Fundamental and technical analysis of financial markets

PhD Thesis

Balázs Torma

Supervisor: László Gerencsér, DSc

Faculty of Informatics,
Eötvös Loránd University
(ELTE IK)

Doctoral School in Informatics,
Foundations and Methods in Informatics PhD Program,
Chairman: Prof. János Demetrovics, Member of MTA

Computer and Automation Research Institute,
Hungarian Academy of Sciences
(MTA SZTAKI)

Budapest, Hungary, 2010

Declaration

Herewith I confirm that all of the research described in this dissertation is my own original work and expressed in my own words. Any use made within it of works of other authors in any form, e.g., ideas, figures, text, tables, are properly indicated through the application of citations and references. I also declare that no part of the dissertation has been submitted for any other degree - either from the Eötvös Loránd University or another institution.

Balázs Torma
Budapest, May 2010

Contents

Acknowledgment

First of all, I would like to express my thanks to my supervisor, Prof. László Gerencsér, for introducing me into the culture of mathematics and for his constant trust and guidance during my studies.

I am very thankful to Prof. Ferenc Friedler for helping me generously to switch back from industry into academics. In general, I feel deeply indebted to the whole Hungarian academic community for the inspiring, open and supportive educational and research environment. In particular, I am grateful to my host institute, the Computer and Automation Research Institute (MTA SZTAKI), and I thank my colleagues in Stochastic Systems Research Group, namely, Dr. Vilmos Prokaj and Dr. Miklós Rásonyi, for several fruitful and motivating discussions and numerous helpful comments on my work. I am very grateful to Dr. Boglárka G.-Tóth and Dr. Vilmos Prokaj for their collaboration in research.

I am also grateful for the PhD scholarship that I received for one year from the Faculty of Informatics (IK) of the Eötvös Loránd University (ELTE) and for the financial support of the Difilton Arc Kft. lead by Sándor Manno. I greatly appreciate the computational resources provided by the Data Mining and Web Search Group at MTA SZTAKI headed by Dr. András Benczúr.

Last but not least, I am very thankful for the continuous and vital support and care of my wife, Emese Ágnes Szűcs.

Chapter 1

Introduction

"Everything should be made as simple as possible, but not simpler."

Albert Einstein

Understanding the working mechanisms of financial markets is crucial to ensure the stability of domestic and global economies. Market participants vary highly with respect to their financial motivation, capital endowment, and their trading activity is often subject to different regulation. Agent-based computational finance (ACF) models provide a means to account for this heterogeneity, inasmuch they incorporate detailed descriptions of agent behaviors and economic processes.

An important application of ACF models is to investigate unobservable economic processes based on observed data. The difficulties in applying ACF models for inference about economic variables are due to their complexity. In order to make inference feasible, analytically tractable agent-based models have been developed recently. The price to pay for tractability is that these models are very abstract, include highly aggregated variables or impose unrealistic conditions - in short, they neglect important characteristics of markets. Consequently, the economic interpretability of empirical results that we get by fitting these simple models to data is limited.

In this thesis we propose a detailed ACF model and develop statistical algorithms based on analytically tractable technical (econometric) models. Furthermore, we apply technical models for analysing some of the key economic factors entering the ACF model. The main mathematical tool of the statistical analysis is model identification by Maximum Likelihood Estimation. A novel local optimization algorithm is developed for efficient off-line identification. We also develop real-time recursive identification algorithms and change-point detection algorithms for some important technical models.

The thesis is organized as follows. The objective of *Chapter 2* is to demonstrate the application of technical models of stock prices on detecting abrupt changes in the dynamics of real price processes. Moreover, we examine what latent economic factors entering financial markets can trigger the change. To elaborate this approach we first propose a fundamental model that extends well-established concepts of agent-based computational finance with a novel element for information arrival processes. Announcements and news about a company trigger analysts to pay more attention to that company, and change its valuation. This, in turn, results in more analysts' announcements about the given company. We model these feedback effects of market news by a discrete time Hawkes process. We show by numerical experiments that in contrast to classical ACF models, stylized facts emerge even by constant market fractions of chartists and fundamentalists as a consequence of chartists trading. We further validate the ACF model using technical models. In particular, a qualitative relationship between the market structure and the best-fitting GARCH(1,1) (General Autoregressive Heteroscedastisity) model is established, which motivates us to apply GARCH(1,1) for indirect statistical inference on the market structure. The use of GARCH models for detecting changes in the market-structure is justified by a well-known principle, stating that change detection is feasible using misspecified models, see discussion in Section 2.1. A real-time change detection method for GARCH processes is presented based on the MDL (Minimum Description Length) approach to modelling. Having the above mentioned relationship between the GARCH(1,1) and the ACF models, we provide an economic interpretation of the GARCH-based change alarms. The performance of the proposed algorithm is evaluated on simulated data. Change-alarms based on real data are reported.

In order to examine the relationship between the market structure and GARCH(1,1) models, we fit a GARCH model on the time series generated by the ACF model by quasi Maximum Likelihood Estimation, first in an off-line manner. We get a non-linear optimization problem, which is well-known to be ill-conditioned in the GARCH case. Motivated by this problem, in *Chapter 3* we develop a new hybrid optimization method which combines the advantages of descent methods and cutting plane approaches. The new method gets fast to near-optimal region by using cutting planes and preserves the good convergence properties of descent methods near the optimum. The method is tested on convex functions, least squares problems and on GARCH parameter estimation by comparing its performance to well-known methods. Numerical experiments show that the proposed method is very efficient on all the examined problem types and performs in average much better than the benchmark methods. We also present a technique with which the dimen-

sion of the GARCH fitting problem can be reduced. Overall, we get a GARCH estimation algorithm that is an order of magnitude faster than the related MATLAB procedure.

We can model self-exciting effects arising on financial markets with Hawkes processes. A Hawkes process is a point process whose intensity is defined via a feedback mechanism where the input is the past of the point process itself. In the standard Hawkes process, when an event occurs, e.g. some news appears on the market, the intensity increases by a constant amount. After the event the intensity is reverting to a minimum value exponentially fast. Modelling news arrival processes can enhance the prediction of price volatility, since, as shown by several empirical studies, the volatility is positively correlated with the intensity of news arrival. Besides modelling information arrival processes, Hawkes processes can be applied to capture the self-exciting property of credit default processes: insolvency of a given company can increase the insolvency probability of another company. In *Chapter 4* we propose Hawkes processes in which the feedback path is defined by a finite dimensional linear system. This model class allows for representing multiple event sources, e.g. several analysts. We propose algorithms for the simulation and real-time estimation of this type of Hawkes processes. Moreover, we investigate the Fisher information matrix of the estimation problem numerically and analytically. In particular, we show that some parts of the diagonal of the asymptotic Fisher information matrix in case of the one-dimensional feedback go to infinity, other parts of the diagonal go to zero as the parameters approach the boundary of the stability domain. As a first step we calculate the limit distribution of the appropriately rescaled intensity process.

Since financial prices are usually quantized data, we may get spurious results from estimation methods neglecting quantization effects. In *Chapter 5* we demonstrate this phenomenon on a toy example, namely, on linear regression. We develop a real-time method for estimating the parameters of a linear regression from quantized observations, when the regressor is finite-valued. The algorithm applies Expectation Maximization (EM) and Markov Chain Monte Carlo techniques. An example application is to regress high-frequency price data on order-book-level supply and demand quantities.

Chapter 2

Change detection in fundamental models using technical models

2.1 Context and motivation

There are basically two competing approaches in modelling stock prices: *technical modelling and fundamental modelling.* Technical or econometric models capture statistical phenomena observed directly on stock prices, while fundamental models consist of mathematical descriptions of behaviors of economic agents affecting the stock price. Fundamental models provide a more detailed description of the price generating mechanism than technical models. On the other hand, technical models are mathematically more tractable. An excellent survey of technical models is given in Taylor (2007), while a similarly outstanding survey for fundamental models is provided in LeBaron (2000); Chen et al. (2009). These works coin the corresponding terminologies as financial econometrics and agent-based computational finance (ACF), respectively.

A drawback of complex fundamental models is that they are not tractable by standard mathematical techniques developed in the literature for system-identification, see Ljung and Söderström (1983); Benveniste et al. (1990); Spall (2003). The reason for this is the exogenous noise driving the price process can not be recovered even if the system dynamics would be known. This situation is analogous to linear stochastic systems with a large number of random sources. In the latter case Kalman filtering leads to an innovation form that is already invertible. In the case of an ACF model we would require non-linear filtering leading to an infinite dimensional problem, which is beyond the scope of Ljung and Söderström (1983); Benveniste et al. (1990); Spall (2003). In short, the identification of

individual key economic factors entering the model is not possible by these standard tools, and we are unaware of any alternative approach of similar sophistication and generality. See also the discussion in Winker et al. (2007) on this.

In spite of analytical intractability, statistical inference on complex fundamental models is possible within limited scope. A basic problem that is tractable in spite of gross uncertainties is the *detection of changes* in the market dynamics. In fact, change detection methods are known to be robust against modelling errors, in other words, detecting changes is possible using mis-specified models, see e.g. Basseville and Nikiforov (1993); Campillo et al. (2000). Obviously, the quality of the detection method depends on the quality of the model class used in modelling the data. A natural model class to be used for detecting changes in an ACF model, or in a real price process is a class of econometric models.

In this chapter we demonstrate the feasibility of the above program. The fundamental model to be used is a significant modification of the fundamental models proposed in the pioneering works of Kirman (1995), Lux (1998) and Brock and Hommes (1998), see Hommes (2006) for a survey including these models. In these ACF models the market is composed of two types of agents: chartists and fundamentalists. Chartists predict future stock prices by extrapolating current trend, while the belief of fundamentalists about future stock prices is affected by the information they receive on the company at hand. Assuming a fixed behavior pattern for individual agents of a given type, modulo random choices, the *market structure* is then defined as the distribution of relative wealth among the two groups of agents. In Kirman (1995) and Lux (1998) market fractions of chartists and fundamentalists evolve endogenously driven by past profitability of these two strategies. Thus, these models suggest that future expectations depend only on profits realized in the past. However, reduction of trading on fundamentals is possible due to decreased predictability, triggered by the burst of an economic bubble, say. Or, increasing trading on fundamentals can be triggered by an announcement about company acquisition, say. Hence in our model we allow future expectations to depend on exogenous effects as well and make the market fractions of chartists and fundamentalist a model parameter.

In general, exogenous uncertainty of fundamentals is caused by market news arriving at random times. Since the reaction to news generates further news, we model the information arrival process as a self-exciting point process, more specifically, a discrete time version of well-known *Hawkes' process*, see Hawkes (1971a).

We have verified by extensive numerical experiments that a number of stylized facts known of real price processes, such as volatility clustering and fat tails of return distri-

butions are reproduced by our model. In the models of Kirman (1995) and Lux (1998), volatility clustering and fat tails of returns distributions are attributed to varying market fractions in time, as in periods dominated by chartists the volatility of returns is higher than in periods when the ratio of fundamentalists is high. In contrast, in our model volatility clustering and fat tails emerge even by constant market fractions, as a consequence of chartist trading and the dynamical properties of the news arrival process.

For the statistical analysis of our ACF model we use the most widely known technical model, the class of Generalized Autoregressive Conditional Heteroscedasticity, or *GARCH models* developed in Engle (1982); Bollerslev (1986). Popular GARCH variants are the Markov switching GARCH (MS-GARCH), see Bauwens et al. (2007) or threshold GARCH (TGARCH), see Zakoian (1994), among other things. In our analysis we fit a GARCH(1,1) model off-line using quasi maximum likelihood to a data-set of 10000 generated by the ACF model. We established a *monotonic relationship* between the market fraction and the GARCH coefficients.

An earlier similar attempt for explaining GARCH parameters is due to Diks and van der Weide. They develop an agent-based model in which an a-synchronous update of agent's beliefs yield a GARCH dynamics of returns, see Diks and van der Weide (2005). This so called Continuous Belief System (CBS) is analytically tractable but still very abstract, see discussion in Diks and van der Weide (2005) in section 4. An example of major simplifications which is not discussed in the paper is that there is no distinction between chartists and fundamentalists. We follow a different route, inasmuch we include much more details that is known on the dynamics of stock markets in our model, and use a technical model for its analysis.

Due to the established relationship between the market fraction and the GARCH coefficients, GARCH models become a natural candidate for change detection in the market structure. In view of real-time needs we present a real-time change detection method for GARCH processes to detect abrupt changes of its dynamics, along the lines of Gerencsér and Baikovicius (1991, 1992) using a Minimum Description Length, or *MDL approach*, see Rissanen (1989). This requires a reliable implementation and testing of a recently developed recursive quasi-maximum-likelihood method, see Gerencsér et al. (2010). See also Aknouche and Guerbyenne (2006) for a recursive least squares based identification method for GARCH models. See Berkes et al. (2004) for a change detection algorithm using quasi-likelihood scores.

In order to outline the feasibility of direct statistical inference using analytically tractable agent-based models, we *review* some of the attempts in recent works addressing identifica-

tion of some models of this class, along with describing major model assumptions. Common to these models is that they implement a low degree of agent heterogeneity (as opposed to our ACF model): heterogeneity among agents is reflected only by two trader types, the concept of agents with different parameterizations within a given type is not included. An additional simplification is that the various agent types are characterized by a common risk aversion factor. All these models rely on the concepts presented in the seminal paper of Brock and Hommes, see Brock and Hommes (1998), except for the model in Alfarano et al. (2005). In Alfarano et al. (2005), a simplified version of the Kirman model is developed and a closed form solution for the distribution of returns is derived. The estimated parameters in the model capture the tendency of traders to switch between the two groups, fundamentalists and noise traders (noise traders are agents whose strategies do not have any systematic component). In Amilon (2008) the model of De Grauwe and Grimaldi (2004) is fitted to data and, among others, the dynamics of the market fraction of chartists and fundamentalists is estimated. Amilo reports that the fit is generally quite poor. In addition, the estimated market fraction in Figure 3(b) in Amilon (2008) indicates that for long periods of time, for approximately a year, only chartists are active on the market, a finding which is at least questionable. The model presented in the following two papers, see Boswijk et al. (2007), Kozhan and Salmon (2009), impose the additional condition that agents have homogeneous expectations about the conditional variance of future returns. Both models also assume that the fundamental value is known to the traders, thus, it is approximated via some observable economic variables (proxies) in the estimation procedure. Boswijk et al. (2007) estimate the mean reversion factor of fundamentalists, the trend extrapolation factor of chartists and market fractions of the two agent types. They found a significant presence of both agent types on the S&P500 stock market. The central question in Kozhan and Salmon (2009) is whether or not agents in the foreign exchange market are uncertainty averse. Chartists are found to be uncertainty averse but fundamentalist are found to be uncertainty neutral, a result which contradicts several empirical studies on uncertainty aversion, see e.g. Mangelsdorff and Weber (1994); Wakker (2001). An interesting result in Kozhan and Salmon (2009) is the statistical evidence showing that the model is capable of predicting the sign of one-step-ahead returns.

The chapter is organized as follows. First the key concepts used in our ACF model are described. The model is validated by showing that it reproduces a number of important stylized facts exhibited by real price processes. In Section 2.4 we present experimental results establishing a qualitative relationships between the market structure and the parameters of the GARCH model that fits best the data. In Section 2.5 we describe a real-time

change-point detection algorithm for GARCH models, its performance is evaluated in Section 2.6. In Section 2.7 we discuss a possible economic interpretation of the change-alarms, and comment on the results of the change detection algorithm when applied on real market data in Section 2.8. We conclude with a brief summary of results.

2.2 The multi-agent stock market model

In our ACF model agents are assumed to trade the stocks of only one company. In each trading period t, say day or hour each agent applies his belief $\hat{p}_t$ to determine his transaction request (d_t, b_t), the requested quantity and the corresponding limit price, where $d_t < 0$ if the agent wants to sell. Then, the stock market exchange matches the transaction requests and determines the new stock price p_t. We apply a pricing algorithm that is commonly used in the closing trading session at real exchanges, such as the NASDAQ, see www.nasdaqtrader.com: for a given price p calculate the total number of transactions that would take place at this price, and choose the price at which trading volume is maximized, see Gerencsér and Mátyás (2007b) for a formal description of the algorithm.

In any period t, agents develop their belief about next period's market price. The belief $\hat{p}_t^i$ of agent i can be interpreted as a one step ahead predictor. E.g., $\hat{p}_t$ could be obtained by using a regression line fitted to past price data. In the following we outline the applied order placing rules, see Appendix A for details. Having $\hat{p}_t$, the agent proceeds with calculating the actual orders (d_t, b_t) in a way that his wealth increases under the assumption of exact prediction $(p_{t+1} = \hat{p}_t)$. It is easy to see that this can be achieved by placing a limit buy order one cent below $\hat{p}$ and placing simultaneously a limit sell order one cent above $\hat{p}$. Then, if $\hat{p} > p$, only the buy order executes, and if $\hat{p} < p$, only the sell order executes. Thus, the transaction request of agent i trading one share consists of orders

$$\begin{aligned} (d, b)_t^{i,1} &= \left(1, \hat{p}_t^i - 0.01\right), \\ (d, b)_t^{i,2} &= \left(-1, \hat{p}_t^i + 0.01\right). \end{aligned}$$

Numerical simulations have shown that the applied ordering rules eventually result in a market price which can be well approximated by the average of the predictions:

$$p_t \approx \frac{1}{N} \sum_{i=1}^{N} \hat{p}_t^i.$$

This finding is in line with considerations in Diks and van der Weide (2005), see eq. (5) in that paper.

A schematic view of the agent model is depicted in Figure 2.1. The superscript C stands for chartists, F for fundamentalists and E for the environment, see further notations below. Next we describe the agent types and their predictor models.

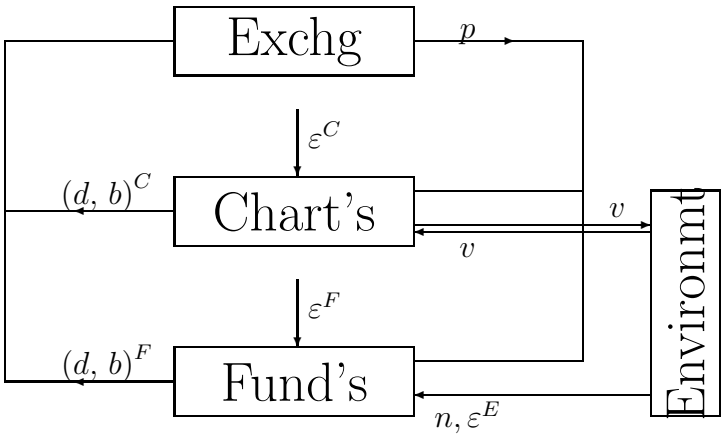

Figure 2.1: The market model

Chartists. Chartists or technical traders determine their belief based on past market prices, without incorporating any economic information about the company. According to the survey of practitioners' trading strategies Kaufman (1998), we can identify two behavior groups. Trend followers (or momentum traders) believe that market prices will continue to move in the direction of the current price dynamics. Traders typically detect the trend by comparing long-term and short-term average market prices, see Chiarella et al. (2006). As opposed to trend following, the mean reverting (or contrarian) behavior is based on the opinion that prices oscillate around some long-term average price.

Let us now formalize the behavior model of chartists, which leans on the concepts summarized in Chen et al. (2009). The number of chartist is denoted by N^C. Let $[\mathcal{E}_q(x)]_t$ denote the exponential averaging operator defined as

$$[\mathcal{E}_q(x)]_0 = x_0, \tag{2.1}$$

$$[\mathcal{E}_q(x)]_t = (1-q)\,[\mathcal{E}_q(x)]_{t-1} + qx_t, \quad t > 0 \tag{2.2}$$

for the process $x_t \in \mathbb{R}$ with forgetting factor $0 < q < 1$.

Define the short-term and long-term average price calculated by chartist i with forgetting factor $0 < \lambda^i < 0.5$ and $2\lambda^i$ as

$$\bar{p}_t^i = \left[\mathcal{E}_{\lambda^i}(p)\right]_t, \quad \bar{\bar{p}}_t^i = \left[\mathcal{E}_{2\lambda^i}(p)\right]_t.$$

Each chartist uses a different λ to filter p_t, thus, we say that λ is an agent specific parameter.

Chartist i defines the trend of the price process as

$$\bar{r}_t^i = \log\left(\frac{\bar{p}_t^i}{\bar{\bar{p}}_t^i}\right). \tag{2.3}$$

The processes $\bar{p}_t$ and $\bar{\bar{p}}_t$ tracks p_t with a delay, the tracking lag is controlled by the forgetting factor. If λ is small, $\bar{p}_t$ and $\bar{\bar{p}}_t$ track p_t slowly, in which case the speed of change of the dynamics of $\bar{r}_t$ is small.

Based on the heuristics above, trend followers denoted by superscript T and mean reverters denoted by superscript M determine their belief by extrapolating past trend as

$$\hat{p}_t^{T,i} = p_{t-1}\exp\left(\alpha^i \bar{r}_{t-1}^i\right), \tag{2.4}$$

$$\hat{p}_t^{M,i} = p_{t-1}\exp\left(-\alpha^i \bar{r}_{t-1}^i\right), \tag{2.5}$$

with the agent specific parameter $\alpha^i > 0$ expressing the degree of trend extrapolation.

Adaptive beliefs. According to fundamental findings of behavioral finance, real traders often do not feel confident with their currently executed trading strategy, they usually switch strategies from time to time caused by behavioral uncertainty. A strategy switch can be triggered by some information they receive from the media or other market participants. There are in fact ranking web sites, e.g. www.collective2.com, where traders publish some basic information about their strategies. They see for example the profitability of each other's strategy and also some general information about its working mechanism. Thus, we assume that chartists can collect information about the average profitability of trend following and mean reverting, denoted by $\bar{v}_t^T$ and $\bar{v}_t^M$, respectively. The aggregated fitness measures are determined in the environment based on individual profits v_t^T and v_t^M chartist achieve with either behavior types:

$$\bar{v}_t^T = \frac{1}{N^C}\sum_{i=1}^{N_C} v_t^{T,i}, \quad \bar{v}_t^M = \frac{1}{N^C}\sum_{i=1}^{N_C} v_t^{M,i}. \tag{2.6}$$

See Appendix A for a detailed description of the calculation of the individual profits.

Let s_t^i indicate whether a given chartist i acts as trend follower ($s_t^i = 1$) or as mean reverter ($s_t^i = -1$). A chartist chooses between the two behaviors following the concepts of the well-known binary choice model introduced by Brock and Hommes (1998) using $\bar{v}^T$ and $\bar{v}^M$ as fitness and $\gamma^i > 0$ intensity of choice parameter that reflects how influential is the overall profitability for the behavior of the given agent:

$$s_t^C = \begin{cases} 1, & \text{w. p.} \quad e^{\gamma \bar{v}_t^T} / \left(e^{\gamma \bar{v}_t^T} + e^{\gamma \bar{v}_t^M} \right) \\ -1, & \text{w. p.} \quad e^{\gamma \bar{v}_t^M} / \left(e^{\gamma \bar{v}_t^T} + e^{\gamma \bar{v}_t^M} \right). \end{cases} \tag{2.7}$$

(Note that in contrast to our model, in Brock and Hommes (1998) the choice is made between fundamentalists and chartists.) Thus, a feedback system arises in which the chartist agent applies a given behavior type with higher probability if it has been more profitable in the recent past for the whole chartist community on average. See Cesa-Bianchi and Lugosi (2006) for a comprehensive treatise of learning using exponentially weighted averages.

Using s_t we can define the belief of a given chartist from (2.4)-(2.5) as

$$\hat{p}_t^{C,i} = p_{t-1} \exp\left(s_t^i \cdot \alpha^i \cdot \bar{r}_{t-1}^i \right). \tag{2.8}$$

Fundamentalists. Fundamentalists develop their belief based on company information, they acquire each day $n_t \in \mathbb{N}$ pieces of public news. The same news are available to all fundamentalist agents. Fundamentalists reinterpret and reevaluate the news received so far each day, in a way that less weight is put on news received earlier in the past. We denote the filtered, normalized n_t by

$$\bar{n}_t^\nu = \frac{[\mathcal{E}_\nu(n)]_t}{E(n_t)}, \tag{2.9}$$

where $0 < \nu < 1$ is the factor of the exponential forgetting and $E(n_t)$ denotes the expected value of n_t. Note that n_t can be observed on real markets, see e.g. finance.yahoo.com for a given firm related financial headlines.

The news evaluation results in the opinion o_t, expressing whether the company is in a better ($o_t > 0$) or worse ($o_t < 0$) shape than the market price p_{t-1} indicates, from the agent's point of view. The opinion of a given fundamentalist agent has a common and an individual component. The common component reflects some aggregated view of the market environment. The private component is agent specific and independent from the

common opinion and the private opinions of other fundamentalists. Our opinion model captures the economic assumption that the variability of investor opinion is positively correlated to the number of news appearing on the market, see Mitchell and Mulherin (1994), Kalev et al. (2004) for empirical evidence. Thus, the common opinion can be formulated as

$$o_t^E = \mu_t + \sigma \bar{n}_t^{\left(\nu^E\right)} \varepsilon_t^E, \quad \text{with} \quad \varepsilon_t^E \sim \mathcal{N}(0,1) \quad \text{i. i. d.}, \tag{2.10}$$

$$\mu_t = -0.0001 \log\left(\frac{p_{t-1}}{p_0}\right), \tag{2.11}$$

where μ_t expectation of news takes care of mean reversion to the constant fundamental value p_0. The private opinion $o^{F,i}$ for fundamentalist i is calculated similarly with an agent specific forgetting factor $\nu^{F,i}$ and i. i. d. noise $\varepsilon_t^{F,i}$, which is independent from ε^E as well. For simplicity, we define the unconditional variance of the common and private opinion denoted by σ^2 to be equal. We can interpret σ^2 as a measure of the uncertainty about fundamentals, expressing the variability of future economic outcomes.

We can finally specify the fundamentalist belief model as

$$\hat{p}_t^{F,i} = p_{t-1} \exp\left(\frac{o_t^{F,i} + o_t^E}{2}\right). \tag{2.12}$$

Here the private and public opinions are weighted equally for simplicity.

The information arrival process n_t. The news process n_t evolves in the environment driven by analyst activity. Our novel model of n_t captures herding among analysts, resulting in a self-exciting phenomenon, see e.g. Welch (2000): announcements and news about a company trigger analysts to follow that company, and thus increase its analyst coverage. This, in turn, results in further analysts' announcements about the given company. Let us now refine and formalize this heuristics using the idea of Hawkes's continuous time point processes, see Hawkes (1971a). Let $x > 0$ denote analysts' coverage, a quantity determining the intensity of news generation. We assume that analyst coverage would erode from period to period by factor $(1 - a)$ with $0 < a < 1$ if no news appeared on the market. On the other hand, the random number of news n_t increase next period's analyst coverage by

factor $b > 0$. Thus, our number of news model can be formulated as

$$x_t = ax_{t-1} + bn_{t-1}, \tag{2.13}$$

$$n_t \sim \text{Poisson}(\mu_t), \tag{2.14}$$

$$\mu_t = m + x_t \tag{2.15}$$

where $m > 0$ is the minimum intensity. Figure 2.2 depicts a typical trajectory of n_t and x_t, we can see some clustering of volatility on the graph of n_t.

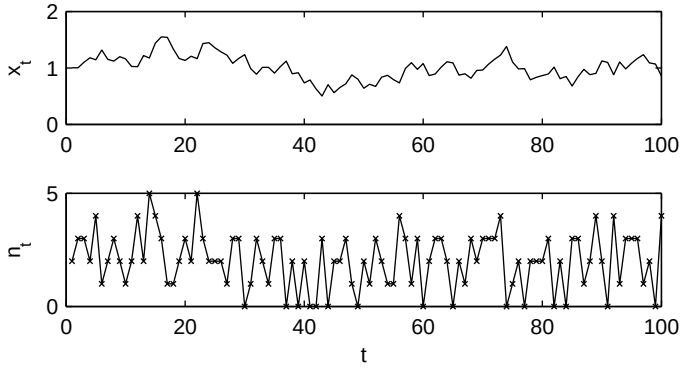

Figure 2.2: A typical trajectory of the new process n_t and analyst coverage x_t.

We can easily examine the stability properties of the discrete Hawkes process. Taking expectations on (2.14)-(2.15) yields

$$\text{E}(x_{t+1}) = (a + b)\text{E}(x_t) + bm, \tag{2.16}$$

from which we see that the process is stable if $a + b < 1$, the expected value explodes linearly if $a + b = 1$ and exponentially if $a + b > 1$. Under stationarity assumptions we can calculate $E(n_t)$ from (2.14)-(2.15), (2.16):

$$\text{E}(n_t) = m + \frac{mb}{1 - a - b}.$$

In our simulations we initialize the state process as $x_0 = mb/(1 - a - b)$.

Summary of design differences. The essential differences between our and classical ACF models are that in our model 1. the behaviors exhibit a higher degree of heterogeneity; 2. the market fraction of chartists and fundamentalists is an (exogenous) parameter; 3. the news arrival process exhibits serial correlations. Let us stress at this point that in accordance with research goals of this chapter we have not optimized the design of the ACF

model for simplicity, but we rather included details, that in our view represent essential characteristics of real markets. In the following, we examine the model by statistical means.

2.3 Model validation

We validate our ACF model by showing experimentally that it reproduces some important stylized facts one can observe in real prices. In addition we analyse the impact of the variation of market fractions on simulated returns with respect to stylized facts. We define returns as

$$r_t = \log\left(\frac{p_t}{p_{t-1}}\right).$$

Chartist and fundamentalist market fractions are defined as

$$w^C = \frac{N^C}{N} \quad \text{and} \quad w^F = \frac{N^F}{N}, \tag{2.17}$$

respectively. All agents are assumed to have the same initial endowment, thus relative weights can be seen as the initial distribution of wealth among chartists and fundamentalists. Put

$$w = \frac{w^C}{w^N}.$$

The main source of agent heterogeneity is the variety of parameters specifying agent's behavior. For simulation purposes, for each agent specific parameter we specify intervals from which we draw random samples uniformly for agents of a given type. Table 2.1 shows the parameter values and intervals applied in the simulations.

	Notation	Meaning	Value
Chart's	λ	Price filtering	$[0.02, 0.03]$
	α	Trend extrapolation	$[0, 5]$
	γ	Intensity of choice	$[1, 5]$
Fund's	ν^F	Private news forgetting factor	$[0.01, 0.03]$
	ν^E	Common news forgetting factor	0.01
News	a	Coverage erosion	0.7
	b	Amplifier factor	0.2
	m	Min. intensity	0.2
Economic factors	w	Agent type fractions	1
	p_0	Fundamental value	100
	σ^2	Uncertainty about fundamentals	0.01

Table 2.1: Baseline parameter values.

The statistics related to the stylized facts are calculated on several simulated price trajectories of length 10^5. In Table 2.2 we report results of hypothesis testing, each cell in the table shows the proportion of rejected null-hypotheses. The name of the tests are indicated in the column header. Table 2.3-2.4 report the averages of some statistics and their standard errors in italics. The first four rows correspond to increasing market fraction settings, other parameters are kept fix on the baseline values; the reported values here are calculated from 20 price series. We apply the following perturbation scheme to perform a sensitivity check on various parameter settings. The fifth row for $w \in (0.5, 1.5)$ contains results based on 100 return series, each generated with parameters perturbed uniformly randomly around the baseline values:

$$q' \sim q\,(1 + \delta u)\,, \quad u \ \sim U[-1, 1], \tag{2.18}$$

where q is the baseline parameter and q' is the applied parameter, $\delta > 0$. We applied $\delta = 0.2$ for parameter λ, $\delta = 0.05$ for parameters a and b, $\delta = 0.5$ for the rest of the parameters. In case of intervals, this perturbation rule is applied on the endpoints of the interval.

For comparison purposes, in the last row (NQ100) separated by double line we report statistics based on 1 minute intraday returns of the NASDAQ 100 index preprocessed analogously as described in Section 2.8. The statistics are calculated on 10 different returns series each spanning a week, from the period 6th December 2009 until 25th February 2010.

Note that all ACF model parameters are kept fixed over time in the simulations. This is an admissible simplification considering that agent behaviors and the economic environment are stable on the short term, a week or month, say, as major regular economic and cash flow related reports are announced quarterly. Prices spanning even such short time frames exhibit the essential stylized facts discussed below, see the statistical analysis of intraday data in Cont (2001); Taylor (2007). Hence it is meaningful to compare simulated prices to real prices with respect to stylized facts, even without more sophisticated models for the dynamics of p_0, σ^2, w. See Alfarano (2006) for an excellent summary of the statistics we apply below for the comparison.

Unit root. The column ADF in Table 2.2 shows that the augmented Dickey-Fuller test typically can not reject the null-hypothesis of unit roots in $log(p_t)$ at 5% significance level. In the column $AR(1)$ in Table 2.3 we see that the root of a fitted zero mean AR(1) model is very close to unity, which is a property that real data exhibit as well.

Autocorrelations in returns. Column $Q(1)$ in Table 2.2 reports results of the Box-Ljung

w	ADF	$Q(1)$	$ARCH(2)$	$ARCH(20)$	$GPH_{(r)}$	$GPH_{(r^2)}$
0	0.05	0.00	0.85	1.00	0.05	1.00
0.5	0.05	0.05	1.00	1.00	0.05	1.00
1	0.05	0.20	1.00	1.00	0.05	1.00
2	0.00	0.40	1.00	1.00	0.20	1.00
(0.5,1.5)	0.07	0.13	0.98	1.00	0.07	0.97
NQ100	0.10	0.20	1.00	1.00	0.00	1.00

Table 2.2: Proportion of rejected null-hypotheses.

test for checking the null-hypothesis of absence of autocorrelations at lag 1, which typically can not be rejected at 5% significance level. Note that a well-known property of the Box-Ljung test is that it rejects the null-hypothesis for heavy tailed distributions more often than the significance level reflects. The type I error accumulates with higher lags. In Table 2.3 $\hat{R}_{(r)}(l)$ denotes the sample autocorrelation of returns at lag l, we calculated very small values. Empirical studies report both statistically insignificant and significant lag 1 sample autocorrelation in returns. The magnitude in the significant cases is approximately 0.1 for daily data and -0.1 for intraday data.

The unit root tests and the measured autocorrelations in returns give statistical evidence for near random walk characteristics of simulated prices. This happens despite model components which introduce serial dependences in returns, such as mean reverting to constant fundamentals and trend extrapolation. This effect can be explained as the random model elements conceal the dependencies, e.g. the effect of trend extrapolation is mitigated by s_t^C in (2.8), because it has an expectation close to zero and it is almost independent from past returns. The value of the intensity of choice parameter γ has a major influence on the autocorrelations in returns, as it controls the weight of trend follower and mean reverter chartists. A high γ can result in a shift of the weight towards trend followers in a long, highly trending period, who then increase the autocorrelation of returns as $E(s_t^C)$ increases.

Volatility clustering. We tested prices for volatility clustering using Engle's ARCH test with lags 2 and 20 and reported results in Table 2.2. The absence of ARCH effects has been typically rejected even at 1% significance level, confirming volatility clustering. The results are in accordance with results of ARCH tests performed on real prices. We also report sample autocorrelations in squared returns for lags 1, 20, 50 denoted by $\hat{R}_{(r^2)}$ in Table 2.3. We can observe that increasing the weight of chartists causes stronger volatility clustering effects. We give a brief heuristic analysis of this effect in the following. The

effect of the trading of a single chartist i is that he pushes p_t towards his belief $p_t^{C,i}$. The magnitude of the price shift caused by the chartist is positively correlated with

$$\log\left(\frac{p_t^{C,i}}{p_{t-1}}\right) = \alpha^i \bar{r}_{t-1},$$

which we easily get from (2.8). Hence, the magnitude of the aggregate impact of the whole chartist community is positively correlated with $\bar{r}_{t-1}$. Thus, the effect of chartist trading can be interpreted as saying that it adjusts the variance of the market price process by an amount that is proportional to the absolute trend and controlled by parameter α. As the trend varies slowly in time due to smoothing controlled by λ, the variance induced by chartists possesses a high inertia. This explains the slow decay of the sample autocorrelation function. Higher autocorrelations by higher chartist presence is a consequence of the stronger aggregate impact of chartists on volatility. The low values in columns $\hat{R}_{(r^2)}$, row $w = 0$ in Table 2.3 reveal that autocorrelations in the filtered news process $\bar{n}_t$ do not increase the autocorrelation in squared returns significantly.

Long memory. We examined whether long-term dependence is exhibited by returns, and squared returns series using the method of Geweke and Porter-Hudak (GPH). We applied a frequency range between $10000^{0.1}$ and $10000^{0.8}$. We report results in columns $GPH_{(r)}$ and $GPH_{(r^2)}$ for returns and squared returns respectively. In Table 2.4, the estimated fractional differencing parameters are shown. Table 2.2 shows that the long memory null for raw returns can not be rejected typically at 5% significance level and it is typically rejected for squared returns at 1% significance level. The estimated fractional differencing parameter for squared returns falls within the range $(0.05, 0.5)$ reported by empirical studies, such as e.g. Alfarano (2006). According to the tests, long memory is exhibited by squared returns but not by raw returns. A possible source of long memory in squared returns is the heterogeneity in agents memory defined by λ, ν, see Diks and van der Weide (2005); Matteo et al. (2004).

Fat tails. In order to check for fat tails in returns distributions, we calculated the popular Hill tail index from the upper 5% and 10 % of the tail. In Table 2.4 we report results matching usual empirical findings according to which the index lies between 2 and 5. Note that intraday returns typically exhibit fatter tails than daily data. Interestingly, for some return series generated with $w = 2$, the estimated 10% index fell below 2, indicating an infinite variance of returns. The variance is known to be finite though, as we apply a cap of 2% on returns in the pricing algorithm. A tail index less than 2 are also reported

in some empirical statistical investigations, see Alfarano (2006).

We can consider heavy tails as a result of mixing different returns distributions arising in two regimes in the price dynamics: one representing calm market periods when the variance of r_t is small since $|\bar{r}_t|$ is small and one representing trending periods, i.e. when the variance of r_t is big since $|\bar{r}_t|$ is big. The difference between the variances in the two regimes controls the tail fatness. A high α reduces the difference in the variances of the two regimes inasmuch it prevents the variance of r_t from getting low in calm market periods. The tail gets heavier when the market fraction of chartists is higher, as the difference in variances in the two regimes increases as the chartist trading increases.

w	$AR(1)$	$\hat{R}_{(r)}(1)$	$\hat{R}_{(r)}(20)$	$\hat{R}_{(r^2)}(1)$	$\hat{R}_{(r^2)}(20)$	$\hat{R}_{(r^2)}(50)$
0	0.9997	0.0052	0.0022	0.0288	0.0271	0.0203
	0.0002	*0.0079*	*0.0094*	*0.0102*	*0.0087*	*0.0113*
0.5	0.9997	0.0007	0.0011	0.0518	0.0396	0.0280
	0.0003	*0.0097*	*0.0079*	*0.0142*	*0.0176*	*0.0112*
1	0.9997	-0.0005	0.0026	0.1270	0.1101	0.0689
	0.0003	*0.0136*	*0.0101*	*0.0234*	*0.0134*	*0.0160*
2	0.9997	-0.0027	-0.0035	0.2899	0.2676	0.2115
	0.0003	*0.0198*	*0.0237*	*0.0437*	*0.0474*	*0.0375*
(0.5,1.5)	0.9996	-0.0017	-0.0018	0.1033	0.0887	0.0668
	0.0004	*0.0132*	*0.0114*	*0.0855*	*0.0761*	*0.0670*
NQ100	0.9984	0.0089	0.0105	0.1849	0.0662	0.0399
	0.0012	*0.0347*	*0.0376*	*0.0655*	*0.0328*	*0.0291*

Table 2.3: Estimated AR(1) coefficient and sample autocorrelation of returns and squared returns with various lags.

Summary. Although the results above indicate that our ACF model is capable of reproducing important stylized facts exhibited by real prices, it is by far not proven that this model is a true specification of the real price generating mechanism. In fact, it is impossible to formally validate an ACF model based on financial data. However, our model is a formal description of heuristics checked in the indicated empirical literature and some of its components are well-established in ACF. Thus, considering the statistical comparison above we are confident that our model can be applied to provide an economic interpretation of statistical phenomena we observe on real prices.

w	*Hill* 5%	*Hill* 10%	$GPH_{(r)}$	$GPH_{(r^2)}$
0	4.4559	3.3065	-0.0021	0.0834
	0.1662	*0.0827*	*0.0149*	*0.0174*
0.5	4.2554	3.1322	-0.0020	0.1116
	0.1371	*0.1089*	*0.0161*	*0.0152*
1	3.7679	2.7640	-0.0002	0.1900
	0.2561	*0.1653*	*0.0174*	*0.0192*
2	1.9251	1.2955	0.0112	0.2410
	0.4648	*0.3498*	*0.0259*	*0.0352*
(0.5,1.5)	3.8173	2.7894	-0.0015	0.1465
	0.8124	*0.6445*	*0.0207*	*0.0632*
NQ100	3.0011	2.3178	-0.0105	0.1866
	0.2476	*0.1654*	*0.0275*	*0.0379*

Table 2.4: The Hill tail index and estimated fractional differencing parameter of returns and squared returns.

2.4 Fitting a GARCH model

Our goal in this section is to find a relationship between the ACF model and a technical model, since it then could possibly allow to infer (inversely) on the ACF model parameters from the fitted technical model parameters. The GARCH model, being a simple volatility model, seems to be suitable for this purpose, since according to numerical experiments performed for validation, the parameters of the ACF model have a big influence on the volatility structure of the generated time series. Thus we examine the effect of the fundamental value p_0, the uncertainty about fundamentals σ^2 introduced in (2.10)-(2.11) and the market fractions w on the parameters of a technical model fitted to the return process.

We consider a GARCH(1, 1) model given by

$$r_t = \sqrt{h_t}\varepsilon_t, \tag{2.19}$$

$$h_t = \sigma_0^* + \varphi^* r_{t-1}^2 + \psi^* h_{t-1}, \tag{2.20}$$

with $\varepsilon_t \sim \mathcal{N}(0,1)$ i. i. d. sequence, and real parameter vector

$$\vartheta^* = \begin{pmatrix} \sigma_0^* \\ \varphi^* \\ \psi^* \end{pmatrix}, \tag{2.21}$$

is such that $\sigma_0^* > 0$ and $\varphi^*, \psi^* \geq 0$. We impose the condition $\varphi^* + \psi^* < 1$, which ensures the existence of a unique stationary solution. (Alternative noise models having fat tails, are the subject of future analysis).

According to the GARCH model, the volatility of the return process r_t is determined by the conditional variance process h_t as in (2.19). The conditional variance process h_t follows a linear dynamics, it is driven by past squared returns.

The conditional quasi log-likelihood function of observations $(r_1, \ldots, r_T)$ is the following:

$$L_T(\vartheta|r_1, \ldots, r_T) = \sum_{t=1}^{T} l_t(\vartheta|r_1, \ldots, r_t) = \tag{2.22}$$

$$= \sum_{t=1}^{T} -\frac{1}{2}\left(\log \hat{h}_t(\vartheta) + \frac{r_t^2}{\hat{h}_t(\vartheta)}\right), \tag{2.23}$$

where $\hat{h}_t$ is calculated by inverting (2.20) for an assumed true parameter ϑ, under the condition

$$\hat{h}_0 = \frac{\sigma_0}{1-\varphi-\psi}, \quad r_t = 0 \quad \text{for} \quad t < 0. \tag{2.24}$$

(Thus h is initialized with the unconditional variance.) Note, that in the case of Gaussian disturbances ε_t (2.23) is the exact conditional likelihood.

Our goal is to examine the relationship between the parameters of the GARCH(1,1) model that matches best the stock returns generated by our ACF model, and the economic factors p_0, σ^2, w. First, we examine the effect of the market fraction w, while keeping p_0, σ^2 fixed. Let

$$r(w) = (r_1(w), \cdots, r_T(w))^T$$

denote the corresponding return series. We fit a GARCH(1,1) model using the standard quasi maximum likelihood method, to obtain the off-line maximum likelihood estimate $\hat{\vartheta}_T$. Applying the methods of Gerencsér et al. (2010) it can be shown that, under appropriate technical conditions, for T tending to infinity, $\hat{\vartheta}_T$ converges to a limit $\hat{\vartheta}$. The latter corresponds to the GARCH model that fits best our ACF model in a metric defined by the quasi-likelihood.

In our simulation experiment with $N = 200$ agents, we generated return series $r(w)$ of length 10^5 with varying w, $0 \leq w \leq 2$, and we fitted a GARCH(1,1) model to each time series using an efficient numerical optimization method developed recently by Torma and G.-Tóth (2010), see Chapter 3. We extracted $\hat{\varphi}$ and $\hat{\psi}$ from the resulted $\hat{\vartheta}$ and plotted

them against w which we can see on Figure 2.3.

The curve on the left shows that $\hat{\varphi}$ tends to get higher by increasing the weight of chartists, indicating a higher impact of returns on volatility. The curve on the right shows that $\hat{\psi}$ tends to get higher by increasing the weight of fundamentalists, indicating an increasing memory for $\hat{h}_t$. We can explain this finding heuristically as follows. In case of $w = 0$, i.e. without any chartist on the market, the estimated latent GARCH volatility $\hat{h}_t$ approximates the dynamics of the filtered news process n_t, which determines the volatility via the trading of fundamentalists, see (2.10). With a fairly high memory of analyst coverage ($a = 0.7$) and a low dependence on past news ($b = 0.2$) we get a high $\hat{\psi}$ and a low $\hat{\varphi}$ on the fundamentalist market. As chartist adjust the volatility based on past returns, increasing the weight of them increases the dependence of $\hat{h}_t$ on past returns and simultaneously decreases the dependence of $\hat{h}_t$ on the filtered news process.

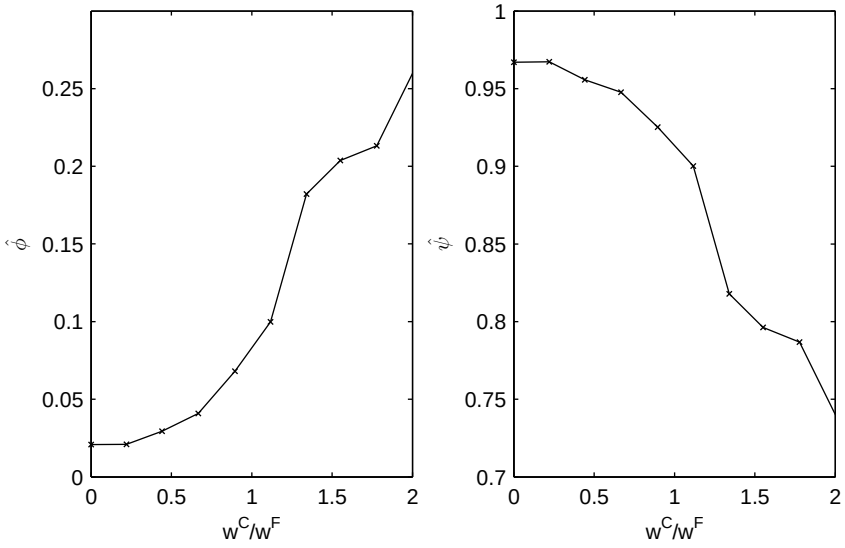

Figure 2.3: Relationship between estimated GARCH coefficients and market fractions with model parameters as in Table 2.1.

Note that we have set the baseline parameters and the upper limit for w so that the fitted GARCH parameters span a range that seems to be realistic considering some documented results of empirical GARCH fitting, see e.g. Cont (2001). We can also realize in Figure 2.3 that $\hat{\varphi} + \hat{\psi}$ is close to 1, which is a typical observation when fitting GARCH to real data.

In the following we examine the dependence of fitted GARCH parameters on p_0, σ^2, w statistically, by linear regression. In a regression experiment we first choose a parameter setting for all ACF model parameters according to (2.18) randomly, then we generate 20 times a return series of length $3 \cdot 10^4$ by varying a selected regressor again according to

(2.18). We fit a GARCH(1,1) model and perform Student's t–test with the null hypothesis being that the regression coefficient is zero. We ran this regression experiment 10 times for each regressor in order to check the sensitivity of the results on the parameter settings.

In Table 2.5 the extrema of the p-values of the hypothesis tests are reported. None of the null-hypotheses for the parameters p_0 and σ^2 can be rejected at a 5% significance level, indicating that $\hat{\varphi}$ and $\hat{\psi}$ are not correlated to p_0, nor to σ^2. On the other hand, it can be rejected in all cases for parameter w on a very small significance level, confirming the dependence we found by eye inspection of Figure 2.3. Numerical investigations indicate the following reasons for this finding. The lack of influence of p_0 on $\hat{\varphi}$ and $\hat{\psi}$ follows from the property that p_0 has no effect on the variance of the returns process at all. As for σ^2, this parameter is a constant scaler of the variance of the returns r_t and also the trend $\bar{r}_t$, hence it only controls $\hat{\sigma}_0$ but not $\hat{\varphi}$ and $\hat{\psi}$.

	$p_0 \sim \hat{\varphi}$	$p_0 \sim \hat{\psi}$	$\sigma \sim \hat{\varphi}$	$\sigma \sim \hat{\psi}$	$w \sim \hat{\varphi}$	$w \sim \hat{\psi}$
min. p-value	0.1433	0.1429	0.0910	0.0908	$1 \cdot 10^{-9}$	$3 \cdot 10^{-9}$
max. p-value	0.9236	0.9266	0.8356	0.8420	$5 \cdot 10^{-5}$	$2 \cdot 10^{-4}$

Table 2.5: Minimum and maximum p-value from the regression experiments.

Thus we come to the following conclusion.

Property 1. *Considering $\hat{\varphi}$, $\hat{\psi}$ as functions of p_0, σ^2, w, our numerical experiments indicate that*

$$\frac{\partial \hat{\varphi}}{\partial w^C} > 0, \qquad \frac{\partial \hat{\psi}}{\partial w^F} > 0.$$

Moreover,

$$\frac{\partial \hat{\varphi}}{\partial p_0} = \frac{\partial \hat{\psi}}{\partial p_0} = 0 \quad \text{and} \quad \frac{\partial \hat{\varphi}}{\partial \sigma} = \frac{\partial \hat{\psi}}{\partial \sigma} = 0,$$

for $p_0 \in (50, 150)$, $\sigma^2 \in (0.005, 0.015)$.

Discussion. At the end of this section, we shall compare our ACF model and the GARCH(1,1) model from a system's architecture point of view. In particular, we examine the main sources of the volatility of the return process. In both models we can identify two heuristic sources of volatility: market returns and the economic environment. In the GARCH case, the hidden volatility process h_t can be considered as a volatility source in the environment. In the ACF model, the number of news, denoted by n_t is generated by the environment, as shown in Figure 2.1. Note that the filtering of n_t in (2.9) increases the memory of the opinion variance process $\bar{n}_t$, which we need to get a high GARCH coefficient $\hat{\psi}$.

The main difference between the two models lies in the relationship between the environmental volatility source and market prices. In the GARCH case the conditional variance $h_t = E(r_t^2)$ depends only on past returns $(r_{t-1}, r_{t-2}, \ldots)$, implying the GARCH assumption that the effect of all economic factors having an impact on the conditional variance are already reflected in past market prices. In contrast to this, the environmental volatility process n_t in the ACF model is independent from market returns. In the light of this difference, it is very interesting that our ACF model reproduces certain properties of a GARCH process so well.

2.5 The GARCH-based change detection algorithm

The established relationship between our ACF model and the best fitting GARCH(1,1) model motivates the use of GARCH(1,1) models for statistical inference on the economic factors entering the ACF model. In particular, GARCH(1,1) models are prime candidates for detecting structural changes in our ACF model.

The setup for GARCH(1,1) is the following: assume that market returns $(r_1, r_2, \ldots)$ are generated by a GARCH(1,1) model with true parameter

$$\vartheta_t^* = \begin{cases} \vartheta_1 & \text{for} \quad t < \tau, \\ \vartheta_2 & \text{for} \quad t \geq \tau, \end{cases}$$

where τ is the change-point. The problem is to estimate τ on-line. Consider first an on-line recursive quasi maximum likelihood estimation algorithm with fixed gain:

$$\hat{\vartheta}_t = \hat{\vartheta}_{t-1} - \beta \hat{H}_t^{-1} g_t, \tag{2.25}$$

$$\hat{H}_t = (1-\beta)\hat{H}_{t-1} + \beta H_t \tag{2.26}$$

for $t > 0$, where g_t, H_t are on-line approximations of the gradient and the Hessian of the negative log-likelihood function, respectively, as described in Gerencsér et al. (2010), and $0 < \beta << 1$ is a small fixed gain. The initial value $\hat{\vartheta}_0$ is given by the user, and we may choose $\hat{H}_0 = I$.

Based on the qualitative characterization of a fairly general class of fixed gain recursive estimators given in Gerencsér (1996) we expect that the tracking error $\hat{\vartheta}_t - \vartheta^*$ is of the order of magnitude $\beta^{1/2}$ for time-invariant systems with $\tau = \infty$. On the other hand increasing β improves the adaptivity of the recursive estimation method at and after τ. For time-

invariant GARCH models with bounded noise ε_t we get

$$\sup_{t>0} E^{\frac{1}{q}} |\hat{\vartheta}_t - \vartheta^*|^q = O\left(\beta^{\frac{1}{2}}\right), \quad 1 \leq q \leq \infty$$

except for an exponentially decaying term, due to initialization effects, see Gerencsér (2006).

Let us now specify the derivatives needed in the recursion. The conditional negative quasi log-likelihood of observation r_t can be written as

$$l_t\left(r_t | \mathcal{F}_{t-1}; \vartheta\right) = \log \hat{h}_t(\vartheta) + \frac{r_t^2}{\hat{h}_t(\vartheta)}. \tag{2.27}$$

Differentiating yields

$$\partial_\vartheta l_t\left(\vartheta\right) = \frac{\partial_\vartheta \hat{h}_t(\vartheta)}{\hat{h}_t(\vartheta)} \left(1 - \frac{r_t^2}{\hat{h}_t(\vartheta)}\right), \tag{2.28}$$

We differentiate (2.28) to get the Hessian, which we can split into a term containing only first order derivatives of $\hat{h}$ and a term containing only second order derivatives of $\hat{h}$:

$$\begin{aligned}
\partial_\vartheta^2 l_t\left(\vartheta\right) &= \frac{\partial_\vartheta^2 \hat{h}_t(\vartheta)\hat{h}_t(\vartheta) - \partial_\vartheta \hat{h}_t(\vartheta)\left(\partial_\vartheta \hat{h}_t(\vartheta)\right)^T}{\hat{h}_t^2(\vartheta)} \left(1 - \frac{r_t^2}{\hat{h}_t(\vartheta)}\right) - \\
&- \frac{\partial_\vartheta \hat{h}_t(\vartheta)\left(\partial_\vartheta \hat{h}_t(\vartheta)\right)^T r_t^2}{\hat{h}_t^3(\vartheta)} = \\
&= D_t^{(1)}(\vartheta) + D_t^{(2)}(\vartheta), \\
D_t^{(1)}(\vartheta) &= c_t \cdot \partial_\vartheta \hat{h}_t(\vartheta)\left(\partial_\vartheta \hat{h}_t(\vartheta)\right)^T, \quad c_t = \hat{h}_t^{-2}(\vartheta)\left(\frac{2r_t^2}{\hat{h}_t(\vartheta)} - 1\right), \\
D_t^{(2)}(\vartheta) &= \frac{\partial_\vartheta^2 \hat{h}_t(\vartheta)}{\hat{h}_t(\vartheta)} \left(1 - \frac{r_t^2}{\hat{h}_t(\vartheta)}\right).
\end{aligned}$$

Now assume for the moment that $\vartheta = \vartheta^*$. Then, running the recursion for $\hat{h}$ with the right initial condition we have $\hat{h} = h$ and

$$\frac{r_t^2}{\hat{h}_t(\vartheta)} = \varepsilon_t^2 \quad \text{for} \quad \vartheta = \vartheta^*$$

from (2.19). Thus we have

$$E\left(D_t^{(2)}(\vartheta)\right) = E\left(E\left(D_t^{(2)}(\vartheta)|\mathcal{F}_{t-1}\right)\right) = \tag{2.29}$$

$$= E\left(\frac{\partial_\vartheta^2 \hat{h}_t(\vartheta)}{\hat{h}_t(\vartheta)} E\left(1-\varepsilon_t^2\right)\right) = \tag{2.30}$$

$$= 0 \quad \text{for} \quad \vartheta = \vartheta^*. \tag{2.31}$$

In (2.30) we used that $\hat{h}_t$ and $\partial_\vartheta^2 \hat{h}_t(\vartheta)$ are $\mathcal{F}_{t-1}$-measurable and ε_t^2 is independent from $\mathcal{F}_{t-1}$.

Based on (2.29)-(2.31) we neglect $D_t^{(2)}$ from the Hessian approximation. Numerical simulations have shown that a quasi Newtonian direction with an approximate Hessian $D^{(1)}$ points typically closer to ϑ^* from a distant ϑ than using a Hessian $D^{(1)} + D^{(2)}$.

Thus, in (2.25)-(2.26) we apply

$$g_t = \partial_\vartheta l_t\left(\hat{\vartheta}_{t-1}\right), \tag{2.32}$$

$$H_t = D^{(1)}\left(\hat{\vartheta}_{t-1}\right). \tag{2.33}$$

The algorithm is extended by a basic resetting mechanism to keep $\hat{\vartheta}$ in the stability domain, see Gerencsér and Mátyás (2007a).

In order to save the computational effort needed for the inversion of $\hat{H}_t$ in (2.25), we can rewrite the recursion (2.26) directly for the inverse $P_t = \hat{H}_t^{-1}$ using the well-known Matrix Inversion Lemma

$$(A+UCV)^{-1} = A^{-1} - A^{-1}U\left(C^{-1} + VA^{-1}U\right)^{-1} VA^{-1}$$

where A, C, U, V are matrices of appropriate size. Thus we get from (2.26)

$$P_t = \frac{1}{1-\beta}\left(P_{t-1} - q_t P_{t-1}\left(\partial_\vartheta \hat{h}_t(\vartheta)\right)\left(\partial_\vartheta \hat{h}_t(\vartheta)\right)^T P_{t-1}\right), \tag{2.34}$$

$$q_t = \left(\frac{1-\beta}{\beta c_t} + \left(\partial_\vartheta \hat{h}_t(\vartheta)\right)^T P_{t-1}\left(\partial_\vartheta \hat{h}_t(\vartheta)\right)\right)^{-1}, \tag{2.35}$$

where $\hat{h}_t$ is the recursive estimation of h_t.

On Figure 2.4 on the left we present the trajectory of the on-line estimates of GARCH(1,1) parameters using $\beta_1 = 0.0005$.

The change detection algorithm itself is based on the Minimum Description Length

(MDL) principle, and is an adaptation of the method for ARMA-processes given in Gerencsér and Baikovicius (1992, 1991).

A central idea of MDL modelling is that the estimated conditional negative quasi log-likelihood can be interpreted as the code length needed for encoding the next incoming observation, say r_t, see Rissanen (1989). Here conditional means conditional on past data. Thus we get a so-called predictive encoding procedure. The choice of the encoding procedure may depend on the assumed position of the change point. The hypothesis giving the shortest code-length will then be accepted.

To carry out this program let us consider the estimated conditional negative quasi log-likelihood for the observation r_t:

$$C_t(\beta) := l_t\left(\hat{\vartheta}_{t-1}(\beta)|r_t\right) = \log \hat{h}_t + \frac{r_t^2}{\hat{h}_t},$$

where $\hat{\vartheta}_{t-1}(\beta)$ is the estimated parameter and $\hat{h}_t$ is the estimated h_t using the recursive estimation algorithm (2.25)-(2.26) with step size β. Thus C_t is interpreted as a code length for encoding r_t. An alternative definition of C_t may be obtained by using an assumed fat-tailed distribution for the noise, such as Student's t-distribution.

The main idea of the proposed change-point detection algorithm is to run two instances of the recursive estimation algorithm with different step sizes $0 < \beta_1 < \beta_2 < 1$. (We apply throughout the chapter $\beta_2 = 2\beta_1$.) Taking into account the above mentioned properties of the fixed gain recursive estimators we note that $\hat{\vartheta}_t(\beta_2)$ has higher variance than $\hat{\vartheta}_t(\beta_1)$ in the initial phase prior to τ, while $\hat{\vartheta}_t(\beta_2)$ has better tracking abilities at and right after τ. Thus in the initial phase $C_t(\beta_2)$ is expected to exceed $C_t(\beta_1)$, on average, while at and right after τ the opposite is true:

$$\mathrm{E}\left(C_t(\beta_1) - C_t(\beta_2)\right) < 0 \quad \text{for} \quad t < \tau, \tag{2.36}$$

$$\mathrm{E}\left(C_t(\beta_1) - C_t(\beta_2)\right) > 0 \quad \text{for} \quad t \quad \text{immediately after} \quad \tau. \tag{2.37}$$

Using these properties, we can use the so-called Hinkley-detector, see Hinkley (1971) to detect the change-point. Let us define the so-called CUSUM statistics by

$$S_t = \sum_{k=1}^{t} \left(C_k(\beta_1) - C_k(\beta_2)\right),$$

and set

$$S_t^* = \min_{k<t} S_k.$$

Then an alarm is generated as soon as

$$S(t) - S_t^* > \rho,$$

where $\rho > 0$ is a prescribed sensitivity threshold. On Figure 2.4 on the right the trajectory of the CUSUM statistics is given, with τ in the middle of the horizon.

Note that the above method can also be applied if ϑ_t^* exhibits a considerable drift, rather than an abrupt change, starting at τ.

2.6 Numerical performance evaluation

We carried out extensive numerical experiments to test the performance of the change detection algorithm. Table 2.6 shows test problems I.-IV. defined by the GARCH parameters before and after the change point. The test problem coefficients are selected to be close to the coefficients one may get when fitting a GARCH(1,1) model to real data. The change in the parameters is pretty small in all test problems. In the well-conditioned settings I.-II. also the unconditional variance of returns changes at the change point, while in the ill-conditioned settings III.-IV. it is constant. Thus, we can consider testing on problem types III.-IV. as a stress test of the algorithm.

I.		II.		III.		IV.	
ϑ_1^*	ϑ_2^*	ϑ_1^*	ϑ_2^*	ϑ_1^*	ϑ_2^*	ϑ_1^*	ϑ_2^*
1	1	1	1	1	1	1	1
0.05	0.1	0.1	0.05	0.05	0.1	0.1	0.05
0.9	0.8	0.8	0.9	0.9	0.85	0.85	0.9

Table 2.6: GARCH parameters of the test problems.

Figure 2.4 shows the trajectories of the estimated parameters and CUSUM test statistics for test problem I using $\beta_1 = 0.0005$. The change-point is in the middle of the time horizon at $\tau = 10^4$. We can see in the figure that the estimated coefficients nicely track the change. Note that in the typical real application the values of ϑ_1 and ϑ_2 are not constant but slightly varying. In this case the downward trend of S_t is less stiff since $E\left(C_t(\beta_1) - C_t(\beta_2)\right)$ is higher.

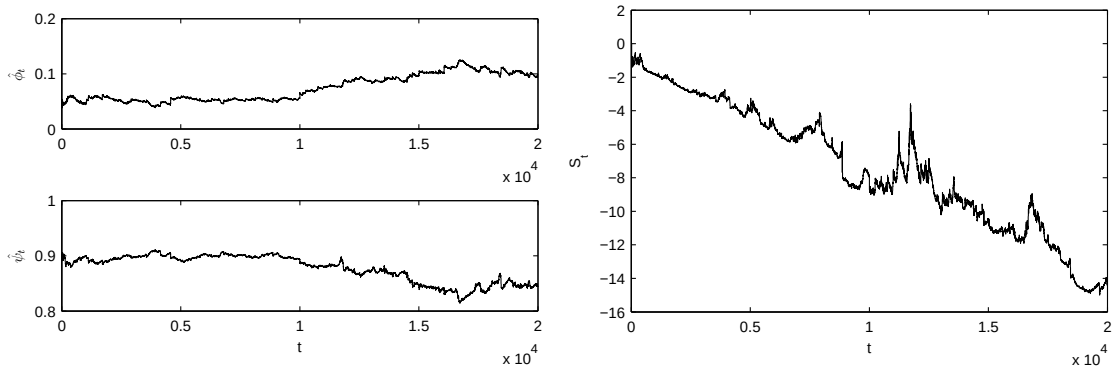

Figure 2.4: GARCH(1,1)-based recursive estimation (left) and CUSUM test statistics (right) on test problem I.

We also calculated the empirical Fisher information matrix

$$\hat{I} = \frac{1}{T} \sum_{t=1}^{T} \frac{\partial l_t(\vartheta^*)}{\partial \vartheta} \left(\frac{\partial l_t(\vartheta^*)}{\partial \vartheta} \right)^T,$$

its eigenvalues λ and its condition number κ based on $T = 20000$ observations generated with parameters defined by test problem I.:

$$\hat{I}_1 = \begin{pmatrix} 0.1444 & 2.4654 & 2.7596 \\ 2.4654 & 49.4789 & 49.6788 \\ 2.7596 & 49.6788 & 54.1695 \end{pmatrix} \quad \lambda_1 = \begin{pmatrix} 0.0027 \\ 2.0971 \\ 101.693 \end{pmatrix} \quad \kappa_1 = 3.7664 \cdot 10^4,$$

$$\hat{I}_2 = \begin{pmatrix} 0.1532 & 1.1356 & 1.3967 \\ 1.1356 & 11.3480 & 11.3907 \\ 1.3967 & 11.3907 & 13.3550 \end{pmatrix} \quad \lambda_2 = \begin{pmatrix} 0.0052 \\ 0.9289 \\ 23.9222 \end{pmatrix} \quad \kappa_2 = 4.6004 \cdot 10^3.$$

The relatively low upper-left value of the Fisher information matrices compared to the other two diagonal values indicates that the algorithm can track the parameters φ^* and ψ^* much faster than the parameter σ_0^*. The condition number is pretty high.

In order to quantify the detection capability of the change-point detection algorithm we estimated three quality measures: the probability that an alarm is generated too early, which is called false alarm probability and defined as

$$Pr(FA) = Pr(\hat{\tau} < \tau),$$

the probability that no alarm is detected until $\tau + \Delta t$, which is called missed alarm probability and defined as

$$Pr(MA) = Pr(\hat{\tau} > \tau + \Delta t),$$

and the expected alarm delay

$$E(\hat{\tau} - \tau | \tau < \hat{\tau} \leq \tau + \Delta t),$$

where $\hat{\tau}$ is the change-point estimator. The parameter Δt expresses the time delay after which a detected change is too late for the user. In experiments calculating the expected delay, we set the time out large to get only a few missed alarms. The effect of the time out setting is then examined separately.

We tested the algorithm with two different step-size settings, $\beta_1 = 0.0005$ and $\beta_1 = 0.001$. In all experiments the length of returns series is $n = 10000$, the change-point is at $\tau = 5000$. For all test problems, we generated 50 times the return series and calculated the corresponding cumulative sums S_t and then estimated the three quality measures from the trajectories S_t.

There is a trade-off in choosing the sensitivity threshold ρ: a high ρ results in a low false alarm probability but yields a higher expected alarm delay and a higher rate of missed alarms at the same time. Thus, in Figure 2.5-2.6 we plot the $Pr(FA)$ (solid) and average delay (dashed) against a varying ρ.

Comparing the two figure columns we can conclude that the effect of the step-size on the average delay for similar false alarm probability levels differ for different test problem types. For the well-conditioned problems, faster forgetting yields slightly better results. For the ill-conditioned problems, slower forgetting yields better results.

Figure 2.7 shows $P(MA)$ vs. Δt applying a ρ which is the smallest that resulted in $P(MA) = 0.1$ in the tests shown in Figure 2.5-2.6; only results with the better forgetting factor are presented.

For comparison purposes we report numerical results also for the Threshold GARCH (TGARCH) model in Figures 2.8-2.9 in a similar fashion as for GARCH except that here only results with one forgetting factor setting are included. The TGARCH model captures asymmetries in the volatility of returns: (2.20) modifies to

$$h_t = \sigma_0^* + \left(\varphi^-\right)^* I\left\{r_{t-1} < 0\right\} r_{t-1}^2 + \varphi^* r_{t-1}^2 + \psi^* h_{t-1},$$

where $I\{x < 0\} = 1$ if $x < 0$ and 0 otherwise. In the TGARCH related experiments we

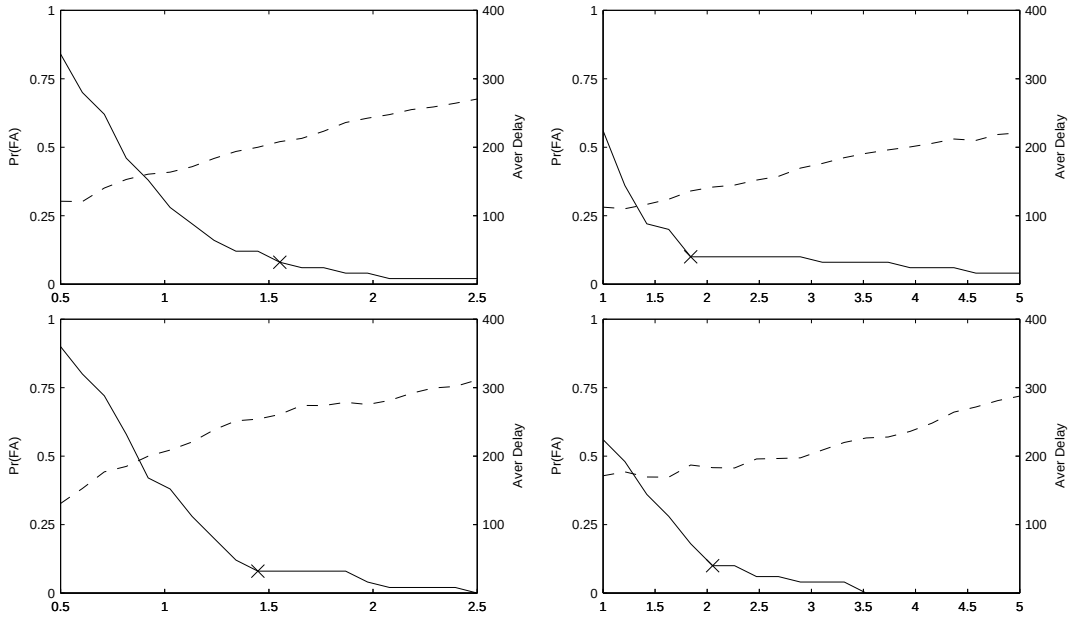

Figure 2.5: GARCH: Results for test problems I.-II. from top to bottom, with $\beta_1 = 0.0005$ (left) and $\beta_1 = 0.001$ (right). Crosses indicate the smallest ρ s. t. $P(MA) = 0.1$.

used the parameter settings from Table 2.6 for σ_0^* and ψ^*, and we set

$$\left(\varphi^-\right)^* = \varphi^* = \frac{2}{3}\varphi_G^*,$$

with φ_G^* being the second coefficient of the parameter vector from Table 2.6.

Since the TGARCH(1,1) model is in general more difficult to estimate than the GARCH(1,1) model, we used smaller forgetting factors in the TGARCH case. We get a slightly worse overall change detection performance for the TGARCH model in comparison to the GARCH model.

Change detection in market fractions. According to Property 1, we would expect a change in parameters of a GARCH(1,1) model fitted to a returns series generated by our ACF model with an abrupt change in the market structure parameter w. We verified this intuition by running our recursive estimation algorithm (2.25)-(2.26) on returns simulated by our ACF model with w satisfying

$$w_t^* = \begin{cases} w_1 = \frac{1}{5} & t < \tau, \\ w_2 = 1 & t \geq \tau. \end{cases}$$

In Figure 2.10 on the left $P(FA)$ is shown against ρ, $\times$ and $+$ indicate the smallest ρ

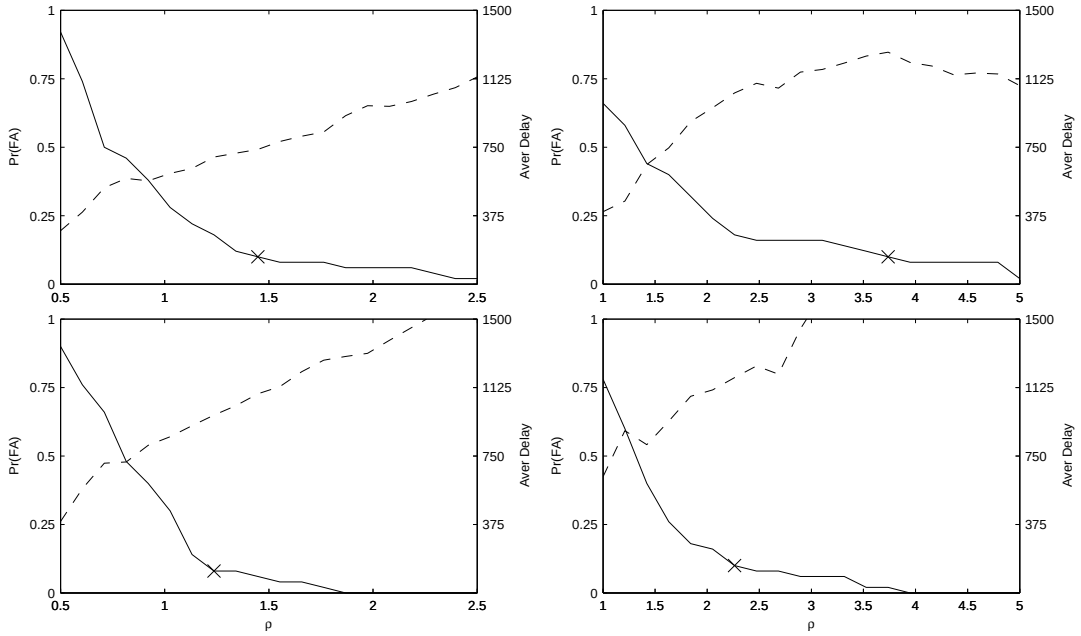

Figure 2.6: GARCH: Results for test problems III.-IV. from top to bottom, with $\beta_1 = 0.0005$ (left) and $\beta_1 = 0.001$ (right). Crosses indicate the smallest ρ s. t. $P(MA) = 0.1$.

at which $P(FA) = 0.1$ and $P(FA) = 0.2$, respectively. On the right we show $P(MA)$ corresponding to the smaller (dashed) and bigger (solid) sensitivity thresholds.

We stress that the GARCH(1,1) model is an incorrect representation of the return series generated by the ACF model. In particular, we found that the distribution of the estimated residuals has a fatter tail than the normally distributed GARCH residuals. The misspecification of the driving noise increases by increasing the weight of chartists. Because of model misspecifications we applied a quite low $\beta_1 = 5 \cdot 10^{-5}$. Examining the algorithm based on a GARCH model with ε_t exhibiting heavy tails, such as the Student's t-distribution, say, is subject of future research.

Regarding the application of the algorithm on real data we conclude that the delay tolerance of the user is a major factor determining the sampling rate of the price data under analysis. For example, a tolerance level of approximately two weeks, which may be characteristic for medium term investors or regulators, requires 15 minute intraday data. See related empirical tests in Section 2.8.

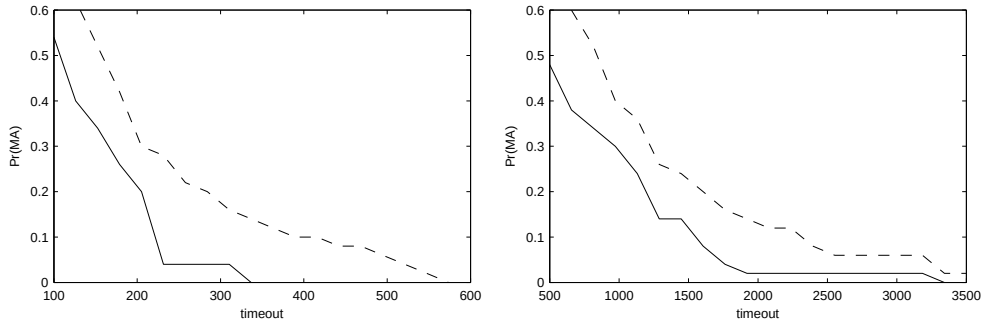

Figure 2.7: GARCH: Missed alarm probability vs. timeout for problem I. (solid), II. (dashed) on the left and III. (solid), IV. (dashed) on the right.

2.7 Interpretation of an alarm

Although the applicability of misspecified models for change detection is a widely accepted principle, a closer look in any specific application is needed. In this section we provide a possible interpretation of an alarm generated by our change detection algorithm using GARCH models.

Property 2. *Let the price process be generated by our ACF model so that the behavior patterns of the agents, and the information generating system do not change over time. On the other hand, let us assume that the parameter vector* (p_0, σ^2, w) *does change at* τ*, but is constant before and after time* τ*, and the value of* w *does change. Then, the parameters* $\hat{\varphi}$*,* $\hat{\psi}$ *of the GARCH(1,1) model that matches best the data before and after* τ *can not be identical. Thus a change detection algorithm based on a GARCH(1,1) model will detect the change in the market fraction parameter* w*.*

Let us now discuss the assumptions formulated in the property.

Fix behaviors. Agent parameters and parameters of the information arrival process represent variables controlling the behavior of portfolio managers and analysts, respectively. We may assume that the underlying behavior of market participants is stable over time naturally, although may change slowly via a learning process. In addition, learning has a stabilizing effect, as demonstrated in Gerencsér and Mátyás (2007b).

Factors that may trigger a change in market fractions. Recall that $\left(w^C, w^F\right)$ denote the distribution of wealth among fundamentalists and chartists. According to standard utility theory, risk averse investors redistribute their wealth if their expectations about profit extent and risk, regarding the two strategies, change. We can conclude, that a change in profitability expectations may imply a change in the agent weights.

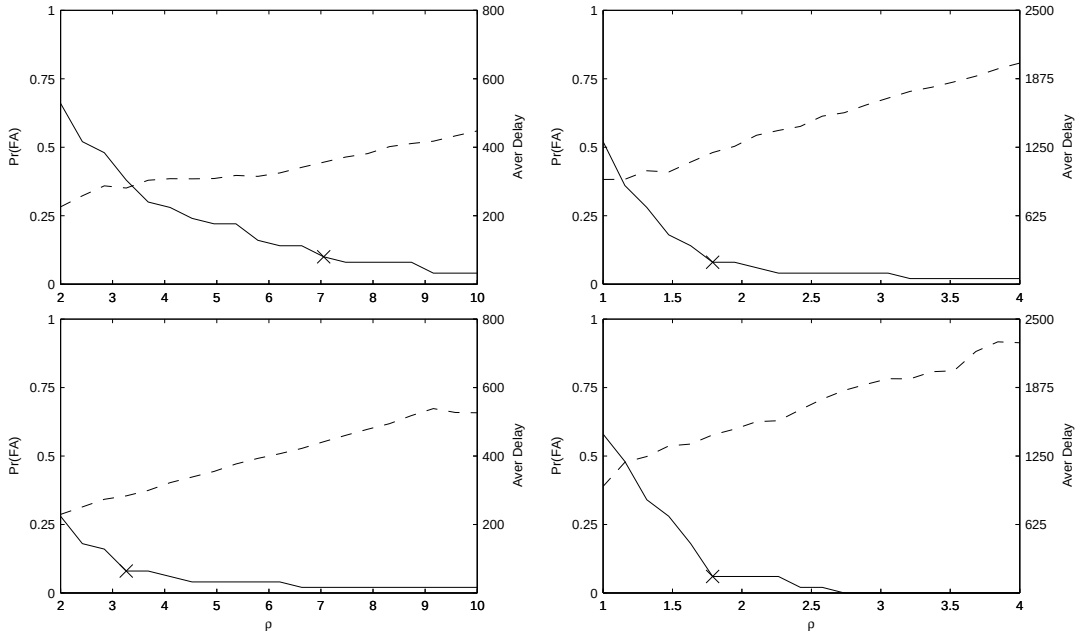

Figure 2.8: TGARCH: Results for problems I.-II. with $\beta_1 = 0.0005$ on the left and III.-IV. with $\beta_1 = 0.00025$ on the right. Crosses indicate the smallest ρ s. t. $P(MA) = 0.1$.

Chartists measure their profitability using past observations. Since the statistical characteristics of the price process are typically stable over time, significant abrupt changes in the expectations of chartists are unlikely.

Profitability expectations of fundamentalists may change if the difference between the current market price and the belief about future fundamental values changes significantly. The main factors determining their future profit expectations can be captured by the fundamental value p_0 and the uncertainty about fundamentals σ. In fact, it seems plausible that public information attracting considerable attention may cause portfolio managers to change their beliefs about the uncertainty of future fundamental values. This may in turn trigger them to redistribute funds between chartist and fundamentalist strategies. Thus, we conclude that p_0 and σ do have an influence on $\hat{\varphi}$ and $\hat{\psi}$, but this influence is realized only via w.

2.8 Empirical findings

We tested our change-point detection algorithm on real market data. We downloaded 15 minute prices of the NASDAQ100 index from the data provider FINAM. As discussed in Section 2.6, the performance of the change detection algorithm suggests a delay of

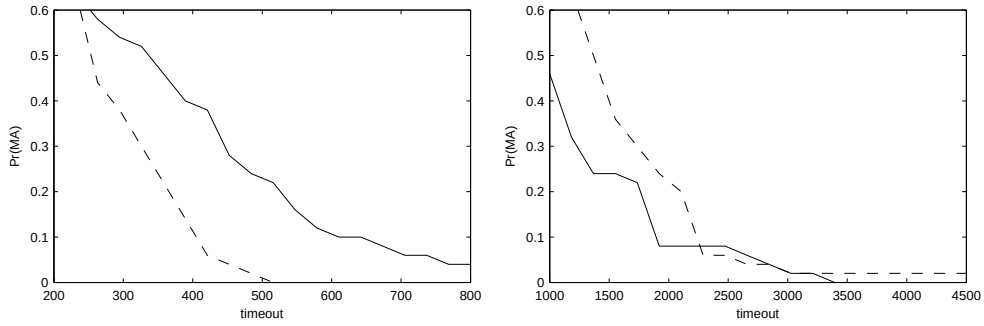

Figure 2.9: TGARCH: Missed alarm probability vs. timeout for problem I. (solid), II. (dashed) on the left and III. (solid), IV. (dashed) on the right.

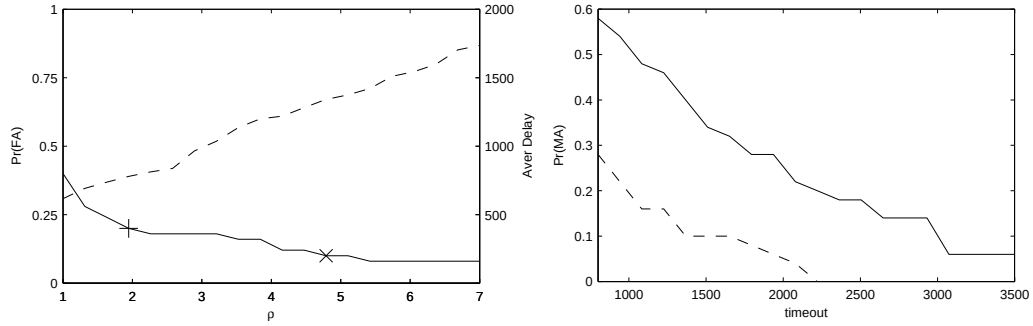

Figure 2.10: Change detection based on returns generated by the ACF model.

approximately 2 weeks for change alarms based on this time scale. The dataset spans the time interval between the 17th December 2002 and 30th October 2009 and contains 44483 data points. In order to exclude the sometimes very large nightly price gaps, we used the daily opening price instead of previous trading day's closing price in calculating the return of the first 15 minute interval of a day.

Figure 2.11 depicts the GARCH parameters fitted recursively with step size $\beta_1 = 0.0004$. In order to avoid spurious estimators due to an inaccurate Hessian in the burn-in phase, we applied a very low $\beta_1 = 0.00002$ for strong averaging for the first 10^5 iterations until approx. August 2004. We set $\hat{\vartheta}_0$ to the result of fitting GARCH(1,1) in an off-line manner on the first 25000 sample points.

Figure 2.12 depicts the results of change-point detection, in which we applied a sensitivity threshold $\rho = 10$. The CUSUM statistics S_t does not show the clear downward trends before the change-point as in the case of simulated data due to model misspecification effects and also because the best fitting GARCH parameters may vary a little even in calm market periods. In order to allow for subsequent alarms, we restarted the Hinkley

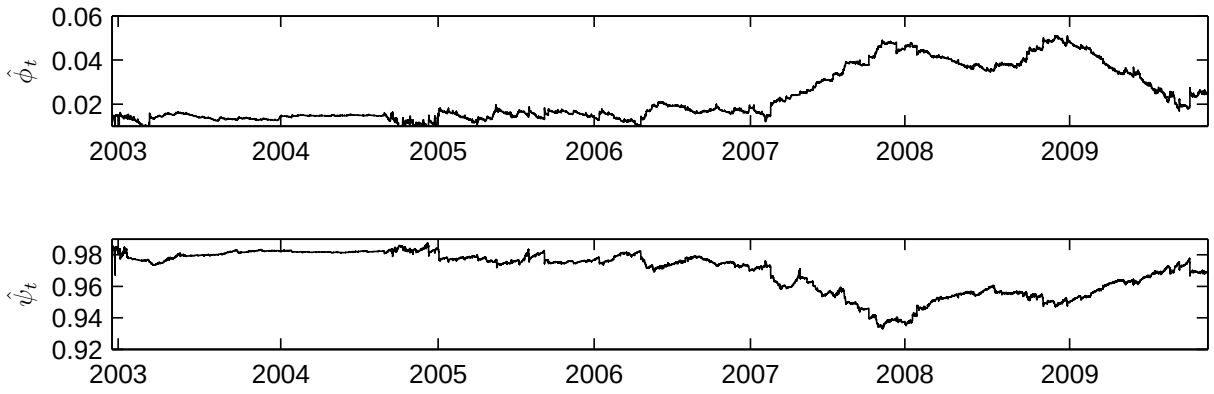

Figure 2.11: GARCH coefficients estimated from NASDAQ100 15 minute returns.

detector after every detected alarm points, the locations of which are indicated by vertical arrows.

In view of Property 2, the change alarms indicate a change in market fractions. Figure 2.11 suggests an immense increase in the share of chartists near the beginning of 2007. The change-point arrows are included in the diagram on the top as well, so that we can examine the effect of the assumed sudden big changes in the market fractions on the market price dynamics. In the price chart on the top, we can realize huge moves of market prices after almost every change-point signal. At some points, the forecast of the trend beginning is astonishingly accurate, for example in October 2007, September 2008, March 2009. The figures suggest that a sudden change in market fractions causes a consistent up or downward trend in market prices.

2.9 Conclusion

In this chapter we present a simulation study, in which we give a possible economic explanation of the coefficients of the GARCH(1,1) model. We generated price returns using a novel agent-based stock market model by varying the market fractions of chartists and fundamentalist and fitted a GARCH(1,1) model. We found a monotonic relationship between market fractions and GARCH(1,1) parameters.

Motivated by the ability of the technical model to reproduce data generated by the fundamental model we use GARCH models to detect changes in the market structure. An alarm indicating a significant increase of chartists' market share may trigger regulators to

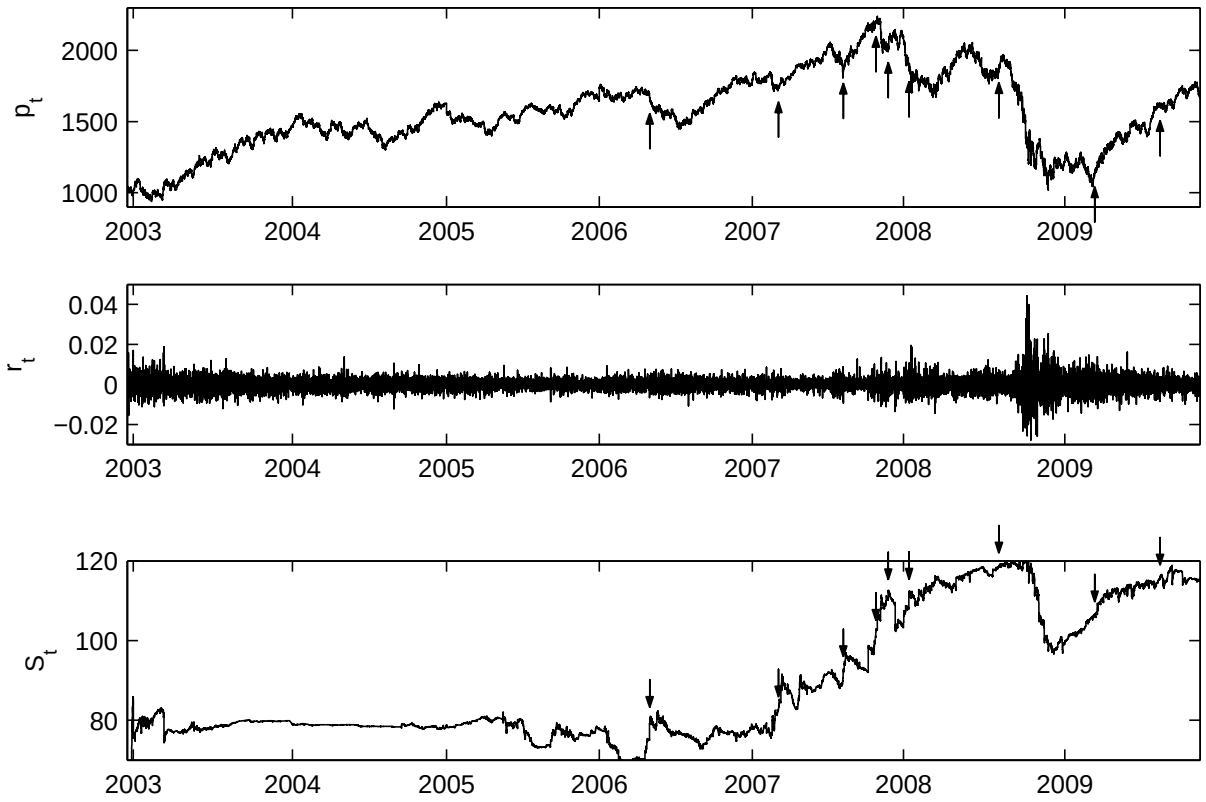

Figure 2.12: NASDAQ100 prices, returns and CUSUM test statistics. Vertical arrows indicate the change-point alarms.

prohibit shorting, in order to damp this form of market destabilizing chartist trading. We present a real-time change detection algorithm applying the Minimum Description Length principle based framework developed by Gerencsér and Baikovicius (1992, 1991). The algorithm contains a novel recursive method for GARCH parameter estimation, which is an efficient extension of the method analysed in Gerencsér et al. (2010).

We have tested the change-point detection algorithm extensively on simulated data. According to the detection performance, a promising application area is to detect structural changes based on intraday or high-frequency financial data. Thus we tried our algorithm on 15 minute data of the NASDAQ100 index and found that alarms on the assumed abrupt changes in the market structure occur just before the price trends become consistent, up or down, indicating that a real change in the market dynamics has indeed occurred.

The in-depth analysis of the robustness of the algorithm against model misspecifications is an interesting subject of future research.

Bibliographical remarks

The main ideas of this chapter were presented at the 3rd International Conference on Computational and Financial Econometrics (CFE'09) held in Limassol, Cyprus, 29-31 October 2009 and at the Erich Schneider Seminar at the Christian-Albrechts-Universität zu Kiel, Germany on the 14th January 2010 on kind invitation by Prof. Dr. Thomas Lux. A corresponding manuscript is due to submit for journal publication. The recursive GARCH estimation component of the change detection algorithm has been accepted for publication in the proceedings of the 19th International Symposium on Mathematical Theory of Networks and Systems, (MTNS 2010), Budapest, Hungary, see Gerencsér et al. (2010).

The GARCH model related theoretical parts of the chapter (Sections 2.4, 2.5, 2.7) are the joint work of László Gerencsér and Balázs Torma, the recursive GARCH estimation algorithm was analysed in cooperation with Zsanett Orlovits. The ACF model related parts, and all statistical and numerical results (Sections 2.2, 2.3, 2.6, 2.8) are by Balázs Torma.

Chapter 3

Efficient off-line model identification

3.1 Introduction

In this chapter we present a general local optimization method, which we used extensively for identification of Generalized Autoregressive Conditional Heteroscedastic (GARCH) models based on the output of the ACF model presented in Chapter 2. The algorithm is very efficient and easy to implement, which made it very suitable to include in the JAVA environment for the ACF model.

The optimization problem we would like to solve is

$$\min_{x \in S} f(x), \tag{3.1}$$

where $f : S \to \mathbb{R}$ is a twice continuously differentiable function with an optimal point x^* inside S, where $S \subset \mathbb{R}^n$ is given by bound constraints. For simplicity, we exclude the cases when x^* can reside on the boundary of S. Let $f^* = f(x^*)$. We would like to solve the optimization problem iteratively, i.e. to find a *minimizing sequence* $x_0, x_1, x_2, \ldots \in S$, such that $f(x_k) \to f^*$ as $k \to \infty$. The calculation of $f(x)$ and its derivatives is assumed to be computationally expensive.

The main motivation of this research is parametric model identification, see Soderstrom and Stoica (1989), with a given system model $\mathcal{M}(\vartheta)$ characterized by a parameter vector $\vartheta \in S$. Given a series of observations from the system, denoted by $y_0, y_1, y_2, \ldots$, we would like to find the parameter vector $\hat{\vartheta}$, such that $\mathcal{M}(\hat{\vartheta})$ best describes the system. Two well-known example of tools solving the identification problem is least squares estimation and maximum likelihood estimation. In both cases, $\hat{\vartheta}$ is determined by minimizing a loss

Input: tolerance level ε, starting point x_0
Initialization: $k = 0$.
repeat
1 *Search Direction.* Determine a descent direction Δx_k.
2 *Line Search.* Choose step size $t > 0$, s. t. $f(x_k) - f(x_k + t\Delta x_k)$ is sufficiently large and $x_k + t\Delta x_k \in S$.
3 *Update.* $x_{k+1} = x_k + t\Delta x_k$, $k = k + 1$.
until $\|t\Delta x_k\| < \varepsilon$.

Algorithm 1: General descent method.

function

$$V(\vartheta, y_0, y_1, y_2, \ldots)$$

in ϑ, which, by choosing $f = V$ and $x = \vartheta$, directly leads to the optimization problem (3.1). Evaluating V and its derivatives for larger systems usually requires processing a huge amount of data, which makes the calculation expensive. An important problem of this type is the identification of GARCH models, see Bollerslev (1986); Engle (1982), which we will address in Section 3.4.3 in more detail.

A widely used iterative scheme to solve optimization problem (3.1) is described by Algorithm 1, see Boyd and Vandenberghe (2004); Fletcher (1980). We call this algorithm framework as *descent method*, because in each iteration, Δx_k satisfies

$$\nabla f\left(x_k\right)^T \Delta x_k < 0, \tag{3.2}$$

where $\nabla f(x)$ is the gradient vector of f at x. Condition (3.2) ensures that Δx_k points in a direction where f decreases at x_k. It is a well-known property of descent methods like the Newton method, that they have a rapid convergence rate near x^*. Thus, we can split the iterations of descent methods into two stages: a *fast convergence phase*, when x_k is already near to x^*, and a *damped phase* before, when $t < 1$ damps $\|\Delta x_k\|$ in order to get a sufficiently small $f(x_k + t\Delta x_k)$.

In Step 2, the line search minimizes

$$f(x_k + t\Delta x_k)$$

in $t > 0$. Let t^* denote the optimal t. The algorithm choosing a t very close to t^* is called *exact line search*. In practice, to save computational time, t^* is only approximated with an *inexact line search* algorithm, in which a step size t is chosen, such that $f(x_k +$

$t\Delta x_k$) decreases sufficiently. Conditions for sufficient decrease are developed for example in Armijo (1966), for a survey of these methods see Shi (2004). The line search approximation is again carried out iteratively. In the damped phase, f may be evaluated at several points on the search line, which is a costly operation in several applications in model identification. In particular, the *backtracking line search*, starting with $t = 1$, reduces t by halving it in a cycle until the objective is decreased sufficiently. Another popular class of inexact line search algorithms is polynomial line search as discussed in Fletcher (1980); Murray and Overton (1978), in which a polynomial is fitted to the objective in the search direction and the minimizer of the polynomial provides the step size. In our numerical experiments, we have chosen backtracking line search for its simplicity and effectiveness in practice.

In the optimization literature, for example in Boyd and Vandenberghe (2004); Fletcher (1980), even for inexact line search algorithms it is considered to be important to approximate t^* well, i.e. to decrease the objective as much as possible. In model identification, as function evaluations are expensive, the main task of a line search algorithm should be to help the descent method to arrive close to x^* fast, i.e. to have a short damped phase. This objective does not necessarily coincide with the one mentioned above, i.e. to determine a nearly optimal step size t^*. It is easy to see that it is actually not even necessary to decrease f in order to get closer to x^* in an iteration. See Auslender et al. (2007); Zhang and Hager (2004) for recent nonmonotone line search techniques that allow some increase of the objective, and are, at the same time, effective in saving function evaluations.

For a convex function f, *localization methods* can also be used for minimization. Localization methods have a completely different mechanism compared to descent methods. The general problem they solve is to find a point in a given set $X \subset S$, implicitly defined by an oracle. The oracle can decide, whether an arbitrary point $y \in S$ is contained in X or not. If for the current point $x_k \notin X$, it provides a separating hyperplane through x_k. A localization polyhedron containing X is shrunk by the hyperplanes iteratively until the oracle says $x_k \in X$ for some k. The query point x_k is determined as some center of the localization polyhedron – it can be for example the Chebyshev center, center of maximum volume ellipsoid or analytic center, see Chapter 8.5 of Boyd and Vandenberghe (2004), and Boyd (2008) for an introductory survey and freely available program code for centering methods. The method applying analytic centers has been chosen for further analysis because of its superior performance. Minimization can be achieved by a localization method by choosing an oracle answering 'yes' if $\|\nabla f(x_k)\|$ is small enough, for a simple example.

The Analytic Center Cutting Plane Method (ACCPM), see Sonnevend (1988), is mainly used for minimization of non-smooth functions, because in the smooth case, when also

Input: $X \subset S$
Initialization: $\bar{\mathbb{H}}_0 := \mathbb{H}_0$
while *true* **do**
1 *Calculate AC.* Set $x_k = \text{argmax}_y \prod_{i=1}^{m} p_i^T(o_i - y)$, with o_i, p_i defining the m halfspaces in $\bar{\mathbb{H}}_k$.
2 *Query oracle.* Is $x_k \in X$? If yes, then break, else determine halfspace $H_{\nabla f(x_k)}$ defined by normal vector $\nabla f(x_k)$ and point x_k.
3 *Shrink polyhedron.* Set $\bar{\mathbb{H}}_{k+1} = \bar{\mathbb{H}}_k \cup H_{\nabla f(x_k)}$, $k = k + 1$.
end

Algorithm 2: Analytic center cutting plane method (ACCPM)

second order derivatives are available, other methods, such as the damped Newton method is known to be much more efficient. In the non-smooth case, ACCPM applies subgradients where f is not differentiable. The ACCPM algorithm is proved to terminate in a finite number of iterations, see the corresponding result in Ye (1997). Note, that Ye (1997) also specifies a Newton type minimization procedure for calculating the analytic center (see Step (1) of Algorithm 2), for which a complexity estimation is also given. ACCPM converges only in the convex case, we are not aware of any extensions for a non-convex objective.

The ACCPM, in the form presented in this chapter by Algorithm 2, approaches x^* without evaluating the objective function, as opposed to damped descent methods. Furthermore, intuitively, the center of a shrinking localization polyhedron may approach x^* in early iteration steps faster, than the search point in a descent method. Figure 3.1 confirms this intuition by comparing ACCPM and the Newton method damped by backtracking line search for the function (3.7) defined later. We measured the computational effort needed by the algorithms for the search point x_k to satisfy $\|x_k - x^*\| < \delta$, for different $\delta > 0$. The evaluation index is based on the number of function evaluations, the number of gradient and Hessian evaluations for the methods. The minimizer was $x^* = (-3, 2)^T$, while the initial polyhedron for ACCPM was defined by the bound constraints $[-10, 3] \times [-6, 8]$. For the damped Newton method, different starting points x_0 were taken equidistantly from a circle around x^* with a radius $r = 5$. We calculated the average of the corresponding efficiency indexes. It can be seen from the figure, that for ca. $\delta > 10^{-2.5}$, ACCPM reaches the target neighborhood faster than the damped Newton method.

Descent methods, in contrast to ACCPM, work for nonconvex objective functions as well and have, in general, a faster asymptotic convergence rate than ACCPM. Figure 3.1 also shows the asymptotic performance superiority of the Newton method over ACCPM.

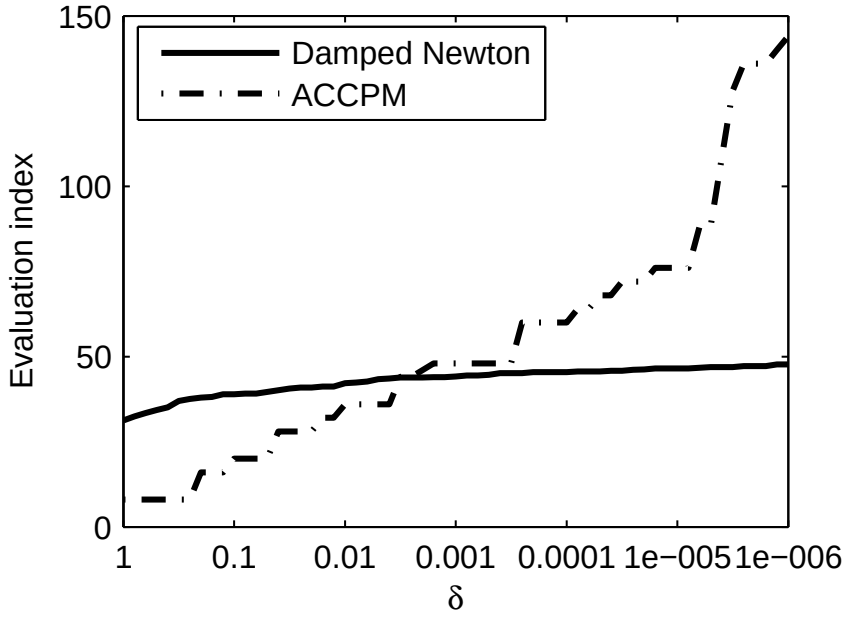

Figure 3.1: Computational effort to reach δ-neighbourhood of x^*

A further drawback of ACCPM is that redundant hyperplanes may be involved in the calculation of the analytic center, as seen in Step 1 of the ACCPM algorithm. The linear inequalities of redundant cutting planes may unnecessarily displace the analytic center. This, in an extreme case, can result in a center that lies close to the boundary of the localization polyhedron, which is quite counter-intuitive. It would be desirable to eliminate redundant cutting planes, but identifying them is a costly operation.

The article is organized as follows. In the next section we present a novel bound constrained optimization algorithm merging the advantages of descent methods and localization methods. In Section 3.3 the convergence of descent methods and our method is discussed. In the subsequent sections we give experimental evidence of the superior performance of our method in comparison with well-established descent methods addressing selected optimization problem types. Namely, in Section 3.4.1, we test our method on convex nonlinear optimization problems and show that the performance of our method is robust regarding the size of the initial polyhedron defined by the bound constraints and regarding the value of the initial search point x_0. In Section 3.4.2 we examine our method on several nonlinear least squares problems. In Section 3.4.3 we identify GARCH models with our method and point out that it is robust against noisy objective functions with many local minima. We conclude the chapter with Section 3.5, in which we summarize our results.

3.2 The Descent Direction Method with Cutting Planes (DDMCP)

In this section we describe a novel hybrid algorithm, which takes steps in descent directions and damps them by cutting planes, if necessary. In each iteration, it calculates a descent direction as a descent method together with a cutting plane like an oracle in a localization method. The algorithm determines the step length to the midpoint of the line segment between the current search point and the boundary, in the previously calculated descent direction, and takes the smaller between this or $t = 1$.

For the detailed description some definitions and notations are needed.

Definition 1. *Let $\mathbb{H}$ denote a set of half-spaces, and $\pi(\mathbb{H})$ denote the encapsulating polyhedron, which is the intersection of all half-spaces in $\mathbb{H}$:*

$$\pi(\mathbb{H}) = \bigcap \{H \in \mathbb{H}\}.$$

Let Δx denote a descent direction in x. We denote by $d_{\Delta x,P}$ the relative length of the line segment starting at x in the direction of Δx, ending at the boundary of the encapsulating polyhedron P:

$$d_{P,\Delta x} = \sup\{\alpha : x + \alpha \Delta x \in P\}.$$

Now we have all the notions necessary to describe our Descent Direction Method With Cutting Planes (DDMCP) in Algorithm 3.

Input: $\delta, \varepsilon, \mathbb{H}_0, x_0$
Initialization: $\bar{\mathbb{H}}_0 := \mathbb{H}_0$, $k = 0$
repeat
1. *Search Direction.* Determine a descent direction Δx_k.
2. *Cleaning.* Remove half-spaces that are too close to x_k in the direction Δx_k: $\bar{\mathbb{H}}_k = \left\{H \in \bar{\mathbb{H}}_k \setminus \mathbb{H}_0 \mid d_{H,\Delta x_k} > \delta \left\|\Delta x_k\right\|\right\} \cup \mathbb{H}_0$.
3. *Step size.* Choose step size $t = \min\left(1, \frac{1}{2} d_{P_k,\Delta x_k}\right)$, where $P_k = \pi(\mathbb{H}_k)$ is the encapsulating polyhedron.
4. *Extend polyhedron.* $\bar{\mathbb{H}}_{k+1} = \bar{\mathbb{H}}_k \cup H_{\nabla f(x_k)}$, where $H_{\nabla f(x_k)}$ is the halfspace defined by the hyperplane through x_k with normal vector $\nabla f(x_k)$.
5. *Update.* $x_{k+1} = x_k + t\Delta x_k$, $k = k + 1$.

until $\|\Delta x_k\| < \varepsilon$.

Algorithm 3: Descent direction method with cutting planes (DDMCP)

Formally, DDMCP can be regarded as a descent method as in Algorithm 1, with a specific technique determining step size t, defined by steps 2-4 of DDMCP. In fact, the main novelty of DDMCP lies in the determination of t. We avoid calling steps 2-4 of DDMCP a 'line search' though, because it does not necessarily decrease the objective, as the term would suggest in most of the optimization literature. In DDMCP, as in most descent methods, the method of determining the search direction is independent from the method of determining the step size.

In Step 1 of DDMCP, the search direction is calculated in the same way as in an appropriate descent method that the user would choose to solve the given minimization problem. In this chapter we have chosen minimization problems, such as convex nonlinear minimization, nonlinear least squares and maximum likelihood estimation, for which there exist well-known descent methods, where the search direction can be calculated from local derivatives only, that is, without using any derivative values from previous iterations. We have compared the appropriate descent method to DDMCP.

The essential idea of DDMCP is formulated in Step 3. In this step the method calculates the midpoint $\bar{x}_k$ of the line segment in the search direction Δx_k from the current search point x_k to the boundary of the current encapsulating polyhedron. Since the midpoint equalizes the distance on the line segment from the borders, $\bar{x}_k$ can be seen as a center of the polyhedron with respect to the current search point and search direction. Thus, the centering idea of localization methods with its intuitive advantages as described in the introduction is included in DDMCP. The euclidean distance of x_k from $\bar{x}_k$ is

$$\|x_k - \bar{x}_k\| = \frac{1}{2} d_{P_k, \Delta x_k} \|\Delta x_k\| .$$

Thus, the formula

$$t = \min\left(1, \frac{1}{2} d_{P_k, \Delta x_k}\right)$$

means, that t is chosen to be 1, if

$$\|\bar{x}_k - x_k\| > \|\Delta x_k\| .$$

In this case we reject $\bar{x}_k$ to be the next search point and we take a shorter step with length $\|\Delta x_k\|$ to allow a rapid convergence inherited from the corresponding descent method. On the other hand, we use the damping effect by accepting $\bar{x}_k$ if

$$\|\bar{x}_k - x_k\| \leq \|\Delta x_k\| .$$

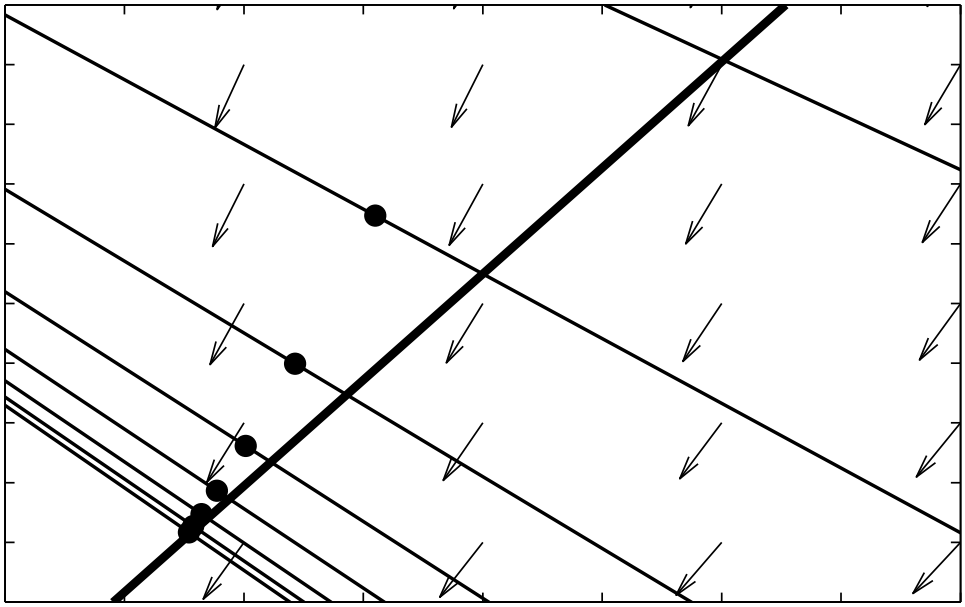

Figure 3.2: A blocking cutting plane

Step 2 in DDMCP ensures that no cutting plane in $\bar{\mathbb{H}}_k \setminus \mathbb{H}_0$ can block the search point to progress further along the search direction. To see how blocking can occur, consider first Step 4, where a cutting plane is added to the localization polyhedron $\bar{\mathbb{H}}_k$. Remember, the cutting plane with normal vector $\nabla f(x_k)$ through search point x_k has a *local* separating property in x_k, that is, in a sufficiently small neighborhood of x_k, the objective function is greater behind the cutting plane than in front of it. For points on the hyperplane far away from x_k, this local separating property may not hold any more. Consider the case when the algorithm approaches a point y on the hyperplane, where y does not have this separating property. In this situation, no matter how close x_k is to y, the search direction points towards the blocking hyperplane.

Without deleting the blocking hyperplanes, as described in Step 2, the algorithm would get stuck at the congestion point y. This situation is exhibited in Figure 3.2 on the gradient field of a two dimensional objective function. Here, the blocking hyperplane is the bold line in the quasi diagonal direction on the picture. The search points are indicated by bullets. In this example, the search direction is the negative gradient, thus, the cutting planes $H_{\nabla f(x_k)}$ are orthogonal to the search direction. In our numerical experiments, the threshold for the distance from a hyperplane was $\delta = 10^{-6}$.

Note that for ease of presentation we have defined DDMCP for bound constraint op-

timization, but it works for convex-constrained problems as well, as we will show via the GARCH example in Chapter 3.4.3. When convex constraints are applied, S is defined by a polyhedron, i.e. by finitely many hyperplanes. The DDMCP algorithm itself does not need to be changed for convex constrained problems. An extension for unconstrained problems can also be considered. In this case, DDMCP needs to be changed, so that a hyperplane from the initial polyhedron $\mathbb{H}_0$ acting as a blocking hyperplane has to be replaced by a new hyperplane with the same normal vector but further away from the current search point.

An important advantage of the proposed method DDMCP over descent methods is that DDMCP does not include any line search steps, and, as a direct consequence, it does not evaluate the objective. The price to pay for this improvement is that a growing set of cutting planes need to be stored and processed. This cost should be insignificant for low and medium scale problems though. As the numerical experiments will show, the performance improvement is high especially when the cost of a function evaluation is high. An other consequence of the missing function evaluations is that DDMCP is more robust against ill-behaved objective functions, such as for example noisy objectives with many local minima. This issue will be elaborated in Section 3.4.3 on the problem of GARCH identification.

As DDMCP can also be viewed as a centering or localization method, it is reasonable to compare it with a well-established member of this family, the ACCPM. One important difference between ACCPM and DDMCP relies in the implementation of the centering idea, which is to find some center of the encapsulating polyhedron $\pi(\bar{\mathbb{H}})$ defined by the set of m half-spaces with linear inequalities

$$p_i^T(y - o_i) \leq 0, \quad i = 1 \dots m.$$

As described in the introduction, in ACCPM, redundant hyperplanes may arise. In contrast, DDMCP automatically neglects redundant cutting planes, it considers only the boundary of $\pi(\bar{\mathbb{H}})$ when determining the new center on the line $x_k + \alpha\Delta x_k$, $\alpha > 0$ and DDMCP does not require any nontrivial optimization procedure to calculate it. Numerical experiments have shown that DDMCP has a faster convergence rate than ACCPM, and DDMCP actually inherits the fast convergence rate from the corresponding descent method. An advantage of ACCPM is that it does not require any initial point.

Figure 3.3 shows the search paths for test function (3.7) generated by the three different algorithms: Newton method damped with backtracking line search, ACCPM and DDMCP with search directions calculated exactly as in the Newton method. The initial point is

$x_0 = (-2, 3)^T$, the minimizer $x^* = (1, 1)^T$. The normalized vector field defined by the search direction is drawn in the background. The initial polyhedron is $[-3, 3] \times [-3, 3]$. One can realize in this example, that, in the first few iterations, the descent method approaches x^* slowly, meaning that $\|x_k - x^*\|$ decreases slowly as k increases. However, if it gets close enough to x^*, it converges very fast, it actually jumps very close to the minimizer. In contrast, ACCPM approaches x^* in early iterations faster, but gets slow in the neighborhood of the minimizer. DDMCP is fast in both convergence phases.

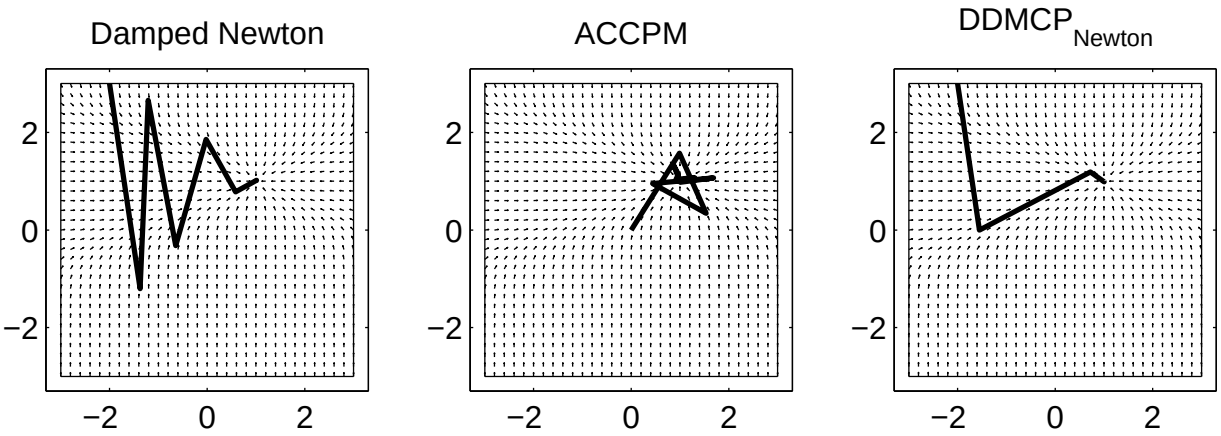

Figure 3.3: Search paths of the different algorithms

3.3 Theoretical aspects

3.3.1 Convergence of descent methods

In this chapter we compare the proposed method with popular descent algorithms often applied in practice for the particular minimization problem type. In all of these methods the descent direction Δx is the solution of

$$B_k \Delta x_k = -\nabla f(x_k), \tag{3.3}$$

where B_k is an $n \times n$ matrix, the concrete form of which is part of the specification of the particular algorithm. The role of B_k is to allow to determine a search direction that takes a local second order approximation of f in x_k into account. It is either the Hessian matrix

$$B_k = \nabla^2 f(x_k), \tag{3.4}$$

or some approximation of it, which we will see in more detail in the next sections.

The descent algorithms are equipped with a simple and efficient line search algorithm, called Armijo backtracking line search, see Armijo (1966). It is based on the Armijo rule ensuring a sufficient decrease of the objective:

$$f(x_k + t\Delta x_k) - f(x_k) < \alpha t \nabla f(x_k)^T \Delta x_k, \tag{3.5}$$

where α is a parameter, typically $\alpha = 10^{-4}$.

The backtracking line search algorithm is given by Algorithm 4, in that t is halved until the candidate search point satisfies the bound constraints *and* (3.5). Theorem 1 gives conditions for the convergence of the overall descent method. The theorem is a straightforward reformulation of Theorem 3.2.4 in Kelley (1999). See Shi (2004) for the convergence analysis of different line search algorithms.

Initialization: $t = 1$
repeat
  $t = t/2$
until $x_k + t\Delta x_k \in S$ and $f(x_k + t\Delta x_k) - f(x_k) < \alpha t \nabla f(x_k)^T \Delta x_k$

Algorithm 4: Armijo line search

Theorem 1. *Let f be bounded from below in S and ∇f be Lipschitz continuous, that is, there exists a positive constant $L \in \mathbb{R}$, such that*

$$\|\nabla f(x) - \nabla f(y)\| \leq L \|x - y\|, \quad \forall x, y \in S.$$

Assume that the matrices B_k are symmetric and positive definite (spd) and there exist constants $\bar{\kappa}$ and λ such that for the condition number $\kappa(B_k) \leq \bar{\kappa}$, and $\|B_k\| = \max_{x \neq 0} \frac{\|B_k x\|}{x} \leq \lambda$ for all k. Then,

$$\lim_{k \to \infty} \nabla f(x_k) = 0.$$

The positive definiteness of B_k is a crucial condition. It ensures the search direction Δx_k to be a descent direction, that is, Δx_k points in a direction where $f(x_k)$ is locally decreasing.

We have explicitly chosen problem types, where B_k is spd by construction. For other problem types a regularization technique is usually applied, that is, after determining B_k, a μI tag is added to it, where I denotes the identity matrix and μ is some appropriately chosen positive real number. This kind of regularization can also ensure to satisfy the other

conditions (uniform boundedness, being well-conditioned) in Theorem 1. Regularization is out of the scope of this work, see for example Kelley (1999), Boyd and Vandenberghe (2004), Fletcher (1980) for details, or Polyak (2007) for a recent result.

3.3.2 Remark on the convergence of DDMCP

The global convergence of descent methods relies on the sufficient decrease of the objective, which is ensured by choosing an appropriate step size in the line search procedure. The convergence of ACCPM relies on the advantageous theoretical properties of the analytic center Ye (1997). DDMCP neither ensures sufficient decrease nor uses analytic centers, thus, we cannot apply any of the two established concepts to prove convergence. In fact, the proof of convergence remains a subject of future work.

In order to keep the fast numerical performance of DDMCP, while enforcing convergence, we can consider combining DDMCP with a convergent descent method as spacer steps. According to the Spacer Step Theorem in Luenberger (1973) Chapter 7.9 the following holds. Given a convergent algorithm A and an algorithm B, which does not increase the objective, we can compose a new algorithm C, in which in some iterations A is applied, in the remaining iterations B. Then, if A is applied infinitely often, then C is also convergent.

Notice that DDMCP, while aspiring to approach x^*, at the same time may also increase the objective, although it does decrease it usually. Having a line search algorithm that decreases the objective function sufficiently, as for example the Armijo backtracking line search, we can apply the following simple alternating technique to ensure convergence by the Spacer Step Theorem (although we did not apply it).

Let N be a positive integer. If DDMCP actually decreased the objective (not necessarily sufficiently), then, inserting a sufficiently decreasing line search algorithm at every lN ($l \in \mathbb{N}$) iteration would ensure convergence of the whole x_k series. To get rid of the problem of non-decreasing DDMCP steps we just have to keep track of the value of the objective function at steps $(l-1)N$ and $lN-1$, and, before making a sufficiently decreasing line search step, ensure non-increase by $x_{lN-1} = x_{(l-1)N}$ if necessary. In the bad case, the last DDMCP steps are rolled back and we lose the progress of the last $N-1$ iterations. In this case we may set $N = \max(N/2, 1)$ expressing that we are loosing faith in DDMCP. Then, in the worst case, if $N = 2^u$ ($u \in \mathbb{N}$) we may have $\sum_{i=1}^{u} 2^i - 1 = 2^{u+1} - u - 1$ iterations that are rolled back.

3.4 Numerical performance evaluation

3.4.1 The convex case

In this section we deal with the convex case, that is, when the function

$$f : S \to \mathbb{R}$$

is convex.

For this type of functions, usually the Newton method damped with Armijo backtracking line search is used, which we denote by Newton-bLS. Therefore, we have compared our DDMCP method with Newton-bLS, such that Newton search direction is used in DDMCP. Our algorithm will be denoted by $\text{DDMCP}_{\text{Newton}}$ in this case.

In the Newton method the matrix B_k in (3.3) is the Hessian matrix, which is always spd in the convex case. Convexity has another useful technical implication: a cutting hyperplane with the gradient as its normal vector does not cut out the minimum point, i.e.

$$\nabla f(x)^T (x - x^*) < 0, \quad \forall x \in S.$$

Note, that despite of this property, we still need to drop hyperplanes too close to the searching point x_k as described in Step 2 in the DDMCP algorithm, because blocking hyperplanes as shown in Figure 3.2 can occur even in the convex case.

For completeness, we present numerical results for the ACCPM too. For simplicity, we installed a trivial oracle answering 'yes' if

$$\|x_k - x^*\| < \varepsilon, \tag{3.6}$$

where ε is the tolerance level. Note that, for our performance comparison purposes, it is sufficient to choose this theoretical oracle already knowing the minimizer. If (3.6) is not satisfied, the oracle returns a cutting plane with $\nabla f(x_k)$ as its normal vector.

Numerical results. For the numerical experiments we used the following two convex test functions:

$$f_1(x, y) = \log\left(1 + e^x\right) - \frac{1}{2}x + \frac{1}{2}\log\left(1 + e^{2y}\right) - \frac{1}{2}y, \tag{3.7}$$

$$f_2(x, y) = (x - \arctan x)\,\operatorname{sign} x + \left(y - \frac{1}{4}\arctan 4y\right)\operatorname{sign} y. \tag{3.8}$$

The form of the functions along the two dimensions is the same, the y coordinate is transformed affinely by factors 2 and 4 respectively. Apart from the affine constants, the partial derivatives for f_1 and f_2 have the form

$$\frac{e^z}{1+e^z}, \quad \frac{z^2}{1+z^2}\operatorname{sign} z,$$

respectively, where z stands for x or y.

The rationale behind these functions is that the (undamped) Newton method diverges if the starting point is far away from the minimum point $(0,0)^T$, thus, damping line search steps are needed for convergence.

Using both of these test functions we generated four test problems, each corresponding to different rows in Table 3.1, the first four to function type (3.7), the second four to function type (3.8). For each problem we defined a different minimum point $c = (c_1, c_2)^T$ as seen in the first two columns and implemented it simply by shifting the appropriate test function as

$$f_i^c(x,y) = f_i(x - c_1, y - c_2), \quad i \in \{1,2\}\,.$$

Thus, a test problem is defined by the function type and the minimum point c. A minimization run was started from 100 different initial points by the Newton-bLS and $\text{DDMCP}_{\text{Newton}}$ for each test problem, where the initial points were placed equidistantly on a circle line around the origin. In each run the number of function evaluations, number of gradient and Hessian evaluations were noted together with an evaluation index built from them as

$$e = \frac{n(n+1)}{2}e_H + ne_g + e_f, \tag{3.9}$$

where e_g, e_H, and e_f stand for the number of gradient, Hessian and function evaluations, respectively, and n is the problem dimension ($n = 2$ in these cases). We calculated the average from the 100 measurements of e_H, e_f, e_g, and e. The performance of ACCPM and $\text{DDMCP}_{\text{Newton}}$ is given as a ratio of the corresponding evaluation index relative to the evaluation index of Newton-bLS. We omitted the metric e_f for $\text{DDMCP}_{\text{Newton}}$ and ACCPM, e_H for ACCPM, because they were always zero for these algorithms. We ran ACCPM only once for each test problem, as it does not need any initial point. The radius r of the circle and also the size parameter d of the initial polyhedron $d[-1,1]^2$ defined by $\mathbb{H}_0$ was also chosen differently depending on c. The termination tolerance was $\varepsilon = 10^{-6}$ for all three methods.

We present the results of the numerical performance comparison in Table 3.1. According to the evaluation index e, $\text{DDMCP}_{\text{Newton}}$ is the most efficient and ACCPM is the least efficient algorithm in most runs. $\text{DDMCP}_{\text{Newton}}$ needed, on average, even less iterations (gradient and Hessian evaluations) than Newton-bLS, confirming the viability of the centering approach. We listed time measurement results as well. The measurements were taken on a 2.8 GHz machine with an Intel Pentium 4 CPU with 2 GB RAM using Matlab for all methods. For algorithm Newton-bLS, in column $\bar{t}_0$ average execution times in milliseconds can be seen. For the other two algorithms we present execution times relative to t_0. We can conclude that also in terms of execution time $\text{DDMCP}_{\text{Newton}}$ is the fastest algorithm out of those tested. ACCPM, on average, needed approximately two orders of magnitude more time than Newton-bLS, while in terms of the evaluation index, it was only slightly worse. This discrepancy can be explained by the fact that the evaluation index does not take into account the effort needed to find the analytic center as described in Step 1 of the ACCPM algorithm.

Table 3.1: Convex minimization: computational effort and execution time comparison

				Newton-bLS				ACCPM			$\text{DDMCP}_{\text{Newton}}$		
c		r	d	$\bar{e}_f$	$\bar{e}_g = \bar{e}_H$	$\bar{e}_0$	$\bar{t}_0$	e_g	$\frac{e}{e_0}$	$\frac{t}{t_0}$	$\bar{e}_g = \bar{e}_H$	$\frac{\bar{e}}{\bar{e}_0}$	$\frac{t}{t_0}$
1	1	2	4	8.02	5.68	36.4	1.56	61	3.35	300.00	4.89	0.67	0.40
2	2	4	8	15.60	6.61	48.6	2.49	56	2.30	144.18	5.87	0.60	0.57
3	3	6	12	24.30	7.26	60.6	3.59	87	2.87	287.19	6.64	0.55	0.40
4	4	8	16	33.31	7.41	70.4	4.84	78	2.22	158.06	7.13	0.51	0.23
Average				20.31	6.74	54.0	3.12	70.5	2.69	222.36	6.13	0.58	0.40
3	3	5	10	38.27	22.28	149.7	7.81	86	1.15	135.98	21.13	0.71	0.48
6	6	10	20	50.12	23.75	168.9	9.65	69	0.82	59.90	22.74	0.67	0.39
9	9	15	30	53.24	24.05	173.5	9.08	95	1.10	142.84	23.80	0.69	0.55
12	12	20	40	61.82	24.59	184.8	9.87	103	1.11	147.21	24.50	0.66	0.54
Average				50.86	23.67	169.2	9.10	88.3	1.04	121.48	23.04	0.68	0.49

We also examined the sensitivity of the performance of $\text{DDMCP}_{\text{Newton}}$ to the size of the initial polyhedron. We fixed $c = (0.5, 0.5)^T$, $r = 2$ and by increasing d we measured the average number of iterations for both test function types. Results for test functions f_1 and f_2 are depicted on Figure 3.4 on the left and on the right, respectively. We can conclude that even for large initial hypercube sizes, $\text{DDMCP}_{\text{Newton}}$ needs only a few iterations more than in the case of a relatively small initial hypercube.

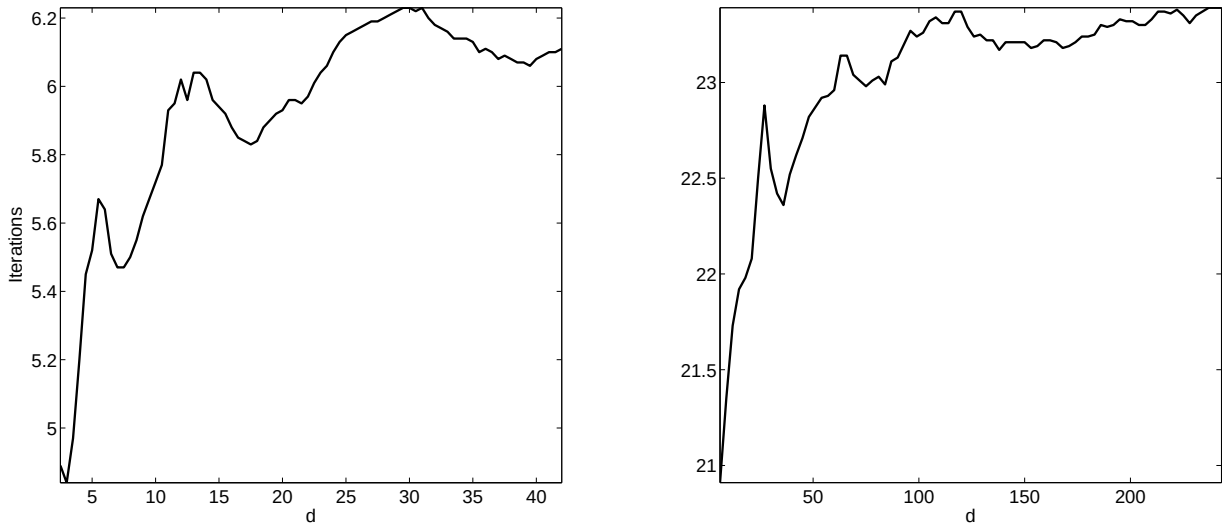

Figure 3.4: Average iteration number necessary vs. size of initial hypercube

3.4.2 A nonconvex case: nonlinear least squares

We consider the nonlinear least-squares problem

$$\min_{x \in S} f(x), \quad f(x) = \frac{1}{2} \sum_{i=1}^{m} r_i^2(x), \quad m > n, \tag{3.10}$$

where each component $r_i : \mathbb{R}^n \to \mathbb{R}$ of the residual vector $r(x)$ is a twice continuously differentiable function. Let $J(x)$ be the Jacobian matrix of $r(x)$. A widely used numerical algorithm to solve (3.10) is the Gauss-Newton algorithm with backtracking Armijo line search Kelley (1999), which is considered to be very effective in practice. We denote this algorithm by GN-bLS. In this method B_k in the equation system (3.3) is

$$B_k = J(x_k)^T J(x_k), \tag{3.11}$$

and the gradient can be computed as

$$\nabla f(x) = J(x)^T r(x).$$

We denote our method using the same descent direction as DDMCP_{GN}.

Numerical results. Table 3.2 lists the nonlinear least squares test problems together with the performance results. The problems are suggested in Moré et al. (1981), though we restrict ourself to test functions with finitely many global minimizers. The initial

parameter points are also defined in Moré et al. (1981). The termination tolerance on the length of the search direction was $\varepsilon = 10^{-6}$. The number in the first column of the table is the problem identifier from Moré et al. (1981), n is the dimension of the problem, m denotes the number of squared residuals to sum. In the fifth column we listed the bound constraints for f, which was a hypercube defining the initial set $\mathbb{H}_0$. At some of the test functions, B_k was ill-conditioned on the negative cone $[-\infty, 0]^n$, thus, without regularization, no search direction was available. As the minimum point x^* resides in the positive cone for these functions and regularization is out of the scope of this work, we got rid of this issue by simply restricting the domain onto the positive cone. Note that one popular regularization method is the *Levenberg-Marquardt* algorithm, see Kelley (1999).

Table 3.2: Least squares: computational effort comparison

	Function				GN-bLS			DDMCP_{GN}	
Id	name	n	m	$\mathbb{H}_0$	e_f	e_g	e_0	e_g	$\frac{e}{e_0}$
32	Linear	10	20	$5[-1,1]^n$	4	3	34	2	0.59
1	Rosenbrock	2	2	$3[-1,1]^n$	42	12	66	4	0.12
7	Helical valley	3	3	$2[-1,1]^n$	18	10	48	26	1.63
13	Powell singular	4	4	$5[-1,1]^n$	44	23	136	22	0.65
8	Bard	3	15	$3[-1,1]^n$	12	7	33	6	0.55
15	Kowalik & Osborne	4	11	$[0,1]^n$	60	31	184	20	0.43
10	Meyer	3	16	$7000[-1,1]^n$	23	11	56	8	0.43
20	Watson	9	31	$8[-1,1]^n$	10	6	64	5	0.70
12	Box	3	10	$15[-1,1]^n$	12	7	33	6	0.55
35	Chebyquad	5	5	$[0,1]^n$	11	6	41	5	0.61
17	Osborne 1	5	33	$6[-1,1]^n$	18	9	63	7	0.56
19	Osborne 2	11	65	$8[0,1]^n$	27	13	170	13	0.84
Average					23.4	11.5	77.3	10.3	0.64

We used the same performance metrics as given in (3.9). Since no Hessian was evaluated, e_H was always zero. The results are summarized in Table 3.2. Both GN-bLS and DDMCP_{GN} have found the same minima for all problems (up to four decimal digits). On problems 'Freudenstein & Roth', 'Jennrich & Sampson', and 'Brown & Dennis' neither of the algorithms converged, because of ill-conditioned Gauss-Newton matrices B_k, so we have excluded them from the table. As seen on the results, our algorithm performs better than GN-bLS on most of the problems, except problem 'Helical valley', where GN-bLS was better. Without this problem, even the number of iterations (gradient evaluations) was less than or equal to the one for GN-bLS, confirming that the centering approach is quite efficient.

Table 3.3: Least squares: computational effort comparison with increased size of $\mathbb{H}_0$

	Function	$\mathbb{H}_0$	GN-bLS			DDMCP_{GN}	
Id	name	$d \in \{10, 10^2, 10^3\}$	e_f	e_g	e_0	e_g	$\frac{e}{e_0}$
32	Linear	$5d[-1,1]^n$	4	3	34	2	0.59
1	Rosenbrock	$3d[-1,1]^n$	44	12	68	3	0.09
7	Helical valley	$2d[-1,1]^n$	21	11	54	30	1.67
13	Powell singular	$5d[-1,1]^n$	44	23	136	22	0.65
8	Bard	$3d[-1,1]^n$	12	7	33	6	0.55
15	Kowalik & Osborne	$d[0,1]^n$	62	32	190	20	0.42
20	Watson	$8d[-1,1]^n$	10	6	64	5	0.70
12	Box	$15d[-1,1]^n$	12	7	33	6	0.55
35	Chebyquad	$d[0,1]^n$	11	6	41	5	0.61
17	Osborne 1	$6d[-1,1]^n$	18	9	63	7	0.56
19	Osborne 2	$8d[0,1]^n$	27	13	170	13	0.84
Average			24.1	11.7	80.5	10.8	0.66

Table 3.3 summarizes results of comparison tests, where we multiplied the side of $\mathbb{H}_0$ by 10^1, 10^2, 10^3. All three sizes gave the same results in terms of the evaluation index, thus, $d > 10$ did not require more computational effort. With extended bounds, we have got similar results as in Table 3.2, with the exception of test problem 'Meyer'. For this test function, matrix B_k was often very close to singular resulting in unusable search directions, so neither of the methods converged. The problem could be solved again by regularization, which we do not consider in this work, thus it has been excluded from this study. Table 3.4 contains time measurement results for different sizes of the bounding boxes. The execution times in case $d \in \{10^1, 10^2, 10^3\}$ were almost the same for all three initial hypercube sizes, so we listed time results only once.

Let us remark that we have also tested Gauss-Newton with a polynomial line search algorithm (see Kelley (1999), Chapter 3.2.1), but in general, the performance results with polynomial line search were even worse than with backtracking line search.

3.4.3 Maximum likelihood estimation of GARCH parameters

In this section we demonstrate the viability of our method on the important problem of estimating the parameters of GARCH models. GARCH models are usually identified by maximum likelihood estimation, see Berkes et al. (2003); Bollerslev and Wooldridge (1992) for results regarding the consistency and asymptotic normality of the estimator. We consider in this chapter the off-line GARCH parameter estimation problem, when the

Table 3.4: Least squares: execution time comparison. The index E indicates the cases of extended $\mathbb{H}_0$'s as given in Table 3.3.

	Function	GN-bLS		DDMCP_{GN}	
Id	name	t_0 (ms)	t_0^E (ms)	$\frac{t}{t_0}$	$\frac{t^E}{t_0^E}$
32	Linear	3.1	2.8	0.2	0.3
1	Rosenbrock	9.8	9.4	0.1	0.2
7	Helical valley	5.5	6.3	2.0	2.0
13	Powell singular	12.2	12.6	0.7	0.7
8	Bard	5.5	5.8	0.4	0.4
15	Kowalik & Osborne	19.4	21.0	0.5	0.4
10	Meyer	8.1	-	0.5	-
20	Watson	10.0	8.1	0.5	0.8
12	Box	4.6	5.0	0.6	0.4
35	Chebyquad	3.3	4.4	1.2	0.6
17	Osborne 1	11.9	11.4	0.6	0.5
19	Osborne 2	35.6	34.2	0.6	0.6
Average		10.7	11.0	0.7	0.6

estimation is carried out after all the observation data is available.

Next, we explain the connection between parameter estimation and nonlinear optimization, and we present a popular optimization method for this problem type. Then we shortly describe the GARCH model along with a simple technique introduced here to reduce the dimension of the optimization problem.

Consider the estimation problem regarding to the univariate stochastic model in form of

$$F_t\left(y_t, \vartheta^*\right) = \varepsilon_t, \quad t \in \mathbb{Z},$$

where $y_t \in \mathbb{R}$ are observations, F_t is a twice-differentiable scalar-valued function, ε_t are independent identically distributed (i. i. d.) normal disturbances and $\vartheta^* \in S \subset \mathbb{R}^d$ is the parameter vector to be estimated, or identified, according to the system theoretic terminology. A popular way of identifying such models is the maximum likelihood estimation, which means the minimization of the negative log-likelihood function $L_T(\vartheta)$, i.e.

$$\hat{\vartheta} = \arg\min_{\vartheta} L_T(\vartheta), \tag{3.12}$$

given the observations $y_1, y_2, \ldots y_T$, where $\hat{\vartheta}$ denotes the estimator, that is, the minimizer point we are searching for. Thus, with notations used throughout this chapter, (3.12) can

be formulated as an optimization problem with objective

$$f(x) = L_T(\vartheta)|_{\vartheta=x}, \quad x \in S.$$

In what follows, we consider only log-likelihood functions, which can be written as

$$L_T(\vartheta) = \sum_{t=1}^{T} l_t(y_t, \vartheta), \tag{3.13}$$

where l_t can be calculated from the joint density function of $(y_1, y_2, \ldots, y_t)$. A widely used algorithm for minimizing a likelihood function with property (3.13) is the BHHH method Berndt et al. (1974) with backtracking Armijo line-search, denoted by BHHH-bLS. The BHHH algorithm specifies matrix B_k in Algorithm 1 as the outer product matrix built from partial derivatives

$$B_k = \sum_{t=1}^{T} \frac{\partial l_t(y_t, \vartheta_k)}{\partial \vartheta} \left(\frac{\partial l_t(y_t, \vartheta_k)}{\partial \vartheta} \right)^T, \tag{3.14}$$

where $\vartheta_k = x_k$. Notice that B_k is asymptotically spd and only the gradients are needed to calculate it, which makes it very useful in practice. We denote our algorithm applying matrices as defined in (3.14) by $\text{DDMCP}_{\text{BHHH}}$.

We consider GARCH(p, q) models given as

$$y_t = \sqrt{h_t}\varepsilon_t, \quad \varepsilon_t \sim \mathcal{N}(0,1), \quad \text{i. i. d.} \tag{3.15}$$

$$h_t = \sigma_0 + \sum_{i=1}^{q} \varphi_i y_{t-i}^2 + \sum_{j=1}^{p} \psi_i h_{t-j}, \tag{3.16}$$

where the real parameter vector to be estimated is

$$(\sigma_0, \varphi_1, \ldots, \varphi_q, \psi_1, \ldots, \psi_p)^T, \tag{3.17}$$

with positive components. We also apply the stationarity condition given in Berkes et al. (2003); Bollerslev (1986) and restrict the parameter domain as

$$\sum_{i=1}^{q} \varphi_i + \sum_{j=1}^{p} \psi_i < 1. \tag{3.18}$$

In a GARCH(p, q) model there are (p+q+1) parameters to estimate. Now we introduce a technique, with which we can reduce the dimension of the optimization problem to (p+q). Let's first denote by σ^2 the unconditional variance which is the expected value of the conditional variance h_t:

$$\sigma^2 = E(h_t), \quad \forall t. \tag{3.19}$$

Under stationarity we can calculate σ^2 from the model parameters as follows. Taking expectation on both sides of (3.16) we can write

$$E(h_t) = \sigma_0 + E(h_t) \sum_{i=1}^{q} \varphi_i + E(h_t) \sum_{j=1}^{p} \psi_i,$$

where we used $E(y_t^2|h_t) = h_t$ and stationarity. Using (3.19), it immediately follows that

$$\sigma^2 = \frac{\sigma_0}{1 - \sum_{i=1}^{q} \varphi_i - \sum_{j=1}^{p} \psi_i}. \tag{3.20}$$

Assume that we can estimate σ^2, let us denote the estimator by $\hat{\sigma}^2$. Then we can use (3.20) to express σ_0 and eliminate it from the parameter vector. Doing so, we arrive at the following modified equation for the latent variance process, which only contains (p+q) unknown variables:

$$\bar{h}_t = \hat{\sigma}^2 \left(1 - \sum_{i=1}^{q} \varphi_i - \sum_{j=1}^{p} \psi_i \right) + \sum_{i=1}^{q} \varphi_i y_{t-i}^2 + \sum_{j=1}^{p} \psi_i \bar{h}_{t-j}. \tag{3.21}$$

For the numerical experiments we need to define the initial values of $\bar{h}_t$ and y_t. We assume that the initial values of h_t and y_t are equal to the unconditional variance σ^2, i.e.

$$\bar{h}_t = y_t^2 = \sigma^2, \quad \forall t \leq 0. \tag{3.22}$$

In our experiments, in order to save iterations in the numerical optimization procedure, we applied this dimension reduction technique, that is, we first estimated σ^2 using

$$\hat{\sigma}^2 = \frac{1}{T} \sum_{t=1}^{T} y_t^2, \tag{3.23}$$

then we took the model in the form of (3.21) to estimate the rest of the parameter vector

$$\vartheta = (\varphi_1, \ldots, \varphi_q, \psi_1, \ldots, \psi_p)^T.$$

Note, that the estimator resulting from this dimension reducing scheme can deviate from the estimator we would get by maximizing the original likelihood explicitly including σ_0.

For completeness, we next specify l_t and its derivatives, which are used to calculate B_k. The negative log-likelihood of the form (3.13), is calculated from

$$l_t(y_t, \vartheta) = \frac{1}{2}\left(\log \bar{h}_t + \frac{y_t^2}{\bar{h}_t}\right), \tag{3.24}$$

where some constants are neglected because they do not play any role in the optimization. Let us now calculate the score function. Differentiating (3.24) yields

$$\frac{\partial l_t}{\partial \vartheta} = \frac{1}{2}\frac{1}{\bar{h}_t}\frac{\partial \bar{h}_t}{\partial \vartheta}\left[\frac{y_t^2}{\bar{h}_t} - 1\right],$$

where the partial derivatives of h_t can be calculated recursively from (3.21) and (3.22) as

$$\frac{\partial \bar{h}_t}{\partial \vartheta} = \begin{cases} \left(y_{t-1}^2, \ldots y_{t-q}^2, \bar{h}_{t-1}, \ldots \bar{h}_{t-p}\right)^T - \hat{\sigma}^2 + \sum_{i=1}^{p} \psi_i \frac{\partial \bar{h}_{t-i}}{\partial \vartheta}, & t > 0 \\ 0, & t \leq 0. \end{cases} \tag{3.25}$$

Note that a Hessian approximation based on (2.33) in Chapter 2 Section 2.5 results a matrix formally very similar to what we get by (3.14), the difference is a scalar factor only. Numerical tests have shown that both formulas yield essentially the same search directions, except for some small intervalls of ϑ, when ϑ^* is close to the boundary of the search domain, where the adjusted version yields better directions. For this reason we adjust (3.14) as follows:

$$B_k = \sum_{t=1}^{T} c_t \cdot \partial_\vartheta \bar{h}_t(\vartheta) \left(\partial_\vartheta \bar{h}_t(\vartheta)\right)^T, \quad c_t = \bar{h}_t^{-2}(\vartheta)\left(\frac{2y_t^2}{\bar{h}_t(\vartheta)} - 1\right).$$

Because of the stochastic nature of the problem, the minimum point x^* of the negative likelihood can fall onto the boundary of the parameter space defined in (3.17) and (3.18). For simplicity, we handled this problem by modifying the termination criteria both for $\text{DDMCP}_{\text{BHHH}}$ and the BHHH-bLS, as the given algorithm also stops when the search point x gets closer to the boundary of the parameter space than a predefined tolerance.

Numerical results. For our numerical experiments regarding the GARCH identification we have taken the simplest model class with $p = q = 1$. The low dimension allowed us to sufficiently cover the parameter space when choosing the true parameter for the model to be identified. The parameter space S can be defined by the following linear inequalities:

$$\mathbb{H}_0 = \left\{ \begin{array}{rcl} (-1,0)y & < & 0, \\ (0,-1)y & < & 0, \\ (1,1)y & < & 1 \end{array} \right\},$$

where $y \in \mathbb{R}^2$.

For each parameter pair $(\varphi_1^*, \psi_1^*)^T$, we generated $T = 10^5$ observations by simulation and estimated the parameters, with an initial value $x_0 = (0.2, 0.2)^T$. For simplicity, the data was generated with a fixed $\sigma_0^* = 1$. In the estimation procedure we first applied the dimension reduction technique and used the optimization methods $\text{DDMCP}_{\text{BHHH}}$ and BHHH-bLS to find the best estimate for the remaining parameter vector $(\varphi_1^*, \psi_1^*)^T$. We use the term 'estimation run' to indicate the two-stage procedure consisting of generating observations then estimating the parameter vector. For each parametrization we made 5 estimation runs. In all of the methods, the termination tolerance was $\varepsilon = 10^{-6}$ both on the length of search direction Δx and on the distance from boundary. We have set the parameter $\alpha = 10^{-8}$ in the Armijo condition (3.5).

For the method BHHH-bLS we modified the line search slightly: we allow at most 6 search steps only. This cap was necessary, because otherwise in some cases we experienced very long line search cycles. These were the consequence of frequent consecutive local minima and maxima of the noisy log-likelihood function along the search direction, causing condition (3.5) to be satisfied for only very small step sizes. A very small step size t made the algorithm actually stuck far from the optimum. The capped algorithm got rid of the possibly tiny steps and despite violating theoretical conditions of convergence, it has not caused any convergence problems in the numerical experiments. Figure 3.5 exhibits the situation of a line search failure at $x_k = \vartheta_k = (0.0525, 0.0822)^T$ on a model with $\vartheta^* = (0.05, 0.05)^T$. The Armijo reference line shows

$$f(x_k) + \alpha t \nabla f(x_k)^T \Delta x_k, \quad t > 0,$$

with tick marks on it indicating the places where the Armijo condition was tested by the algorithm. The Armijo condition is satisfied for a step size t, where the objective

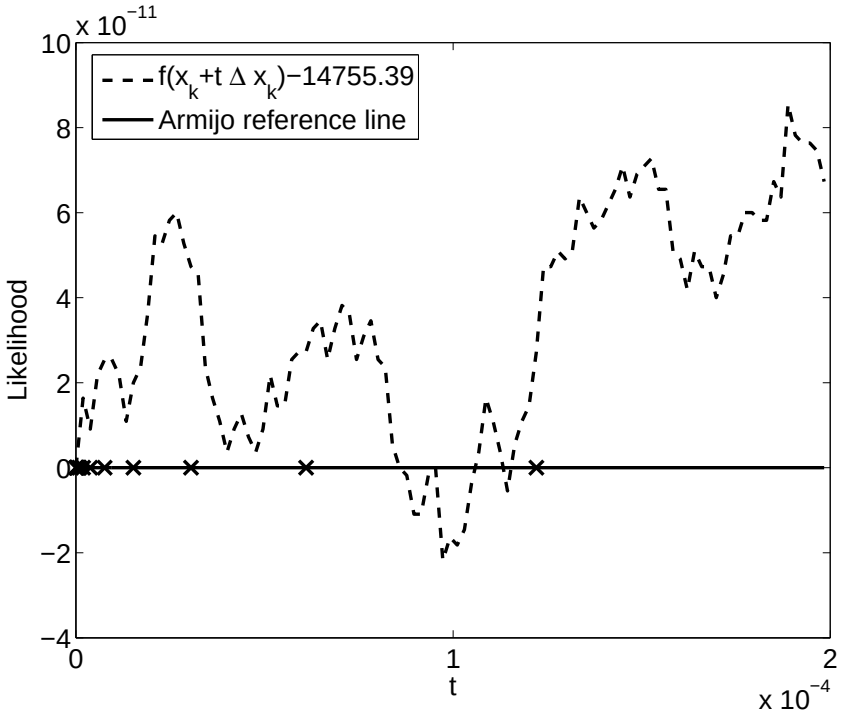

Figure 3.5: Backtracking line search fails on the noisy likelihood

$f(x_k + t\Delta x_k)$ is below this line, which is not the case for any of the step sizes marked on the line.

For an additional comparison, we have also tested the constrained nonlinear minimization method from the Optimization Toolbox in Matlab 7.0.0, see Crummey et al. (1991). This method is used by Matlab's state of the art GARCH Toolbox for model identification. We did not test the GARCH toolbox directly, in order to be able to apply the dimension reduction technique, to have the same problem dimensions for all algorithms under comparison. The Matlab method is a quite different algorithm than our $\text{DDMCP}_{\text{BHHH}}$ and BHHH-bLS. It is a sequential quadratic programming method using the BFGS formula for updating the Hessian approximation. An essential difference is that the BFGS formula updates B_k using B_{k-1}, while the two BHHH based methods do not use any derivative values from previous iterations for calculating B_k. Please consult Crummey et al. (1991) for details of the Matlab method.

The rows of Table 3.5 contain the performance results of the estimation runs. We listed the number of function evaluations, gradient evaluations and the evaluation index ratios as introduced in (3.9). In terms of average of the evaluation index, the proposed method is clearly better than the other two algorithms. In comparison to BHHH-bLS,

Table 3.5: GARCH parameter estimation: performance comparison

ψ_1^*	ψ_1^*	BHHH-bLS			Matlab			$\text{DDMCP}_{\text{BHHH}}$				$\hat{\Sigma}_\infty$	
		e_f	e_g	e_0	e_f	e_g	$\frac{e}{e_0}$	e_g	$\frac{e}{e_0}$	$\hat{\varphi}_1 - \varphi_1^*$	$\hat{\psi}_1 - \psi_1^*$		
		12	12	36	40	7	1.50	21	1.17	-0.0058	-0.05*	1.4	-2.4
		8	8	24	54	12	3.25	8	0.67	-0.0127	0.0589		
0.05	0.05	13	13	39	31	7	1.15	21	1.08	0.0085	−0.05*		
		18	17	52	55	12	1.52	17	0.65	-0.0114	0.0679	-2.4	412.9
		24	12	48	50	11	1.50	13	0.54	-0.0210	0.1669		
		8	8	24	50	9	2.83	7	0.58	-0.1276	0.0066	0.2	-0.2
		8	8	24	55	10	3.13	8	0.67	-0.0662	-0.0086		
0.90	0.05	8	8	24	55	10	3.13	7	0.58	-0.0921	0.0037		
		7	7	21	52	10	3.43	8	0.76	-0.0520	0.0026	-0.2	0.3
		8	8	24	55	10	3.13	7	0.58	-0.0762	-0.0018		
		8	8	24	11	2	0.63	8	0.67	-0.0006	0.0010*	0.3	-0.6
		9	9	27	61	12	3.15	9	0.67	0.0068	-0.0215		
0.05	0.90	11	9	29	56	10	2.62	10	0.69	0.0043	0.0002		
		7	7	21	59	11	3.86	8	0.76	0.0142	-0.0151	-0.6	1.9
		9	9	27	11	2	0.56	9	0.67	-0.0103	0.0176*		
		9	9	27	47	9	2.41	8	0.59	-0.0273	0.1236	1.8	-2.8
		9	8	25	35	6	1.88	7	0.56	0.0096	-0.0301		
0.15	0.15	13	12	37	44	9	1.68	11	0.59	-0.0131	-0.0307		
		9	9	27	39	7	1.96	8	0.59	0.0060	0.0253	-2.8	45.5
		6	6	18	37	7	2.83	5	0.56	0.0086	-0.0535		
		6	6	18	50	9	3.78	7	0.78	0.0106	-0.0097	0.8	-1.3
		6	6	18	48	9	3.67	6	0.67	0.0040	-0.0312		
0.15	0.75	9	7	23	52	9	3.04	7	0.61	-0.0236	0.0384		
		7	7	21	56	11	3.71	8	0.76	0.0088	-0.0025	-1.3	2.9
		7	7	21	52	10	3.43	9	0.86	0.0042	-0.0102		
		6	6	18	46	8	3.44	7	0.78	-0.0384	-0.0080	2.7	-3.1
		13	7	27	47	9	2.41	6	0.44	-0.0900	0.0079		
0.75	0.15	7	7	21	47	9	3.10	7	0.67	-0.0656	-0.0099		
		8	7	22	45	8	2.77	6	0.55	-0.0386	-0.0236	-3.1	3.7
		8	8	24	62	12	3.58	6	0.50	0.0246	0.0010		
		8	8	24	50	9	2.83	7	0.58	-0.0180	0.0127	1.8	-2.7
		8	8	24	37	7	2.13	7	0.58	-0.0206	-0.0128		
0.30	0.30	6	6	18	42	8	3.22	5	0.56	0.0395	-0.0275		
		7	6	19	42	8	3.05	5	0.53	0.0253	-0.0467	-2.7	10.0
		7	7	21	50	10	3.33	6	0.57	0.0176	-0.0062		
		9	9	27	53	11	2.78	8	0.59	0.0165	-0.0147	1.2	-1.7
		8	8	24	48	10	2.83	8	0.67	0.0212	0.0031		
0.30	0.60	15	9	33	53	11	2.27	6	0.36	-0.0101	0.0045		
		7	7	21	50	10	3.33	7	0.67	0.0173	-0.0043	-1.7	2.6
		9	8	25	57	12	3.24	7	0.56	-0.0215	0.0217		
		7	7	21	53	10	3.48	8	0.76	-0.0069	0.0126	1.3	-1.6
		9	9	27	47	9	2.41	7	0.52	0.0423	-0.0207		
0.60	0.30	8	8	24	47	9	2.71	8	0.67	0.0393	-0.0115		
		7	7	21	50	11	3.43	8	0.76	-0.0165	0.0234	-1.6	2.1
		9	8	25	44	9	2.48	7	0.56	-0.0150	-0.0171		
Average		8.27	7.55	23.4	43.4	8.39	2.50	7.61	0.60	-0.01	0.00		

40% of the computational effort is saved by our algorithm on average. In almost all of the cases BHHH-bLS achieved better performance than the general Matlab optimization routine. Table 3.6 shows the average duration time needed by a BHHH-bLS estimation for each parametrization. We list for the Matlab and $\text{DDMCP}_{\text{BHHH}}$ duration ratios relative to BHHH-bLS. On average, $\text{DDMCP}_{\text{BHHH}}$ is approximately two times faster than BHHH-bLS and 4.5 times faster than Matlab. Running the Matlab rutine without dimension reduction, our algorithm is approximatly an order of magnitude faster. Note that in case of the GARCH model, function evaluations are especially costly, because evaluating the log-likelihood contains enormous number of logarithm calculations.

Table 3.5 also shows the estimated parameters $\hat{\varphi}_1$, $\hat{\psi}_1$ of our algorithm, which matched the results of the other two algorithms with a precision of 10^{-4} in most of the runs. Deviations are listed in Table 3.7, the corresponding rows in Table 3.5 are marked with asterisk at the column $\hat{\psi}_1 - \psi_1^*$. Deviations arose near the boundary of the parameter space. In the case of $\vartheta^* = (0.05, 0.05)^T$, in two out of five estimation runs the algorithms stopped for getting closer to the boundary than the specified tolerance. In the case of $\vartheta^* = (0.05, 0.90)^T$, in two estimation runs, the Matlab routine stopped prematurely, after only two iterations at $\hat{\vartheta} = (0, 1)^T$. In contrast, BHHH-bLS and $\text{DDMCP}_{\text{BHHH}}$ have found the optimum near the true parameters.

As for the accuracy of the estimation, we can realize, that $\hat{\psi}_1$ varies considerably, when the true model parameters φ_1^*, ψ_1^* are both small, especially in case of $\vartheta^* = (0.05, 0.05)^T$. In order to examine this property we calculated an approximation of the asymptotic covariance matrix as

$$\hat{\Sigma}_\infty(\vartheta^*) = \hat{I}(\vartheta^*)^{-1} = \left(\frac{1}{T} \sum_{t=1}^{T} \frac{\partial l_t(\vartheta^*)}{\partial \vartheta} \left(\frac{\partial l_t(\vartheta^*)}{\partial \vartheta} \right)^T \right)^{-1},$$

where $T = 10^6$ and $\hat{I}$ denotes the approximated Fisher information matrix. We can realize that the asymptotic variance of $\hat{\psi}_1$ corresponding to the lower right corner value of $\hat{\Sigma}_\infty$ is high in the high variation cases, indicating that this is a model inherent property. Therefore, we can conclude that GARCH models with small coefficients are more difficult to identify.

3.5 Conclusion

In this chapter we have presented a hybrid algorithm DDMCP for bound constrained optimization. The main novelty over descent methods is the way of damping. Instead

Table 3.6: GARCH: average estimation time comparison

φ_1^*	ψ_1^*	$t_{\text{BHHH-bLS}}$ (sec)	$\frac{t_{\text{Matlab}}}{t_{\text{BHHH-bLS}}}$	$\frac{t_{\text{DDMCP}_{\text{BHHH}}}}{t_{\text{BHHH-bLS}}}$
0.05	0.05	0.4376	1.39	0.59
0.90	0.05	0.2596	2.56	0.48
0.05	0.90	0.2840	1.79	0.51
0.15	0.15	0.2940	1.79	0.44
0.15	0.75	0.2218	2.90	0.56
0.75	0.15	0.2408	2.57	0.48
0.30	0.30	0.2342	2.39	0.47
0.30	0.60	0.2838	2.38	0.43
0.60	0.30	0.2564	2.41	0.49
Average		0.2791	2.24	0.49

Table 3.7: Deviations in $\hat{\varphi}_1$, $\hat{\psi}_1$

φ_1^*	ψ_1^*	BHHH-bLS		Matlab		$\text{DDMCP}_{\text{BHHH}}$	
		$\hat{\varphi}_1$	$\hat{\psi}_1$	$\hat{\varphi}_1$	$\hat{\psi}_1$	$\hat{\varphi}_1$	$\hat{\psi}_1$
		0.0396	0.0000	0.0406	0.0000	0.0442	0.0000
0.05	0.05	0.0580	0.0000	0.0597	0.0000	0.0585	0.0000
		0.0494	0.9010	0.0000	1.0000	0.0494	0.9010
0.05	0.90	0.0397	0.9176	0.0000	1.0000	0.0397	0.9176

of line search, which ensures sufficient decrease of the objective, DDMCP uses separating hyperplanes to bound the step size in the search direction. Performance gain is achieved by inheriting the centering idea of localization methods in the damped phase, while preserving the rapid convergence rate of descent methods when the search point is sufficiently close to the minimizer. Furthermore, DDMCP does not evaluate the objective function.

We have shown empirically the efficiency of the method. We tested DDMCP for three problem types: convex nonlinear minimization, nonlinear least squares minimization and GARCH model identification. For all the three problem types, for the majority of the test problems, the performance of DDMCP was superior to the corresponding well established descent methods with backtracking line search. For each problem type, we examined the robustness of DDMCP from a different perspective.

In the case of convex nonlinear minimization, we compared DDMCP to the Newton method by starting it with different initial values and we examined how sensitive the performance is against the size of the initial polyhedron.

For the least squares problem type, we applied it against the Gauss-Newton method on a large set of different problems developed specifically for optimization test purposes.

In the GARCH case, the objective function exhibits numerous local minima and maxima around the global minimizer. We can conclude that DDMCP is not too sensitive to the initial value x_0 and the size of the initial polyhedron. DDMCP works well for the different test problems. The numerical results have also shown that DDMCP is quite robust against local minima and maxima in the noisy objective function, unlike the BHHH method with backtracking line search.

Bibliographic remarks

The paper based on the work presented in this chapter has been accepted for publication in the Central European Journal of Operations Research, see Torma and G.-Tóth (2010).

The algorithm and the test concept are based on the ideas of Balázs Torma, the numerical experiments were conducted by him as well. The work has been carried out with cooperation of Boglárka G.-Tóth.

Chapter 4

Modelling information arrival processes

4.1 Introduction

4.1.1 Context and motivation

Stochastic systems driven by point processes arise in many applications, see Daley and Vere-Jones (2003a) for a modern and comprehensive treatise. The present investigations were motivated by application areas in finance. First, as demonstrated in the description of the ACF model in Section 2.2, point processes are intuitive models for news arrival processes. Since the observable intensity of news arrival can be used as a proxy for price volatility, see Kalev et al. (2004), modelling the news arrival dynamics is an important subject of investigations. In addition, news arrival models are a widely unexplored research area. Second, stochastic systems driven partially by point processes are widely used in financial mathematics, in particular to study credit risk processes on bond markets. In this case credit events (defaults) results in jumps in the state of the system. Letting $p(t) = p(t, T)$ be the price of a zero-coupon T-bond at time $t \leq T$ the price dynamics, written in multiplicative form in terms of returns, is in its simplest form

$$dp(t) = p(t-)(\alpha dt + \sigma dW(t) + \delta dN(t)),$$

where $N(t)$ is a counting process, counting the number of credit events up to time t. By letting T vary, and using a suitable re-parametrization in terms of so-called forward rates we get the extension of the celebrated HJM (Heath-Jarrow-Morton) model. Good

references for bond markets with jumps are Giesecke et al. (2004); Runggaldier (2003).

In consequence of analysts' herding behavior, see Welch (2000), market news appearing on the market about a company can call analysts' attention on that company and thus make analysts generate more news. This intuition can be captured by a homogeneous self-exciting point process $N(t)$, also called Hawkes-processes, or Poisson cluster-process, see Hawkes (1971b,a), where $N(t)$ counts the news events up to time t. This is defined, in its simplest form, by the feedback system

$$dN(t) = \lambda(t)dt + dm(t) \tag{4.1}$$

$$d\lambda(t) = -a(\lambda(t) - m)dt + \sigma dN(t), \tag{4.2}$$

where $\lambda(t)$ is the intensity process, $dm(t)$ is a kind of white noise (a martingale differential), $a > 0$ takes care of mean-reversion, and $m > 0$ is a steady-state value of the intensity. Self-excitation is due to the fact that there is a positive correlation between the intensity and the event process: an increase in the number of events temporally increases the intensity of the event process. This is counter-balanced by a mean-reversion effect: the intensity would tend to return to its stationary value m if no events takes place.

The above jump-diffusion model extends naturally to multi-variable models. The identification (calibration) of these models, in particular the estimation of cross-effects is a hot area of research. This leads to a the problem of the estimation of the internal dynamics of the Hawkes process.

A recursive maximum-likelihood estimation will be developed and tested. The simulation of Hawkes processes itself is a non-trivial problem, see Moller and Rasmussen (2005), Moller and Rasmussen (2006). The weak point of these simulation methods is that they are inherently non-dynamic. An alternative simulation technique will also be presented in this chapter. Finally, the Fisher information matrix of the estimation problem will be investigated.

4.1.2 Point processes

This introduction is based on Runggaldier (2003) and Bremaud (1981). A point process is an increasing sequence of random times T_n defined over some probability space $(\Omega, \mathcal{F}, P)$ with $T_0 = 0$ and $T_n < T_{n+1}$ if $T_n < \infty$. A common assumption is that the process is non-explosive, meaning that $\lim T_n = \infty$. Alternatively, the associated counting process $N(t) := \#\{i : T_i \leq t\}$, counting the number of events up to time t, is also called a point

process. The internal history of a point process is defined by $\mathcal{F}^N(t) = \sigma\{N(s) : s \leq t\}$. In general, a filtration $\mathcal{F}(t)$ is called a history, if it is a refinement of $\mathcal{F}^N(t)$. An additional technical assumption is that $N(t)$ is integrable, i.e., $\mathrm{E}\,(N(t)) < +\infty$ for all t.

Homogeneous Poisson process. A prime example for a point process is the homogeneous Poisson process with constant, finite intensity λ: $N(t)$ adapted to $\mathcal{F}(t)$ is called homogeneous Poisson process if $N_0 = 0$, $N(t)$ has independent stationary increments and

$$N(t+h) - N(t) \sim Poisson(\lambda h), \quad h > 0.$$

A well-known property of Poisson processes is that

$$M(t) = N(t) - \lambda t$$

is a martingale, also called as compensated Poisson process. Indeed, using the properties from the definition, for any $0 \leq s < t$ we have

$$\mathrm{E}\,(M(t) - M(s) \mid \mathcal{F}(s)) = \mathrm{E}\,(N(t) - N(s) \mid \mathcal{F}(s)) - \mathrm{E}\,(\lambda \cdot (t-s) \mid \mathcal{F}(s)) = 0.$$

Another well-known property of Poisson processes is that the waiting time until the next occurrence follows the exponential distribution with parameter λ.

General point processes. For general point processes the (stochastic and/or time-variant) intensity is defined as follows: let $(N(t))$ be a point process, adapted to $\mathcal{F}(t)$, and let $\lambda(t)$ be a non-negative, locally integrable and $\mathcal{F}(t)$-progressively measurable process. If for all non-negative, adapted and left-continuous process $C(t)$

$$\mathrm{E}\left(\int_0^\infty C(s)dN(s)\right) = \mathrm{E}\left(\int_0^\infty C(s)\lambda(s)ds\right) \tag{4.3}$$

holds, then we say that $N(t)$ admits the $(P, \mathcal{F}(t))$-*intensity* $\lambda(t)$, see Bremaud (1981) for details. (Progressive measurability is a technical condition, meaning basically that the whole trajectory is measurable: for all $T > 0$, $\lambda(t)\chi_{[0,T]}(t)$ is $\mathcal{B}(\mathbb{R}) \times \mathcal{F}(T)$-measurable, where χ is the indicator function.) The integral on the left hand side is to be interpreted as follows:

$$\int_0^t C(s)\ dN(s) = \sum_{n=1}^{N(t)} C(T_n).$$

An intuitive way to construct a point process with intensity $\lambda(t)$ is as follows. Consider

a Poisson process $N(x, y)$ on the plane with unit intensity. In this case the number of points on every two disjoint areas are independently distributed according to the Poisson distribution with intensities equaling the Lebesgue measure of the areas. Then, $N(t)$ can be considered as the number of points under $\lambda(t)$:

$$N(t) = \int_{[0,t]} \int_{[0,\lambda(s)]} dN(s, .)ds.$$

4.1.3 Hawkes processes

The Hawkes process $(N(t))_{t\in\mathbb{R}}$ is a self-exciting point process. Its intensity λ depends on the past of the process through the formula

$$\lambda(t) = m + \int_{(-\infty,t)} g(t-u)dN(u), \tag{4.4}$$

where $m \geq 0$ and $g : [0, \infty) \to [0, \infty)$.

A necessary condition for (4.4) to have a stationary solution with $\lambda(t) \in L^1$ is that

$$\int_0^\infty g(u)du < 1. \tag{4.5}$$

This condition is sufficient for the existence of a stationary process with the structure given above, see Hawkes and Oakes (1974).

In this chapter we consider a class of Hawkes processes, proposed in Gerencsér et al. (2008b). In this model the intensity λ satisfies the linear state equations

$$dx(t) = -Ax(t)dt + bdN(t), \tag{4.6}$$
$$\lambda(t) = c^T x(t-) + m. \tag{4.7}$$

with a matrix A and vectors b, c, such that the system's impulse response is non-negative, i.e.

$$g(u) = c^T e^{-Au} b \geq 0, \quad \text{for all } u \geq 0. \tag{4.8}$$

We consider only matrices A, for which $-A$ is stable. The sample path of the state process x is right continuous with left limits. The left limit at a given time t is denoted by $x(t-)$.

In this model the coordinates of x represent activities of different analysts (or different analyst groups) with respect to a given company (or industry sector). The parameter b

controls how much analyst coverage increases on news events and parameter a expresses the degree of the coverage deterioration.

The stability condition (4.5) for system (4.6)-(4.7) reads as

$$\int_0^\infty c^T e^{-At} b dt = c^T A^{-1} b < 1. \tag{4.9}$$

The expected value of the intensity can be calculated form (4.4) using (4.3) as follows:

$$\mathrm{E}\,(\lambda) \;=\; m + \mathrm{E}\,(\lambda) \int_0^\infty c^T e^{-At} b dt, \tag{4.10}$$

thus

$$\mathrm{E}\,(\lambda) \;=\; \frac{m}{1 - c^T A^{-1} b}. \tag{4.11}$$

From (4.11) we see that the intensity process is transient if (4.9) does not hold.

The log-likelihood of the observation $(N(t))_{0 \le t \le T}$ can be written as

$$L_T(\vartheta) = \int_0^T -\hat{\lambda}(t) dt + \int_0^T \ln \hat{\lambda}(t) dN(t)$$

where $\hat{\lambda}(t) = \hat{\lambda}(t, \vartheta)$ is the solution of (4.6)-(4.7) with the observed point process $N(t)$ and parameter vector ϑ, see e.g. in Daley and Vere-Jones (2003b). The Fisher information contained in the observation of $(N(t))_{0 \le t \le T}$ is

$$I_T(\vartheta) = \mathrm{E}\left(-\partial_\vartheta^2 L_T(\vartheta) \right) = \mathrm{E}\left(\int_0^T \partial_\vartheta^2 \hat{\lambda}(t) dt - \int_0^T \partial_\vartheta^2 \ln \hat{\lambda}(t) dN(t) \right).$$

We obtain that the Fisher information contained in the observation can be written as

$$I_T(\vartheta) = \int_0^T \mathrm{E}\left(\frac{(\partial_\vartheta \hat{\lambda}(t))(\partial_\vartheta \hat{\lambda}(t))^T}{\hat{\lambda}^2(t)} \lambda(t) \right) dt. \tag{4.12}$$

Assuming that the initial state $x(0)$ is known, then $\hat{\lambda}(t) = \lambda(t)$ when ϑ is the true parameter. Note that if ϑ equals the true parameter, with arbitrary initialization we have $\hat{\lambda}(t) = \lambda(t) + c^T e^{-At} x(0)$, thus the error decays exponentially fast. From identity (4.12) and the ergodicity we can see that I_T/T has a limit, which we call the time–normalized or

asymptotic Fisher information and denote by $I(\vartheta)$, i.e.

$$I(\vartheta) = \mathrm{E}\left(\frac{\lambda_\vartheta \lambda_\vartheta^T}{\lambda}\right),$$

where the expectation is taken with respect to the stationary distribution and λ_ϑ stands for the derivative $\partial_\vartheta \hat{\lambda}$. Recall that $\lambda_\vartheta(t)$ is the derivative of the calculated intensity.

Standard Hawkes processes. In the simplest case as introduced in Hawkes (1971a), (4.6)-(4.7) reduces to

$$d\lambda(t) = -a(\lambda(t) - m)dt + \sigma dN(t) \tag{4.13}$$

where a, $\sigma = bc$, m are positive real parameters, $\vartheta^T = (a, \sigma, m)$. For the standard Hawkes process the stability condition simplifies to $\sigma < a$.

4.2 Simulation of Hawkes processes

In order to be able to examine Hawkes processes we have to simulate them with given parameters. This seemingly innocent problem is in fact quite challenging, see Moller and Rasmussen (2005), Moller and Rasmussen (2006). Unfortunately, these simulation methods are inherently off-line. An alternative procedure is to actually generate the sequence of events, such as news or credit events using the following observation: if t_n is a Poisson process with deterministic intensity $\lambda(t)$, then with

$$M(t) = \int_0^t \lambda_u du$$

the process $\tau_n = M(t_n)$ is a homogeneous Poisson-process with intensity 1. Thus, we can generate t_n by the inverse mapping

$$t_n = M^{-1}(\tau_n).$$

Let us now assume that T_n has been constructed for the Hawkes process with given dynamics. Let τ_n be as above. Then solve (4.6)-(4.7) with $dN_t = 0$, and noting that the solution has a simple, closed form, define T_{n+1} by the equation

$$\int_{T_n}^{T_{n+1}} \lambda_u du = \tau_{n+1} - \tau_n.$$

For $T_n \leq t < T_{n+1}$ the dynamics of x can be written from (4.6) as

$$x(t) = e^{-A(T_n - t)} \left(x(T_n-) + b \right). \tag{4.14}$$

Put $\Delta T_{n+1} = T_{n+1} - T_n$. Thus, having $x(T_n-)$, T_n, our goal is to calculate ΔT_{n+1}. Substituting (4.14) into (4.7) we get

$$\int_{T_n}^{T_{n+1}} \lambda_u du = m\Delta T_{n+1} + c^T \left(I - e^{-A\Delta T_{n+1}} \right) \left(x(T_n-) + b \right) = \nu_{n+1},$$

where $\nu_{n+1} \sim Exponential(1)$ i. i. d. The above equation can be solved by standard numerical root-finder procedures.

Figure 4.1 shows the trajectories of x and λ of a simulated Hawkes process with parameters

$$A = \begin{pmatrix} 0.08 & 0 \\ 0 & 0.02 \end{pmatrix}, \quad b = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \quad c = \begin{pmatrix} 0.02 \\ 0.01 \end{pmatrix}, \quad m = 0.02.$$

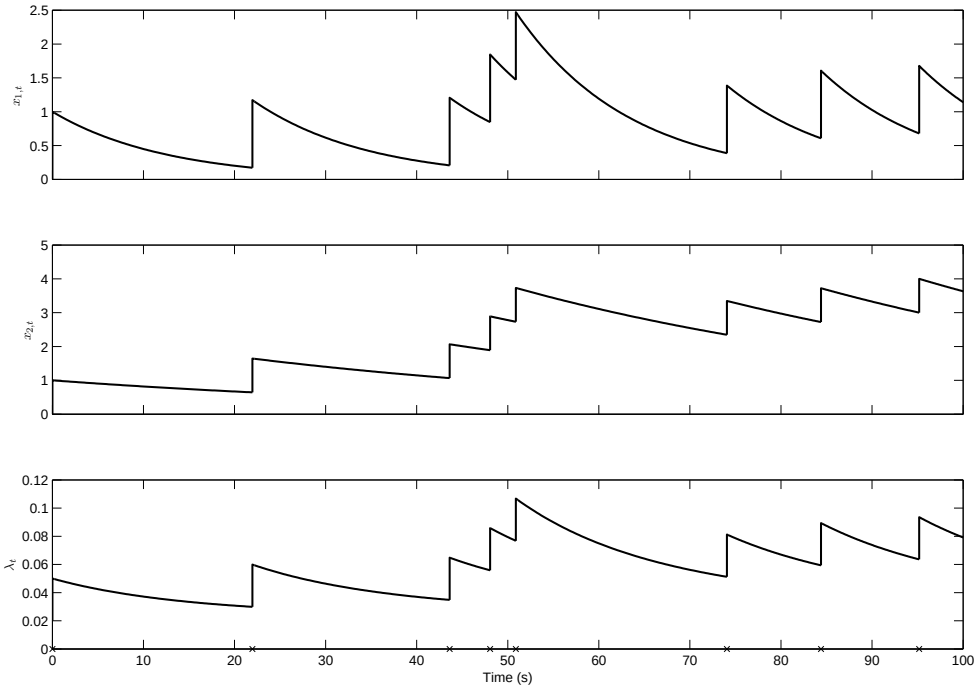

Figure 4.1: A simulated Hawkes process. The symbol $\times$ indicates the event points.

4.3 Identification of Hawkes models

To develop a maximum-likelihood estimator choose an arbitrary parameter ϑ, and define the frozen-parameter process $\hat{\lambda}(t) = \hat{\lambda}(t, \vartheta)$ by

$$\hat{\lambda}(t) \;=\; m + \int_{(0,t]} g(t-u, \vartheta) dN(u), \tag{4.15}$$

with initial condition $\hat{\lambda}(0) = m$. The asymptotic *negative* log-likelihood function is defined as

$$W(\vartheta) = \lim \frac{1}{T} \int_0^T \left(\log \hat{\lambda}(t)\ dN(t) - \hat{\lambda}(t)\ dt \right) = \mathrm{E}\left((\log \hat{\lambda}(t)) \cdot\ \lambda(t) - \hat{\lambda}(t) \right), \tag{4.16}$$

assuming stationary initialization for $\hat{\lambda}(t) = \hat{\lambda}(t, \vartheta)$. Here we made use of the formal calculus $\mathrm{E}\left(f(t) dN(t) \mid \mathcal{F}(t) \right) = f(t)\lambda(t)dt$ for any left-continuous, adapted process $f(t)$.

The maximum-likelihood estimate for Hawkes processes has been studied in depth in Ogata and Akaike (1982). The asymptotic properties of the maximum likelihood estimates of point processes have been discussed in Ogata (1978). Differentiating with respect to ϑ leads to the likelihood equation

$$\frac{\partial}{\partial \vartheta} L_T(\vartheta) = \int_0^T \frac{\partial}{\partial \vartheta} \hat{\lambda}_t \cdot \left(\frac{dN_t}{\hat{\lambda}_t} -\ dt \right) = 0. \tag{4.17}$$

In asymptotic terms we would get

$$\frac{\partial}{\partial \vartheta} W(\vartheta) = -\mathrm{E}\left(\hat{\lambda}_\vartheta(t) \cdot \left(\frac{\lambda(t)}{\hat{\lambda}(t)} - 1 \right) \right), \tag{4.18}$$

assuming stationary initialization for $\hat{\lambda}_t = \hat{\lambda}_t(\vartheta)$. Obviously, this becomes 0 for $\vartheta = \vartheta^*$. The asymptotic Fisher information matrix is

$$I(\vartheta^*) = \frac{\partial^2}{\partial \vartheta^2} W(\vartheta)_{|\vartheta=\vartheta^*} = \mathrm{E}\left(\frac{1}{\hat{\lambda}(t, \vartheta^*)} \cdot \hat{\lambda}_\vartheta(t, \vartheta^*)\ \hat{\lambda}_\vartheta^T(t, \vartheta^*) \right) \tag{4.19}$$

with the right initialization, see also Ogata (1978). In practice, a wrong initialization does not really matter, since the error decays exponentially fast. The computation of the gradient process $\hat{\lambda}_{\vartheta t} = \frac{\partial}{\partial \vartheta} \hat{\lambda}_t$ is straightforward.

The process $\hat{\lambda}_t$ and its derivatives are explicitly computable between any two events by solving a linear differential equation with constant coefficients. Having $\hat{\lambda}(t)$ and $\hat{\lambda}_\vartheta(t)$ at our disposal, the off-line ML (maximum likelihood) estimate can be easily obtained by a gradient method.

The conceptual framework of the quasi-Newton-type recursive maximum likelihood method (RML) in continuous time is given as follows:

$$d\hat{\vartheta}(t) = \frac{1}{t}\hat{H}^{-1}(t)g(t) \tag{4.20}$$

$$d\hat{H}(t) = \frac{1}{t}\left(H(t) - \hat{H}(t)\right), \tag{4.21}$$

where $\hat{\vartheta}(t)$ denotes the current estimate at time t and $g(t)$ and $H(t)$ are online approximations of the gradient and Hessian of the asymptotic negative log-likelihood function, respectively. The initial Hessian $\hat{H}(0)$, and the initial parameter $\hat{\vartheta}(0)$ is set by the user, usually $\hat{H}(0) = I$. In case of our Hawkes processes (4.6)-(4.7) we apply

$$g(t) = \tilde{\lambda}_\vartheta(t) \cdot \left(\frac{dN(t)}{\tilde{\lambda}(t)} - \ dt\right) \tag{4.22}$$

$$H(t) = \frac{1}{\tilde{\lambda}(t)} \cdot \tilde{\lambda}_\vartheta(t)\ \tilde{\lambda}_\vartheta^T(t), \tag{4.23}$$

where $\tilde{\lambda}(t), \tilde{\lambda}_\vartheta(t)$ are on-line estimates of $\hat{\lambda}(t, \vartheta(t)), \hat{\lambda}_\vartheta(t, \vartheta(t))$, respectively. Note that, instead of (4.21), we can apply the following recursion to calculate $\hat{H}^{-1}$ without matrix inversion:

$$d\hat{H}^{-1}(t) = -\hat{H}^{-1}(t)d\hat{H}(t)\hat{H}^{-1}(t) = \tag{4.24}$$

$$= \frac{1}{t}\left(\hat{H}^{-1}(t) - \hat{H}^{-1}(t)\frac{1}{\tilde{\lambda}(t)}\tilde{\lambda}_\vartheta(t)\ \tilde{\lambda}_\vartheta^T(t)\hat{H}^{-1}(t)\right) = \tag{4.25}$$

$$= \frac{1}{t}\left(\hat{H}^{-1}(t) - \frac{1}{\tilde{\lambda}(t)}\left(\hat{H}^{-1}(t)\tilde{\lambda}_\vartheta(t)\right)\left(\hat{H}^{-1}(t)\tilde{\lambda}_\vartheta(t)\right)^T\right), \tag{4.26}$$

where we used that $\hat{H}^{-1}(t)$ is symmetric. Applying recursion (4.26) saves the computational time needed for the matrix inversion.

Identification of the standard Hawkes model. Let us now complete the differential equation system (4.20)-(4.23) with the specifications for $\tilde{\lambda}(t), \tilde{\lambda}_\vartheta(t)$. For simplicity, we shall consider first the standard Hawkes model (4.13). Let the true parameters be denoted

by $\vartheta^* = (a^*, \sigma^*, m^*)$. The frozen-parameter process $\hat{\lambda}(t, \vartheta)$ is then defined by

$$d\hat{\lambda}(t) = a(\hat{\lambda}(t) - m)dt + \sigma dN(t). \tag{4.27}$$

Differentiating (4.27) with respect to $\vartheta = (a, m, \sigma)$ we get:

$$d\hat{\lambda}_{at} = a\hat{\lambda}_a(t)\ dt + \left(\hat{\lambda}(t) - m\right)\ dt \tag{4.28}$$

$$d\hat{\lambda}_\sigma(t) = a\hat{\lambda}_\sigma(t)\ dt + dN(t) \tag{4.29}$$

$$d\hat{\lambda}_m(t) = a\hat{\lambda}_m(t)\ dt - a\ dt. \tag{4.30}$$

Recall that $\hat{\vartheta}(t) = (\hat{a}(t), \hat{\sigma}(t), \hat{m}(t))$ denotes the current estimate at time t, see (4.20). Define

$$d\tilde{\lambda}(t) = \hat{a}(t)\left(\tilde{\lambda}(t) - \hat{m}(t)\right)dt + \hat{\sigma}(t)dN(t), \tag{4.31}$$

and similarly for the approximations of the derivatives:

$$d\tilde{\lambda}_a(t) = \hat{a}(t)\tilde{\lambda}_a(t)\ dt + \left(\tilde{\lambda}(t) - \hat{m}(t)\right)dt \tag{4.32}$$

$$d\tilde{\lambda}_\sigma(t) = \hat{a}(t)\tilde{\lambda}_\sigma(t)\ dt + dN(t) \tag{4.33}$$

$$d\tilde{\lambda}_m(t) = \hat{a}(t)\tilde{\lambda}_m(t)\ dt - \hat{a}(t)\ dt. \tag{4.34}$$

Thus, solving the differential equation system given by (4.20)-(4.23) and (4.31)-(4.34) gives the quasi-RML estimate $\hat{\vartheta}(t)$. Note that we apply a basic resetting mechanism at T_n to keep $\hat{\vartheta}$ in the stability domain, see Gerencsér and Mátyás (2007a).

Figure 4.2 demonstrates the convergence of $\hat{\vartheta}(t)$ in a standard Hawkes model with $\vartheta^* = (0.35, 0.3, 0.1)$.

Identification of the general Hawkes model. In the class of Hawkes models (4.6)-(4.7) we are interested in estimating the true system parameters A^*, b^*, c^*, m^*. We can simplify presentation by considering these variables as a known function of some real parameter vector ϑ:

$$A = A(\vartheta), \quad b = b(\vartheta), \quad c = c(\vartheta), \quad m = m(\vartheta). \tag{4.35}$$

For a simple example, ϑ could be a vector collecting the elements of A, b, c, m with mappings connecting the corresponding elements of ϑ and A, b, c, m.

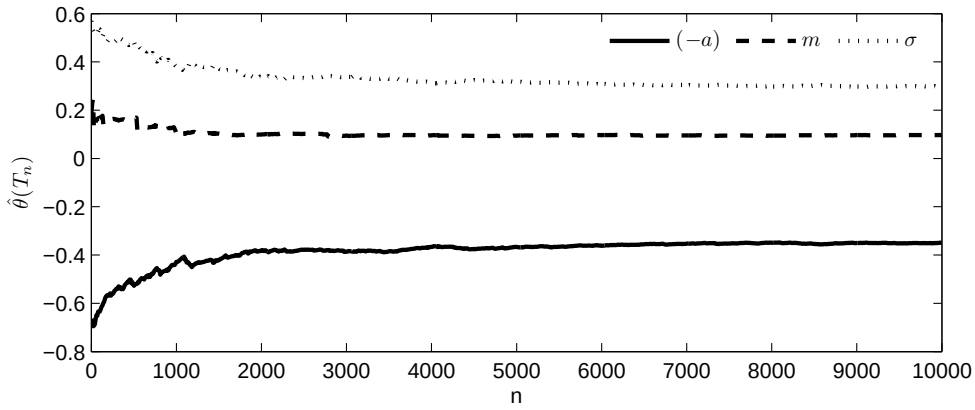

Figure 4.2: Recursive estimation of the parameters for the test problem

Then, for any ϑ, define

$$d\hat{x}(t) = -A\hat{x}(t)dt + bdN(t), \tag{4.36}$$

$$\hat{\lambda}(t) = c^T\hat{x}(t-) + m, \tag{4.37}$$

with $\hat{x}(0) = 0$. Differentiating (4.36)-(4.37) with respect to ϑ we get:

$$d\hat{x}_\vartheta(t) = -A_\vartheta\hat{x}(t)dt - A\hat{x}_\vartheta(t)dt + b_\vartheta dN(t), \tag{4.38}$$

$$\hat{\lambda}_\vartheta(t) = c_\vartheta^T\hat{x}(t-) + c^T\hat{x}_\vartheta(t) + m_\vartheta. \tag{4.39}$$

Put

$$\hat{X}(t) = \begin{pmatrix} \hat{x}(t) \\ \hat{x}_\vartheta(t) \end{pmatrix}, \quad \hat{\Lambda}(t) = \begin{pmatrix} \hat{\lambda}(t) \\ \hat{\lambda}_\vartheta(t) \end{pmatrix}.$$

Then we can write (4.36)-(4.39) in matrix form:

$$d\hat{X}(t) = -\begin{pmatrix} A & 0 \\ A_\vartheta & A \end{pmatrix}\hat{X}(t)dt + \begin{pmatrix} b \\ b_\vartheta \end{pmatrix}dN(t), \tag{4.40}$$

$$d\hat{\Lambda}(t) = \begin{pmatrix} c^T & 0 \\ c_\vartheta^T & c^T \end{pmatrix}\hat{X}(t) + \begin{pmatrix} m \\ m_\vartheta \end{pmatrix}. \tag{4.41}$$

Finally, having the current estimate $\hat{\vartheta}(t)$ at hand, we approximate $\hat{X}(t, \hat{\vartheta}(t))$ and $\hat{\Lambda}(t, \hat{\vartheta}(t))$ by

$$\tilde{X}(t) = \begin{pmatrix} \tilde{x}(t) \\ \tilde{x}_\vartheta(t) \end{pmatrix}, \quad \tilde{\Lambda}(t) = \begin{pmatrix} \tilde{\lambda}(t) \\ \tilde{\lambda}_\vartheta(t) \end{pmatrix}, \tag{4.42}$$

respectively, for which

$$d\tilde{X}(t) = -\begin{pmatrix} \hat{A}(t) & 0 \\ \hat{A}_\vartheta(t) & \hat{A}(t) \end{pmatrix} \tilde{X}(t)dt + \begin{pmatrix} \hat{b}(t) \\ \hat{b}_\vartheta(t) \end{pmatrix} dN(t), \tag{4.43}$$

$$d\tilde{\Lambda}(t) = \begin{pmatrix} \hat{c}^T(t) & 0 \\ \hat{c}_\vartheta^T(t) & \hat{c}^T(t) \end{pmatrix} \tilde{X}(t) + \begin{pmatrix} \hat{m}(t) \\ \hat{m}_\vartheta(t) \end{pmatrix}. \tag{4.44}$$

Thus, solving the differential equation system given by (4.20)-(4.23) and (4.42)-(4.44) gives the quasi-RML estimate $\hat{\vartheta}(t)$.

Identifiability issues. Let the invertible matrix T define a similarity transformation. Let us apply T on the state x of the system defined by (A^*, b^*, c^*, m^*). Note that applying a similarity transformation has no effect on the impulse response function as it only defines an alternative basis for the state vector. However, it transforms the system parameters into $(T^{-1}(A^*)T, T^{-1}(b^*), (c^*)^T T, m^*)$. Thus, since T can be chosen arbitrarily, there are infinitely many system parametrizations describing the observed point process $N(t)$. Hence we have to reduce the dimension of the parameter space in order the system to be identifiable. To this end, we assume that

(i) A^* is diagonal,

(ii) $b^* = (1, 1)^T$.

Simulation results. We present simulation results for a model with two-dimensional x, where $\vartheta = (A_{11}, A_{22}, c_1, c_2, m)$. Figure 4.3 demonstrates the convergence of $\hat{\vartheta}(t)$, the true model parameters are as defined in (4.2). The empirical eigenvalues of the Fisher information matrix calculated from data T_n generated by this model are

$$z = \begin{pmatrix} 0.0586 \\ 0.5915 \\ 3.3189 \\ 10.3902 \\ 234.6782 \end{pmatrix}.$$

The condition number is moderately high (4004.7) in this setting. All of the eigenvalues are significantly bigger than zero, suggesting that the model is indeed identifiable. For comparison, we mention that we calculated almost zero ($1.22 \cdot 10^{-5}$) for the smallest eigenvalue in an overparametrized case, when ϑ includes the elements of b as well.

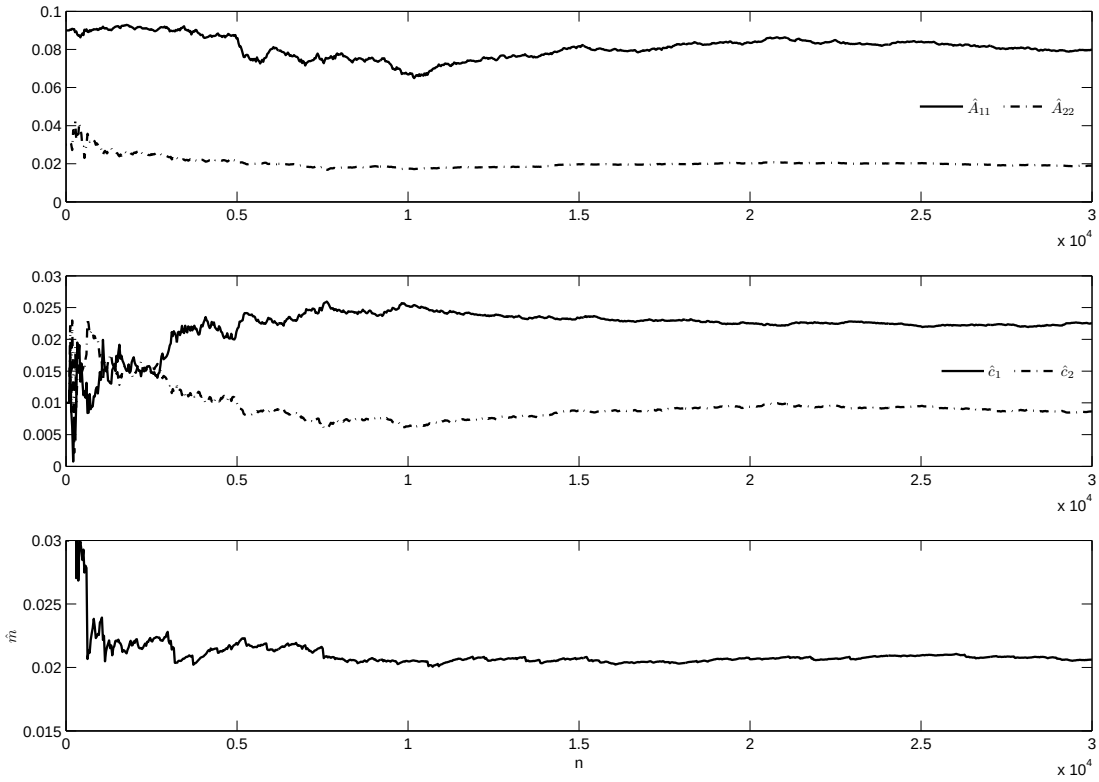

Figure 4.3: $\hat{\vartheta}(T_n)$ in the Hawkes model with two-dimensional state.

4.4 The Fisher information matrix

Under general conditions the asymptotic accuracy of the identification is determined by the asymptotic Fisher information. Our goal in this section is to understand the behavior of the Fisher information matrix near the boundary of the stability domain. The investigations in this chapter are carried out in the standard Hawkes model, see Prokaj and Torma (2010) for related analysis in the model defined (4.6)-(4.7). We first present some observations regarding the limiting behavior of the Fisher information, which we then examine theoretically.

Recall that in the standard case, (4.6)-(4.7) reduces to

$$d\lambda(t) = -a(\lambda(t) - m)dt + \sigma dN(t)$$

where a, $\sigma = bc$ and m are positive real parameters. In this model the parameter vector is $\vartheta = (a, \sigma, m)$. For the standard Hawkes process the stability condition simplifies to $\sigma < a$. Notice, that the value of m does not play any role in the stability condition.

Figure 4.4 illustrates the effect of approaching criticality. The value of $\sigma = 0.3$ and $m = 0.1$ is kept fixed. On the left hand side of Figure 4.4, $a = 0.35$, while on the right it is $a = \sigma + 10^{-6}$. The density of events and also $\mathrm{E}(\lambda)$ is much larger when the parametrization

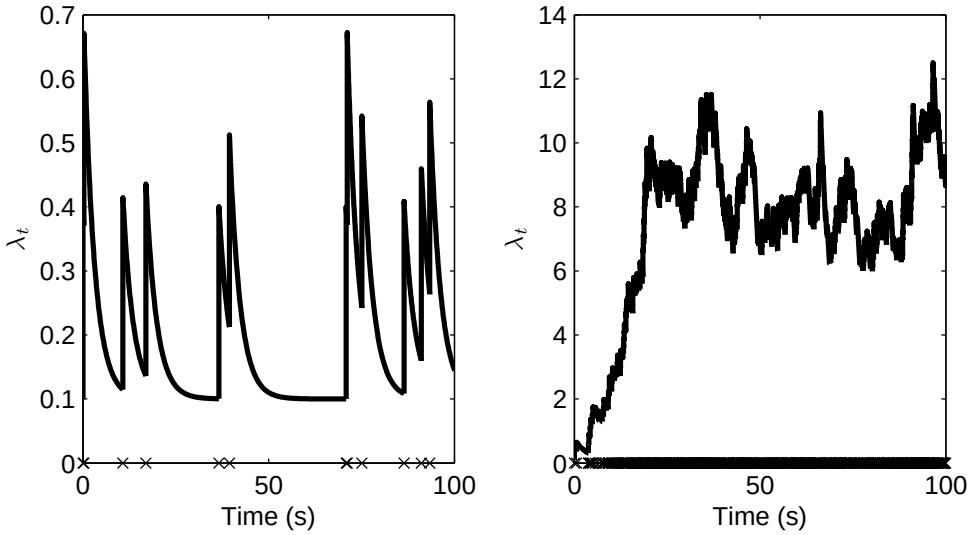

Figure 4.4: Intensity in the stable (left) and the nearly unstable (right) case.

is close to the boundary of the stability domain, i.e., when $a - \sigma$ is small. Moreover, the nearly unstable intensity process shows diffusion characteristics.

Let us now turn to the time–normalized Fisher information with respect to the parameter a, i.e.,

$$I(a) = \mathrm{E}\left(\frac{\lambda_a^2}{\lambda}\right).$$

To evaluate $I(a)$ the joint law of λ_a and λ is needed. In Figure 4.5 the scatter plot of λ_a against λ is shown with decreasing decay factor a, where $a = 1$ is the critical value. We can see that the cloud gets narrower as a gets closer to 1. This indicates an increasing correlation between λ and λ_a. It is easy to calculate the correlation coefficient, which indeed tends to -1 as a goes to 1, see Proposition 4.4.2 below.

Comparing the expected values of λ and λ_a one can see that they have the same order of magnitude, see (4.45) below. Then, at least at a heuristic level, we can expect that $\lambda_a^2/\lambda \approx \lambda$ and $I(a) \approx \mathrm{E}(\lambda)$ for $a - \sigma \approx 0$. This shows that the time–normalized Fisher information $I(a)$ goes to infinity as a approaches the critical value.

In a similar manner one easily finds from Lemma 1 below that $\lambda_a \approx \lambda_\sigma \approx \lambda \approx (a-\sigma)^{-1}$:

$$\mathrm{E}(\lambda) = \frac{am}{a-\sigma}, \quad \mathrm{E}(\lambda_a) = \frac{-m\sigma}{a(a-\sigma)}, \quad \mathrm{E}(\lambda_\sigma) = \frac{m}{a-\sigma}. \tag{4.45}$$

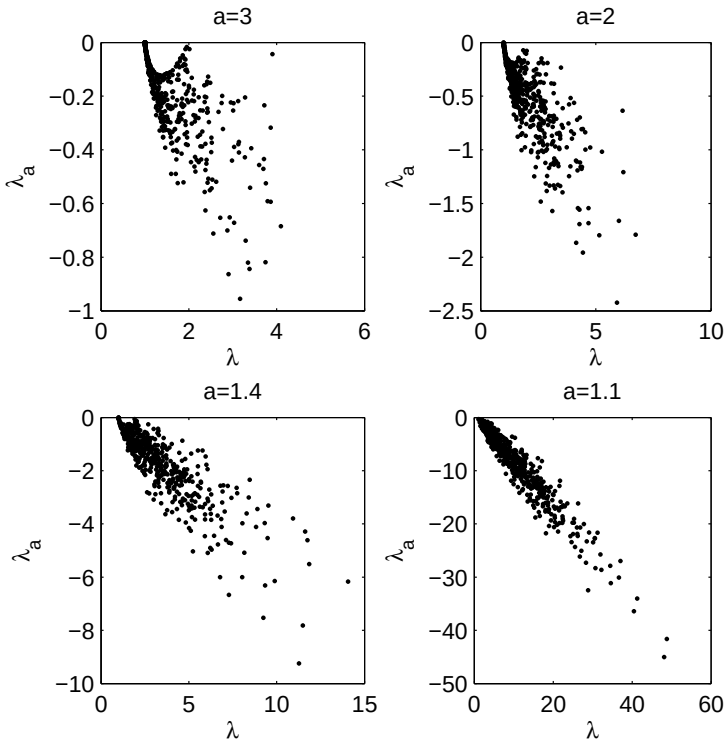

Figure 4.5: λ_a vs. λ in a standard Hawkes model with $\sigma = m = 1$.

Thus, the rescaled Fisher information matrix with respect to parameters a and σ has the form

$$\lim_{a-\sigma\to 0}(a-\sigma)I(a,\sigma) = vv^T$$

where v is a vector with non-zero elements.

Let us now make these findings precise. The first step is to investigate the limiting behavior of the intensity process. Similar investigations have been carried out for branching processes with immigration in Ispány et al. (2005), Ispány and Pap (2007). See also the discussion on the analogies in the introduction of Prokaj and Torma (2010). Next, in Theorem 2 we calculate the stationary distribution of the appropriately rescaled intensity process, in which we use the following differentiation rule:

Proposition 4.4.1. *Consider the (right continuous) process η which satisfies*

$$d\eta_t = a_t dt + \sigma dN_t,$$

where N_t is a point process and a continuously differentiable function f. Then, for $s < t$

$$f(\eta_t) - f(\eta_s) = \int_s^t f'(\eta_u) a_u du + \int_s^t [f(\eta_u) - f(\eta_{u-})]\, dN_u.$$

or in the differential form

$$df(\eta_t) = f'(\eta_t) a_t dt + [f(\eta_t) - f(\eta_{t-})]\, dN_t.$$

Theorem 2. *Consider the stationary point process given in (4.13). Let σ_0 be a positive real number. Then, $(a-\sigma)\lambda$ converges to $\Gamma_{\frac{2m}{\sigma_0}, \frac{2}{\sigma_0^2}}$ in distribution, as a and σ approach σ_0 such that $\sigma < a$.*

Proof. In order to calculate the characteristic function, first we determine the dynamics of $e^{i\alpha\lambda(t)}$. Applying Proposition 4.4.1 with $f(x) = e^{i\alpha x}$, we can write

$$de^{i\alpha\lambda(t)} = -i\alpha e^{i\alpha\lambda(t)} a(\lambda(t) - m) d_t + [e^{i\alpha(\lambda(t)+\sigma)} - e^{i\alpha\lambda(t)}] dN_t.$$

Taking expectation at both sides and applying (4.3) we get

$$0 = E[-i\alpha e^{i\alpha\lambda(t)} a(\lambda(t) - m) + e^{i\alpha\lambda(t)}(e^{i\alpha\sigma} - 1)\lambda(t)],$$

where we set the left side to zero, since the mean change is zero under stationarity . With $(\log\varphi)'(\alpha)$ denoting the derivative of the logarithm of the characteristic function $\varphi(\alpha)$ we have

$$(\log\varphi)'(\alpha) = \frac{\alpha a m}{e^{i\alpha\sigma} - 1 - i\alpha a}. \tag{4.46}$$

Applying basic calculus and elementary properties of the characteristic function we can write from (4.46)

$$\log\varphi(\alpha(a-\sigma)) = \int_0^{\alpha(a-\sigma)} \frac{xam}{e^{ix\sigma} - 1 - ixa} dx. \tag{4.47}$$

Let us now introduce

$$y = \frac{x}{a-\sigma}$$

and change variables in (4.47) to get

$$\log\varphi_{(a-\sigma)\lambda(t)}(\alpha) = \int_0^{\alpha} \frac{yam}{\frac{e^{i\sigma y(a-\sigma)} - 1 - iy(a-\sigma)a}{(a-\sigma)^2}} dy. \tag{4.48}$$

Applying

$$-iy(a-\sigma)a = -iy(a-\sigma)(\sigma+(a-\sigma)) = -iy\sigma(a-\sigma) - iy(a-\sigma)^2$$

in the denominator of the integrand we can rewrite (4.48) as

$$\log \varphi_{(a-\sigma)\lambda(t)}(\alpha) = \int_0^\alpha \frac{yam}{(iy)^2 D(y,a,\sigma) - iy} dy. \tag{4.49}$$

with

$$D(y,a,\sigma) = \frac{e^{\sigma iy(a-\sigma)} - 1 - \sigma iy(a-\sigma)}{[iy(a-\sigma)]^2}.$$

Let us now take limit $a, \sigma \to \sigma_0$, $\sigma < a$ on (4.49):

$$\lim_{\substack{a,\sigma\to\sigma_0 \\ \sigma<a}} \log \varphi_{(a-\sigma)\lambda(t)}(\alpha) = \int_0^\alpha \frac{y\sigma_0 m}{(iy)^2 \frac{\sigma_0^2}{2} - iy} dy = \tag{4.50}$$

$$= -\frac{2m}{\sigma_0} \log\left(1 - \frac{i\alpha\sigma_0^2}{2}\right). \tag{4.51}$$

In (4.50) we used

$$\lim_{\substack{a,\sigma\to\sigma_0 \\ \sigma<a}} D(y,a,\sigma) = \lim_{z\searrow 0} \frac{e^{\sigma_0 z} - 1 - \sigma_0 z}{z^2} =$$

$$= \lim_{z\searrow 0} \frac{(1 + \sigma_0 z + \frac{(\sigma_0 z)^2}{2} + \ldots) - 1 - \sigma_0 z}{z^2} = \frac{\sigma_0^2}{2},$$

with $z = iy(a-\sigma)$.

It follows from (4.51) that the characteristic function in the limit is

$$\lim_{\substack{a,\sigma\to\sigma_0 \\ \sigma<a}} \varphi_{(a-\sigma)\lambda(t)}(\alpha) = \left(1 - \frac{i\alpha\sigma_0^2}{2}\right)^{-\frac{2m}{\sigma_0}}$$

which is the characteristic function of the Gamma-distribution with parameters as in the theorem. □

Theorem 2 can be seen as a special case of Theorem 2 in Prokaj and Torma (2010) with a slightly different normalization factor. The next result, Theorem 3 says that the time-normalized Fisher information gets infinite as we approach the boundary of the stability domain.

Theorem 3. *Consider the stationary solution of (4.13). Let σ_0 be a positive real number. Then,*

$$\lim_{\substack{a,\sigma\to\sigma_0\\ \sigma<a}} \mathrm{E}\left(\frac{\lambda_a^2}{\lambda}\right) = \lim_{\substack{a,\sigma\to\sigma_0\\ \sigma<a}} \mathrm{E}\left(\frac{\lambda_\sigma^2}{\lambda}\right) = \infty,$$

moreover $\liminf(a-\sigma)\mathrm{E}\left(\lambda_a^2\lambda^{-1}\right) > 0$ *and similarly for* λ_σ.

The proof is based on the key observation that the intensity process and its derivatives are fully correlated in the limit, which is presented in Proposition 4.4.2. For this to show we need the derivatives of λ and a method for calculating the moments and covariance of the stationary λ, λ_a, λ_σ.

Put

$$\Lambda_t = \begin{pmatrix} \lambda_t - m \\ \lambda_{at} \\ \lambda_{\sigma t} \end{pmatrix}.$$

We can formulate the standard Hawkes model and its derivatives in matrix form as shown in the following lemma.

Lemma 1 (Derivatives)**.**

$$d\Lambda_t = -A\Lambda_t dt + b dN_t,$$

where

$$A = \begin{bmatrix} a & 0 & 0 \\ 1 & a & 0 \\ 0 & 0 & a \end{bmatrix}, \quad b = \begin{bmatrix} \sigma \\ 0 \\ 1 \end{bmatrix}.$$

Lemma 2 (Moments, covariance)**.** *Let j,k,l denote nonnegative integers, $s = j+k+l \geq 1$. Then, we have*

$$\begin{gathered} -a(j+k+l)\mathrm{E}\left(\lambda_a^j\lambda_\sigma^k\lambda^l\right) + jm\mathrm{E}\left(\lambda_a^{j-1}\lambda_\sigma^k\lambda^l\right) - \\ -j\mathrm{E}\left(\lambda_a^{j-1}\lambda_\sigma^k\lambda^{l+1}\right) + aml\mathrm{E}\left(\lambda_a^j\lambda_\sigma^k\lambda^{l-1}\right) + \\ +\mathrm{E}\left(\lambda_a^j\left(\lambda_\sigma+1\right)^k\lambda^{l+1}\right) + \mathrm{E}\left(\lambda_a^j\lambda_\sigma^k\lambda\left(\lambda+\sigma\right)^l\right) - \\ -\mathrm{E}\left(\lambda_a^j\lambda_\sigma^k\lambda^{l+1}\right) = 0. \end{gathered} \tag{4.52}$$

Proof. We get by differentiating $\lambda_a^j(t)\lambda_\sigma^k(t)\lambda^l(t)$

$$d\lambda_a^j(t)\lambda_\sigma^k(t)\lambda^l(t) = d\lambda_a^j(t)\lambda_\sigma^k(t)\lambda^l(t) + d\lambda_\sigma^k(t)\lambda_a^j(t)\lambda^l(t) + d\lambda^l(t)\lambda_a^j(t)\lambda_\sigma^k(t). \tag{4.53}$$

We can apply Proposition 4.4.1 on the derivatives given by Lemma 1 to calculate $d\lambda^l(t)$,

$d\lambda_a^j(t)$, $d\lambda_\sigma^k(t)$:

$$d\lambda^l(t) = l\lambda^{l-1}(t)(-a\lambda(t) + am)dt + \left[(\lambda(t-) + \sigma)^l - \lambda^l(t-)\right] dN(t), \tag{4.54}$$

$$d\lambda_a^j(t) = j\lambda_a^{j-1}(t)(m - \lambda(t) - a\lambda_a(t))dt, \tag{4.55}$$

$$d\lambda_\sigma^k(t) = k\lambda_\sigma^{k-1}(t)(-a\lambda_\sigma(t))dt + \left[(\lambda_\sigma(t-) + 1)^k - \lambda_\sigma^k(t-)\right] dN(t). \tag{4.56}$$

After inserting the right hand side of the equations (4.54), (4.55), (4.56) into (4.53), we take expectation using (4.3). Note that the random variables in the expectations are all integrable, see Proposition 10 in Prokaj and Torma (2010). Under stationarity the average change is zero, that is

$$d\mathrm{E}\left(\lambda_a^j(t)\lambda_\sigma^k(t)\lambda^l(t)\right) = 0.$$

Ordering the terms yields the stated equation of the moments. □

Proposition 4.4.2 (Joint distribution of scaled λ, λ_a, λ_σ)**.** *Consider the stationary solution of (4.13). Let σ_0 be a positive real number. Then, $((a-\sigma)\lambda, -\sigma(a-\sigma)\lambda_a, a(a-\sigma)\lambda_\sigma)$ converges in distribution to (X, X, X) with $X \sim \Gamma_{\frac{2m}{\sigma_0}, \frac{2}{\sigma_0^2}}$, as a and σ approach σ_0 such that $\sigma < a$.*

Proof. We show that the L^2 norm of the scaled differences

$$(a - \sigma)(\lambda - (-\sigma\lambda_a))$$

and

$$(a - \sigma)(\lambda - a\lambda_\sigma)$$

vanishes in the limit. For the L^2 norm of the first difference we have

$$\begin{aligned}&\lim_{\substack{a,\sigma\to\sigma_0\\ \sigma<a}} \mathrm{E}\left(((a-\sigma)\lambda + \sigma(a-\sigma)\lambda_a)^2\right) = \\ &= \lim_{\substack{a,\sigma\to\sigma_0\\ \sigma<a}} \left[(a-\sigma)^2(\mathrm{E}\left(\lambda^2\right) + 2\sigma\mathrm{E}\left(\lambda_a\lambda\right) + \sigma^2\mathrm{E}\left(\lambda_a^2\right))\right],\end{aligned} \tag{4.57}$$

for the second we have

$$\begin{aligned}&\lim_{\substack{a,\sigma\to\sigma_0\\ \sigma<a}} \mathrm{E}\left(((a-\sigma)\lambda - a(a-\sigma)\lambda_\sigma)^2\right) = \\ &= \lim_{\substack{a,\sigma\to\sigma_0\\ \sigma<a}} \left[(a-\sigma)^2(\mathrm{E}\left(\lambda^2\right) - 2a\mathrm{E}\left(\lambda_a\lambda\right) + a^2\mathrm{E}\left(\lambda_\sigma^2\right))\right].\end{aligned} \tag{4.58}$$

To calculate the expectations in (4.57) and (4.58), using Lemma 2 we build a linear equation system, in which the equations are of type (4.52) with all (j,k,l) triples from

$$\{(j',j',k')|0 \leq j',k',l' \leq 2, \quad j'+k'+l' \geq 1\}\,.$$

Solving the equation system for the expectations gives

$$\mathrm{E}\left(\lambda^2\right) = \frac{am\left(2am+\sigma^2\right)}{2(a-\sigma)^2}, \tag{4.59}$$

$$\mathrm{E}\left(\lambda_a^2\right) = \frac{m\sigma^2\left(a^2+4am-2m\sigma\right)}{2a^2\left(2a-\sigma\right)(a-\sigma)^2}, \tag{4.60}$$

$$\mathrm{E}\left(\lambda_\sigma^2\right) = \frac{m\left(2a+4am-2m\sigma-3a\sigma+2\sigma\right)}{2\left(2a-\sigma\right)(a-\sigma)^2}, \tag{4.61}$$

$$\mathrm{E}\left(\lambda_a\lambda\right) = -\frac{\sigma m(a\sigma-2m\sigma+4am)}{2(a-\sigma)^2(2a-\sigma)}, \tag{4.62}$$

$$\mathrm{E}\left(\lambda_\sigma\lambda\right) = \frac{am\left(4am-2m\sigma+\sigma\right)}{2(a-\sigma)^2(2a-\sigma)}. \tag{4.63}$$

We substitute in (4.57) and (4.58), take limit to get zero for both L^2 norms. □

Proof of Theorem 3. We consider first the Fisher information in parameter a. By Fatou-lemma and Proposition 4.4.2 we have

$$\liminf_{\substack{a,\sigma\to\sigma_0\\ \sigma<a}} \mathrm{E}\left(\frac{[-\sigma(a-\sigma)\lambda_a]^2}{(a-\sigma)\lambda}\right) \geq \mathrm{E}\left(\liminf_{\substack{a,\sigma\to\sigma_0\\ \sigma<a}} \frac{[-\sigma(a-\sigma)\lambda_a]^2}{(a-\sigma)\lambda}\right) = \mathrm{E}\left(\frac{X^2}{X}\right), \tag{4.64}$$

where $X \sim \Gamma_{\frac{2m}{\sigma_0},\frac{2}{\sigma_0^2}}$. It follows, that

$$\liminf_{\substack{a,\sigma\to\sigma_0\\ \sigma<a}} \sigma^2(a-\sigma)\mathrm{E}\left(\frac{\lambda_a^2}{\lambda}\right) \geq \mathrm{E}\left(X\right).$$

From this we conclude. For the other limit, the proof goes analogously with analyzing

$$\mathrm{E}\left(\frac{[a(a-\sigma)\lambda_\sigma]^2}{(a-\sigma)\lambda}\right)$$

in the limit. □

In the following theorem we examine the Fisher information with respect to the parameter m. Heuristically, we would expect that $I(m)$ vanishes near the stability boundary,

since the variance of λ gets there large while m remains constant.

Theorem 4. *Consider the stationary solution of (4.13). Let σ_0 be a positive real number. Then,*

$$\lim_{\substack{a,\sigma\to\sigma_0\\ \sigma<a}} \mathrm{E}\left(\frac{\lambda_m^2}{\lambda}\right) = 0.$$

Proof. Differentiating (4.13) we get

$$d\lambda_m(t) = -a(\lambda_m(t) - 1)dt,$$

the stationary solution of which is $\lambda_m(t) = 1$. Thus,

$$I(m) = \mathrm{E}\left(\frac{\lambda_m^2}{\lambda}\right) = \mathrm{E}\left(\frac{1}{\lambda}\right).$$

Since

$$\frac{1}{\lambda} \le \frac{1}{m},$$

and $\lim \frac{1}{\lambda} = 0$ in distribution, we have $\lim \mathrm{E}\left(\frac{1}{\lambda}\right) = 0$ by the Dominated Convergence Theorem. □

For the standard Hawkes process with parameter $a > 0$ we can give rather precise estimation for the Fisher information. This is based on the identities given in the next statement.

Proposition 4.4.3. *Consider the stationary point process given in (4.13) with $m = \sigma > 0$. Then, for any $k, l \in \mathbb{Z}$, $k \ge 0$ we have $\lambda_a^k(\lambda - m)^l \in L^1(\Omega)$ and*

$$\mathrm{E}\left(\lambda_a^k \lambda^{l+1}\right) = (a(k+l) + m)\mathrm{E}\left(\lambda_a^k(\lambda - m)^l\right) + k\mathrm{E}\left(\lambda_a^{k-1}(\lambda - m)^{l+1}\right) + \mathrm{E}\left(\lambda_a^k(\lambda - m)^{l+1}\right). \quad (4.65)$$

Proof. For $k, l \in \mathbb{Z}$, $k \ge 0$ and $\varepsilon > 0$, the integrability of $\lambda_a^k(\lambda - m + \varepsilon)^l$ follows from Proposition 10 in Prokaj and Torma (2010).

Write the dynamics of $\lambda_a^k(t)(\lambda(t) - m + \varepsilon)^l = x_2^k(t)(x_1 + \varepsilon)^l(t)$ using Proposition 1 and

the change of variable formula:

$$\begin{aligned} d(x_1+\varepsilon)^l(t) &= -al(x_1+\varepsilon)^{l-1}(t)x_1(t)dt+ \\ &\qquad \left((x_1(t-)+\sigma+\varepsilon)^l-(x_1(t-)+\varepsilon)^l\right)dN(t), \\ dx_2^k(t) &= -kx_2^{k-1}(t)\left(x_1(t)+ax_2(t)\right)dt \\ d(x_1(t)+\varepsilon)^l x_2^k(t) &= (x_1+\varepsilon)^l(t)dx_2^k(t)+x_2^k(t)d(x_1+\varepsilon)^l(t) \end{aligned}$$

Since the process $(x_1(t)+\varepsilon)^l x_2^k(t)$ is stationary and in L^1 for all t we have that the mean change is zero. Writing this out, but omitting the actual time t, we obtain that

$$\begin{aligned} -k\mathrm{E}\left(x_2^{k-1}x_1(x_1+\varepsilon)^l\right)-ka\mathrm{E}\left(x_2^k(x_1+\varepsilon)^l\right)-la\mathrm{E}\left(x_1(x_1+\varepsilon)^{l-1}x_2^k\right)+ \\ \mathrm{E}\left(x_2^k(x_1+\sigma+\varepsilon)^l\lambda\right)-\mathrm{E}\left(x_2^k(x_1+\varepsilon)^l\lambda\right)=0. \end{aligned}$$

Rearranging and letting $\varepsilon \to 0+$ gives the relation (4.65) by $\sigma = m$. For $l \geq 0$ the Dominated Convergence Theorem, for $l < 0$ the Beppo-Levi Theorem can be used to see that we can take the limit inside the expectation.

For a given l, the integrability of $\lambda_a^k(\lambda-m)^l$ for all $k \geq 0$ follows from Proposition 10 in Prokaj and Torma (2010) if $l \geq 0$, while for $l < 0$ from (4.65) by induction on $-l$. □

Theorem 5. *In model (4.13) with fixed $m = 1$ and $\sigma = 1$, $a > 1$ we have*

$$\frac{2}{(a-1)a(a+1)}-1<\mathrm{E}\left(\frac{\lambda_a^2}{\lambda}\right)<\frac{2}{(a-1)a(a+1)}.$$

Proof. First note that

$$\frac{2}{(a-1)a(a+1)}=\mathrm{E}\left(\frac{\lambda_a^2}{\lambda-1}\right).$$

This can be easily seen by applying Proposition 4.4.3 with $k=2$, $l=-1$ and with $k=1$, $k=0$, which yield

$$\mathrm{E}\left(\frac{\lambda_a^2}{\lambda-1}\right)=-\frac{2}{a+1}\mathrm{E}\left(\lambda_a\right)$$

and

$$\mathrm{E}\left(\lambda_a\right)=-\frac{1}{a(a-1)},$$

respectively. Thus, we have to prove

$$\mathrm{E}\left(\frac{\lambda_a^2}{\lambda-1}\right)-1<\mathrm{E}\left(\frac{\lambda_a^2}{\lambda}\right)<\mathrm{E}\left(\frac{\lambda_a^2}{\lambda-1}\right). \tag{4.66}$$

The second inequality is trivial, because

$$\frac{\lambda_a^2}{\lambda} < \frac{\lambda_a^2}{\lambda - 1}.$$

Let us now work out the first inequality. Applying Proposition 4.4.3 with $k = 2$, $l = -2$ yields

$$\mathrm{E}\left(\frac{\lambda_a^2}{\lambda}\right) = 2\mathrm{E}\left(\frac{\lambda_a}{\lambda - 1}\right) + \mathrm{E}\left(\frac{\lambda_a^2 \lambda}{(\lambda - 1)^2}\right). \tag{4.67}$$

We can calculate the expectation in the first term of the right hand side of (4.67) as

$$\mathrm{E}\left(\frac{\lambda_a}{\lambda - 1}\right) = \mathrm{E}\left(\frac{\lambda_a(\lambda - \lambda + 1)}{\lambda - 1}\right) = \mathrm{E}\left(\frac{\lambda_a \lambda}{\lambda - 1} - \lambda_a\right) = -1, \tag{4.68}$$

where

$$\mathrm{E}\left(\frac{\lambda_a \lambda}{\lambda - 1}\right) = \mathrm{E}\left(\lambda_a\right) - 1$$

from Proposition 4.4.3 applied with $k = 1$, $l = -1$. For the second term we have

$$\begin{aligned} &\mathrm{E}\left(\frac{\lambda_a^2}{\lambda - 1}\frac{\lambda}{\lambda - 1}\right) - \mathrm{E}\left(\frac{\lambda_a^2}{\lambda - 1}\right) = \mathrm{E}\left(\frac{\lambda_a^2}{(\lambda - 1)^2}\right) = \\ &= \mathrm{E}\left(\frac{\lambda_a}{(\lambda - 1)}\right)^2 + D^2\left(\frac{\lambda_a}{\lambda - 1}\right) = 1 + D^2\left(\frac{\lambda_a}{\lambda - 1}\right), \end{aligned} \tag{4.69}$$

where $D^2(x)$ denotes the variance of x. Combining (4.67) with (4.69) we get

$$\mathrm{E}\left(\frac{\lambda_a^2}{\lambda}\right) = \mathrm{E}\left(\frac{\lambda_a^2}{\lambda - 1}\right) - 1 + D^2\left(\frac{\lambda_a}{\lambda - 1}\right). \tag{4.70}$$

Since the variance is nonnegative, the first inequality in (4.66) holds as well. □

Numerical results. Next, we present a simulation experiment. The time–normalized Fisher information matrix is approximated with a time average

$$\hat{I}(\vartheta) = \frac{1}{T}\int_0^T \frac{\partial_\vartheta \hat{\lambda}(t, \vartheta) \partial_\vartheta \hat{\lambda}(t, \vartheta)^T}{\hat{\lambda}(t, \vartheta)} dt$$

for T large in a long simulation of the standard Hawkes process. We keep the parameters $\sigma = 0.3$ and $m = 0.1$ fixed. Figure 4.6 shows the diagonal elements of this empirical matrix as a approaches σ from above. The Fisher information with respect to parameters a and σ is a decreasing function of $a - \sigma$, while $\hat{I}(m)$ is increasing. The graphs are in accordance

with the analytical results mentioned in the Theorem 3, 4, namely that $I(a)$, $I(\sigma)$ tend to infinity and $I(m)$ tends to zero as $a - \sigma \to 0$ with m fixed.

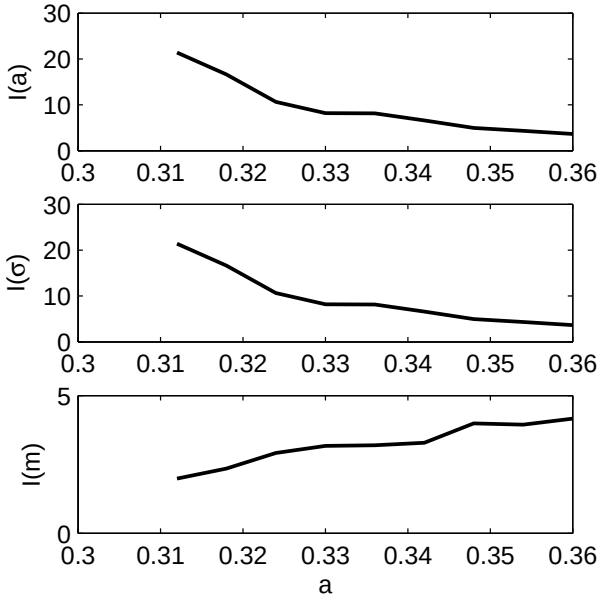

Figure 4.6: Diagonals of the empirical Fisher information matrix in the standard Hawkes case.

From a practical point of view the inverse of the Fisher information matrix $I^{-1}(\vartheta)$ is even more important then $I(\vartheta)$ itself, since $I^{-1}(\vartheta)$ indicates the accuracy of parameter estimation. For example, in the standard Hawkes model asymptotic normality holds for the maximum likelihood estimator, see Ogata (1978). The asymptotic covariance matrix is $I^{-1}(\vartheta)$. Note also that in the standard Hawkes case the overparametrization issue is resolved by introducing $\sigma = bc$.

The inverse of the Fisher information matrix with $a = 0.312$, its eigenvalues z and the condition number κ are

$$\hat{I}^{-1}(a,\sigma,m) = \begin{pmatrix} 0.8737 & 0.8134 & 0.2007 \\ 0.8134 & 0.8059 & 0.1188 \\ 0.2007 & 0.1188 & 0.6605 \end{pmatrix}, \quad z = \begin{pmatrix} 0.0210 \\ 0.6156 \\ 1.7034 \end{pmatrix}, \quad \kappa = 81.11.$$

The parameters a and σ can be estimated by the maximum likelihood method approximately equally accurately, the estimation errors with respect to these two parameters are highly correlated in this nearly unstable case (the correlation coefficient is 0.9694). Moreover, the condition number is moderately high indicating that standard iterative numerical

procedures are applicable for maximum likelihood estimation of ϑ in this model.

In Figure 4.7 the trace of $\hat{I}^{-1}(\vartheta)$ is shown. Simple theoretical considerations imply that $\mathrm{Tr}\left(I^{-1}(\vartheta)\right)$ should first decrease and then go to infinity as ϑ approaches criticality with m fixed. The curve confirms decreasing but it is incomplete on the left due to the immense computational burden arising very close to criticality.

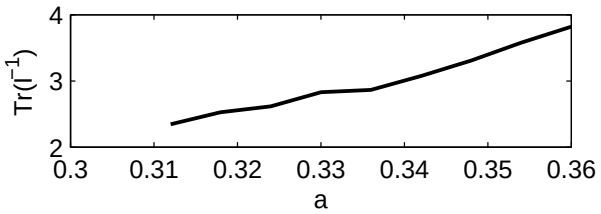

Figure 4.7: The trace of the empirical asymptotic covariance matrix in the standard Hawkes case.

4.5 Conclusion

In this chapter a Hawkes model was presented, which captures the self-exciting effects of market news. We developed algorithms for the simulation and recursive identification of this model. We investigated the Fisher information matrix in the simplest (standard Hawkes) model. In particular, we showed that parts of the diagonal of the asymptotic Fisher information matrix go to infinity, other parts go to zero as the parameters approach the boundary of the stability domain. As a first step we calculated the limit distribution of the appropriately rescaled intensity process.

The presented model describes the dynamics of news arrival, where the news are generated by multiple analysts with respect to a single company. A natural extension of the model is to allow for interdependencies between companies, i.e. to capture the assumption that news about a company can increase the analyst coverage of another company. To this end, we can introduce multiple Poisson-channels as follows:

$$dx(t) = -Ax(t)dt + BdN(t), \tag{4.71}$$

$$\lambda(t) = C^T x(t-) + m, \tag{4.72}$$

where $m, \lambda(t)$ are vector-valued and A, B, C are matrices of appropriate dimensions.

It would be highly interesting to check empirically how well our news arrival model

describes real processes. The major obstacle of empirical investigations is that historical news data is difficult to acquire.

Bibliographical remarks

The linear state Hawkes model, the algorithms for simulation of Hawkes processes and recursive identification have been presented at the 18th International symposium on Mathematical Theory of Networks and Systems at Virginia Tech, Blacksburg, Virginia, USA held in July 28-August 1, 2008, see Gerencsér et al. (2008b), authored by Gerencsér, L., Matias, C., Vágó, Zs., Torma, B., Weiss, B. The Fisher information matrix related results have been presented at the International Conference on Probability and Statistics with Applications, dedicated to the 100th anniversary of the birthday of Béla Gyires; the accompanying paper has been accepted for publication in Publicationes Mathematicae, Debrecen, see Prokaj and Torma (2010).

The linear state Hawkes model was proposed by László Gerencsér, its application to model news arrival processes was proposed by Balázs Torma. The simulation and identification methods (Section 4.2, 4.3) are based on the ideas of László Gerencsér, numerical investigations have been carried out by Balázs Torma. The theoretical investigations regarding the Fisher information matrix (Section 4.4) have been carried out by Vilmos Prokaj and Balázs Torma in cooperation, all numerical experiments have been conducted by Balázs Torma.

Chapter 5

Regression analysis based on high-frequency data

5.1 Introduction

A general property of security prices, such as stock prices is that the price is some integer multiple of the so-called minimum price increment or tick-size, say h. For example, on the NASDAQ stock exchange, $h = 0.01\$$. On futures markets a higher h is applied: on the Chicago Mercantile Exchange, say, $h = 12.50\$$ for the ES Mini S&P500 Futures contract. The need for a minimum price increment is a consequence of pricing algorithms applied on the exchanges, as they aggregate demand and supply on equidistant price levels, where the distance of two consecutive price levels is h. Thus we can interpret the market price as a price observed under aggregation. The aggregation in this form is also called quantization. The loss of information due to quantization is especially high when h is large relatively to the price volatility. This is the case when dealing with high-frequency data, that is, the price process is sampled at a high frequency, 100 times a second, say. The analysis of orderbook data, such as exploring relationships between the price dynamics and order quantities at various price levels, has gained high attention recently.

A scalar quantizer is defined as a mapping q from $\mathbb{R}$ to a discrete, finite or countable set $\mathcal{Y} \subset \mathbb{R}$, representing the so-called quantization levels, assigning to each $x \in \mathbb{R}$ its quantized version

$$y = q(x). \tag{5.1}$$

The simplest scalar quantizer is the uniform quantizer, where the set of quantization levels is given by the integer multiples of a fixed, positive number h, called the sensitivity of the

quantizer, and if x is a real number then we set

$$q(x) = kh \quad \text{for} \quad I_k = \{kh - h/2 < x \leq kh + h/2\}. \tag{5.2}$$

A more realistic model for quantization is a quantizer with saturation, see Brockett and Liberzon (2000), defined as above in the range

$$-(M + 1/2)h < x \leq (M + 1/2)h,$$

with M being a positive integer, and setting $q(x) = \pm Mh$ outside the above interval. Thus there are altogether $2M + 3$ quantization domains, and we will denote them again by I_k, with $k \in K$, where K is the set of possible indices. See Widrow and Kollár (2008) for a recent survey of the statistical theory of quantization.

Motivated by the application described above, in this chapter we consider the regression problem of reconstructing the coefficient vector $\vartheta^* \in \mathbb{R}^d$ of the finite-valued regressor process $(\psi) = \psi_n \in \mathbb{R}^d$ when measured with additive Gaussian noise, followed by quantization. I.e. the observed values are of the form

$$y_n = q(\psi_n^T \vartheta^* + e_n), \tag{5.3}$$

where e_n is an i.i.d. Gaussian sequence with mean 0 and *known variance* $\sigma^2 = (\sigma^*)^2$, see e.g. Masry and Cambanis (1980). Let $J = \{\bar{\psi}_1, \dots \bar{\psi}_M\}$ denote the set ψ_n can take values from. To get rid of overparametrization issues we further assume that J generates $\mathbb{R}^d$ and $P(\psi_n = \bar{\psi}_j > 0)$ for all n and $j = 1 \dots M$. The assumed knowledge of σ^2 may be unrealistic in many applications, but it greatly simplifies the presentation. We shall discuss the possibility of handling unknown σ-s later.

An efficient randomized EM-method to solve the off-line maximum-likelihood estimation problem, based on say N observations, has been developed in Finesso et al. (1999a). In the course of this procedure we generate a sequence of estimators ϑ_t that converge to the off-line maximum likelihood estimator $\hat{\vartheta}_N$ almost surely, under appropriate technical conditions. A real-time version of this method, exhibiting excellent convergence properties, has been developed in Finesso et al. (1999b). In the real-time scheme we generate a sequence of estimators $\hat{\vartheta}_t$ such that $\hat{\vartheta}_t$ converges to ϑ^* almost surely, under appropriate technical conditions.

It is well-known form the theory of stochastic approximation, that, in the case of a weighted stochastic gradient method based on the maximum-likelihood estimation, the

best available asymptotic covariance is the inverse of the Fisher information, see Benveniste et al. (1990) Section 3.2.3. This is achieved by a stochastic Newton-method.

5.2 The EM-method for estimating ϑ^*

Consider first the case of off-line estimation, i.e. when the number of samples N is fixed. For each y in the observation set define the quantization domain $I(y) = \{x : q(x) = y\}$. Write as usual

$$\varphi(x; \psi, \vartheta, \sigma^2) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x - \psi^T \vartheta)^2}{2\sigma^2}},$$

where $\psi = \psi_n$ is the regressor. Then for any ϑ and ψ the ϑ-probability of observing y is, with $\sigma^2 = (\sigma^*)^2$,

$$P(I(y); \psi, \vartheta) = \int_{I(y)} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x - \psi^T \vartheta)^2}{2\sigma^2}} dx = \int_{I(y)} \varphi(x; \psi, \vartheta, \sigma^2) dx. \tag{5.4}$$

Having $\psi^N = (\psi_1, ..., \psi_N)$, for any ϑ the logarithm of the ϑ-probability of observing $y^N = (y_1, ..., y_N)$ is

$$L_N(y^N; \psi^N, \vartheta) = \sum_{n=1}^{N} \log P(I(y_n); \psi_n, \vartheta) = \sum_{n=1}^{N} L(y_n; \psi_n, \vartheta). \tag{5.5}$$

In Figure 5.1 we plot the expected likelihood function against the running parameter ϑ, with σ kept fixed at σ^* (left), and against the running parameter σ with ϑ kept fixed at ϑ^* (right). Two one-dimensional problems are considered: problem I. (solid line) with parameters

$$\vartheta^* = 0.5, \quad {\sigma^*}^2 = 0.08,$$

and problem II. (dashed line) with parameters

$$\vartheta^* = 0, \quad {\sigma^*}^2 = 0.1.$$

In all experiments we assume $h = 1$, and $M = 10$, here we apply the regressor $\psi = 1$ for simplicity. According to these figures the expected likelihood function is likely to be concave with a unique maximum.

The ML estimator $\hat{\vartheta}_N$ is obtained by maximizing $L_N(y^N; \psi^N, \vartheta)$, or solving the likeli-

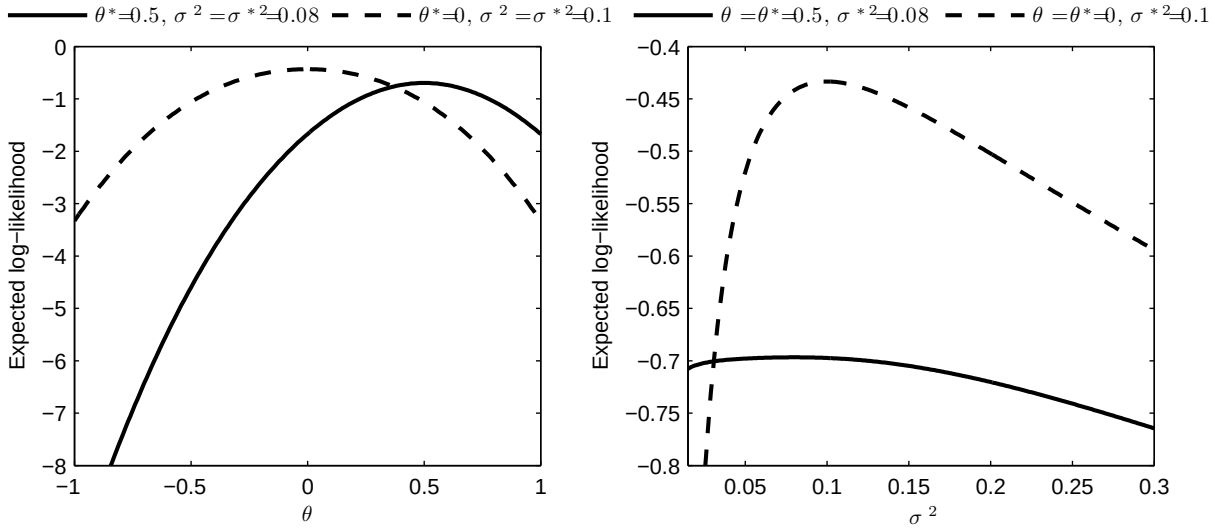

Figure 5.1: The expected log-likelihood function.

hood equation

$$\frac{d}{d\vartheta} L_N(y^N; \psi^N, \vartheta) = 0. \tag{5.6}$$

Introducing the conditional density of x given y

$$\varphi(x \mid y; \psi, \vartheta, \sigma^2) = \frac{\varphi(x; \psi, \vartheta, \sigma^2)}{P(I(y); \psi, \vartheta)} \chi_{I(y)}(x), \tag{5.7}$$

where $\chi_E(x)$ denotes the indicator of the set E, the likelihood equation is equivalent to the following:

The quantized normal equation:

$$\left(\sum_{n=1}^{N} \psi_n \psi_n^T \right) \vartheta = \sum_{n=1}^{N} \psi_n \int x \varphi(x \mid y_n; \psi_n, \vartheta, \sigma^2) dx. \tag{5.8}$$

Notice that this equation is non-linear in ϑ. To solve the likelihood equation would require the computation of an integral in each step of the iteration, which is not feasible if ϑ^* is vector-valued.

The EM-method: This difficulty has been circumvented in Finesso et al. (1999a) by using a Markov Chain Monte Carlo ($MCMC$) method for computing the integrals. Since the likelihood of the observations $x_n = \psi_n^T \vartheta^* + e_n$ is easily obtained, and $y_n = q(x_n)$, a natural approach to solve the likelihood equation is to use the EM-method. Following the basic steps of the EM-method we replace the log-likelihood function by an auxiliary

function (see e.g. Dempster et al. (1977); Mclachlan and Krishnan (1996))

$$Q(y;\vartheta,\bar{\vartheta}) = \mathrm{E}_{\bar{\vartheta}}[\log P(X,\vartheta,\sigma^2) \mid y] = \mathrm{E}\,[\log P(X,\vartheta,\sigma^2) \mid y;\ \bar{\vartheta}], \tag{5.9}$$

where $\bar{\vartheta}$ is a current best estimate, and the random variable $X = \psi^T\bar{\vartheta} + e$ is the unknown assumed state given regressor ψ. Thus we have

$$Q(y;\psi,\vartheta,\bar{\vartheta}) = \int_{I(y)} -\log(\sqrt{2\pi}\sigma) - \frac{(x-\psi^T\vartheta)^2}{2\sigma^2}\varphi(x;y,\psi,\bar{\vartheta},\sigma^2)dx. \tag{5.10}$$

For N independent observations we set

$$Q_N(y^N;\vartheta,\bar{\vartheta}) = \sum_{n=1}^{N} Q(y_n;\vartheta,\bar{\vartheta}) = \mathrm{E}\,[\log P(X^N,\vartheta) \mid y^N;\ \bar{\vartheta}], \tag{5.11}$$

where $X_n = \psi_n^T\bar{\vartheta} + e_n$ is the unknown assumed state at time n. The so-called M-step, maximizing Q_N in ϑ, gives an updated estimate that will replace $\bar{\vartheta}$.

To simplify the notations consider now the case of uniform quantization without saturation. Let I_k be the k-th interval: $I_k = \{x : q(x) = kh\}$, and let

$$N_{j,k} = \#\{n : 1 \le n \le N,\ y_n \in I_k,\ \psi_n = \bar{\psi}_j\} \tag{5.12}$$

be the number of times that kh is observed in the sequence y^N and at the same time the regressor takes the value $\bar{\psi}_j$. Then the M-step is equivalent to solving the *linear* equation

$$\frac{d}{d\vartheta}Q_N(y^N;\psi^N,\vartheta,\bar{\vartheta}) = \sum_j\sum_k N_{j,k}\int_{I_k}\frac{\bar{\psi}_j(x-\bar{\psi}_j^T\vartheta)}{\sigma^2}\,\varphi(x \mid kh;\bar{\psi}_j,\bar{\vartheta},\sigma^2)dx = 0. \tag{5.13}$$

Note that all information on the data is now contained in the counting numbers $N_{j,k}$. We can write (5.13) as

$$\sum_j\sum_k N_{j,k}\,\bar{\psi}_j\bar{\psi}_j^T\vartheta\int_{I_k}\varphi(x \mid kh;\bar{\psi}_j,\bar{\vartheta},\sigma^2)dx = \sum_j\sum_k N_{j,k}\int_{I_k}\bar{\psi}_j x\,\varphi(x \mid kh;\bar{\psi}_j,\bar{\vartheta},\sigma^2)dx.$$

Taking into account that the integral of a density function is one, we arrive at the following updating formula:

The M-step:

$$\Psi_N \vartheta = \sum_{j,k} N_{j,k} \bar{\psi}_j \int_{I_k} x\, \varphi(x \mid kh; \bar{\psi}_j, \overline{\vartheta}, \sigma^2) dx, \tag{5.14}$$

where

$$\Psi_N = \left(\sum_{n=1}^{N} \psi_n \psi_n^T \right).$$

In the course of the EM-method we set $\bar{\vartheta} = \vartheta_t$, and we get $\vartheta = \vartheta_{t+1}$.

Basic inequalities. The basic inequality connecting the likelihood function and the Q-function is the following: for any y and for given fixed $\bar{\vartheta}$ we have for any ϑ

$$L(y, \vartheta) \geq Q(y; \vartheta, \bar{\vartheta}) + D(\bar{\vartheta} || \vartheta) + H(\bar{\vartheta}), \tag{5.15}$$

where $D(\bar{\vartheta}||\vartheta) \geq 0$ for all ϑ. (In fact $D(\bar{\vartheta}||\vartheta)$ is a divergence between two conditional probability densities, and $H(\bar{\vartheta})$ is an entropy, which depends only on $\bar{\vartheta}$.) It follows that the function $L(y, \vartheta) - Q(y; \vartheta, \bar{\vartheta})$ is minimized at $\vartheta = \bar{\vartheta}$, thus, if $\bar{\vartheta}$ is interior relative to the the parameter domain then, we have for any N, y^N, ψ^N

$$Q'_N(y^N; \psi^N, \vartheta, \vartheta) = \frac{\partial}{\partial \vartheta} Q_N(y^N; \psi^N, \vartheta, \bar{\vartheta})_{|\bar{\vartheta}=\vartheta} = \frac{\partial}{\partial \vartheta} L(y^N; \psi^N, \vartheta). \tag{5.16}$$

It follows that the solution of the likelihood equation $\frac{\partial}{\partial \vartheta} L_N(y^N; \psi^N, \vartheta) = 0$ is obtained by solving the equation

$$Q'_N(y^N; \psi^N, \vartheta, \vartheta) = 0. \tag{5.17}$$

Asymptotic log-likelihood. Assuming ergodic regressors, $\bar{N}_{j,k} = \lim_{N\to\infty} \frac{N_{j,k}(N)}{N}$ has a limit for every $j \in J$ (here we made the dependence of $N_{j,k}$ on N explicit). Then, using $Y_j = q(\bar{\psi}_j^T \vartheta^* + e)$, the asymptotic log-likelihood function can be written in the following form:

$$\begin{aligned}
\bar{L}(\vartheta) &= \lim_{N\to\infty} \frac{1}{N} L_N(y^N, \vartheta) = & (5.18)\\
&= \lim_{N\to\infty} \frac{1}{N} \sum_{n=1}^{N} \log P(I(y_n), \vartheta) = & (5.19)\\
&= \lim_{N\to\infty} \frac{1}{N} \sum_{j\in J} \sum_{k\in K} N_{j,k} \log P(I(k); \bar{\psi}_j, \vartheta) = & (5.20)\\
&= \sum_{j\in J} \sum_{k\in K} \bar{N}_{j,k} \log P(I(k); \bar{\psi}_j, \vartheta). & (5.21)
\end{aligned}$$

Note that we have that

$$\frac{\partial}{\partial\vartheta}\bar{L}(\vartheta)_{|\vartheta=\vartheta^*} = \sum_{j,k}\bar{N}_j P(I(k);\bar{\psi}_j,\vartheta^*)\frac{\frac{\partial}{\partial\vartheta}P(I(k);\bar{\psi}_j,\vartheta)_{|\vartheta=\vartheta^*}}{P(I(k);\bar{\psi}_j,\vartheta^*)} = \tag{5.22}$$

$$= \sum_j \bar{N}_j \frac{\partial}{\partial\vartheta}\sum_k P(I(k);\bar{\psi}_j,\vartheta)_{|\vartheta=\vartheta^*} = \sum_j \bar{N}_j\cdot 0 = \tag{5.23}$$

$$= 0, \tag{5.24}$$

where $\bar{N}_j = \sum_k \bar{N}_{j,k} = \lim N_j/N$, $N_j = \#\{n : 1 \le n \le N,\ \psi_n = \bar{\psi}_j\}$.

Similarly, define the asymptotic Q-function, with $X_j = \psi_j^T\vartheta^* + e$ and $Y_j = q(X_j)$, as

$$\bar{Q}(\vartheta,\bar{\vartheta}) = \sum_{j,k}\bar{N}_{j,k}\ Q(Y_j = kh,\vartheta,\bar{\vartheta}) = \tag{5.25}$$

$$= \sum_{j,k}\bar{N}_{j,k}\ \mathrm{E}\ [\log P(X_j,\vartheta)|Y_j = kh,\bar{\vartheta}]. \tag{5.26}$$

To relate the asymptotic conditional Q function to the asymptotic conditional likelihood function the simplest, although formal, procedure is to divide both sides of (5.16) by N, and take limit to get

$$\bar{Q}'(\vartheta,\vartheta) = \frac{\partial}{\partial\vartheta}\bar{Q}(\vartheta,\bar{\vartheta})_{|\bar{\vartheta}=\vartheta} = \frac{\partial}{\partial\vartheta}\bar{L}(\vartheta). \tag{5.27}$$

Thus the asymptotic problem of determining ϑ^* can be formulated, as solving the equation

$$\bar{Q}'(\vartheta,\vartheta) = 0. \tag{5.28}$$

This equation could be derived directly, but the context of the EM-method gives a convenient computational framework that will be exploited subsequently.

The asymptotic Fisher information. The asymptotic Fisher information for the problem of estimating ϑ^* with known $\sigma = \sigma^*$ will be denoted

$$I^* = -\frac{\partial^2}{\partial\vartheta^2}\bar{L}(\vartheta)|_{\vartheta=\vartheta^*}. \tag{5.29}$$

It is well-known that it can be expressed via the score function

$$\frac{\partial}{\partial\vartheta}L(y_n;\vartheta) = \int_{I(y_n)}\frac{\psi_n(x-\psi_n^T\vartheta)}{\sigma^2}\,\varphi(x\mid y_n;\psi_n,\vartheta,\sigma^2)dx \tag{5.30}$$

as

$$I^* = \lim_{N\to\infty} \frac{1}{N} \sum_{n=1}^{N} \left(\frac{\partial}{\partial \vartheta} L(y_n; \vartheta) \right) \left(\frac{\partial}{\partial \vartheta} L(y_n; \vartheta) \right)^T |_{\vartheta=\vartheta^*}. \tag{5.31}$$

Equivalently, we can write, taking into account (5.16),

$$I^* = \lim_{N\to\infty} \frac{1}{N} \sum_{n=1}^{N} Q'(y_n; \vartheta^*, \vartheta^*)\ Q'(y_n; \vartheta^*, \vartheta^*)^T, \tag{5.32}$$

or as

$$I^* = \sum_{j\in J} \sum_{k\in K} \bar{N}_{j,k} \frac{\bar{\psi}_j \bar{\psi}_j^T}{\sigma^4} \left(\int_{I(k)} \left(x - \bar{\psi}_j^T \vartheta^* \right) \varphi(x|kh, \bar{\psi}_j, \vartheta^*, \sigma^2) dx \right)^2. \tag{5.33}$$

Note that on the limiting case, when h tends to 0, we get

$$I^* = \frac{\psi\psi^T}{\sigma^2} = \frac{\psi\psi^T}{(\sigma^*)^2},$$

as expected. It is easy to see that in any case the loss in information due to quantization decreases the Fisher information, i.e

$$I^* \leq \frac{\psi\psi^T}{\sigma^2}. \tag{5.34}$$

5.3 A randomized EM-method

The integrals on the right hand side of (5.14) are expectations with respect to a conditional Gaussian density, and it is therefore natural to approximate them using a Markov Chain Monte Carlo (MCMC) algorithm, see Hammersley and Handscomb (1967); Metropolis et al. (1953). A combination of the latter with the EM-algorithm leads to a stochastic approximation scheme called a randomized EM-method, first presented in Finesso et al. (1999a,b)). A similar method has been developed independently for the problem of log-linear regression in Solo (1999).

The MCMC method. Thus to compute $\int_{I_k} x\, \varphi(x \mid kh, \bar{\psi}_j; \bar{\vartheta}, \sigma^2)\ dx$ we generate an ergodic Markov chain $\bar{\xi}_t^{j,k}(\bar{\vartheta})$ on the state-space I_k, which is an interval of length h (in case of no saturation), such that its invariant measure is $\varphi(x \mid kh, \bar{\psi}_j; \bar{\vartheta}, \sigma^2)$ or $\varphi(x \mid I_k, \bar{\psi}_j; \bar{\vartheta}, \sigma^2)$. For this purpose we use the Metropolis-Hastings method, with un-normalized target density

$$\tau(x) = \tau(x, \bar{\vartheta}) = e^{-(x-\bar{\psi}_j^T \bar{\vartheta})^2/(2\sigma^2)} \chi_{I(y_k)}(x).$$

Let the initial transition kernel for the Metropolis-Hastings method be $q(x, y) = 1/h$ for all $x, y \in I(y_k)$, i.e. let the initial chain be simply an i.i.d. sequence with uniform distribution. Then we have a classic Metropolis algorithm defined by the acceptance probabilities

$$\alpha(x, y; \bar{\vartheta}) = \min\left\{\frac{\tau(y, \bar{\vartheta})}{\tau(x, \bar{\vartheta})}, 1\right\} = \min\left\{e^{\frac{-(y-x)(y+x-2\bar{\psi}^T\bar{\vartheta})}{2\sigma^2}}, 1\right\}. \tag{5.35}$$

For the generation of $\bar{\xi}_\ell^{j,k}(\bar{\vartheta})$ we will need an i.i.d sequence of random vectors $(U_l, V_l), l \geq 1$ with uniform distribution on $[0, 1] \times [0, 1]$, independent also of the initial state $\bar{\xi}_0^{j,k}(\bar{\vartheta}) = \bar{\xi}_0^{j,k}$. The first component, U_l, is used to generate the next sate of the initial chain, while the second component, V_l, is used to realize the acceptance or rejection. We will thus use the following shorthand notation for the generation of $\bar{\xi}_\ell^{j,k}(\bar{\vartheta})$:

The frozen parameter Markov chain on I_k with regressor $\bar{\psi}_j$:

$$\bar{\xi}_{\ell+1}^{j,k}(\bar{\vartheta}) = F(\bar{\xi}_\ell^{j,k}(\bar{\vartheta}), U_{l+1}, V_{l+1}; \bar{\vartheta}). \tag{5.36}$$

Here F depends on $\bar{\vartheta}$ via the acceptance probability $\alpha(x, y; \bar{\vartheta})$.

Let K_N be the set of indices of quantization domains that show up in the observation sequence of length N, let J_N be the set of indices of regressor values that show up in the regressor sequence of length N. Let $k \in K_N$, $j \in J_N$ and let the current state of the corresponding Markov chain be $\bar{\xi}_\ell^{j,k}(\bar{\vartheta})$. Then for large L a good approximation of (5.14) is given by

$$\psi_N \vartheta = \sum_{k \in K_N} \sum_{j \in J_N} N_{j,k}\, \bar{\psi}_j\, \frac{1}{L} \sum_{\ell=1}^{L} \bar{\xi}_\ell^{j,k}(\bar{\vartheta}). \tag{5.37}$$

Allowing time-variation. When the above approximation is applied in an EM-iteration it is reasonable to run the EM-algorithm and the $MCMC$ method in parallel. Let us now write $\bar{\vartheta} = \vartheta_t$, with ϑ_t still to be specified, and consider the *time-varying* Markovian dynamics

$$\xi_{t+1}^{j,k} = F(\xi_t^{j,k}, U_{t+1}, V_{t+1}; \vartheta_t). \tag{5.38}$$

Here ϑ_t is the current approximation of the the maximum-likelihood estimator $\hat{\vartheta}_N$, which

is in turn updated by an approximation of (5.37) as follows:

$$\Psi_N \vartheta_{t+1} = \sum_{k \in K_N} \sum_{j \in J_N} N_{j,k}\, \bar{\psi}_j \, \frac{1}{t+1} \sum_{m=1}^{t+1} \xi_m^{j,k}. \tag{5.39}$$

The above algorithm, defined by (5.38) and (5.39) is called a randomized EM-method. The following simple derivation yields a recursive equation for (5.39).

$$\begin{aligned}
\Psi_N \vartheta_{t+1} &= \sum_{j,k} N_{j,k}\, \bar{\psi}_j \, \frac{1}{t+1} \sum_{m=1}^{t+1} \xi_m^{j,k} = && (5.40)\\
&= \frac{t}{t+1} \sum_{j,k} N_{j,k} \bar{\psi}_j \frac{1}{t} \sum_{m=1}^{t} \xi_m^{j,k} + \frac{1}{t+1} \sum_{j,k} N_{j,k} \bar{\psi}_j \xi_{t+1}^{j,k} = && (5.41)\\
&= \frac{t}{t+1} \Psi_N \vartheta_t + \frac{1}{t+1} \sum_{j,k} N_{j,k} \bar{\psi}_j \xi_{t+1}^{j,k} = && (5.42)\\
&= \Psi_N \vartheta_t \left(1 - \frac{1}{t+1}\right) + \frac{1}{t+1} \sum_{j,k} N_{j,k} \bar{\psi}_j \xi_{t+1}^{j,k} = && (5.43)\\
&= \Psi_N \vartheta_t + \frac{1}{t+1} \sum_{j,k} N_{j,k} \left(\bar{\psi}_j \xi_{t+1}^{j,k} - \Psi_N \vartheta_t\right). && (5.44)
\end{aligned}$$

Multiplying with Ψ_N^{-1} from the left yields the following equation :
A randomized EM-method:

$$\vartheta_{t+1} = \vartheta_t + \frac{1}{t+1} \sum_{k \in K_N} \sum_{j \in J_N} N_{j,k} \left(\Psi_N^{-1} \bar{\psi}_j \xi_{t+1}^{j,k} - \vartheta_t\right). \tag{5.45}$$

Let us stress again that the number of observations is fixed, and ϑ_t is expected to converge to $\hat{\vartheta}_N$, rather than to ϑ^*.

5.4 A real-time recursive randomized EM-method

Consider now the situation when the N is not fixed, instead we have a flow of data, and for each new measurement the estimator of ϑ^* will be updated. Since, for increasing N, every integer k will eventually occur in the observation sequence, we would need to generate an infinite number of Markov-chains. This is not practical. Hence we confine ourselves to the case of quantization with saturation. The price of this is that the state space is non-compact, and the generation of the $MCMC$ method below and above the saturation

level requires extra care. We mitigate this problem by choosing a fairly wide saturation interval, so that the probability of the state to become below or above the saturation level is negligible. The quantization intervals will be denoted by I_k as before with $k \in K$. If $|K|$ is large then it is unreasonable to update all the Markovian states at all time. Instead, at any time T, we update a single Markov chain, say $\bar{\xi}^{j,k}(\bar{\vartheta})$, where $k = k_T$ is the index of the current observation and $j = j_T$ is the index of the current regressor value.

The first step is to modify the approximation to the M-step (5.37) so as to take into account the real time T. Let $N_{j,k,T}$ denote the number of visits to the domain I_k for a given regressor value $\bar{\psi}_j$ up to time $T = N$, i.e. set

$$N_{j,k,T} = \#\{n : \ x_n \in I_k, \psi_n = \bar{\psi}_j, n \leq T\}. \tag{5.46}$$

A convenient and reasonable approximation of (5.37) is obtained if we set $L = N_{j,k,T}$ for the quantization domain I_k and regressor value $\bar{\psi}_j$, namely then (5.37) reduces to:

The M-step for increasing sample size:

$$\Psi_T \vartheta = \sum_{j \in J} \sum_{k \in K} \sum_{t=1}^{N_{j,k,T}} \bar{\psi}_j \bar{\xi}_t^{j,k}(\bar{\vartheta}). \tag{5.47}$$

Synchronization. To synchronize the internal times of the individual Markov chains $\bar{\xi}_t^{j,k}(\bar{\vartheta})$ let us define, for each j, k, a new, piecewise constant extension of $\bar{\xi}_t^{j,k}(\bar{\vartheta})$ as follows: first let $Z_t^{j,k}$ be the indicator of the event $(x_t \in I_k) \cap (\psi_t = \bar{\psi}_j)$, i.e.

$$Z_t^k = \chi_{I_k}(x_t) \cdot \chi_{\{\psi_t = \bar{\psi}_j\}}.$$

Define the new Markov chain $\bar{\xi}_t^{\circ,j,k} = \bar{\xi}_t^{\circ,j,k}(\bar{\vartheta})$ so that it stays constant at any time t, unless $x_t \in I_k$ and $\psi_t = \bar{\psi}_j$, and then follows the dynamics of $\bar{\xi}_t^{j,k}(\bar{\vartheta})$. Thus we get:

$$\bar{\xi}_{t+1}^{\circ,j,k} = Z_t^{j,k} F(\bar{\xi}_t^{\circ,j,k}, U_{t+1}, V_{t+1}; \bar{\vartheta}) + (1 - Z_t^{j,k}) \bar{\xi}_t^{\circ,j,k}. \tag{5.48}$$

Let the initial condition be $\bar{\xi}_0^{\circ,j,k} = \bar{\xi}_0^{j,k}$. Then $(\bar{\xi}_t^{\circ,j,k}, Z_t^{j,k})$ is a Markov-process for each j, k, and so is

$$(\bar{\xi}_t^{\circ}, Z_t) = (\bar{\xi}_t^{\circ,j,k}, Z_t^{j,k}), \quad k \in K, j \in J.$$

Also, the processes $(\bar{\xi}_t^{\circ,j,k}, Z_t^{j,k})$ are independent as k and j vary. Thus we can write the M-step (5.47) as

The M-step for the synchronized Markov-chain:

$$\Psi_T\vartheta = \sum_{j\in J}\sum_{k\in K}\sum_{t=1}^{T} Z_t^{j,k}\bar{\xi}_t^{\circ j,k}(\bar{\vartheta})\psi_t, \tag{5.49}$$

Our goal in this section is to develope an on-line recursive quasi maximum likelihood estimation algorithm, which takes the following general form:

$$\hat{\vartheta}_{t+1} = \hat{\vartheta}_t - \frac{1}{t}\hat{H}_{t+1}^{-1}g_{t+1}, \tag{5.50}$$

$$\hat{H}_{t+1} = \hat{H}_t + \frac{1}{t}\left(H_{t+1} - \hat{H}_t\right), \tag{5.51}$$

where g_t, H_t are on-line approximations of the gradient and the Hessian of asymptotic negative likelihood function, respectively. In our case the EM-method is involved in the maximum likelihood procedure, thus we consider first the derivatives of $\bar{Q}(\vartheta,\bar{\vartheta})$; recall that

$$\frac{\partial}{\partial\vartheta}\bar{Q}(\vartheta,\vartheta) = \bar{Q}'(\vartheta,\vartheta) = \frac{\partial}{\partial\vartheta}\bar{L}(\vartheta).$$

Differentiating (5.26) with respect to ϑ we get

$$-\frac{\partial}{\partial\vartheta}\bar{Q}(\vartheta,\bar{\vartheta}) = \sum_{j,k}\bar{N}_{j,k}\frac{\bar{\psi}_j}{\sigma^2}\int_{I(k)}(x-\bar{\psi}_j^T\vartheta)\varphi(kh;\bar{\psi}_j,\bar{\vartheta},\sigma^2)dx. \tag{5.52}$$

It is easy to see that in case of a stationary regressor we have $\mathrm{E}Z_t^{j,k} = \bar{N}_{j,k}$. In addition, assuming stationary initialization for $\bar{\xi}_t^{\circ j,k}(\bar{\vartheta})$, the integral in (5.52) equals

$$\mathrm{E}\left(\bar{\xi}_t^{\circ j,k}(\bar{\vartheta}) - \bar{\psi}_j^T\vartheta\right).$$

Thus, let us define for each t

$$G_t(\vartheta,\bar{\vartheta}) = \sum_{j\in J}\sum_{k\in K} Z_t^{j,k}\frac{\bar{\psi}_j}{\sigma^2}(\bar{\xi}_t^{\circ,j,k}(\bar{\vartheta}) - \vartheta) \tag{5.53}$$

for which we then have that

$$-\frac{\partial}{\partial\vartheta}\bar{Q}(\vartheta,\bar{\vartheta}) = \mathrm{E}G_t(\vartheta,\bar{\vartheta}). \tag{5.54}$$

To get a real-time randomized EM-method we proceed in the usual manner: let $\hat{\vartheta}_t$ be

the estimate of ϑ^* at time t. Then generate the next state of a *non-homogeneous* Markov chain $(\xi_{t+1}^{\circ,j,k})$ by

$$\xi_{t+1}^{\circ,j,k} = Z_t^{j,k} F(\xi_t^{j,k}, U_{t+1}, V_{t+1}; \hat{\vartheta}_t) + (1 - Z_t^{j,k})\xi_t^{\circ,j,k}. \tag{5.55}$$

To update $\hat{\vartheta}_t$ we use a stochastic Newton method to maximize $\bar{L}(\vartheta)$. First we estimate the gradient $\frac{\partial}{\partial\vartheta}\bar{L}(\hat{\vartheta}_t) = \bar{Q}'(\hat{\vartheta}_t, \hat{\vartheta}_t)$ by $G_{t+1}(\hat{\vartheta}_t, \hat{\vartheta}_t)$, which in turn is estimated on-line, see (5.53), by

$$g_{t+1} = -\sum_{j\in J}\sum_{k\in K} Z_{t+1}^{j,k}\frac{\bar{\psi}_j}{\sigma^2}(\xi_{t+1}^{\circ,j,k} - \bar{\psi}_j^T\hat{\vartheta}_t) = -\frac{\psi_{t+1}}{\sigma^2}(\xi_{N_{j',k',t+1}}^{j',k'} - \psi_{t+1}^T\hat{\vartheta}_t), \tag{5.56}$$

where $k' = k'_{t+1}$ and $j' = j'_{t+1}$ are the indexes observed at time $t+1$. The definition of the non-homogenous Markov chain $\xi_t^{j,k}$ is self-explanatory.

Recall that H_t is specified such that its real-time average $\hat{H}_t$ approximates the asymptotic Fisher information matrix I^*, see (5.31). Using similar arguments as for (5.54) we can easily see that, assuming stationary initialization for $\bar{\xi}_t^{\circ j,k}(\vartheta^*)$, we have

$$I^* = \mathrm{E}\left(G_t(\vartheta^*, \vartheta^*)G_t^T(\vartheta^*, \vartheta^*)\right).$$

Thus we first estimate $G_t(\vartheta^*, \vartheta^*)G_t^T(\vartheta^*, \vartheta^*)$ by $G_t(\hat{\vartheta}_t, \hat{\vartheta}_t)G_t^T(\hat{\vartheta}_t, \hat{\vartheta}_t)$, which is in turn estimated on-line by

$$H_t = g_t g_t^T. \tag{5.57}$$

In summary, the real-time stochastic Newton method is given by the equations (5.50)-(5.51), (5.56)-(5.57).

The ideas behind this algorithm are similar to those presented in Finesso et al. (2000), but without justification for its convergence. The above derivation lends to a direct application of the BMP theory, see Gerencsér et al. (2008a).

5.5 Estimating the variance

Let us now return to our original model $y_n = q(\psi_n^T\vartheta^* + e_n)$ and consider now the case when σ^*, the variance of the additive noise is unknown. It is easy to see that the M-step of the

EM-method leads to the following updating formulas

$$\Psi_N \vartheta = \sum_{j\in J}\sum_{k\in K} N_{j,k}\bar{\psi}_j \int_{I_k} x\,\varphi(x \mid kh; \bar{\psi}_j, \overline{\vartheta}, \overline{\sigma}^2)dx, \tag{5.58}$$

$$\sigma^2 = \sum_{j\in J}\sum_{k\in K} \frac{N_{j,k}}{N} \int_{I_k} (x-\vartheta)^2\,\varphi(x \mid kh; \bar{\psi}_j, \overline{\vartheta}, \overline{\sigma}^2)dx. \tag{5.59}$$

Notice in (5.58) that ϑ does not depend on σ, thus we can solve the above equations successively. Then in analogy with the estimation of the regression coefficient, we arrive at a real-time, randomized EM-method, in which (5.50)- (5.51), (5.56)- (5.57) are extended with the following gradient method estimating σ^*:

$$\hat{\sigma}^2_{t+1} = \hat{\sigma}^2_t + \frac{1}{t+1}((\xi^{k'}_{N_{k',t+1}} - \hat{\vartheta}_{t+1})^2 - \hat{\sigma}^2_t), \tag{5.60}$$

where the dynamics of the time-varying Markov-chain now depends both on $\hat{\vartheta}_t$ and $\hat{\sigma}^2_t$, and $\hat{\sigma}^2_{t+1}$ is applied in (5.50)- (5.51), (5.56)- (5.57).

To convert the above procedure into a fully stochastic Newton-method we need to estimate the $(d+1)\times(d+1)$ Fisher information matrix, say I^*, which now contains elements related to $\hat{\sigma}^2$ in addition, in row $d+1$ and column $d+1$, say. In order to examine these additional elements we have carried out numerical experiments to calculate the 2×2 Fisher information matrix $\check{I}(\eta,\sigma)$ in the simplified model

$$y_n = q(\eta + e_n), \quad e_n \sim \mathcal{N}(0,\sigma^2) \quad \text{i.i.d.},$$

with the scalar location parameter $\eta\in\mathbb{R}$, see Gerencsér et al. (2008a). We found that the off-diagonal elements of $\check{I}(\eta,\sigma)$ are zero. From this finding and the chain rule it follows that $I^*_{d+1,i} = I^*_{i,d+1} = 0$, $i = 1\dots d$, thus we only need to estimate the scalar Fisher information $I^*_{d+1,d+1} = (i^\sigma)^*$ to get a fully stochastic Newton-method. The estimation of $(i^\sigma)^*$ can be carried out along the lines described above for the Fisher infromation with respect to parameter ϑ. We can summarize the algorithm as follows:

Real-time stochastic Newton-method estimating ϑ^* and σ^*:

$$\begin{aligned}
\hat{\vartheta}_{t+1} &= \hat{\vartheta}_t + \tfrac{1}{t}\hat{H}^{-1}_{t+1} g_{t+1}, & \hat{\sigma}^2_{t+1} &= \hat{\sigma}^2_t + \tfrac{1}{t} g^\sigma_{t+1}/\hat{i}^\sigma_{t+1},\\
\hat{H}_{t+1} &= \hat{H}_t + \tfrac{1}{t}\left(g^T_{t+1}g_{t+1} - \hat{H}_t\right), & \hat{i}^\sigma_{t+1} &= \hat{i}^\sigma_t + \tfrac{1}{t}\left((g^\sigma_{t+1})^2 - \hat{i}^\sigma_t\right),\\
g_{t+1} &= \psi_{t+1}(\xi^{j',k'}_{N_{j',k',t+1}} - \psi^T_{t+1}\hat{\vartheta}_t)/\hat{\sigma}^2_t, & g^\sigma_{t+1} &= (\xi^{j',k'}_{N_{j',k',t+1}} - \psi^T_{t+1}\hat{\vartheta}_t)^2 - \hat{\sigma}^2_t.
\end{aligned}$$

To save computational time for the inversion of $\hat{H}$, we can apply the Matrix Inversion Lemma along the lines of the related derivation in the GARCH case in Section 2.5.

5.6 Numerical experiments

To demonstrate the viability of our method we present a simulation experiment. In the test problem we have $d = 2$, the true parameters are

$$\vartheta^* = \begin{pmatrix} -0.9 \\ 1.1 \end{pmatrix}, \quad \sigma^* = 0.3.$$

We generated the regressor process ψ_t according to the following model: ψ_t are independent random variables distributed uniformly randomly over $J = \{\bar{\psi}_1, \bar{\psi}_2\}$ with

$$\bar{\psi}_1 = \begin{pmatrix} 0.4 \\ 0.6 \end{pmatrix}, \quad \bar{\psi}_2 = \begin{pmatrix} 0.7 \\ 0.3 \end{pmatrix}.$$

Figure 5.2 depicts $\hat{\vartheta}_t$ and $\hat{\sigma}_t^2$ estimated recursively from simulated data of length $N = 5{\cdot}10^5$. The figure shows that $\hat{\vartheta}_t$ and $\hat{\sigma}_t^2$ converge nicely to ϑ_t^* and $(\sigma^*)_t^2$, respectively.

For comparison purposes we also calculated the least squares estimator $\hat{\vartheta}^\circ$:

$$\hat{\vartheta}^\circ = \left(\sum_{n=1}^{N} \psi_n \psi_n^T \right)^{-1} \sum_{n=1}^{N} \psi_n^T y_n = \begin{pmatrix} -0.7459 \\ 0.9136 \end{pmatrix}.$$

The off-line least squares estimator shows a significant bias

$$\vartheta^* - \hat{\vartheta}^\circ = \begin{pmatrix} -0.2541 \\ 0.1864 \end{pmatrix},$$

because it does not take quantization effects into account (the variance of $\hat{\vartheta}^\circ$ is less then 0.0001 when $N = 5{\cdot}10^5$). Note that this bias does not decrase by increasing the sample size. The bias of the off-line least squares estimator induced by quantization mainly depends on the magnitude of σ^*/h: the bias decreases as σ^*/h increases.

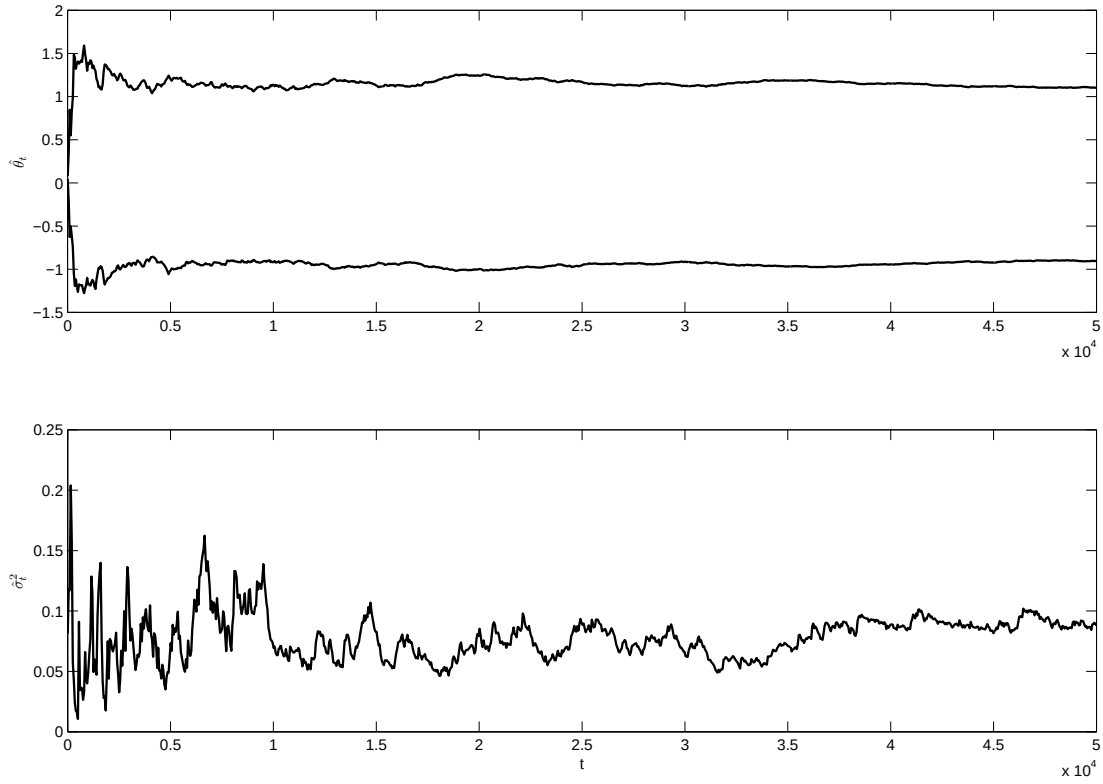

Figure 5.2: The convergence of the estimators $\hat{\vartheta}_t$ and $\hat{\sigma}_t^2$.

5.7 Conclusion

In this chapter we presented a recursive algorithm for estimating the coefficients of a linear regression from quantized observations. Numerical experiments indicate that the bias of the standard linear regression estimator can be significant, while our method yields unbiased estimators.

Another important model in high-frequency finance is the following:

$$x_{t+1} = \mu^* + x_t + e_{t+1}, \quad e_{t+1} \sim \mathcal{N}(0, (\sigma^*)^2) \quad \text{i. i. d.}$$

This model can be seen as the discrete time version of the well-known Brownian motion, where μ^* controls the trend, σ^* controls the volatility. An interesting (and more involved) problem of future research is to recursively estimate (μ^*, σ^*) from quantized observations, i.e. when

$$y_{t+1} = q(x_{t+1}).$$

Bibliographical remarks

The algorithm presented in this chapter is an adaptation of the algorithm developed for estimating the parameters of a Gaussian random variable from quantized observations, published in Communications in Information and Systems, see Gerencsér et al. (2008a), to the quantized linear regression problem.

The original algorithm relies on the ideas of László Gerencsér and Lorenzo Finesso, numerical investigations carried out by Ildikó Kmecs have been reproduced and extended by Balázs Torma. The algorithm has been extended with the estimation of the variance parameter by Balázs Torma. The adaptation of the original algorithm to the linear regression problem presented in this chapter has been the work of Balázs Torma, initiated by László Gerencsér.

Appendix A

Transaction requests

In this appendix we provide some more technical insight on agents' order placing rules. According to the belief $\hat{p}_t$ the agent calculates his transaction request as follows. Recall first our trading protocol, in which the agent has to submit orders before he knows at which price p_t he will actually transact. In the following, we derive simple order placing rules to ensure the rational trading goal that under the assumption of exact prediction ($p_{t+1} = \hat{p}_t$) and self-financing portfolios, the transaction results in a higher wealth:

$$B_t + \hat{p}_t S_t \geq B_{t-1} + \hat{p}_t S_{t-1}, \tag{A.1}$$

where B_t, S_t denotes cash and stock amounts, respectively. We can rewrite condition (A.1) as

$$\begin{aligned} B_{t-1} - d_t p_t + (S_{t-1} + d_t)\,\hat{p}_t - (B_{t-1} + S_{t-1}\hat{p}_t) = \\ = d_t(\hat{p}_t - p_t) \geq 0, \end{aligned} \tag{A.2}$$

where it is worth recalling that the transaction takes place at price p_t. Unsurprisingly, inequality (A.2) suggests that the agent should buy if the market price turns out to be less than the belief, or the agent should sell if the opposite is true.

It is quite easy to achieve this using limit orders. One only has to place a limit buy order one cent below $\hat{p}$ and simultaneously place a limit sell order one cent above $\hat{p}$. Then, if $\hat{p} > p$, only the buy order executes, and if $\hat{p} < p$, only the sell order executes, hence (A.2) is satisfied. If p happens to equal $\hat{p}$, the agent does not transact. The working mechanism of this order placing strategy is similar to the "straddle" strategy widely used by option traders.

The last component of our transaction request model is the concept of trading positions. Opening or entering a long (short) position is a term used in practice for buying (selling) stocks that the trader wants later to sell (buy or rebuy). A trader closes or exits a position by transacting in the direction opposite to the opening transaction of that position. By closing a position the trader realizes the book profit or loss he made on the position so far. The aggregated effect of opening and closing transactions of a given position is zero on the stock amount the trader holds, because the two transactions cancel out each other. Assume for example, that the agent has closed all his positions by the time t^+. Then $S_{t^+} = S_0$, i.e. he has the same amount of stocks on the brokerage account as after the initial endowment.

To complete the description of the order placing rules we have yet to specify the order quantities. In our model, every trading position contains exactly one stock. Thus, at time t^-, the agent has

$$|S_{t^-} - S_0| \tag{A.3}$$

open positions, the sign of the difference indicating whether long (+) or short (−) open positions. Myopic agents would like to close open positions as soon as possible, which they try to achieve by extending the demand in the order of the appropriate direction: an agent increases the quantity in the sell order if he has a surplus in shares or he increases the quantity in the buy order in case of shortfalls. Thus, the transaction request of agent i, $1 \leq i \leq N$ consists of $N_i \equiv 2$ orders

$$\begin{aligned}
(d,b)_t^{i,1} &= \left(1 + \left(S_{t^-}^i - S_0^i\right)^-, \hat{p}_t^i - 0.01\right), \\
(d,b)_t^{i,2} &= \left(-1 - \left(S_{t^-}^i - S_0^i\right)^+, \hat{p}_t^i + 0.01\right).
\end{aligned}$$

Appendix B

The artificial stock market simulator

In this appendix we briefly describe the simulator program we developed for investigating the agent-based model presented in Section 2.2. It deserves an introductory review because it has been designed as a general, easily customizable object-oriented framework capable of hosting other orderbook-level ACF models.

The program is written in the Java programing language. We next outline the main functionalities along with the main components of the framework; related basis classes are indicated in italics.

1. Market Exchange: The market *Exchange* gathers the orders of trading agents in the *OrderBook* corresponding to the traded stock. It uses *ClearingAlgorithm* to match the orders of agents and books the transactions on agents' accounts.

2. Trading Agent: Trading strategies are implemented in classes of type *Behavior* by the method *determineDemand*, which submits a single or several *Order*s to the market exchange. The class *TradingAccount* keeps track of the stock and cash balance of the agents and also checks budget constraints before trading.

3. Data acquisition: All market entities (orderbook, behaviors, accounts) use a unified concept to gather data for analysis. In each trading period, classes inherited from *SimuComponent* record a snapshot of the state of the market and other trading related variables, e.g account balances, order quantities, limit prices, trading profits, stock price, trading volume, bid-ask spread. Time series of these variables are available for analysis after the simulation.

4. Market setup: Parameter settings, such as for example the length of simulation horizon, behavior parameters and the market fractions of different agent types, initial

endowment of agents can be defined via the class *SimuConf*. In addition, some basic statistics to be calculated on the generated time series can be specified here.

The typical steps of analysis using the program are as follows. First the market setup needs to be defined by creating a class of type *SimuConf*. After compiling and starting the program a simple user interface appears where the user can start the simulation. After running a simulation the result chart appears where various time series can be selected and plotted. Time series can be exported into a plain text file for further analysis in a statistical software environment, such as MATLAB or R. In order to save time, the user can start simulations with different parameter settings simultaneously on a multi-processor machine.

Short summary

In this thesis I propose a detailed ACF model and I present statistical algorithms based on technical (econometric) models. The ACF model is analysed using technical models. The fundamental model extends well-established concepts of agent-based computational finance with a novel element for information arrival processes, which is modelled by a discrete time version of Hawkes processes. I show by numerical experiments that in contrast to classical ACF models, stylized facts emerge even by constant market fractions of chartists and fundamentalists. I further validate the ACF model using a widely accepted technical model, the General Autoregressive Heteroscedastisity (GARCH) model. In particular, a qualitative relationship between the market structure and the best-fitting GARCH(1,1) model is established, which motivates to apply GARCH(1,1) for indirect statistical inference on the market structure. The use of GARCH models for detecting changes in the market structure is justified by a general principle, stating that change detection is feasible using misspecified models. A real-time change detection method for GARCH processes is presented based on the MDL (Minimum Description Length) approach to modelling. I provide an economic interpretation of the GARCH-based change alarms. The performance of the proposed algorithm is evaluated on simulated data. Change-alarms based on real data are reported.

Motivated by the problem of quasi Maximum Likelihood Estimation of GARCH parameters, I propose a new efficient nonlinear optimization method which combines the advantages of descent methods and cutting plane approaches. I also present a technique with which the dimension of the GARCH fitting problem can be reduced.

For modelling the dynamics of information arrival, I propose Hawkes processes in which the feedback path is defined by a finite dimensional linear system. I propose algorithms for the simulation and real-time estimation of this type of Hawkes processes. I show that some parts of the diagonal of the asymptotic Fisher information matrix in case of the one-dimensional feedback go to infinity, other parts of the diagonal go to zero as the parameters approach the boundary of the stability domain. As a first step I calculate the limit distribution of the appropriately rescaled intensity process.

Finally I present a real-time method for estimating the parameters of a linear regression from quantized observations, when the regressor is finite-valued. The algorithm applies Expectation Maximization and Markov Chain Monte Carlo techniques.

Rövid összefoglaló (in hungarian)

Az értekezésben egy új multiágens tőzsdemodellt és matematikailag kezelhető technikai (ökonometriai) modellek alapján fejlesztett statisztikai algoritmusokat mutatok be, illetve ezek segítségével elemzem a multiágens modellt. A fundamentális modell az irodalomban elterjedt multiágens modellek koncepcióit egészíti ki egy új hírfolyamatmodellel, amely egy diszkrét idejű verziója a jól ismert Hawkes-folyamatoknak. Numerikus kísérletekben megmutatom, hogy a modell reprodukálja a piaci idősorokban fellelhető tipikus mintákat, az ún. stilizált tényeket a fundamentalisták és technikai kereskedők fix aránya mellett is, szemben klasszikus modellekkel. A tőzsdemodell validálásának egy további lépéseként a generált árfolyamatot egy széles körben elfogadott ökonometriai modell, a General Autoregressive Heteroscedastisity (GARCH) modell segítségével elemzem. Ezen belül kvantitatív összefüggést létesítek a piac struktúrája és a legjobban illeszkedő GARCH modell között. Ezen összefüggés alapján a GARCH(1,1) modellt használom a piacstruktúrában bekövetkező változás detektálására, kihasználva a változás-detektálás probléma jól ismert tulajdonságát, mely szerint az végrehajtható rosszul specifikált modell felhasználásával is. Bemutatok egy GARCH alapú, valós idejű változás-detektáló algoritmust, amely a Minimális Leíróhossz módszer elveit követi. A GARCH alapú riasztást közgazdaságilag értelmezem. Az algoritmust sikeresen alkalmaztam a fenti tőzsdemodellben létrejövő piaci struktúra megváltozásának detektálására illetve futtattam valódi árfolyamadatokon is.

A GARCH modell paramétereinek kvázi Maximum Likelihood alapú becslésére egy új, általános nemlineáris optimalizáló algoritmust javasolok, amely vágósík alapú technikák és (kvázi-) Newton módszerek keveréke. Bemutatok egy technikát, amellyel a GARCH illesztési probléma dimenziója csökkenthető.

A pénzügyi piacokon fellépő hírfolyamatokat egy ún. Hawkes folyamattal modellezzük, melyekben a visszacsatolást az eseményfolyamat és az intenzitás között egy véges dimenziós lineáris rendszerrel írjuk le. Algoritmusokat javasolok e Hawkes folyamatok szimulációjára és rekurzív identifikációjára. Egydimenziós visszacsatolású Hawkes folyamatokra megmutatom, hogy a Fisher információs mátrix diagonálisának egy része a végtelenbe tart, másik része nullához, ha a paraméterekkel a stabilitási tartomány széléhez tartunk. Első lépésként kiszámolom a megfelelően átskálázott folyamat intenzitásának határeloszlását.

Végül bemutatok egy algoritmust, amellyel egy lineáris regresszió paramétereit becsülhetjük kvantált megfigyelésekből, ha a regresszor értékkészlete véges. Az algoritmus Expectation Maximization és Markov Chain Monte Carlo módszereken alapul.

Bibliography

Aknouche, A., Guerbyenne, H., 2006. Recursive estimation of GARCH models. Communications in statistics. Simulation and computation 35 (4), 925–938.

Alfarano, S., 2006. An agent-based stochastoc volatility model. Ph.D. thesis, Christian-Albrechts-Universität zu Kiel.

Alfarano, S., Lux, T., Wagner, F., 2005. Estimation of agent-based models: The case of an asymmetric herding model. Computational Economics 26, 19–49.

Amilon, H., 2008. Estimation of an adaptive stock market model with heterogeneous agents. Journal of Empirical Finance 15 (2), 342 – 362.

Armijo, L., 1966. Minimization of functions having Lipschitz continuous first partial derivatives. Pacific J. Math 16 (1), 1–3.

Auslender, A., Silva, P. J., Teboulle, M., 2007. Nonmonotone projected gradient methods based on barrier and euclidean distances. Comput. Optim. Appl. 38 (3), 305–327.

Basseville, M., Nikiforov, I., 1993. Detection of Abrupt Changes: Theory and Application. Prentice Hall.

Bauwens, L., Preminger, A., Rombouts, J. V., 2007. Theory and inference for a markov switching garch model. Cahiers de recherche 07-09, HEC Montréal, Institut d'économie appliquée.

Benveniste, A., Métivier, M., Priouret, P., 1990. Adaptive algorithms and stochastic approximations. Springer-Verlag, Berlin.

Berkes, I., Gombay, E., Horváth, L., Kokoszka, P., 2004. Sequential change-point detection in GARCH(p,q) models. Econometric Theory 20 (06), 1140–1167.

Berkes, I., Horváth, L., Kokoszka, P., 2003. GARCH processes: structure and estimation. Bernoulli 9, 201–217.

Berndt, E., Hall, B., Hall, R., Hausman, J., 1974. Estimation and inference in nonlinear structural models. Annals of Economic and Social Measurement 3, 653–665.

Bollerslev, T., 1986. Generalized autoregressive conditional heteroscedasticity. Journal of Econometrics 31 (3), 307–327.

Bollerslev, T., Wooldridge, J. M., 1992. Quasi-maximum likelihood estimation and inference in dynamic models with time-varying covariances. Econometric Reviews 11, 143–172.

Boswijk, P., Hommes, C., Manzan, S., 2007. Behavioral heterogeneity in stock prices. Journal of Economic Dynamics and Control 31, 1938–1970.

Boyd, S., 2008. Notes on subgradient methods. Unpublished note for EE364B available at www.stanford.edu/class/ee364b/.

Boyd, S., Vandenberghe, L., 2004. Convex Optimization. Cambridge University Press.

Bremaud, P., 1981. Point processes and queues. Martingale dynamics. Springer Series in Statistics. Springer, New York.

Brock, W. A., Hommes, C. H., 1998. Heterogeneous beliefs and routes to chaos in a simple asset pricing model. Journal of Economic Dynamics and Control 22, 1235–1274.

Brockett, R., Liberzon, D., 2000. Quantized feedback stabilization of linear systems. IEEE Transactions on Automatic Control 75 (7), 1279–1289.

Campillo, F., Kutoyants, Y., Gland, F. L., 2000. Small noise asymptotics of the glr test for off-line change detection in misspecified diffusion processes. Stochastics and Stochastics Reports 70 (1–2), 109–129.

Cesa-Bianchi, N., Lugosi, G., 2006. Prediction, Learning, and Games. Cambridge University Press, New

York, NY, USA.

Chen, S. H., Chang, C. L., Du, Y. R., 2009. Agent-based economic models and econometrics. Knowledge Engineering Review, forthcoming.

Chiarella, C., He, X.-Z., Hommes, C., 2006. A dynamic analysis of moving average rules. Journal of Economic Dynamics and Control 30 (9-10), 1729 – 1753.

Cont, R., 2001. Empirical properties of asset returns: stylized facts and statistical issues. Quantitative Finance 1, 223–236.

Crummey, T. P., Farshadnia, R., Fleming, P. J., Grace, A. C. W., Hancock, S. D., 1991. An optimization toolbox for MATLAB. IEE conference publication 2 (332), 744–749.

Daley, D. J., Vere-Jones, D., 2003a. An introduction to the theory of point processes. Vol. I: Elementary theory and methods. 2nd ed. Probability and Its Applications. Springer, New York.

Daley, D. J., Vere-Jones, D., 2003b. An introduction to the theory of point processes. Vol. I. Springer-Verlag.

De Grauwe, P., Grimaldi, M., 2004. Exchange rate puzzles: A tale of switching attractors. Working Paper Series 163, Sveriges Riksbank (Central Bank of Sweden).

Dempster, A. P., Laird, N. M., Rubin, D. B., 1977. Maximum likelihood from incomplete data via the EM algorithm(with discussion). J. Roy. Statist. Soc. Ser. 39, 1–38.

Diks, C., van der Weide, R., 2005. Herding, a-synchronous updating and heterogeneity in memory in a cbs. Journal of Economic Dynamics and Control 29, 741–763.

Engle, R. F., 1982. Autoregressive conditional heteroscedasticity with estimates of the variance of united kingdom inflation. Econometrica 50 (4), 987–1007.

Finesso, L., Gerencsér, L., Kmecs, I., 1999a. Estimation of parameters from quantized noisy observations. In: Proceedings of the European Control Conference ECC'99, AM-3, F589, Karlsuhe, Germany. 6p.

Finesso, L., Gerencsér, L., Kmecs, I., 1999b. A randomized EM-algorithm for estimating quantized linear gaussian regression. In: Proceedings of the 38th IEEE Conference on Decision and Control. CDC'99, Phoenix, Arizona, USA. pp. 5100–5101.

Finesso, L., Gerencsér, L., Kmecs, I., 2000. A recursive randomized EM-algorithm for estimation under quantization error. In: Proceedings of the American Control Conference, Chicago, Illinois. IEEE Control System Society, pp. 790–791.

Fletcher, R., 1980. Practical Methods of Optimization. Volume 1: Unconstrained Optimization, 1st Edition. John Wiley and Sons Ltd.

Gerencsér, L., 1996. On fixed gain recursive estimation processes. J. of Mathematical Systems, Estimation and Control 6, 355–358, retrieval code for full electronic manuscript: 56854.

Gerencsér, L., 2006. A representation theorem for the error of recursive estimators. SIAM J. Control and Optimization 44, 2123–2188.

Gerencsér, L., Baikovicius, J., 1991. Change-point detection using stochastic complexity. identification and system parameter estimation. Selected papers from the 9th IFAC-IFORS Symposium on Budapest 1, 73–78.

Gerencsér, L., Baikovicius, J., 1992. Change-point detection as model selection. Informatica 3, 3–20.

Gerencsér, L., Kmecs, I., Torma, B., 2008a. Quantization with adaptation, estimation of gaussian linear models. Communications in Information and Systems 8 (The Brockett Legacy Special Issue. In Honor of Roger W. Brockett in honor of his 75th birthday (guest eds.: John Baillieul, John S. Baras, Anthony

Bloch, P.S. Krishnaprasad and Jan C. Willems)), 223–244.

Gerencsér, L., Matias, C., Vágó, Z., Torma, B., Weiss, B., 2008b. Self-exciting point processes with applications in finance and medicine. In: 18th International symposium on Mathematical Theory of Networks and Systems.

Gerencsér, L., Mátyás, Z., 2007a. Almost sure and L_q-convergence of the re-initialized bmp scheme. In: Castanon, D., Spall, J. (Eds.), Proc. 46th IEEE Conference on Decision and Control, Invited session, New Orleans, LA, USA, 13-15 December 2007. Omnipress, pp. 965–974, wEB11.4.

Gerencsér, l., Mátyás, Z., 2007b. A behavioral stock market model. Mathematical Methods of Operations Research 67, 43–63.

Gerencsér, L., Orlovits, Z., Torma, B., 2010. Recursive estimation of GARCH processes. The 19th International Symposium on Mathematical Theory of Networks and Systems, (MTNS 2010), Budapest, Hungary, forthcoming.

Giesecke, K., Goldberg, L., Backshall, T., 2004. Credit Risk Modeling. In: Fabozzi, F. (Ed.), Handbook of Fixed Income Securities. Wiley.

Hammersley, J., Handscomb, D., 1967. Monte-Carlo methods. London: Methuen & Co.

Hawkes, A., 1971a. Spectra of some self-exciting and mutually exciting point processes. Biometrika 58, 83–90.

Hawkes, A. G., 1971b. Point spectra of some mutually exciting point processes. J. Roy. Statist. Soc. Ser. B 33, 438–443.

Hawkes, A. G., Oakes, D., 1974. A cluster process representation of a self-exciting process. J. appl. Probab. 11, 493–503.

Hinkley, D., 1971. Inference about a change-point from cummulative sum tests. Biometrika 58, 509–523.

Hommes, C. H., 2006. Heterogeneous agent models in economics and finance. In: Tesfatsion, L., Judd, K. (Eds.), Handbook of Computational Economics. Vol. 2. Elsevier, pp. 1109 – 1186.

Ispány, M., Pap, G., 2007. Weak convergence of step processes and an application for critical multitype branching processes with immigration. arXiv:math/0701803v1.

Ispány, M., Pap, G., van Zuijlen, M. C. A., 2005. Fluctuation limit of branching processes with immigration and estimation of the means. Advances in Applied Probability 37, 523–538.

Kalev, P. S., Liu, W.-M., Pham, P. K., Jarnecic, E., 2004. Public information arrival and volatility of intraday stock returns. Journal of Banking and Finance 28 (6), 1441–1467.

Kaufman, P. J., 1998. Trading systems and methods. Wiley Trading.

Kelley, C. T., 1999. Iterative methods for optimization. Frontiers in applied mathematics. SIAM, Philadelphia, PA, USA.

Kirman, A., 1995. The behaviour of the foreign exchange market. Bank of England Quarterly Bulletin 15, 286–293.

Kozhan, R., Salmon, M., 2009. Uncertainty aversion in a heterogeneous agent model of foreign exchange rate formation. Journal of Economic Dynamics and Control 33 (5), 1106 – 1122.

LeBaron, B., 2000. Agent-based comuptational finance: suggested readings and early research. Journal of Economic Dynamics and Control 24, 679–702.

Ljung, L., Söderström, T., 1983. Theory and practice of recursive identification. The MIT Press.

Luenberger, D. G., 1973. Introduction to linear and nonlinear programming. Addison-Wesley Publishing Company, Inc.

Lux, T., 1998. The socio-economic dynamics of speculative markets: interacting agents, chaos, and the fat tails of return distribution. Journal of Economic Behavior and Organization 33, 143–165.

Mangelsdorff, L., Weber, M., 1994. Testing choquet expected utility. Journal of Economic Behavior & Organization 25 (3), 437 – 457.

Masry, E., Cambanis, S., 1980. Signal Identification after Noisy Nonlinear Transformation. IEEE Transactions on Information Theory IT-26, 50–58.

Matteo, T. D., Aste, T., Dacorogna, M. M., 2004. Using the scaling analysis to characterize financial markets. Finance 0402014, EconWPA.

Mclachlan, G. J., Krishnan, T., 1996. The EM Algorithm and Extensions. Wiley-Interscience.

Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A., Teller, E., 1953. Equations of state calculations by fast computing machines. J. Chem. Phys. 21, 1087–1091.

Mitchell, M. L., Mulherin, J. H., 1994. The impact of public information on the stock market. The Journal of Finance 49, 923–950.

Moller, J., Rasmussen, J. G., 2005. Perfect simulation of Hawkes processes. Adv. Appl. Probab. 37 (3), 629–646.

Moller, J., Rasmussen, J. G., 2006. Approximate simulation of Hawkes processes. Methodol. Comput. Appl. Probab. 8 (1), 53–64.

Moré, J. J., Garbow, B. S., Hillstrom, K. E., 1981. Testing unconstrained optimization software. ACM Trans. Math. Softw. 7 (1), 17–41.

Murray, W., Overton, M. L., 1978. Steplength algorithms for minimizing a class of nondifferentiable functions. Tech. rep., Stanford University, Stanford, CA, USA.

Ogata, Y., 1978. The asymptotic behaviour of maximum likelihood estimators for stationary point processes. Ann. Inst. Stat. Math. 30, 243–261.

Ogata, Y., Akaike, H., 1982. On linear intensity models for mixed doubly stochastic Poisson and self-exciting point processes. J. R. Stat. Soc., Ser. B 44, 102–107.

Polyak, R. A., 2007. Regularized Newton method for unconstrained convex optimization. Math. Prog.Published online.

Prokaj, V., Torma, B., 2010. Identification of almost unstable Hawkes processes. Publicationes Mathematicae Debrecen, forthcoming.

Rissanen, J., 1989. Stochastic complexity in statistical inquiry. World Scientific Publisher.

Runggaldier, W., 2003. Jump Diffusion Models. In: Rachev, S. (Ed.), Handbook of Heavy Tailed Distributions in Finance. Elsevier/North-Holland, pp. 169–209, handbooks in Finance, Book 1 (W.Ziemba Series Ed.).

Shi, Z.-J., 2004. Convergence of line search methods for unconstrained optimization. Appl. Math. Comput. 157 (2), 393–405.

Soderstrom, T., Stoica, P., 1989. System Identification. Prentice Hall International.

Solo, V., 1999. Adaptive algorithms and Markov chain Monte Carlo methods. In: Proceedings of the 38th IEEE Conference on Decision and Control, Phoenix, Arizona. IEEE Control System Society, pp. 1775–1778.

Sonnevend, G., 1988. New algorithms in convex programming based on a notion of "centre" (for systems of analytic inequalities) and on rational extrapolation. In: Hoffman, K., Hiriat-Urruty, J., Lemarechal, C., Zowe, J. (Eds.), Trends in Mathematical Optimization: Proceedings of the 4th French-German

Conference in Optimization in Irsee. Vol. 84. Birkhauser Verlag, pp. 311–327.

Spall, J. C., 2003. Introduction to Stochastic Search and Optimization. John Wiley & Sons, Inc., New York, NY, USA.

Taylor, S. J., 2007. Asset Price Dynamics, Volatility, and Prediction. Princeton University Press.

Torma, B., G.-Tóth, B., 2010. An efficient descent direction method with cutting planes. Central European Journal of Operations Research, forthcoming.

Wakker, P. P., 2001. Testing and characterizing properties of nonadditive measures through violations of the sure-thing principle. Econometrica 69 (4), 1039–1059.

Welch, I., 2000. Herding among security analysts. Journal of Financial Economics 58, 369–396.

Widrow, B., Kollár, I., 2008. Quantization Noise in Digital Computation, Signal Processing, and Control. Cambridge University Press.

Winker, P., Gilli, M., Jeleskovic, V., 2007. An objective function for simulation based inference on exchange rate data. Journal of Economic Interaction and Coordination 2, 125–145.

Ye, Y., 1997. Complexity analysis of the analytic center cutting plane method that uses multiple cuts. Math. Program. 78 (1), 85–104.

Zakoian, J.-M., 1994. Threshold heteroskedastic models. Journal of Economic Dynamics and Control 18 (5), 931–955.

Zhang, H., Hager, W. W., 2004. A nonmonotone line search technique and its application to unconstrained optimization. SIAM J. on Optimization 14 (4), 1043–1056.