}]} \pi(A_i = 2l|\mathcal{F}_{i-1})\pi(D_i = l|A_i = 2l, \mathcal{F}_{i-1})$, an upward or downward price change of size $y_i \in \{1, \cdots, N\}$ with probability $\pi(Y_i = y_i|\mathcal{F}_{i-1}) = \sum_{l=0}^{[\frac{N-y_i}{2}]} \pi(A_i = y_i + 2l|\mathcal{F}_{i-1})\pi(D_i = y_i + l|A_i = y_i + 2l, \mathcal{F}_{i-1})$ and $\pi(Y_i = -y_i|\mathcal{F}_{i-1}) = \sum_{l=0}^{[\frac{N-y_i}{2}]} \pi(A_i = y_i + 2l|\mathcal{F}_{i-1})\pi(D_i = l|A_i = y_i + 2l, \mathcal{F}_{i-1})$, respectively; $[u]$ truncates the fractional part of the number u towards zero.

The overall log-likelihood of the binomial AD model is given by equation (3.8), and

$$\ell(\boldsymbol{\theta}_{\mathrm{A}}) \triangleq \sum_{i \in 1:n} \log \pi(A_i = \alpha_i|\mathcal{F}^{\mathrm{A}}_{i-1}, \boldsymbol{\theta}_{\mathrm{A}}),\ \ \ell(\boldsymbol{\theta}_{\mathrm{D}}) \triangleq \sum_{\substack{i \in 1:n \\ \forall i: A_i > 0}} \log \pi(D_i = d_i|A_i > 0, \mathcal{F}^{\mathrm{D}}_{i-1}, \boldsymbol{\theta}_{\mathrm{D}}),$$

where $\alpha_i \in \{0, \cdots, N\}$ and $d_i \in \{1, \cdots, A_i\}$.

5.3 Bayesian analysis of the AR(1) component model with Student's t errors

The general estimation procedure (as well as the predictive performance) is similar with its corresponding model with Bernoulli observations that described in the previous chapters. For this reason, is presented only the proposed model with Student's t errors and some details, for example, the posterior distribution of the estimated parameters.

The binomial AR(1) activity model in its centered form is given through

$$A_i | h_i^{\mathrm{A}} \sim \mathrm{Binomial}(N, \pi_{A_i}) \tag{5.1}$$

$$\mathrm{logit}(\pi_{A_i}) = h_i^{\mathrm{A}} \tag{5.2}$$

$$h_i^{\mathrm{A}} = \boldsymbol{x}_{i-1}^{\mathrm{A},\top}\boldsymbol{\beta}_{\mathrm{A}} + \phi_{\mathrm{A}}(h_{i-1}^{\mathrm{A}} - \boldsymbol{x}_{i-2}^{\mathrm{A},\top}\boldsymbol{\beta}_{\mathrm{A}}) + \sigma_i^{\mathrm{A}}\sqrt{\lambda_i^{\mathrm{A}}}\varepsilon_i^{\mathrm{A}}, \;\; \varepsilon_i^{\mathrm{A}} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1), \tag{5.3}$$

where $\pi_{A_i} \triangleq \pi(A_i = 1 | h_i^{\mathrm{A}})$ and $h_1^{\mathrm{A}} | \phi_{\mathrm{A}}, \sigma_{\mathrm{A}}, \lambda_1^{\mathrm{A}} \sim \mathcal{N}(0, \lambda_1^{\mathrm{A}}\sigma_{\mathrm{A}}^2/(1-\phi_{\mathrm{A}}^2))$. Note that $\eta_i \triangleq \sqrt{\lambda_i^{\mathrm{A}}}\varepsilon_i^{\mathrm{A}} \sim t_\nu(0,1)$ and $\lambda_i^{\mathrm{A}} \sim \mathrm{G}^{-1}(\nu/2, \nu/2)$, where $t_\nu(0,1)$ denotes the Student t distribution with mean 0, variance 1 and ν degrees of freedom, $\mathrm{G}^{-1}(\alpha, \beta)$ denotes the inverse gamma distribution with shape and scale parameters α and β, respectively.

The joint distribution of the latent path, $\boldsymbol{h}_{\mathrm{A}} | \boldsymbol{\theta}_{\mathrm{A}}, \boldsymbol{\lambda}_{\mathrm{A}}$, follows a multivariate normal distribution with mean $\boldsymbol{m}_{\mathrm{A}}$ and precision matrix $\boldsymbol{Q}_{\mathrm{A}}$ $(n \times n)$ which is a tridiagonal matrix in which the primary diagonal is formed by the elements $\tau_{\mathrm{A}}\{(1-\phi_{\mathrm{A}}^2)/\lambda_1^{\mathrm{A}} + \phi_{\mathrm{A}}^2/\lambda_2^{\mathrm{A}}, (1/\lambda_2^{\mathrm{A}} + \phi_{\mathrm{A}}^2/\lambda_3^{\mathrm{A}}), \cdots, (1/\lambda_{n-1}^{\mathrm{A}} + \phi_{\mathrm{A}}^2/\lambda_n^{\mathrm{A}}), 1/\lambda_n^{\mathrm{A}}\}$, while the diagonal above the principal diagonal is the vector $\tau_{\mathrm{A}}\{-\phi_{\mathrm{A}}/\lambda_2^{\mathrm{A}}, -\phi_{\mathrm{A}}/\lambda_3^{\mathrm{A}}, \cdots, -\phi_{\mathrm{A}}/\lambda_n^{\mathrm{A}}, \}$, where and $\boldsymbol{\lambda}_{\mathrm{A}} \triangleq (\lambda_1^{\mathrm{A}}, \cdots, \lambda_n^{\mathrm{A}})$. In the 3-block sampler, ϕ_{A}, τ_{A} and the blocks of $\boldsymbol{\beta}_{\mathrm{A}}$, $\boldsymbol{\lambda}_{\mathrm{A}}$ and ν are sampled separately. In C, the parameters τ_{A} and $\boldsymbol{\beta}_{\mathrm{A}}$ are drawn with Gibbs from their corresponding conditional posterior given by

$$\tau_{\mathrm{A}} | \boldsymbol{h}_{\mathrm{A}}, \boldsymbol{\alpha}, \phi_{\mathrm{A}}, \boldsymbol{\beta}_{\mathrm{A}} \sim \mathrm{Gamma}\left(\gamma_0 + \frac{n}{2}, \frac{1}{G}\right), \;\; \boldsymbol{\beta}_{\mathrm{A}} | \boldsymbol{h}_{\mathrm{A}}, \boldsymbol{\alpha}, \phi_{\mathrm{A}}, \tau_{\mathrm{A}} \sim \mathcal{N}_{d_{\mathrm{A}}}\left(\boldsymbol{\mu}, \boldsymbol{\Sigma}^{-1}\right),$$

where $G \triangleq \delta_0 + \frac{1}{2}\sum_{i=1}^{n} \frac{(h_{*i}^{\mathrm{A}} - \boldsymbol{x}_{*i}^{\mathrm{A},\top}\boldsymbol{\beta}_{\mathrm{A}})^2}{\lambda_i}$, $\boldsymbol{\mu} \triangleq \tau_{\mathrm{A}}\boldsymbol{\Sigma}^{-1}\boldsymbol{x}_*^{\mathrm{A}}\Lambda\boldsymbol{h}_*^{\mathrm{A}}$ and $\boldsymbol{\Sigma} \triangleq \frac{1}{\sigma_0^2}\mathbb{I}_{d_{\mathrm{A}}} + \tau_{\mathrm{A}}\boldsymbol{x}_*^{\mathrm{A}}\Lambda\boldsymbol{x}_*^{\mathrm{A},\top}$; Λ

is a diagonal matrix with all diagonal elements the vector $(\lambda_1^{\mathrm{A}}, \cdots, \lambda_n^{\mathrm{A}})$. The marginal posterior of λ_i^{A} is given through

$$\lambda_i^{\mathrm{A}} | \boldsymbol{h}_{\mathrm{A}}, \boldsymbol{\alpha}, \phi_{\mathrm{A}}, \boldsymbol{\beta}_{\mathrm{A}}, \boldsymbol{\lambda}_{-i} \sim \mathrm{G}^{-1}\left(\frac{\nu+1}{2}, \frac{\nu}{2} + \frac{\tau_{\mathrm{A}}(h_{*i}^{\mathrm{A}} - \boldsymbol{x}_{*i}^{\mathrm{A},\top}\boldsymbol{\beta}_{\mathrm{A}})^2}{2}\right),$$

in which $\boldsymbol{\lambda}_{-i}$ denotes the elements of $\boldsymbol{\lambda}_{\mathrm{A}}$ excluding λ_i^{A}. In NC, the conditional posterior density of ϕ_{A} is not known in closed form, thus it is approximated with a Gaussian random walk Metropolis algorithm, while the marginal posterior of λ_i^{A} is given through

$$\lambda_i^{\mathrm{A}} | \tilde{\boldsymbol{h}}_{\mathrm{A}}, \boldsymbol{\alpha}, \phi_{\mathrm{A}}, \boldsymbol{\lambda}_{-i}, \nu \sim \mathrm{G}^{-1}\left(\frac{\nu+1}{2}, \frac{\nu + (\tilde{h}_i^{\mathrm{A}} - \phi_{\mathrm{A}}\tilde{h}_{i-1}^{\mathrm{A}})^2}{2}\right), \quad i \geq 2$$

while its scale parameter equals $(\nu + (1-\phi_{\mathrm{A}}^2)\tilde{h}_1^{\mathrm{A}})/2$ for $i = 1$. In both parameterizations, due to non-conjugacy of the chosen priors, ϕ_{A} and ν are updated with the Gaussian random walk Metropolis algorithm. The full conditional posterior of the degrees of freedom parameter, only depends on $\boldsymbol{\lambda}_{\mathrm{A}}$, and its log density is given through

$$\log \pi(\nu | \boldsymbol{\lambda}_{\mathrm{A}}) \propto \log \pi(\nu) + \log \pi(\boldsymbol{\lambda}_{\mathrm{A}} | \nu),$$

where $\log \pi(\boldsymbol{\lambda}_{\mathrm{A}} | \nu) = \frac{\nu n}{2}\log(\nu/2) - n\log(\Gamma(\nu/2)) - \frac{\nu}{2}\sum_{i=1}^{n}(\log(\lambda_i^{\mathrm{A}}) + 1/\lambda_i^{\mathrm{A}})$ and $\pi(\nu)$ denotes the prior density of ν.

In the interweaving sampler, the reparameterization between the centered and non-centered is the same as described in the previous section, hence the degrees of freedom as well as the auxiliar vector $\boldsymbol{\lambda}_{\mathrm{A}}$ are not redrawn.

5.4 Simulation study

We perform a simulation study to investigate the effectiveness of our algorithms similar to the one described in Section 4.4. Suppose that the random variables $A_1, \cdots, A_n$ are independent with conditional distribution $A_i | h_i \sim \mathrm{Binomial}(N, \pi_i)$, where

$$\mathrm{logit}(\pi_i) = h_i,$$
$$h_i = \beta_0 + \phi(h_{i-1} - \beta_0) + \varepsilon_i, \;\; \varepsilon_i \sim \mathcal{N}(0, 1/\tau)$$

$h_1 \sim \mathcal{N}(\beta_0, 1/\tau(1-\phi^2))$ and $i \in \{1, 2, \cdots, n\}$. The parameter β_0 is equal to -2 and the parameters ϕ and τ vary on $\{0, 0.55, 0.65, 0.75, 0.85, 0.95\} \times \{1, 2, 5, 10, 30\}$. The sample size, n, is fixed to $6,000$ and $N \in \{2, 5, 10\}$.

To speed up the convergence, we use priors with means equalling the true values, more specifically, following Kastner and Frühwirth-Schnatter [2014], the transformed parameter ψ is assumed to follow a Beta distribution with parameters (a_0, b_0), where $a_0 = 40$ and $b_0 = 80/(1+\phi_{\text{true}}) - 40$. For the parameter τ, we use a gamma distribution with parameters (γ_0, δ_0), where $\gamma_0 = 10$ and $\delta_0 \in \{0.1, 0.2, 0.5, 1, 3\}$ for $\tau \in \{1, 2, 5, 10, 30\}$, respectively. Finally, for the constant parameter, we use a normal distribution with mean β_{true} and variance 10.

We use $N = 100,000$ MCMC draws after a burn-in of $30,000$ for each data set. Starting values are set to the true values. We apply four sampling schemes: C, NC, CNC, NCC. Besides, we use the 2-block and 3-block sampler, the '1-1' and 'all' method. The performance of the MCMC sampling methods are evaluate by the ESS/sec.

The simulation results for all datasets and algorithms considered are presented in Tables 5.1 - 5.9 along with the mean running time (in minutes). The quarantines in the tables are averaged over twenty runs under different seeds initializations. The mean running time (in minutes) that describes the amount of time it takes for the C(1-1), NC(1-1), CNC(1-1), NCC(1-1), C(all), NC(all), CNC(all), NCC(all) algorithm to complete 130,000 iterations is 2.78, 2.82, 3.19, 3.12, 2.05, 2.03, 2.48, 2.32 for the 3-block sampler and 2.79, 2.86, 3.30, 3.24, 2.01, 2.09, 2.51,2.47 for the 2-block sampler, respectively.

Concerning the simulation efficiency of ϕ (Tables 5.1, 5.4, 5.7), the results show that keeping ϕ fixed, it seems that as the value of τ increases, the sampling efficiency of ϕ deteriorates. Keeping τ fixed, as the value of ϕ decreases, the sampling efficiency gradually decline, and, in some cases rise when $\phi = 0$. Note that, as ϕ drops form 0.95 to 0.85, the sampling efficiency sometimes increases. Besides, as the number of N decreases, the sampling efficiency of ϕ decreases. Comparing the sampling schemes, it seems that the interweaving samplers exhibit lower autocorrelation (and therefore a higher ESS/sec), while the centered parameterization gives similar values in some cases, for most underlying parameter values. Comparing the latent path strategy, the results show that as the value of the conditional precision increases it is preferable to update the components of the latent state in one step, but the value of τ for this change depends on the value of N; as N increases, this value increases. Finally, by comparing

Table 5.1: Performance of MCMC samplers as measured by ESS/sec for $\pi(\phi|\boldsymbol{h}, \boldsymbol{\alpha}, \tau, \beta_0)$ for the simulated datasets where $N = 2$. Higher is better. The quantities are averaged over twenty runs under different seeds initializations. All algorithms run 130,000 iterations after a burn-in of $30,000$ for each data set. The latent state is updated using '1-1' and 'all' method. The parameter vector is updated by applying the '2-bl' and '3-bl' sampler.

Alg.	τ									
	1		2		5		10		30	
	3-bl	2-bl	3-bl	2-bl	3-bl	2-bl	3-bl	2-bl	3-bl	2-bl
$\phi = 0.95$										
C(1-1)	5.63	3.12	4.22	2.35	2.25	1.23	1.07	0.61	0.32	0.27
NC(1-1)	1.90	1.34	1.26	1.09	0.72	0.59	0.46	0.50	0.20	0.17
CNC(1-1)	5.38	3.58	4.67	2.73	2.56	1.80	1.05	0.99	0.44	0.32
NCC(1-1)	5.13	3.49	4.18	2.69	2.40	1.42	0.92	0.86	0.29	0.36
C(all)	1.43	1.25	1.67	2.08	1.58	1.72	1.92	1.61	1.46	0.80
NC(all)	0.27	0.75	0.41	0.65	0.64	0.89	0.64	1.09	1.35	1.27
CNC(all)	2.20	1.53	2.18	1.80	2.38	1.90	1.85	1.99	1.93	1.45
NCC(all)	1.48	0.83	2.17	1.20	2.15	1.51	2.31	1.46	1.57	1.30
$\phi = 0.85$										
C(1-1)	4.77	2.32	2.01	1.21	1.06	0.40	0.63	0.36	0.25	0.20
NC(1-1)	2.66	2.62	1.44	1.12	0.71	0.72	0.30	0.36	0.18	0.16
CNC(1-1)	5.05	3.65	2.68	2.16	1.14	0.94	0.57	0.42	0.26	0.29
NCC(1-1)	5.35	3.78	2.63	1.89	1.05	0.83	0.57	0.45	0.23	0.21
C(all)	1.51	1.43	1.50	1.14	1.14	0.77	0.86	0.70	0.81	0.76
NC(all)	0.47	0.56	0.41	0.70	0.49	0.65	0.68	0.74	0.91	1.02
CNC(all)	1.35	1.10	1.36	1.05	1.05	0.97	0.99	0.87	0.95	0.81
NCC(all)	1.42	1.31	1.17	1.06	0.98	0.84	1.03	0.92	0.81	0.69
$\phi = 0.75$										
C(1-1)	3.29	1.58	1.57	0.69	0.80	0.58	0.53	0.34	0.31	0.22
NC(1-1)	2.88	2.39	1.38	1.10	0.63	0.43	0.42	0.35	0.22	0.22
CNC(1-1)	4.34	3.35	1.97	1.64	0.82	0.77	0.50	0.44	0.28	0.26
NCC(1-1)	4.14	2.77	1.95	1.57	0.74	0.61	0.50	0.45	0.19	0.18
C(all)	1.06	1.08	0.72	0.75	0.73	0.50	0.65	0.48	0.88	0.55
NC(all)	0.47	0.55	0.54	0.60	0.54	0.62	0.66	0.72	0.64	0.82
CNC(all)	1.11	1.04	1.01	0.73	0.68	0.66	0.82	0.72	0.73	0.85
NCC(all)	1.09	0.93	0.94	0.77	0.72	0.62	0.74	0.63	0.69	0.60
$\phi = 0.65$										
C(1-1)	2.92	1.90	1.45	1.05	0.68	0.36	0.59	0.29	0.32	0.29
NC(1-1)	2.71	1.22	1.36	0.50	0.66	0.62	0.48	0.32	0.25	0.25
CNC(1-1)	3.14	2.74	1.45	1.47	0.72	0.55	0.51	0.45	0.32	0.28
NCC(1-1)	3.54	2.84	1.43	1.48	0.64	0.69	0.48	0.33	0.29	0.38
C(all)	0.68	0.86	0.62	0.67	0.56	0.50	0.74	0.45	0.72	0.53
NC(all)	0.59	0.58	0.42	0.56	0.57	0.60	0.63	0.60	0.59	0.59
CNC(all)	0.79	0.82	0.81	0.67	0.67	0.70	0.68	0.70	0.88	0.79
NCC(all)	0.85	0.86	0.76	0.63	0.69	0.59	0.72	0.61	0.80	0.61
$\phi = 0.55$										
C(1-1)	2.63	2.14	1.26	1.11	0.87	0.65	0.60	0.46	0.45	0.32
NC(1-1)	2.55	1.06	1.10	0.54	0.78	0.38	0.57	0.38	0.42	0.33
CNC(1-1)	3.18	2.66	1.44	1.34	0.72	0.64	0.44	0.45	0.41	0.35
NCC(1-1)	3.08	2.44	1.37	1.26	0.67	0.82	0.49	0.51	0.31	0.31
C(all)	0.74	0.85	0.61	0.56	0.60	0.54	0.71	0.60	1.02	0.53
NC(all)	0.61	0.64	0.50	0.59	0.60	0.58	0.77	0.57	0.85	0.46
CNC(all)	0.82	0.71	0.71	0.61	0.79	0.62	0.61	0.72	0.61	0.83
NCC(all)	0.78	0.72	0.64	0.61	0.66	0.64	0.72	0.57	0.79	0.82
$\phi = 0.0$										
C(1-1)	3.77	2.58	2.15	1.53	1.46	1.06	1.15	0.99	1.11	0.92
NC(1-1)	2.66	1.64	1.92	0.91	1.24	0.64	1.10	0.60	0.90	0.48
CNC(1-1)	2.76	3.01	1.60	1.99	0.87	1.17	0.94	1.02	0.85	1.07
NCC(1-1)	3.27	3.47	1.74	1.75	1.17	1.22	0.91	1.08	0.96	1.11
C(all)	0.98	0.90	0.92	0.85	1.03	0.74	1.08	1.09	1.22	1.15
NC(all)	0.91	0.91	0.84	0.64	0.89	0.75	0.99	0.70	1.16	0.84
CNC(all)	0.90	0.78	0.81	0.65	0.95	0.92	0.94	0.96	0.76	1.14
NCC(all)	0.73	0.79	0.48	0.80	0.74	0.84	1.17	1.03	1.11	1.46

Table 5.2: Performance of MCMC samplers as measured by ESS/sec for $\pi(\tau|\boldsymbol{h}, \boldsymbol{\alpha}, \phi, \beta_0)$ for the simulated datasets where $N = 2$. Higher is better. The quantities are averaged over twenty runs under different seeds initializations. All algorithms run 130,000 iterations after a burn-in of $30,000$ for each data set. The latent state is updated using '1-1' and 'all' method. The parameter vector is updated by applying the '2-bl' and '3-bl' sampler.

Alg.	τ									
	1		2		5		10		30	
	3-bl	2-bl	3-bl	2-bl	3-bl	2-bl	3-bl	2-bl	3-bl	2-bl
$\phi = 0.95$										
C(1-1)	1.21	0.85	1.12	0.80	0.71	0.56	0.50	0.34	0.19	0.14
NC(1-1)	1.43	1.08	0.82	0.78	0.55	0.46	0.39	0.40	0.27	0.18
CNC(1-1)	5.14	3.75	3.40	2.17	1.80	1.26	0.95	0.82	0.36	0.39
NCC(1-1)	4.83	3.62	3.25	2.16	1.86	1.11	0.88	0.68	0.51	0.39
C(all)	0.38	0.43	0.66	0.55	0.76	0.81	0.91	0.90	0.90	0.53
NC(all)	0.38	0.43	0.31	0.46	0.46	0.63	0.60	0.88	1.62	1.44
CNC(all)	1.44	1.32	1.79	1.50	1.56	1.38	2.01	1.67	1.80	1.61
NCC(all)	1.26	1.29	1.40	1.11	1.71	1.12	1.60	1.26	2.37	1.42
$\phi = 0.85$										
C(1-1)	1.63	1.02	0.96	0.68	0.64	0.29	0.40	0.25	0.40	0.19
NC(1-1)	2.39	2.15	1.48	1.02	0.87	0.83	0.46	0.54	0.67	0.74
CNC(1-1)	4.90	3.68	2.93	2.18	1.38	1.11	0.97	0.76	1.27	1.22
NCC(1-1)	4.91	3.06	2.74	1.87	1.35	1.03	0.87	0.64	0.98	0.81
C(all)	0.57	0.54	0.66	0.51	0.66	0.48	0.61	0.47	0.72	0.44
NC(all)	0.46	0.56	0.43	0.78	0.59	0.77	1.09	1.23	4.65	4.47
CNC(all)	1.49	1.25	1.33	1.10	1.36	0.97	1.71	1.42	5.29	2.70
NCC(all)	1.39	1.26	1.24	1.10	1.18	1.08	1.59	1.25	3.66	2.59
$\phi = 0.75$										
C(1-1)	1.53	0.88	0.94	0.50	0.49	0.33	0.41	0.14	0.35	0.14
NC(1-1)	3.00	2.29	1.66	1.34	1.04	0.82	0.94	0.87	3.59	1.92
CNC(1-1)	4.89	3.35	2.55	1.73	1.38	1.12	1.53	1.32	4.27	2.29
NCC(1-1)	4.46	2.85	2.46	1.75	1.30	0.98	1.48	1.02	3.28	2.11
C(all)	0.51	0.43	0.43	0.48	0.53	0.34	0.57	0.42	0.64	0.40
NC(all)	0.54	0.68	0.61	0.75	0.97	0.99	2.01	1.88	11.70	5.47
CNC(all)	1.33	1.07	1.29	0.90	1.37	1.14	2.52	1.61	11.06	4.57
NCC(all)	1.28	1.15	1.16	0.90	1.26	1.08	2.23	1.54	11.60	5.75
$\phi = 0.65$										
C(1-1)	1.39	0.72	0.85	0.36	0.51	0.24	0.44	0.19	0.40	0.15
NC(1-1)	3.32	2.07	1.91	1.26	1.85	1.31	3.61	2.12	7.27	3.04
CNC(1-1)	4.19	2.80	2.17	1.83	2.00	1.09	3.22	2.14	8.29	3.32
NCC(1-1)	4.06	2.82	2.08	1.84	1.74	1.31	2.81	1.47	8.26	2.63
C(all)	0.35	0.37	0.43	0.38	0.48	0.35	0.54	0.37	0.69	0.38
NC(all)	0.69	0.76	0.58	0.83	1.70	1.49	4.03	2.66	23.32	5.98
CNC(all)	1.16	1.02	1.23	0.93	2.03	1.54	4.62	2.46	23.90	6.62
NCC(all)	1.14	1.08	1.10	0.95	1.82	1.44	4.02	2.46	19.60	5.44
$\phi = 0.55$										
C(1-1)	1.27	0.70	0.70	0.39	0.55	0.19	0.44	0.17	0.39	0.10
NC(1-1)	3.47	2.44	2.25	1.62	3.03	1.64	5.94	2.41	14.81	4.37
CNC(1-1)	4.23	2.59	2.82	1.82	3.54	1.85	6.34	3.05	14.21	3.77
NCC(1-1)	4.44	2.99	2.32	1.74	3.24	1.84	6.22	2.11	14.47	2.29
C(all)	0.39	0.34	0.43	0.31	0.49	0.33	0.53	0.35	0.70	0.36
NC(all)	0.86	0.87	0.85	0.94	2.31	1.92	8.78	3.84	34.87	6.61
CNC(all)	1.38	1.16	1.36	1.00	3.55	1.77	8.77	3.48	30.41	6.01
NCC(all)	1.31	1.18	1.31	0.92	2.68	2.00	7.65	2.97	27.50	4.98
$\phi = 0.0$										
C(1-1)	1.46	0.92	0.70	0.28	0.47	0.19	0.48	0.11	0.45	0.13
NC(1-1)	8.72	4.68	8.11	3.70	9.83	2.87	15.08	3.51	32.40	3.32
CNC(1-1)	9.54	4.23	8.13	2.80	10.08	2.83	15.39	3.34	31.41	3.69
NCC(1-1)	9.71	4.00	7.67	2.95	9.74	1.68	14.81	1.98	31.40	1.79
C(all)	0.34	0.30	0.29	0.28	0.44	0.30	0.53	0.31	0.47	0.31
NC(all)	4.18	3.01	4.47	2.58	8.91	3.28	16.91	4.27	46.28	5.83
CNC(all)	3.23	2.77	3.74	2.37	7.65	3.14	14.70	3.33	39.07	5.19
NCC(all)	3.05	2.69	3.52	2.23	6.49	2.98	14.14	4.12	41.33	3.84

Table 5.3: Performance of MCMC samplers as measured by ESS/sec for $\pi(\beta_0|\boldsymbol{h}, \boldsymbol{\alpha}, \phi, \tau)$ for the simulated datasets where $N = 2$. Higher is better. The quantities are averaged over twenty runs under different seeds initializations. All algorithms run 130,000 iterations after a burn-in of $30,000$ for each data set. The latent state is updated using '1-1' and 'all' method. The parameter vector is updated by applying the '2-bl' and '3-bl' sampler.

Alg.	τ									
	1		2		5		10		30	
	3-bl	2-bl	3-bl	2-bl	3-bl	2-bl	3-bl	2-bl	3-bl	2-bl
$\phi = 0.95$										
C(1-1)	30.25	30.59	17.46	24.18	9.27	14.21	4.69	6.17	1.02	1.41
NC(1-1)	0.12	0.11	0.18	0.12	0.18	0.16	0.19	0.17	0.21	0.25
CNC(1-1)	127.54	122.17	133.15	132.83	135.00	140.48	143.16	137.89	120.22	87.81
NCC(1-1)	112.05	96.82	123.56	122.82	133.53	138.98	138.35	139.38	116.85	90.13
C(all)	11.27	16.34	10.32	10.38	10.87	11.26	11.26	12.58	9.82	10.00
NC(all)	0.05	0.05	0.06	0.07	0.16	0.18	0.37	0.44	2.18	1.91
CNC(all)	107.61	72.12	142.96	92.93	171.80	139.25	210.31	184.75	184.16	156.24
NCC(all)	107.61	58.83	135.12	85.05	162.57	131.93	203.78	173.71	177.33	150.25
$\phi = 0.85$										
C(1-1)	9.65	9.97	4.53	4.89	1.76	1.98	1.05	1.28	0.54	0.58
NC(1-1)	1.27	1.12	1.63	1.47	1.89	2.18	2.75	2.54	4.39	4.06
CNC(1-1)	56.51	52.95	43.87	38.05	34.72	28.37	33.20	25.63	23.01	13.61
NCC(1-1)	57.59	52.95	40.47	32.59	34.29	28.03	31.68	24.15	18.82	17.16
C(all)	3.44	2.57	2.98	2.66	2.87	2.52	3.07	2.59	3.51	3.22
NC(all)	0.35	0.30	0.73	0.79	2.32	2.16	6.02	6.13	22.69	23.37
CNC(all)	31.30	28.26	28.05	23.68	34.91	31.55	51.81	38.54	64.15	43.36
NCC(all)	28.52	22.97	27.04	22.45	31.51	29.87	53.52	41.46	47.56	41.80
$\phi = 0.75$										
C(1-1)	4.70	4.00	2.64	2.11	1.24	1.14	0.84	0.64	0.44	0.53
NC(1-1)	4.25	3.45	4.52	4.70	7.22	5.65	10.38	8.00	15.25	13.32
CNC(1-1)	31.09	22.41	22.83	15.16	21.63	15.09	23.05	16.11	38.20	20.53
NCC(1-1)	29.13	21.31	19.88	14.88	20.52	11.57	22.28	14.28	33.75	18.02
C(all)	1.48	1.38	1.36	1.34	1.65	1.11	1.94	1.48	2.72	2.30
NC(all)	1.13	1.17	2.46	2.25	7.82	6.69	16.37	14.02	47.80	37.27
CNC(all)	11.71	11.33	12.05	8.48	20.05	16.49	38.14	22.74	62.86	44.80
NCC(all)	12.39	10.50	13.00	11.87	19.76	15.38	32.74	23.88	50.91	43.52
$\phi = 0.65$										
C(1-1)	3.18	2.30	2.04	1.21	1.10	0.93	0.91	0.71	0.48	0.46
NC(1-1)	6.68	5.72	7.16	6.36	13.81	10.38	20.54	13.82	31.80	27.21
CNC(1-1)	17.07	11.55	16.42	10.37	20.27	11.38	27.02	14.51	44.92	31.16
NCC(1-1)	17.20	12.54	16.45	9.93	19.84	9.80	26.48	12.52	45.72	28.17
C(all)	0.71	0.78	0.99	0.83	1.46	0.84	1.80	1.13	2.32	1.92
NC(all)	2.23	2.09	4.02	3.77	14.29	10.04	30.03	19.17	70.31	45.59
CNC(all)	8.19	5.58	9.63	6.60	19.24	13.88	35.71	20.84	66.32	48.81
NCC(all)	7.71	6.11	8.79	6.96	16.13	13.10	31.56	18.84	74.53	46.03
$\phi = 0.55$										
C(1-1)	2.32	1.91	1.45	1.11	1.16	0.69	0.88	0.82	0.61	0.36
NC(1-1)	9.57	7.15	11.56	6.89	16.97	9.98	24.55	17.10	54.25	40.51
CNC(1-1)	15.85	8.60	13.78	7.81	20.04	9.58	29.46	13.94	64.56	38.11
NCC(1-1)	15.13	9.58	12.84	7.60	19.74	8.80	28.84	12.93	62.40	40.47
C(all)	0.70	0.54	0.74	0.60	1.18	0.79	1.48	1.34	2.04	1.88
NC(all)	3.32	2.89	6.22	4.93	15.37	11.54	36.06	19.34	89.53	64.62
CNC(all)	7.15	5.00	7.13	6.30	16.30	11.06	35.04	21.82	76.74	64.64
NCC(all)	5.71	4.75	7.20	5.59	15.79	10.35	33.05	18.64	92.24	58.53
$\phi = 0.0$										
C(1-1)	2.08	1.55	1.01	0.55	0.87	0.54	0.99	0.45	0.57	0.53
NC(1-1)	11.76	7.01	12.06	6.42	20.02	7.87	44.86	17.00	94.06	71.68
CNC(1-1)	13.48	6.94	12.48	4.83	19.87	7.47	42.42	17.46	82.15	79.38
NCC(1-1)	13.83	6.37	12.10	5.01	19.37	5.27	44.37	11.73	84.54	67.48
C(all)	0.45	0.38	0.44	0.35	0.91	0.60	1.28	0.98	0.78	0.91
NC(all)	5.54	4.64	7.04	4.40	18.15	8.97	50.26	22.65	132.09	107.50
CNC(all)	6.75	4.52	6.06	4.39	15.91	8.44	42.62	20.95	102.95	90.71
NCC(all)	4.89	4.47	5.53	4.27	13.00	7.83	43.66	21.71	116.83	97.86

Table 5.4: Performance of MCMC samplers as measured by ESS/sec for $\pi(\phi|\boldsymbol{h}, \boldsymbol{\alpha}, \tau, \beta_0)$ for the simulated datasets where $N = 5$. Higher is better. The quantities are averaged over twenty runs under different seeds initializations. All algorithms run 130,000 iterations after a burn-in of $30,000$ for each data set. The latent state is updated using '1-1' and 'all' method. The parameter vector is updated by applying the '2-bl' and '3-bl' sampler.

Alg.	τ									
	1		2		5		10		30	
	3-bl	2-bl	3-bl	2-bl	3-bl	2-bl	3-bl	2-bl	3-bl	2-bl
$\phi = 0.95$										
C(1-1)	8.94	4.25	9.76	4.63	4.35	2.96	2.57	1.80	0.96	0.49
NC(1-1)	2.35	2.20	2.53	2.39	0.87	1.31	0.78	0.57	0.33	0.37
CNC(1-1)	8.40	5.78	8.77	5.54	5.33	4.01	2.61	2.01	0.77	0.82
NCC(1-1)	7.48	4.26	8.47	4.75	4.81	3.24	2.35	1.92	0.72	0.70
C(all)	1.95	1.79	2.02	2.22	2.52	2.20	2.47	1.56	1.50	0.81
NC(all)	0.38	0.53	0.33	0.57	0.91	0.77	0.81	0.72	1.12	0.76
CNC(all)	1.49	1.03	2.62	1.89	2.98	2.37	2.41	1.85	1.90	1.33
NCC(all)	0.85	1.24	2.00	1.87	2.28	2.03	2.29	1.80	1.67	1.09
$\phi = 0.85$										
C(1-1)	13.29	6.96	7.10	3.77	2.16	1.60	1.15	0.71	0.64	0.31
NC(1-1)	6.01	4.56	3.17	2.86	1.48	1.43	0.76	0.68	0.37	0.32
CNC(1-1)	12.96	8.37	7.00	4.48	2.77	2.33	1.35	1.15	0.47	0.42
NCC(1-1)	12.68	7.55	7.15	4.81	2.61	2.01	1.17	0.88	0.38	0.36
C(all)	2.19	2.04	1.94	1.56	1.28	1.00	0.90	0.61	0.75	0.43
NC(all)	0.82	0.95	0.96	0.80	0.72	0.57	0.66	0.56	0.82	0.85
CNC(all)	1.76	1.64	1.83	1.43	1.65	1.08	1.13	0.91	0.86	0.73
NCC(all)	1.93	1.45	1.96	1.71	1.38	1.00	1.13	0.93	0.85	0.68
$\phi = 0.75$										
C(1-1)	11.19	5.73	5.18	2.78	1.69	1.27	1.00	0.54	0.59	0.34
NC(1-1)	7.36	4.60	3.48	2.90	1.43	0.90	0.62	0.41	0.39	0.34
CNC(1-1)	11.21	7.54	5.30	4.23	2.12	1.61	0.77	0.83	0.48	0.37
NCC(1-1)	11.16	7.35	5.80	4.37	2.12	1.43	0.85	0.80	0.38	0.39
C(all)	1.60	1.73	1.45	1.39	0.78	0.83	0.65	0.48	0.75	0.53
NC(all)	1.27	0.94	1.12	0.75	0.75	0.57	0.60	0.47	0.87	0.74
CNC(all)	1.79	1.39	1.42	1.35	1.05	0.80	0.89	0.67	0.82	0.81
NCC(all)	1.43	1.40	1.21	1.20	1.03	0.73	0.87	0.82	0.83	0.67
$\phi = 0.65$										
C(1-1)	9.66	5.93	4.26	3.10	1.42	1.21	1.07	0.70	0.54	0.41
NC(1-1)	7.58	4.30	3.71	2.35	1.44	0.66	0.87	0.41	0.44	0.36
CNC(1-1)	9.97	7.91	4.80	3.65	1.63	1.24	0.74	0.70	0.37	0.33
NCC(1-1)	10.34	7.99	4.64	3.64	1.77	1.48	0.86	0.95	0.34	0.45
C(all)	1.39	1.69	1.06	1.28	0.64	0.63	0.62	0.55	0.76	0.52
NC(all)	1.26	1.08	1.01	0.82	0.67	0.53	0.66	0.52	0.80	0.69
CNC(all)	1.43	1.38	1.21	1.04	0.87	0.69	0.82	0.66	0.79	0.74
NCC(all)	1.11	1.22	1.03	1.10	0.85	0.67	0.78	0.76	0.69	0.65
$\phi = 0.55$										
C(1-1)	9.22	4.77	3.99	1.81	1.60	1.12	0.90	0.71	0.69	0.47
NC(1-1)	7.30	5.90	3.69	3.02	1.54	0.72	0.79	0.49	0.49	0.38
CNC(1-1)	9.29	7.44	4.19	3.81	1.27	1.36	0.74	0.78	0.46	0.57
NCC(1-1)	9.29	8.08	3.83	3.55	1.25	1.33	0.77	0.76	0.49	0.39
C(all)	1.33	1.47	0.95	1.08	0.60	0.67	0.69	0.74	0.82	0.54
NC(all)	1.32	1.02	1.02	0.75	0.58	0.58	0.64	0.57	0.84	0.77
CNC(all)	1.41	1.30	1.12	0.85	0.83	0.72	0.74	0.62	0.71	0.80
NCC(all)	1.06	1.15	1.03	1.00	0.79	0.57	0.75	0.62	0.69	0.64
$\phi = 0.0$										
C(1-1)	8.61	6.45	4.07	3.30	1.81	1.60	1.40	1.02	1.18	0.90
NC(1-1)	8.59	4.67	3.77	1.60	1.43	0.99	1.25	0.88	1.04	0.64
CNC(1-1)	7.38	7.50	3.54	3.74	1.39	1.73	0.95	1.34	0.85	1.11
NCC(1-1)	8.35	8.17	3.67	3.81	1.58	1.81	1.23	1.40	1.07	1.10
C(all)	1.26	1.47	1.04	0.98	0.77	0.71	0.72	0.70	1.00	0.81
NC(all)	1.44	1.29	1.07	0.73	0.92	0.90	1.04	1.10	1.08	1.34
CNC(all)	1.04	1.04	0.85	1.09	0.71	0.67	0.91	0.84	0.98	1.05
NCC(all)	1.18	1.17	0.92	0.93	0.65	0.85	0.77	0.84	1.17	1.18

Table 5.5: Performance of MCMC samplers as measured by ESS/sec for $\pi(\tau|\boldsymbol{h}, \boldsymbol{\alpha}, \phi, \beta_0)$ for the simulated datasets where $N = 5$. Higher is better. The quantities are averaged over twenty runs under different seeds initializations. All algorithms run 130,000 iterations after a burn-in of $30,000$ for each data set. The latent state is updated using '1-1' and 'all' method. The parameter vector is updated by applying the '2-bl' and '3-bl' sampler.

Alg.	τ									
	1		2		5		10		30	
	3-bl	2-bl	3-bl	2-bl	3-bl	2-bl	3-bl	2-bl	3-bl	2-bl
$\phi = 0.95$										
C(1-1)	3.38	2.35	3.08	2.00	1.66	1.20	1.16	0.78	0.53	0.28
NC(1-1)	1.58	1.68	1.16	1.24	0.72	0.75	0.57	0.39	0.29	0.35
CNC(1-1)	9.14	5.78	6.45	4.19	3.27	2.44	1.71	1.34	0.71	0.71
NCC(1-1)	8.59	4.99	6.19	4.65	3.67	2.56	1.91	1.47	0.68	0.59
C(all)	0.76	0.55	0.80	0.77	0.96	1.04	1.22	0.79	0.96	0.54
NC(all)	0.33	0.45	0.30	0.35	0.50	0.46	0.63	0.46	1.07	0.72
CNC(all)	1.59	1.80	1.72	1.41	1.97	1.56	1.86	1.33	1.59	1.16
NCC(all)	1.58	1.08	1.70	1.41	1.62	1.52	1.57	1.28	1.83	1.27
$\phi = 0.85$										
C(1-1)	5.44	3.07	3.45	2.01	1.20	0.94	0.76	0.49	0.39	0.20
NC(1-1)	3.61	3.13	2.47	2.27	1.46	1.38	0.86	0.76	0.63	0.51
CNC(1-1)	11.99	6.91	7.27	4.46	3.06	2.36	1.63	1.05	0.93	0.60
NCC(1-1)	11.27	6.84	6.82	3.73	2.84	2.04	1.40	1.25	0.79	0.62
C(all)	0.81	0.73	0.92	0.60	0.80	0.59	0.62	0.36	0.60	0.27
NC(all)	0.66	0.72	0.79	0.67	0.70	0.58	0.81	0.65	1.41	1.41
CNC(all)	2.07	1.85	1.95	1.42	1.51	1.16	1.34	1.11	1.60	1.32
NCC(all)	1.88	1.51	1.72	1.61	1.55	1.01	1.28	1.12	1.47	1.15
$\phi = 0.75$										
C(1-1)	5.88	2.69	2.64	1.58	1.01	0.50	0.70	0.26	0.41	0.20
NC(1-1)	5.59	3.62	3.42	2.77	1.70	1.39	0.90	0.68	1.56	1.20
CNC(1-1)	12.07	7.14	6.47	4.48	2.94	1.92	1.36	1.23	2.11	1.27
NCC(1-1)	11.52	6.65	6.30	4.39	2.66	1.72	1.30	1.04	1.62	1.55
C(all)	0.91	0.67	0.86	0.71	0.60	0.56	0.53	0.31	0.57	0.31
NC(all)	1.15	0.86	1.17	0.79	1.02	0.67	0.97	0.84	3.48	2.28
CNC(all)	2.21	1.72	1.60	1.66	1.34	1.02	1.32	1.18	3.90	2.55
NCC(all)	1.93	1.61	1.58	1.42	1.29	1.05	1.32	1.03	3.05	2.22
$\phi = 0.65$										
C(1-1)	5.29	2.60	2.75	1.47	0.95	0.49	0.56	0.29	0.41	0.13
NC(1-1)	6.79	5.33	4.19	3.31	2.04	1.47	1.78	1.24	3.30	2.29
CNC(1-1)	11.91	7.69	6.34	4.04	2.61	1.92	1.86	1.48	4.50	2.19
NCC(1-1)	12.12	8.10	5.83	4.03	2.64	1.58	1.79	1.40	4.09	1.63
C(all)	0.71	0.79	0.68	0.67	0.54	0.44	0.53	0.35	0.51	0.28
NC(all)	1.59	1.16	1.25	1.00	0.96	0.74	1.39	1.33	5.83	3.64
CNC(all)	1.97	1.85	1.56	1.45	1.34	1.03	1.65	1.31	6.22	3.34
NCC(all)	1.51	1.63	1.39	1.46	1.12	1.07	1.97	1.55	6.67	3.51
$\phi = 0.55$										
C(1-1)	6.05	2.88	2.85	1.29	0.97	0.48	0.49	0.28	0.43	0.18
NC(1-1)	7.55	6.01	4.91	3.47	2.91	1.58	2.48	1.72	6.20	3.14
CNC(1-1)	12.95	8.96	6.19	4.26	2.87	2.10	3.30	2.07	7.94	2.82
NCC(1-1)	12.49	8.01	5.97	4.45	2.56	1.85	2.91	2.08	7.10	2.00
C(all)	0.77	0.66	0.65	0.69	0.48	0.43	0.53	0.39	0.48	0.27
NC(all)	1.84	1.42	1.53	1.16	1.16	1.14	2.41	2.09	11.63	5.03
CNC(all)	2.06	1.94	1.70	1.49	1.68	1.21	3.29	2.07	10.44	3.25
NCC(all)	1.74	1.85	1.65	1.49	1.59	1.22	2.56	1.74	11.28	4.27
$\phi = 0.0$										
C(1-1)	7.46	3.67	3.17	1.61	0.91	0.46	0.60	0.25	0.42	0.17
NC(1-1)	17.98	12.26	15.54	8.33	11.04	4.16	11.20	3.71	17.07	3.03
CNC(1-1)	25.21	14.19	17.37	8.85	11.31	4.44	11.82	3.39	19.71	3.30
NCC(1-1)	24.65	13.14	16.93	8.67	10.46	2.99	9.79	2.54	16.69	2.08
C(all)	0.74	0.69	0.73	0.56	0.46	0.38	0.41	0.33	0.46	0.35
NC(all)	6.80	6.02	5.94	4.91	6.39	4.28	7.44	3.12	17.97	3.49
CNC(all)	6.04	5.81	5.44	4.67	5.30	3.22	8.41	4.24	20.44	4.91
NCC(all)	6.41	4.95	5.26	4.30	5.39	3.60	7.64	3.73	19.68	4.35

Table 5.6: Performance of MCMC samplers as measured by ESS/sec for $\pi(\beta_0|\boldsymbol{h}, \boldsymbol{\alpha}, \phi, \tau)$ for the simulated datasets where $N = 5$. Higher is better. The quantities are averaged over twenty runs under different seeds initializations. All algorithms run 130,000 iterations after a burn-in of $30,000$ for each data set. The latent state is updated using '1-1' and 'all' method. The parameter vector is updated by applying the '2-bl' and '3-bl' sampler.

Alg.	τ									
	1		2		5		10		30	
	3-bl	2-bl	3-bl	2-bl	3-bl	2-bl	3-bl	2-bl	3-bl	2-bl
$\phi = 0.95$										
C(1-1)	144.84	116.90	136.26	148.98	70.13	94.61	30.45	36.78	5.08	5.83
NC(1-1)	0.09	0.14	0.13	0.12	0.12	0.12	0.11	0.10	0.15	0.15
CNC(1-1)	204.73	171.56	279.15	240.69	288.09	268.69	280.71	249.48	210.73	192.20
NCC(1-1)	184.23	184.73	249.93	258.79	306.95	282.16	280.48	261.87	217.08	197.48
C(all)	161.59	162.20	125.36	125.28	52.94	49.47	27.58	31.63	16.06	19.23
NC(all)	0.04	0.04	0.05	0.04	0.07	0.09	0.11	0.17	0.59	0.80
CNC(all)	230.95	153.68	336.16	294.35	336.11	339.02	349.89	331.12	269.74	251.73
NCC(all)	248.58	231.14	324.45	303.67	375.44	338.95	360.21	332.49	299.81	276.36
$\phi = 0.85$										
C(1-1)	60.48	56.98	32.79	27.78	10.21	9.54	3.95	5.38	1.18	1.51
NC(1-1)	0.93	0.90	1.09	1.02	1.61	1.50	1.71	1.93	3.55	3.77
CNC(1-1)	157.38	146.67	143.84	123.47	108.04	92.54	80.31	62.45	52.79	39.89
NCC(1-1)	155.23	145.05	141.97	129.38	106.37	102.66	79.44	71.84	53.56	45.05
C(all)	9.49	8.42	8.31	8.29	5.62	6.08	4.81	5.68	3.80	3.58
NC(all)	0.11	0.17	0.22	0.22	0.79	0.87	2.17	2.05	10.65	10.88
CNC(all)	88.75	64.25	108.26	100.21	87.20	84.47	78.26	77.02	68.43	64.36
NCC(all)	97.60	109.18	113.67	116.91	105.50	85.55	92.09	75.31	81.24	62.03
$\phi = 0.75$										
C(1-1)	37.04	29.98	17.79	13.56	5.59	4.11	2.34	2.39	1.18	1.06
NC(1-1)	2.78	2.40	3.69	3.90	5.13	5.30	7.55	7.20	15.14	12.72
CNC(1-1)	106.93	86.94	87.27	67.87	57.73	35.37	44.40	31.99	42.53	35.49
NCC(1-1)	105.35	93.30	83.83	75.59	46.54	36.69	43.31	28.54	43.64	37.11
C(all)	4.78	3.90	3.99	3.27	2.83	2.74	2.62	2.34	2.61	2.35
NC(all)	0.37	0.35	0.94	1.01	3.16	2.86	7.58	7.40	32.15	27.83
CNC(all)	54.26	40.63	49.91	44.99	38.96	31.49	41.16	33.35	63.11	50.44
NCC(all)	62.27	50.35	49.30	45.63	37.72	31.75	40.79	37.24	68.14	54.60
$\phi = 0.65$										
C(1-1)	24.01	20.24	11.76	7.69	3.33	3.64	2.18	1.96	1.13	1.00
NC(1-1)	5.75	6.28	9.04	7.92	13.78	10.10	18.17	15.43	29.51	20.35
CNC(1-1)	77.57	61.61	56.36	40.92	36.19	22.55	36.43	23.24	50.42	30.80
NCC(1-1)	77.47	71.19	55.31	42.17	31.82	22.37	34.30	22.83	48.05	29.09
C(all)	3.37	2.41	2.74	2.16	2.04	1.72	2.08	1.56	2.28	2.15
NC(all)	0.94	1.02	2.35	2.23	6.19	5.85	16.37	13.68	50.34	36.63
CNC(all)	34.56	27.29	28.13	22.91	22.65	19.23	31.14	24.31	64.09	46.28
NCC(all)	36.71	32.25	29.21	24.13	22.71	17.63	33.08	27.66	66.94	47.11
$\phi = 0.55$										
C(1-1)	21.05	15.24	9.25	5.95	3.09	2.51	1.46	1.95	1.01	1.00
NC(1-1)	9.67	8.66	14.73	12.32	18.99	12.11	25.67	17.26	37.93	28.21
CNC(1-1)	63.10	49.10	45.05	28.02	32.83	17.82	34.59	21.61	53.52	37.31
NCC(1-1)	61.85	55.35	43.35	30.45	30.30	17.71	32.82	19.38	53.87	31.60
C(all)	2.52	1.66	2.09	1.79	1.57	1.26	1.75	1.58	2.07	2.01
NC(all)	1.75	1.83	4.21	3.63	9.98	9.05	24.73	17.26	61.53	44.83
CNC(all)	26.89	20.96	20.07	18.06	19.88	14.59	31.87	25.02	65.38	45.57
NCC(all)	29.14	24.80	22.50	16.17	20.91	15.36	32.59	23.12	70.95	50.46
$\phi = 0.0$										
C(1-1)	12.23	7.76	5.59	3.92	2.22	1.70	1.64	2.01	1.07	1.45
NC(1-1)	23.10	19.95	26.25	17.13	25.34	13.05	33.85	15.52	74.59	38.15
CNC(1-1)	39.86	27.57	32.55	19.67	26.73	13.25	33.82	14.08	67.83	43.92
NCC(1-1)	39.15	29.28	31.10	19.82	26.81	10.85	33.19	12.44	70.42	41.15
C(all)	1.25	1.03	1.22	0.95	1.07	0.83	1.31	0.93	1.30	1.60
NC(all)	8.52	8.74	12.11	10.67	17.41	12.55	31.19	19.28	92.10	61.33
CNC(all)	14.28	13.87	13.05	12.38	15.10	11.22	26.36	16.24	77.30	58.33
NCC(all)	14.85	11.80	13.93	11.57	16.59	11.10	27.90	17.48	88.33	58.06

Table 5.7: Performance of MCMC samplers as measured by ESS/sec for $\pi(\phi|\boldsymbol{h}, \boldsymbol{\alpha}, \tau, \beta_0)$ for the simulated datasets where $N = 10$. Higher is better. The quantities are averaged over twenty runs under different seeds initializations. All algorithms run 130,000 iterations after a burn-in of $30,000$ for each data set. The latent state is updated using '1-1' and 'all' method. The parameter vector is updated by applying the '2-bl' and '3-bl' sampler.

Alg.	τ									
	1		2		5		10		30	
	3-bl	2-bl	3-bl	2-bl	3-bl	2-bl	3-bl	2-bl	3-bl	2-bl
$\phi = 0.95$										
C(1-1)	16.43	11.95	18.05	8.23	11.92	3.39	5.92	1.29	1.78	0.35
NC(1-1)	4.17	7.67	4.25	5.09	2.73	2.64	1.81	1.37	0.57	0.49
CNC(1-1)	16.41	15.30	18.38	11.12	12.73	4.81	6.08	2.25	1.58	0.78
NCC(1-1)	16.20	14.01	16.22	9.95	10.31	4.77	5.78	2.15	1.38	0.62
C(all)	3.25	3.43	3.52	2.99	4.09	1.92	3.32	1.32	2.44	0.68
NC(all)	0.88	1.30	0.92	1.01	0.67	0.78	0.91	0.71	0.67	0.52
CNC(all)	2.86	2.51	4.00	3.00	4.03	2.08	3.47	1.48	2.63	0.93
NCC(all)	1.73	2.44	3.97	2.61	3.57	1.60	4.07	1.29	2.04	0.84
$\phi = 0.85$										
C(1-1)	27.20	11.95	16.33	8.23	5.92	3.39	2.41	1.29	0.76	0.35
NC(1-1)	11.88	7.67	7.51	5.09	3.53	2.64	1.41	1.37	0.50	0.49
CNC(1-1)	24.86	15.30	17.02	11.12	6.37	4.77	3.10	2.25	0.86	0.62
NCC(1-1)	24.50	14.01	15.73	9.95	6.36	4.81	3.15	2.15	0.78	0.78
C(all)	2.56	3.43	2.71	2.99	1.97	1.92	1.42	1.32	0.83	0.68
NC(all)	1.14	1.30	1.09	1.01	0.72	0.78	0.70	0.71	0.63	0.52
CNC(all)	2.50	2.51	2.29	3.00	1.84	2.08	1.56	1.48	1.12	0.84
NCC(all)	1.79	2.44	2.83	2.61	2.14	1.60	1.66	1.29	1.17	0.93
$\phi = 0.75$										
C(1-1)	26.04	11.43	13.65	6.84	4.04	2.75	1.79	1.15	0.71	0.34
NC(1-1)	13.64	9.12	8.49	6.25	3.44	2.83	1.55	1.41	0.62	0.54
CNC(1-1)	23.82	16.51	13.38	9.38	5.24	4.15	2.55	1.82	0.71	0.73
NCC(1-1)	23.15	15.43	13.23	9.22	5.26	3.64	2.40	1.71	0.63	0.57
C(all)	2.54	2.98	2.27	2.14	1.53	1.35	0.89	0.87	0.68	0.55
NC(all)	1.26	1.34	1.06	0.98	0.82	0.80	0.64	0.59	0.74	0.61
CNC(all)	2.02	2.55	2.10	2.28	1.45	1.40	1.13	1.04	0.89	0.77
NCC(all)	1.96	2.44	1.75	1.99	1.36	1.22	1.08	0.95	0.76	0.69
$\phi = 0.65$										
C(1-1)	22.93	10.49	10.98	5.23	3.71	1.77	1.65	0.62	0.82	0.38
NC(1-1)	15.72	9.67	8.52	5.95	3.41	2.84	1.58	1.39	0.69	0.49
CNC(1-1)	20.93	15.24	11.84	8.69	4.16	3.56	1.86	1.52	0.67	0.63
NCC(1-1)	20.82	15.36	10.87	8.56	4.06	3.33	1.73	1.71	0.67	0.75
C(all)	2.55	2.44	2.02	1.94	1.07	1.19	0.70	0.78	0.67	0.55
NC(all)	1.36	1.23	1.13	0.94	0.84	0.76	0.80	0.55	0.73	0.56
CNC(all)	1.55	2.14	1.59	1.87	1.20	1.18	0.93	0.91	0.82	0.75
NCC(all)	1.52	2.26	1.45	1.64	1.16	1.10	0.89	0.84	0.79	0.79
$\phi = 0.55$										
C(1-1)	19.87	9.76	9.33	4.74	3.10	1.29	1.44	0.60	0.95	0.47
NC(1-1)	15.28	10.36	8.67	5.92	3.14	2.65	1.63	1.40	0.69	0.74
CNC(1-1)	19.72	14.73	10.75	8.25	3.68	2.97	1.67	1.47	0.58	0.59
NCC(1-1)	18.78	14.60	9.80	7.86	3.72	3.12	1.50	1.38	0.67	0.73
C(all)	2.03	1.92	1.29	1.64	1.01	0.95	0.74	0.75	0.69	0.58
NC(all)	1.45	1.40	1.21	1.04	0.86	0.69	0.82	0.71	0.85	0.64
CNC(all)	1.90	2.07	1.45	1.62	1.11	1.03	0.83	0.82	0.79	0.83
NCC(all)	1.81	1.90	1.07	1.42	1.03	1.02	0.79	0.75	0.59	0.69
$\phi = 0.0$										
C(1-1)	17.60	9.99	8.16	4.42	3.00	1.51	1.97	0.76	1.26	0.68
NC(1-1)	17.10	11.11	8.35	6.61	2.60	1.99	1.48	1.26	1.04	1.06
CNC(1-1)	15.10	14.25	6.98	7.04	2.60	2.83	1.61	1.65	1.23	1.27
NCC(1-1)	14.95	14.21	7.49	7.61	2.85	2.67	1.42	1.50	1.05	1.05
C(all)	1.80	1.59	1.41	1.16	0.86	0.90	0.82	0.79	1.19	0.70
NC(all)	1.82	1.74	1.46	1.29	1.06	0.90	0.90	1.00	1.14	0.97
CNC(all)	1.37	1.13	0.91	0.95	0.70	0.83	0.92	0.83	1.09	1.02
NCC(all)	1.52	1.47	1.07	1.06	0.75	0.77	0.53	0.77	1.01	0.90

Table 5.8: Performance of MCMC samplers as measured by ESS/sec for $\pi(\tau|\boldsymbol{h}, \boldsymbol{\alpha}, \phi, \beta_0)$ for the simulated datasets where $N = 10$. Higher is better. The quantities are averaged over twenty runs under different seeds initializations. All algorithms run 130,000 iterations after a burn-in of $30,000$ for each data set. The latent state is updated using '1-1' and 'all' method. The parameter vector is updated by applying the '2-bl' and '3-bl' sampler.

Alg.	τ									
	1		2		5		10		30	
	3-bl	2-bl	3-bl	2-bl	3-bl	2-bl	3-bl	2-bl	3-bl	2-bl
$\phi = 0.95$										
C(1-1)	7.11	4.79	6.90	4.25	4.21	2.44	2.45	1.77	1.02	0.61
NC(1-1)	1.94	1.86	1.63	1.45	1.22	1.12	0.97	0.72	0.41	0.30
CNC(1-1)	15.17	9.64	11.86	6.92	6.31	4.03	3.58	2.41	1.20	0.99
NCC(1-1)	14.72	8.42	11.66	6.85	6.92	4.11	3.50	2.35	1.13	1.00
C(all)	0.69	0.66	1.01	0.97	1.11	1.10	1.18	1.03	1.22	0.83
NC(all)	0.29	0.20	0.30	0.23	0.32	0.34	0.37	0.38	0.50	0.58
CNC(all)	1.40	1.29	1.70	1.62	1.85	1.41	1.84	1.42	1.65	1.38
NCC(all)	1.26	1.20	1.78	1.46	1.66	1.38	2.37	1.37	2.09	1.17
$\phi = 0.85$										
C(1-1)	11.87	7.08	7.45	4.68	3.54	1.76	1.77	0.81	0.64	0.30
NC(1-1)	4.72	4.17	3.75	3.45	2.67	2.18	1.35	1.35	1.03	0.59
CNC(1-1)	21.08	11.68	13.46	8.63	5.81	4.30	3.31	2.22	0.60	0.80
NCC(1-1)	20.28	12.56	13.12	8.17	5.84	4.11	3.20	2.28	1.00	0.98
C(all)	0.70	0.87	0.90	0.99	0.97	1.02	0.90	0.90	0.60	0.47
NC(all)	0.54	0.56	0.59	0.58	0.59	0.63	0.69	0.69	0.87	0.66
CNC(all)	2.03	2.20	2.17	2.08	1.82	1.68	1.77	1.35	1.53	1.14
NCC(all)	2.11	2.17	1.91	1.96	1.56	1.45	1.56	1.42	1.44	1.09
$\phi = 0.75$										
C(1-1)	13.09	6.80	7.63	3.94	2.72	1.94	1.33	0.56	0.59	0.20
NC(1-1)	6.63	5.07	6.12	4.83	3.33	2.64	1.91	1.60	1.16	0.77
CNC(1-1)	22.73	14.13	14.57	8.93	6.33	4.07	3.31	2.15	1.42	1.12
NCC(1-1)	22.49	13.34	13.24	8.13	6.04	4.01	3.09	2.04	1.17	1.21
C(all)	0.90	1.07	1.00	0.94	0.96	0.84	0.75	0.58	0.54	0.38
NC(all)	0.83	0.93	0.93	0.83	0.88	0.86	0.95	0.68	1.41	1.01
CNC(all)	2.18	2.57	2.11	2.37	1.57	1.63	1.35	1.31	1.77	1.23
NCC(all)	1.94	2.26	1.82	2.02	1.61	1.40	1.35	1.15	1.65	1.28
$\phi = 0.65$										
C(1-1)	13.76	6.77	6.99	3.67	2.65	1.10	1.23	0.43	0.60	0.20
NC(1-1)	8.62	6.81	7.57	5.50	4.10	3.28	2.45	1.82	2.47	1.27
CNC(1-1)	24.22	14.47	15.11	9.05	5.84	3.93	3.05	2.46	2.51	1.91
NCC(1-1)	23.87	13.24	13.21	8.87	5.75	4.41	2.90	2.10	2.25	1.53
C(all)	0.93	1.07	0.97	0.81	0.86	0.79	0.62	0.51	0.54	0.40
NC(all)	1.14	1.12	1.25	1.03	1.13	0.94	1.30	0.85	3.35	1.84
CNC(all)	2.15	3.25	1.97	2.20	1.65	1.62	1.38	1.33	2.92	2.17
NCC(all)	2.17	2.67	1.80	2.22	1.59	1.37	1.32	1.32	2.53	1.81
$\phi = 0.55$										
C(1-1)	13.12	6.50	6.93	3.20	2.31	0.91	0.98	0.36	0.63	0.25
NC(1-1)	10.31	7.62	8.59	6.07	4.44	3.20	2.81	1.93	4.78	2.44
CNC(1-1)	25.84	15.36	15.93	9.09	6.06	4.53	3.46	2.47	5.56	2.72
NCC(1-1)	24.67	14.22	13.98	9.31	5.78	3.96	3.30	2.10	4.83	1.86
C(all)	0.95	0.96	0.82	0.78	0.79	0.67	0.61	0.54	0.54	0.36
NC(all)	1.55	1.46	1.51	1.32	1.38	1.03	1.83	1.24	5.64	3.55
CNC(all)	2.99	3.14	2.15	2.38	1.56	1.69	1.64	1.57	5.40	3.55
NCC(all)	2.93	2.66	2.02	2.04	1.60	1.59	1.63	1.40	4.65	3.01
$\phi = 0.0$										
C(1-1)	16.97	7.79	8.79	4.19	2.53	1.16	1.03	0.45	0.54	0.25
NC(1-1)	22.17	15.74	22.74	14.24	16.19	7.99	13.96	5.06	16.72	3.85
CNC(1-1)	43.67	24.16	32.24	16.64	17.65	7.57	13.48	4.48	16.05	3.87
NCC(1-1)	43.67	22.52	32.04	15.97	16.26	7.09	12.86	4.01	15.77	2.80
C(all)	0.99	0.80	1.02	0.89	0.69	0.61	0.61	0.38	0.60	0.32
NC(all)	6.62	6.22	8.32	7.20	6.86	5.16	7.81	4.65	16.44	4.85
CNC(all)	10.18	7.43	8.46	6.55	6.41	4.87	7.22	4.06	14.65	4.44
NCC(all)	9.18	7.39	8.00	5.96	6.59	4.35	7.48	4.13	13.48	4.54

Table 5.9: Performance of MCMC samplers as measured by ESS/sec for $\pi(\beta_0|\boldsymbol{h}, \boldsymbol{\alpha}, \phi, \tau)$ for the simulated datasets where $N = 10$. Higher is better. The quantities are averaged over twenty runs under different seeds initializations. All algorithms run 130,000 iterations after a burn-in of $30,000$ for each data set. The latent state is updated using '1-1' and 'all' method. The parameter vector is updated by applying the '2-bl' and '3-bl' sampler.

Alg.	τ									
	1		2		5		10		30	
	3-bl	2-bl	3-bl	2-bl	3-bl	2-bl	3-bl	2-bl	3-bl	2-bl
$\phi = 0.95$										
C(1-1)	30.09	24.37	16.77	13.49	6.51	4.58	3.71	2.82	2.13	2.19
NC(1-1)	23.67	22.27	33.16	27.61	38.33	27.27	45.58	26.70	76.77	51.65
CNC(1-1)	74.64	62.55	61.86	45.69	47.34	28.95	47.44	24.82	70.09	51.32
NCC(1-1)	75.81	62.92	62.48	49.18	45.11	29.47	47.80	23.08	70.87	51.00
C(all)	1.61	1.73	1.77	1.87	1.60	1.44	1.59	1.26	1.72	1.95
NC(all)	4.85	4.43	10.40	10.57	21.06	17.90	35.39	24.63	94.30	66.00
CNC(all)	26.47	26.05	28.04	25.61	25.30	21.17	36.67	24.69	77.77	55.10
NCC(all)	33.29	24.20	29.56	22.04	26.94	21.96	38.30	23.88	83.10	63.17
$\phi = 0.85$										
C(1-1)	151.25	169.61	104.61	120.74	39.34	45.56	18.94	14.94	4.06	3.99
NC(1-1)	0.62	0.75	0.83	0.88	1.39	1.48	1.81	2.08	3.19	2.87
CNC(1-1)	256.39	237.52	242.35	227.49	193.61	179.19	153.93	139.21	99.60	79.79
NCC(1-1)	267.97	236.94	248.82	226.70	193.28	192.32	148.89	145.13	107.61	95.28
C(all)	9.80	22.72	10.83	19.51	8.87	12.30	6.44	9.33	4.74	6.63
NC(all)	0.09	0.05	0.12	0.11	0.26	0.43	0.82	0.98	4.14	3.68
CNC(all)	174.26	164.86	230.35	192.27	206.01	186.11	162.16	141.35	113.82	101.66
NCC(all)	221.69	159.53	257.20	243.19	230.26	182.78	172.56	157.89	124.07	111.03
$\phi = 0.75$										
C(1-1)	111.72	112.49	60.20	61.02	21.02	21.67	8.33	6.49	2.30	1.96
NC(1-1)	2.00	1.86	2.79	2.79	4.51	4.78	6.79	6.32	11.84	10.95
CNC(1-1)	202.77	185.02	172.66	150.53	125.96	104.37	86.09	65.93	63.55	47.96
NCC(1-1)	206.38	189.99	167.85	162.65	113.77	107.85	84.72	72.27	68.99	48.06
C(all)	6.76	10.19	6.63	8.95	4.51	5.21	3.68	4.21	3.07	4.08
NC(all)	0.14	0.12	0.31	0.35	1.38	1.21	3.51	3.00	16.19	14.31
CNC(all)	103.45	90.73	113.05	120.16	86.30	74.98	64.87	62.44	70.62	60.79
NCC(all)	124.26	113.25	130.28	101.56	109.25	77.54	80.47	62.57	85.46	57.01
$\phi = 0.65$										
C(1-1)	79.28	78.36	42.62	40.00	14.76	11.12	5.59	4.97	1.90	1.89
NC(1-1)	3.35	3.12	5.71	5.63	10.17	9.41	13.65	13.05	25.41	21.23
CNC(1-1)	164.48	143.55	132.49	114.12	92.47	71.06	67.14	47.72	65.12	41.13
NCC(1-1)	164.23	151.34	132.61	119.38	84.38	75.25	64.06	52.35	61.98	40.28
C(all)	5.63	6.82	4.30	4.95	3.31	3.58	2.81	3.06	2.68	3.02
NC(all)	0.35	0.29	0.90	0.96	2.98	2.63	7.21	6.52	29.41	28.12
CNC(all)	80.34	75.65	73.28	74.41	51.63	48.30	47.61	43.92	66.43	49.83
NCC(all)	89.31	78.63	91.35	68.08	64.80	44.41	54.83	39.86	71.27	52.28
$\phi = 0.55$										
C(1-1)	64.20	57.17	32.64	28.23	10.54	7.45	4.27	3.80	2.02	1.95
NC(1-1)	5.67	6.02	9.59	9.69	16.42	14.57	23.02	18.34	38.89	33.73
CNC(1-1)	141.58	119.06	107.61	89.65	68.50	52.01	54.75	38.78	64.45	43.00
NCC(1-1)	139.49	126.35	109.01	92.16	69.88	59.56	57.99	41.14	68.44	41.41
C(all)	3.87	4.60	3.29	3.42	2.70	2.61	2.31	2.50	2.42	2.50
NC(all)	0.71	0.69	1.63	1.60	5.27	4.47	11.88	11.63	46.52	37.20
CNC(all)	62.11	60.80	54.70	50.30	43.71	35.76	41.93	34.20	68.25	55.80
NCC(all)	72.20	61.65	68.63	51.40	46.80	35.10	45.39	38.55	77.44	53.98
$\phi = 0.0$										
C(1-1)	30.09	24.37	16.77	13.49	6.51	4.58	3.71	2.82	2.13	2.19
NC(1-1)	23.67	22.27	33.16	27.61	38.33	27.27	45.58	26.70	76.77	51.65
CNC(1-1)	74.64	62.55	61.86	45.69	47.34	28.95	47.44	24.82	70.09	51.32
NCC(1-1)	75.81	62.92	62.48	49.18	45.11	29.47	47.80	23.08	70.87	51.00
C(all)	1.61	1.73	1.77	1.87	1.60	1.44	1.59	1.26	1.72	1.95
NC(all)	4.85	4.43	10.40	10.57	21.06	17.90	35.39	24.63	94.30	66.00
CNC(all)	26.47	26.05	28.04	25.61	25.30	21.17	36.67	24.69	77.77	55.10
NCC(all)	33.29	24.20	29.56	22.04	26.94	21.96	38.30	23.88	83.10	63.17

the blocking strategy the 3-block sampler shows higher ESS/sec most of the time. As the sample size increases, the sampling efficiency deteriorates, for example, when $n = 6,000$ the ESS/sec is on average two and half times larger than its corresponding value when the sample size is doubled.

Concerning the simulation efficiency of τ (Tables 5.2, 5.5, 5.8), it seems that for some values of τ, as ϕ decreases, the sampling efficiency may increase, decrease or witness a gradual decline, followed by a rise. Besides, as the number of N decreases, the sampling efficiency of τ decreases. Comparing the sampling schemes, the result show that the non-centered parameterization performs better compared to the centered (in most cases), and the ESS/sec is similar to the interweaving samplers which have the smallest autocorrelation for both parameterizations for most of the parameters values. Comparing the method for updating the latent state, the results report that for $N = 2$ ($N \in \{5, 10\}$) it is preferable to update the components of the latent state in one step for $\tau > 10$ ($\tau > 30$). Besides, by comparing the blocking strategy the 3-block sampler shows lower autocorrelation most of the time. The above results hold irrespective of the sample size or the choice of β_0. Finally, as the sample size increases, the sampling efficiency deteriorates, for example, when $n = 6,000$ the ESS/sec is on average two times larger than its corresponding value when the sample size is doubled.

Concerning the simulation efficiency of β_0 (Tables 5.3, 5.6, 5.9), the results show that in most cases, keeping ϕ fixed, as the value of τ increases, the sampling efficiency initially declines and then rises. On the other hand, for $\tau \geq 10$ ($\tau = 30$), as ϕ decreases, the ESS/sec declines (initially deteriorate and then increase). Besides, as the number of N increases, the sampling efficiency of β_0 increases. Comparing the latent path strategy, the results show that as the value of the conditional precision increases it is preferable to update the components of the latent state in one step, but the value of τ for this change depends on the value of N; as N increases, this value increases. Comparing the sampling schemes, it seems that the interweaving samplers have higher ESS/sec, whereas the baseline of the interweaving strategy is of minor influence. Comparing the non-centered and centered parameterization, the results show that keeping ϕ fixed, it seems that as the value of τ increases, the non-centered parameterization exhibits lower autocorrelation than the centered (in most cases where $\phi < 0.85$). Comparing the blocking strategy, the 3-block sampler show higher ESS/sec. Finally, as the sample size increases, the sampling efficiency deteriorates, for example, when $n = 6,000$ the ESS/sec is on average two

times larger than its corresponding value when the sample size is doubled.

5.5 Real data results

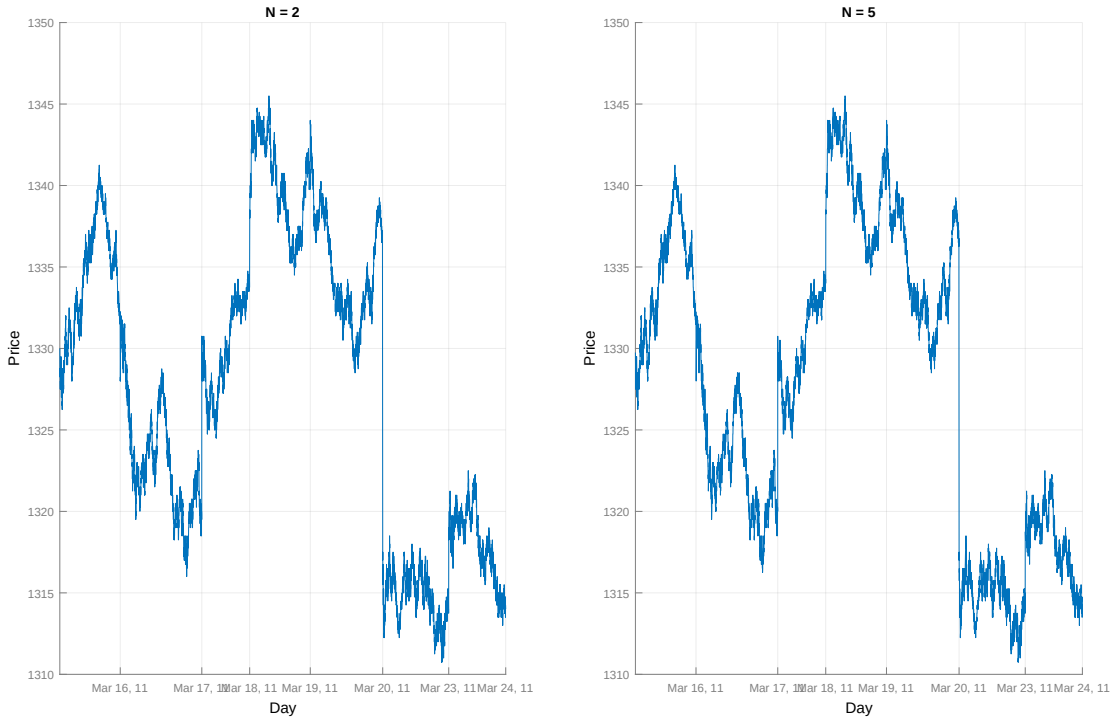


Figure 5.1: Time series plot of ES prices. The sample period covers Monday to Friday from May 16th 2011 to May 24th 2011, 9:00 a.m. to 5:00 p.m. eastern time. The labels along the x-axis depict the last transaction of the corresponding day. Tick size equals \$0.25.

The ES data are compressed into trade bars of size two and five. The total number of observations is $262,356$ from which $39,924$ $(222,432)$ are reported on the morning (afternoon) time period with $12,100$ $(74,957)$ active trades when $N = 2$. Similarly, when $N = 5$, the observations are $104,926$ from which $15,961$ $(88,965)$ are reported on the morning (afternoon) with $8,203$ $(46,965)$ active trades. The tick size is considered to be 1/4th of a dollar for each segment. In Figure 5.1, the ES is plotted from May 16 through May 24, 9 a.m. to 5 p.m. for both trade bar sizes. Besides, Table 1.3 shows the percentage of the transaction price changes between 09:00 and 17:00 after the aggregation. Variable definitions are presented in Table 5.10.

If we do not make a specific reference to which data segment we are referring to, it means that we imply both examined time periods; The same it is true for the trade bar sizes. The curves

Table 5.10: Variable Definitions.

Variable Name	Description
	Continuous Variables
t_i	the time at which the last trade of the ith trade bar occurs
Δt_i	$\triangleq t_i - t_{i-1}$
τ_i	$\triangleq \log(\Delta t_i + 1)$
$V_i^{\text{b,j}}$	the log total volume on the jth best bid quote right after the last trade of the ith trade bar
$V_i^{\text{a,j}}$	the log total volume on the jth best ask quote right after the last trade of the ith trade bar
V_i^{mo}	the log volume of the last trade of the ith trade bar
	Discrete Variables
SP_i	the spread (in tick) instantaneously after the ith trade bar
	Dummy Variables
Up_i	1: if at the end of the ith trade bar the price moves down, 0: moves down
BMO_i	1: if the last trade of the ith trade bar is a buy market order, 0: otherwise
MO_i	type of market data update action of the last trade of the ith trade bar 1: change, 0: delete

Index j corresponds to the jth limit order level, $j = 1, 2$. Index i corresponds to the ith trade bar.

with blue (black) color refer to the morning (afternoon) period. The Bayesian estimates reported in the tables represent posterior means and standard deviations in parenthesis. For the maximum likelihood in parenthesis is reported the asymptotic standard error.

5.5.1 Part A

In order to investigate whether the probability of a price movement is predictable based on past sequences of the trading process and limit orders, we include some possible covariates given by $\boldsymbol{x}_{i-1}^{\text{A},\top} = (1, A_{ii}, D_{ii}, \text{Up}_{ii}, \tau_{ii}, \text{BMO}_{ii}, V_{ii}^{\text{mo}}, SP_{ii}, V_{ii}^{\text{b,1}}, V_{ii}^{\text{b,2}}, V_{ii}^{\text{a,1}}, V_{ii}^{\text{a,2}}, \text{MO}_{ii})$ where $ii \in \{i-2, i-1\}$. After testing out insignificant explanatory variables we end up with $\boldsymbol{x}_{i-1}^{\text{A},\top} = A_{i-1}$ and $\boldsymbol{x}_{i-1}^{\text{A},\top} = (A_{i-1}, A_{i-2})$ during the morning and afternoon period, respectively.

As a short summary, the length of burn-in, acceptance rate, and thinning to draw 2000 MCMC thinned samples for the activity binomial GLM, GLARMA, AR(1), WN and RW(1) model are shown in Table 5.11. The parameters of the observation driven models are estimated

Table 5.11: Length of burn-in, acceptance rate, and thinning to draw 2000 MCMC thinned samples for the activity binomial GLM, GLARMA, AR(1), WN and RW(1) model. The parameters of the observation driven models are estimated (in one step) using the AM algorithm, while the parameters of all parameter driven model except the WN are estimated with the CNC algorithm combined with the 3-block sampler in which the latent path is updated using the '1-1' method. The AR(1)'s ratio shows the acceptance rate of $(\phi, \tau, \boldsymbol{\beta})$, while the WN's and RW's ratio show the acceptance rate of $(\tau, \boldsymbol{\beta})$.

Model		$N = 2$		$N = 5$	
Binomial -		Morning	Afternoon	Morning	Afternoon
GLM	burn-in	1e+4	1e+4	1e+4	1e+4
	thinning	50	50	50	50
	ratio (in %)	22.02	16.96	22.24	16.14
GLARMA	burn-in	2e+4	2e+4	2e+4	2e+4
	thinning	100	100	100	100
	ratio (in %)	16.90	28.46	24.96	27.13
AR(1)	burn-in	2e+4	2e+4	2e+4	2e+4
	thinning	300	900	900	900
	ratio (in %)	(47.19,37.78, 21.80)	(49.22,40.83, 29.09)	(41.39,43.78, 19.09)	(41.02,41.59, 27.41)
WN	burn-in	2e+4	2e+4	2e+4	2e+4
	thinning	200	200	200	200
	ratio (in %)	(41.78,22.15)	(38.83,29.55)	(44.91,32.67)	(42.40,27.98)
RW(1)	burn-in	2e+4	2e+4	2e+4	2e+4
	thinning	2000	1500	800	120
	ratio (in %)	(33.96,24.24)	(43.18,31.89)	(47.90,36.06)	(47.16,36.83)

(in one step) using the AM algorithm, while the parameters of all parameter driven model except the WN are estimated with the CNC algorithm combined with the 3-block sampler in which the latent path is updated using the '1-1' method. IBIS is initialized with MCMC draws based on the first 1000 (500) observations during the morning (afternoon) period.

Tables 5.12 - 5.15 list the parameter estimates of the activity model, along with the running time. Firstly, it is clear that AM and IBIS sample estimates agree well with the maximum likelihood estimates and by increasing the number of particles does not significantly affect the estimated parameters, meanwhile, the computational cost grows with the number of particles. Judging by the influence of the regression parameters, the tables show that the number of active

Table 5.12: MCMC parameter estimation of the activity binomial GLM with the ES, May 16th to May 20th. 'Run-time' returns the execution time (in minutes). The Bayesian estimates represent posterior means and standard deviations in parenthesis based on 2000 draws. Model: $A_i \sim \text{Binomial}(N, \pi_i)$, $\text{logit}(\pi_i) = \boldsymbol{x}_{i-1}^\top \boldsymbol{\beta}$ and $N \in \{2, 5\}$. Morning: $\boldsymbol{x}_{i-1}^\top = A_{i-1}$. Afternoon: $\boldsymbol{x}_{i-1}^\top = (A_{i-1}, A_{i-2})$. Priors: $\boldsymbol{\beta} \sim \mathcal{N}_3(\mathbf{0}_3, 10^3\mathbf{I}_3)$. Parameter estimation algorithm: AM.

	$N = 2$		$N = 5$	
Parameter	Morning	Afternoon	Morning	Afternoon
β_0	-1.877	-1.917	-1.882	-1.961
	(0.015)	(0.007)	(0.018)	(0.009)
β_1	1.034	1.143	0.683	0.979
	(0.021)	(0.009)	(0.023)	(0.009)
β_2	-	0.377	-	0.139
	-	(0.009)	-	(0.009)
Run-time	1.17	6.32	0.59	2.44

movements is positive for all past period values, indicating that past active movements tend to increase the change of subsequent active movements, but the influence decays down at lag two. However, the RW(1) model returns a negative β_2 value for the afternoon period when $N = 5$. Regarding the posterior mean of ϕ, when $N = 2$ the AR1's value is negative during the morning and insignificant on the afternoon subset, while when $N = 5$ it is almost 0.3. On the other hand, GLARMA's value approximates one when $N = 5$ and it is more than 0.8 when $N = 2$. Concerning the posterior mean of the conditional standard deviation of the latent process, we can see that its value is small for the random walk model of order one except for the afternoon period with trade bar of size five which is 0.8. The corresponding values for the rest parameter driven models is around to 1.4 (1.2) over the morning (afternoon) dataset.

Table 5.13: Maximum likelihood, AM and IBIS parameter estimation for the activity binomial GLARMA model with the ES, May 16th to May 20th over the morning period. 'Run-time' returns the execution time (in minutes). The Bayesian estimates represent posterior means and standard deviations in parenthesis. For MLE, in parenthesis, is reported the asymptotic standard error. AM is initialized with mle yielding 2000 draws. Model: $A_i \sim \text{Binomial}(N, \pi_i)$ with $\text{logit}(\pi_i) = \boldsymbol{x}_{i-1}^\top \boldsymbol{\beta} + Z_i$, $Z_i = \phi Z_{i-1} + \delta \varepsilon_{i-1}$, $\varepsilon_i = (A_i - N\pi_i)/\sqrt{N\pi_i(1-\pi_i)}$, $Z_0 = \varepsilon_0 = 0$. $\boldsymbol{x}_{i-1}^\top = A_{i-1}$. Priors: $(\phi+1)/2 \sim \mathcal{B}e(0.5, 0.5)$, $\log(\delta) \sim \mathcal{N}(0, 10^3)$, $\boldsymbol{\beta} \sim \mathcal{N}_3(\mathbf{0}_3, 10^3 \mathbf{I}_3)$.

			IBIS		
			Number of Particles		
Parameter	MLE	AM	500	1000	2000
Panel A: $N = 2$					
ϕ	0.824	0.825	0.825	0.825	0.825
	(0.022)	(0.027)	(0.026)	(0.026)	(0.027)
δ	0.099	0.098	0.098	0.098	0.098
	(0.011)	(0.013)	(0.013)	(0.013)	(0.013)
β_0	-1.798	-1.799	-1.799	-1.799	-1.799
	(0.017)	(0.019)	(0.020)	(0.020)	(0.020)
β_1	0.741	0.745	0.742	0.745	0.745
	(0.034)	(0.040)	(0.039)	(0.041)	(0.041)
Run-time	0.00	7.65	3.28	6.49	12.47
Panel B: $N = 5$					
ϕ	0.977	0.976	0.977	0.976	0.976
	(0.005)	(0.007)	(0.007)	(0.008)	(0.007)
δ	0.018	0.019	0.019	0.019	0.019
	(0.002)	(0.003)	(0.003)	(0.003)	(0.003)
β_0	-1.854	-1.855	-1.851	-1.853	-1.854
	(0.018)	(0.024)	(0.023)	(0.022)	(0.023)
β_1	0.623	0.623	0.618	0.621	0.622
	(0.019)	(0.025)	(0.023)	(0.022)	(0.024)
Run-time	0.00	1.80	1.31	2.55	5.09

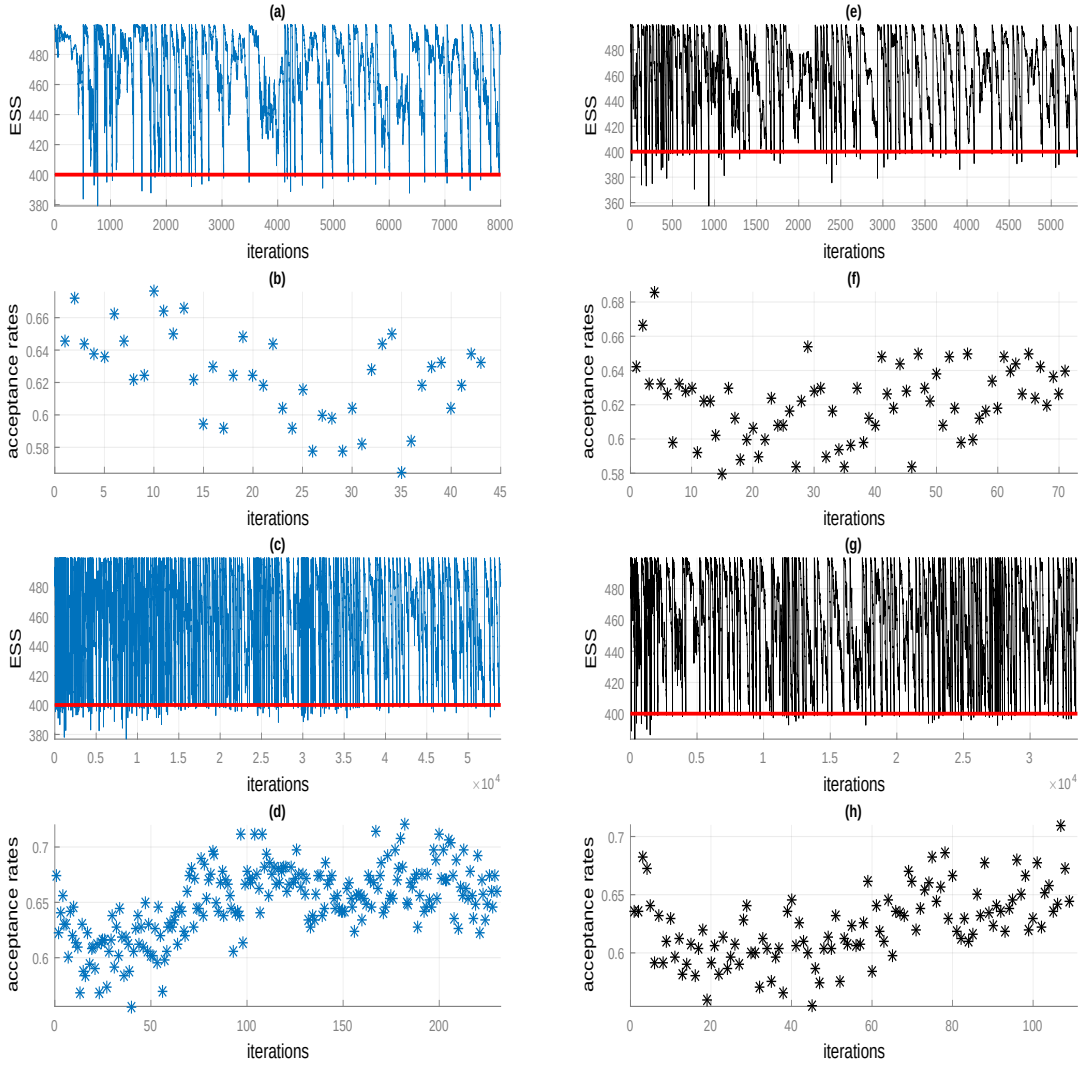
Table 5.14: Maximum likelihood, AM and IBIS parameter estimation for the activity binomial GLARMA model for ES, May 16th to May 20th over the afternoon period. 'Run-time' returns the execution time (in minutes). The Bayesian estimates represent posterior means and standard deviations in parenthesis. For MLE, in parenthesis, is reported the asymptotic standard error. AM is initialized with mle yielding 2000 draws. Model: $A_i \sim \text{Binomial}(N, \pi_i)$ with $\text{logit}(\pi_i) = \boldsymbol{x}_{i-1}^\top \boldsymbol{\beta} + Z_i$, $Z_i = \phi Z_{i-1} + \delta \varepsilon_{i-1}$, $\varepsilon_i = (A_i - N\pi_i)/\sqrt{N\pi_i(1-\pi_i)}$, $Z_0 = \varepsilon_0 = 0$. $\boldsymbol{x}_{i-1}^\top = (A_{i-1}, A_{i-2})$. Priors: $(\phi+1)/2 \sim \mathcal{B}e(0.5, 0.5)$, $\log(\delta) \sim \mathcal{N}(0, 10^3)$, $\boldsymbol{\beta} \sim \mathcal{N}_3(\mathbf{0}_3, 10^3 \mathbf{I}_3)$.

			IBIS		
			Number of Particles		
Parameter	MLE	AM	500	1000	2000
Panel A: $N = 2$					
ϕ	0.858	0.858	0.858	0.858	0.858
	(0.006)	(0.006)	(0.006)	(0.006)	(0.006)
δ	0.126	0.126	0.126	0.126	0.126
	(0.005)	(0.006)	(0.005)	(0.005)	(0.005)
β_0	-1.752	-1.753	-1.752	-1.752	-1.753
	(0.009)	(0.010)	(0.010)	(0.010)	(0.010)
β_1	0.826	0.827	0.826	0.827	0.827
	(0.014)	(0.010)	(0.016)	(0.016)	(0.016)
β_2	0.163	0.164	0.164	0.163	0.163
	(0.010)	(0.011)	(0.012)	(0.011)	(0.011)
Run-time	0.01	43.18	26.84	45.19	82.39
Panel B: $N = 5$					
ϕ	0.989	0.989	0.989	0.989	0.989
	(0.001)	(0.002)	(0.002)	(0.002)	(0.002)
δ	0.012	0.012	0.012	0.012	0.012
	(0.001)	(0.001)	(0.001)	(0.001)	(0.001)
β_0	-1.925	-1.926	-1.927	-1.925	-1.925
	(0.009)	(0.012)	(0.012)	(0.011)	(0.012)
β_1	0.942	0.942	0.942	0.942	0.942
	(0.008)	(0.010)	(0.010)	(0.010)	(0.010)
β_2	0.103	0.103	0.103	0.102	0.103
	(0.007)	(0.010)	(0.010)	(0.010)	(0.009)
Run-time	0.00	16.62	8.95	16.67	32.41

Table 5.15: MCMC parameter estimation of the activity binomial AR(1), WN and RW(1) model with the ES, May 16th to May 20th. The last model is estimated by the algorithm CNC combined with the 3-block sampler in which the latent path is updated using the 1-1'method. 'Run-time' returns the execution time in hours. The Bayesian estimates represent posterior means and standard deviations in parenthesis based on 2000 samples. Model: $A_i \sim \text{Binomial}(N, \pi_i)$ with $\text{logit}(\pi_i) = h_i$, $h_i = \boldsymbol{x}_{i-1}^\top \boldsymbol{\beta} + \phi(h_{i-1} - \boldsymbol{x}_{i-2}^\top \boldsymbol{\beta}) + \varepsilon_i$, $\varepsilon_i \sim \mathcal{N}(0, 1/\tau)$. Morning: $\boldsymbol{x}_{i-1}^\top = A_{i-1}$. Afternoon: $\boldsymbol{x}_{i-1}^\top = (A_{i-1}, A_{i-2})$. Priors: $(\phi + 1)/2 \sim \mathcal{B}e(0.5, 0.5)$, $\tau \sim \text{Gamma}(1\text{e}{-3}, 1\text{e}{+3})$, $\boldsymbol{\beta} \sim \mathcal{N}_3(\mathbf{0}_3, 10^3\mathbf{I}_3)$. * indicates not significant parameter at the 5% level.

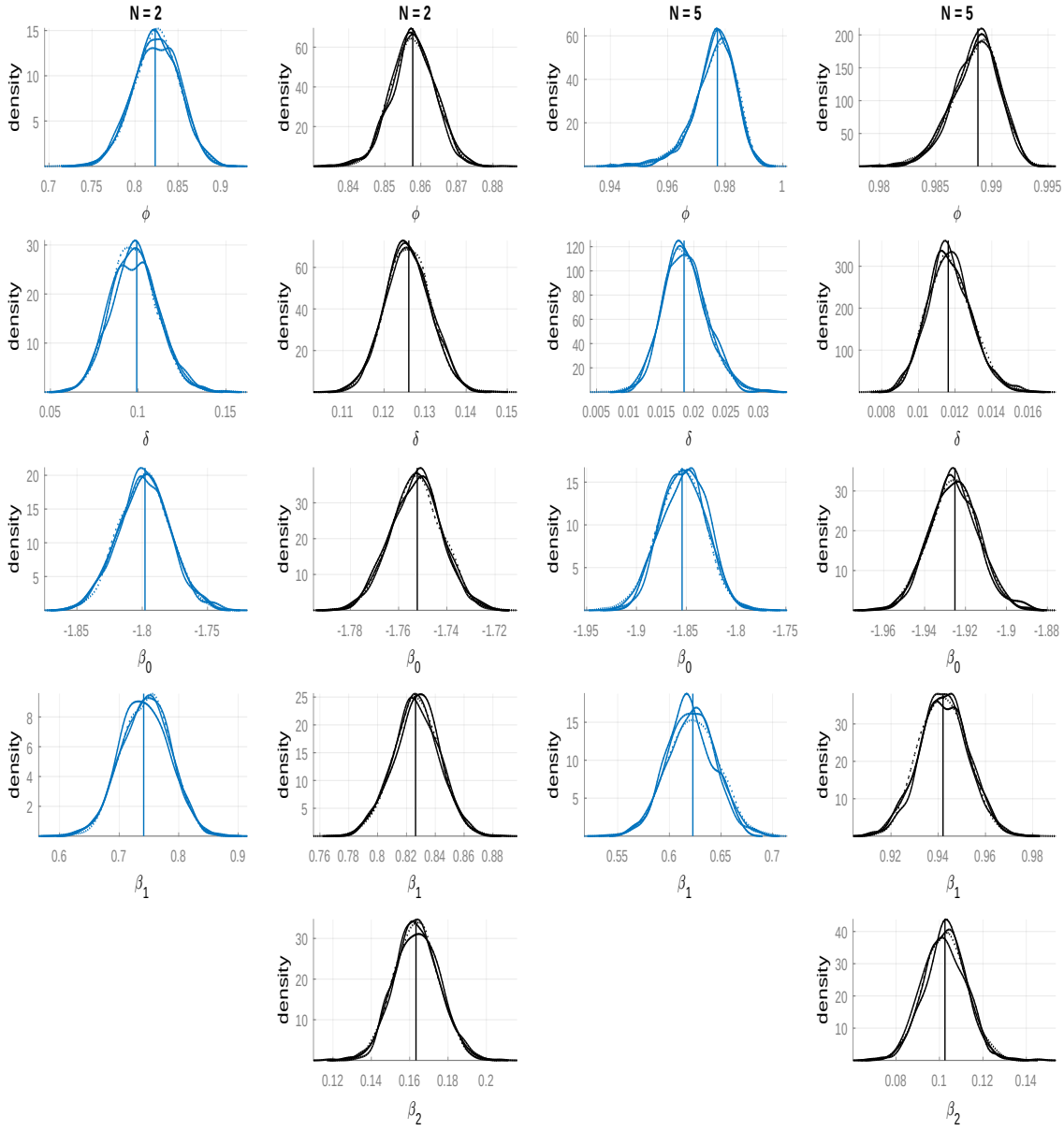
	$N = 2$			$N = 5$		
Parameter	AR(1)	WN	RW(1)	AR(1)	WN	RW(1)
Panel A: Morning period						
ϕ	-0.186	-	-	0.244	-	-
	(0.029)	-	-	(0.052)	-	-
σ	1.420	1.441	0.018	1.100	1.124	0.047
	(0.033)	(0.031)	(0.003)	(0.025)	(0.023)	(0.007)
β_0	-2.601	-2.521	-1.947	-2.146	-2.308	-1.754
	(0.034)	(0.030)	(0.094)	(0.046)	(0.029)	(0.231)
β_1	1.647	1.370	0.982	0.508	0.829	0.598
	(0.054)	(0.034)	(0.023)	(0.077)	(0.033)	(0.024)
Run-time	1.441	0.920	5.863	1.752	0.391	0.941
Panel B: Afternoon period						
ϕ	-0.016*	-	-	0.261	-	-
	(0.015)	-	-	(0.018)	-	-
σ	1.404	1.405	0.013	1.184	1.209	0.811
	(0.013)	(0.013)	(0.001)	(0.010)	(0.009)	(0.012)
β_0	2.575	-2.570	-1.996	-2.298	-2.486	-1.231*
	(0.015)	(0.013)	(0.152)	(0.023)	(0.015)	(1.182)
β_1	1.539	1.516	1.110	0.849	1.233	0.133
	(0.027)	(0.015)	(0.010)	(0.032)	(0.016)	(0.016)
β_2	0.522	0.531	0.345	0.211	0.186	-0.374
	(0.016)	(0.013)	(0.010)	(0.016)	(0.015)	(0.014)
Run-time	27.749	6.166	34.266	9.835	2.185	1.439

Figure 5.2 shows the ESS along the iterations as well as the acceptance rates of the move steps with 500 particles for the activity binomial GLM. Since the resampling threshold is set to 80%, a resample-move is triggered when ESS drops below 400 (red line). As expected, the frequency of the resample-moves steps seems to decrease over time. On the other hand, the acceptance rate is quite high along the iterations. Similar results are obtained for the other particles of the corresponding plots as well as for the GLARMA process; these figures are not presented here.



Model: $A_i \sim \text{Binomial}(N, \pi_i)$, $\text{logit}(\pi_i) = \boldsymbol{x}_{i-1}^\top \boldsymbol{\beta}$, $\boldsymbol{x}_{i-1}^\top = (1, \tau_{i-1}, V_{i-1}^{\text{b,l}}, V_{i-1}^{\text{a,l}}, V_{i-1}^{\text{mo}}, \text{Act}_{i-1})$ and $N \in \{2, 5\}$. Priors: $\boldsymbol{\beta} \sim \mathcal{N}_5(\mathbf{0}_5, 10^3 \mathbf{I}_5)$.

Figure 5.2: Results of IBIS estimation for the activity binomial GLM with the ES, May 16th to May 20th, during the morning (blue) and afternoon period (black), $N = 2$ (rows 1, 2) and $N = 5$ (rows 3, 4). ESS along the iterations and acceptance rate at each move step. The number of particles is 500 and a resample-move step is triggered when ESS drops below 400 (red line).

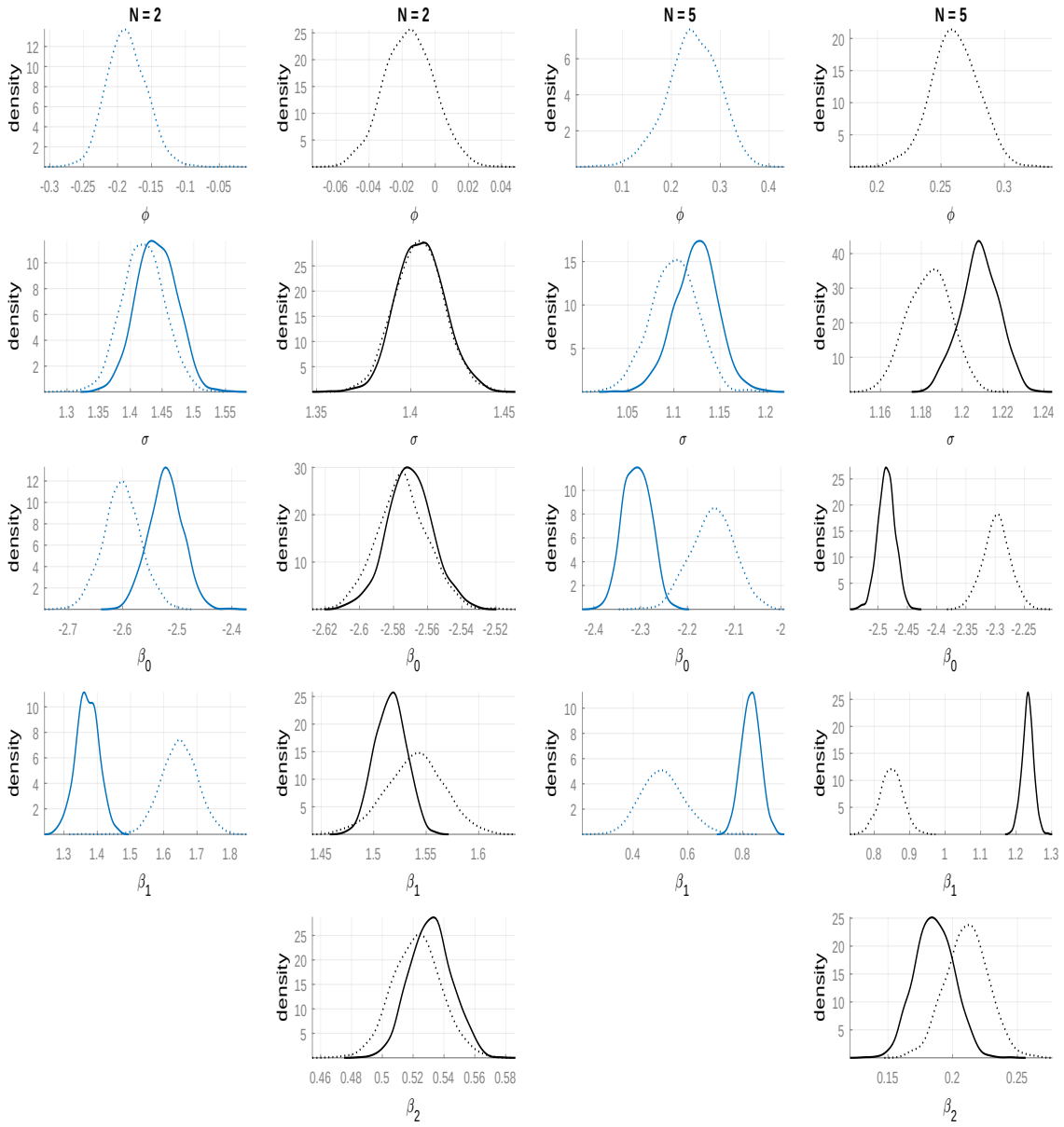


Model: $A_i \sim \text{Binomial}(N, \pi_i)$ with $\text{logit}(\pi_i) = \boldsymbol{x}_{i-1}^\top \boldsymbol{\beta} + Z_i$, $Z_i = \phi Z_{i-1} + \delta \varepsilon_{i-1}$, $\varepsilon_i = (A_i - N\pi_i)/\sqrt{N\pi_i(1-\pi_i)}$, $Z_0 = \varepsilon_0 = 0$. Morning: $\boldsymbol{x}_{i-1}^\top = A_{i-1}$. Afternoon: $\boldsymbol{x}_{i-1}^\top = (A_{i-1}, A_{i-2})$. Priors: $(\phi + 1)/2 \sim \mathcal{B}e(0.5, 0.5)$, $\tau \sim \text{Gamma}(1\text{e}{-3}, 1\text{e}{+3})$, $\boldsymbol{\beta} \sim \mathcal{N}_d d(\mathbf{0}_d, 10^d \mathbf{I}_d)$.

Figure 5.3: Bayesian marginal posterior densities for parameters estimated by the activity binomial GLARMA model, estimated from AM (dotted curve) and IBIS (solid, dashed and dash-dotted curve for 500, 1000 and 2000 particles, respectively) samples using a kernel density. Vertical lines represent the mle. The data set we are analyzing is ES, May 16th to May 20th during the morning (blue curve) and afternoon period (black curve).

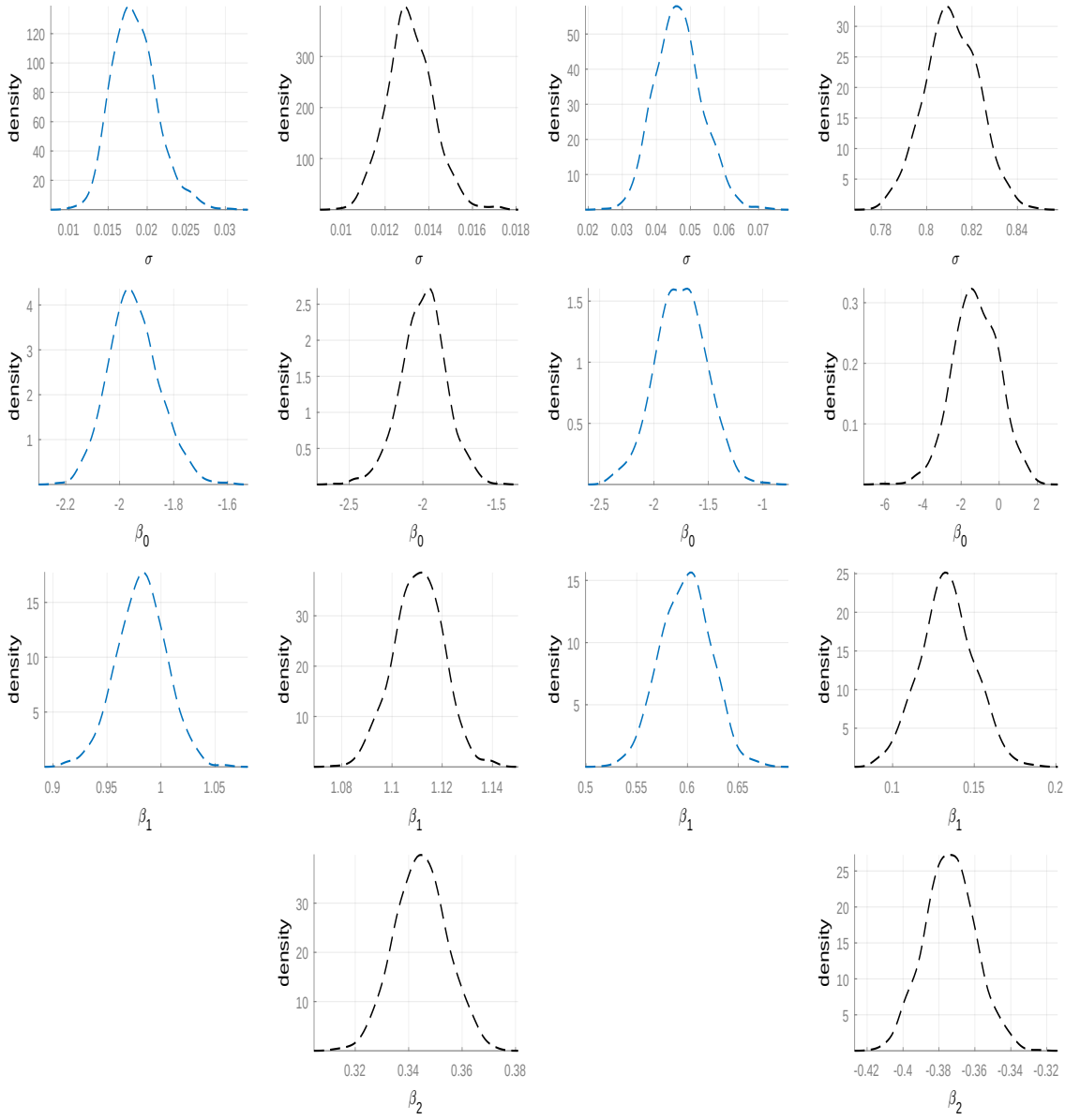
Figure 5.3 illustrates the marginal posterior densities of the estimated parameters for the the activity binomial GLARMA model estimated from AM and IBIS samples using a kernel density; the two methods produce nearly identical posterior distributions. Note that by increas-

ing the number of particles we obtain similar estimates. The figure also marks the maximum likelihood estimators; it seems that they coincide with the posterior mode. Figures 5.4-5.5 illustrate the marginal posterior densities of the estimated parameters for the binomial parameters driven models.



Model: $A_i \sim \text{Binomial}(N, \pi_i)$ with $\text{logit}(\pi_i) = h_i, h_i = \boldsymbol{x}_{i-1}^\top \boldsymbol{\beta} + \phi(h_{i-1} - \boldsymbol{x}_{i-2}^\top \boldsymbol{\beta}) + \varepsilon_i, \varepsilon_i \sim \mathcal{N}(0, 1/\tau)$.
Morning: $\boldsymbol{x}_{i-1}^\top = A_{i-1}$. Afternoon: $\boldsymbol{x}_{i-1}^\top = (A_{i-1}, A_{i-2})$.

Figure 5.4: Marginal posterior densities (with 2000 MCMC thinned samples) for parameters estimated by the activity binomial AR(1) (dotted curve) and WN (solid curve) model.

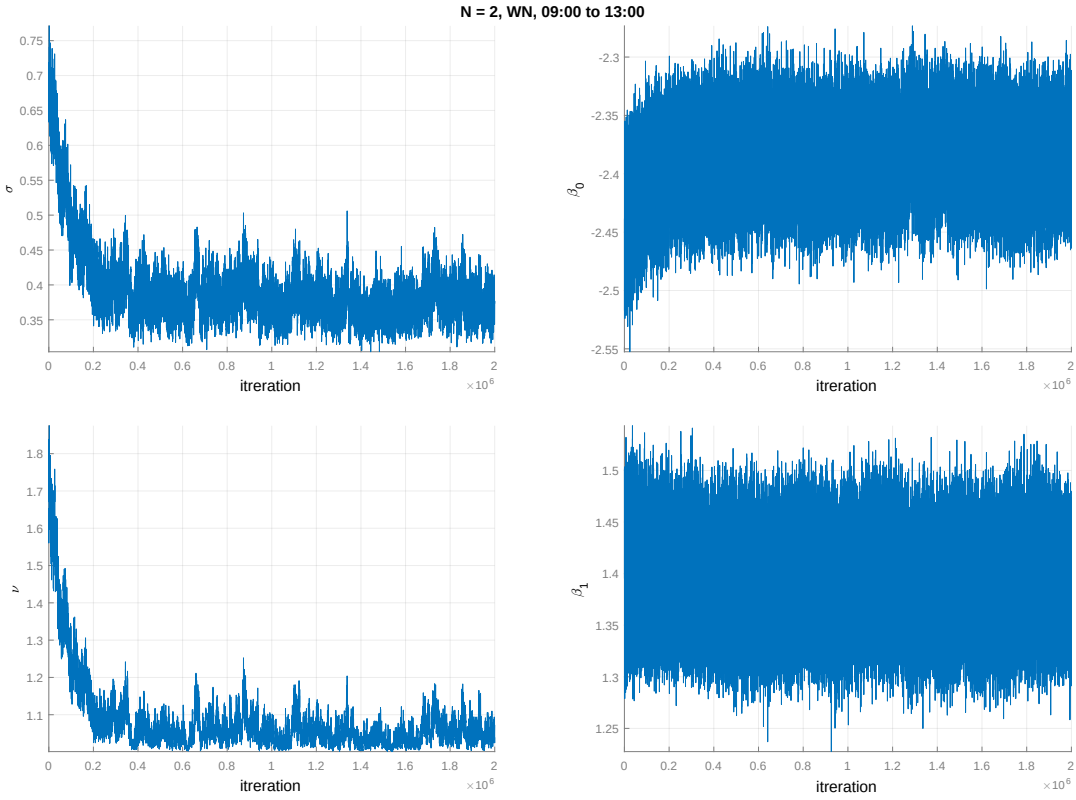


Model: $A_i \sim \text{Binomial}(N, \pi_i)$ with $\text{logit}(\pi_i) = h_i$, $h_i = \boldsymbol{x}_{i-1}^\top \boldsymbol{\beta} + (h_{i-1} - \boldsymbol{x}_{i-2}^\top \boldsymbol{\beta}) + \varepsilon_i$, $\varepsilon_i \sim \mathcal{N}(0, 1/\tau)$. Morning: $\boldsymbol{x}_{i-1}^\top = A_{i-1}$. Afternoon: $\boldsymbol{x}_{i-1}^\top = (A_{i-1}, A_{i-2})$. Priors: $(\phi + 1)/2 \sim \mathcal{B}e(0.5, 0.5)$, $\tau \sim \text{Gamma}(1\text{e}{-3}, 1\text{e}{+3})$, $\boldsymbol{\beta} \sim \mathcal{N}_d d(\mathbf{0}_d, 10^d \mathbf{I}_d)$.

Figure 5.5: Marginal posterior densities (with 2000 MCMC thinned samples) for parameters estimated by the activity binomial RW(1) model.

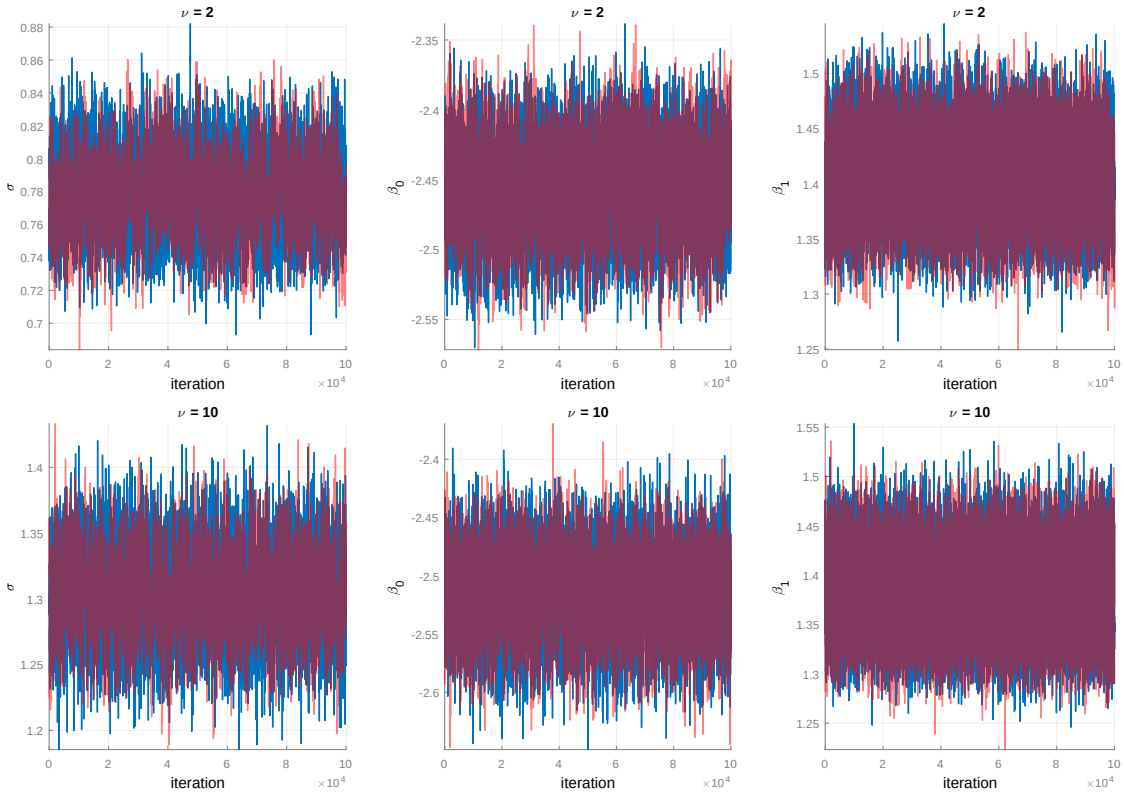
We have also considered the WN model with Student's t errors with ν degrees of freedom. However, the MCMC algorithms produce posterior samples with very high autocorrelation, for example, Figure 5.6 displays trace plots for parameters during the morning with $N = 2$. It appears that the regression parameters are less correlated, while the serial correlation of the rest parameters is stronger even for long lags. When $N = 5$, only the parameter ν appears to

have poor mixing, while the rest parameters are mixing well. The significant autocorrelation suggests that chains require a thinning interval which is calculated as the maximum value of a vector whose the jth element is the first lag that the serial correlation of the jth parameter drops to zero. However, this value is high implying a high computational cost. Instead of thinning the posterior chains, we pre-specify ν as a known constant to deal with high autocorrelation and we notice that the mixing behaviour improves. For example, Figure 5.7 illustrates MCMC trace plots for two and ten degrees of freedom. The samples are generated by the CNC algorithm combined with the 3-block sampler in which the latent path is updated with the '1-1' and the all' (red curve) method. All algorithms run 100,000 iterations after a burn-in of $50,000$. The figure shows that the algorithms achieve well mixing chain.



Model: $A_i \sim \text{Binomial}(2, \pi_i)$ with $\text{logit}(\pi_i) = h_i$, $h_i = \boldsymbol{x}_{i-1}^\top \boldsymbol{\beta} + (h_{i-1} - \boldsymbol{x}_{i-2}^\top \boldsymbol{\beta}) + \sigma\varepsilon_i$, $\varepsilon_i \sim t_\nu(0,1)$. Morning: $\boldsymbol{x}_{i-1}^\top = A_{i-1}$. Priors: $\tau \sim \text{Gamma}(1\text{e}{-3}, 1\text{e}{+3})$, $1/\nu \sim \text{Uniform}(0,1)$, $\boldsymbol{\beta} \sim \mathcal{N}_2(\mathbf{0}_2, 10^3\mathbf{I}_2)$.

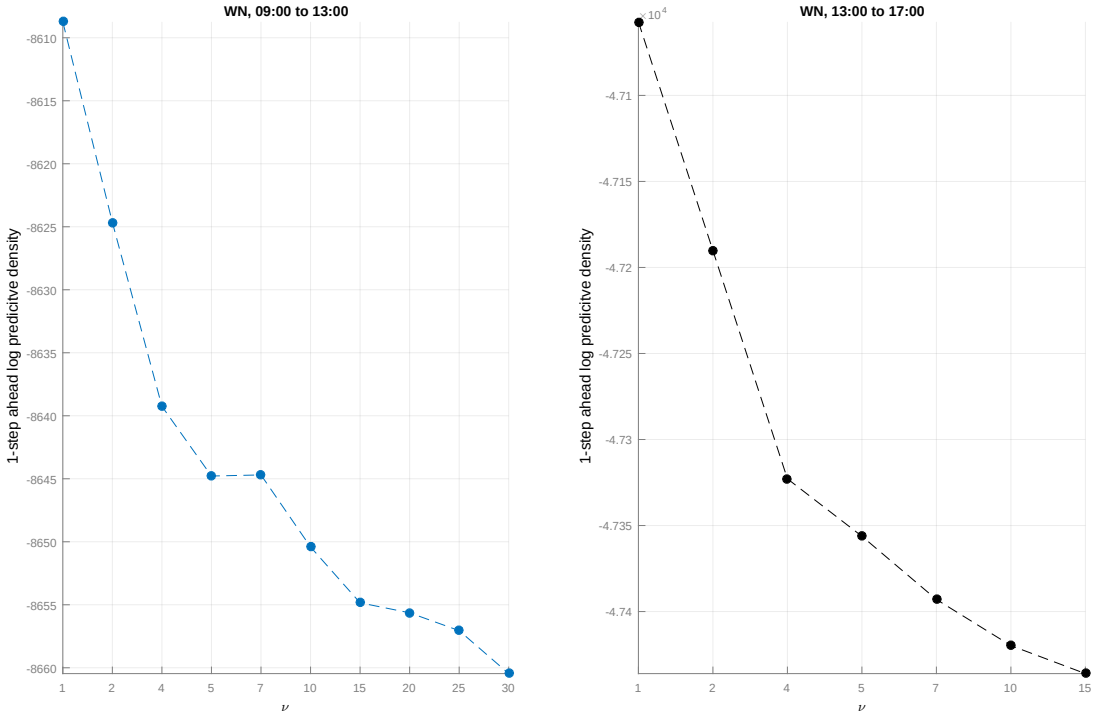
Figure 5.6: MCMC trace plots for parameters estimated by the activity binomial WN model with Student's t errors with unknown degrees of freedom with the ES, May 16th to May 20th, during the morning, $N = 2$.



Model: $A_i \sim \text{Binomial}(N, \pi_i)$ with $\text{logit}(\pi_i) = h_i$, $h_i = \boldsymbol{x}_{i-1}^\top \boldsymbol{\beta} + (h_{i-1} - \boldsymbol{x}_{i-2}^\top \boldsymbol{\beta}) + \sigma\varepsilon_i$, $\varepsilon_i \sim t_\nu(0, 1)$. Morning: $\boldsymbol{x}_{i-1}^\top = A_{i-1}$. Priors: $\tau \sim \text{Gamma}(1\text{e}{-3}, 1\text{e}{+3})$, $\boldsymbol{\beta} \sim \mathcal{N}_2(\mathbf{0}_2, 10^3\mathbf{I}_2)$.

Figure 5.7: MCMC trace plots for parameters estimated by the activity binomial WN model with Student's t errors with $\nu = \{2, 10\}$ with the ES, May 16th to May 20th, during the morning, $N = 2$.

Figure 5.8 displays the effect of the choice of ν for trade bar of size two, based on the (one-step ahead) log predictive score of a thinned chain with 500 draws for the last two days of the dataset. The figure suggests that it is maximized for one degree of freedom for all time periods. When $N = 5$, the results suggest that it is maximized for values of ν greater than 30, approximating the WN component model with normal errors. Regarding the latent path strategy, the results (which are not presented here) show that it is better to use the 1-1 strategy and as the value of ν increases the ESS/sec of the parameters increases.

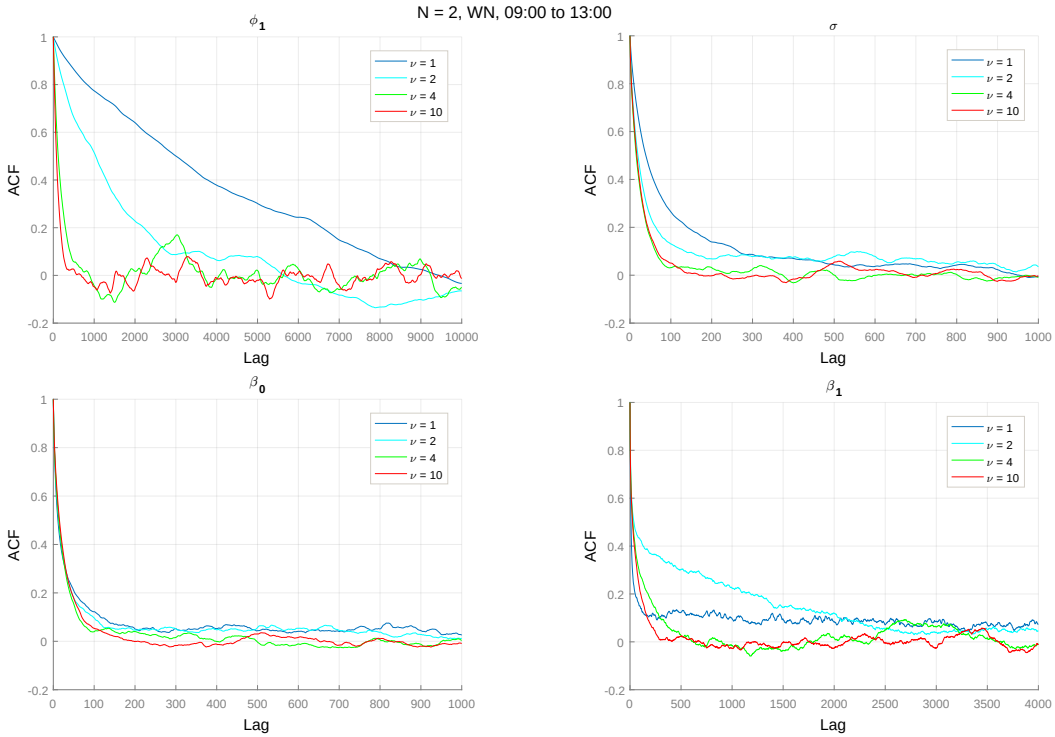


Model: $A_i \sim \text{Binomial}(2, \pi_i)$ with $\text{logit}(\pi_i) = h_i$, $h_i = \boldsymbol{x}_{i-1}^\top \boldsymbol{\beta} + (h_{i-1} - \boldsymbol{x}_{i-2}^\top \boldsymbol{\beta}) + \sigma\varepsilon_i$, $\varepsilon_i \sim t_\nu(0, 1)$. Morning: $\boldsymbol{x}_{i-1}^\top = A_{i-1}$. Priors: $\tau \sim \text{Gamma}(1\text{e}{-3}, 1\text{e}{+3})$, $\boldsymbol{\beta} \sim \mathcal{N}_2(\mathbf{0}_2, 10^3\mathbf{I}_2)$.

Figure 5.8: Forecasting performance: (one-step ahead) log predictive score versus ν for the activity binomial model with the ES, May 23rd & 24th 2011, 9 a.m. - 1 p.m. (left panel) and 1 p.m. - 5 p.m. (right panel), $N = 2$.

Concerning the AR(1) model with Student's t errors with ν degrees of freedom, if ν is not known the sampler generates also highly autocorrelated samples. However, keeping ν fixed, the autocorrelation of the estimated parameters in some cases is still high. Figure 5.9 displays the MCMC autocorrelation plots during the morning with $N = 2$ and $\nu \in \{1, 2, 4, 10\}$. All algorithms run 100,000 iterations after a burn-in of $50,000$. When $\nu = 1$ or 2, it appears that the serial correlation of the ϕ parameter is strong even for long lags, and drops more quickly for larger degrees of freedom. During the afternoon period, we conclude to similar results, however the serial correlations are a little larger. To deal with autocorrelation, we pre-specify ν as a known constant and when $\nu \in \{1, 2\}$ we also keep ϕ parameter fixed and examine the (one-step ahead) log predictive score. The results show that for trade bars of size two, the best choice of ν is one with $\phi = -0.1$ (0) during the morning (afternoon) period, while for trade bars of size five the best choice of ν appears to be greater than 30, approximating the AR(1)

component model with normal errors. However, we left the model incomplete.



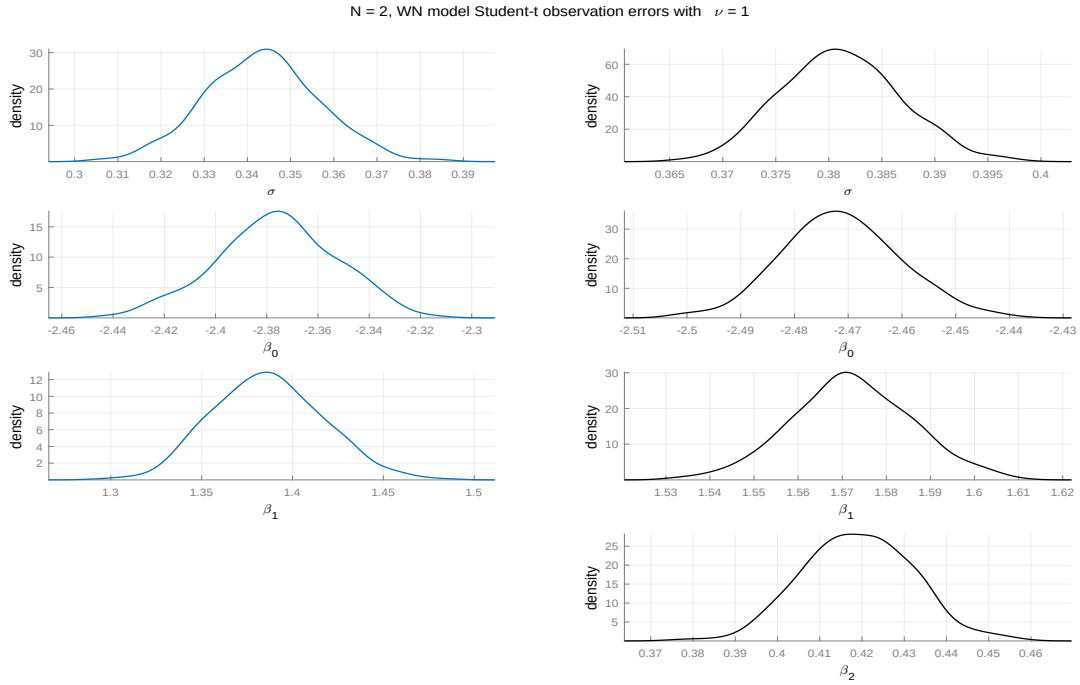
Model: $A_i \sim \text{Binomial}(2, \pi_i)$ with $\text{logit}(\pi_i) = h_i$, $h_i = \boldsymbol{x}_{i-1}^\top \boldsymbol{\beta} + (h_{i-1} - \boldsymbol{x}_{i-2}^\top \boldsymbol{\beta}) + \sigma\varepsilon_i$, $\varepsilon_i \sim t_\nu(0, 1)$. Morning: $\boldsymbol{x}_{i-1}^\top = A_{i-1}$. Priors: $\tau \sim \text{Gamma}(1\text{e}{-3}, 1\text{e}{+3})$, $\boldsymbol{\beta} \sim \mathcal{N}_2(\mathbf{0}_2, 10^3\mathbf{I}_2)$.

Figure 5.9: MCMC autocorrelation plots of the activity binomial AR(1) model with Student's t errors with ν degrees of freedom with the ES, May 16th to May 20th, during the morning, $N = 2$.

Table 5.16 lists the parameter estimates of the price activity binomial WN model for trade bars of size two, using the CNC algorithm combined with the 3-block sampler in which the latent path is updated one at time, conditional on all other values of the state process and parameters, along with the running time (in hours). The Bayesian estimates reported in the table represent posterior means and standard deviations in parenthesis. The algorithm runs 320,000 (500,000) iterations for the morning (afternoon) dataset with burn-in of 20,000 iterations and the autocorrelation is reduced by retaining only every 160th (250th) iteration of the chain. The posterior mean of σ is 0.34 (0.38) for the morning (afternoon) dataset, which is four time smaller that the corresponding model with normal errors. The influence of the regression parameters are positive and slightly smaller than the corresponding model with Gaussian errors. Figure 5.10 illustrates the marginal posterior densities.

Table 5.16: MCMC parameter estimation of the activity binomial WN model with Student's t errors with one degree of freedom with the ES, May 16th to May 20th, $N = 2$. The model is estimated by the algorithm CNC combined with the 3-block sampler in which the latent path is updated using the 1-1'method. 'Run-time' returns the execution time in hours. The Bayesian estimates represent posterior means and standard deviations in parenthesis based on 2000 samples. Model: $A_i \sim \text{Binomial}(2, \pi_i)$ with $\text{logit}(\pi_i) = h_i$, $h_i = \boldsymbol{x}_{i-1}^\top \boldsymbol{\beta} + \phi(h_{i-1} - \boldsymbol{x}_{i-2}^\top \boldsymbol{\beta}) + \sigma\varepsilon_i$, $\varepsilon_i \sim t_1(0,1)$. Morning: $\boldsymbol{x}_{i-1}^\top = A_{i-1}$. Afternoon: $\boldsymbol{x}_{i-1}^\top = (A_{i-1}, A_{i-2})$. Priors: $\tau \sim \text{Gamma}(1\text{e}{-3}, 1\text{e}{+3})$, $\boldsymbol{\beta} \sim \mathcal{N}_d(\mathbf{0}_d, 10^3\mathbf{I}_d)$.

	Parameter			
	σ	β_0	β_1	β_2
Morning	0.343	-2.376	1.386	-
	(0.013)	(0.023)	(0.030)	-
Afternoon	0.381	-2.471	1.572	0.419
	(0.006)	(0.011)	(0.014)	(0.013)
Run-time	Morning: 1.15 hr		Afternoon: 12.05 hrs	



Model: $A_i \sim \text{Binomial}(2, \pi_i)$ with $\text{logit}(\pi_i) = h_i$, $h_i = \boldsymbol{x}_{i-1}^\top \boldsymbol{\beta} + \phi(h_{i-1} - \boldsymbol{x}_{i-2}^\top \boldsymbol{\beta}) + \sigma\varepsilon_i$, $\varepsilon_i \sim t_1(0,1)$. Morning: $\boldsymbol{x}_{i-1}^\top = A_{i-1}$. Afternoon: $\boldsymbol{x}_{i-1}^\top = (A_{i-1}, A_{i-2})$. Priors: $\tau \sim \text{Gamma}(1\text{e}{-3}, 1\text{e}{+3})$, $\boldsymbol{\beta} \sim \mathcal{N}_d(\mathbf{0}_d, 10^3\mathbf{I}_d)$.

Figure 5.10: Marginal posterior densities (with 2000 MCMC thinned samples) for parameters estimated by the activity binomial WN model with Student's t errors with one degree of freedom. The data set we are analyzing is ES, May 16th to May 20th during the morning (blue curve) and afternoon period (black curve).

5.5.2 Part B

In order to investigate whether the probability of a price movement is predictable based on past sequences of the trading process and limit orders, we include the covariates

$$\boldsymbol{x}_{i-1}^{\mathrm{D},\top} = (1, A_{ii}, D_{ii}, \mathrm{Up}_{ii}, \tau_{ii}, \mathrm{BMO}_{ii}, V_{ii}^{\mathrm{mo}}, SP_{ii}, V_{ii}^{\mathrm{b},1}, V_{ii}^{\mathrm{b},2}, V_{ii}^{\mathrm{a},1}, V_{ii}^{\mathrm{a},2}, \mathrm{MO}_{ii})$$

where $ii \in \{i-2, i-1\}$. After testing out insignificant explanatory variables we end up with $\boldsymbol{x}_{i-1}^{\mathrm{D},\top} = (1, V_{i-1}^{\mathrm{b},1}, V_{i-1}^{\mathrm{a},1}, \mathrm{Up}_{i-1}, \mathrm{BMO}_{i-1})$.

About the observation driven component model, the results of the GLARMA model show that the ϕ and δ coefficients are not significant during the morning period, so only a GLM component model is discussed for the morning dataset. Relative to the parameter driven component model, trace plots of the AR(1) model with normal errors show that the samples from the full conditional of the estimated persistence parameter are not well mixed with very strong autocorrelation. For this reason, we only examine a RW(1) and a WN model. Besides, we have also considered the corresponding models with Student's t errors with unknown ν degrees of freedom. Due to the bad mixing and convergence behaviour of ν, it is pre-specified as a known constant and the mixing behaviour improves. The best choice of the degrees of freedom is determined by optimizing the model's one-step ahead predictive likelihood. However, the results suggest that it is maximized for values of the conditional variance, $\sigma^2 \triangleq 1/\tau$ greater than 30, approximating the component models with normal errors.

As a short summary, the length of burn-in, acceptance rate, and thinning to draw 2000 MCMC thinned samples for the different models are shown in Table 5.17. The observation driven models are estimated (in one step) using the AM algorithm, while the parameter driven models are estimated with the CNC algorithm combined with the 3-block sampler in which the latent path is updated in one move. In order to reduce the correlations, the regression parameters are updated in groups which are $\beta_0, (\beta_1, \beta_2), (\beta_3, \beta_4)$. IBIS algorithm is initialized with MCMC draws based on the first 1000 observations.

Table 5.17: Length of burn-in, acceptance rate, and thinning to draw 2000 MCMC thinned samples for the upward movements GLM, GLARMA, RW(1) and WN model. The parameters of the first two models are estimated (in one step) using the AM algorithm. The parameters of the last two models are estimated with the CNC algorithm combined with the 3-block sampler using the 1-1'method, while the regression parameters are updated in groups. The ratio of the parameter driven models show the acceptance rate of $(\tau, \beta_0, \beta_{1:2}, \beta_{3:4})$.

Model		$N = 2$		$N = 5$	
Binomial -		Morning	Afternoon	Morning	Afternoon
GLM	burn-in	2e+4	2e+4	2e+4	2e+4
	thinning	130	150	100	150
	ratio (in %)	15.85	14.27	15.55	14.41
GLARMA	burn-in	-	5e+4	-	6e+4
	thinning	-	120	-	100
	ratio (in %)	-	31.27	-	32.73
RW(1)	burn-in	2e+4	2e+4	2e+4	2e+4
	thinning	1800	1800	1800	1300
	ratio (in %)	(38.12,31.99, 19.93,24.68)	(41.13,41.64, 26.18,22.17)	(41.13,41.64, 26.18,22.17)	(44.24,41.25, 22.11,21.72)
WN	burn-in	2e+4	2e+4	2e+4	2e+4
	thinning	700	1400	1100	1400
	ratio (in %)	(38.28,43.798, 30.46,24.55)	(41.49,44.34, 24.82,22.17)	(49.76,41.49, 28.63,24.76)	(41.49,44.34, 24.82,22.17)

Tables 5.18 - 5.20 list the parameter estimates of the activity binomial models, along with the running time. Firstly, it is clear that AM and IBIS sample estimates agree well with the maximum likelihood estimates and by increasing the number of particles does not significantly affect the estimated parameters, meanwhile, the computational cost grows with the number of particles. Judging by the influence of the regression parameters, the tables show that the influence of lagged movement, is negative indicating that if the price moved on the last trade bar then there is a large chance that this movement will be reversed if there is an active batch trade. Relative to the impact of buy market orders, we observe a significant negative influence at lag

Table 5.18: AM parameter estimates of the upward movements binomial GLM with the ES, May 16th to May 20th. 'Run-time' returns the execution time (in minutes). The Bayesian estimates represent posterior means and standard deviations in parenthesis based on 2000 draws. Model: $D_i|A_i > 0 \sim \text{Binomial}(A_i, \pi_i)$ with $\text{logit}(\pi_i) = \boldsymbol{x}_{i-1}^\top \boldsymbol{\beta}$, $\boldsymbol{x}_{i-1}^\top = (1, V_{i-1}^{\text{b,1}}, V_{i-1}^{\text{a,1}}, \text{Up}_{i-1}, \text{BMO}_{i-1})$ and $A_i \in \{1, \cdots, N\}$, $N \in \{2, 5\}$. Priors: $\boldsymbol{\beta} \sim \mathcal{N}_5(\mathbf{0}_5, 10^3\mathbf{I}_5)$.

Parameter	MLE	AM	IBIS Number of Particles		
			500	1000	2000
Panel A: Morning, $N = 2$					
β_0	1.135	1.132	1.135	1.136	1.130
	(0.141)	(0.141)	(0.141)	(0.139)	(0.144)
β_1	0.134	0.135	0.135	0.135	0.136
	(0.018)	(0.018)	(0.018)	(0.018)	(0.018)
β_2	-0.198	-0.198	-0.198	-0.198	-0.198
	(0.019)	(0.019)	(0.019)	(0.019)	(0.019)
β_3	-0.198	-0.198	-0.197	-0.199	-0.197
	(0.046)	(0.046)	(0.045)	(0.047)	(0.046)
β_4	-1.471	-1.473	-1.471	-1.471	-1.473
	(0.046)	(0.046)	(0.043)	(0.044)	(0.046)
Run-time	0.000	1.445	0.536	0.951	1.851
Panel B: Morning, $N = 5$					
β_0	0.588	0.594	0.602	0.585	0.587
	(0.125)	(0.125)	(0.115)	(0.120)	(0.120)
β_1	0.062	0.061	0.061	0.062	0.062
	(0.016)	(0.016)	(0.014)	(0.016)	(0.015)
β_2	-0.101	-0.102	-0.103	-0.101	-0.101
	(0.017)	(0.017)	(0.016)	(0.016)	(0.016)
β_3	-0.104	-0.103	-0.103	-0.103	-0.102
	(0.041)	(0.041)	(0.040)	(0.039)	(0.040)
β_4	-0.693	-0.693	-0.691	-0.691	-0.692
	(0.042)	(0.042)	(0.040)	(0.042)	(0.041)
Run-time	0.000	0.870	0.311	0.592	1.163
Panel C: Afternoon, $N = 2$					
β_0	1.296	1.296	1.293	1.301	1.301
	(0.082)	(0.082)	(0.082)	(0.085)	(0.084)
β_1	0.135	0.135	0.135	0.134	0.134
	(0.009)	(0.009)	(0.009)	(0.009)	(0.009)
β_2	-0.177	-0.177	-0.177	-0.178	-0.178
	(0.009)	(0.009)	(0.009)	(0.009)	(0.010)
β_3	-0.201	-0.202	-0.201	-0.201	-0.201
	(0.019)	(0.019)	(0.018)	(0.018)	(0.019)
β_4	-1.853	-1.853	-1.854	-1.852	-1.853
	(0.019)	(0.019)	(0.018)	(0.018)	(0.019)
Run-time	0.000	6.102	3.736	6.509	12.224
Panel D: Afternoon, $N = 5$					
β_0	0.618	0.620	0.622	0.617	0.620
	(0.072)	(0.072)	(0.068)	(0.073)	(0.072)
β_1	0.057	0.056	0.056	0.056	0.056
	(0.008)	(0.008)	(0.008)	(0.007)	(0.008)
β_2	-0.089	-0.090	-0.090	-0.089	-0.090
	0.008)	(0.008)	(0.007)	(0.008)	(0.008)
β_3	-0.080	-0.080	-0.079	-0.080	-0.080
	(0.016)	(0.016)	(0.016)	(0.017)	(0.017)
β_4	-0.751	-0.751	-0.751	-0.750	-0.752
	(0.017)	(0.017)	(0.016)	(0.017)	(0.017)
Run-time	0.000	3.861	2.082	3.721	6.855

Table 5.19: Maximum likelihood, AM and IBIS parameter estimation of the upward movements GLARMA model with the ES, May 16th to May 20th over the afternoon period. 'Run-time' returns the execution time (in minutes). The Bayesian estimates represent posterior means and standard deviations in parenthesis. For MLE, in parenthesis, is reported the asymptotic standard errors and the solver runs from multiple initialization points. Model: $D_i|A_i > 0 \sim \text{Binomial}(A_i, \pi_i)$ with $\text{logit}(\pi_i) = \boldsymbol{x}_{i-1}^\top \boldsymbol{\beta} + Z_i$, $Z_i = \phi Z_{i-1} + \delta\varepsilon_{i-1}$, $\varepsilon_i = (D_i - A_i\pi_i)/\sqrt{A_i\pi_i(1-\pi_i)}$, $Z_0 = \varepsilon_0 = 0$, $\boldsymbol{x}_{i-1}^\top = (1, V_{i-1}^{\text{b,1}}, V_{i-1}^{\text{a,1}}, \text{Up}_{i-1}, \text{BMO}_{i-1})$ and $A_i \in \{1, \cdots, N\}$, $N \in \{2, 5\}$. Priors: $(\phi+1)/2 \sim \mathcal{B}e(0.5, 0.5)$, $\log(\delta) \sim \mathcal{N}(0, 10^3)$, $\boldsymbol{\beta} \sim \mathcal{N}_5(\mathbf{0}_5, 10^3\mathbf{I}_5)$.

Parameter	MLE	AM	IBIS Number of Particles		
			500	1000	2000
Panel A: $N = 2$					
ϕ	0.981	0.981	0.981	0.981	0.981
	(0.005)	(0.004)	(0.004)	(0.004)	(0.004)
δ	0.033	0.033	0.033	0.033	0.033
	(0.005)	(0.004)	(0.004)	(0.004)	(0.004)
β_0	1.471	1.468	1.469	1.469	1.468
	(0.126)	(0.101)	(0.099)	(0.099)	(0.101)
β_1	0.151	0.152	0.152	0.151	0.152
	(0.012)	(0.010)	(0.010)	(0.010)	(0.010)
β_2	-0.219	-0.218	-0.219	-0.218	-0.218
	(0.014)	(0.011)	(0.011)	(0.011)	(0.011)
β_3	-0.193	-0.193	-0.193	-0.194	-0.193
	(0.025)	(0.019)	(0.019)	(0.019)	(0.019)
β_4	-1.908	-1.908	-1.907	-1.908	-1.906
	(0.025)	(0.020)	(0.019)	(0.021)	(0.020)
Run-time	0.034	19.331	2.882	4.783	9.173
Panel B: $N = 5$					
ϕ	0.973	0.971	0.973	0.972	0.972
	(0.018)	(0.010)	(0.010)	(0.010)	(0.010)
δ	0.027	0.028	0.027	0.028	0.028
	(0.012)	(0.006)	(0.006)	(0.006)	(0.006)
β_0	0.720	0.722	0.723	0.716	0.722
	(0.166)	(0.083)	(0.080)	(0.082)	(0.084)
β_1	0.063	0.063	0.062	0.064	0.063
	(0.016)	(0.009)	(0.008)	(0.008)	(0.008)
β_2	-0.109	-0.109	-0.109	-0.109	-0.109
	(0.018)	(0.009)	(0.009)	(0.009)	(0.009)
β_3	-0.077	-0.077	-0.077	-0.076	-0.077
	(0.035)	(0.017)	(0.016)	(0.017)	(0.017)
β_4	-0.793	-0.792	-0.790	-0.793	-0.792
	(0.036)	(0.018)	(0.018)	(0.019)	(0.017)
Run-time	0.021	11.192	4.230	7.693	15.322

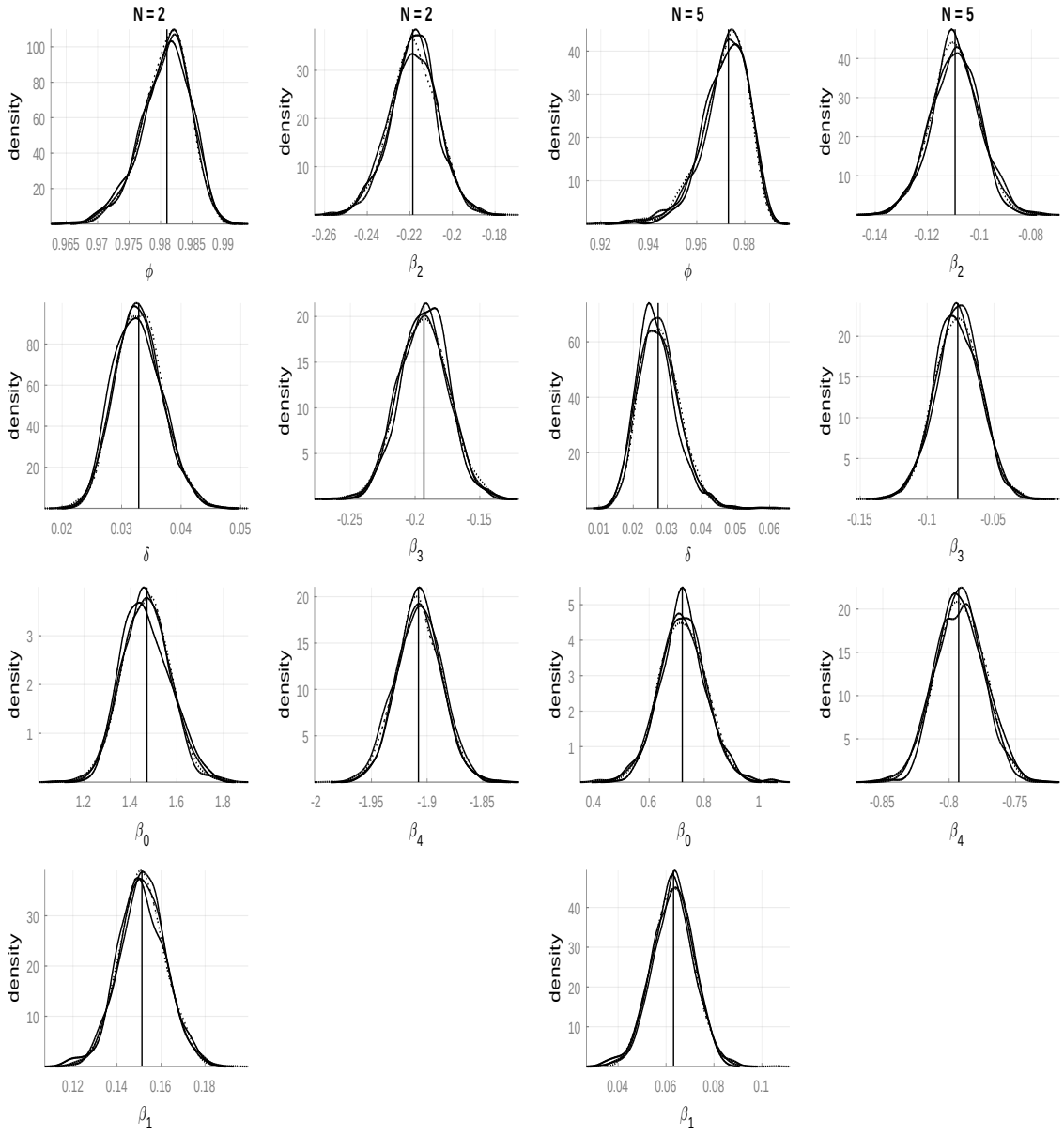
Table 5.20: MCMC parameter estimation of the upward movements binomial WN and RW(1) model for ES, May 16th to May 20th, estimated by the algorithm CNC combined with the 3-block sampler in which the latent path is updated using the 1-1'method. 'Run-time' returns the execution time in hours. The Bayesian estimates represent posterior means and standard deviations in parenthesis based on 2000 samples. Model: $D_i|A_i > 0 \sim \text{Binomial}(A_i, \pi_i)$ with $\text{logit}(\pi_i) = h_i$, $h_i = \boldsymbol{x}_{i-1}^\top \boldsymbol{\beta} + \phi(h_{i-1} - \boldsymbol{x}_{i-2}^\top \boldsymbol{\beta}) + \varepsilon_i$, $\varepsilon_i \sim \mathcal{N}(0, 1/\tau)$, $\boldsymbol{x}_{i-1}^\top = (1, V_{i-1}^{\text{b,l}}, V_{i-1}^{\text{a,l}}, \text{Up}_{i-1}, \text{BMO}_{i-1})$. Priors: $(\phi + 1)/2 \sim \mathcal{B}e(0.5, 0.5)$, $\tau \sim \text{Gamma}(1\text{e}{-3}, 1\text{e}{+3})$, $\boldsymbol{\beta} \sim \mathcal{N}_5(\mathbf{0}_5, 10^3\mathbf{I}_5)$.

	WN				RW(1)			
	$N = 2$		$N = 5$		$N = 2$		$N = 5$	
Parameter	Morn.	Aft.	Morn.	Aft.	Morn.	Aft.	Morn.	Aft.
σ	0.042	0.025	0.034	0.020	0.011	0.007	0.011	0.006
	(0.016)	(0.006)	(0.011)	(0.004)	(0.002)	(0.001)	(0.002)	(0.000)
β_0	1.138	1.292	0.591	0.619	1.083	1.495	0.563	0.716
	(0.147)	(0.084)	(0.128)	(0.073)	(0.176)	(0.134)	(0.155)	(0.109)
β_1	0.135	0.135	0.062	0.056	0.137	0.133	0.062	0.055
	(0.018)	(0.009)	(0.016)	(0.008)	(0.018)	(0.010)	(0.016)	(0.008)
β_2	-0.199	-0.177	-0.101	-0.089	0.204	-0.196	-0.103	-0.099
	(0.020)	(0.009)	(0.017)	(0.008)	(0.020)	(0.010)	(0.018)	(0.009)
β_3	-0.197	-0.201	-0.103	-0.080	-0.193	-0.199	-0.105	-0.080
	(0.046)	(0.019)	(0.042)	(0.017)	(0.047)	(0.018)	(0.042)	(0.017)
β_4	-1.473	-1.854	-0.694	-0.752	-1.484	-1.871	-0.698	-0.760
	(0.046)	(0.019)	(0.042)	(0.017)	(0.048)	(0.020)	(0.042)	(0.018)
Run-time	38.296	461.057	46.538	249.126	161.164	1025.064	70.227	447.698

one, hence the odds of an up movement are larger for sell market orders than buy market orders. Besides, bid (ask) volume at lag one increases (reduces) the chance that the price movement will be upward. According to the estimation results, the posterior mean of ϕ of the GLARMA model approximates one during the afternoon period. Concerning the posterior mean of the conditional standard deviation of the latent path, we can see that WN's value is 0.04 on the morning, meanwhile, the RW's values is smaller (0.01).

Figures 5.11 - 5.12 illustrate the marginal posterior densities of the estimated parameters for the GLARMA and parameter driven model, respectively, using a kernel density, confined to

the morning partition and afternoon partition. It is observed that the posterior distributions of

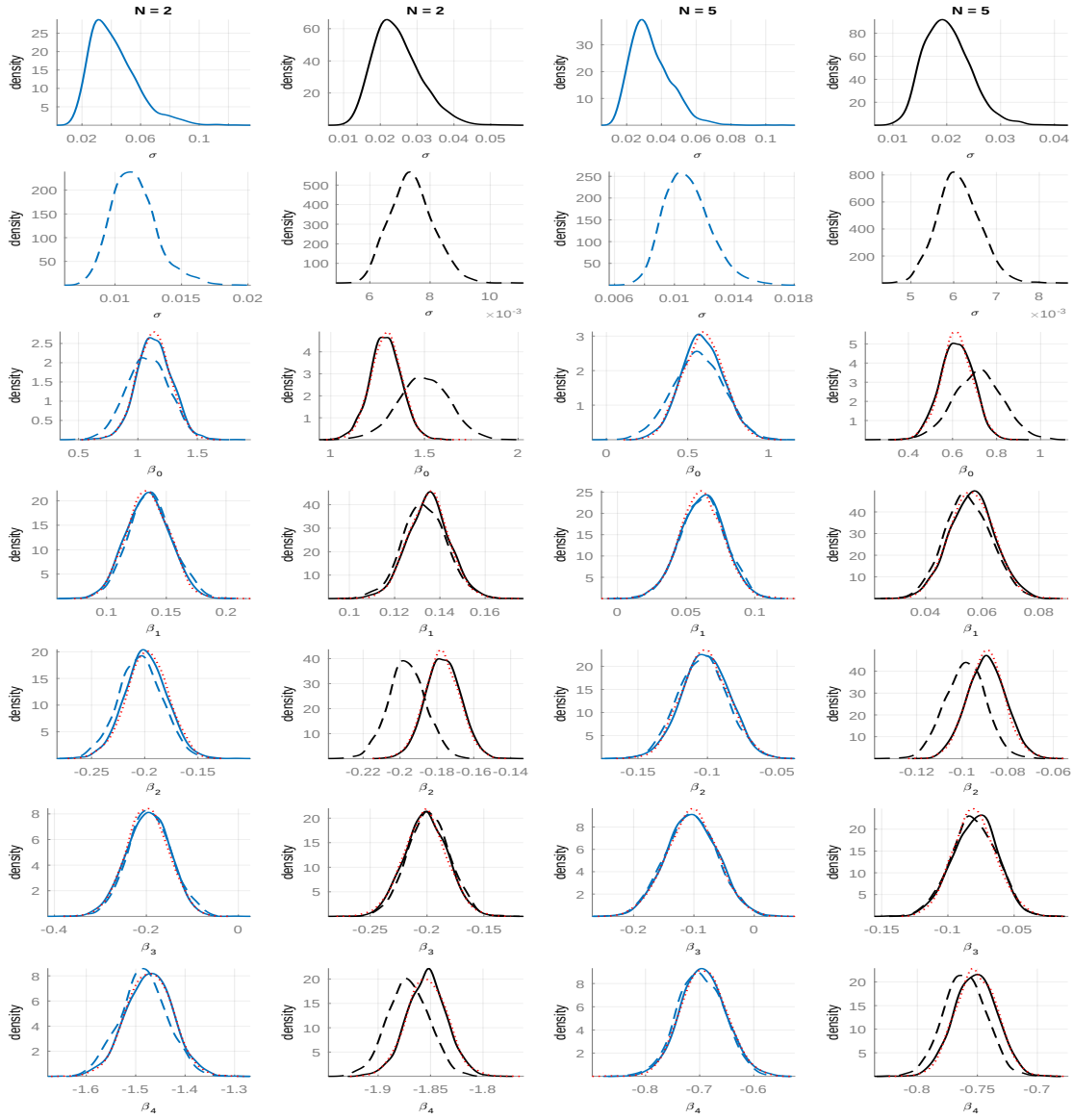


Model: $D_i|A_i > 0 \sim \text{Binomial}(A_i, \pi_i)$ with $\text{logit}(\pi_i) = \boldsymbol{x}_{i-1}^\top \boldsymbol{\beta} + Z_i$, $Z_i = \phi Z_{i-1} + \delta\varepsilon_{i-1}$, $\varepsilon_i = (D_i - A_i\pi_i)/\sqrt{A_i\pi_i(1-\pi_i)}$, $Z_0 = \varepsilon_0 = 0$, $\boldsymbol{x}_{i-1}^\top = (1, V_{i-1}^{\text{b,1}}, V_{i-1}^{\text{a,1}}, \text{Up}_{i-1}, \text{BMO}_{i-1})$. Priors: $(\phi+1)/2 \sim \mathcal{B}e(0.5, 0.5)$, $\log(\delta) \sim \mathcal{N}(0, 10^3)$, $\boldsymbol{\beta} \sim \mathcal{N}_5(\mathbf{0}_5, 10^3\mathbf{I}_5)$.

Figure 5.11: Bayesian marginal posterior densities for parameters estimated by the upward movements GLARMA model, estimated from AM (dotted curve) and IBIS (solid, dashed and dash-dotted curve for 500, 1000 and 2000 particles, respectively) samples using a kernel density. Vertical lines represent the mle. The data set we are analyzing is ES, May 16th to May 20th during the afternoon period.

the regression parameters, through the GLM (red curve), WN and RW(1) models, are almost identical. The results of the WN model with Student's t errors show that the parameter ν appears

to have poor mixing, while the rest parameters are mixing well. As in the previous section, we pre-specify ν as a known constant to deal with high autocorrelation. The results show that the best choice of ν appears to be greater than 30, approximating the corresponding component model with normal errors.



Model: $D_i|A_i > 0 \sim \text{Binomial}(A_i, \pi_i)$ with $\text{logit}(\pi_i) = h_i$, $h_i = \boldsymbol{x}_{i-1}^\top \boldsymbol{\beta} + \phi(h_{i-1} - \boldsymbol{x}_{i-2}^\top \boldsymbol{\beta}) + \varepsilon_i$, $\varepsilon_i \sim \mathcal{N}(0, 1/\tau)$, $\boldsymbol{x}_{i-1}^\top = (1, V_{i-1}^{\text{b,l}}, V_{i-1}^{\text{a,l}}, \text{Up}_{i-1}, \text{BMO}_{i-1})$. Priors: $(\phi + 1)/2 \sim \mathcal{B}e(0.5, 0.5)$, $\tau \sim \text{Gamma}(1\text{e}{-3}, 1\text{e}{+3})$, $\boldsymbol{\beta} \sim \mathcal{N}_5(\mathbf{0}_5, 10^3 \mathbf{I}_5)$.

Figure 5.12: Marginal posterior densities (with 2000 MCMC thinned samples) for parameters estimated by the upward movements RW(1) (dotted curve) and WN (solid curve) model, estimated with the algorithm CNC combined with the 3-block sampler in which the latent path is updated using the 1-1'method. The data set we are analyzing is ES, May 16th to May 20th during the morning (blue curve) and afternoon period (black curve). Red curves denote the corresponding GLMs.

Table 5.21: Relative frequencies of price changes of ES in multiple of tick size, from May 23th 2011 to May 24th 2011, after the trade aggregation. The results are presented as percentages. tick =$0.25.

	$N = 2$				$N = 5$			
	Morning		Afternoon		Morning		Afternoon	
Number (tick)	23/5	24/5	23/5	24/5	23/5	24/5	23/5	24/5
-2	0.06	0.13	0.03	0.02	0.19	0.62	0.12	0.10
-1	12.59	12.11	12.33	11.98	16.47	15.73	14.52	14.32
0	74.92	75.54	75.30	76.09	67.06	67.19	70.81	71.36
1	12.20	11.95	12.30	11.87	15.82	15.62	14.38	14.12
2	0.23	0.27	0.04	0.03	0.84	0.84	0.16	0.10

5.5.3 Predicting price movements

In the section we predict price changes from May 23 to May 24 during the morning and afternoon period. We analyze each day separately allowing us to assess the predicted accuracy over the two days separately. The estimation procedure is described in Section 4.5.3. Table 5.21 shows the percentage of the transaction price changes, after the trade aggregation, for the two sub-periods. It seems that most of the transactions are the same as their previous values, while the distribution of transaction price changes is roughly symmetric. Furthermore, both downward and upward moves greater than one tick occur too rarely.

Tables 5.22 - 5.23 provide the MSE and MAE for price changes reported for the next two days separately with $N = 2$ and $N = 5$, respectively. The first column (A-B) gives the process that is used for each price component, for example, 'GLM-GLARMA' means that the activity component is modeled with a binomial GLM and the upward movements are modeled with a binomial GLARMA model. The last table row (Lag1) reports the corresponding values using a model that assume that the best prediction for tomorrow's market price is simply today's price. Concerning the trade bar of size two, Table 5.22 suggests that throughout the morning the MSE (MAE) is equal for all approaches but GLARMA-GLM. During the afternoon period, the best results for MSE and MAE belong to the GLM-GLM and WN-WN, followed by GLM-GLARMA and WN-RW with differences of 0.1. The worst performance is produced by GLARMA-GLM, RW-WN and RW-RW with differences of approximately 0.3. The comparison of the best performance models and the lag1 model indicates that MSE (MAE) is approximately

Table 5.22: MSE and MAE of one-step-ahead forecasts for predicting price movements with $N = 2$ over May 23th to May 24th. For the activity component model, during the afternoon the parameter ϕ of the AR(1) process is not significant. For the upward movements, during the morning, the parameter ϕ of the GLARMA process is not significant. Lag1 reports the MSE (MAE) values using a model that assume that the best prediction for tomorrow's market price is simply today's. 'WN(t)' defines the white noise model with Student's t errors.

	MSE				MAE			
	Morning		Afternoon		Morning		Afternoon	
A - B	23/5	24/5	23/5	24/5	23/5	24/5	23/5	24/5
GLM - GLM	0.346*	0.349*	0.302*	0.296*	0.332*	0.331*	0.297*	0.293*
GLARMA - GLM	0.576	0.558	0.587	0.568	0.445	0.434	0.439	0.429
GLARMA - GLARMA	-	-	0.575	0.601	-	-	0.437	0.445
GLM - GLARMA	-	-	0.303	0.297	-	-	0.298	0.294
AR1 - WN	0.346*	0.349*	-	-	0.332*	0.331*	-	-
WN - WN	0.346*	0.349*	0.302*	0.296*	0.332*	0.331*	0.297*	0.293*
RW - WN	0.346*	0.349*	0.480	0.478	0.332*	0.331*	0.402	0.399
WN - RW	0.346*	0.349*	0.303	0.297	0.332*	0.331*	0.298	0.294
RW - RW	0.346*	0.349*	0.481	0.460	0.332*	0.331*	0.403	0.389
AR1 - RW	0.346*	0.349*	-	-	0.332*	0.331*	-	-
WN(t) - WN	0.346*	0.349*	0.302*	0.296*	0.332*	0.331*	0.297*	0.293*
WN(t) - RW	0.346*	0.349*	0.303	0.297	0.332*	0.331*	0.297*	0.293*
Lag1	0.708	0.689	0.715	0.689	0.497	0.486	0.492	0.478

two (one and a half) times larger. Concerning the trade bar of size five, Table 5.23 suggests that the RW-WN model has the lowest values, but nevertheless they are still approximately equal to the lag1 model during the morning. On the other hand, during the afternoon the MSE/MAE is two and a half times larger. However this result is misleading because the model almost always predict the most common class, or in other words the zero class.

Concerning the MSEs (MAEs), we can conclude that the binomial AD model is clearly superior than the lag1 model for the trade bar of size two, but for the trade bar of size five it is not. Considering the execution time when $N = 2$, since the MSEs (MAEs) of the best parameter and observation driven models are equal we conclude that it is better to use a binomial GLM for both price factors.

We compare the predicted price changes with the price changes in the original sample data by using four scalar performance measures, namely, accuracy, sensitivity (or recall), preci-

Table 5.23: MSE and MAE of one-step-ahead forecasts for predicting price movements with $N = 5$ over May 23th to May 24th. For the upward movements, during the morning, the parameter ϕ of the GLARMA process is not significant.

	MSE				MAE			
	Morning		Afternoon		Morning		Afternoon	
A - B	23/5	24/5	23/5	24/5	23/5	24/5	23/5	24/5
GLM - GLM	1.328	1.412	1.084	1.058	0.996	1.031	0.869	0.859
GLARMA - GLM	1.328	1.392	1.109	1.109	0.996	1.022	0.886	0.891
GLARMA - GLARMA	-	-	1.115	1.114	-	-	0.889	0.893
GLM - GLARMA	-	-	1.118	1.115	-	-	0.889	0.893
AR1 - WN	1.328	0.996	1.303	1.278	1.403	1.027	1.004	0.995
WN - WN	1.324	0.994	0.963	0.950	1.403	1.027	0.796	0.772
RW - WN	0.901*	0.940*	0.301*	0.296*	0.806*	0.882*	0.295*	0.289*
WN - RW	1.325	1.394	0.964	0.946	0.995	1.022	0.796	0.792
RW - RW	1.020	1.175	0.302	0.296*	0.811	0.886	0.295*	0.289*
AR1 - RW	1.331	1.392	1.303	1.276	0.999	1.022	1.004	0.994
Lag1	0.906	0.941	0.814	0.808	0.627	0.633	0.573	0.564

sion and specificity derived from the confusion matrix [Hajmeer and Basheer, 2003]. Accuracy is a ratio of correctly predicted observations to the total observations. Sensitivity (of the jth event) is the ratio of j-events correctly predicted, where $j \in \{-2, -1, 0, 1, 2\}$. In fact, out of all the j-events, how many of them have been predicted by the algorithm. Specificity (of the jth event) is the ratio of not j-events correctly predicted. Precision (of the jth event) is the percentage of correct j-predictions out of all j-predictions, which are used to calculate the model's ability to classify j-predictions correctly.

The results of the empirical research and their statistical processing are summarized in Tables 5.24 - 5.25. Clearly the model fails to provide a good prediction of upward and downward moves greater than one tick, since there are no instances where the algorithm declares these cases; a reason might be that these moves occur too rarely. In the following we only present the results for May 23 which are similar to May 24. Besides, when $N = 2$ we use the binomial GLM-GLM which is the top-performing choice, and when $N = 5$ during the morning (afternoon) we use the RW-WN (WN-WN) model which is the (second) top-performing choice. If we do not make a specific reference to which data segment we are referring to, it means that

Table 5.24: Prediction accuracy of GLM-GLM measured by confusion matrix over May 23th to May 24th with $N = 2$. The results are presented as percentages. Notes: Sensitivity (sens), specificity (spec), precision (prec) and accuracy (Acc). '-' indicates not a number.

	May 23						May 24					
	Morning			Afternoon			Morning			Afternoon		
Class	Sens	Prec	Spec	Sens	Prec	Spec	Sens	Prec	Spec	Sens	Prec	Spec
-2	0	-	100	0	-	100	0	-	100	0	-	100
-1	51.65	40.32	88.64	61.30	48.89	90.20	50.27	41.53	89.79	61.79	48.66	90.24
0	73.78	82.46	54.08	76.30	84.75	61.89	75.30	82.25	51.39	77.00	85.19	61.76
1	52.07	39.00	88.32	61.05	47.19	89.58	47.29	36.50	88.30	60.66	47.47	90.02
2	0	-	100	0	-	100	0	-	100	0	-	100
Acc	68.22			72.35			68.66			72.91		

we imply both examined time periods; the same is true for the different trade bar sizes.

Firstly, judging by the prediction accuracy, it can be seen that the accuracy decreases as we increase the size of aggregated trades and is affected by the time period. More specifically, when $N = 5$, the prediction ability of the RW-WN and WN-WN model is less than 30%. Despite the fact that this prediction ability is not significant, the model works slightly better than a random guessing. On the other hand, when $N = 2$, the total accuracy of the GLM-GLM yields much better results which are slightly more than 68% and 72% on the morning and afternoon, respectively.

Similarly to accuracy, the sensitivity is affected both of the size of aggregated trades and the time period. More specifically, when $N = 2$, sensitivity in identifying up/down moves over the afternoon achieves 61%, or in other words, 4 of every 10 up/down moves, in reality, are missed by our model and 6 are correctly identified as up/down moves, while during the morning the corresponding value drops. On the other hand, the recall of zero state is over 70%, hence less than 30% of non-zero moves are incorrectly classified as zero moves. When $N = 5$, the ability of the model to identify correctly each event drops to 30%, meanwhile, the recall of zero moves on May 24 drops to 20 on the morning (afternoon) segment.

Concerning the specificity, the RW-WN and WN-WN model achieve specificity of 65% when identifying the non-up/down moves, i.e. 4 of every 10 non-up/down moves in reality are miss-labeled as up/down moves and 6 are correctly labeled as non-up/down moves. The speci-

ficity of zero moves is drops to 60%. When $N = 2$, the ratio of non-up/down moves incorrectly identified as up/down moves is low (10% on average). However, specificity of 54.08% (61.80%) in identifying zero moves during the morning (afternoon) is achieved. It means that the model allows to identify non-zero moves as zero moves at a rate of 46% (38%).

Table 5.25: Prediction accuracy of WN-WN measured by confusion matrix over May 23th to May 24th with $N = 5$. The results are presented as percentages. Notes: Sensitivity (sens), specificity (spec), precision (prec) and accuracy (Acc). '-' indicates not a number.

	May 23						May 24					
	Morning			Afternoon			Morning			Afternoon		
Class	Sens	Prec	Spec	Sens	Prec	Spec	Sens	Prec	Spec	Sens	Prec	Spec
-2	0	-	100	0	-	100	0	-	100	0	-	100
-1	30.32	15.47	67.32	30.06	13.63	63.39	28.62	12.34	65.45	29.04	12.38	65.65
0	30.02	60.00	59.26	21.08	59.39	70.48	27.24	61.70	58.99	26.40	60.73	57.47
1	27.72	12.83	64.63	36.20	13.99	58.79	29.29	12.01	63.98	30.12	12.01	63.74
2	0	-	100	0	-	100	0	-	100	0	-	100
Acc	29.51			25.11			27.24			27.63		

Regarding the precision, the results show that the up/down moves are harder to classify than the zero moves. For example, when $N = 5$, out of the total observations that the model predicts as up/down moves on the morning only 15% are correct, and gets a little worse on the afternoon. On the other hand, precision of 60% in identifying zero moves is achieved. When $N = 2$, more than 80% out of zero predictive moves are correctly classified. On the other hand, around the half out of the upward (downward) predictive moves are truly an upward (downward) move on the afternoon, meanwhile, this value drops to 40% on the morning.

The model which is used for estimating the activity component is crucial (as expected) for the final results of the empirical research. For example, let assume that the values of the activity process on May 23 to May 24 is known, so we only need to predict the upward movements. The models correctly identified more than 80% on each trade bar size. When $N = 2$, the recall of zero state is 90% and rises to 99% when $N = 5$, while the recall of up/down one tick moves is on average 80%. The recall of up/down two tick moves in on average 30% (<10%)for trade bars of size two (five).

Chapter 6

Conclusion and Future Directions

In this chapter some of the most important, general conclusions derived in each chapter of the thesis are summarized and directions for further work are given.

6.1 Summary

We employ the AD model, introduced by Rydberg and Shephard [1998a, 2003], which defines the price change as the product of a binary process for a change in price or not, and a second binary process for a positive or negative change when one occurs. For both components of the price process, the dynamics are modeled using a GLM and a GLARMA model with a logistic link function. Both maximum likelihood and Bayesian estimation via MCMC and SMC methods is conducted. We perform a simulation study to investigate the performance of the proposed algorithms for the estimation of the model.

We extend the approach of Rydberg and Shephard [1998a, 2003] by modelling each price component sequentially using a parameter driven binary generalized model with an AR(1) and a WN in the mean, and estimate each of these components via MCMC methods. We consider two different parametrizations: the centered parametrization and the non-centered parametrization. Our simulation experiments show that the convergence rate of the MCMC algorithms deteriorates when the autoregressive coefficient, ϕ, decreases to an absolute value or the conditional precision, τ, increases. It becomes clear that the interweaving samplers [Yu and Meng, 2011] exhibit lower autocorrelation (and therefore a higher ESS/sec) with respect to all parameters at a very little extra computational cost, whereas the baseline of the interweaving strategy is of minor influence. Regarding the latent path strategy, the results show that as the value of the

conditional precision increases it is preferable to update the components of the latent state in one step, while for the smaller values it is preferable to update its components one at a time. Comparing the blocking strategy of the parameters, the 3-block sampler shows higher ESS/sec most of the time. Finally, as the sample size increases, the sampling efficiency deteriorates.

We apply the above models to the ES. The results show that the influence of lagged price activity, is negative for all past period values, indicating that past active trades tend to decrease the probability of subsequent movements in the price, while this reduction decays down at lag two. Lagged log-durations have a very dramatic positive impact on the chance that a trade moves the transaction price. Besides, a smaller but negative impact is made by the log trading volume. For the quoted volumes on the previous best level, we find a positive impact on the activity process. More specifically, the effect of the buying volume is slightly larger than the impact of the selling volume. The standard deviations are smaller for the afternoon dataset compared to the morning dataset. Furthermore, the influence of lagged price direction, is negative indicating that if the price moved on the last trade then there is a large chance that this movement will be reversed if there is an active trade. Concerning the impact of buy market orders, we observe a significant negative at influence lag one, hence the odds of an up movement are larger for sell market orders than buy market order at lag one. Bid (ask) volume at lag one reduces (increases) the chance that the price movement will be upwards, while the reverse is true at the second lag.

One-step ahead predictions for the next two trading days for each component are obtained. Regarding each component model, the results show that the predictive ability of the proposed models is the same, indicating that there is not much difference between the models, with Brier score (BS) almost two times smaller than a BS of value 0.25 which indicates a random guess. Regarding the information set, the BS of the direction model using only past values of the processes is 3.5 (7) times smaller than the corresponding wider information set during the morning (afternoon) period. However, the improvement of the activity model is significantly lower. Considering the execution time, we conclude that it is better to use a Bernoulli GLM for both price factors. The direction model with the wider information set has a good accuracy, sensitivity and specificity which are over 90%, and some of them even reach 99%. About the activity model, the ability of the model to classify correctly the trade activity is over 75% for the plain model, and it improves 3% with the wider set. Regarding the sensitivity, the results show that the active trades are harder to classify than the inactive. More specifically, 6 of every 10

active trades, in reality, are missed by our model and 4 are correctly identified as active. On the other hand, less than 20% of non-active trades are incorrectly classified as active. Regarding the precision, out of the total observations that the model predicts as active (inactive) trades over 40% (85%) are correct. Regarding the specificity, the model allows to identify active moves as inactive at a rate less than 60%, and non active trades as active at a rate less than 20%. The corresponding measures using the wider set is a little better.

We compare the predicted price changes with the price changes in the original sample data using the activity GLM and direction GLM. The results show that the price direction moves are harder to classify than the zero moves, while on the afternoon the predictive measures are slightly better. More specifically, the ability of the model to classify correctly the price changes is 75%. Less than 20% of non-zero moves are incorrectly classified as zero moves, while 6 of every 10 up (down) moves, in reality, are missed by our model and 4 are correctly identified as up (down) moves. Finally, more than 85% out of zero predictive moves are correctly classified, while this value drops to over 40% in identifying the non-zero price moves. Comparing the model with an alternative model that assume that the best prediction for tomorrow's market price is simply today's price we can conclude the AD model is clearly superior for the trade bar of size two since the MSE (MAE) is approximately two (one and a half) times smaller.

We propose the binomial AD model for the analysis of trade by trade price changes. The model is applied on data where a transaction can move the trade price one tick up or one tick down or not at all. Following the idea of decomposition, we define the price change as the subtraction of two binomial processes. The first process denotes the number of active trades over a short trade interval (denoted trade bar) of a fixed size, and the second process indicates the number of upward moves (if at least one change occurs) during the trade bar. Based on this decomposition, we model each component sequentially using a generalized linear model with logit link function, as well as a first order GLARMA model to investigate serial dependence in the data. Besides, we consider an AR(1), a WN and a random walk of order one latent process into the logit link function, and use MCMC techniques for estimation.

We perform a simulation study to investigate the effectiveness of the binomial parameter driven model using different algorithms. The results show that as the number of trials, N, decreases, the sampling efficiency of all parameters decreases. Concerning the simulation efficiency of the autoregressive coefficient the results show that keeping ϕ fixed, it seems that as the

value of τ increases, the sampling efficiency of ϕ deteriorates. Keeping τ fixed, as the value of ϕ decreases, the sampling efficiency gradually decline. For most underlying parameter values, the interweaving samplers exhibit lower autocorrelation with respect to all parameters, whereas the baseline of the interweaving strategy is of minor influence. Regarding the method for updating the latent state, the results show that as the value of the conditional precision increases it is preferable to update the components of the latent state in one move, rather than one at time conditional on all the others values of the state process (and on the parameter vector). However, the value of τ for this change depends on the value of trials; as N increases, this value increases. About the blocking strategy of the parameter vector, the 3-block sampler shows higher ESS/sec most of the time for all parameters. Finally, as the sample size increases, the sampling efficiency of the parameters deteriorates.

The binomial AD model is applied on ES data using trade bars of size two and five during the morning and afternoon time period. The time gaps between trading days are ignored. We investigate the impact of two lagged values of: the number of active trades, the number of upward moves, the logarithmic duration, the logarithmic traded volume, the bid-ask spread, the logarithmic bid and ask volume on the two best observed quote levels of the LOB, a not aggressive market order dummy variable, a buy market order dummy variable and a dummy variable denoting that a trade moves up the price, on the subsequent price movement for each price component. Regarding the activity process, the results suggest that the number of active trades at the previous trade bar is significant for both time periods, while its penultimate value is significant only for the afternoon subset. Past active movements tend to increase the change of subsequent active movements, while the influence decays down at lag two. Regarding the upward movement process, bid (ask) volume at lag one increases (reduces) the chance that the price movement will be upward, while if the price moved on the last trade then there is a large chance that this movement will be reversed if there is an active trade bar. Finally, the odds of an up movement are larger if the past active trade is a sell market order.

One-step ahead predictions for the next two trading days are obtained. In order to assess the predictive performance of the binomial AD model with the alternative serial dependence structures we apply the MSE and MAE criterion. About the trade bar of size two, the results suggest that throughout the morning the MSE (MAE) is equal for all approaches except the GLARMA-GLM (the notation means that the activity component is modeled with a binomial

GLARMA model and the upward movements are modeled with a binomial GLM model). During the afternoon period, the best results for MSE and MAE belong to the GLM-GLM and WN-WN, followed by GLM-GLARMA and WN-RW with differences of 0.1. The worst performance is produced by GLARMA-GLM, RW-WN and RW-RW with differences of approximately 0.3. Considering the execution time, we conclude that it is better to use a binomial GLM for both price factors. About the trade bar of size five, the results show that the RW-WN (WN-WN) model during the morning (afternoon) perform better. Comparing the top-performing choice with an alternative model that assume that the best prediction for tomorrow's market price is simply today's price we can conclude the binomial AD model is clearly superior for the trade bar of size two since the MSE (MAE) is approximately two (one and a half) times smaller, but for the trade bar of size five it is not.

We compare the predicted price changes with the price changes in the original sample data by using the accuracy which is a ratio of correctly predicted observations to the total observations. It can be seen that the accuracy decreases as we increase the size of aggregated trades and is affected by the time period. More specifically, when $N = 5$, the prediction ability of the RW-WN and WN-WN model is less than 30%. Despite the fact that this prediction ability is not significant, the model works slightly better than a random guessing. On the other hand, when $N = 2$, the total accuracy of the GLM-GLM yields much better results which are slightly more than 68% and 72% on the morning and afternoon, respectively.

Furthermore, we compare three scalar performance measures, namely, sensitivity (or recall), precision and specificity derived from the confusion matrix. Unfortunately, the model fails to provide a good prediction of upward and downward moves greater than one tick, since there are no instances where the algorithm declares these cases. This is because the activity model has a tendency to underestimate the extreme values. When $N = 2$, sensitivity in identifying up/down one tick move over the afternoon achieves 61%, or in other words, 4 of every 10 up/down moves, in reality, are missed by our model and 6 are correctly identified as up/down one tick moves, while during the morning the corresponding value drops. On the other hand, the recall of zero state is over 70%, hence less than 30% of non-zero moves are correctly classified as zero moves. When $N = 5$, the ability of the model to identify correctly each event drops to 30%, meanwhile, the recall of zero moves on May 24 drops to 20 on the afternoon segment. Concerning the specificity, when $N = 2$, the model allows to identify non-zero moves as zero

moves at a rate of 46% (38%) during the morning (afternoon) period, while the ratio of non-up (down) one tick moves incorrectly identified as up (down) one tick moves is 10%. When $N = 5$, 4 of every 10 non-up (down) one tick moves in reality are miss-labeled as up (down) one tick moves and 6 are correctly labeled as non-up (down) one tick moves, while the specificity of zero moves is 60%. Regarding the precision, the results show that the up (down) one tick moves are harder to classify than the zero moves. When $N = 2$, more than 80% out of zero predictive moves are correctly classified. On the other hand, around the half out of the upward (downward) predictive moves are truly an upward (downward) one tick move on the afternoon, meanwhile, this value drops to 40% on the morning. When $N = 5$, precision of 60% in identifying zero moves is achieved. However, out of the total observations that the model predicts as up (down) one tick moves only 15% are correct on the morning, and gets a little worse on the afternoon.

6.2 Further developments

The work presented in this thesis extended the existing empirical literature on trade by trade price changes by explicitly accounting for prices discreteness of the tick data. There are still much work to be done, both in extending and perfecting our study which is presented below.

Modelling the proportion of excess of zeros. Our models have the tendency to underestimate the extreme values in the tails. Therefore, in order to add more flexibility to the tails of the Binomial distribution, we may use a ZIB regression model, proposed by Hall [2000], which is a statistical model to fit binary data with excessive zeros. It is a mixture of observation of only zeros and a weighted binomial distribution. Two unknown parameters in ZIB are probability of observation from only zeros and probability of success in binomial distribution, and for regression, logit link functions can be imposed on these two parameters to incorporate covariates. The analysis may run with different parameters each day, which are assumed to be i.i.d. following a normal distribution.

Alternative processes. We may consider alternative processes such as an autoregressive process of order two and check if the performance of classifiers will be affected or not.

Online estimation. So far we have assumed that all data have been collected before performing any inference, and when we predict a new observation we keep the parameters fixed equal to the

estimated coefficients and we propagate the particles using suitable particle filter algorithms. In contrast to MCMC methods, SMC are useful for online estimation problems, or in other words, if we want to update our analysis after each new observation comes in. Numerous approaches for online estimation (where the parameters are estimated as well) have been proposed; Kantas et al. [2015] presents a comprehensive review.

Alternative data. Studying price changes in other futures market, or stocks traded for example at NYSE with various trade intensities. Besides, it is interested to apply the model to more balanced data and examine its performance of identifying each class.

Bivariate model for price changes and time duration. When combined with a propel model for the time between trades, for example, the autoregressive conditional duration proposed by Engle and Russell [1998] the analysis provides a complete model for the evolution of prices in real time. Following the idea of decomposition, the joint conditional density of the price change and the duration is decomposed into the product of the conditional density of the mark and the marginal density of the arrival times, both conditioned on the past filtration of the joint information set.

Appendices

Appendix A

Data and matching algorithm

This appendix describes the procedure to match trade and limit order book data as well as to identify each order book event as a limit order, cancel order or market order; the process is implemented in MATLAB 2013b.

The front month E-mini S & P 500 futures contracts were recorded electronically on a continuous 23-hour schedule on the Globex between January and December 2011. Globex is active from Monday to Friday. The data is provided as virtual folders, one folder for each month. Within a folder `MAT`-files are included, one file for each day (file names are in the format `YYYMMDD.mat`). Each file contains the following variables: `arr`, `deltaVol`, `lob`, `myTime` with sizes narr$\times$14, narr$\times$1, nlob$\times$60 and nlob$\times$1, respectively. Always narr$>$nlob.

Each row in the `arr` matrix records every trade and change of the order book whenever it occurs, while the variable `lob` stands for the LOB and it was rebuilt by `arr`. In what follows, all operations involving r must be performed for all $r \in 1 : \text{narr}$. Furthermore, immediately after the beginning of the r-loop set `if arr(r,1)~=-1, l=l+1; end`, where l is initialized with zero and takes values in $\{1, 2, \cdots, \text{nlob}\}$. Index l corresponds to the lth order book event.

The variable `lob` contains actual bid and ask prices and their corresponding volumes up to ten levels; the first six columns give information for the best bid/ask quotes, the second six columns contain similar information for the second level of the book etc. The associated date and time is stored in the variable `myTime`. To reformat the date and time, and also show the milliseconds the MATLAB command `datestr(myTime,'dd-mm-yyyy HH:MM:SS.FFF')` is used.

The column `arr(:,1)` takes the values $\{-1, 1, 2, \cdots, 10\}$; if `arr(r,1)==-1` it indicates that at date and time `arr(r,2)` a trade occurred with price `arr(r,7)` and volume

arr(r,8). Whether arr(r,11)==1(2), then the trade was a buy (sell) market order. On the other hand, if arr(r,1)==j it shows that a change took place on the jth level in the book where $j \in 1:10$; if the bid (ask) size has changed, then the bid (ask) price arr(r,3) (arr(r,5)) and the bid (ask) volume arr(r,4) (arr(r,6)) will be real values, otherwise it will be NaN.

The variable deltaVol contains both integers and NaN values. If deltaVol(r,1)>0, it means that the volume at the arr(r,1)th price level has increased and the lth event is always a limit order; if arr(r,3:4) (arr(r,5:6)) are real numbers then it is a bid (ask) limit order. Similarly, when deltaVol(r,1)<0 then the volume on the arr(r,1)th level in the book has decreased and the lth event may be a cancel order, market order or both. Besides, NaN value is returned only when arr(r,1)==-1.

Here it is assumed that deltaVol(r,1)<0. In order to characterize the lth event we continue as follows: firstly, we examine whether the change occurred on the best level of the order book. If it was not, then it is always a cancellation; see the elements arr(r,3:4) and arr(r,5:6) to decide if it is a bid or an ask cancel order. Conversely, if arr(r,1)==1, initially we examine if arr(r-1,1)==-1. If it is, since a buy (sell) market order generally will execute at or near the best ask (bid) price, the lth line in arr should have real values on the ask (bid) side and NaN values on the bid (ask) side. However, if we observe the opposite, then the lth event is characterised as a bid (ask) cancel order. When arr(r-1,1)~=-1, then the lth event may be a trade.

Now we emphasize in the case where arr(r,1)==1, deltaVol(r,1)<0 and arr(r-1,1)~= -1. Set trades_nearby=MOindex+find(arr((MOindex+1):r,1)==-1), where MOindex is initialized with zero. This variable denotes the 'closest' trades up to r that has not yet been combined with a LOB. You may observe that some elements in arr(trades_near by,11) take the value 1 and others the value 2. If the bid (ask) price and volume in the rth line in the arr matrix are real numbers, then it is a sell (buy) market order; consequently, only the elements of trades_nearby such that arr(trades_nearby,11) equals 2 (1) are kept. If the vector trades_nearby is empty, then the lth event is characterized as a cancel order; otherwise set jtrades=find(arr(trades_nearby,7)==arr(r,3)(arr(r,5))). Again, if the resulting vector jtrades is empty the lth event is classified as a cancel order, otherwise set trades_nearby=trades_nearby(jtrades) and the lth event is characterized as a trade. The corresponding traded volume is tradedVol=sum(arr(trades_nearby,8)). Let mySize=-

`deltaVol(r)-tradedVol`. If `mySize>0`, then there is a cancelling order plus a trading. On the contrary, if `mySize<0`, then `tradedVol=-deltaVol(r)`. If the lth event has been characterized as a trade set `MOindex=trades_nearby(end,1)`. Note that it is possible to combine multiple trades with the same trade price with a LOB observation.

Bibliography

Carlos A Abanto-Valle and Dipak K Dey. State space mixed models for binary responses with scale mixture of normal distributions links. *Computational Statistics & Data Analysis*, 71: 274–287, 2014.

Carlos A Abanto-Valle, Dipak K Dey, and Xun Jiang. Binary state space mixed models with flexible link functions: a case study on deep brain stimulation on attention reaction time. *Statistics and Its Interface*, 8(2):187–194, 2015.

John Aitchison and Samuel D Silvey. The generalization of probit analysis to the case of multiple responses. *Biometrika*, 44(1/2):131–140, 1957.

Hirotugu Akaike. A new look at the statistical model identification. In *Selected Papers of Hirotugu Akaike*, pages 215–222. Springer, 1974.

MA Al-Osh and Aus A Alzaid. First-order integer-valued autoregressive (inar (1)) process. *Journal of Time Series Analysis*, 8(3):261–275, 1987.

AA Alzaid and M Al-Osh. An integer-valued pth-order autoregressive structure (inar (p)) process. *Journal of Applied Probability*, pages 314–324, 1990.

Abdulhamid A Alzaid and Maha A Omair. On the poisson difference distribution inference and applications. *Bulletin of the Malaysian Mathematical Sciences Society. Second Series*, 33(1): 17–45, 2010.

Abdulhamid A Alzaid and Maha A Omair. Poisson difference integer valued autoregressive model of order one. *Bulletin of the Malaysian Mathematical Sciences Society*, 37(2):465–485, 2014.

Stanislav Anatolyev and Nikolay Gospodinov. Modeling financial return dynamics via decomposition. *Journal of Business & Economic Statistics*, 28(2):232–245, 2010.

Jonas Andersson and Dimitris Karlis. A parametric time series model with covariates for integers in z. *Statistical Modelling*, 14(2):135–156, 2014.

Christophe Andrieu, Éric Moulines, et al. On the ergodicity properties of some adaptive mcmc algorithms. *The Annals of Applied Probability*, 16(3):1462–1505, 2006.

Yves Atchadé, Gersende Fort, et al. Limit theorems for some adaptive mcmc algorithms with subgeometric kernels. *Bernoulli*, 16(1):116–154, 2010.

István Barra, Agnieszka Borowska, and Siem Jan Koopman. Bayesian dynamic modeling of high-frequency integer price changes. *Journal of Financial Econometrics*, 16(3):384–424, 2018.

Luc Bauwens and Pierre Giot. The logarithmic acd model: an application to the bid-ask quote process of three nyse stocks. *Annales d'Economie et de Statistique*, pages 117–149, 2000.

Luc Bauwens and David Veredas. The stochastic conditional duration model: A latent factor model for the analysis of financial durations, forthcoming in journal of econometrics. 1999.

Luc Bauwens, Fausto Galli, and Pierre Giot. The moments of log-acd models. 2003.

Luc Bauwens, Pierre Giot, Joachim Grammig, and David Veredas. A comparison of financial duration models via density forecasts. *International Journal of Forecasting*, 20(4):589–609, 2004.

T. Bayes. An essay towards solving a problem in the doctrine of chances. *Philosophical Transactions of the Royal Society of London*, 53:370–418, 1763. Reprinted in *Biometrika* 45, 298–315, 1958. Reprinted in S. J. Press, *Bayesian Statistics*, 189–217, Wiley, New York, 1989.

Ernst R Berndt, Bronwyn H Hall, Robert E Hall, and Jerry A Hausman. Estimation and inference in nonlinear structural models. In *Annals of Economic and Social Measurement, Volume 3, number 4*, pages 653–665. NBER, 1974.

Alexandros Beskos, Gareth Roberts, Andrew Stuart, and Jochen Voss. Mcmc methods for diffusion bridges. *Stochastics and Dynamics*, 8(03):319–350, 2008.

Katarzyna Bien, Ingmar Nolte, and Winfried Pohlmeier. An inflated multivariate integer count hurdle model: an application to bid and ask quote dynamics. *Journal of Applied Econometrics*, 26(4):669–707, 2011.

Gail Blattenberger and Frank Lad. Separating the brier score into calibration and refinement components: A graphical exposition. *The American Statistician*, 39(1):26–32, 1985.

Szabolcs Blazsek and Adrian Licht. Dynamic conditional score models: a review of their applications. *Applied Economics*, 52(11):1181–1199, 2020.

Tim Bollerslev. Generalized autoregressive conditional heteroskedasticity. *Journal of econometrics*, 31(3):307–327, 1986.

Glenn W Brier. Verification of forecasts expressed in terms of probability. *Monthly weather review*, 78(1):1–3, 1950.

Christian T Brownlees, Fabrizio Cipollini, and Giampiero M Gallo. Multiplicative error models. *Available at SSRN 1852285*, 2011.

Patrizia Campagnoli, Giovanni Petris, and Sonia Petrone. *Dynamic Linear Models with R*. Springer, 2009.

Bradley P Carlin, Nicholas G Polson, and David S Stoffer. A monte carlo approach to nonnormal and nonlinear state-space modeling. *Journal of the American Statistical Association*, 87(418): 493–500, 1992.

James Carpenter, Peter Clifford, and Paul Fearnhead. Improved particle filter for nonlinear problems. *IEE Proceedings-Radar, Sonar and Navigation*, 146(1):2–7, 1999.

Leopoldo Catania, Roberto Di Mari, and Paolo Santucci de Magistris. Dynamic discrete mixtures for high frequency prices. *Available at SSRN 3349118*, 2019.

Siddhartha Chib and Bradley P Carlin. On mcmc sampling in hierarchical longitudinal models. *Statistics and Computing*, 9(1):17–26, 1999.

Nicolas Chopin. A sequential particle filter method for static models. *Biometrika*, 89(3):539–552, 2002.

Nicolas Chopin et al. Central limit theorem for sequential monte carlo methods and its application to bayesian inference. *The Annals of Statistics*, 32(6):2385–2411, 2004.

Simon L Cotter, Gareth O Roberts, Andrew M Stuart, and David White. Mcmc methods for functions: modifying old algorithms to make them faster. *Statistical Science*, pages 424–446, 2013.

David R Cox, Gudmundur Gudmundsson, Georg Lindgren, Lennart Bondesson, Erik Harsaae, Petter Laake, Katarina Juselius, and Steffen L Lauritzen. Statistical analysis of time series: Some recent developments [with discussion and reply]. *Scandinavian Journal of Statistics*, pages 93–115, 1981.

Drew Creal, Siem Jan Koopman, and André Lucas. A dynamic multivariate heavy-tailed model for time-varying volatilities and correlations. *Journal of Business & Economic Statistics*, 29 (4):552–563, 2011.

Drew Creal, Siem Jan Koopman, and André Lucas. Generalized autoregressive score models with applications. *Journal of Applied Econometrics*, 28(5):777–795, 2013.

Claudia Czado and Stephan Haug. An acd-ecogarch (1, 1) model. *Journal of Financial Econometrics*, 8(3):335–344, 2010.

Claudia Czado and Andreas Kolbe. Empirical study of intraday option price changes using extended count regression models. 2004.

Claudia Czado and Peter X-K Song. State space mixed models for longitudinal observations with binary and binomial responses. *Statistical Papers*, 49(4):691–714, 2008.

Claudia Czado, Gernot Müller, et al. Ordinal-and continuous-response stochastic volatility models for price changes: An empirical comparison. In *Statistical Modelling and Regression Structures*, pages 301–320. Springer, 2010.

Paul Damien, Petros Dellaportas, Nicholas G Polson, and David A Stephens. *Bayesian theory and applications*. OUP Oxford, 2013.

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

Stefanos Dimitrakopoulos and Mike Tsionas. Ordinal-response garch models for transaction data: A forecasting exercise. *International Journal of Forecasting*, 35(4):1273–1287, 2019.

Georges Dionne and Xiaozhou Zhou. The dynamics of ex-ante high-frequency liquidity: An empirical analysis. *Available at SSRN 2718637*, 2016.

Randal Douc and Olivier Cappé. Comparison of resampling schemes for particle filtering. In *ISPA 2005. Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis, 2005.*, pages 64–69. IEEE, 2005.

Arnaud Doucet and Adam M Johansen. A tutorial on particle filtering and smoothing: Fifteen years later. *Handbook of nonlinear filtering*, 12(656-704):3, 2009.

Arnaud Doucet, Simon Godsill, and Christophe Andrieu. On sequential monte carlo sampling methods for bayesian filtering. *Statistics and computing*, 10(3):197–208, 2000.

Arnaud Doucet, Nando De Freitas, and Neil Gordon. An introduction to sequential monte carlo methods. In *Sequential Monte Carlo methods in practice*, pages 3–14. Springer, 2001.

Feike C Drost and Bas J M Werker. Semiparametric duration models. *Journal of Business & Economic Statistics*, 22(1):40–50, 2004.

William Dunsmuir and Jieyi He. Marginal estimation of parameter driven binomial time series models. *Journal of Time Series Analysis*, 38(1):120–144, 2017.

William TM Dunsmuir, David J Scott, et al. The glarma package for observation-driven time series regression of counts. *Journal of Statistical Software*, 67(7):1–36, 2015.

WTM Dunsmuir and JY He. Detecting serial dependence in binomial time series ii: Observation driven models. *arXiv preprint arXiv:1606.00984*, 2016.

James Durbin and Siem Jan Koopman. *Time series analysis by state space methods*. Oxford university press, 2012.

Robert Engle. New frontiers for arch models. *Journal of Applied Econometrics*, 17(5):425–446, 2002.

Robert Engle and Giampiero Gallo. A model for intra-daily volatility with multiple indicators. *Unpublished manuscript*, 2002.

Robert F Engle. The econometrics of ultra-high-frequency data. *Econometrica*, 68(1):1–22, 2000.

Robert F Engle and Jeffrey R Russell. Autoregressive conditional duration: a new model for irregularly spaced transaction data. *Econometrica*, pages 1127–1162, 1998.

Robert F Engle and Jeffrey R Russell. Analysis of high frequency financial data. *Handbook of financial econometrics*, 2004.

Ludwig Fahrmeir. Posterior mode estimation by extended kalman filtering for multivariate dynamic generalized linear models. *Journal of the American Statistical Association*, 87(418): 501–509, 1992.

Augusto Fasano, Giovanni Rebaudo, Daniele Durante, and Sonia Petrone. A closed-form filter for binary time series. *arXiv preprint arXiv:1902.06994*, 2019.

Paul Fearnhead. Mcmc for state-space models. 2011.

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

Sylvia Frühwirth-Schnatter. Efficient bayesian parameter estimation for state space models based on reparameterizations. *State Space and Unobserved Component Models: Theory and Applications*, pages 123–151, 2004.

Alan E Gelfand, Sujit K Sahu, and Bradley P Carlin. Efficient parametrisations for normal linear mixed models. *Biometrika*, 82(3):479–488, 1995.

Andrew Gelman, Gareth O Roberts, Walter R Gilks, et al. Efficient metropolis jumping rules. *Bayesian statistics*, 5(599-608):42, 1996.

Stuart Geman and Donald Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on pattern analysis and machine intelligence*, (6): 721–741, 1984.

Walter R Gilks and Carlo Berzuini. Following a moving target—monte carlo inference for dynamic bayesian models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(1):127–146, 2001.

Walter R Gilks, Sylvia Richardson, and David Spiegelhalter. *Markov chain Monte Carlo in practice*. Chapman and Hall/CRC, 1995.

Geof H Givens and Jennifer A Hoeting. *Computational statistics*, volume 703. John Wiley & Sons, 2012.

Geoffrey Grimmett, David Stirzaker, et al. *Probability and random processes*. Oxford university press, 2001.

David Gunawan, Khue-Dung Dang, Matias Quiroz, Robert Kohn, and Minh-Ngoc Tran. Subsampling sequential monte carlo for static bayesian models. *arXiv preprint arXiv:1805.03317*, 2018.

Heikki Haario, Eero Saksman, Johanna Tamminen, et al. An adaptive metropolis algorithm. *Bernoulli*, 7(2):223–242, 2001.

M Hajmeer and I Basheer. Comparison of logistic regression and neural network-based classifiers for bacterial growth. *Food Microbiology*, 20(1):43–55, 2003.

Daniel B Hall. Zero-inflated poisson and binomial regression with random effects: a case study. *Biometrics*, 56(4):1030–1039, 2000.

Andrew C Harvey. *Dynamic models for volatility and heavy tails: with applications to financial and economic time series*, volume 52. Cambridge University Press, 2013.

W Keith Hastings. Monte carlo sampling methods using markov chains and their applications. 1970.

Jerry A Hausman, Andrew W Lo, and A Craig MacKinlay. An ordered probit analysis of transaction stock prices. *Journal of financial economics*, 31(3):319–379, 1992.

Nikolaus Hautsch, Peter Malec, and Melanie Schienle. Capturing the zero: a new class of zero-augmented distributions and multiplicative error processes. *Journal of Financial Econometrics*, 12(1):89–121, 2014.

Joseph Oscar Irwin. The frequency distribution of the difference between two independent variates following the same poisson distribution. *Journal of the Royal Statistical Society*, 100 (3):415–416, 1937.

Andrew H Jazwinski. *Stochastic processes and filtering theory*. Courier Corporation, 2007.

Simon J Julier and Jeffrey K Uhlmann. Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, 92(3):401–422, 2004.

Simon J Julier, Jeffrey K Uhlmann, and Hugh F Durrant-Whyte. A new approach for filtering nonlinear systems. In *Proceedings of 1995 American Control Conference-ACC'95*, volume 3, pages 1628–1632. IEEE, 1995.

Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1):35–45, 1960.

Nikolas Kantas, Arnaud Doucet, Sumeetpal S Singh, Jan Maciejowski, Nicolas Chopin, et al. On particle methods for parameter estimation in state-space models. *Statistical science*, 30 (3):328–351, 2015.

Dimitris Karlis and Ioannis Ntzoufras. Bayesian analysis of the differences of count data. *Statistics in medicine*, 25(11):1885–1905, 2006.

Robert E Kass. Bayes factors in practice. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 42(5):551–560, 1993.

Robert E Kass and Adrian E Raftery. Bayes factors. *Journal of the american statistical association*, 90(430):773–795, 1995.

Robert E Kass, Bradley P Carlin, Andrew Gelman, and Radford M Neal. Markov chain monte carlo in practice: a roundtable discussion. *The American Statistician*, 52(2):93–100, 1998.

Gregor Kastner and Sylvia Frühwirth-Schnatter. Ancillarity-sufficiency interweaving strategy (asis) for boosting mcmc estimation of stochastic volatility models. *Computational Statistics & Data Analysis*, 76:408–423, 2014.

Heikki Kauppi and Pentti Saikkonen. Predicting us recessions with dynamic binary response models. *The Review of Economics and Statistics*, 90(4):777–791, 2008.

Alexander R Kent. High frequency price decomposition and trade arrivals. 2015.

Sangjoon Kim, Neil Shephard, and Siddhartha Chib. Stochastic volatility: likelihood inference and comparison with arch models. *The review of economic studies*, 65(3):361–393, 1998.

Augustine Kong, Jun S Liu, and Wing Hung Wong. Sequential imputations and bayesian missing data problems. *Journal of the American statistical association*, 89(425):278–288, 1994.

Siem Jan Koopman, DD Creal, André Lucas, et al. A general framework for observation driven time-varying parameter models. 2008.

Siem Jan Koopman, Rutger Lit, and André Lucas. The dynamic skellam model with applications. 2014.

Siem Jan Koopman, Andre Lucas, and Marcel Scharth. Predicting time-varying parameters with parameter-driven and observation-driven models. *Review of Economics and Statistics*, 98(1):97–110, 2016.

Siem Jan Koopman, Rutger Lit, and André Lucas. Intraday stochastic volatility in discrete price changes: The dynamic skellam model. *Journal of the American Statistical Association*, 112 (520):1490–1503, 2017.

Siem Jan Koopman, Rutger Lit, André Lucas, and Anne Opschoor. Dynamic discrete copula models for high-frequency stock price changes. *Journal of Applied Econometrics*, 33(7): 966–985, 2018.

Roman Liesenfeld, Ingmar Nolte, and Winfried Pohlmeier. Modelling financial transaction price movements: a dynamic integer count data model. *Empirical Economics*, 30(4):795–825, 2006.

Simone Manganelli. *Volume, volatility and the price impact of trades*. PhD thesis, PhD Dissertation, UCSD, 2000.

P McCullagh and JA Nelder. 1989. *Generalized linear models*, 37, 1983.

Robert E McCulloch and Ruey S Tsay. Nonlinearity in high-frequency financial data and hierarchical models. *Studies in Nonlinear Dynamics & Econometrics*, 5(1), 2001.

Nicholas Metropolis and Stanislaw Ulam. The monte carlo method. *Journal of the American statistical association*, 44(247):335–341, 1949.

Nicholas Metropolis, Arianna W Rosenbluth, Marshall N Rosenbluth, Augusta H Teller, and Edward Teller. Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6):1087–1092, 1953.

John Mullahy. Specification and testing of some modified count data models. *Journal of econometrics*, 33(3):341–365, 1986.

Gernot Müller and Claudia Czado. An autoregressive ordered probit model with application to high-frequency financial data. *Journal of Computational and Graphical Statistics*, 14(2): 320–338, 2005.

Gernot Müller and Claudia Czado. Stochastic volatility models for ordinal-valued time series with application to finance. *Statistical Modelling*, 9(1):69–95, 2009.

Radford M Neal. Annealed importance sampling. *Statistics and computing*, 11(2):125–139, 2001.

Henri Nyberg. Forecasting the direction of the us stock market with dynamic binary probit models. *International Journal of Forecasting*, 27(2):561–578, 2011.

Omiros Papaspiliopoulos. *Non-centered parametrizations for hierarchical models and data augmentation*. PhD thesis, Dept. Mathematics and Statistics, Lancaster Univ., 2003.

Omiros Papaspiliopoulos, Gareth O Roberts, and Martin Sköld. Non-centered parameterizations for hierarchical models and data augmentation (with discussion). In *Bayesian Statistics 7 (J. M. Bernardo, M. J. Bayarri, J. O. Berger, A. P. Dawid, D. Heckerman, A. F. M. Smith and M. , eds.)*, pages 307–326. Oxford University Press, New York, 2003.

Michael K Pitt and Neil Shephard. Filtering via simulation: Auxiliary particle filters. *Journal of the American statistical association*, 94(446):590–599, 1999.

Michael K Pitt and Neil Shephard. Auxiliary variable based particle filters. In *Sequential Monte Carlo methods in practice*, pages 273–293. Springer, 2001.

Harri Pönkä. Predicting the direction of us stock markets using industry returns. *Empirical Economics*, 52(4):1451–1480, 2017.

Greg Ridgeway and David Madigan. A sequential monte carlo method for bayesian analysis of massive datasets. *Data Mining and Knowledge Discovery*, 7(3):301–319, 2003.

Christian Robert and George Casella. *Monte Carlo statistical methods*. Springer Science & Business Media, 2013.

Gareth O Roberts and Jeffrey S Rosenthal. Coupling and ergodicity of adaptive markov chain monte carlo algorithms. *Journal of applied probability*, 44(2):458–475, 2007.

Gareth O Roberts and Jeffrey S Rosenthal. Examples of adaptive mcmc. *Journal of Computational and Graphical Statistics*, 18(2):349–367, 2009.

Gareth O Roberts, Andrew Gelman, Walter R Gilks, et al. Weak convergence and optimal scaling of random walk metropolis algorithms. *The annals of applied probability*, 7(1):110–120, 1997.

Gareth O Roberts, Jeffrey S Rosenthal, et al. Optimal scaling for various metropolis-hastings algorithms. *Statistical science*, 16(4):351–367, 2001.

Jeffrey Russell and Robert Engle. Econometric analysis of discrete-valued irregularly-spaced financial transactions data using a new autoregressive conditional multinomial model. 1998.

Jeffrey R Russell and Robert F Engle. A discrete-state continuous-time model of financial transactions prices and times: The autoregressive conditional multinomial–autoregressive conditional duration model. *Journal of Business & Economic Statistics*, 23(2):166–180, 2005.

JR Russell and RF Engle. Analysis of high (frequency data. in ait (sahalia, y. and lp hansen,(eds) handbook of financial econo (metrics, 2010.

Tina Hviid Rydberg and Neil Shephard. Dynamics of trade-by-trade price movements: decomposition and models. 1998a. Working Paper, Presented at Workshop on Econometrics and Finance, Issac Newton Institute, Cambridge University.

Tina Hviid Rydberg and Neil Shephard. Modelling trade-by-trade price movements of multiple assets using multivariate compound poisson processes. *Available at SSRN 182248*, 1999.

Tina Hviid Rydberg and Neil Shephard. A modelling framework for the prices and times of trades made on the new york stock exchange. *Nonlinear and nonstationary signal processing*, pages 217–246, 2000.

Tina Hviid Rydberg and Neil Shephard. Dynamics of trade-by-trade price movements: decomposition and models. *Journal of Financial Econometrics*, 1(1):2–25, 2003.

Golnaz Shahtahmassebi. Bayesian modelling of ultra high-frequency financial data. 2011.

Golnaz Shahtahmassebi and Rana Moyeed. Bayesian modelling of integer data using the generalised poisson difference distribution. *International Journal of Statistics and Probability*, 3(1):35, 2014.

N Shephard. Generalized linear autoregressions. nuffield college, 1995.

Neil Shephard. *Stochastic volatility: selected readings*. Oxford University Press on Demand, 2005.

Neil Shephard and Sangjoon Kim. [bayesian analysis of stochastic volatility models]: Comment. *Journal of Business & Economic Statistics*, 12(4):406–410, 1994.

John G Skellam. The frequency distribution of the difference between two poisson variates belonging to different populations. *Journal of the Royal Statistical Society. Series A (General)*, 109(Pt 3):296–296, 1946.

M Sklar. Fonctions de repartition an dimensions et leurs marges. *Publ. inst. statist. univ. Paris*, 8:229–231, 1959.

Peter Xue-Kun Song. Monte carlo kalman filter and smoothing for multivariate discrete state space models. *Canadian Journal of Statistics*, 28(3):641–652, 2000.

Anthony S Tay, Christopher Ting, Yiu Kuen Tse, and Mitchell Warachka. Transaction-data analysis of marked durations and their implications for market microstructure. 2004.

Luke Tierney. Markov chains for exploring posterior distributions. *the Annals of Statistics*, pages 1701–1728, 1994.

M Titsias. Contribution to the discussion of the paper by girolami and calderhead. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(2):197–199, 2011.

Michalis K Titsias and Omiros Papaspiliopoulos. Auxiliary gradient-based sampling algorithms. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 80(4): 749–767, 2018.

Ruey S Tsay. *Analysis of financial time series*, volume 543. John wiley & sons, 2005.

Rudolph Van Der Merwe, Arnaud Doucet, Nando De Freitas, and Eric A Wan. The unscented particle filter. In *Advances in neural information processing systems*, pages 584–590, 2001.

John von Neumann. Various techniques used in connection with random digits. *John von Neumann, Collected Works*, 5:768–770, 1963.

Rongning Wu and Yunwei Cui. A parameter-driven logit regression model for binary time series. *Journal of Time Series Analysis*, 35(5):462–477, 2014.

Joey Wenling Yang and Jerry Parwada. Predicting stock price movements: an ordered probit analysis on the australian securities exchange. *Quantitative Finance*, 12(5):791–804, 2012.

Yaming Yu and Xiao-Li Meng. To center or not to center: That is not the question—an ancillarity–sufficiency interweaving strategy (asis) for boosting mcmc efficiency. *Journal of Computational and Graphical Statistics*, 20(3):531–570, 2011.

Michael Yuanjie Zhang, Jeffrey R Russell, and Ruey S Tsay. A nonlinear autoregressive conditional duration model with applications to financial transaction data. *Journal of Econometrics*, 104(1):179–207, 2001.

Walter Zucchini, Iain L MacDonald, and Roland Langrock. *Hidden Markov models for time series: an introduction using R*. Chapman and Hall/CRC, 2017.