UNIVERSITY OF CYPRUS

DEPARTMENT OF MATHEMATICS AND STATISTICS

STATISTICAL THEORY FOR MIXED POISSON TIME SERIES MODELS

Vasiliki Christou

SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY
AT
UNIVERSITY OF CYPRUS
NICOSIA, CYPRUS
NOVEMBER 2013

To my family.

For their endless love, support and encouragement.

Abstract

The goal of this thesis is the statistical analysis of mixed Poisson count time series models. The necessity for such an investigation arises from the fact that count dependent sequences, which appear in several scientific fields like medical, environmental or financial applications, they often face the phenomenon of overdispersion, that is the sample variance of the data collection is larger than the corresponding mean.

In particular, we study inference for count time series regression models which include a feedback mechanism. We study probabilistic properties and quasi-likelihood estimation for this class of processes. We show that the resulting estimators are consistent and asymptotically normally distributed. The key observation in developing the theory is a mean parameterized form of the mixed Poisson process.

In addition, we employ different criteria to study the prediction problem. In particular, we provide probabilistic forecasts based on the assumption of Poisson or negative binomial distribution, which they fall within the framework of mixed Poisson processes and we propose the use of Probability Integral Transformation (PIT) histogram, marginal calibration plot and scoring rules to assess the predictive performance and rank the competing forecast models.

In the last part of this thesis we discuss the testing problem for linearity in the context of such models. We employ the score test, suitably adjusted, as it is based on a quasi-likelihood function. Our methodology covers both cases where parameters can be identifiable or non identifiable under the null.

Περίληψη

Σκοπός αυτής της εργασίας είναι η στατιστική ανάλυση μικτών Poisson μοντέλων χρονοσειρών, τα οποία λαμβάνουν ακέραιες τιμές. Η αναγκαιότητα μιας τέτοιας μελέτης προκύπτει από το γεγονός ότι οι ακέραιες εξαρτημένες ακολουθίες, οι οποίες εμφανίζονται σε πολλά επιστημονικά πεδία όπως η ιατρική, η μελέτη του περιβάλλοντος ή οι οικονομικές εφαρμογές, πολύ συχνά αντιμετωπίζουν το φαινόμενο της υπερδιακύμανσης, δηλαδή η δειγματική διασπορά της συλλογής δεδομένων είναι μεγαλύτερη από την αντίστοιχη μέση τιμή.

Ειδικότερα, μελετούμε συμπερασματολογία για μοντέλα παλινδρόμησης ακέραιων χρονοσειρών, τα οποία περιλαμβάνουν ένα μηχανισμό ανάδρασης. Μελετούμε πιθανοθεωρητικές ιδιότητες και εκτίμηση μέσω της ημιπιθανοφάνειας για αυτή την κλάση διαδικασιών. Αποδεικνύουμε ότι οι προκύπτουσες εκτιμήτριες είναι συνεπείς και ασυμπτωτικά κανονικά κατανεμημένες. Η βασική παρατήρηση στην ανάπτυξη της θεωρίας είναι η κατάλληλη παραμετροποιημένη μορφή της μέσης τιμής της μικτής Poisson διαδικασίας.

Επιπλέον, χρησιμοποιούμε διάφορα κριτήρια για να μελετήσουμε το πρόβλημα της πρόβλεψης. Ειδικότερα, παρέχουμε πιθανοθεωρητικές προβλέψεις βασιζόμενες στην υπόθεση της Poisson ή της αρνητικής διωνυμικής κατανομής, οι οποίες εμπίπτουν στο πλαίσιο των μικτών Poisson διαδικασιών και προτείνουμε τη χρήση του ιστογράμματος μετασχηματισμού πιθανότητας, της καμπύλης περιθωριακής βαθμονόμησης και των κανόνων scoring για να αξιολογήσουμε την απόδοση πρόβλεψης και να κατατάξουμε τα ανταγωνιστικά μοντέλα πρόβλεψης.

Στο τελευταίο μέρος αυτής της εργασίας συζητούμε το πρόβλημα ελέγχου γραμμικότητας στο πλαίσιο αυτών των μοντέλων. Χρησιμοποιούμε την ελεγχοσυνάρτηση score, κατάλληλα προσαρμοσμένη, αφού βασίζεται στην ημιπιθανοφάνεια. Η μεθοδολογία μας καλύπτει και τις δύο περιπτώσεις όπου οι παράμετροι είναι προσδιορίσιμες ή μη, κάτω από τη μηδενική υπόθεση.

Acknowledgements

I take this opportunity to express my deepest appreciation and my spacial gratitude to my supervisor, Prof. Konstantinos Fokianos, who has been instrumental in the successful completion of this work. He invested his full effort in guiding me in achieving my goal. His patience, motivation, enthusiasm, stimulating suggestions, unconditional support and consistent encouragement, helped me to coordinate my project. Without his valuable guidance and persistent help, this dissertation would not have been possible.

In addition, I would like to thank all the other members of the examining committee, Prof. C. Francq, Prof. D. Karlis, Prof. E. Paparoditis and Prof. T. Sapatinas, for several comments and suggestions that improve this work.

I would also like to thank the Research Promotion Foundation of Cyprus for the financial support granted through my doctoral studies (TEXNOLOGIA/THEPIS/0609(BE)/02).

Special thanks to all the academic and administrative staff of the Department of Mathematics and Statistics of the University of Cyprus, for their support and help all these ten years of my undergraduate and postgraduate studies.

Furthermore, I would like to acknowledge the use of resources provided by the European Grid Infrastructure. In particular, I acknowledge computing support by Prof. M. Dikaiakos and Ms. A. Balla (CyGrid).

Last but not least, I wish to express my love and gratitude to my family for their understanding and endless love, through the duration of my studies. I am grateful for their constant support and help.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

During recent years there has been considerable interest in models for time series of counts. The necessity for such an investigation arises from the fact that count dependent sequences appear in several diverse scientific fields, including medical, environmental or financial applications. The main goal of this thesis is the development of statistical methodology for count time series modeling, estimation, inference, prediction and testing.

The study and the analysis of count time series poses several problems and questions. For instance, a common distribution that is used in practice to model the response time series, is the Poisson distribution. Such an assumption is sensible because the Poisson distribution is the simplest discrete distribution, yet its properties are satisfactory to cover a large class of problems. However, to the best of our knowledge, the appropriateness of the Poisson assumption has not been discussed properly in the literature. One of our contributions, is to examine whether the Poisson assumption can be verified by real data. Towards this goal, Figure 1.1 shows the Probability Integral Transformation (PIT) histograms, see Czado et al. [13], for transactions data of a certain stock. The PIT will be discussed in detail in Chapter 4, but the main message of Figure 1.1 is that the Poisson assumption fails to model these data and a better approach is provided by the negative binomial distribution *suitably* parameterized. In fact, Figure 1.1 motivates our contribution, in the sense that the PIT plot based on negative binomial distribution shows better fit of the data than the corresponding PIT based on Poisson. To understand why is this so, note that the negative binomial process belongs to the class of mixed Poisson processes; see Mikosch [64]. These processes can be used for modeling transactions data

(and more generally count time series) where the total number of transactions can be thought as a collection of individual transactions which correspond to different trading strategies.

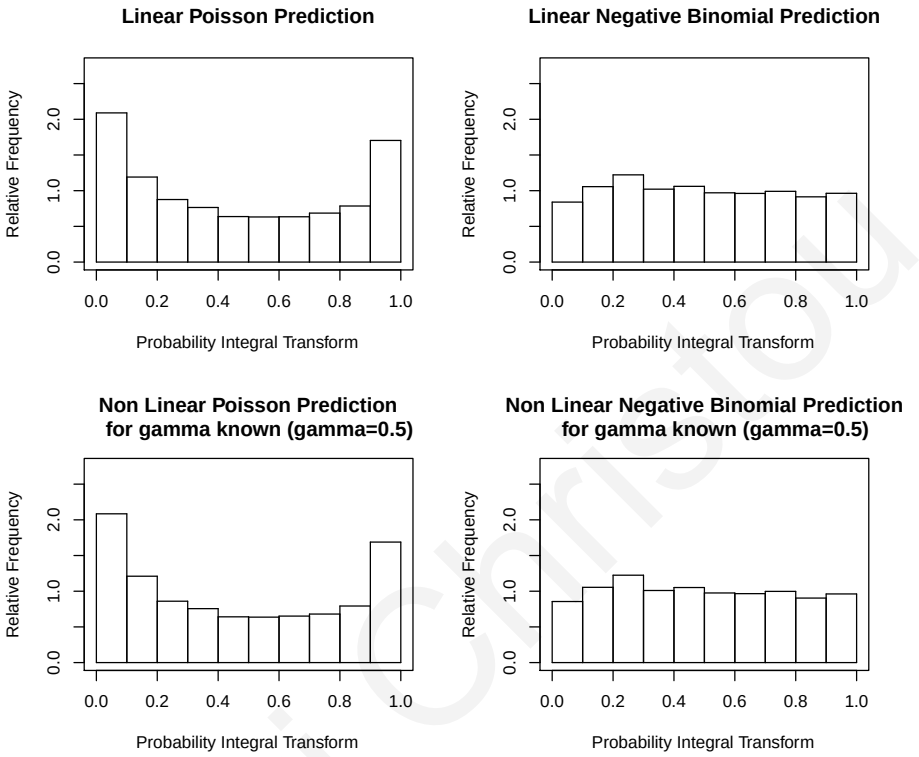

Figure 1.1: PIT histograms applied to the number of transactions per minute for the stock Ericsson B for the time period between July 2nd and July 22nd, 2002. From top to bottom: PIT histograms for model (3.6) and model (3.11) for $\gamma = 0.5$. Left Plots: The conditional distribution is the Poisson. Right Plots: The conditional distribution is the negative binomial.

More specifically, we will be studying models that include a feedback mechanism, similar to Fokianos et al. [34]. It has been proved that this class of models successfully accommodates count time series whose autocorrelation function decays slowly towards zero. The inclusion of the feedback mechanism makes parsimonious modeling possible. This is in line with GARCH modeling of financial returns, whereby past values of volatility are fed back into the system; see Bollerslev [5]. The context of this thesis is quite different from the previous contribution, in many aspects. First, we extend Fokianos et al. [34], to mixed Poisson count time series by suitably parameterizing its mean. Using the observation that a mixed Poisson process has stationary increments, we can discover ergodicity conditions that do not depend upon any additional parameters regarding the variance of the mix-

ing variable. The main instrument on deriving such conditions, is the notion of weak dependence (Dedecker et al. [21]) in connection to the recent contribution by Doukhan and Wintenberger [27]. These works give verifiable conditions for studying ergodicity of a general class of processes. Based on these theoretical results and inspired by the success of Gaussian likelihood in modeling and estimation of GARCH models, *we propose estimation of the regression parameters based on quasi-likelihood function obtained from the Poisson distribution*. It turns out, that the correct mean specification yields consistent estimates whose asymptotic covariance matrix is given by a sandwich type matrix; a basic result in GARCH literature for the Gaussian quasi-likelihood function. The variance of the mixing variable which appears in the expression of the sandwich type matrix is estimated consistently by two standard estimation methods. In this approach, we avoid complicated likelihood functions and at the same time, it is still possible to obtain consistent estimators whose standard error can be robustly estimated; White [73]. Moreover, the results of estimation enable construction of PIT plots based on the negative binomial distribution, which falls within the framework of mixed Poisson processes, instead of Poisson. The PIT plots provides a graphical diagnostic tool to assess any distributional assumptions.

Models that are similar to the ones we consider in this contribution, but for Poisson data, have been studied recently by Neumann [66], Fokianos and Tjøstheim [37] and Doukhan et al. [23]. Related literature regarding log-linear models for time series of counts include the works by Zeger and Qaqish [76], Brumback et al. [9], Fahrmeir and Tutz [29], Kedem and Fokianos [56], Davis et al. [16], Jung et al. [55] and more recently Fokianos and Tjøstheim [36], Woodard et al. [74] and Douc et al. [22]. Negative binomial distributed models have been studied by Davis and Wu [15], Zhu [77] and Davis and Liu [17]. The mean parametrization that is used in these articles is quite different from our approach. In particular, we study observation driven models, as opposed to Davis and Wu [15] who study parameter driven models (for a classification between observation and parameter driven models, see Cox [12]). The articles by Zhu [77] and Davis and Liu [17] study observation driven models for count time series in the context of negative binomial distribution. However, their parametrization is quite different from the one we use, as we will discuss below. Questions regarding ergodicity and stationarity of mixed Poisson processes, as well as asymptotic maximum likelihood theory have not been addressed in

the literature, except in Davis and Liu [17] for the case of negative binomial but with a different parametrization.

In the second part of this contribution we study the prediction problem in the context of count time series, one of the most crucial aspects in time series analysis. More specifically, we follow the recent methodology of Czado et al. [13], where various tools for predictive model assessment are developed for independent but not identically distributed data. We show that these methods can also be applied for count dependent data. We will take a similar point of view; that is predictors are probabilistic in the sense that a probability distribution can adequately describe their basic properties; see Dawid [19].

To predict future values of the response, the most natural way (in terms of mean square error) is to employ the conditional mean after estimating any unknown parameters by using maximum likelihood. However, the performance of such forecast is largely unknown in the literature; for an exception see Jung and Tremayne [54]. This contribution fills this gap by studying the behavior of such predictor using a variety of scoring rules. In addition, we study the same problem for negative binomial time series. At the end, we can in fact assess which distribution is more suitable for the data at hand.

We focus on Poisson and negative binomial distributions since these are occurred in applications more frequently; however the methods can be applied to other discrete distributions, like mixtures of Poisson, provided that they are suitably parameterized. We also show that models which include a feedback mechanism considerably reduce the number of parameters for fitting and predicting models with strong autocorrelation, regardless of the chosen response distribution.

In the last part of this thesis we are particularly interested in testing linearity against two special classes of nonlinear alternatives for count time series data. Evaluating the significance of added variables in a regression equation can be carried out using the likelihood ratio test, Wald test or score (Lagrange Multiplier) test under quasi-likelihood theory. The score test is often a very convenient tool because it does not require estimation of the nonlinear model and only requires the estimation of the constrained model under the null hypothesis. However, careful application of the methodology requires suitable adjustment of the score test, since it is based on quasi-likelihood methodology. Note that, all aforementioned types of test statistics are asymptotically equivalent (cf. Francq and Zakoïan

[40, Ch. 8]).

Two major classes of nonlinear models are considered, which both of them nest the linear model. The first class consists of models which do not face the problem of non identifiability, that is all the parameters of the model are identified under the null hypothesis. For this class of models and under the null hypothesis of linearity, the score test statistic possesses an asymptotic $\mathcal{X}^2$ distribution. The second class of nonlinear models consists of models in which a nonnegative nuisance parameter exists under the alternative hypothesis but not when the null holds. In this particular case, the testing problem is nonstandard and the classical asymptotic theory for the score test does not apply.

Chapter 2

Basic Tools and Definitions

2.1 Introduction

In this chapter we define important tools and definitions for the sequel.

Definition 2.1.1 *(Probability mass function of the Poisson distribution)*
Suppose that Y is a Poisson distributed random variable with mean equal to λ, $\lambda > 0$. Then the probability mass function (pmf) of Y, for $y = 0, 1, 2, \ldots$, is given by

$$f_Y(y) = \frac{e^{-\lambda}\lambda^y}{y!}.$$

Definition 2.1.2 *(Probability density function of the gamma distribution)*
Suppose that Z is a gamma distributed random variable with parameters $\lambda > 0$ and $\nu > 0$. Then the probability density function of Z, for $z > 0$, is given by

$$f_Z(z) = \frac{1}{\Gamma(\nu)}\left(\frac{\nu}{\lambda}\right)^{\nu} z^{\nu-1}e^{-(\frac{\nu}{\lambda}z)}.$$

Definition 2.1.3 *(Probability mass function of the negative binomial distribution)*
Suppose that Y is a negative binomial distributed random variable with parameters $\nu > 0$ and $p \in (0, 1)$. Then the pmf of Y, for $y = 0, 1, 2, \ldots$, is given by

$$f_Y(y) = \frac{\Gamma(\nu + y)}{\Gamma(y + 1)\Gamma(\nu)}p^{\nu}(1 - p)^y. \tag{2.1}$$

The mean and the variance of Y are given by $\mathrm{E}(Y) = \nu(1-p)/p$ and $\mathrm{Var}(Y) = \nu(1-p)/p^2$,

respectively.

The negative binomial distribution arises as a mixture of a Poisson distribution where the mixing distribution of the Poisson rate is a gamma distribution. That is, we can view the negative binomial as a conditional Poisson(z) distribution, given $Z = z$, where Z is itself a random variable, distributed according to gamma with parameters λ and ν. To calculate the pmf of Y for $y = 0, 1, 2, \ldots$, we follow the steps below:

$$\begin{aligned}
f_Y(y) &= \int_0^\infty f_{Y|Z}(y|z).f_Z(z)dz \\
&= \int_0^\infty \frac{e^{-z}z^y}{y!}\frac{1}{\Gamma(\nu)}\big(\frac{\nu}{\lambda}\big)^\nu z^{\nu-1}e^{-\frac{\nu}{\lambda}z}dz \\
&= \frac{1}{\Gamma(\nu)y!}\Big(\frac{\nu}{\lambda}\Big)^\nu \int_0^\infty z^{\nu+y-1}e^{-\left(\frac{\nu}{\lambda}+1\right)z}dz \\
&= \frac{1}{\Gamma(\nu)y!}\Big(\frac{\nu}{\lambda}\Big)^\nu \frac{\Gamma(\nu+y)}{\big(\nu/\lambda+1\big)^{\nu+y}} \\
&= \frac{\Gamma(\nu+y)}{\Gamma(\nu)\Gamma(y+1)}\Big(\frac{\nu/\lambda}{\nu/\lambda+1}\Big)^\nu\Big(\frac{1}{\nu/\lambda+1}\Big)^y \\
&= \frac{\Gamma(\nu+y)}{\Gamma(\nu)\Gamma(y+1)}\Big(\frac{\nu}{\nu+\lambda}\Big)^\nu\Big(\frac{\lambda}{\nu+\lambda}\Big)^y
\end{aligned}$$

Comparing the previous display with the pmf of the negative binomial distribution given by (2.1), we realize that Y has the negative binomial distribution with $p = \nu/(\nu+\lambda)$. Hence, the mean and the variance of Y are given by $\mathrm{E}(Y) = \lambda$ and $\mathrm{Var}(Y) = \lambda + \lambda^2/\nu$.

Definition 2.1.4 *(Poisson process),* Brockwell and Davis [8, Def. 1.7.3]
A Poisson process with mean rate $\lambda > 0$ is a process $\{N(t), t \geq 0\}$ satisfying the conditions

- $N(0) = 0$,
- $N(t_2) - N(t_1),\ N(t_3) - N(t_2),\ \ldots, N(t_n) - N(t_{n-1})$, are independent for every $n \in \{3, 4, \ldots\}$ and every $\boldsymbol{t} = (t_1, \ldots, t_n)'$ such that $0 \leq t_1 < t_2 < \ldots < t_n$,
- $N(t) - N(s)$ has the Poisson distribution with mean $\lambda(t - s)$ for $t \geq s$.

Definition 2.1.5 *(Autocovariance function),* Brockwell and Davis [8, Def. 1.3.1]
Let $\{Y_t, t \in T\}$ be a process such that $\mathrm{Var}(Y_t) < \infty$ for each $t \in T$. Then the autocovariance function $\gamma_Y(\cdot, \cdot)$ of $\{Y_t\}$ is defined by

$$\gamma_Y(r, s) = \mathrm{Cov}(Y_r, Y_s) = \mathrm{E}[(Y_r - \mathrm{E}(Y_r))(Y_s - \mathrm{E}(Y_s))], \qquad r, s \in T.$$

Definition 2.1.6 *(White noise process),* Brockwell and Davis [8, Def. 3.1.1]
The process $\{\epsilon_t\}$ is said to be white noise with mean 0 and variance σ^2, written

$$\{\epsilon_t\} \sim \mathrm{WN}(0, \sigma^2),$$

if and only if $\{\epsilon_t\}$ has zero mean and covariance function

$$\gamma(h) = \begin{cases} \sigma^2 & \text{if } h = 0, \\ 0 & \text{if } h \neq 0. \end{cases}$$

Definition 2.1.7 *(Stationarity),* Brockwell and Davis [8, Def. 1.3.2]
The time series $\{Y_t, t \in \mathbb{Z}\}$, with index set $\mathbb{Z} = \{0, \pm 1, \pm 2, \ldots\}$, is said to be stationary if

- $\mathrm{E}|Y_t|^2 < \infty, \qquad \forall\ t \in \mathbb{Z},$
- $\mathrm{E}(Y_t) = \mu, \qquad \forall\ t \in \mathbb{Z},$
- $\gamma_Y(r, s) = \gamma_Y(r + t, s + t), \qquad \forall\ r, s, t \in \mathbb{Z}.$

Definition 2.1.8 *(Strict Stationarity),* Brockwell and Davis [8, Def. 1.3.3]
The time series $\{Y_t,\ t \in \mathbb{Z}\}$, is said to be strictly stationary if the joint distributions of vectors $(Y_{t_1}, \ldots, Y_{t_k})'$ and $(Y_{t_1+h}, \ldots, Y_{t_k+h})'$ are the same for all positive integers k and for all $t_1, \ldots, t_k, h \in \mathbb{Z}$.

Definition 2.1.9 *(Ergodic stationary process),* Francq and Zakoïan [40, Def. A.1]
A strictly stationary process $\{Y_t,\ t \in \mathbb{Z}\}$, is said to be ergodic if and only if, for any Borel set B and any integer k,

$$\frac{1}{n} \sum_{t=1}^{n} 1_B(Y_t, Y_{t+1}, \ldots, Y_{t+k}) \to P((Y_1, \ldots, Y_{1+k}) \in B)$$

with probability 1.

Definition 2.1.10 *(Contraction)*

A contraction mapping or contraction on a metric space (M, d) is a function $f : M \to M$, with the property that there is some nonnegative real number $k < 1$ such for all x and y in M,

$$d(f(x), f(y)) \leq kd(x, y).$$

Definition 2.1.11 *(ARMA(p,q) process),* Brockwell and Davis [8, Def. 3.1.2]

The process $\{Y_t,\ t \in \mathbb{Z}\}$ is said to be an AutoRegressive Moving Average ARMA(p, q) process if $\{Y_t\}$ is stationary and if for every t,

$$Y_t - \sum_{r=1}^{p} \phi_r Y_{t-r} = \epsilon_t + \sum_{r=1}^{q} \theta_r \epsilon_{t-r}, \tag{2.2}$$

where $\{\epsilon_t\} \sim WN(0, \sigma^2)$. We say that $\{Y_t\}$ is an ARMA(p, q) process with mean μ if $\{Y_t - \mu\}$ is an ARMA(p, q) process.

Example 2.1.1 *(Autocovariance function of ARMA(1,1) process),*

Consider Definition 2.1.11 for $p = q = 1$. The autocovariance function of an ARMA(1,1) process is given by

$$\gamma(|h|) = \mathrm{Cov}\left[Y_t, Y_{t+|h|}\right] = \begin{cases} \dfrac{1 + 2\theta_1\phi_1 + \theta_1^2}{1 - \phi_1^2}\ \sigma^2, & h = 0, \\ \dfrac{(1 + \theta_1\phi_1)(\theta_1 + \phi_1)}{1 - \phi_1^2}\phi_1^{|h|-1}\ \sigma^2, & |h| \geq 1. \end{cases}$$

Definition 2.1.12 *(GARCH(p,q) process),* Francq and Zakoïan [40, Def. 2.1]

A process $\{\epsilon_t\}$ is said to be a Generalized AutoRegressive Conditional Heteroscedastic process - GARCH(p, q) if its first two conditional moments exist and satisfy:

- $\mathrm{E}(\epsilon_t | \epsilon_u, u < t) = 0, \quad t \in \mathbb{Z}$,
- There exist constants $\omega, \alpha_i, i = 1, \ldots, q$ and $\beta_j, j = 1, \ldots, p$ such that

$$\sigma_t^2 = \mathrm{Var}(\epsilon_t | \epsilon_u, u < t) = \omega + \sum_{i=1}^{q} \alpha_i \epsilon_{t-i}^2 + \sum_{j=1}^{p} \beta_j \sigma_{t-j}^2, \quad t \in \mathbb{Z}.$$

Definition 2.1.13 *(Mixed Poisson process),* Mikosch [64, Def. 2.3.1]

Let $\tilde{N}$ be a standard homogeneous Poisson process, that is a Poisson process with rate equal to 1. Let $\mu(u)$ be the mean value function of a Poisson process on $[0, \infty)$. Suppose further that Z is a positive random variable independent of $\tilde{N}$. Then, the process

$$N(u) = \tilde{N}(Z\mu(u)) = \tilde{N}(0, Z\mu(u)], \quad u \geq 0,$$

is called a mixed Poisson process. The variable Z is called the mixing variable.

Theorem 2.1.1 *(Mean value theorem)*

Let $f : U \subseteq \mathbb{R}^n \to \mathbb{R}$ be differentiable, and $\boldsymbol{a},\ \boldsymbol{b} \in U$, such that $[\boldsymbol{a}, \boldsymbol{b}] \subseteq U$. Then $\exists\ \boldsymbol{c} \in (\boldsymbol{a}, \boldsymbol{b})$ such that

$$f(\boldsymbol{b}) - f(\boldsymbol{a}) = \nabla f(\boldsymbol{c}) \cdot (\boldsymbol{b} - \boldsymbol{a}).$$

It holds that

$$|f(\boldsymbol{b}) - f(\boldsymbol{a})| \leq \sup_{\boldsymbol{x} \in (\boldsymbol{a}, \boldsymbol{b})} |\nabla f(\boldsymbol{x})| \cdot |\boldsymbol{b} - \boldsymbol{a}|.$$

Lemma 2.1.1 *(Cesaro′s lemma)*

Let b_n be a monotonic increasing sequence of positive real numbers such that $b_n \to \infty$. Let x_n be a sequence such that $x_n \to x$. Then

$$\frac{1}{b_n} \sum_{t=1}^{n} (b_t - b_{t-1}) x_t \to x \quad \text{as} \quad n \to \infty.$$

Definition 2.1.14 *(Probability Integral Transformation)*

Suppose that X is a nonnegative integer-valued random variable, that is $X = 1, 2, \ldots$ Let $\gamma_i = P(X = i)$ and $p_i = P(X \leq i) = \gamma_1 + \gamma_2 + \ldots + \gamma_i$ for $i = 1, 2, \ldots$ It holds that $p_i - p_{i-1} = P(X \leq i) - P(X \leq i - 1) = P(X = i) = \gamma_i$. Denote by $U(a, b)$ the uniform distribution in (a, b).

Let Y be a random variable as follows:

$$Y = \begin{cases} U(0, p_1) & \text{when } X = 1, \\ U(p_1, p_2) & \text{when } X = 2, \\ U(p_2, p_3) & \text{when } X = 3, \\ \quad \vdots \end{cases} \tag{2.3}$$

We will show that Y has the standard uniform distribution. Consider the cumulative distribution function of Y,

$$\begin{aligned} P(Y \leq y) &= \sum_{i=1}^{\infty} P(U(p_{i-1}, p_i) \leq y | X = i) P(X = i) \\ &= \sum_{i=1}^{\infty} \gamma_i P(U(p_{i-1}, p_i) \leq y | X = i), \end{aligned} \tag{2.4}$$

where $P(U(p_{i-1}, p_i) \leq y | X = i)$ is equal to

$$P(U(p_{i-1}, p_i) \leq y | X = i) = \begin{cases} 0 & \text{if } y < p_{i-1}, \\ (y - p_{i-1})/(p_i - p_{i-1}) & \text{if } p_{i-1} \leq y < p_i, \\ 1 & \text{if } p_i \leq y < 1. \end{cases} \tag{2.5}$$

Therefore, from (2.4) and (2.5) we have that

$$P(Y \leq y) = \begin{cases} 0 & \text{if } y < 0, \\ y & \text{if } 0 \leq y < 1, \\ 1 & \text{if } y \geq 1, \end{cases}$$

and thus, Y is a standard uniform random variable.

Chapter 3

Inference for Count Time Series Models

3.1 Introduction

This chapter introduces the general framework for count time series analysis. We consider mixed Poisson autoregressions for the statistical analysis of this class of time series. This approach enlarges considerably the framework of the most common used distributional assumption, the Poisson-based analysis of count time series, as discussed by Kedem and Fokianos [56], Davis et al. [16] and Fokianos et al. [34] among others.

We study inference for count time series regression models which include a feedback mechanism. We study probabilistic properties and quasi-likelihood estimation for these models and we show that the resulting estimators are consistent and asymptotically normally distributed. In addition, we discuss some ergodicity and stationarity conditions for such models by employing the notion of weak dependence (see Dedecker et al. [21]). The key observation in developing the theory is a mean parameterized form of the models that fall under the framework of mixed Poisson processes.

3.2 Autoregressive Modeling

In what follows, we denote by $\|h\|_\infty = \sup_{\boldsymbol{x}\in\mathbb{R}^d} |h(\boldsymbol{x})|$ for the Euclidean space $\mathbb{R}^d$ and $h : \mathbb{R}^d \to \mathbb{R}$.

Assume that $\{Y_t, t \in \mathbb{Z}\}$ is a count time series and let $\{\lambda_t, t \in \mathbb{Z}\}$ be a sequence of mean processes. Denote by $\mathcal{F}_t^{Y,\lambda}$ the past of the process up to and including time t, that is $\mathcal{F}_t^{Y,\lambda} = \sigma(Y_s, s \leq t, \lambda_{1-q}, \ldots, \lambda_0)$, for some $q > 0$. In this chapter, we will study general theory for estimation for the following class of count time series models defined by

$$Y_t = \tilde{N}_t(0, Z_t\lambda_t] = N_t(0, \lambda_t], \quad t \in \mathbb{Z}. \tag{3.1}$$

In the above, $\tilde{N}_t$ is a standard homogeneous Poisson process (that is a Poisson process with rate equal to 1) and $\{Z_t\}$ denotes a sequence of independent and identically distributed (iid) positive random variables with mean 1, which are independent of $\tilde{N}_t$. In addition, we assume that λ_t is measurable with respect to $\{Y_s, s < t\}$ and Z_t is independent of $\{Y_s, s < t\}$. The family of processes belonging to (3.1) is called mixed Poisson process; see Definition 2.1.13, and it counts the number of events on the time interval $(0, \lambda_t]$.

We will now calculate the mean and the variance of the above process.

- Mean of the mixed Poisson process:

$$\begin{aligned}
\mathrm{E}(Y_t|\mathcal{F}_{t-1}^{Y,\lambda}) &= \mathrm{E}\{\mathrm{E}(Y_t|Z_t, \mathcal{F}_{t-1}^{Y,\lambda})\} = \mathrm{E}\{\mathrm{E}(\tilde{N}_t(0, Z_t\lambda_t]|Z_t, \mathcal{F}_{t-1}^{Y,\lambda})\} \\
&= \mathrm{E}(Z_t\lambda_t|\mathcal{F}_{t-1}^{Y,\lambda}), \quad \text{since } \tilde{N}_t \text{ is a Poisson process with rate 1} \\
&= \lambda_t \mathrm{E}(Z_t) \\
&= \lambda_t, \quad \text{since } \mathrm{E}(Z_t) \text{ is equal to 1.}
\end{aligned}$$

- Variance of the mixed Poisson process:

$$\begin{aligned}
\mathrm{Var}(Y_t|\mathcal{F}_{t-1}^{Y,\lambda}) &= \mathrm{E}\{\mathrm{Var}(Y_t|Z_t, \mathcal{F}_{t-1}^{Y,\lambda})\} + \mathrm{Var}\{\mathrm{E}(Y_t|Z_t, \mathcal{F}_{t-1}^{Y,\lambda})\} \\
&= \mathrm{E}\{\mathrm{Var}(\tilde{N}_t(0, Z_t\lambda_t]|Z_t, \mathcal{F}_{t-1}^{Y,\lambda})\} + \mathrm{Var}\{\mathrm{E}(\tilde{N}_t(0, Z_t\lambda_t]|Z_t, \mathcal{F}_{t-1}^{Y,\lambda})\} \\
&= \mathrm{E}(Z_t\lambda_t|\mathcal{F}_{t-1}^{Y,\lambda}) + \mathrm{Var}(Z_t\lambda_t|\mathcal{F}_{t-1}^{Y,\lambda}) \\
&= \lambda_t \mathrm{E}(Z_t) + \mathrm{Var}(Z_t)\lambda_t^2 \\
&= \lambda_t + \sigma_Z^2\lambda_t^2.
\end{aligned}$$

Note that if $\sigma_Z^2 > 0$ then the variance of the mixed Poisson process is greater than

the corresponding mean, so it takes into account overdispersion, a common phenomenon occurred repeatedly in the analysis of such data; see Dean et al. [20] and Lawless [60] for instance.

Two important distributions fall within the framework of (3.1) and they are routinely employed for the analysis of count time series. Namely, the Poisson distribution given by

$$\mathrm{P}[Y_t = y \mid \mathcal{F}_{t-1}^{Y,\lambda}] = \frac{\exp(-\lambda_t)\lambda_t^y}{y!}, \quad y = 0, 1, 2, \ldots \tag{3.2}$$

and the negative binomial distribution given by

$$\mathrm{P}[Y_t = y \mid \mathcal{F}_{t-1}^{Y,\lambda}] = \frac{\Gamma(\nu + y)}{\Gamma(y+1)\Gamma(\nu)} \Big(\frac{\nu}{\nu + \lambda_t}\Big)^{\nu} \Big(\frac{\lambda_t}{\nu + \lambda_t}\Big)^{y}, \quad y = 0, 1, 2, \ldots, \tag{3.3}$$

where $\nu > 0$. It is obvious that (3.2) is a special case of (3.1) when $\{Z_t\}$ is a sequence of degenerate random variables with mean 1. Furthermore (3.3) is a special case of (3.1) when $\{Z_t\}$ are iid gamma with mean 1 and variance $1/\nu$.

It can be shown that (3.3) tends to (3.2), when $\nu \to \infty$. Indeed, using Sterling's formula we have that

$$\begin{aligned}
(3.3) &= \frac{\Gamma(\nu + y)}{\Gamma(y+1)\Gamma(\nu)} \Big(\frac{\nu}{\nu + \lambda_t}\Big)^{\nu} \Big(\frac{\lambda_t}{\nu + \lambda_t}\Big)^{y} \\
&\approx \frac{\sqrt{\frac{2\pi}{\nu+y}}\Big(\frac{\nu+y}{e}\Big)^{\nu+y}}{y!\sqrt{\frac{2\pi}{\nu}}\Big(\frac{\nu}{e}\Big)^{\nu}} \Big(\frac{1}{1 + \lambda_t/\nu}\Big)^{\nu} \frac{\lambda_t^y}{(\nu + \lambda_t)^y} \\
&= \sqrt{\frac{\nu}{\nu + y}}\Big(\frac{\nu + y}{\nu}\Big)^{\nu} e^{-y} \Big(\frac{\nu + y}{\nu + \lambda_t}\Big)^{y} \Big(1 + \frac{\lambda_t}{\nu}\Big)^{-\nu} \frac{\lambda_t^y}{y!} \\
&= \sqrt{\frac{\nu}{\nu + y}}\Big(1 + \frac{y}{\nu}\Big)^{\nu} e^{-y} \Big(\frac{\nu + y}{\nu + \lambda_t}\Big)^{y} \Big(1 + \frac{\lambda_t}{\nu}\Big)^{-\nu} \frac{\lambda_t^y}{y!} \\
&\overset{\nu\to\infty}{\longrightarrow} \frac{e^{-\lambda_t}\lambda_t^y}{y!} = (3.2)
\end{aligned}$$

Regardless of the choice of $Z's$, the conditional mean of $\{Y_t\}$, as given by (3.1), is always equal to λ_t, the mean of the Poisson distribution. This is a key observation which allows for consistent estimation when using quasi-likelihood inference based on the Poisson assumption, as we advance in Section 3.3. However, the conditional variance of the

Poisson distribution is equal to λ_t, whereas the conditional variance of (3.1) is equal to $\lambda_t + \sigma_Z^2 \lambda_t^2$. Hence, the mixed Poisson takes into account overdispersion more properly. Although, simple Poisson models, as those studied by Fokianos et al. [34], can take into account overdispersion, it is anticipated that modeling based on mixed Poisson will improve considerably the fit in several cases.

The proposed modeling approach that we take is along the lines of Fokianos et al. [34]. We assume that the mean process depends upon its past values and the past values of the response. In other words, we assume that

$$\lambda_t = f(Y_{t-1}, \ldots, Y_{t-p}, \lambda_{t-1}, \ldots, \lambda_{t-q}), \tag{3.4}$$

where $f(\cdot, \cdot)$ is a parametric function defined on $\mathbb{N}_0^p \times \mathbb{R}_+^q$ and taking values on $(0, \infty)$, with $\mathbb{N}_0 = \{0, 1, 2, \ldots\}$. As we shall see below, there are several examples falling in this class of models; the linear model (see Example 3.2.1) being the most prominent. We consider models that include a feedback mechanism for the mean process $\{\lambda_t\}$ because this class of models successfully accommodates count time series whose autocorrelation function decays slowly. The inclusion of the feedback mechanism makes parsimonious modeling possible. In fact, we will see in Chapter 4, by a simulated example, that models which include a feedback mechanism outperform models without the feedback mechanism in terms of prediction, even in cases where the true data generating process depends exclusively on a large number of lagged variables of the response. Here, it should be mentioned that both (3.2) and (3.3) can be employed as conditional response distributions in connection with model (3.4).

In particular, we focus on the specific case of $p = q = 1$. Then the proposed model is rephrased as

$$Y_t = \tilde{N}_t(0, Z_t \lambda_t] = N_t(0, \lambda_t], \qquad \lambda_t = f(Y_{t-1}, \lambda_{t-1}), \qquad t \in \mathbb{Z}, \tag{3.5}$$

with $\tilde{N}_t$ and Z_t as discussed before. The last display shows that the hidden process $\{\lambda_t\}$ is determined by past functions of lagged responses and the initial value λ_0; this fact can be easily seen after repeated substitution in (3.5). Therefore model (3.5) belongs to the class of observation driven models in the sense of Cox [12]. In addition, (3.5) is quite analogous

to GARCH where the observation equation links the data with the unobserved process.

Our first result offers verifiable conditions for proving existence and stationarity of model (3.5). Furthermore, it is proved, under mild conditions, that the processes under consideration have finite moments. This task is carried out by employing the concept of weak dependence, see Doukhan and Louhichi [26], Dedecker et al. [21] and Doukhan and Wintenberger [27]. The main assumption of the following theorem is a contraction property of the function $f(\cdot)$ in (3.5), see also Fokianos et al. [34], Fokianos and Tjøstheim [37] and Neumann [66]. The proof of the following theorem is postponed to the Appendix.

Theorem 3.2.1 Consider model (3.5). Assume that there exist constants α_1, α_2 of non-negative real numbers such that

$$|f(y,\lambda) - f(y',\lambda')| \leq \alpha_1|\lambda - \lambda'| + \alpha_2|y - y'|.$$

Suppose that $\alpha = \alpha_1 + \alpha_2 < 1$. Then there exists a unique causal solution $\{(Y_t, \lambda_t), t \in \mathbb{Z}\}$ to model (3.5) which is weakly dependent, stationary, ergodic and satisfies $\mathrm{E}\|(Y_0, \lambda_0)\|^r < \infty$, for any $r \in \mathbb{N}$.

Some concrete examples that fall in the above framework are discussed below.

Example 3.2.1 Consider the following simple linear model with

$$\lambda_t = d + a_1\lambda_{t-1} + b_1 Y_{t-1}. \tag{3.6}$$

By a slight abuse of language, this is an example of an integer GARCH model of order $(1,1)$, see Ferland et al. [31]. Fokianos et al. [34] proved that a perturbed version of this model is geometrically ergodic with moments of any order when $0 < a_1 + b_1 < 1$. The unperturbed version has been studied by Neumann [66] and Doukhan et al. [23] who showed weak dependence and existence of moments under the same conditions.

All the above contributions are under the assumption of Poisson distribution. Theorem 3.2.1 shows that under the same conditions, model (3.6) is ergodic and stationary which possesses moments of any order, *but under the mixed Poisson assumption.* It is important to mention that the stationary region avoids dependence on the variance of the mixing variable; this is a crucial fact for applications since we only need to be concerned with the

appropriate choice of the regression parameters (compare with the approach by Zhu [77] for the particular case of the negative binomial). In addition, when developing optimization algorithms for fitting such models, we do not need to take into account any additional nuisance parameters.

It is illuminating to consider the simple linear model (3.6), which is our benchmark, in more detail. First of all, if we consider the particular case where $t \geq 1$, by repeated substitution we have that

$$\begin{aligned}
\lambda_t &= d + a_1\lambda_{t-1} + b_1 Y_{t-1} \\
&= d + a_1(d + a_1\lambda_{t-2} + b_1 Y_{t-2}) + b_1 Y_{t-1} \\
&= d + a_1 d + a_1^2\lambda_{t-2} + a_1 b_1 Y_{t-2} + b_1 Y_{t-1} \\
&= \dots \\
&= d\frac{1-a_1^t}{1-a_1} + a_1^t\lambda_0 + b_1\sum_{i=0}^{t-1} a_1^i Y_{t-i-1}. \qquad (3.7)
\end{aligned}$$

The last display shows that the hidden process $\{\lambda_t\}$ is determined by past functions of lagged responses and the initial value λ_0. Therefore model (3.6) belongs to the class of observation driven models in the sense of Cox [12]. Representation (3.7) explains further the reason that model (3.6) offers a parsimonious way of modeling count time series data whose autocorrelation function decays slowly. The process $\{\lambda_t\}$ depends on a large number of lagged response values so it is expected to provide a more parsimonious model than a model of the form (3.10), which will be discussed later.

Consider now the following decomposition,

$$\begin{aligned}
Y_t &= \lambda_t + (Y_t - \lambda_t) \\
&= \lambda_t + \varepsilon_t \\
&= d + a_1\lambda_{t-1} + b_1 Y_{t-1} + \varepsilon_t \\
&= d + a_1(Y_{t-1} - \varepsilon_{t-1}) + b_1 Y_{t-1} + \varepsilon_t \\
&= d + (a_1 + b_1)Y_{t-1} - a_1\varepsilon_{t-1} + \varepsilon_t.
\end{aligned}$$

Therefore,

$$\Big(Y_t - \frac{d}{1-(a_1+b_1)}\Big) = (a_1+b_1)\Big(Y_{t-1} - \frac{d}{1-(a_1+b_1)}\Big) + \varepsilon_t - a_1\varepsilon_{t-1},$$

which indicates that $\{Y_t\}$ is an ARMA(1,1) model and when $0 < a_1 + b_1 < 1$, is stationary with mean $E(Y_t) = E(\lambda_t) \equiv \mu = d/(1-(a_1+b_1))$.
Assuming that $\{Y_t\}$ is stationary and its conditional distribution is either the Poisson or the negative binomial, we will calculate the mean and the variance of the residuals ε_t.

- Mean:
  $\mathrm{E}(\varepsilon_t) = \mathrm{E}(Y_t - \lambda_t) = \mathrm{E}[\mathrm{E}(Y_t - \lambda_t|\mathcal{F}_{t-1}^{Y,\lambda})] = 0$

- Variance for the Poisson assumption:

$$\begin{aligned}
\mathrm{Var}(\varepsilon_t) &= \mathrm{Var}[\mathrm{E}(\varepsilon_t|\mathcal{F}_{t-1}^{Y,\lambda})] + \mathrm{E}[\mathrm{Var}(\varepsilon_t|\mathcal{F}_{t-1}^{Y,\lambda})] \\
&= \mathrm{E}[\mathrm{Var}(\varepsilon_t|\mathcal{F}_{t-1}^{Y,\lambda})], \text{ since } \mathrm{Var}[\mathrm{E}(\varepsilon_t|\mathcal{F}_{t-1}^{Y,\lambda})] = 0 \\
&= \mathrm{E}[\mathrm{E}(\varepsilon_t^2|\mathcal{F}_{t-1}^{Y,\lambda})] \\
&= \mathrm{E}[\mathrm{E}((Y_t - \lambda_t)^2|\mathcal{F}_{t-1}^{Y,\lambda})] \\
&= \mathrm{E}[\mathrm{Var}(Y_t|\mathcal{F}_{t-1}^{Y,\lambda})] \\
&= \mathrm{E}(\lambda_t) = \mu
\end{aligned}$$

These results state that the sequence $\{\varepsilon_t\}$ is a white noise sequence. Using the ARMA(1,1) representation, we finally calculate the autocovariance function of $\{Y_t\}$. For $h = 0$,

$$\begin{aligned}
\gamma(0) &= \frac{1 + 2\theta_1\phi_1 + \theta_1^2}{1-\phi_1^2}\,\mathrm{Var}(\varepsilon_t), \qquad \text{with } \phi_1 = a_1 + b_1 \text{ and } \theta_1 = -a_1 \\
&= \frac{1 - 2a_1(a_1+b_1) + a_1^2}{1-(a_1+b_1)^2}\,\mu \\
&= \frac{1 - (a_1+b_1)^2 + b_1^2}{1-(a_1+b_1)^2}\,\mu
\end{aligned}$$

and for $|h| \geq 1$,

$$\begin{aligned}\gamma(|h|) &= \frac{(1+\theta_1\phi_1)(\theta_1+\phi_1)}{1-\phi_1^2}\phi_1^{|h|-1}\ \mathrm{Var}(\varepsilon_t) \\ &= \frac{b_1(1-a_1(a_1+b_1))}{1-(a_1+b_1)^2}\ (a_1+b_1)^{|h|-1}\mu.\end{aligned}$$

Hence, if the conditional distribution of the process is the Poisson, then the autocovariance function is given by

$$\mathrm{Cov}\left[Y_t, Y_{t+|h|}\right] = \begin{cases} \dfrac{1-(a_1+b_1)^2+b_1^2}{1-(a_1+b_1)^2}\ \mu, & h=0, \\ \dfrac{b_1(1-a_1(a_1+b_1))}{1-(a_1+b_1)^2}\ \mu(a_1+b_1)^{|h|-1}, & |h|\geq 1. \end{cases}$$

- Variance for the negative binomial assumption:

$$\begin{aligned}\mathrm{Var}(\varepsilon_t) &= \mathrm{Var}[\mathrm{E}(\varepsilon_t|\mathcal{F}_{t-1}^{Y,\lambda})] + \mathrm{E}[\mathrm{Var}(\varepsilon_t|\mathcal{F}_{t-1}^{Y,\lambda})] \\ &= \mathrm{E}[\mathrm{Var}(\varepsilon_t|\mathcal{F}_{t-1}^{Y,\lambda})],\ \text{since } \mathrm{Var}[\mathrm{E}(\varepsilon_t|\mathcal{F}_{t-1}^{Y,\lambda})]=0 \\ &= \mathrm{E}[\mathrm{E}(\varepsilon_t^2|\mathcal{F}_{t-1}^{Y,\lambda})] \\ &= \mathrm{E}[\mathrm{E}((Y_t-\lambda_t)^2|\mathcal{F}_{t-1}^{Y,\lambda})] \\ &= \mathrm{E}[\mathrm{Var}(Y_t|\mathcal{F}_{t-1}^{Y,\lambda})] \\ &= \mathrm{E}(\lambda_t+\lambda_t^2/\nu) \\ &= \mathrm{E}(\lambda_t)+\mathrm{E}(\lambda_t^2)/\nu \\ &= \mu+\mathrm{E}(\lambda_t^2)/\nu \end{aligned} \tag{3.8}$$

Therefore, we need to calculate $\mathrm{E}(\lambda_t^2)$ which is denoted by $\mu^{(2)}$.

$$\begin{aligned}
\mu^{(2)} &= \mathrm{E}(\lambda_t^2) \\
&= \mathrm{E}(d + a_1\lambda_{t-1} + b_1 Y_{t-1})^2 \\
&= \mathrm{E}(d + a_1\lambda_{t-1} + b_1(Y_{t-1} - \lambda_{t-1}) + b_1\lambda_{t-1})^2 \\
&= d^2 + (a_1+b_1)^2\mathrm{E}(\lambda_{t-1}^2) + b_1^2\mathrm{E}(Y_{t-1} - \lambda_{t-1})^2 + 2d(a_1+b_1)\mathrm{E}(\lambda_{t-1}) \\
&= d^2 + (a_1+b_1)^2\mu^{(2)} + b_1^2\mathrm{Var}(\varepsilon_t) + 2d(a_1+b_1)\mu, \quad \text{assuming stationarity of } \{\lambda_t\} \\
&= d^2 + (a_1+b_1)^2\mu^{(2)} + b_1^2(\mu + \mu^{(2)}/\nu) + 2d(a_1+b_1)\mu
\end{aligned}$$

Hence, $\mu^{(2)} = (d^2 + b_1^2\mu + 2d(a_1+b_1)\mu)/(1 - (a_1+b_1)^2 - b_1^2/\nu)$. Substituting $\mu^{(2)}$ and $d = \mu(1 - a_1 - b_1)$ in (3.8), we have that,

$$\begin{aligned}
\mathrm{Var}(\varepsilon_t) &= \mu + \mu^{(2)}/\nu \\
&= \mu + \frac{1}{\nu}\Big[\frac{\mu^2(1-a_1-b_1)^2 + b_1^2\mu + 2\mu(1-a_1-b_1)(a_1+b_1)\mu}{1-(a_1+b_1)^2 - b_1^2/\nu}\Big] \\
&= \frac{1}{\nu}\Big[\frac{\nu\mu(1-(a_1+b_1)^2 - b_1^2/\nu) + \mu^2(1-a_1-b_1)^2 + b^2\mu + 2\mu(1-a_1-b_1)(a_1+b_1)\mu}{1-(a_1+b_1)^2 - b_1^2/\nu}\Big] \\
&= \frac{1}{\nu}\Big[\frac{\nu\mu(1-(a_1+b_1)^2) + \mu^2 - \mu^2(a_1+b_1)^2}{1-(a_1+b_1)^2 - b_1^2/\nu}\Big] \\
&= \frac{1-(a_1+b_1)^2}{1-(a_1+b_1)^2 - b_1^2/\nu}\,(\mu + \frac{\mu^2}{\nu}).
\end{aligned}$$

Therefore, if the distributional assumption is the negative binomial, then the autocovariance function is given by

$$\mathrm{Cov}\left[Y_t, Y_{t+|h|}\right] = \begin{cases} \dfrac{(1-(a_1+b_1)^2 + b_1^2)}{1-(a_1+b_1)^2 - b_1^2/\nu}\Big(\mu + \dfrac{\mu^2}{\nu}\Big), & h = 0, \\ \dfrac{b_1(1-a_1(a_1+b_1))}{1-(a_1+b_1)^2 - b_1^2/\nu}\Big(\mu + \dfrac{\mu^2}{\nu}\Big)(a_1+b_1)^{|h|-1}, & |h| \geq 1. \end{cases}$$

Comparing the above expressions of the autocovariance function for the two distributional assumptions, we conclude that both models allow for overdispersion for count data, but the negative binomial based model should fit better larger values of the autocovariance function.

Consider the following model

$$\lambda_t = f(Y_{t-1}, \ldots, Y_{t-p}, \lambda_{t-1}, \ldots, \lambda_{t-q}) = d + \sum_{i=1}^{q} a_i \lambda_{t-i} + \sum_{j=1}^{p} b_j Y_{t-j}, \tag{3.9}$$

where $d, a_i, b_j > 0$ for all i, j so that the mean process is positive. With some slight abuse of terminology, we can refer to (3.9) as an integer GARCH model of order (p, q) at least for the case of the Poisson distribution where the conditional mean equals to the conditional variance.

Proposition 3.2.1 For model (3.9), a necessary condition on the parameters $a_i,\ i = 1, \ldots, q$ and $b_j,\ j = 1, \ldots, p$ to yield a second-order stationary process, is $0 < \sum_{i=1}^{q} a_i + \sum_{j=1}^{p} b_j < 1$.

The proof is analogous to Proposition 1 of Ferland et al. [31] and it is postponed to the Appendix. Moreover, Doukhan et al. [24] have shown that for Poisson models the mean relationship of (3.5) can be replaced by the infinite memory model $\lambda_t = f(Y_{t-1}, Y_{t-2}, \ldots)$ by assuming a contraction property on $f(\cdot)$ similar to the condition stated in Theorem 3.2.1; see also Doukhan and Kengne [25]. The proof of Theorem 3.2.1 in the Appendix shows that the contraction property yields weak dependence under the mixed Poisson assumption for models of infinite memory; see Doukhan et al. [24]. Regardless the assumed distribution form, the model still enjoys weak dependence and the asymptotic theory for maximum likelihood estimation can be developed along the lines of Bardet and Wintenberger [3]. For models like (3.9), and more generally models of infinite memory which include the feedback from the mean process, weak dependence is quite challenging to be proved; see Doukhan et al. [24] for more.

Another particular case of (3.9) is given by setting $p = 5$ and $q = 0$; that is

$$\lambda_t = d + \sum_{i=1}^{5} b_i Y_{t-i}. \tag{3.10}$$

This is an example of an integer AutoRegressive model of order 5 (integer $AR(5)$ model). It follows that the required condition for model (3.10) to be stationary with moments of any order, is given by $0 < \sum_{i=1}^{5} b_i < 1$. We include this model for the sake of comparison

with models that contain a feedback mechanism like (3.6). We shall see in Chapter 4 that even though the data generating process follows (3.10), it is still more efficient to work with model (3.6) in terms of prediction.

Example 3.2.2 Now, consider the following nonlinear specification for (3.5) (cf. Gao et al. [42])

$$\lambda_t = \frac{d}{(1+\lambda_{t-1})^{\gamma}} + a_1\lambda_{t-1} + b_1 Y_{t-1}, \tag{3.11}$$

provided that all the parameters d, a_1, b_1, γ are positive. The inclusion of the parameter γ introduces a nonlinear perturbation, in the sense that small values of the parameter γ cause (3.11) to approach model (3.6). Moderate values of γ introduce a stronger perturbation. To apply Theorem 3.2.1, consider the multivariate form of the mean value theorem and note that

$$\begin{aligned} |f(y,\lambda) - f(y',\lambda')| &\leq \left\|\frac{\partial f}{\partial \lambda}\right\|_{\infty} |\lambda - \lambda'| + \left\|\frac{\partial f}{\partial y}\right\|_{\infty} |y - y'| \\ &= \left\|\frac{\partial f}{\partial \lambda}\right\|_{\infty} |\lambda - \lambda'| + b_1 |y - y'|. \\ \frac{\partial f(y,\lambda)}{\partial \lambda} &= a_1 - \frac{d\gamma}{(1+\lambda)^{\gamma+1}}. \end{aligned}$$

Since all the parameters are assumed to be positive and $\lambda > 0$, choosing $\alpha = \max\{a_1, d\gamma - a_1\} + b_1$ in Theorem 3.2.1, yields the asserted conclusions under the mixed Poisson assumption. Once again, we observe that the final conclusions do not depend upon σ_Z^2.

Example 3.2.3 Another interesting example of a nonlinear model which yields an exponential autoregressive model for time series of counts (Haggan and Ozaki [47]) is given by

$$\lambda_t = d + (a_1 + c_1 \exp(-\gamma\lambda_{t-1}^2))\lambda_{t-1} + b_1 Y_{t-1}, \tag{3.12}$$

and assuming that $d, a_1, c_1, b_1, \gamma > 0$; see Fokianos and Tjøstheim [37] and Doukhan et al. [23] who study in detail the above model by either employing an approximation lemma or the notion of weak dependence. Several other examples are provided by the class of Smooth

Transition AutoRegressive (STAR) models of which the exponential autoregressive model is a special case (cf. Teräsvirta et al. [71]). It turns out that, when the mixed Poisson assumption is imposed to the conditional distribution of the responses, then Theorem 3.2.1 is true with $0 < a_1 + b_1 + c_1 < 1$; this fact can be proved along the lines of Doukhan et al. [23]. Note again that the parameter σ_Z^2 does not affect the region of ergodicity and stationarity of the joint process $\{(Y_t, \lambda_t),\ t \in \mathbb{Z}\}$.

Note that all conditions required for ergodicity of models (3.11) and (3.12) and more generally (3.5), do not depend upon the variance of Z's. This is an important point because modeling of the regression parameters does not depend on the parameter σ_Z^2; hence software implementation can be more easily carried out.

These examples show the wide applicability of the results to the analysis of integer-valued time series.

3.3 Inference

Maximum likelihood inference for the Poisson model (3.2) and negative binomial model (3.3) has been developed by Fokianos et al. [34] and Christou and Fokianos [11], respectively. Both the above studies develop estimation based on the Poisson likelihood function which for Poisson model (3.2) it is obviously the true likelihood. However, for the negative binomial model this method resembles similarities with quasi-likelihood inference developed for the estimation of GARCH models. For instance Berkes et al. [4], Francq and Zakoïan [39], Mikosch and Straumann [65] and Bardet and Wintenberger [3] among others, study the Gaussian likelihood function irrespectively of the assumed distribution for the error sequence. Such an approach yields consistent estimators of regression parameters under a correct mean specification and it bypasses complicated likelihood functions; see Zeger and Liang [75], Godambe and Heyde [44], Heyde [51].

For the case of mixed Poisson models (3.1), it is impossible, in general, to have readily available a likelihood function. Hence, we resort to a quasi maximum likelihood (QMLE) methodology and we propose to use the Poisson-based score estimating function.

3.3.1 Estimation of Regression Parameters

Suppose that $\{Y_t, t = 1, 2, \ldots, n\}$ denotes the observed count time series. For defining properly the quasi-conditional maximum likelihood estimator (QMLE), consider the Poisson conditional likelihood function and log-likelihood function, respectively, as in Fokianos et al. [34], where $\boldsymbol{\theta}$ denotes the unknown parameter vector,

$$L_n(\boldsymbol{\theta}) = \prod_{t=1}^{n} \frac{\exp(-\lambda_t(\boldsymbol{\theta}))\lambda_t^{Y_t}(\boldsymbol{\theta})}{Y_t!},$$

$$l_n(\boldsymbol{\theta}) = \sum_{t=1}^{n} l_t(\boldsymbol{\theta}) = \sum_{t=1}^{n} \left(Y_t \log \lambda_t(\boldsymbol{\theta}) - \lambda_t(\boldsymbol{\theta})\right). \tag{3.13}$$

Remark 3.3.1 Consider model (3.5) and the associated negative binomial pmf given by (3.3). Then the true log-likelihood function is given by

$$l_n(\boldsymbol{\theta}, \nu) = \sum_{t=1}^{n} \left\{ \log\left(\frac{\Gamma(\nu + Y_t)}{\Gamma(Y_t + 1)\Gamma(\nu)}\right) + \nu \log\left(\frac{\nu}{\nu + \lambda_t(\boldsymbol{\theta})}\right) + Y_t \log\left(\frac{\lambda_t(\boldsymbol{\theta})}{\nu + \lambda_t(\boldsymbol{\theta})}\right) \right\}.$$

Direct maximization of the above quantity with respect to both $\boldsymbol{\theta}$ and ν can be implemented but this might be challenging for some cases. Because our aim is to extend the theory in the general class of mixed Poisson processes we opt for an alternative which avoids complicated likelihood calculations. Hence, it is sensible to approach the problem of estimation by using a quasi-likelihood function, the Poisson-based likelihood function in our case. Inference becomes more challenging under the general setup of mixed Poisson models.

The quasi-score function is defined by

$$\boldsymbol{S}_n(\boldsymbol{\theta}) \;=\; \frac{\partial l_n(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \sum_{t=1}^{n} \frac{\partial l_t(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \sum_{t=1}^{n} \left(\frac{Y_t}{\lambda_t(\boldsymbol{\theta})} - 1\right) \frac{\partial \lambda_t(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}. \tag{3.14}$$

For the linear model (3.6), $\partial\lambda_t(\boldsymbol{\theta})/\partial\boldsymbol{\theta}$ is a three-dimensional vector with components given by

$$\frac{\partial \lambda_t}{\partial d} = 1 + a_1 \frac{\partial \lambda_{t-1}}{\partial d}, \quad \frac{\partial \lambda_t}{\partial a_1} = \lambda_{t-1} + a_1 \frac{\partial \lambda_{t-1}}{\partial a_1}, \quad \frac{\partial \lambda_t}{\partial b_1} = Y_{t-1} + a_1 \frac{\partial \lambda_{t-1}}{\partial b_1}.$$

For the nonlinear model (3.11), and for γ known, $\partial\lambda_t(\boldsymbol{\theta})/\partial\boldsymbol{\theta}$ is a three-dimensional vector with components given by

$$\frac{\partial\lambda_t}{\partial d} = \frac{1}{(1+\lambda_{t-1})^{\gamma}} + \Big(a_1 - \frac{d\gamma}{(1+\lambda_{t-1})^{\gamma+1}}\Big)\frac{\partial\lambda_{t-1}}{\partial d}, \;\; \frac{\partial\lambda_t}{\partial a_1} = \lambda_{t-1} + \Big(a_1 - \frac{d\gamma}{(1+\lambda_{t-1})^{\gamma+1}}\Big)\frac{\partial\lambda_{t-1}}{\partial a_1},$$

and $\dfrac{\partial\lambda_t}{\partial b_1} = Y_{t-1} + \Big(a_1 - \dfrac{d\gamma}{(1+\lambda_{t-1})^{\gamma+1}}\Big)\dfrac{\partial\lambda_{t-1}}{\partial b_1}$.

For the exponential model (3.12), and for γ known, $\partial\lambda_t(\boldsymbol{\theta})/\partial\boldsymbol{\theta}$ is a four-dimensional vector with components given by

$$\begin{aligned}
\frac{\partial\lambda_t}{\partial d} &= 1 + a_1\frac{\partial\lambda_{t-1}}{\partial d} + (1-2\gamma\lambda_{t-1}^2)c_1\exp(-\gamma\lambda_{t-1}^2)\frac{\partial\lambda_{t-1}}{\partial d},\\
\frac{\partial\lambda_t}{\partial a_1} &= \Big(1 - 2\gamma c_1\lambda_{t-1}\exp(-\gamma\lambda_{t-1}^2)\frac{\partial\lambda_{t-1}}{\partial a_1}\Big)\lambda_{t-1} + (a_1 + c_1\exp(-\gamma\lambda_{t-1}^2))\frac{\partial\lambda_{t-1}}{\partial a_1},\\
\frac{\partial\lambda_t}{\partial c_1} &= \Big(1 - 2\gamma c_1\lambda_{t-1}\frac{\partial\lambda_{t-1}}{\partial c_1}\Big)\exp(-\gamma\lambda_{t-1}^2)\lambda_{t-1} + (a_1 + c_1\exp(-\gamma\lambda_{t-1}^2))\frac{\partial\lambda_{t-1}}{\partial c_1}, \text{ and}\\
\frac{\partial\lambda_t}{\partial b_1} &= Y_{t-1} + a_1\frac{\partial\lambda_{t-1}}{\partial b_1} + (1-2\gamma\lambda_{t-1}^2)c_1\exp(-\gamma\lambda_{t-1}^2)\frac{\partial\lambda_{t-1}}{\partial b_1}.
\end{aligned}$$

Note that

$$\begin{aligned}
\mathrm{E}(\boldsymbol{S}_n(\boldsymbol{\theta})|\mathcal{F}_{t-1}^{Y,\lambda}) &= \mathrm{E}\Big(\sum_{t=1}^{n}\Big(\frac{Y_t}{\lambda_t(\boldsymbol{\theta})} - 1\Big)\frac{\partial\lambda_t(\boldsymbol{\theta})}{\partial\boldsymbol{\theta}}\Big|\mathcal{F}_{t-1}^{Y,\lambda}\Big)\\
&= \sum_{t=1}^{n}\frac{1}{\lambda_t(\boldsymbol{\theta})}\frac{\partial\lambda_t(\boldsymbol{\theta})}{\partial\boldsymbol{\theta}}\mathrm{E}(Y_t - \lambda_t(\boldsymbol{\theta})|\mathcal{F}_{t-1}^{Y,\lambda})\\
&= 0, \text{ since } \mathrm{E}(Y_t - \lambda_t(\boldsymbol{\theta})|\mathcal{F}_{t-1}^{Y,\lambda}) = 0.
\end{aligned}$$

The solution of the system of nonlinear equations $\boldsymbol{S}_n(\boldsymbol{\theta}) = \mathbf{0}$, if it exists, yields the QMLE of $\boldsymbol{\theta}$ which we denote by $\hat{\boldsymbol{\theta}}$. By further differentiation of the score equations (3.14) we obtain the Hessian matrix

$$\begin{aligned}
\boldsymbol{H}_n(\boldsymbol{\theta}) &= -\sum_{t=1}^{n}\frac{\partial^2 l_t(\boldsymbol{\theta})}{\partial\boldsymbol{\theta}\partial\boldsymbol{\theta}'}\\
&= \sum_{t=1}^{n}\frac{Y_t}{\lambda_t^2(\boldsymbol{\theta})}\Big(\frac{\partial\lambda_t(\boldsymbol{\theta})}{\partial\boldsymbol{\theta}}\Big)\Big(\frac{\partial\lambda_t(\boldsymbol{\theta})}{\partial\boldsymbol{\theta}}\Big)' - \sum_{t=1}^{n}\Big(\frac{Y_t}{\lambda_t(\boldsymbol{\theta})} - 1\Big)\frac{\partial^2\lambda_t(\boldsymbol{\theta})}{\partial\boldsymbol{\theta}\partial\boldsymbol{\theta}'}. \quad (3.15)
\end{aligned}$$

An important quantity involved in asymptotic theory is the conditional information

matrix defined by

$$\begin{aligned}
\boldsymbol{G}_n(\boldsymbol{\theta}) &= \sum_{t=1}^{n} \mathrm{Var}\Big(\frac{\partial l_t(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\Big|\mathcal{F}_{t-1}^{Y,\lambda}\Big) = \sum_{t=1}^{n} \mathrm{Var}\Big[\Big(\frac{Y_t}{\lambda_t(\boldsymbol{\theta})}-1\Big)\frac{\partial \lambda_t(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\Big|\mathcal{F}_{t-1}^{Y,\lambda}\Big] \\
&= \sum_{t=1}^{n} \frac{1}{\lambda_t^2(\boldsymbol{\theta})}\left(\frac{\partial \lambda_t(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right)\left(\frac{\partial \lambda_t(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right)' \mathrm{Var}(Y_t|\mathcal{F}_{t-1}^{Y,\lambda}) \\
&= \sum_{t=1}^{n} \frac{1}{\lambda_t^2(\boldsymbol{\theta})}\left(\frac{\partial \lambda_t(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right)\left(\frac{\partial \lambda_t(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right)' (\lambda_t+\sigma_Z^2\lambda_t^2) \\
&= \sum_{t=1}^{n}\Big(\frac{1}{\lambda_t(\boldsymbol{\theta})}+\sigma_Z^2\Big)\left(\frac{\partial \lambda_t(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right)\left(\frac{\partial \lambda_t(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right)'.
\end{aligned}$$

In particular, as mentioned before, the conditional variance of the Poisson process is given by λ_t, whereas for the case of the negative binomial is equal to $\lambda_t+\lambda_t^2/\nu$. Therefore the conditional information matrices for these two particular distributional assumptions are given by

$$\boldsymbol{G}_n(\boldsymbol{\theta}) = \sum_{t=1}^{n} \frac{1}{\lambda_t(\boldsymbol{\theta})}\left(\frac{\partial \lambda_t(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right)\left(\frac{\partial \lambda_t(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right)',$$

and

$$\boldsymbol{G}_{1,n}(\boldsymbol{\theta}) = \sum_{t=1}^{n}\Big(\frac{1}{\lambda_t(\boldsymbol{\theta})}+\frac{1}{\nu}\Big)\left(\frac{\partial \lambda_t(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right)\left(\frac{\partial \lambda_t(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right)',$$

respectively.

To study the asymptotic properties of the QMLE we take advantage of the fact that the log-likelihood function is three times differentiable; this methodology is similar in spirit of Fokianos et al. [34] but with an essential difference. In Fokianos et al. [34], the authors are using a perturbed model to derive the asymptotic properties of the maximum likelihood estimator of $\boldsymbol{\theta}$ for the Poisson model, which makes the approach complicated. In the following theorem we use the contraction principle for $f(\cdot)$ that was employed in the proof of Theorem 3.2.1. In addition, some additional standard assumptions shown in the Appendix make possible the approximation of the derivatives of $f(\cdot)$ by linear functions of its arguments. These assumptions are commonly used in estimation of nonlinear time

series models and they are relatively mild. The proof employs again the notion of weak dependence. The final result asserts that the QMLE is asymptotically normal with a sandwich type variance matrix; a standard result also in GARCH literature. The theorem covers the case of model (3.5). For a proof regarding model (3.9), methods like those employed by Berkes et al. [4] and Francq and Zakoïan [39] can be employed to study asymptotic inference.

Theorem 3.3.1 Given model (3.5) and under the assumptions of Theorem 3.2.1 and **A-1-A-3** listed in the Appendix, there exists an open neighborhood $O = O(\boldsymbol{\theta_0})$ of the true value $\boldsymbol{\theta_0}$ such that the QMLE $\hat{\boldsymbol{\theta}}$ is consistent and asymptotically normal. In other words

$$\sqrt{n}(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta_0}) \xrightarrow{D} \mathcal{N}(\mathbf{0}, \boldsymbol{G}^{-1}(\boldsymbol{\theta_0})\boldsymbol{G}_1(\boldsymbol{\theta_0})\boldsymbol{G}^{-1}(\boldsymbol{\theta_0})),$$

where the matrices $\boldsymbol{G}$ and $\boldsymbol{G}_1$ are given by

$$\boldsymbol{G}(\boldsymbol{\theta}) = \mathrm{E}\left(\frac{1}{\lambda_t(\boldsymbol{\theta})}\left(\frac{\partial \lambda_t}{\partial \boldsymbol{\theta}}\right)\left(\frac{\partial \lambda_t}{\partial \boldsymbol{\theta}}\right)'\right), \tag{3.16}$$

and

$$\boldsymbol{G}_1(\boldsymbol{\theta}) = \mathrm{E}\left(\left(\frac{1}{\lambda_t(\boldsymbol{\theta})} + \sigma_Z^2\right)\left(\frac{\partial \lambda_t}{\partial \boldsymbol{\theta}}\right)\left(\frac{\partial \lambda_t}{\partial \boldsymbol{\theta}}\right)'\right). \tag{3.17}$$

Given the Poisson model, $\boldsymbol{G}_1 = \boldsymbol{G}$ since $\sigma_Z^2 = 0$ and therefore

$$\sqrt{n}(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta_0}) \xrightarrow{D} \mathcal{N}(\mathbf{0}, \boldsymbol{G}^{-1}(\boldsymbol{\theta_0})).$$

In the above, expectation is taken with respect to the stationary distribution. All the above quantities can be computed, and they are employed for constructing predictions, confidence interval and so on. To show the validity of Theorem 3.3.1 it is enough to show the following Lemma.

Lemma 3.3.1 Under the assumptions of Theorem 3.2.1 we have the following results, as $n \to \infty$:

(i) The score function defined by (3.14) satisfies

$$\frac{1}{\sqrt{n}}\boldsymbol{S}_n(\boldsymbol{\theta_0}) \xrightarrow{D} \mathcal{N}(\mathbf{0}, \boldsymbol{G}_1(\boldsymbol{\theta_0})),$$

where $\xrightarrow{D}$ stands for convergence in distribution.

(ii) The Hessian matrix defined by (3.15) satisfies

$$\frac{1}{n}\boldsymbol{H}_n(\boldsymbol{\theta_0}) \xrightarrow{p} \boldsymbol{G}(\boldsymbol{\theta_0}),$$

where $\xrightarrow{p}$ stands for convergence in probability.

(iii) Within the neighborhood $O(\boldsymbol{\theta_0})$ of the true value

$$\max_{i,j,k=1,2,3} \sup_{\boldsymbol{\theta}\in O(\boldsymbol{\theta_0})} \left| \frac{1}{n}\sum_{t=1}^{n} \frac{\partial^3 l_t(\boldsymbol{\theta})}{\partial\boldsymbol{\theta}_i \partial\boldsymbol{\theta}_j \partial\boldsymbol{\theta}_k} \right| \leq M_n,$$

such that $M_n \xrightarrow{p} M$, with M a constant.

Remark 3.3.2 In practise, maximization of (3.13) is implemented by assuming some fixed or random value of λ_0, say $\lambda_0 = 0$ or $\lambda_0 = \bar{Y}$. Then, technically speaking, if we denote by $\tilde{\lambda}_0$ the starting value of process (3.5) we observe the log-likelihood function

$$\tilde{l}_n(\boldsymbol{\theta}) = \sum_{t=1}^{n} \tilde{l}_t(\boldsymbol{\theta}) = \sum_{t=1}^{n} \left(Y_t \log \tilde{\lambda}_t(\boldsymbol{\theta}) - \tilde{\lambda}_t(\boldsymbol{\theta}) \right), \tag{3.18}$$

with $\tilde{\lambda}_t(\boldsymbol{\theta}) = f(Y_{t-1}, \tilde{\lambda}_{t-1}(\boldsymbol{\theta}))$. We would like to approximate (3.18) by (3.13) with $\{(Y_t, \lambda_t), t \in \mathbb{Z}\}$ the stationary and ergodic process obtained by Theorem 3.2.1. Lemma A-2 in the Appendix shows that indeed this is the case. Similar results hold for GARCH models, see Berkes et al. [4], Francq and Zakoïan [39], Mikosch and Straumann [65] and Meitz and Saikkonen [63]. Although (3.18) is the observed likelihood function, we will use (3.13) in the proofs of Theorem 3.3.1 and Lemma 3.3.1.

3.3.2 Estimation of σ_Z^2

The previous subsection shows how to obtain the QMLE of the regression parameters. If $\sigma_Z^2 > 0$, it remains to estimate this additional parameter. One of several possible estimators of σ_Z^2, proposed by Gourieroux et al. [45, 46], is given by

$$\hat{\sigma}_Z^2 = \frac{1}{n}\sum_{t=1}^{n} \frac{[(Y_t - \hat{\lambda}_t)^2 - \hat{\lambda}_t]}{\hat{\lambda}_t^2}, \tag{3.19}$$

where $\hat{\lambda}_t = \lambda_t(\hat{\boldsymbol{\theta}})$. This is simply a moment-based estimator. The motivation to consider this particular estimator for σ_Z^2 is given by the fact that

$$\mathrm{E}\Big[\frac{(Y_t - \lambda_t)^2 - \lambda_t}{\lambda_t^2}\Big] = \sigma_Z^2.$$

The above display is proved by using properties of iterated expectation and substituting the mean and variance of the mixed Poisson process, i.e.

$$\begin{aligned}
\mathrm{E}\Big[\frac{(Y_t - \lambda_t)^2 - \lambda_t}{\lambda_t^2}\Big] &= \mathrm{E}\Big[\mathrm{E}\Big(\frac{(Y_t - \lambda_t)^2 - \lambda_t}{\lambda_t^2}\Big|\mathcal{F}_{t-1}^{Y,\lambda}\Big)\Big] \\
&= \mathrm{E}\Big[\frac{\mathrm{Var}(Y_t|\mathcal{F}_{t-1}^{Y,\lambda}) - \lambda_t}{\lambda_t^2}\Big] \\
&= \mathrm{E}\Big[\frac{\lambda_t + \sigma_Z^2\lambda_t^2 - \lambda_t}{\lambda_t^2}\Big] \\
&= \sigma_Z^2.
\end{aligned}$$

Another approach for obtaining an estimator of σ_Z^2 is by solving the equation

$$\sum_{t=1}^{n} \frac{(Y_t - \hat{\lambda}_t)^2}{\hat{\lambda}_t(1 + \sigma_Z^2\hat{\lambda}_t)} = n - m, \tag{3.20}$$

where m denotes the dimension of $\boldsymbol{\theta}$ and $\hat{\lambda}_t = \lambda_t(\hat{\boldsymbol{\theta}})$. Such an estimator can be calculating by existing software and it is consistent because it is obtained by equating the Pearson chi-square statistic to its degrees of freedom (cf. Dean et al. [20] and Breslow [7]).

Note that the first estimator is a moment based estimator and can be used regardless the response distribution. The second estimator is based on the Pearson chi-square test

statistic and therefore carries more information about the data generating process.

In particular, we recognize that for the case of negative binomial, $\sigma_Z^2 = 1/\nu$ and the above estimator is standard in applications, see Cameron and Trivedi [10, Ch. 3] and Lawless [60] for instance.

3.4 Simulations

In this section, the proposed approach of quasi maximum likelihood estimation is illustrated empirically by means of a simulated study. We consider fitting the linear model (3.6) and the nonlinear model (3.11) to data generated from the mixed Poisson process with mixing variable Z from gamma, chi-square, uniform, binomial or geometric distribution. All results are based on 1000 runs. Calculation of the maximum likelihood estimators is carried out by optimizing the log-likelihood function (3.13) by a quasi-Newton method using the `constrOptim()` function of `R`; see R Development Core Team [68] for instance. We consider sample sizes of $n = 200, n = 500$ and $n = 1000$ throughout the simulations. Estimation is implemented by discarding the first 500 observations so that the process is within its stationary region.

3.4.1 Mixing Variable Z from Gamma Distribution

Recall again that the mixed Poisson process with mixing variable Z from a gamma distribution with mean 1 and variance $\sigma_Z^2 = 1/\nu$, sets up a negative binomial distribution. Therefore, we generate data from the linear model (3.6) and the nonlinear model (3.11) according to the pmf of the negative binomial given by (3.3). The linear model (3.6) is generated with true values $(d, a_1, b_1) = (0.5, 0.4, 0.5)$ and for different values of the parameter ν. Note that the parameter values where chosen to satisfy $a_1 + b_1 = 0.9$ which is close to the sum $\hat{a}_1 + \hat{b}_1$ obtained from the estimates of the real data examples given in the next section. In addition, this selection of the set of parameter values satisfies the condition $0 < a_1 + b_1 < 1$, which is necessary for the ergodicity and stationarity of the process. Table 3.1 shows the results from estimation of the linear model (3.6). In all cases, we note that the maximum likelihood estimators are consistent and as the sample size increases the approximation improves. Some deviations are observed for the estimator $\hat{d}$ but this

issue, which also occurs for the ordinary GARCH models, has been observed before by Fokianos et al. [35]; see their Figure 1(a).

Besides estimation results, we also report the standard errors of the estimators obtained by simulation (first row in parentheses) and the true standard errors of the estimators obtained by calculating the matrices $\boldsymbol{G}$ and $\boldsymbol{G}_1$–recall Theorem 3.3.1–at the true values (second row in parentheses). In all cases, we do not observe any gross discrepancies between theory and simulation for the linear model, especially for large sample sizes. Note that all standard errors decrease as the sample size increases. To investigate further the finite sample performance of the Hessian matrix–see the second assertion of Lemma 3.3.1–we evaluate both matrices at the true value of $\boldsymbol{\theta}_0$ for the linear model. Table 3.2 shows the comparison between the Hessian matrix and $\boldsymbol{G}$ for 1000 simulations. We conclude that the convergence is quite satisfactory.

As far as the estimator of the parameter ν is concerned, we note that for all cases considered, the estimator $\hat{\nu}_2$ based on (3.20) is superior to the estimator $\hat{\nu}_1$ from (3.19) because it gives estimates closer to the true parameter ν. Furthermore, Figure 3.1 shows histograms and qq-plots for the sampling distribution of the standardized maximum likelihood estimators of the linear model (3.6). The plots indicate the adequacy of the normal approximation. In addition, Kolmogorov-Smirnov test for the sampling distribution of the standardized QMLE, gives large p-values, $0.213, 0.824$ and 0.783 for testing against the standard normal distribution. This is consistent with the findings from Figure 3.1. Similar results were obtained for the nonlinear model (3.11). Using the same set of parameters, and setting $\gamma = 0.5$, we see from Tables 3.3, 3.4 and from Figure 3.2 the asserted asymptotic normality and the convergence of the Hessian matrix to $\boldsymbol{G}$.

Note that, when $\nu \to \infty$, the pmf given by (3.3) tends to a standard Poisson pmf. Hence, it is instructive to compare the two matrices $\boldsymbol{G}$ and $\boldsymbol{G}_1$ given by (3.16) and (3.17) respectively. To do so, we calculate the ratio of the standard errors of the estimators $\hat{d}, \hat{a}_1, \hat{b}_1$ obtained by inversion of the matrix $\boldsymbol{G}$ to the standard errors of the estimators $\hat{d}, \hat{a}_1, \hat{b}_1$ obtained from the sandwich matrix $\boldsymbol{G}^{-1}\boldsymbol{G}_1\boldsymbol{G}^{-1}$, for both linear and nonlinear model, using the same set of parameter values. For this comparison, we generate data by varying $\nu \in \{1, \ldots, 15\}$. Figure 3.3 shows the results of this exercise; when ν increases all the ratios tend to unity because of the fact that the sandwich matrix $\boldsymbol{G}^{-1}\boldsymbol{G}_1\boldsymbol{G}^{-1}$ tends

QMLE			Estimators of ν		Sample Size	True ν
$\hat{d}$	$\hat{a}_1$	$\hat{b}_1$	$\hat{\nu}_1$	$\hat{\nu}_2$	n	ν
0.666	0.368	0.480	2.177	2.074	200	2
(0.267)	(0.111)	(0.087)	(0.531)	(0.443)		
(0.219)	(0.108)	(0.104)				
0.567	0.387	0.492	2.067	2.018	500	
(0.142)	(0.072)	(0.064)	(0.299)	(0.266)		
(0.140)	(0.071)	(0.070)				
0.537	0.394	0.493	2.033	2.007	1000	
(0.097)	(0.049)	(0.044)	(0.221)	(0.199)		
(0.102)	(0.052)	(0.051)				
0.666	0.367	0.486	3.372	3.155	200	3
(0.262)	(0.105)	(0.083)	(0.924)	(0.727)		
(0.210)	(0.099)	(0.090)				
0.561	0.392	0.490	3.118	3.037	500	
(0.141)	(0.064)	(0.057)	(0.503)	(0.418)		
(0.132)	(0.064)	(0.059)				
0.531	0.396	0.495	3.061	3.026	1000	
(0.095)	(0.046)	(0.040)	(0.332)	(0.284)		
(0.094)	(0.046)	(0.043)				
0.656	0.366	0.494	5.668	5.192	200	5
(0.260)	(0.097)	(0.077)	(1.967)	(1.417)		
(0.205)	(0.092)	(0.080)				
0.566	0.385	0.498	5.319	5.127	500	
(0.130)	(0.058)	(0.049)	(1.110)	(0.856)		
(0.126)	(0.058)	(0.051)				
0.531	0.394	0.499	5.166	5.071	1000	
(0.088)	(0.041)	(0.034)	(0.729)	(0.574)		
(0.089)	(0.042)	(0.036)				

Table 3.1: Quasi maximum likelihood estimators and their standards errors (in parentheses) for the linear model (3.6) when $(d, a_1, b_1) = (0.5, 0.4, 0.5)$, $\nu \, \epsilon \, \{2, 3, 5\}$ and for different sample sizes. The first row in parentheses reports the standard errors of the estimators obtained by simulation and the second row in parentheses gives the standard errors of the estimators obtained by calculating the matrices $\boldsymbol{G}$ and $\boldsymbol{G}_1$ at the true values. Results are based on 1000 simulations.

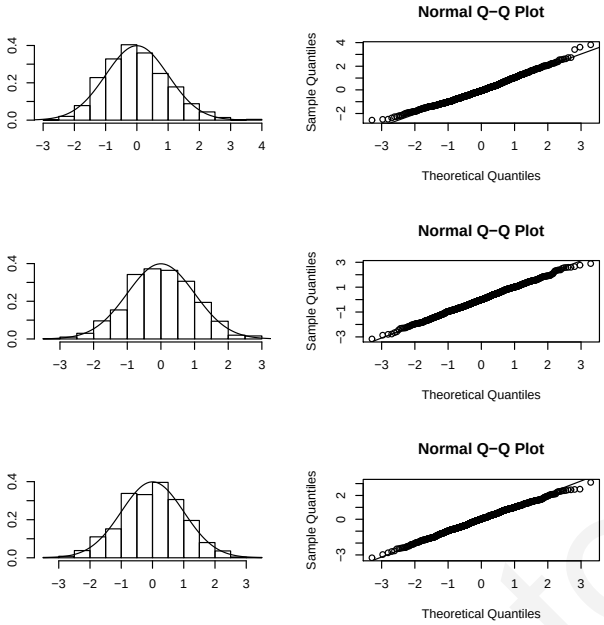

Figure 3.1: From top to bottom: Histograms and qq-plots of the sampling distribution of the standardized estimators of $\hat{\boldsymbol{\theta}} = (\hat{d}, \hat{a}_1, \hat{b}_1)$ for the linear model (3.6) when the true values are $(d, a_1, b_1) = (0.5, 0.4, 0.5)$ and $\nu = 5$. Superimposed is the standard normal density function. The results are based on 1000 data points and 1000 simulations.

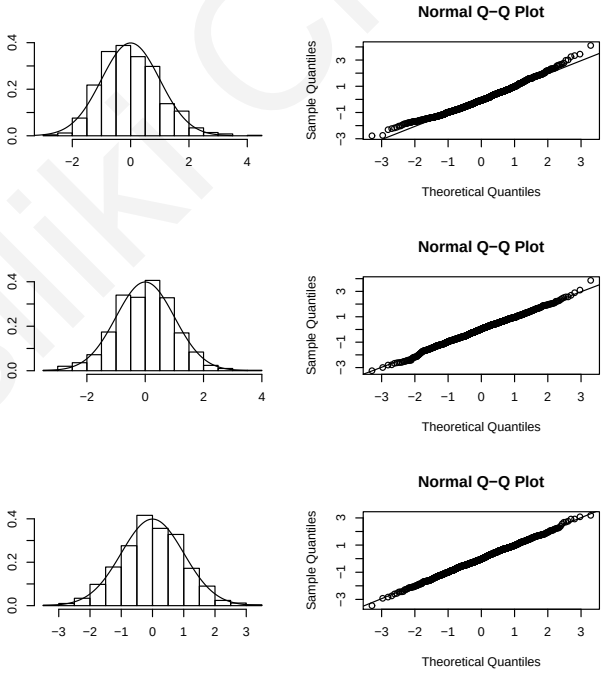

Figure 3.2: From top to bottom: Histograms and qq-plots of the sampling distribution of the standardized estimators of $\hat{\boldsymbol{\theta}} = (\hat{d}, \hat{a}_1, \hat{b}_1)$ for the nonlinear model (3.11) when the true values are $(d, a_1, b_1) = (0.5, 0.4, 0.5)$, $\gamma = 0.5$ and $\nu = 5$. Superimposed is the standard normal density function. The results are based on 1000 data points and 1000 simulations.

	$\nu = 2$	$\nu = 3$	$\nu = 5$
n=200	$\begin{pmatrix} 0.001 & 0.003 & 0.003 \\ 0.003 & 0.016 & 0.014 \\ 0.003 & 0.014 & 0.015 \end{pmatrix}$	$\begin{pmatrix} 0.001 & 0.003 & 0.003 \\ 0.003 & 0.017 & 0.015 \\ 0.003 & 0.015 & 0.016 \end{pmatrix}$	$\begin{pmatrix} 0.001 & 0.003 & 0.003 \\ 0.003 & 0.017 & 0.015 \\ 0.003 & 0.015 & 0.017 \end{pmatrix}$
n=1000	$\begin{pmatrix} 0.001 & 0.003 & 0.002 \\ 0.003 & 0.017 & 0.014 \\ 0.002 & 0.014 & 0.015 \end{pmatrix}$	$\begin{pmatrix} 0.001 & 0.003 & 0.003 \\ 0.003 & 0.017 & 0.016 \\ 0.003 & 0.016 & 0.017 \end{pmatrix}$	$\begin{pmatrix} 0.001 & 0.003 & 0.002 \\ 0.003 & 0.015 & 0.013 \\ 0.002 & 0.013 & 0.013 \end{pmatrix}$
n=200	$\begin{pmatrix} 0.001 & 0.003 & 0.002 \\ 0.003 & 0.018 & 0.014 \\ 0.002 & 0.014 & 0.015 \end{pmatrix}$	$\begin{pmatrix} 0.001 & 0.003 & 0.003 \\ 0.003 & 0.018 & 0.015 \\ 0.003 & 0.015 & 0.016 \end{pmatrix}$	$\begin{pmatrix} 0.001 & 0.003 & 0.003 \\ 0.003 & 0.017 & 0.016 \\ 0.003 & 0.016 & 0.016 \end{pmatrix}$
n=1000	$\begin{pmatrix} 0.001 & 0.003 & 0.002 \\ 0.003 & 0.017 & 0.014 \\ 0.002 & 0.014 & 0.015 \end{pmatrix}$	$\begin{pmatrix} 0.001 & 0.003 & 0.002 \\ 0.003 & 0.018 & 0.016 \\ 0.002 & 0.016 & 0.017 \end{pmatrix}$	$\begin{pmatrix} 0.001 & 0.003 & 0.002 \\ 0.003 & 0.015 & 0.013 \\ 0.002 & 0.013 & 0.013 \end{pmatrix}$

Table 3.2: Empirical illustration of the second assertion of Lemma 3.3.1 for the linear model (3.6) with true values $(d, a_1, b_1) = (0.5, 0.4, 0.5)$, $\nu \, \epsilon \, \{2, 3, 5\}$ and for different sample sizes. The upper panel reports the Hessian matrix. The lower panel reports the entries of matrix $\boldsymbol{G}$. Results are based on 1000 simulations.

to the $\boldsymbol{G}^{-1}$ matrix. However, the convergence for the nonlinear model is faster than that of the linear model.

3.4.2 Mixing Variable Z from Chi-Square, Uniform, Binomial or Geometric Distribution

We consider now a mixing variable from the $\mathcal{X}_1^2$, $U(0,2)$, $Binomial(10, 1/10)$ or $Geometric(5/10)$. Note that the mean of the mixing variable is always equal to 1. We fit the linear model (3.6) and the nonlinear model (3.11) with true values $(d, a_1, b_1) = (0.5, 0.3, 0.4)$ and $\gamma = 0.5$ for the nonlinear model. This selection of the parameters satisfies the conditions $0 < a_1 + b_1 < 1$ and $\alpha = \max\{a_1, d\gamma - a_1\} + b_1$, which are necessary in order to have ergodic and stationary model in each case. Tables 3.5 and 3.6 show the results for the estimators of the regression parameters. For both models, the estimators are consistent and their standard errors decrease as the sample size increases. In addition, the variance of the mixing variable Z is estimated consistently. Furthermore, Figure 3.4 shows histograms and qq-plots for the sampling distribution of the standardized maximum likelihood estimators of the linear model (3.6), when the mixing variable follows the $U(0,2)$

QMLE			Estimators of ν		Sample Size	True ν
$\hat{d}$	$\hat{a}_1$	$\hat{b}_1$	$\hat{\nu}_1$	$\hat{\nu}_2$	n	ν
0.612	0.382	0.478	2.316	2.096	200	2
(0.236)	(0.107)	(0.093)	(0.965)	(0.615)		
(0.201)	(0.104)	(0.104)				
0.551	0.393	0.489	2.112	2.024	500	
(0.139)	(0.067)	(0.061)	(0.434)	(0.337)		
(0.128)	(0.068)	(0.069)				
0.526	0.397	0.493	2.060	2.015	1000	
(0.092)	(0.051)	(0.047)	(0.281)	(0.225)		
(0.092)	(0.049)	(0.050)				
0.618	0.377	0.484	3.601	3.177	200	3
(0.246)	(0.103)	(0.088)	(1.731)	(1.045)		
(0.196)	(0.096)	(0.091)				
0.544	0.393	0.492	3.207	3.055	500	
(0.129)	(0.061)	(0.055)	(0.776)	(0.575)		
(0.122)	(0.061)	(0.059)				
0.514	0.400	0.494	3.102	3.027	1000	
(0.084)	(0.045)	(0.042)	(0.506)	(0.391)		
(0.087)	(0.044)	(0.042)				
0.619	0.380	0.492	6.211	5.198	200	5
(0.233)	(0.090)	(0.079)	(2.987)	(1.877)		
(0.190)	(0.091)	(0.082)				
0.543	0.393	0.495	5.671	5.164	500	
(0.129)	(0.060)	(0.053)	(2.321)	(1.357)		
(0.118)	(0.057)	(0.052)				
0.520	0.396	0.497	5.276	5.064	1000	
(0.085)	(0.040)	(0.035)	(1.211)	(0.805)		
(0.083)	(0.040)	(0.037)				

Table 3.3: Quasi maximum likelihood estimators and their standards errors (in parentheses) for the nonlinear model (3.11) when $(d, a_1, b_1) = (0.5, 0.4, 0.5)$, $\nu \, \epsilon \, \{2, 3, 5\}$, $\gamma = 0.5$ and for different sample sizes. The first row in parentheses reports the standard errors of the estimators obtained by simulation and the second row in parentheses gives the standard errors of the estimators obtained by calculating the matrices $\boldsymbol{G}$ and $\boldsymbol{G}_1$ at the true values. Results are based on 1000 simulations.

distribution. The plots indicate the adequacy of the normal approximation.

3.5 Case Studies

In this section, we apply the above theory to real data examples. For all the collections, the mean is always less than the corresponding variance. In other words, the data exhibits overdispersion. Motivated by this fact, we fit the linear model (3.6) and the nonlinear

	$\boldsymbol{\nu = 2}$	$\boldsymbol{\nu = 3}$	$\boldsymbol{\nu = 5}$
n=200	$\begin{pmatrix} 0.001 & 0.002 & 0.001 \\ 0.002 & 0.009 & 0.007 \\ 0.001 & 0.007 & 0.007 \end{pmatrix}$	$\begin{pmatrix} 0.001 & 0.002 & 0.001 \\ 0.002 & 0.008 & 0.007 \\ 0.001 & 0.007 & 0.007 \end{pmatrix}$	$\begin{pmatrix} 0.000 & 0.001 & 0.001 \\ 0.001 & 0.011 & 0.010 \\ 0.001 & 0.010 & 0.012 \end{pmatrix}$
n=1000	$\begin{pmatrix} 0.001 & 0.001 & 0.001 \\ 0.001 & 0.009 & 0.007 \\ 0.001 & 0.007 & 0.008 \end{pmatrix}$	$\begin{pmatrix} 0.001 & 0.002 & 0.001 \\ 0.002 & 0.009 & 0.008 \\ 0.001 & 0.008 & 0.009 \end{pmatrix}$	$\begin{pmatrix} 0.001 & 0.002 & 0.001 \\ 0.002 & 0.007 & 0.006 \\ 0.001 & 0.006 & 0.006 \end{pmatrix}$
n=200	$\begin{pmatrix} 0.001 & 0.002 & 0.001 \\ 0.002 & 0.008 & 0.007 \\ 0.001 & 0.007 & 0.007 \end{pmatrix}$	$\begin{pmatrix} 0.001 & 0.001 & 0.001 \\ 0.001 & 0.008 & 0.007 \\ 0.001 & 0.007 & 0.007 \end{pmatrix}$	$\begin{pmatrix} 0.000 & 0.001 & 0.001 \\ 0.001 & 0.011 & 0.010 \\ 0.001 & 0.010 & 0.011 \end{pmatrix}$
n=1000	$\begin{pmatrix} 0.001 & 0.002 & 0.001 \\ 0.002 & 0.009 & 0.007 \\ 0.001 & 0.007 & 0.008 \end{pmatrix}$	$\begin{pmatrix} 0.001 & 0.001 & 0.001 \\ 0.001 & 0.009 & 0.008 \\ 0.001 & 0.008 & 0.009 \end{pmatrix}$	$\begin{pmatrix} 0.001 & 0.002 & 0.001 \\ 0.002 & 0.007 & 0.006 \\ 0.001 & 0.006 & 0.007 \end{pmatrix}$

Table 3.4: Empirical illustration of the second assertion of Lemma 3.3.1 for the nonlinear model (3.11) with true values $(d, a_1, b_1) = (0.5, 0.4, 0.5)$, $\gamma = 0.5$, $\nu \in \{2, 3, 5\}$ and for different sample sizes. The upper panel reports the Hessian matrix. The lower panel reports the entries of matrix $\boldsymbol{G}$. Results are based on 1000 simulations.

model (3.11) to the data using the negative binomial distribution as discussed before. To initialize the recursions, we set $\lambda_0 = 0$ and $\partial\lambda_0/\partial\boldsymbol{\theta} = 0$ for the linear model and $\lambda_0 = 1$ and $\partial\lambda_0/\partial\boldsymbol{\theta} = 1$ for the nonlinear model. In addition, for the nonlinear model we choose the value of the parameter γ to be 0.5. Using the quasi-likelihood methodology outlined in Section 3.3 we obtain the quasi maximum likelihood estimators for the regression parameters.

3.5.1 Transactions Data

The left plot of Figure 3.5 shows the total number of transactions per minute for the stock Ericsson B for the time period between July 2nd and July 22nd, 2002. We consider the first ten days of reported data, that is 460 observations. The right plot of Figure 3.5 shows the autocorrelation function (acf) for these data. The mean of the series is 9.909 while the sample variance is 32.836, that is the data are overdispersed. The results of the quasi maximum likelihood estimators for the regression parameters are summarized in Table 3.7. Together with the estimators of d, a_1 and b_1, we give standard errors where the first row of standard errors are under the negative binomial distribution and are obtained by

Mixing Variable Z	QMLE			$\hat{\sigma}_Z^2$	Sample Size	True σ_Z^2
	$\hat{d}$	$\hat{a}_1$	$\hat{b}_1$	$\hat{\sigma}_Z^2$	n	σ_Z^2
$\mathcal{X}_1^2$	0.591	0.260	0.356	1.983	200	2
	(0.213)	(0.181)	(0.130)	(0.735)		
	0.546	0.280	0.377	2.008	500	
	(0.148)	(0.129)	(0.088)	(0.476)		
	0.527	0.289	0.388	2.014	1000	
	(0.105)	(0.094)	(0.067)	(0.316)		
$U(0,2)$	0.579	0.253	0.394	0.340	200	0.333
	(0.184)	(0.139)	(0.082)	(0.104)		
	0.527	0.286	0.396	0.334	500	
	(0.118)	(0.093)	(0.052)	(0.066)		
	0.511	0.293	0.399	0.334	1000	
	(0.077)	(0.062)	(0.036)	(0.045)		
$Binomial(10, 1/10)$	0.583	0.256	0.379	0.920	200	0.9
	(0.198)	(0.156)	(0.098)	(0.280)		
	0.534	0.282	0.392	0.909	500	
	(0.119)	(0.098)	(0.065)	(0.133)		
	0.513	0.294	0.393	0.899	1000	
	(0.084)	(0.068)	(0.045)	(0.091)		
$Geometric(5/10)$	0.586	0.264	0.359	2.026	200	2
	(0.225)	(0.183)	(0.128)	(0.784)		
	0.540	0.283	0.379	2.023	500	
	(0.137)	(0.121)	(0.086)	(0.379)		
	0.523	0.291	0.387	2.007	1000	
	(0.099)	(0.085)	(0.066)	(0.259)		

Table 3.5: Quasi maximum likelihood estimators and their standards errors (in parentheses) for the linear model (3.6) when $(d, a_1, b_1) = (0.5, 0.3, 0.4)$ and for different sample sizes. The mixing variable Z becomes from the $\mathcal{X}_1^2$, $U(0,2)$, $Binomial(10, 1/10)$ or $Geometric(5/10)$ distribution. Results are based on 1000 simulations.

Mixing Variable Z	QMLE			$\hat{\sigma}_Z^2$	Sample Size	True σ_Z^2
	$\hat{d}$	$\hat{a}_1$	$\hat{b}_1$	$\hat{\sigma}_Z^2$	n	σ_Z^2
$\mathcal{X}_1^2$	0.563	0.276	0.357	1.993	200	2
	(0.234)	(0.188)	(0.135)	(0.631)		
	0.535	0.288	0.377	1.997	500	
	(0.151)	(0.124)	(0.095)	(0.349)		
	0.514	0.296	0.388	2.012	1000	
	(0.101)	(0.086)	(0.071)	(0.247)		
$U(0,2)$	0.542	0.282	0.383	0.337	200	0.333
	(0.185)	(0.142)	(0.088)	(0.139)		
	0.522	0.291	0.393	0.336	500	
	(0.113)	(0.089)	(0.056)	(0.085)		
	0.508	0.296	0.398	0.335	1000	
	(0.077)	(0.059)	(0.040)	(0.058)		
$Binomial(10, 1/10)$	0.557	0.275	0.379	0.914	200	0.9
	(0.192)	(0.150)	(0.102)	(0.238)		
	0.525	0.291	0.391	0.909	500	
	(0.118)	(0.093)	(0.068)	(0.149)		
	0.514	0.295	0.394	0.909	1000	
	(0.085)	(0.066)	(0.050)	(0.149)		
$Geometric(5/10)$	0.560	0.281	0.348	2.066	200	2
	(0.233)	(0.189)	(0.132)	(0.539)		
	0.535	0.287	0.378	2.048	500	
	(0.145)	(0.120)	(0.096)	(0.386)		
	0.519	0.293	0.386	2.015	1000	
	(0.097)	(0.082)	(0.068)	(0.221)		

Table 3.6: Quasi maximum likelihood estimators and their standards errors (in parentheses) for the nonlinear model (3.11) when $(d, a_1, b_1) = (0.5, 0.3, 0.4)$, $\gamma = 0.5$ and for different sample sizes. The mixing variable Z becomes from the $\mathcal{X}_1^2$, $U(0,2)$, $Binomial(10, 1/10)$ or $Geometric(5/10)$ distribution. Results are based on 1000 simulations.

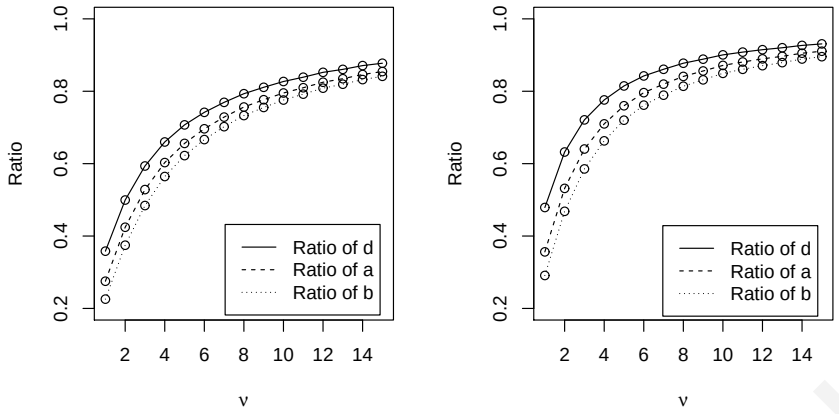

Figure 3.3: Left plot: Comparison of the standard errors of quasi maximum likelihood estimators for the linear model (3.6). Right plot: Comparison of the standard errors of quasi maximum likelihood estimators for the nonlinear model (3.11). The true values are $(d, a_1, b_1) = (0.5, 0.4, 0.5)$ and $\gamma = 0.5$. Data are generated employing the negative binomial distribution (3.3) for different values of ν. The results are based on 1000 data points and 1000 simulations.

the sandwich matrix $\boldsymbol{G}^{-1}(\hat{\boldsymbol{\theta}})\boldsymbol{G}_1(\hat{\boldsymbol{\theta}})\boldsymbol{G}^{-1}(\hat{\boldsymbol{\theta}})$, where the second line corresponds to the case of Poisson distribution. Standard errors for $\hat{\nu}_i$, $i = 1, 2$ have been computed by employing parametric bootstrap. Given $\hat{\boldsymbol{\theta}}$ and $\hat{\nu}_i$, $i = 1, 2$, generate a large number of count time series models by means of (3.6) or (3.11) and using (3.3). For each of the simulated count time series, carry out the QMLE and get an estimator of ν_i using either (3.19) or (3.20). The standard error of these replications is reported in Table 3.7, underneath $\hat{\nu}_i$, $i = 1, 2$.

Compared with the results obtained by Fokianos et al. [34], we note that the regression coefficients are the same; this is an immediate consequence of the model specification (3.6) and the Poisson log-likelihood function. However, the standard errors of the estimators are inflated due to estimation of the parameter ν. In addition, note that the sum $\hat{a}_1 + \hat{b}_1$ is close to unity. This observation indicates some evidence of non stationarity when we fit these types of models to the transactions data.

In order to examine the adequacy of the fit, we consider the Pearson residuals defined by $e_t = (Y_t - \lambda_t)/\sqrt{(\lambda_t + \lambda_t^2/\nu_2)}$. Under the correct model, the sequence e_t is a white noise sequence with constant variance, see Kedem and Fokianos [56, Sec. 1.6.3]. We estimate the Pearson residuals by substituting λ_t by $\lambda(\hat{\boldsymbol{\theta}})$ and ν_2 by $\hat{\nu}_2$. Figure 3.6 demonstrates that the predicted values defined by $\hat{Y}_t = \lambda(\hat{\boldsymbol{\theta}})$ approximate the observed process reasonably well. The right plots of the same figure depict the cumulative periodogram plots (cf. Brockwell and Davis [8, Sec. 10.2]) and illustrate the whiteness of the Pearson residuals

in both the linear and nonlinear model.

	QMLE			Estimators of ν	
	$\hat{d}$	$\hat{a}_1$	$\hat{b}_1$	$\hat{\nu}_1$	$\hat{\nu}_2$
Linear Model (3.6)	0.581	0.745	0.199	7.022	7.158
	(0.236)	(0.047)	(0.035)	(0.985)	(0.940)
	(0.149)	(0.030)	(0.022)		
Nonlinear Model (3.11)	1.327	0.774	0.186	7.127	7.229
	(0.508)	(0.041)	(0.034)	(0.929)	(0.915)
	(0.324)	(0.026)	(0.021)		

Table 3.7: Quasi maximum likelihood estimators and their standards errors (in parentheses) for the linear model (3.6) and the nonlinear model (3.11) with $\gamma = 0.5$, for the total number of transactions per minute for the stock Ericsson B for the time period between July 2nd and July 22nd, 2002. The total number of observations is 460.

3.5.2 Measles Data

The left plot of Figure 3.7 shows the monthly number of measles at Sheffield for the period between September 8th, 1978 and April 17th, 1987. The total number of observations is 450. The right plot of Figure 3.7 shows the autocorrelation function of those data. Apparently, the autocorrelation plot shows the high degree of correlation among observations and hence we anticipate that either linear or a nonlinear model would accommodate suitably those data. The sample mean and variance of this particular data collection are 17.151 and 265.781 respectively, and this fact shows that the data are overdispersed. The results of the estimation are summarized in Table 3.8. Together with the estimators of d, a_1 and b_1, we report standard errors where the first row of standard errors are under the negative binomial distribution and the second line corresponds to the case of Poisson distribution. Standard errors for $\hat{\nu}_i$, $i = 1, 2$ have been computed by employing parametric bootstrap.

In addition, Figure 3.8 shows the adequacy of the fit and the whiteness of the Pearson residuals.

3.5.3 Breech Births Data

The third example to be considered is a count time series reported by Zucchini and MacDonald [78]. These data correspond to the number of monthly breech births in Edendale hospital of Pietermaritzburg in South Africa from February 1977 to January 1986. The size

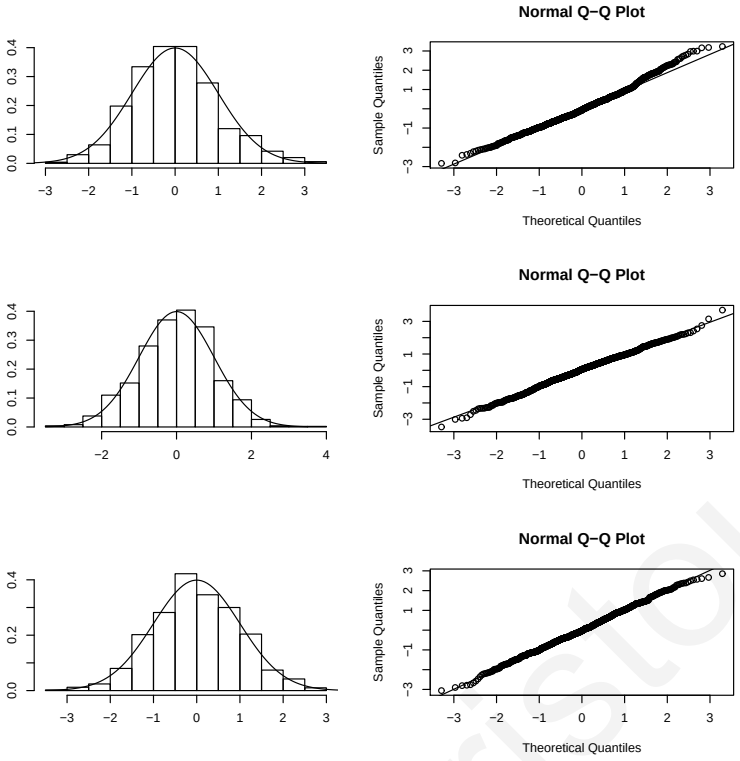

Figure 3.4: From top to bottom: Histograms and qq-plots of the sampling distribution of the standardized estimators of $\hat{\boldsymbol{\theta}} = (\hat{d}, \hat{a}_1, \hat{b}_1)$ for the linear model (3.6) when the true values are $(d, a_1, b_1) = (0.5, 0.3, 0.4)$ and $Z \sim U(0, 2)$. Superimposed is the standard normal density function. The results are based on 1000 data points and 1000 simulations.

	QMLE			Estimators of ν	
	$\hat{d}$	$\hat{a}_1$	$\hat{b}_1$	$\hat{\nu}_1$	$\hat{\nu}_2$
Linear Model (3.6)	0.720	0.490	0.469	4.853	5.309
	(0.235)	(0.057)	(0.055)	(0.617)	(0.650)
	(0.122)	(0.024)	(0.023)		
Nonlinear Model (3.11)	1.549	0.506	0.469	4.816	5.239
	(0.522)	(0.054)	(0.055)	(0.631)	(0.650)
	(0.292)	(0.023)	(0.022)		

Table 3.8: Quasi maximum likelihood estimators and their standards errors (in parentheses) for the linear model (3.6) and the nonlinear model (3.11) with $\gamma = 0.5$, for the monthly number of measles at Sheffield for the period between September 8th, 1978 and April 17th, 1987. The total number of observations is 450.

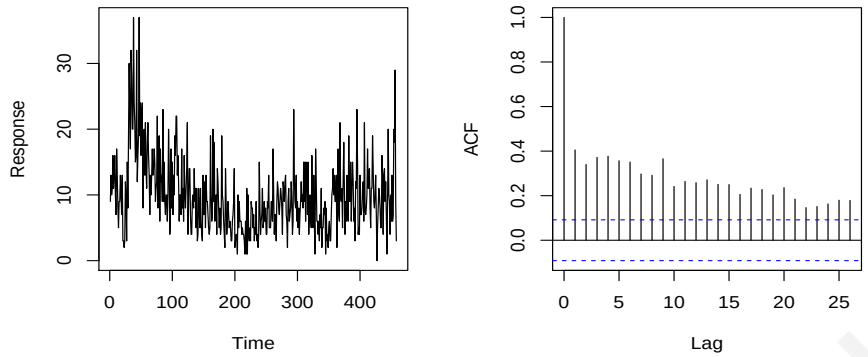

Figure 3.5: Left Plot: Number of transactions per minute for the stock Ericsson B for the time period between July 2nd and July 22nd, 2002. Right Plot: The autocorrelation function for the transactions data.

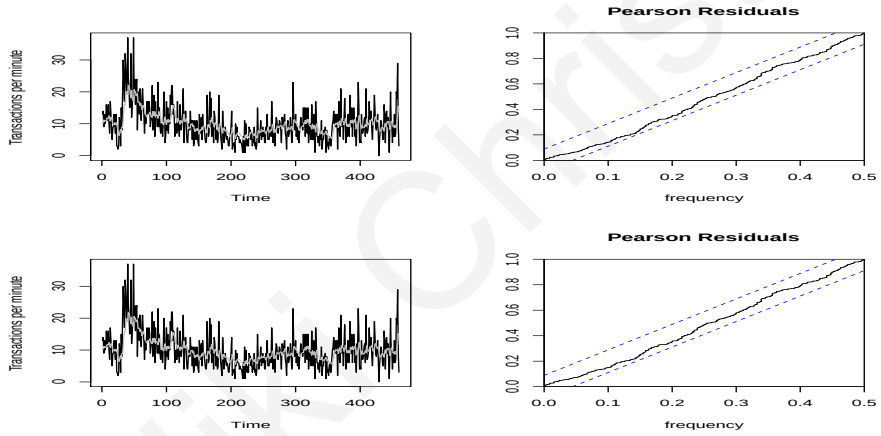

Figure 3.6: Left Plots: Observed and predicted (grey) number of transactions per minute for the stock Ericsson B for the time period between July 2nd and July 22nd, 2002, if we fit the linear model (3.6) and the nonlinear model (3.11), respectively. Right Plots: Cumulative periodogram plots of the Pearson residuals if we fit the linear model (3.6) and the nonlinear model (3.11), respectively, to the transactions data.

of this particular time series is $n = 108$. Figure 3.9 shows the data and the corresponding autocorrelation function. Here, we note that there is a reduced degree of autocorrelation among successive observations. Nevertheless, we can operate as in the previous examples and following the same methodology, we obtain Table 3.9. Note that the overdispersion phenomenon is occurred also in this collection, since the sample mean is 18.176 and the sample variance is equal to 62.240.

Furthermore, the adequacy of the fit and the whiteness of the Pearson residuals are demonstrated by Figure 3.10.

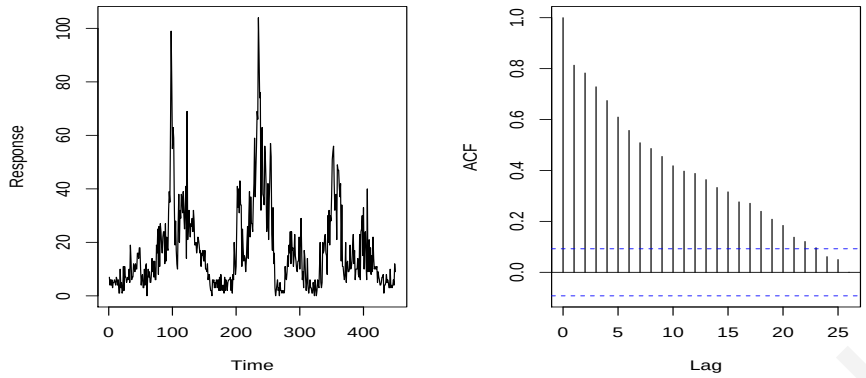

Figure 3.7: Left Plot: Number of measles in Sheffield for the time period between September 8th, 1978 and April 17th, 1987. Right Plot: The autocorrelation function for the measles data.

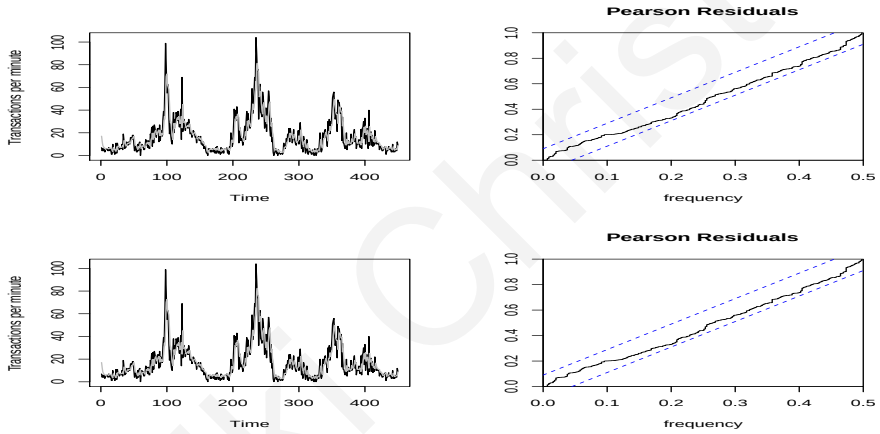

Figure 3.8: Left Plots: Observed and predicted (grey) number of measles in Sheffield for the time period between September 8th, 1978 and April 17th, 1987, if we fit the linear model (3.6) and the nonlinear model (3.11), respectively. Right Plots: Cumulative periodogram plots of the Pearson residuals if we fit the linear model (3.6) and the nonlinear model (3.11), respectively, to the measles data.

3.5.4 Homicides Data

Another data example that we consider, corresponds to the numbers of deaths recorded at the Salt River state mortuary at Cape Town, for the period time 1986 - 1991 (reported by Zucchini and MacDonald [78]). The length of the time series is 312. The left plot of Figure 3.11 shows the time series, while the right plot shows the autocorrelation function of the data. Operating as in the previous examples and following the same methodology, we obtain Table 3.10. Note that the overdispersion phenomenon is occurred also in this collection, since the sample mean is 2.628 and the sample variance is equal to 6.588.

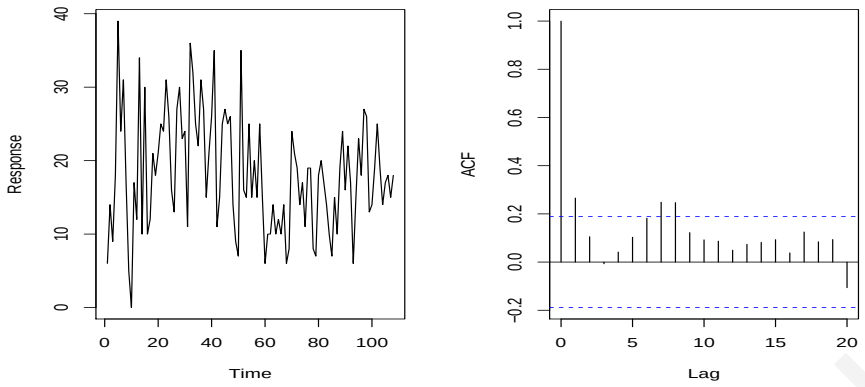

Figure 3.9: Left Plot: Number of monthly breech births in Edendale hospital of Pietermaritzburg in South Africa from February 1977 to January 1986. Right Plot: The autocorrelation function for the births data.

	QMLE			Estimators of ν	
	$\hat{d}$	$\hat{a}_1$	$\hat{b}_1$	$\hat{\nu}_1$	$\hat{\nu}_2$
Linear Model (3.6)	11.753	0.099	0.261	7.885	7.686
	(4.209)	(0.254)	(0.103)	(1.963)	(1.869)
	(2.406)	(0.144)	(0.055)		
Nonlinear Model (3.11)	12.703	0.662	0.181	8.031	7.797
	(5.131)	(0.119)	(0.084)	(2.027)	(1.801)
	(3.138)	(0.069)	(0.046)		

Table 3.9: Quasi maximum likelihood estimators and their standards errors (in parentheses) for the linear model (3.6) and the nonlinear model (3.11) if $\gamma = 0.5$, for the number of monthly breech births in Edendale hospital of Pietermaritzburg in South Africa from February 1977 to January 1986. The total number of observations is 108.

In addition, Figure 3.12 shows the adequacy of the fit and the whiteness of the Pearson residuals.

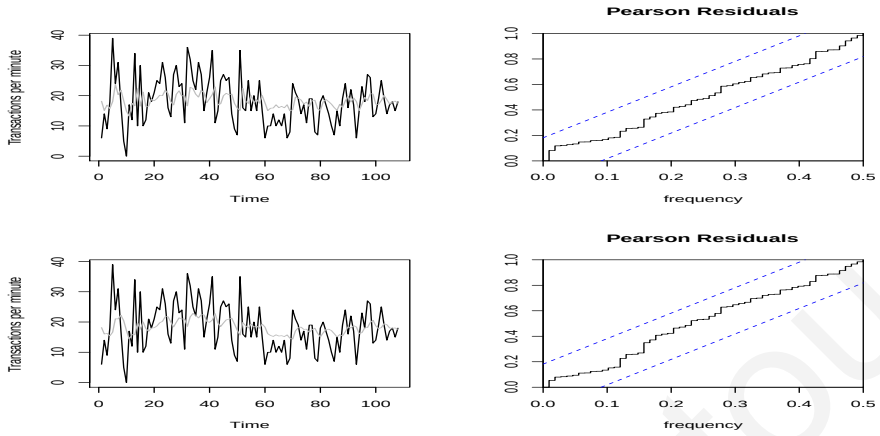

Figure 3.10: Left Plots: Observed and predicted (grey) number of monthly breech births in Edendale hospital of Pietermaritzburg in South Africa from February 1977 to January 1986, if we fit the linear model (3.6) and the nonlinear model (3.11), respectively. Right Plots: Cumulative periodogram plots of the Pearson residuals if we fit the linear model (3.6) and the nonlinear model (3.11), respectively, to the births data.

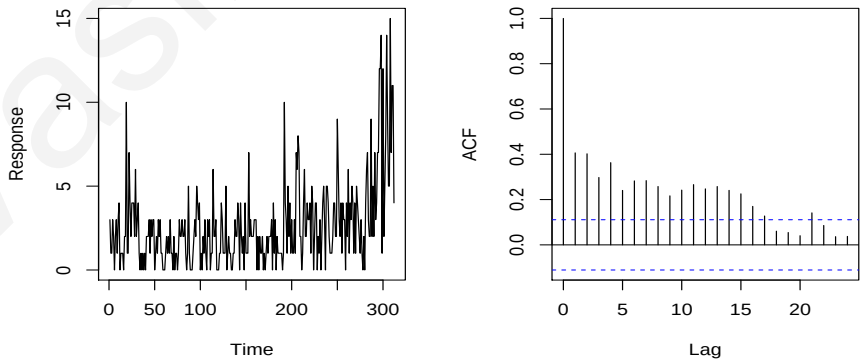

Figure 3.11: Left Plot: Number of deaths recorded at the Salt River state mortuary at Cape Town, for the period time 1986 - 1991. Right Plot: The autocorrelation function for the deaths data.

	QMLE			Estimators of ν	
	$\hat{d}$	$\hat{a}_1$	$\hat{b}_1$	$\hat{\nu}_1$	$\hat{\nu}_2$
Linear Model (3.6)	0.120	0.795	0.168	3.463	3.825
	(0.089)	(0.064)	(0.043)	(0.933)	(0.772)
	(0.065)	(0.048)	(0.032)		
Nonlinear Model (3.11)	0.090	0.853	0.137	3.499	3.853
	(0.091)	(0.046)	(0.036)	(0.908)	(0.762)
	(0.068)	(0.035)	(0.027)		

Table 3.10: Quasi maximum likelihood estimators and their standards errors (in parentheses) for the linear model (3.6) and the nonlinear model (3.11) if $\gamma = 0.5$, for the number of deaths recorded at the Salt River state mortuary at Cape Town, for the period time 1986 - 1991. The total number of observations is 312.

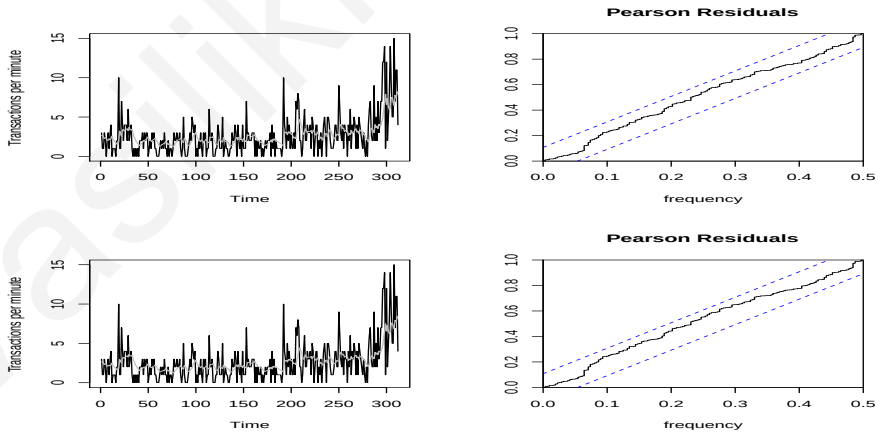

Figure 3.12: Left Plots: Observed and predicted (grey) number of deaths recorded at the Salt River state mortuary at Cape Town, for the period time 1986 - 1991, if we fit the linear model (3.6) and the nonlinear model (3.11), respectively. Right Plots: Cumulative periodogram plots of the Pearson residuals if we fit the linear model (3.6) and the nonlinear model (3.11), respectively, to the deaths data.

Appendix A – Proofs

Lemma A-1 Let Ψ be a negative binomial random variable with pmf given by (3.3) such that its mean is equal to λ. Let $k \in \mathbb{N}$. Then, the moments of order k of Ψ are given by

$$\mathrm{E}(\Psi^k) = \sum_{j=1}^{k} c_j(\nu)\lambda^j, \qquad \text{with} \;\; c_1(\nu) = 1. \tag{A-1}$$

Proof of Lemma A-1

Following Jain and Consul [53], the kth moment of a negative binomial random variable, is given by the following recurrence relation:

$$M_k(\nu) = \frac{\nu\lambda}{\nu+\lambda} \sum_{j=0}^{k-1} \binom{k-1}{j} \Big[M_j(\nu) + \frac{1}{\nu} M_{j+1}(\nu) \Big], \qquad \text{with} \;\; M_0(\nu) = 1, \tag{A-2}$$

where $M_k(\nu) = \mathrm{E}(\Psi^k)$. By induction, we will show that Lemma A-1 holds true.
For $k = 1$, obviously $\mathrm{E}(\Psi) = \lambda$. Suppose that it holds for k. We will show that it holds for $k+1$, i.e.

$$\mathrm{E}(\Psi^{k+1}) = \sum_{j=1}^{k+1} c_j(\nu)\lambda^j.$$

Using (A-2) we have that

$$\begin{aligned}
\mathrm{E}(\Psi^{k+1}) = M_{k+1}(\nu) &= \frac{\nu\lambda}{\nu+\lambda} \sum_{j=0}^{k} \binom{k}{j} \Big[M_j(\nu) + \frac{1}{\nu} M_{j+1}(\nu) \Big] \\
&= \frac{\nu\lambda}{\nu+\lambda} \Big[\sum_{j=0}^{k} \binom{k}{j} M_j(\nu) + \frac{1}{\nu} \sum_{j=0}^{k} \binom{k}{j} M_{j+1}(\nu) \Big] \\
&= \frac{\nu\lambda}{\nu+\lambda} \Big[\sum_{j=0}^{k} \binom{k}{j} M_j(\nu) + \frac{1}{\nu} \sum_{j=0}^{k-1} \binom{k}{j} M_{j+1}(\nu) + \frac{1}{\nu} M_{k+1}(\nu) \Big]
\end{aligned}$$

Hence,

$$\begin{aligned}
M_{k+1}(\nu) &= \lambda\Big[\sum_{j=0}^{k}\binom{k}{j}M_j(\nu)+\frac{1}{\nu}\sum_{l=1}^{k}\binom{k}{l-1}M_l(\nu)\Big] \\
&= \lambda\Big[1+\sum_{j=1}^{k}\binom{k}{j}M_j(\nu)+\frac{1}{\nu}\sum_{j=1}^{k}\binom{k}{j-1}M_j(\nu)\Big] \\
&= \lambda\Big[1+\sum_{j=1}^{k}\Big[\binom{k}{j}+\frac{1}{\nu}\binom{k}{j-1}\Big]M_j(\nu)\Big] \\
&= \lambda\Big[1+\sum_{j=1}^{k}\delta_j(\nu)\sum_{i=1}^{j}c_i(\nu)\lambda^i\Big] \\
&= \lambda\Big[1+\sum_{i=1}^{k}c_i^*(\nu)\lambda^i\Big] \\
&= \lambda+\sum_{i=1}^{k}c_i^*(\nu)\lambda^{i+1} \\
&= \sum_{i=1}^{k+1}c_i^{**}(\nu)\lambda^i, \quad \text{with} \;\; c_1^{**}(\nu)=1.
\end{aligned}$$

□

Proof of Theorem 3.2.1

First we show that if $\alpha<1$, then there exists a weakly dependent strictly stationary process $\{(Y_t,\lambda_t)\}$ which belongs to L^1. We verify condition 3.1 of Doukhan and Wintenberger [27]. The other two conditions of the same paper hold trivially.

Define $X_t=(Y_t,\lambda_t)$ and note that model (3.5) is expressed as

$$X_t=(\tilde{N}_t(0,Z_tf(Y_{t-1},\lambda_{t-1})],f(Y_{t-1},\lambda_{t-1}))=F(X_{t-1},\varepsilon_t)$$

where $\varepsilon_t=(\tilde{N}_t,Z_t)$ is an iid sequence. Then, for a vector $\boldsymbol{x}=(y,\lambda)\in\mathbb{N}_0\times(0,\infty)$, define

$\|\boldsymbol{x}\|_\epsilon = |y| + \epsilon|\lambda|$, for every $\epsilon > 0$. With $\boldsymbol{x'}$ defined analogously, we have that

$$\begin{aligned}
\mathrm{E}\|F(\boldsymbol{x}, \varepsilon_t) - F(\boldsymbol{x}', \varepsilon_t)\|_\epsilon &= \mathrm{E}|\tilde{N}_t(0, Z_t f(\boldsymbol{x})] - \tilde{N}_t(0, Z_t f(\boldsymbol{x}')]| + \epsilon|f(\boldsymbol{x}) - f(\boldsymbol{x}')| \\
&= \mathrm{E}\Big\{\mathrm{E}|\tilde{N}_t(0, Z_t f(\boldsymbol{x})] - \tilde{N}_t(0, Z_t f(\boldsymbol{x}')]|\Big| Z_t\Big\} + \epsilon|f(\boldsymbol{x}) - f(\boldsymbol{x}')| \\
&= \mathrm{E}(Z_t|f(\boldsymbol{x}) - f(\boldsymbol{x}')|) + \epsilon|f(\boldsymbol{x}) - f(\boldsymbol{x}')| \\
&= (1+\epsilon)|f(\boldsymbol{x}) - f(\boldsymbol{x}')| \\
&\leq (1+\epsilon)\max(\frac{\alpha_1}{\epsilon}, \alpha_2)\|\boldsymbol{x} - \boldsymbol{x}'\|_\epsilon.
\end{aligned}$$

The second equality follows from conditioning, the third equality holds because of the Poisson process properties and the fourth equality is true because the random variable Z_t is positive with mean 1. Choosing $\epsilon = \alpha_1/\alpha_2$, yields that $(1+\epsilon)\max(\alpha_1/\epsilon, \alpha_2) = \alpha_1 + \alpha_2 \equiv \alpha < 1$. Hence, we have shown that if $\alpha_1 + \alpha_2 < 1$, then there exists a weakly dependent strictly stationary process $\{(Y_t, \lambda_t)\}$ with finite first order moments.

Now, we will show that if $\alpha^r < 1$, then $\{(Y_t, \lambda_t)\}$ has moments of order r, $r \in \mathbb{N}$. Obviously if $\alpha < 1$, then $\alpha^r < 1$. Set $\|\boldsymbol{x}\|_r = (|y|^r + |\lambda|^r)^{1/r}$. Observe that

$$\begin{aligned}
|f(\boldsymbol{x})| = |f(\boldsymbol{x}) - f(\mathbf{0}) + f(\mathbf{0})| &\leq |f(\boldsymbol{x}) - f(\mathbf{0})| + |f(\mathbf{0})| \\
&\leq \alpha_1|\lambda| + \alpha_2|y| + |f(\mathbf{0})| \\
&= g(\boldsymbol{x}) + f(\mathbf{0}).
\end{aligned}$$

Therefore,

$$|f(\boldsymbol{x})|^r \leq g^r(\boldsymbol{x}) + \sum_{j=0}^{r-1} \binom{r}{j} g^j(\boldsymbol{x}) f^{r-j}(\mathbf{0}) = g^r(\boldsymbol{x}) + R(\boldsymbol{x}).$$

For $i = 1, \ldots, r$ we obtain

$$g^i(\boldsymbol{x}) = (\alpha_1|\lambda| + \alpha_2|y|)^i = \alpha^i\Big(\frac{\alpha_1}{\alpha}|\lambda| + \frac{\alpha_2}{\alpha}|y|\Big)^i.$$

Therefore, by Jensen's inequality we have that

$$\mathrm{E}(g^i(\boldsymbol{x})) = \alpha^i \mathrm{E}\Big(\frac{\alpha_1}{\alpha}|\lambda| + \frac{\alpha_2}{\alpha}|y|\Big)^i \leq \alpha^{i-1}(\alpha_1 \mathrm{E}|\lambda_t|^i + \alpha_2 \mathrm{E}|Y_t|^i) \leq \alpha^i \mathrm{E}\|X_0\|^i.$$

Hence,

$$\begin{aligned}
\mathrm{E}\|X_t\|^r = \mathrm{E}(Y_t^r + \lambda_t^r) &= \mathrm{E}\{\mathrm{E}\{Y_t^r + \lambda_t^r | \mathcal{F}_{t-1}^{Y,\lambda}\}\} \\
&= \mathrm{E}\{\sum_{j=1}^{r} c_j(\nu)\lambda_t^j + \lambda_t^r\} \\
&= (c_r(\nu)\mathrm{E}(\lambda_t^r) + \mathrm{E}(\lambda_t^r)) + \sum_{j=1}^{r-1} c_j(\nu)\mathrm{E}(\lambda_t^j) \\
&= (c_r(\nu)+1)\mathrm{E}(\lambda_t^r) + \sum_{j=1}^{r-1} c_j(\nu)\mathrm{E}(\lambda_t^j).
\end{aligned}$$

Using induction and the proof of Proposition 6 of Ferland et al. [31], we have that $\mathrm{E}(\lambda_t^r) < \infty$. Therefore, $\mathrm{E}\|X_t\|^r < \infty$ for $r \in \mathbb{N}$.

Finally, by Doukhan and Wintenberger [27], there exists a measurable function ϕ such that $X_t = \phi(\varepsilon_t, \varepsilon_{t-1}, \ldots)$ for all $t \in \mathbb{Z}$. The ergodicity follows from the Proposition 4.3 in Krengel [59], which states that if $(E, \mathcal{E})$ and $(\tilde{E}, \tilde{\mathcal{E}})$ are measurable spaces, $\{v_t, t \in \mathbb{Z}\}$ is a stationary ergodic sequence of $E-$valued random elements and $\phi : (E^{\mathbb{N}}, \mathcal{E}^{\mathbb{N}}) \mapsto (\tilde{E}, \tilde{\mathcal{E}})$ is a measurable function, then the sequence $\{\tilde{v}_t, t \in \mathbb{Z}\}$ defined by $\tilde{v}_t = \phi(v_t, v_{t-1}, \ldots)$ is a stationary, ergodic process. □

Proof of Proposition 3.2.1

Let the two polynomials $A(\mathcal{B}) = 1 - a_1\mathcal{B} - \ldots - a_q\mathcal{B}$ and $B(\mathcal{B}) = b_1\mathcal{B} + \ldots + b_p\mathcal{B}$, where $\mathcal{B}$ is the backshift operator. Suppose that the roots of $A(z) = 0$ lie outside the unit circle. Then $\sum_{i=1}^{q} a_i < 1$. Indeed, suppose that $\sum_{i=1}^{q} a_i \geq 1$. Then $A(0) = 1$ and $A(1) = 1 - a_1 - \ldots - a_q \leq 0$. Hence, there exists a root between $(0, 1]$, which contradicts the hypothesis.

Under the assumption of $\sum_{i=1}^{q} a_i < 1$, the operator $A(\mathcal{B})$ has an inverse, denoted by $A^{-1}(\mathcal{B})$, and then we have that $\lambda_t - \sum_{i=1}^{q} a_i\lambda_{t-i} = d + \sum_{j=1}^{p} b_j Y_{t-j}$ is equivalent to $A(\mathcal{B})\lambda_t = d + B(\mathcal{B})Y_t$. Therefore,

$$\begin{aligned}
\lambda_t &= A^{-1}(\mathcal{B})(d + B(\mathcal{B})Y_t) \\
&= dA^{-1}(1) + A^{-1}(\mathcal{B})B(\mathcal{B})Y_t \\
&= dA^{-1}(1) + H(\mathcal{B})Y_t,
\end{aligned}$$

where $H(z) = A^{-1}(z)B(z) = \sum_{j=1}^{\infty} \psi_j z^j$.

Then, we have that

$$\begin{aligned} \mu &= \mathrm{E}(Y_t) = \mathrm{E}[\mathrm{E}(Y_t|\mathcal{F}_{t-1}^{Y,\lambda})] \\ &= \mathrm{E}(dA^{-1}(1) + \sum_{j=1}^{\infty} \psi_j Y_{t-j}) \\ &= \frac{d}{A(1)} + \mu \frac{B(1)}{A(1)}. \end{aligned}$$

Hence, $\mu = d/(A(1) - B(1)) = d/(1 - \sum_{i=1}^{q} a_i - \sum_{j=1}^{p} b_j)$ and therefore, the parameters $a_i,\ i = 1, \ldots, q$ and $b_j,\ j = 1, \ldots, p$ of the nonnegative integer-valued process $\{Y_t\}$ must satisfy necessarily the condition

$$\sum_{i=1}^{q} a_i + \sum_{j=1}^{p} b_j < 1.$$

□

Assumptions for Theorem 3.3.1 The following list of assumptions are used in proving Theorem 3.3.1 and Lemma 3.3.1:

Assumption A-1 The parameter $\boldsymbol{\theta}$ belongs to a compact set Θ and $\boldsymbol{\theta_0}$ belongs to the interior of Θ. In addition, the function $f(\cdot)$ satisfies the condition

$$f(y, \lambda, \boldsymbol{\theta}) \geq C_1 > 0,$$

for some constant C_1 and for all $\lambda > 0$ and $y \in \mathbb{N}_0$.

Assumption A-2 The components of $\partial f / \partial \boldsymbol{\theta}$ are linearly independent.

Assumption A-3 The function $f(\cdot)$ is four times differentiable with respect to $\boldsymbol{\theta}$ and λ.

In addition if $x^* = (\boldsymbol{\theta}, \lambda) = (\theta_1, \ldots, \theta_m, \lambda)$, then

$$\begin{aligned}
\Big|\frac{\partial f(y,\lambda,\boldsymbol{\theta})}{\partial x_i^*} - \frac{\partial f(y',\lambda',\boldsymbol{\theta})}{\partial x_i^*}\Big| &\leq \alpha_{1i}|\lambda-\lambda'| + \alpha_{2i}|y-y'\Big|, \quad i = 1,\ldots,m+1. \\
\Big|\frac{\partial^2 f(y,\lambda,\boldsymbol{\theta})}{\partial x_i^* \partial x_j^*} - \frac{\partial^2 f(y',\lambda',\boldsymbol{\theta})}{\partial x_i^* \partial x_j^*}\Big| &\leq \alpha_{1ij}|\lambda-\lambda'| + \alpha_{2ij}|y-y'|, \quad i,j = 1,\ldots,m+1. \\
\Big|\frac{\partial^3 f(y,\lambda,\boldsymbol{\theta})}{\partial x_i^* \partial x_j^* \partial x_k^*} - \frac{\partial^3 f(y',\lambda',\boldsymbol{\theta})}{\partial x_i^* \partial x_j^* \partial x_k^*}\Big| &\leq \alpha_{1ijk}|\lambda-\lambda'| + \alpha_{2ijk}|y-y'|, \quad i,j,k = 1,\ldots,m+1.
\end{aligned}$$

Furthermore $\mathrm{E}|\partial f(0,0,\boldsymbol{\theta})/\partial x_i^*| < \infty$, $\mathrm{E}|\partial^2 f(0,0,\boldsymbol{\theta})/\partial x_i^* \partial x_j^*| < \infty$ and $\mathrm{E}|\partial^3 f(0,0,\boldsymbol{\theta})/\partial x_i^* \partial x_j^* \partial x_k^*| < \infty$ for all i,j,k in $\{1,2,\ldots,m+1\}$ and $\sum\limits_i(\alpha_{1i}+\alpha_{2i})$, $\sum\limits_{i,j}(\alpha_{1ij}+\alpha_{2ij})$, $\sum\limits_{i,j,k}(\alpha_{1ijk}+\alpha_{2ijk}) < \infty$.

Before proceeding to the proof of Lemma (3.3.1) we first show the validity of Remark 3.3.2.

Lemma A-2 Under the conditions of Theorem 3.2.1 and assumptions A-1 and A-2

$$\sup_{\boldsymbol{\theta}\in\Theta}\Big|\frac{1}{n}l_n(\boldsymbol{\theta}) - \frac{1}{n}\tilde{l}_n(\boldsymbol{\theta})\Big| \xrightarrow{a.s.} 0 \quad \text{as } n \to \infty$$

following the notation of Remark 3.3.2.

Proof of Lemma A-2

Let $\lambda_t = f(Y_{t-1}, \lambda_{t-1})$ denote the stationary and ergodic solution of (3.5). Let $\tilde{\lambda}_t = f(Y_{t-1}, \tilde{\lambda}_{t-1})$ be the process obtained with some starting value $\tilde{\lambda}_0$. Then,

$$\begin{aligned}
|\lambda_t(\boldsymbol{\theta}) - \tilde{\lambda}_t(\boldsymbol{\theta})| &= |f(Y_{t-1}, \lambda_{t-1}(\boldsymbol{\theta})) - f(Y_{t-1}, \tilde{\lambda}_{t-1}(\boldsymbol{\theta}))| \\
&\leq \alpha_1|\lambda_{t-1}(\boldsymbol{\theta}) - \tilde{\lambda}_{t-1}(\boldsymbol{\theta})| \\
&\leq \alpha_1^2|\lambda_{t-2}(\boldsymbol{\theta}) - \tilde{\lambda}_{t-2}(\boldsymbol{\theta})| \\
&\leq \ldots \\
&\leq \alpha_1^t|\lambda_0(\boldsymbol{\theta}) - \tilde{\lambda}_0(\boldsymbol{\theta})|,
\end{aligned}$$

by using the contraction assumption of Theorem 3.2.1.

Therefore,

$$\sup_{\boldsymbol{\theta}\in\Theta}|\lambda_t(\boldsymbol{\theta}) - \tilde{\lambda}_t(\boldsymbol{\theta})| \leq K\alpha_1^t, \quad \forall\ t,$$

almost surely from the compactness of Θ, for some positive constant K.

Now, for any $x, y > 0$ it holds that $|\log(x/y)| \leq |x-y|/\min(x,y)$. We obtain that

$$\begin{aligned} |l_n(\boldsymbol{\theta}) - \tilde{l}_n(\boldsymbol{\theta})| &\leq \frac{1}{n}\Big\{\sum_{t=1}^{n} Y_t |\log\lambda_t(\boldsymbol{\theta}) - \log\tilde{\lambda}_t(\boldsymbol{\theta})| + \sum_{t=1}^{n}|\lambda_t(\boldsymbol{\theta}) - \tilde{\lambda}_t(\boldsymbol{\theta})|\Big\} \\ &\leq \frac{1}{n}\Big\{\sum_{t=1}^{n}\frac{Y_t|\lambda_t(\boldsymbol{\theta}) - \tilde{\lambda}_t(\boldsymbol{\theta})|}{\min(\lambda_t(\boldsymbol{\theta}), \tilde{\lambda}_t(\boldsymbol{\theta}))} + \sum_{t=1}^{n}|\lambda_t(\boldsymbol{\theta}) - \tilde{\lambda}_t(\boldsymbol{\theta})|\Big\} \\ &\leq \frac{1}{n}\frac{1}{C_1}\sum_{t=1}^{n} Y_t|\lambda_t(\boldsymbol{\theta}) - \tilde{\lambda}_t(\boldsymbol{\theta})| + \frac{1}{n}\sum_{t=1}^{n}|\lambda_t(\boldsymbol{\theta}) - \tilde{\lambda}_t(\boldsymbol{\theta})|, \end{aligned}$$

by using Assumption A-1.

Hence,

$$\begin{aligned} \sup_{\boldsymbol{\theta}\in\Theta}|l_n(\boldsymbol{\theta}) - \tilde{l}_n(\boldsymbol{\theta})| &\leq \frac{1}{n}\sum_{t=1}^{n}\Big(\frac{Y_t}{C_1}+1\Big)|\lambda_t(\boldsymbol{\theta}) - \tilde{\lambda}_t(\boldsymbol{\theta})| \\ &\leq \frac{K}{n}\sum_{t=1}^{n}\Big(\frac{Y_t}{C_1}+1\Big)\alpha_1^t. \end{aligned}$$

But using Markov′s inequality for every $\epsilon > 0$ we have that

$$\sum_{t=1}^{\infty} P\Big(\alpha_1^t\Big(\frac{Y_t}{C_1}+1\Big) > \epsilon\Big) \leq \sum_{t=1}^{\infty}\frac{\alpha_1^{st}\mathrm{E}\Big(Y_t/C_1+1\Big)^s}{\epsilon^s} < \infty.$$

If we take $b_n = n$ and $x_t = \alpha_1^t(Y_t/C_1+1)$ in Cesaro′s lemma (see Lemma 2.1.1), then we have that $\alpha_1^t(Y_t/C_1+1) \xrightarrow{a.s.} 0$, as $t\to\infty$ and therefore the proof is concluded. □

Proof of Lemma 3.3.1

Assume that model (3.5) holds true with $\lambda_t(\boldsymbol{\theta}) = f(Y_{t-1}, \lambda_{t-1}, \boldsymbol{\theta})$ and denote by $m = \dim(\boldsymbol{\theta})$.

(i) Recall that the score function is given by (3.14). Denote by $W_t(\boldsymbol{\theta}) = (Y_t/\lambda_t(\boldsymbol{\theta}) - 1)$. Then $\partial l_t(\boldsymbol{\theta})/\partial\boldsymbol{\theta} = W_t(\boldsymbol{\theta})\partial\lambda_t(\boldsymbol{\theta})/\partial\boldsymbol{\theta}$. At the true value $\boldsymbol{\theta} = \boldsymbol{\theta_0}$, we obtain that $\mathrm{E}(W_t|\mathcal{F}_{t-1}^{Y,\lambda}) = 0$ and $\mathrm{Var}(W_t|\mathcal{F}_{t-1}^{Y,\lambda}) = 1/\lambda_t + \sigma_Z^2 < \infty$, by assumption A-1.

Now, we need to show that $\mathrm{E}|\partial\lambda_t(\boldsymbol{\theta})/\partial\boldsymbol{\theta}| < \infty$. But

$$\frac{\partial\lambda_t(\boldsymbol{\theta})}{\partial\theta_i} = \frac{\partial f(Y_{t-1}, \lambda_{t-1}(\boldsymbol{\theta}), \boldsymbol{\theta})}{\partial\lambda_{t-1}(\boldsymbol{\theta})}\frac{\partial\lambda_{t-1}(\boldsymbol{\theta})}{\partial\theta_i} + \frac{\partial f(Y_{t-1}, \lambda_{t-1}(\boldsymbol{\theta}), \boldsymbol{\theta})}{\partial\theta_i}, \quad i = 1, \ldots, m.$$

Therefore, we can write with obvious notation

$$A_t = B_{t-1}A_{t-1} + C_{t-1}, \quad t \geq 1$$

and by repeated substitutions, we obtain that

$$A_t = \Big\{ \prod_{i=0}^{t-1} B_{t-(i+1)} \Big\} A_0 + \sum_{k=1}^{t-1} \Big\{ \prod_{j=1}^{k} B_{t-j} \Big\} C_{t-(k+1)} + C_{t-1}.$$

Using the fact that $|f(y,\lambda) - f(y,\lambda')| \leq \alpha_1 |\lambda - \lambda'|$ (by Theorem 3.2.1) we obtain that

$$\Big| \frac{\partial \lambda_t(\boldsymbol{\theta})}{\partial \theta_i} \Big| \leq \alpha_1^{t-1} \Big| \frac{\partial \lambda_0(\boldsymbol{\theta})}{\partial \theta_i} \Big| + \sum_{k=0}^{t-1} \alpha_1^k \Big| \frac{\partial f(\lambda_{t-(k+1)}(\boldsymbol{\theta}), Y_{t-(k+1)}, \boldsymbol{\theta})}{\partial \theta_i} \Big| + C,$$

with C being a constant. Hence, by assumption A-3, we obtain that

$$\begin{aligned} \Big| \frac{\partial \lambda_t(\boldsymbol{\theta})}{\partial \theta_i} \Big| &\leq \alpha_1^{t-1} \Big| \frac{\partial \lambda_0(\boldsymbol{\theta})}{\partial \theta_i} \Big| + \sum_{k=0}^{t-1} \alpha_1^k \{ \alpha_{1i} \lambda_{t-(k+1)} + \alpha_{2i} Y_{t-(k+1)} \} + C \\ &= \alpha_1^{t-1} \Big| \frac{\partial \lambda_0(\boldsymbol{\theta})}{\partial \theta_i} \Big| + \alpha_{1i} \sum_{k=0}^{t-1} \alpha_1^k \lambda_{t-(k+1)} + \alpha_{2i} \sum_{k=0}^{t-1} \alpha_1^k Y_{t-(k+1)} + C. \end{aligned}$$

Theorem 3.2.1 guarantees that $\mathrm{E}|\partial \lambda_t(\boldsymbol{\theta})/\partial \boldsymbol{\theta}| < \infty$. Therefore, an application of central limit theorem for martingales (Hall and Heyde [48, Cor. 3.1]) shows that

$$\frac{1}{\sqrt{n}} \boldsymbol{S}_n \xrightarrow{D} \mathcal{N}(0, \boldsymbol{G}_1),$$

as $n \to \infty$. The matrix $\boldsymbol{G}_1$ is the limit of

$$\frac{1}{n} \sum_{t=1}^{n} \mathrm{E} \Big\{ W_t^2 \left(\frac{\partial \lambda_t}{\partial \boldsymbol{\theta}} \right) \left(\frac{\partial \lambda_t}{\partial \boldsymbol{\theta}} \right)' \Big| \mathcal{F}_{t-1}^{Y,\lambda} \Big\}$$

which is equal to (3.17). The conditional Lindeberg's condition holds by noting

$$\frac{1}{n} \sum_{t=1}^{n} \mathrm{E} \left(\|\partial l_t / \partial \boldsymbol{\theta}\|^2 \, I \left(\|\partial l_t / \partial \boldsymbol{\theta}\| > \sqrt{n}\delta \right) | \mathcal{F}_{t-1}^{Y,\lambda} \right) \leq \frac{1}{n^2 \delta^2} \sum_{t=1}^{n} \mathrm{E} \left(\|\partial l_t / \partial \boldsymbol{\theta}\|^4 | \mathcal{F}_{t-1}^{Y,\lambda} \right) \to 0,$$

since $\mathrm{E}\,\|\partial l_t/\partial\boldsymbol{\theta}\|^4 < \infty$, because of Theorem 3.2.1 which guarantees existence of moments.

(ii) We consider the following matrices:

$$\mathbf{A}_1 = \frac{1}{n}\sum_{t=1}^{n}\frac{Y_t}{\lambda_t^2}\left(\frac{\partial\lambda_t}{\partial\boldsymbol{\theta}}\right)\left(\frac{\partial\lambda_t}{\partial\boldsymbol{\theta}}\right)'$$

and

$$\mathbf{A}_2 = \frac{1}{n}\sum_{t=1}^{n}\Big(\frac{Y_t}{\lambda_t}-1\Big)\frac{\partial^2\lambda_t}{\partial\boldsymbol{\theta}\partial\boldsymbol{\theta}'}$$

For the first matrix, we obtain that

$$\mathbf{A}_1 \xrightarrow{p} \mathrm{E}\Big\{\frac{Y_t}{\lambda_t^2}\left(\frac{\partial\lambda_t}{\partial\boldsymbol{\theta}}\right)\left(\frac{\partial\lambda_t}{\partial\boldsymbol{\theta}}\right)'\Big\} = \mathbf{G}$$

after taking iterated expectations (the matrix $\mathbf{G}$ has been defined by (3.16)). The matrix $\mathbf{G}$ is positive definite because of assumption A-2. It is enough to show that $\mathbf{A}_2$ converges in probability to zero. But this is true, provided that $\mathrm{E}|\partial^2\lambda_t(\boldsymbol{\theta})/\partial\theta_i\partial\theta_j| < \infty$.

However,

$$\frac{\partial^2\lambda_t}{\partial\theta_i\partial\theta_j} = \frac{\partial\lambda_{t-1}}{\partial\theta_i}\frac{\partial\lambda_{t-1}}{\partial\theta_j}\frac{\partial^2 f}{\partial\lambda_{t-1}^2}+\frac{\partial\lambda_{t-1}}{\partial\theta_i}\frac{\partial^2 f}{\partial\lambda_{t-1}\partial\theta_j}+\frac{\partial f}{\partial\lambda_{t-1}}\frac{\partial^2\lambda_{t-1}}{\partial\theta_i\partial\theta_j}+\frac{\partial^2 f}{\partial\lambda_{t-1}\partial\theta_i}\frac{\partial\lambda_{t-1}}{\partial\theta_j}+\frac{\partial^2 f}{\partial\theta_i\partial\theta_j}.$$

Assumption A-3 guarantees that all these terms are bounded by linear function of both $\{\lambda_t\}$ and $\{Y_t\}$ and therefore we obtain the required result.

(iii) The verification of these results is based upon observing that is enough to show that $\mathrm{E}|\partial^3\lambda_t(\boldsymbol{\theta})/\partial\theta_i\partial\theta_j\partial\theta_k| < \infty$. Tedious algebra shows that

$$\frac{\partial^3\lambda_t(\boldsymbol{\theta})}{\partial\theta_i\partial\theta_j\partial_k} = I + II + III + IV + V, \tag{A-3}$$

where

$$
\begin{aligned}
I &= \frac{\partial^2\lambda_{t-1}}{\partial\theta_i\partial\theta_k}\frac{\partial\lambda_{t-1}}{\partial\theta_j}\frac{\partial^2 f}{\partial\lambda_{t-1}^2} + \frac{\partial\lambda_{t-1}}{\partial\theta_i}\frac{\partial^2\lambda_{t-1}}{\partial\theta_j\partial\theta_k}\frac{\partial^2 f}{\partial\lambda_{t-1}^2} + \frac{\partial\lambda_{t-1}}{\partial\theta_i}\frac{\partial\lambda_{t-1}}{\partial\theta_j}\frac{\partial\lambda_{t-1}}{\partial\theta_k}\frac{\partial^3 f}{\partial\lambda_{t-1}^3} \\
&+ \frac{\partial\lambda_{t-1}}{\partial\theta_i}\frac{\partial\lambda_{t-1}}{\partial\theta_j}\frac{\partial^3 f}{\partial\lambda_{t-1}^2\partial\theta_k}, \\
II &= \frac{\partial^2\lambda_{t-1}}{\partial\theta_i\partial\theta_k}\frac{\partial^2 f}{\partial\lambda_{t-1}\partial\theta_j} + \frac{\partial\lambda_{t-1}}{\partial\theta_i}\frac{\partial\lambda_{t-1}}{\partial\theta_k}\frac{\partial^3 f}{\partial\lambda_{t-1}^2\partial\theta_j} + \frac{\partial\lambda_{t-1}}{\partial\theta_i}\frac{\partial^3 f}{\partial\lambda_{t-1}\partial\theta_j\partial\theta_k}, \\
III &= \frac{\partial^2 f}{\partial\lambda_{t-1}^2}\frac{\partial^2\lambda_{t-1}}{\partial\theta_i\partial\theta_j}\frac{\partial\lambda_{t-1}}{\partial\theta_k} + \frac{\partial^2 f}{\partial\lambda_{t-1}\partial\theta_k}\frac{\partial^2\lambda_{t-1}}{\partial\theta_i\partial\theta_j} + \frac{\partial f}{\partial\lambda_{t-1}}\frac{\partial^3\lambda_{t-1}}{\partial\theta_i\partial\theta_j\partial\theta_k}, \\
IV &= \frac{\partial\lambda_{t-1}}{\partial\theta_j}\frac{\partial\lambda_{t-1}}{\partial\theta_k}\frac{\partial^3 f}{\partial\lambda_{t-1}^2\partial\theta_i} + \frac{\partial\lambda_{t-1}}{\partial\theta_j}\frac{\partial^3 f}{\partial\lambda_{t-1}\partial\theta_i\partial\theta_k} + \frac{\partial^2 f}{\partial\lambda_{t-1}\partial\theta_i}\frac{\partial^2\lambda_{t-1}}{\partial\theta_j\partial\theta_k}, \\
V &= \frac{\partial^3 f}{\partial\lambda_{t-1}\partial\theta_i\partial\theta_j}\frac{\partial\lambda_{t-1}}{\partial\theta_k} + \frac{\partial^3 f}{\partial\theta_i\partial\theta_j\partial\theta_k}.
\end{aligned}
$$

Consider the first summand of I in (A-3). Then, from Assumption A-3 we have that

$$
\begin{aligned}
\Big|\frac{\partial^2\lambda_{t-1}(y,\lambda,\boldsymbol{\theta})}{\partial\theta_i\partial\theta_k}\Big| &\leq \Big|\frac{\partial^2\lambda_{t-1}(0,0,\boldsymbol{\theta})}{\partial\theta_i\partial\theta_k}\Big| + \alpha_{1ik}|\lambda| + \alpha_{2ik}|y|, &\text{(A-4)}\\
\Big|\frac{\partial\lambda_{t-1}(y,\lambda,\boldsymbol{\theta})}{\partial\theta_j}\Big| &\leq \Big|\frac{\partial\lambda_{t-1}(0,0,\boldsymbol{\theta})}{\partial\theta_j}\Big| + \alpha_{1j}|\lambda| + \alpha_{2j}|y|, &\text{(A-5)}\\
\Big|\frac{\partial^2 f(y,\lambda,\boldsymbol{\theta})}{\partial\lambda_{t-1}^2}\Big| &\leq \Big|\frac{\partial^2 f(0,0,\boldsymbol{\theta})}{\partial\lambda_{t-1}^2}\Big| + \alpha_{1(m+1)(m+1)}|\lambda| + \alpha_{2(m+1)(m+1)}|y|. &\text{(A-6)}
\end{aligned}
$$

From (A-4), (A-5) and (A-6), we obtain that

$$
\begin{aligned}
\Big|\frac{\partial^2\lambda_{t-1}}{\partial\theta_i\partial\theta_k}\frac{\partial\lambda_{t-1}}{\partial\theta_j}\frac{\partial^2 f}{\partial\lambda_{t-1}^2}\Big| &\leq C_1^* + C_2^*|\lambda| + C_3^*|y| \\
&+ C_4^*|\lambda||y| + C_5^*|\lambda|^2 * C_6^*|y|^2 \\
&+ C_7^*|\lambda|^2|y| + C_8^*|\lambda||y|^2 + C_9^*|\lambda|^3 + C_{10}^*|y|^3,
\end{aligned}
$$

where $C_1^*, C_2^*, C_3^*, C_4^*, C_5^*, C_6^*, C_7^*, C_8^*, C_9^*, C_{10}^*$ are nonnegative real numbers.

Therefore, from Theorem 3.2.1 we have that

$$
\mathrm{E}\Big|\frac{\partial^2\lambda_{t-1}}{\partial\theta_i\partial\theta_k}\frac{\partial\lambda_{t-1}}{\partial\theta_j}\frac{\partial^2 f}{\partial\lambda_{t-1}^2}\Big| < \infty.
$$

Similarly, it can be shown that all the remaining summands of all the terms in (A-3),

have finite moments. □

Appendix B – R codes

R code regarding the linear model (3.6) for the construction of Tables 3.1, 3.2, top panels of Tables 3.7, 3.8, 3.9, 3.10, Figure 3.1, left plot of Figure 3.3, Figures 3.5, 3.7, 3.9, 3.11 and top plots of Figures 3.6, 3.8, 3.10 and 3.12.

```
#########################################################################
## Simulate the linear model using the negative binomial distribution ##
#########################################################################
linearnegbin.ts=function(d,a,b,size,nu)
{
y=rep(NA,size)
mu=rep(NA,size)
mu[1]=1 #initial value
y[1]=rnbinom(1,size=nu,mu=mu[1])
for (t in 2:size)
{
mu[t]=d+a*mu[t-1]+b*y[t-1]
y[t]=rnbinom(1,size=nu,mu=mu[t])
}
return(cbind(y,mu))
}


#####################################################
## (Poisson) Quasi-likelihood for the linear model ##
#####################################################
liklinear.poisson=function(theta,data)
{
lambda=rep(NA,length(data))
loglik=rep(NA,length(data))
lambda[1]=0
loglik[1]=0
for (t in 2:length(data))
```

```
{
lambda[t]=theta[1]+theta[2]*lambda[t-1]+theta[3]*data[t-1]
if (lambda[t]==0) loglik[t]= 0 else
if (lambda[t] >0) loglik[t]= -data[t]*log(lambda[t])+lambda[t]
}
final=sum(loglik)
}

#####################################################
## (Poisson) Score function for the linear model ##
#####################################################
scorelinear.poisson=function(theta,data)
{
lambda=rep(NA,length(data))
lambda[1]=0
first=rep(NA,length(data))
first[1]=0
second=rep(NA,length(data))
second[1]=0
third=rep(NA,length(data))
third[1]=0
s1=rep(NA,length(data))
s2=rep(NA,length(data))
s3=rep(NA,length(data))
for (t in 2:length(data))
{
lambda[t]=theta[1]+theta[2]*lambda[t-1]+theta[3]*data[t-1]
first[t]=(1+theta[2]*first[t-1])
second[t]=(lambda[t-1]+theta[2]*second[t-1])
third[t]=(data[t-1]+theta[2]*third[t-1])
s1[t]=-((data[t]/lambda[t])-1)*first[t]
```

```
s2[t]=-((data[t]/lambda[t])-1)*second[t]
s3[t]=-((data[t]/lambda[t])-1)*third[t]
}
ss1=sum(s1[-1])
ss2=sum(s2[-1])
ss3=sum(s3[-1])
score=c(ss1,ss2,ss3)
}

#####################################################################
## Information matrix based on the Poisson distribution-G matrix ##
#####################################################################
information.poisson=function(theta,data)
{
lambda=rep(NA,length(data))
lambda[1]=1
first=rep(NA,length(data))
first[1]=1
second=rep(NA,length(data))
second[1]=1
third=rep(NA,length(data))
third[1]=1
Information=matrix(0,nrow=3,ncol=3)
s1=rep(NA,length(data))
s2=rep(NA,length(data))
s3=rep(NA,length(data))
for (t in 2:length(data))
{
lambda[t]=theta[1]+theta[2]*lambda[t-1]+theta[3]*data[t-1]
first[t]=(1+theta[2]*first[t-1])
second[t]=(lambda[t-1]+theta[2]*second[t-1])
```

```
third[t]=(data[t-1]+theta[2]*third[t-1])
s1[t]=first[t]
s2[t]=second[t]
s3[t]=third[t]
var.comp=(1/sqrt(lambda[t]))*c(s1[t],s2[t],s3[t])
Information=Information+var.comp%*%t(var.comp)
}
return(Information)
}

#########################################################################
## Information matrix based on the negative binomial distribution ##
## G_{1} matrix                                                        ##
#########################################################################
information.negbin=function(theta,data,nu)
{
lambda=rep(NA,length(data))
lambda[1]=1
first=rep(NA,length(data))
first[1]=1
second=rep(NA,length(data))
second[1]=1
third=rep(NA,length(data))
third[1]=1
Information=matrix(0,nrow=3,ncol=3)
s1=rep(NA,length(data))
s2=rep(NA,length(data))
s3=rep(NA,length(data))
for (t in 2:length(data))
{
lambda[t]=theta[1]+theta[2]*lambda[t-1]+theta[3]*data[t-1]
```

```
first[t]=(1+theta[2]*first[t-1])
second[t]=(lambda[t-1]+theta[2]*second[t-1])
third[t]=(data[t-1]+theta[2]*third[t-1])
s1[t]=first[t]
s2[t]=second[t]
s3[t]=third[t]
var.comp=(sqrt(1/lambda[t]+1/nu))*c(s1[t],s2[t],s3[t])
Information=Information+var.comp%*%t(var.comp)
}
return(Information)
}

###################################
## Constrains to obtain the QMLE ##
## d>0, a>0, b>0, 0<a+b<1        ##
###################################
uilinear=matrix(0,nrow=4,ncol=3) #matrix for the linear constraints
uilinear[1,1]=1
uilinear[2,2]=1
uilinear[3,3]=1
uilinear[4,2]=-1
uilinear[4,3]=-1
cilinear=rep(0,4) #constant vector for the linear constraints
cilinear[4]=-1

###################################
## QMLE of d, a and b           ##
## Calculation of \hat{\lambda} ##
## Estimation of parameter \nu  ##
###################################
library(MASS)
```

```
calculate.parlinear=function(d,a,b,size,nu,sim,epsilon=0.001)
{
results1=matrix(NA,nrow=sim,ncol=3)
est.nu1=rep(NA,sim)
est.nu2=rep(NA,sim)
ratio=matrix(NA,nrow=sim,ncol=3)
for (i in 1:sim)
{
# Estimation
data.test=linearnegbin.ts(d,a,b,size,nu)[501:size,1]
arma_fit=arima(data.test,order=c(1,0,1),method="CSS")
ma_1=min(max(arma_fit$coef["ma1"],-1+epsilon),0-epsilon)
ar_1=min(max(arma_fit$coef["ar1"],0+epsilon-ma_1),1-epsilon)
sigma_sq=max(arma_fit$sigma2,epsilon)
theta_init=c(sigma_sq*(1-ar_1),-ma_1,ar_1+ma_1)
results1[i,]=constrOptim(theta=theta_init,f=liklinear.poisson,
             grad=scorelinear.poisson,data=data.test,ui=uilinear,
             ci=cilinear,outer.iterations=100,outer.eps=1e-05,
             method="BFGS")$par
hat.d=results1[i,1]
hat.a=results1[i,2]
hat.b=results1[i,3]
# Calculation of \hat{\lambda}
lambda=rep(NA,length(data.test))
lambda[1]=mean(data.test)
for (t in 2:length(data.test))
{
lambda[t]=hat.d+hat.a*lambda[t-1]+hat.b*data.test[t-1]
}
# Estimation of \nu
est.nu1[i]=(mean(((data.test-lambda)^{2}-lambda)/(lambda^{2})))^{-1}
```

```
est.nu2[i]=theta.mm(data.test,lambda,length(data.test)-3)
# Calculation of the ratio of standard errors of the estimators
sd1=sqrt(diag(ginv(information.poisson(c(hat.d,hat.a,hat.b),data.test))))
sd2=sqrt(diag(ginv(information.poisson(c(hat.d,hat.a,hat.b),data.test))
    %*%information.negbin(c(hat.d,hat.a,hat.b),data.test,est.nu2[i])
    %*%ginv(information.poisson(c(hat.d,hat.a,hat.b),data.test))))
ratio[i,]=sd1/sd2
}
return(list(results1,est.nu1,est.nu2,ratio))
}

mle.negbin=calculate.parlinear(0.5,0.4,0.5,1500,5,1000)

mean.est=apply(mle.negbin[[1]],2,mean)
mean.est

sd.est=apply(mle.negbin[[1]],2,sd)
sd.est

mean(mle.negbin[[2]])
sd(mle.negbin[[2]])

mean(mle.negbin[[3]])
sd(mle.negbin[[3]])

mean.ratio=apply(mle.negbin[[4]],2,mean)
mean.ratio
```

Construction of Figure 3.1.

```
############################
## Histograms and qq-plots ##
```

```
############################
d.data=(mle.negbin[[1]][,1]-mean.est[1])/sd.est[1]
a.data=(mle.negbin[[1]][,2]-mean.est[2])/sd.est[2]
b.data=(mle.negbin[[1]][,3]-mean.est[3])/sd.est[3]

par(mfrow=c(3,2))
hist(d.data,prob=T,ylab="",xlab="",main="")
lines(seq(-3.5,3.5,length=100),dnorm(seq(-3.5,3.5,length=100)))
qqnorm(d.data)
qqline(d.data)
hist(a.data,prob=T,ylab="",xlab="",main="")
lines(seq(-3.5,3.5,length=100),dnorm(seq(-3.5,3.5,length=100)))
qqnorm(a.data)
qqline(a.data)
hist(b.data,prob=T,ylab="",xlab="",main="")
lines(seq(-3.5,3.5,length=100),dnorm(seq(-3.5,3.5,length=100)))
qqnorm(b.data)
qqline(b.data)

########################################
## Kolmogorov-Smirnov test for the     ##
## normality of the standardized data  ##
########################################
ks.test(d.data,"pnorm")
ks.test(a.data,"pnorm")
ks.test(b.data,"pnorm")
```

Construction of left plot of Figure 3.3.

```
mat=matrix(NA,nrow=15,ncol=3)
for(j in 1:15)
{
```

```
mle.negbin=calculate.parlinear(0.5,0.4,0.5,1500,j,1000)
mean.ratio=apply(mle.negbin[[4]],2,mean)
mat[j,]=mean.ratio
}
mat

x=seq(1,15)
y1=mat[,1]
plot(x,y1,xlim=c(1,15),ylim=c(0.2,1),xlab=expression(nu),ylab="Ratio")
lines(lowess(x,y1,f=0.2))

y2=mat[,2]
points(x,y2)
lines(lowess(x,y2,f=0.2),lty=2)

y3=mat[,3]
points(x,y3)
lines(lowess(x,y3,f=0.2),lty=3)

leg.names=c("Ratio of d","Ratio of a", "Ratio of b")
legend(locator(1),leg.names,lty=1:3)
```

Construction of the second rows in parentheses of Table 3.1 and Table 3.2.

```
f=function(d,a,b,size,nu,sim)
{
final=matrix(0,nrow=3,ncol=3)
hessian.matrix=matrix(0,nrow=3,ncol=3)
G.matrix=matrix(0,nrow=3,ncol=3)
for (i in 1:sim)
{
data=linearnegbin.ts(d,a,b,size,nu)[501:size,1]
```

```
theta=c(d,a,b)
G=information.poisson(theta,data)/length(data)
G1=information.negbin(theta,data,nu)/length(data)
sandwich=ginv(G)%*%G1%*%ginv(G)/length(data)
hess=constrOptim(theta=theta,f=liklinear.poisson,grad=scorelinear.poisson,
     data=data,ui=uilinear,ci=cilinear,control=list(maxit=0),method="BFGS",
     hessian=TRUE)$hessian/length(data)
final=final+sandwich
hessian.matrix=hessian.matrix+hess
G.matrix=G.matrix+G
final=final
hessian.matrix=hessian.matrix/sim
G.matrix=G.matrix/sim
}
return(list(final,hessian.matrix,G.matrix))
}


result=f(0.5,0.4,0.5,1500,5,1000)
sd=sqrt(diag(result[[1]]/1000))
round(sd,4)
result[[2]]
result[[3]]
```

Construction of top panels of Tables 3.7, 3.8, 3.9, 3.10, Figures 3.5, 3.7, 3.9, 3.11 and top plots of Figures 3.6, 3.8, 3.10 and 3.12.

```
########################
## Transactions data ##
########################
ericsson=read.table("ericssonfirstday.txt",header=T,dec=",",as.is=F)
names(ericsson)
transactions=ericsson$trans
```

```
ydata=transactions

##################
## Measles data ##
##################
data=read.table("ewcitmeas.txt",header=T,dec=",",as.is=F)
names(data)
ydata=data$Sheffield[1601:2050]

#################
## Births data ##
#################
data=read.table("births.txt",header=T,dec=",",as.is=F)
names(data)
ydata=data$bb

####################
## Homicides data ##
####################
data=read.table("homicides.txt",header=T,dec=",",as.is=F)
names(data)
ydata=data$X0

# Time series and acf function
par(mfrow=c(1,2))
plot(ydata,type="l",xlab="Time",ylab="Response")
acf(ydata,main="",xlab="Lag",ylab="ACF")

# Estimation of d, a, b
epsilon=0.001
arma_fit=arima(ydata,order=c(1,0,1),method="CSS")
```

```
ma_1=min(max(arma_fit$coef["ma1"],-1+epsilon),0-epsilon)
ar_1=min(max(arma_fit$coef["ar1"],0+epsilon-ma_1),1-epsilon)
sigma_sq=max(arma_fit$sigma2,epsilon)
theta_init=c(sigma_sq*(1-ar_1),-ma_1,ar_1+ma_1)
results=constrOptim(theta=theta_init,f=liklinear.poisson,
        grad=scorelinear.poisson,data=ydata,ui=uilinear,
        ci=cilinear,outer.iterations=100,outer.eps=1e-05,method="BFGS")
results1=results$par
hat.d=results1[1]
hat.a=results1[2]
hat.b=results1[3]
# Calculation of \hat{\lambda}
lambda=rep(NA,length(ydata))
lambda[1]=mean(ydata)
for (t in 2:length(ydata))
{
lambda[t]=hat.d+hat.a*lambda[t-1]+hat.b*ydata[t-1]
}
# Estimation of \nu
est.nu1=(mean(((ydata-lambda)^{2}-lambda)/(lambda^{2})))^{-1}
est.nu2=theta.mm(ydata,lambda,length(ydata)-length(results1))
# Standard errors of the estimators
sd1=sqrt(diag(ginv(information.poisson(c(hat.d,hat.a,hat.b),ydata))))
sd2=sqrt(diag(ginv(information.poisson(c(hat.d,hat.a,hat.b),ydata))
    %*%information.negbin(c(hat.d,hat.a,hat.b),ydata,est.nu2)
    %*%ginv(information.poisson(c(hat.d,hat.a,hat.b),ydata))))

##################################
## Cumulative periodogram plot ##
##################################
cpgramnew=function(ts,taper=0.1,main=paste("Series: ",
```

```
          deparse(substitute(ts))),ci.col="blue")
{
main
if (NCOL(ts)>1)
stop("only implemented for univariate time series")
x=as.vector(ts)
x=x[!is.na(x)]
x=spec.taper(scale(x,TRUE,FALSE),p=taper)
y=Mod(fft(x))^2/length(x)
y[1]=0
n=length(x)
x=(0:(n/2))*frequency(ts)/n
if (length(x)%%2==0)
{
n=length(x)-1
y=y[1:n]
x=x[1:n]
}
else
y=y[seq(along=x)]
xm=frequency(ts)/2
mp=length(x)-1
crit=1.358/(sqrt(mp)+0.12+0.11/sqrt(mp))
plot(x,cumsum(y)/sum(y),type="s",xlim=c(0,xm),ylim=c(0,1),xaxs="i",
     yaxs="i",xlab="frequency",ylab="")
lines(c(0,xm*(1-crit)),c(crit,1),col=ci.col,lty=2)
lines(c(xm*crit,xm),c(0,1-crit),col=ci.col,lty=2)
title(main=main)
invisible()
}
```

```
#######################
## Pearson residuals ##
#######################
resnb=(ydata-lambda)/sqrt(lambda+lambda^2/est.nu2)

plot(ydata,type="l",col=1,xlab="Time",ylab="Transactions per minute")
lines(lambda,type="l",col="grey")
cpgramnew(resnb,main="Pearson Residuals")
```

R code regarding the nonlinear model (3.11) for the construction of Tables 3.3, 3.4, bottom panels of Tables 3.7, 3.8, 3.9, 3.10, Figure 3.2, right plot of Figure 3.3 and bottom plots of Figures 3.6, 3.8, 3.10 and 3.12.

```
##############################################
## Simulate the nonlinear model             ##
## using the negative binomial distribution ##
##############################################
nonlinnegbin.gamma.ts=function(d,a,gamma,b,size,nu)
{
y=rep(NA,size)
mu=rep(NA,size)
mu[1]=1 #initial value
y[1]=rnbinom(1,size=nu,mu=mu[1])
for (t in 2:size)
{
mu[t]=d/((1+mu[t-1])^(gamma))+a*mu[t-1]+b*y[t-1]
y[t]=rnbinom(1,size=nu,mu=mu[t])
}
return(cbind(y,mu))
}

##################################################
```

```
## (Poisson) Quasi-likelihood for the nonlinear  ##
## model for gamma known                         ##
################################################
liknonlin.gamma.poisson=function(theta,gamma,data)
{
lambda=rep(NA,length(data))
loglik=rep(NA,length(data))
lambda[1]=1
loglik[1]=0
for (t in 2:length(data))
{
lambda[t]=theta[1]/((1+lambda[t-1])^(gamma))+theta[2]*lambda[t-1]
         +theta[3]*data[t-1]
if (lambda[t]==0) loglik[t]= 0 else
if (lambda[t] >0) loglik[t]= -data[t]*log(lambda[t])+lambda[t]
}
final=sum(loglik)
}

################################################
## (Poisson) Score function for the nonlinear ##
## model for gamma known                      ##
################################################
scorenonlin.gamma.poisson=function(theta,gamma,data)
{
lambda=rep(NA,length(data))
lambda[1]=1
first=rep(NA,length(data))
first[1]=1
second=rep(NA,length(data))
second[1]=1
```

```
third=rep(NA,length(data))
third[1]=1
s1=rep(NA,length(data))
s2=rep(NA,length(data))
s3=rep(NA,length(data))
for (t in 2:length(data))
{
lambda[t]=theta[1]/((1+lambda[t-1])^(gamma))+theta[2]*lambda[t-1]
         +theta[3]*data[t-1]
first[t]=1/((1+lambda[t-1])^(gamma))+(theta[2]-theta[1]*gamma
        *((1+lambda[t-1])^(-gamma-1)))*first[t-1]
second[t]=lambda[t-1]+(theta[2]-theta[1]*gamma*((1+lambda[t-1])^(-gamma-1)))
         *second[t-1]
third[t]=(theta[2]-theta[1]*gamma*((1+lambda[t-1])^(-gamma-1)))*third[t-1]
        +data[t-1]
s1[t]=-((data[t]/lambda[t])-1)*first[t]
s2[t]=-((data[t]/lambda[t])-1)*second[t]
s3[t]=-((data[t]/lambda[t])-1)*third[t]
}
ss1=sum(s1[-1])
ss2=sum(s2[-1])
ss3=sum(s3[-1])
score=c(ss1,ss2,ss3)
}

#####################################################################
## Information matrix based on the Poisson distribution-G matrix ##
#####################################################################
information.poisson=function(theta,gamma,data)
{
lambda=rep(NA,length(data))
```

```
lambda[1]=1
first=rep(NA,length(data))
first[1]=1
second=rep(NA,length(data))
second[1]=1
third=rep(NA,length(data))
third[1]=1
Information=matrix(0,nrow=3,ncol=3)
s1=rep(NA,length(data))
s2=rep(NA,length(data))
s3=rep(NA,length(data))
for (t in 2:length(data))
{
lambda[t]=theta[1]/((1+lambda[t-1])^(gamma))+theta[2]*lambda[t-1]
         +theta[3]*data[t-1]
first[t]=1/((1+lambda[t-1])^(gamma))+(theta[2]-theta[1]*gamma
        *((1+lambda[t-1])^(-gamma-1)))*first[t-1]
second[t]=lambda[t-1]+(theta[2]-theta[1]*gamma*((1+lambda[t-1])^(-gamma-1)))
         *second[t-1]
third[t]=(theta[2]-theta[1]*gamma*((1+lambda[t-1])^(-gamma-1)))*third[t-1]
        +data[t-1]
s1[t]=first[t]
s2[t]=second[t]
s3[t]=third[t]
var.comp=(1/sqrt(lambda[t]))*c(s1[t],s2[t],s3[t])
Information=Information+var.comp%*%t(var.comp)
}
return(Information)
}

#####################################################################
```

```
## Information matrix based on the negative binomial distribution ##
## G_{1} matrix                                                    ##
#####################################################################
information.negbin=function(theta,gamma,data,nu)
{
lambda=rep(NA,length(data))
lambda[1]=1
first=rep(NA,length(data))
first[1]=1
second=rep(NA,length(data))
second[1]=1
third=rep(NA,length(data))
third[1]=1
Information=matrix(0,nrow=3,ncol=3)
s1=rep(NA,length(data))
s2=rep(NA,length(data))
s3=rep(NA,length(data))
for (t in 2:length(data))
{
lambda[t]=theta[1]/((1+lambda[t-1])^(gamma))+theta[2]*lambda[t-1]
         +theta[3]*data[t-1]
first[t]=1/((1+lambda[t-1])^(gamma))+(theta[2]-theta[1]*gamma
       *((1+lambda[t-1])^(-gamma-1)))*first[t-1]
second[t]=lambda[t-1]+(theta[2]-theta[1]*gamma*((1+lambda[t-1])^(-gamma-1)))
         *second[t-1]
third[t]=(theta[2]-theta[1]*gamma*((1+lambda[t-1])^(-gamma-1)))*third[t-1]
        +data[t-1]
s1[t]=first[t]
s2[t]=second[t]
s3[t]=third[t]
var.comp=(sqrt(1/lambda[t]+1/nu))*c(s1[t],s2[t],s3[t])
```

```
Information=Information+var.comp%*%t(var.comp)
}
return(Information)
}

###################################
## QMLE of d, a and b           ##
## Calculation of \hat{\lambda} ##
## Estimation of parameter \nu  ##
###################################
library(MASS)
calculate.parnonlinear=function(d,a,gamma,b,size,nu,sim,epsilon=0.001)
{
results1=matrix(NA,nrow=sim,ncol=3)
est.nu1=rep(NA,sim)
est.nu2=rep(NA,sim)
ratio=matrix(NA,nrow=sim,ncol=3)
for (i in 1:sim)
{
# Estimation
data.test=nonlinnegbin.gamma.ts(d,a,gamma,b,size,nu)[501:size,1]
arma_fit=arima(data.test,order=c(1,0,1),method="CSS")
ma_1=min(max(arma_fit$coef["ma1"],-1+epsilon),0-epsilon)
ar_1=min(max(arma_fit$coef["ar1"],0+epsilon-ma_1),1-epsilon)
sigma_sq=max(arma_fit$sigma2,epsilon)
theta_init=c(sigma_sq*(1-ar_1),-ma_1,ar_1+ma_1)
results=constrOptim(theta=theta_init,f=liklinear.poisson,
        grad=scorelinear.poisson,data=data.test,ui=uilinear,
        ci=cilinear,outer.iterations=100,
        outer.eps=1e-05,method="BFGS")$par
theta_init1=c(results[1],results[2],results[3])
```

```
results1[i,]=constrOptim(theta=theta_init1,f=liknonlin.gamma.poisson,
             grad=scorenonlin.gamma.poisson,data=data.test,gamma=gamma,
             ui=uilinear,ci=cilinear,outer.iterations=100,
             outer.eps=1e-05,method="BFGS")$par
hat.d=results1[i,1]
hat.a=results1[i,2]
hat.b=results1[i,3]
# Calculation of \hat{\lambda}
lambda=rep(NA,length(data.test))
lambda[1]=mean(data.test)
for (t in 2:length(data.test))
{
gamma=gamma
lambda[t]=hat.d*((1+lambda[t-1])^(-gamma))
         +hat.a*lambda[t-1]+hat.b*data.test[t-1]
}
# Estimation of \nu
est.nu1[i]=(mean(((data.test-lambda)^{2}-lambda)/(lambda^{2})))^{-1}
est.nu2[i]=theta.mm(data.test,lambda,length(data.test)-3)
# Calculation of the ratio of standard errors of the estimators
sd1=sqrt(diag(ginv(information.poisson(c(hat.d,hat.a,hat.b),gamma,
         data.test))))
sd2=sqrt(diag(ginv(information.poisson(c(hat.d,hat.a,hat.b),gamma,data.test))
    %*%information.negbin(c(hat.d,hat.a,hat.b),gamma,data.test,est.nu2[i])
    %*%ginv(information.poisson(c(hat.d,hat.a,hat.b),gamma,data.test))))
ratio[i,]=sd1/sd2
}
return(list(results1,est.nu1,est.nu2,ratio))
}

mle.negbin=calculate.parnonlinear(0.5,0.4,0.5,0.5,1500,5,1000)
```

```
mean.est=apply(mle.negbin[[1]],2,mean)
mean.est

sd.est=apply(mle.negbin[[1]],2,sd)
sd.est

mean(mle.negbin[[2]])
sd(mle.negbin[[2]])

mean(mle.negbin[[3]])
sd(mle.negbin[[3]])

mean.ratio=apply(mle.negbin[[4]],2,mean)
mean.ratio
```

Construction of Figure 3.2.

```
############################
## Histograms and qq-plots ##
############################
d.data=(mle.negbin[[1]][,1]-mean.est[1])/sd.est[1]
a.data=(mle.negbin[[1]][,2]-mean.est[2])/sd.est[2]
b.data=(mle.negbin[[1]][,3]-mean.est[3])/sd.est[3]

par(mfrow=c(3,2))
hist(d.data,prob=T,ylab="",xlab="",main="")
lines(seq(-3.5,3.5,length=100),dnorm(seq(-3.5,3.5,length=100)))
qqnorm(d.data)
qqline(d.data)
hist(a.data,prob=T,ylab="",xlab="",main="")
lines(seq(-3.5,3.5,length=100),dnorm(seq(-3.5,3.5,length=100)))
```

```
qqnorm(a.data)
qqline(a.data)
hist(b.data,prob=T,ylab="",xlab="",main="")
lines(seq(-3.5,3.5,length=100),dnorm(seq(-3.5,3.5,length=100)))
qqnorm(b.data)
qqline(b.data)
```

Construction of right plot of Figure 3.3.

```
mat=matrix(NA,nrow=15,ncol=3)
for(j in 1:15)
{
mle.negbin=calculate.parnonlinear(0.5,0.4,0.5,0.5,1500,j,1000)
mean.ratio=apply(mle.negbin[[4]],2,mean)
mat[j,]=mean.ratio
}
mat

x=seq(1,15)
y1=mat[,1]
plot(x,y1,xlim=c(1,15),ylim=c(0.2,1),xlab=expression(nu),ylab="Ratio")
lines(lowess(x,y1,f=0.2))

y2=mat[,2]
points(x,y2)
lines(lowess(x,y2,f=0.2),lty=2)

y3=mat[,3]
points(x,y3)
lines(lowess(x,y3,f=0.2),lty=3)

leg.names=c("Ratio of d","Ratio of a","Ratio of b")
```

```
legend(locator(1),leg.names,lty=1:3)
```

Construction of the second rows in parentheses of Table 3.3 and Table 3.4.

```
f=function(d,a,gamma,b,size,nu,sim)
{
final=matrix(0,nrow=3,ncol=3)
hessian.matrix=matrix(0,nrow=3,ncol=3)
G.matrix=matrix(0,nrow=3,ncol=3)
for (i in 1:sim)
{
data=nonlinnegbin.gamma.ts(d,a,gamma,b,size,nu)[501:size,1]
theta=c(d,a,b)
G=information.poisson(theta,gamma,data)/length(data)
G1=information.negbin(theta,gamma,data,nu)/length(data)
sandwich=ginv(G)%*%G1%*%ginv(G)/length(data)
hess=constrOptim(theta=theta,gamma=gamma,f=liknonlin.gamma.poisson,
     grad=scorenonlin.gamma.poisson,data=data,ui=uilinear,ci=cilinear,
     control=list(maxit=0),method="BFGS",hessian=TRUE)$hessian/length(data)
final=final+sandwich
hessian.matrix=hessian.matrix+hess
G.matrix=G.matrix+G
final=final
hessian.matrix=hessian.matrix/sim
G.matrix=G.matrix/sim
}
return(list(final,hessian.matrix,G.matrix))
}

result=f(0.5,0.4,0.5,0.5,1500,5,1000)
sd=sqrt(diag(result[[1]]/1000))
round(sd,4)
```

```
result[[2]]
result[[3]]
```

Construction of bottom panels of Tables 3.7, 3.8, 3.9, 3.10 and bottom plots of Figures 3.6, 3.8, 3.10 and 3.12.

```
#########################
## Transactions data ##
#########################
ericsson=read.table("ericssonfirstday.txt",header=T, dec=",", as.is=F)
names(ericsson)
transactions=ericsson$trans
ydata=transactions

###################
## Measles data ##
###################
data=read.table("ewcitmeas.txt",header=T,dec=",",as.is=F)
names(data)
ydata=data$Sheffield[1601:2050]

#################
## Births data ##
#################
data=read.table("births.txt",header=T,dec=",",as.is=F)
names(data)
ydata=data$bb

####################
## Homicides data ##
####################
data=read.table("homicides.txt",header=T,dec=",",as.is=F)
```

```
names(data)
ydata=data$X0

# Estimation of d, a, b
epsilon=0.001
arma_fit=arima(ydata,order=c(1,0,1),method="CSS")
ma_1=min(max(arma_fit$coef["ma1"],-1+epsilon),0-epsilon)
ar_1=min(max(arma_fit$coef["ar1"],0+epsilon-ma_1),1-epsilon)
sigma_sq=max(arma_fit$sigma2,epsilon)
theta_init=c(sigma_sq*(1-ar_1),-ma_1,ar_1+ma_1)
results=constrOptim(theta=theta_init,f=liklinear.poisson,
        grad=scorelinear.poisson,data=ydata,ui=uilinear,ci=cilinear,
        outer.iterations=100,outer.eps=1e-05,method="BFGS")$par
results1=constrOptim(theta=c(results[1],results[2],results[3]),
         f=liknonlin.gamma.poisson,grad=scorenonlin.gamma.poisson,
         data=ydata,gamma=0.5,outer.iterations=100,ui=uilinear,ci=cilinear,
         outer.eps=1e-05,method="BFGS")
hat.d=results1$par[1]
hat.a=results1$par[2]
hat.b=results1$par[3]
# Calculation of \hat{\lambda}
lambda=rep(NA,length(ydata))
lambda[1]=mean(ydata)
for (t in 2:length(ydata))
{
gamma=0.5
lambda[t]=hat.d/((1+lambda[t-1])^(gamma))+hat.a*lambda[t-1]+hat.b*ydata[t-1]
}
# Estimation of \nu
est.nu1=(mean(((ydata-lambda)^{2}-lambda)/(lambda^{2}) ))^{-1}
est.nu2=theta.mm(ydata,lambda,length(ydata)-3)
```

```
# Standard errors of the estimators
sd1=sqrt(diag(ginv(information.poisson(c(hat.d,hat.a,hat.b),0.5,ydata))))
sd2=sqrt(diag(ginv(information.poisson(c(hat.d,hat.a,hat.b),gamma=0.5,ydata))
    %*%information.negbin(c(hat.d,hat.a,hat.b),gamma=0.5,ydata,est.nu2)
    %*%ginv(information.poisson(c(hat.d,hat.a,hat.b),gamma=0.5,ydata))))

########################
## Pearson residuals ##
########################
resnb=(ydata-lambda)/sqrt(lambda+lambda^2/est.nu2)

plot(ydata,type="l",col=1,xlab="Time",ylab="Transactions per minute")
lines(lambda,type="l",col="grey")
cpgramnew(resnb,main="Pearson Residuals")
```

If we want to simulate data from a mixed Poisson process, whose mixing variable Z is not necessarily negative binomial distributed, we work analogously as before. The differences in the R code are shown below. This is how we construct Tables 3.5, 3.6 and Figure 3.4.

```
##########################################
## Simulate a mixed Poisson - linear model ##
##########################################
linear.mixed.poisson.ts=function(d,a1,b1,size)
{
y=rep(NA,size)
mu=rep(NA,size)
z=rep(NA,size)
#z=rchisq(n=size,df=1)              #chisquare
#z=rbinom(n=size,size=10,prob=1/10) #binomial
z=runif(size,0,2)                  #uniform
#z=rgeom(size,5/10)                #geometric
mu[1]=1
```

```
y[1]=rpois(1,z[1]*mu[1])
for (t in 2:size)
{
mu[t]=d+a1*mu[t-1]+b1*y[t-1]
y[t]=rpois(1,z[t]*mu[t])
}
return(cbind(y,mu,z))
}

####################################
## Constrains to obtain the QMLE  ##
## d>0, a_1>0, b_1>0, 0<a_1+b_1<1 ##
####################################
constrainslinear.poisson=function(z)
{
uilinear=matrix(0,nrow=4,ncol=3) #matrix for the linear constraints
uilinear[1,1]=1
uilinear[2,2]=1
uilinear[3,3]=1
uilinear[4,2]=-1
uilinear[4,3]=-1
cilinear=rep(0,4) #constant vector for the linear constraints
cilinear[4]=-1/mean(z)
return(list(uilinear,cilinear))
}

####################################
## QMLE of d, a and b             ##
## Calculation of \hat{\lambda}  ##
####################################
library(MASS)
```

```
mixedpoisson.linear=function(d,a1,b1,size,sim,epsilon=0.001)
{
results1=matrix(NA,nrow=sim,ncol=3)
var_z=rep(NA,sim)
for (i in 1:sim)
{
# Estimation
data1=linear.mixed.poisson.ts(d,a1,b1,size)
data.test=data1[501:size,1]
z=data1[501:size,3]
arma_fit=arima(data.test,order=c(1,0,1),method="CSS")
ma_1=min(max(arma_fit$coef["ma1"],-1+epsilon),0-epsilon)
ar_1=min(max(arma_fit$coef["ar1"],0+epsilon-ma_1),1-epsilon)
sigma_sq=max(arma_fit$sigma2,epsilon)
theta_init=c(sigma_sq*(1-ar_1),-ma_1,ar_1+ma_1)
uilinear=constrainslinear.poisson(z)[[1]]
cilinear=constrainslinear.poisson(z)[[2]]
results1[i,]=constrOptim(theta=theta_init,f=liklinear.poisson,
             grad=scorelinear.poisson,data=data.test,ui=uilinear,
             ci=cilinear,z=z,outer.iterations=100,outer.eps=1e-05,
             method="BFGS")$par
hat.d=results1[i,1]
hat.a=results1[i,2]
hat.b=results1[i,3]
# Calculation of \hat{\lambda}
lambda=rep(NA, length(data.test))
lambda[1]=mean(data.test)
for (t in 2:length(data.test))
{
lambda[t]=hat.d+hat.a*lambda[t-1]+hat.b*data.test[t-1]
}
```

```
# Estimation of \sigma_Z^2
var_z[i]=1/theta.mm(data.test,lambda,length(data.test)-3)
}
return(list(results1,var_z))
}

################################################
## Simulate a mixed Poisson - nonlinear model ##
################################################
nonlinear.mixed.poisson.ts=function(d,a1,gamma,b1,size)
{
y=rep(NA,size)
mu=rep(NA,size)
z=rep(NA,size)

#z=rchisq(n=size,df=1)             #chisquare
#z=rbinom(n=size,size=10,prob=1/10) #binomial
z=runif(size,0,2)                 #uniform
#z=rgeom(size,5/10)                #geometric
mu[1]=1
y[1]=rpois(1,z[1]*mu[1])
for (t in 2:size)
{
mu[t]=d/((1+mu[t-1])^(gamma))+a1*mu[t-1]+b1*y[t-1]
y[t]=rpois(1,z[t]*mu[t])
}
return(cbind(y,mu,z))
}

mixedpoisson.nonlinear=function(d,a1,gamma,b1,size,sim,epsilon=0.001)
{
```

```
results1=matrix(NA,nrow=sim,ncol=3)
var_z=rep(NA,sim)
for (i in 1:sim)
{
# Estimation
data1=nonlinear.mixed.poisson.ts(d,a1,gamma,b1,size)
data.test=data1[501:size,1]
z=data1[501:size,3]
arma_fit=arima(data.test,order=c(1,0,1),method="CSS")
ma_1=min(max(arma_fit$coef["ma1"],-1+epsilon),0-epsilon)
ar_1=min(max(arma_fit$coef["ar1"],0+epsilon-ma_1),1-epsilon)
sigma_sq=max(arma_fit$sigma2,epsilon)
theta_init=c(sigma_sq*(1-ar_1),-ma_1,ar_1+ma_1)
uilinear=constrainslinear.poisson(z)[[1]]
cilinear=constrainslinear.poisson(z)[[2]]
results=constrOptim(theta=theta_init,f=liklinear.poisson,
        grad=scorelinear.poisson,data=data.test,ui=uilinear,ci=cilinear,
        outer.iterations=100,outer.eps=1e-05,method="BFGS")$par
theta_init1=c(results[1],results[2],results[3])
results1[i,]=constrOptim(theta=theta_init1,f=liknonlin.gamma.poisson,
             grad=scorenonlin.gamma.poisson,data=data.test,gamma=gamma,
             ui=uilinear,ci=cilinear,outer.iterations=100,
             outer.eps=1e-05,method="BFGS")$par
hat.d=results1[i,1]
hat.a=results1[i,2]
hat.b=results1[i,3]
# Calculation of \hat{\lambda}
lambda=rep(NA,length(data.test))
lambda[1]=mean(data.test)
for (t in 2:length(data.test))
{
```

```
gamma=gamma
lambda[t]=hat.d*((1+lambda[t-1])^(-gamma))+hat.a*lambda[t-1]
          +hat.b*data.test[t-1]
}
# Estimation of \sigma_Z^2
var_z[i]=1/theta.mm(data.test,lambda,length(data.test)-3)
}
return(list(results1,var_z))
}
```

Chapter 4

On Count Time Series Prediction

4.1 Introduction

This chapter studies the prediction problem in the context of count time series. More specifically, we follow the recent methodology of Czado et al. [13], where various tools for predictive model assessment are developed for independent but not identically distributed data. We show that these methods can also be applied for count dependent data. We focus on Poisson and negative binomial distributions since these are occurred in applications more frequently (and they are special cases of the mixed Poisson class of models); however the methods can be applied to other discrete distributions provided that they are suitably parameterized, as discussed in Chapter 3, and the distribution of the mixing variable is known. We assess the predictive performance of the proposed models by extending the tools developed by Czado et al. [13]. More specifically, we address the problem of examining probabilistic calibration, marginal calibration and sharpness of the proposed models.

4.2 Autoregressive Modeling and Inference

As outlined in Section 3.2 of Chapter 3, we consider count time series models that are defined by (3.1) with mean specified by (3.4). In particular, in this chapter we focus on count data whose distribution given the past is either the Poisson defined by (3.2) or the negative binomial given by (3.3). As stated before, the negative binomial distribution is

suitably parameterized to have equal mean as the Poisson distribution, that is λ_t. However, its conditional variance is equal to $\lambda_t + \lambda_t^2/\nu$ and it is greater than λ_t, which corresponds to the conditional variance of the Poisson. This shows that the negative binomial will tend to fit overdispersed data better.

Models (3.6), (3.10) and (3.11) defined in Section 3.2 of Chapter 3 are used to fit the data in this chapter. Quasi maximum likelihood methodology outlined in the previous chapter is used to obtain the consistent estimators of the regression parameters for each model.

4.3 Assessment of the Predictive Performance

A major issue of time series analysis is to provide forecasts for future quantities. In our study, since $E(Y_t \mid \mathcal{F}_{t-1}^{Y,\lambda}) = \lambda_t$, the optimal predictor of Y_t given its past, is given by the conditional expectation λ_t, in terms of mean square error. Obviously, in applications we employ $\hat{\lambda}_t = \lambda_t(\hat{\boldsymbol{\theta}})$. We focus exclusively on this predictor and the associated probability models as introduced by (3.2) and (3.3). Regardless the chosen response distribution, it is clear that $\hat{\lambda}_t$ is identical for both cases provided that we employ the QMLE obtained by (3.14).

Following Gneiting et al. [43], we take the point of view that predictions should be probabilistic in nature. In addition, they should strive to maximize the sharpness of the predictive distribution subject to calibration. Calibration, refers to the statistical consistency between the predictive distribution and the observations, and it is a joint property of the forecasts and the values that utilize. The notion of sharpness refers to the concentration of the predictive distribution and is a property of the forecasts only. It follows that if the predictive distribution is more concentrated, then the forecasts obtained are sharper. Therefore, this will yield better predictions subject to calibration. In this section we provide diagnostic tools to evaluate the predictive performance.

4.3.1 Assessment of Probabilistic Calibration

The key tool in assessing probabilistic calibration is the nonrandomized Probability Integral Transformation (PIT) histogram as discussed by Czado et al. [13]. This tool is used

for checking the statistical consistency between the predictive distribution and the distribution of the observations. If the observation is drawn from the predictive distribution, then the PIT has a standard uniform distribution.

In the case of count data, the predictive distribution is discrete and therefore the PIT is no longer uniform. To remedy this, several authors (see for instance Handcock and Morris [49]) have suggested a randomized PIT. Specifically, if P is the predictive distribution, $x \sim P$ is the observed count and v is standard uniform and independent of x, then

$$u = P_{x-1} + v(P_x - P_{x-1}), \tag{4.1}$$

is standard uniform, where we define $P_{-1} = 0$. Czado et al. [13] recently proposed a non randomized uniform version of the PIT (see Definition 2.1.14) replacing the randomized PIT value in (4.1) by

$$F(u|x) = \begin{cases} 0 & u < P_{x-1}, \\ (u - P_{x-1})/(P_x - P_{x-1}) & P_{x-1} \leq u < P_x, \\ 1 & u \geq P_x. \end{cases}$$

Aggregating over a relevant set of n predictions we obtain the mean PIT,

$$\bar{F}(u) = \frac{1}{n} \sum_{i=1}^{n} F^{(i)}(u|x^{(i)}), \quad 0 \leq u \leq 1,$$

where $F^{(i)}$ is based on the predictive distribution $P^{(i)}$ and the observed count $x^{(i)}$.

Then the mean PIT is compared to the cumulative distribution function (cdf) of the standard uniform distribution. The comparison is performed by plotting a non randomized PIT histogram, which can be used as a diagnostic tool. After selecting the number of bins, J, compute

$$f_j = \bar{F}(\frac{j}{J}) - \bar{F}(\frac{j-1}{J})$$

for equally spaced bins $j = 1, \ldots, J$, plot the histogram with height f_j for bin j and check for uniformity. Deviations from uniformity hint at reasons for forecasting failures and model deficiencies. U-shaped histograms point at underdispersed predictive distributions, hump or inverse U-shaped histograms indicate overdispersion and uniformity of histograms

hints well-calibrated predictive distributions.

To construct the plots of the PIT histograms shown in Section 4.5 we operate as follows. We fit model (3.5) to the data by using the quasi-likelihood function (3.13). After obtaining consistent estimators for the regression parameters, we estimate the mean process λ_t by $\hat{\lambda}_t = \lambda_t(\hat{\boldsymbol{\theta}})$ and the parameter ν by either $\hat{\nu}_1$ or $\hat{\nu}_2$. Based on the simulation results of the previous chapter, we suggest to use the estimator $\hat{\nu}_2$. Then, the PIT is based on the conditional cumulative distribution

$$F(u|Y_t = y) = \begin{cases} 0 & u < P_{y-1}, \\ (u - P_{y-1})/(P_y - P_{y-1}) & P_{y-1} \leq u < P_y, \\ 1 & u \geq P_y, \end{cases}$$

where P_y is equal to the conditional cdf either of the Poisson distribution evaluated at $\hat{\lambda}_t$, or of the negative binomial evaluated at $\hat{\lambda}_t$ and $\hat{\nu}_2$. Subsequently, the mean PIT to be used for plotting the histogram is given by

$$\bar{F}(u) = \frac{1}{n} \sum_{t=1}^{n} F^{(t)}(u|y_t), \quad 0 \leq u \leq 1.$$

4.3.2 Assessment of Marginal Calibration

We now turn to the question of assessing marginal calibration. Suppose that the observed time series $\{Y_t,\ t = 1, \dots n\}$ is stationary which follows marginally the cdf $G(\cdot)$. In addition, we assume that a candidate forecaster picks a probabilistic forecast in the form of a predictive cdf $P_t(x) = \mathrm{P}(Y_t \leq x \mid \mathcal{F}_{t-1}^{Y,\lambda})$. In the case that we study, we have $P_t(\cdot)$ to be either the cdf of a Poisson with mean $\hat{\lambda}_t$ or of a negative binomial distribution evaluated at $\hat{\lambda}_t$ and $\hat{\nu}$. We follow Gneiting et al. [43] to assess marginal calibration by comparing the average predictive cdf

$$\bar{P}(x) = \frac{1}{n} \sum_{t=1}^{n} P_t(x), \qquad x \in \mathbb{R},$$

to the empirical cdf of the observations given by

$$\hat{G}(x) = \frac{1}{n}\sum_{t=1}^{n} 1(Y_t \leq x), \qquad x \in \mathbb{R}.$$

To display the marginal calibration plot, we plot the difference of the two cdf

$$\bar{P}(x) - \hat{G}(x), \qquad x \in \mathbb{R}, \tag{4.2}$$

for each of the various forecasters. If the marginal calibration hypothesis is true, then we expect minor fluctuations about 0. Major excursions from zero hint that the forecaster lacks marginal calibration. To see this, note that for x fixed and $P_t(x)$ as before, we obtain

$$\bar{P}(x) = \frac{1}{n}\sum_{t=1}^{n} P_t(x) \quad \xrightarrow{p} \quad \mathrm{E}(P(Y_t \leq x \mid \mathcal{F}_{t-1}^{Y,\lambda})) = P(Y_t \leq x),$$

because of ergodicity of $\{Y_t\}$. In addition,

$$\hat{G}(x) = \frac{1}{n}\sum_{t=1}^{n} 1(Y_t \leq x) \quad \xrightarrow{p} \quad \mathrm{E}(1(Y_t \leq x)) = P(Y_t \leq x).$$

Therefore, $\bar{P}(x)$ and $\hat{G}(x)$ converge in probability to the marginal cdf of $\{Y_t\}$. Hence, plot (4.2) would indicate agreement (or disagreement) between the predictive distribution and the marginal empirical distribution of the observed counts.

4.3.3 Assessment of Sharpness

Addressing sharpness is accomplished via scoring rules. These rules provide numerical scores and form summary measures for the assessment of the predictive performance. In addition, scoring rules help us to rank the competing forecast models. They are negatively oriented penalties that the forecaster wishes to minimize, see also Czado et al. [13].

Assume that $P_t(\cdot)$ denotes the predictive cdf as before. The score will be denoted, in general, by $s(P_t, Y_t)$. Put $p_y = \mathrm{P}(Y_t = y \mid \mathcal{F}_{t-1}^{Y,\lambda})$, for $y = 0, 1, 2, \ldots$; recall (3.2) and (3.3). We consider seven different examples of scoring rules, as in Czado et al. [13], for the case of count time series.

- Logarithmic score

$$logs(P_t, Y_t) = -\log p_y. \tag{4.3}$$

- Quadratic or Brier score

$$qs(P_t, Y_t) = -2p_y + \|p\|^2, \tag{4.4}$$

where $\|p\|^2 = \sum_{y=0}^{\infty} p_y^2$.

- Spherical score

$$sphs(P_t, Y_t) = -\frac{p_y}{\|p\|}. \tag{4.5}$$

- Ranked probability score

$$rps(P_t, Y_t) = \sum_{x=0}^{\infty} (P_t(x) - 1(Y_t \leq x))^2. \tag{4.6}$$

- Dawid-Sebastiani score

$$dss(P_t, Y_t) = \Big(\frac{Y_t - \mu_{P_t}}{\sigma_{P_t}}\Big)^2 + 2\log \sigma_{P_t}, \tag{4.7}$$

where μ_{P_t} and σ_{P_t} are the mean and the standard deviation of the predictive distribution P_t, respectively.

- Normalized squared error score

$$nses(P_t, Y_t) = \Big(\frac{Y_t - \mu_{P_t}}{\sigma_{P_t}}\Big)^2. \tag{4.8}$$

- Squared error score

$$ses(P_t, Y_t) = (Y_t - \mu_{P_t})^2. \tag{4.9}$$

It is clear, from the above formulation, that the squared error score is identical for both the Poisson and the negative binomial distribution since the conditional means are equal. Note that the normalized square error score is formed by the so called Pearson residuals obtained after fitting a model. We will compare all those scores in the next sections.

4.4 Simulations

In the following, we present a limited simulation study where we examine the predictive properties of models (3.6), (3.10) and (3.11). All simulations are based on 1000 runs and the data have been generated according to the negative binomial model (3.3) with mean specified by (3.4) and $\nu = 2$. We consider sample sizes of $n = 200$ and $n = 1000$, throughout the simulations.

4.4.1 Models With and Without Feedback Mechanism

To show that in fact models with a feedback mechanism are more parsimonious than models without a feedback mechanism, in terms of prediction, consider models (3.6) and (3.10), for instance. We will use the scoring rules, as outlined in Section 4.3.3, to measure their predictive power.

Suppose that we sample data from the negative binomial distribution (3.3) with $\nu = 2$ and the mean process λ_t is modeled by means of (3.10) with $d = 1,\ b_1 = 0.01,\ b_2 = 0.05,\ b_3 = 0.01,\ b_4 = 0.2,\ b_5 = 0.3$. For this example, we fit models (3.6) and (3.10) following the methodology outlined in Section 3.3 of Chapter 3. After obtaining the QMLE, we calculate all scoring rules based on negative binomial probabilistic prediction. Table 4.1 shows the results of the scoring rules for the two models. All rules point out to the adequacy of parsimonious model (3.6); the difference between the normalized squared error scores of models (3.6) and (3.10) is not significant. This simple exercise shows that the inclusion of the feedback mechanism in (3.6) takes into account dependence more properly, even though the data have been generated by (3.10). Hence, model (3.6) seems more suitable to fit data that show slowly decaying sample autocorrelation function and additionally offers a parsimonious parametrization.

		Scoring Rules						
	n	*logs*	*qs*	*sphs*	*rps*	*dss*	*nses*	*ses*
Model (3.6)	200	**1.871**	**-0.188**	**-0.434**	**1.056**	**2.521**	1.002	**4.620**
Model (3.10)		2.060	-0.165	-0.420	1.120	2.648	**0.998**	5.115
Model (3.6)	1000	**1.999**	**-0.178**	**-0.423**	**1.291**	**2.714**	1.004	**6.246**
Model (3.10)		2.352	-0.087	-0.365	1.412	2.979	**0.999**	7.492

Table 4.1: Mean scores for models (3.6) and (3.10). The data have been generated by model (3.10) with true values $d = 1,\ b_1 = 0.01,\ b_2 = 0.05,\ b_3 = 0.01,\ b_4 = 0.2,\ b_5 = 0.3$ and $\nu = 2$. Models are compared by the mean logarithmic, quadratic, spherical, ranked probability, Dawid-Sebastiani, normalized squared error and squared error scores. Bold face numbers in each column indicate the minimum value obtained between the two models.

4.4.2 Linear and Nonlinear Models

We generate now data according to the linear model (3.6) and the nonlinear model (3.11). For the linear model (3.6), data are generated with true values $(d, a_1, b_1) = (0.5, 0.4, 0.5)$ from (3.3) with $\nu = 2$. For each simulation we divide the data into two sets; a training data set and a test data set. The training data set is used to fit the model and to estimate the unknown parameters, while the test data set is employed for prediction. The training data set consists of the first 65%, 75% or 85% of the observations of the full data collection and the remaining data points form the test data set for each split. We also use the whole observed time series to study the in-sample performance of the predictor. Calculation of the maximum likelihood estimators, regardless wether we split the data or not, is carried out by applying the quasi-likelihood methodology as outlined in Section 3.3 of Chapter 3. After obtaining the QMLE $\hat{\boldsymbol{\theta}} = (\hat{d}, \hat{a}_1, \hat{b}_1)$, and using the fact that $\mathrm{E}(Y_t \mid \mathcal{F}_{t-1}^{Y,\lambda}) = \lambda_t$, we predict Y_t from $\hat{\lambda}_t = \lambda_t(\hat{\boldsymbol{\theta}})$. We consider two different probabilistic forecasters whose predictive distribution is either the Poisson or the negative binomial. For each split of the data, both forecasters predict the response of the corresponding test data set. Thereafter, we calculate the mean score of six scoring rules given by (4.3) – (4.8). Then, we compute the pairwise difference between mean scores for negative binomial and Poisson for each of the numerical measures; in other words we compute the mean score obtained from the Poisson minus the mean score obtained from the negative binomial.

Since the true data generating process follows the negative binomial distribution, we expect that the forecaster whose predictive distribution is assumed to be the negative

binomial, will obtain smaller values for each mean score than those obtained from the forecast whose predictive distribution is Poisson. To verify this assertion, using the results from all simulations, we produce a table of proportions of positive differences for each split of the data. For purposes of comparison, we construct all the above results for the full data collection as well. The left tables of Table 4.2 show the results.

n=200

	Split Percentage			
Score	65%	75%	85%	Full data
logs	0.954	0.907	0.844	1.000
qs	0.687	0.689	0.704	0.686
sphs	0.671	0.660	0.700	0.680
rps	0.762	0.750	0.719	0.803
dss	0.828	0.759	0.686	1.000
nses	0.999	1.000	1.000	1.000

	Split Percentage			
Score	65%	75%	85%	Full data
logs	0.892	0.841	0.773	0.999
qs	0.637	0.656	0.647	0.611
sphs	0.618	0.629	0.603	0.607
rps	0.680	0.645	0.650	0.656
dss	0.734	0.699	0.609	0.998
nses	0.997	0.998	0.999	1.000

n=1000

	Split Percentage			
Score	65%	75%	85%	Full data
logs	1.000	1.000	1.000	1.000
qs	0.813	0.792	0.779	0.836
sphs	0.815	0.786	0.771	0.847
rps	0.938	0.936	0.928	0.961
dss	1.000	0.999	0.998	1.000
nses	1.000	1.000	1.000	1.000

	Split Percentage			
Score	65%	75%	85%	Full data
logs	1.000	0.999	1.000	1.000
qs	0.732	0.729	0.746	0.766
sphs	0.732	0.733	0.749	0.748
rps	0.894	0.857	0.860	0.936
dss	0.999	0.994	0.980	1.000
nses	1.000	1.000	1.000	1.000

Table 4.2: Proportions of the positive differences between the scoring rules obtained from the Poisson distribution and the scoring rules obtained from the negative binomial distribution. For the left tables, data are generated from the linear model (3.6) for $(d, a_1, b_1) = (0.5, 0.4, 0.5)$, $\nu = 2$ and for sample sizes $n = 200$ and $n = 1000$. For the right tables, data are generated from the nonlinear model (3.11) for the same set of parameters and setting $\gamma = 0.5$. Results are based on 1000 simulations.

Most of the proportions of these tables are approaching unity. This fact implies that for most of the simulation runs, the negative binomial forecast outperforms the corresponding Poisson prediction.

Note that both the quadratic and spherical scores (see (4.4) and (4.5), respectively) yield, in most of the cases, the smaller values for the obtained proportions. In addition, the logarithmic score (4.3), the Dawid-Sebastiani score (4.7) and the normalized squared error score (4.8) yield proportions which approximate unity, regardless of the splitting of the data. We conclude that these scores seem to be more appropriate for correctly identifying the true data generating process. Figure 4.1 shows the boxplots of all scores (4.3) – (4.8) for the two forecasters and for each split of the data for the case of linear model (3.6).

White boxplots correspond to the Poisson prediction; grey boxplots correspond to the negative binomial forecast. It is obvious that the negative binomial forecast is superior to the Poisson forecast for each case. In addition, Figure 4.1 indicates that the performance of all scoring rules does not depend upon the particular partition of the data set.

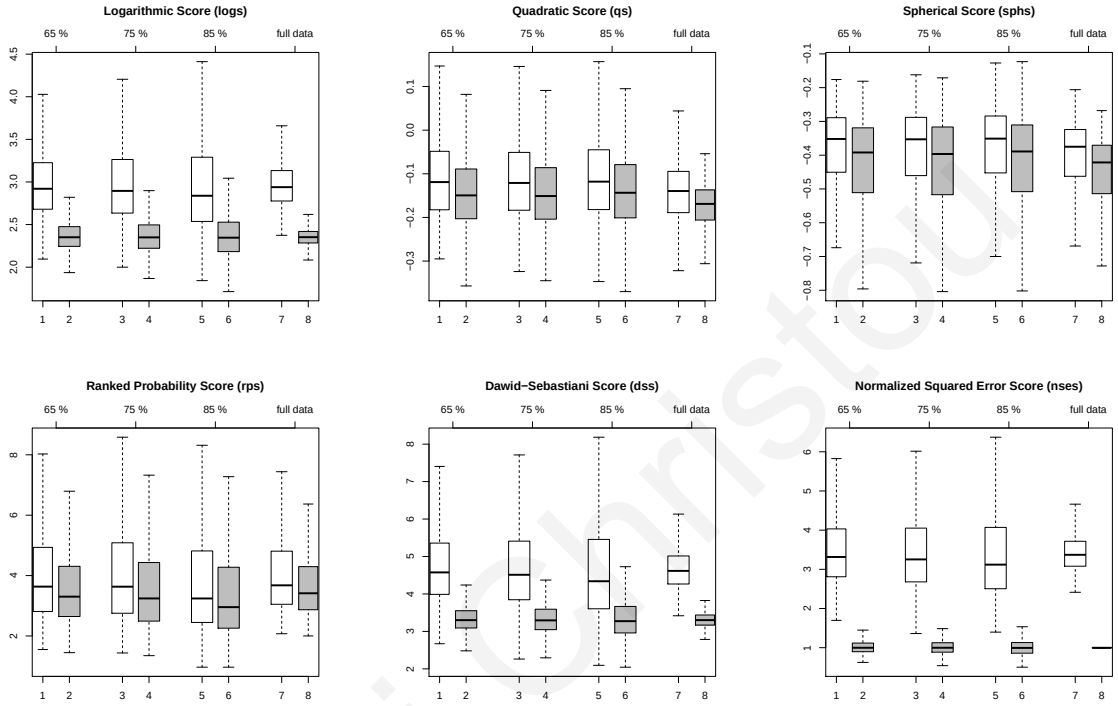

Figure 4.1: Boxplots for the mean scores given by (4.3) – (4.8). White plots are for the Poisson forecast, while grey plots are for the negative binomial prediction. Data are generated from the linear model (3.6) when the true values are $(d, a_1, b_1) = (0.5, 0.4, 0.5)$ and $\nu = 2$. The results are based on 1000 data points and 1000 simulations.

Table 4.3 reports the scoring rules calculated from the full data collection in this simulation experiment, where the two forecasters are compared by the mean scores given by (4.3) – (4.9). Scores from this table show clear preference towards the negative binomial over the Poisson prediction. Note that the predictive mean is equal (by model (3.5)) for both distributions and therefore the mean squared error score is the same for the two forecasts. Table 4.4 reports results when the true data generating process follows the Poisson distribution. As we note, all scoring rules are similar and therefore we suggest that it is preferable to employ the Poisson forecast when such phenomenon is observed.

			Scoring Rules						
	Forecaster	n	*logs*	*qs*	*sphs*	*rps*	*dss*	*nses*	*ses*
Linear Model (3.6)	Poisson	200	1.714	-0.267	-0.566	1.276	2.077	1.786	3.919
	NegBin		**1.599**	**-0.283**	**-0.571**	**1.183**	**1.819**	**0.985**	**3.919**
Nonlinear Model (3.11)	Poisson		1.425	-0.334	-0.624	0.938	1.488	1.554	2.303
	NegBin		**1.359**	**-0.345**	**-0.629**	**0.885**	**1.338**	**0.984**	**2.303**
Linear Model (3.6)	Poisson	1000	2.989	-0.139	-0.423	4.625	4.722	3.480	44.951
	NegBin		**2.354**	**-0.170**	**-0.469**	**3.958**	**3.307**	**0.997**	**44.951**
Nonlinear Model (3.11)	Poisson		2.234	-0.226	-0.529	2.676	3.167	2.466	19.181
	NegBin		**1.921**	**-0.250**	**-0.564**	**2.360**	**2.466**	**0.997**	**19.181**

Table 4.3: Scoring rules calculated for the linear model (3.6) and the nonlinear model (3.11) when data are generated from the negative binomial distribution with true values $(d, a_1, b_1) = (0.5, 0.4, 0.5)$, $\gamma = 0.5$, $\nu = 2$ and $n = 200$ or $n = 1000$. Results are based on 1000 simulations. For both models, the two forecasters are compared by the mean logarithmic, quadratic, spherical, ranked probability, Dawid-Sebastiani, normalized squared error and squared error scores. Bold face numbers in each column indicate the minimum value obtained between the two forecasters.

			Scoring Rules						
	Forecaster	n	*logs*	*qs*	*sphs*	*rps*	*dss*	*nses*	*ses*
Linear Model (3.6)	Poisson	200	2.123	-0.152	-0.414	2.302	2.463	0.991	4.954
	NegBin		2.122	-0.152	-0.415	2.290	2.462	0.947	4.954
Nonlinear Model (3.11)	Poisson		1.775	-0.224	-0.503	1.494	1.824	0.989	2.713
	NegBin		1.774	-0.224	-0.504	1.485	1.823	0.945	2.713
Linear Model (3.6)	Poisson	1000	2.128	-0.152	-0.420	2.441	2.476	0.998	4.991
	NegBin		2.128	-0.152	-0.420	2.436	2.475	0.980	4.991
Nonlinear Model (3.11)	Poisson		1.780	-0.228	-0.507	1.580	1.838	0.999	2.750
	NegBin		1.780	-0.228	-0.507	1.575	1.837	0.980	2.750

Table 4.4: Scoring rules calculated for the linear model (3.6) and the nonlinear model (3.11) when data are generated from the Poisson distribution with true values $(d, a_1, b_1) = (0.5, 0.4, 0.5)$, $\gamma = 0.5$ and $n = 200$ or $n = 1000$. Results are based on 1000 simulations. For both models, the two forecasters are compared by the mean logarithmic, quadratic, spherical, ranked probability, Dawid-Sebastiani, normalized squared error and squared error scores.

Furthermore, Figure 4.2 depicts the marginal calibration plot for both forecast distributions; recall (4.2). It is clear that the negative binomial is superior to the Poisson distribution. The solid line of the graph (which corresponds to the negative binomial forecast) illustrates small deviations from zero, as it should be expected. Similar conclusions hold for the nonlinear model (3.11). Using the same set of parameters as for the linear model and setting $\gamma = 0.5$, we verified empirically from the right tables of Table 4.2, Table 4.3, and Figure 4.3 the superiority of the negative binomial prediction.

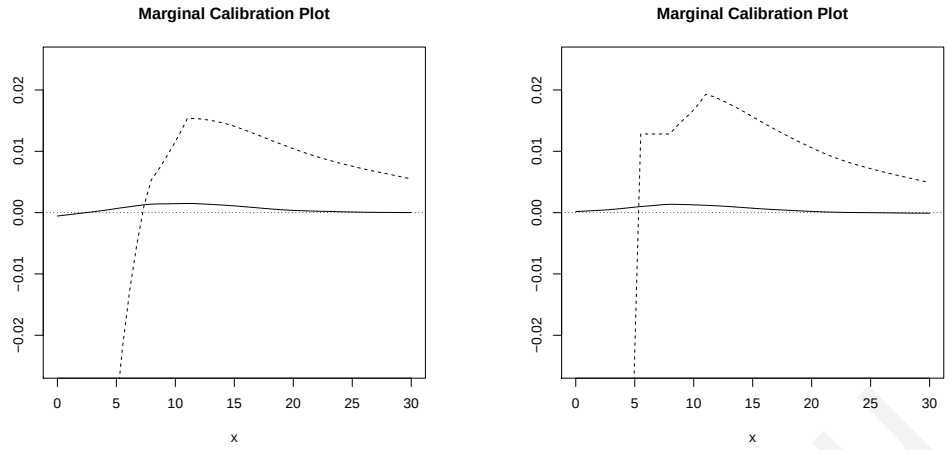

Figure 4.2: Left Plot: Marginal calibration plot for the linear model (3.6). Data are generated with true values $(d, a_1, b_1) = (0.5, 0.4, 0.5)$ and $\nu = 2$. Right Plot: Marginal calibration plot for the nonlinear model (3.11). Data are generated with true values $(d, a_1, \gamma, b_1) = (0.5, 0.4, 0.5, 0.5)$ and $\nu = 2$. The results are based on 1000 data points and 1000 simulations. Solid line corresponds to the negative binomial prediction, while dashed line is for the Poisson forecast.

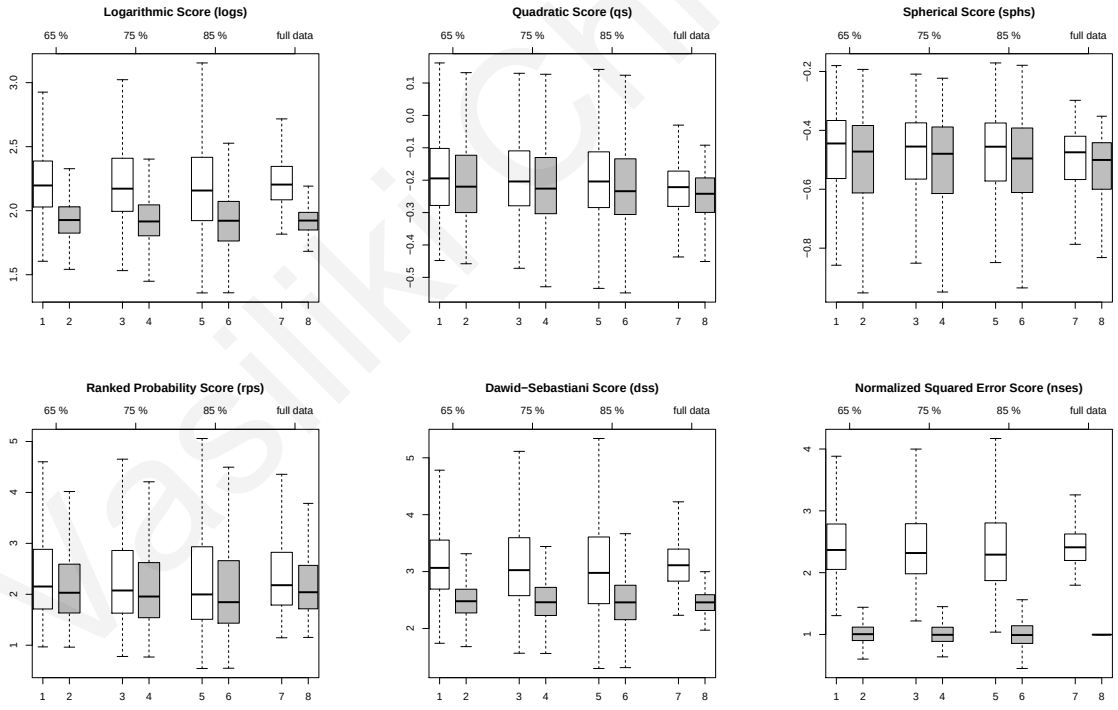

Figure 4.3: Boxplots for the mean scores given by (4.3) – (4.8). White plots correspond to the Poisson forecasts, while grey plots are for the negative binomial prediction. Data are generated from the nonlinear model (3.11) when the true values are $(d, a_1, \gamma, b_1) = (0.5, 0.4, 0.5, 0.5)$ and $\nu = 2$. The results are based on 1000 data points and 1000 simulations.

4.5 Case Studies

In order to examine whether or not negative binomial is a better forecaster than the Poisson predictor for some real data applications, we use the machinery developed in Section 4.3, namely the nonrandomized PIT histogram, the marginal calibration plot and the various scoring rules. We fit the linear model (3.6) and the nonlinear model (3.11) to the four different data collections introduced in Section 3.5 of Chapter 3.

4.5.1 Transactions Data

We fit the linear model (3.6) and the nonlinear model (3.11) to the data by using the quasi-likelihood methodology outlined in Section 3.3 of Chapter 3 and we obtain the QMLE for the regression parameters. The results are summarized in Table 3.7. Substituting these estimators to the expression of λ_t, we estimate the mean process. After obtaining $\hat{\lambda}_t$, we construct the PIT histograms, the marginal calibration plots and mean scores for the two models. The left plots of Figure 4.4 show the PIT histograms when the fit is based on Poisson distribution, for both linear and nonlinear models. Apparently, the plots show deviations from the Poisson distribution indicating underdispersed predictive distribution. The right plots indicate no apparent deviations from the uniformity; these plots are based on the negative binomial distribution.

Furthermore, to assess marginal calibration and sharpness of the prediction, we compute the scoring rules (4.3) – (4.9) and we construct the marginal calibration plot for these data. Table 4.5 shows the mean scores for the linear and nonlinear model. In addition, Figure 4.5 depicts the marginal calibration plots for the transactions data. The results in all cases indicate that the negative binomial distribution fits the data better than the Poisson distribution.

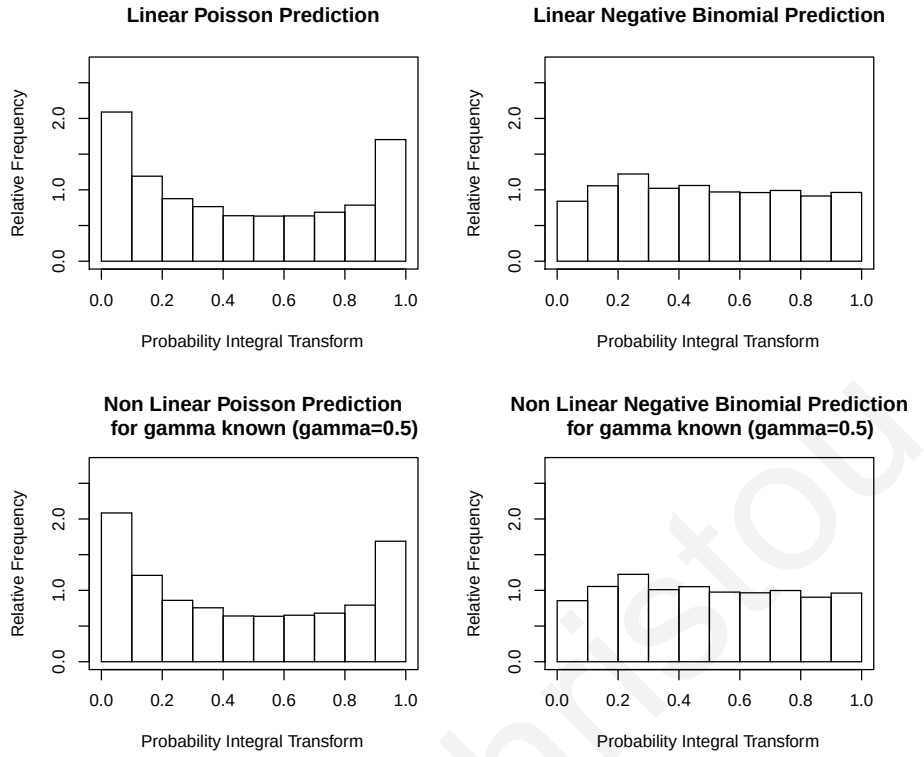

Figure 4.4: PIT histograms applied to the number of transactions per minute for the stock Ericsson B for the time period between July 2nd and July 22nd, 2002. From top to bottom: PIT histograms for model (3.6) and model (3.11) for $\gamma = 0.5$. Left Plots: The conditional distribution is the Poisson. Right Plots: The conditional distribution is the negative binomial.

		Scoring Rules						
	Forecaster	*logs*	*qs*	*sphs*	*rps*	*dss*	*nses*	*ses*
Linear Model (3.6)	Poisson	3.126	-0.076	-0.276	3.633	4.585	2.326	23.477
	NegBin	**2.902**	**-0.080**	**-0.292**	**3.284**	**4.112**	**0.993**	**23.477**
Nonlinear Model (3.11)	Poisson	3.123	-0.075	-0.274	3.605	4.579	2.318	23.435
	NegBin	**2.901**	**-0.080**	**-0.289**	**3.267**	**4.107**	**0.985**	**23.435**

Table 4.5: Scoring rules calculated for the transactions data after fitting the linear model (3.6) and the nonlinear model (3.11) for $\gamma = 0.5$. The two forecasters are compared by the mean logarithmic, quadratic, spherical, ranked probability, Dawid-Sebastiani, normalized squared error and squared error scores. Bold face numbers in each column indicate the minimum value obtained between the two forecasters.

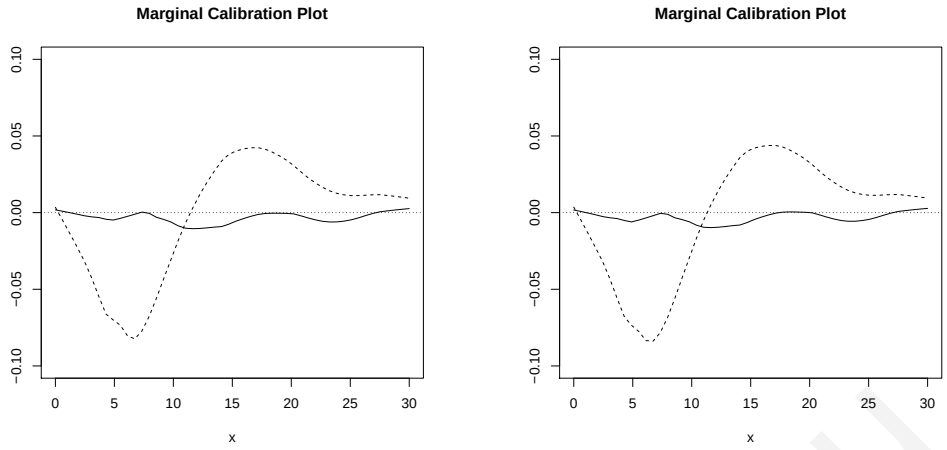

Figure 4.5: Left Plot: Marginal calibration plot for the transactions data if we fit the linear model (3.6). Right Plot: Marginal calibration plot for the transactions data if we fit the nonlinear model (3.11). Solid line corresponds to the negative binomial prediction, while dashed line is for the Poisson forecast.

4.5.2 Measles Data

Recall the measles data introduced in the previous chapter. We operate as in the previous example and following the same methodology, we obtain Figures 4.6 and 4.7 and Table 4.6. The data analysis in this case shows again that the negative binomial prediction is superior to the Poisson forecaster.

		Scoring Rules						
	Forecaster	*logs*	*qs*	*sphs*	*rps*	*dss*	*nses*	*ses*
Linear Model (3.6)	Poisson	3.920	-0.015	-0.189	11.435	6.242	3.677	78.844
	NegBin	**3.254**	**-0.033**	**-0.197**	**11.160**	**4.853**	**0.993**	**78.844**
Nonlinear Model (3.11)	Poisson	3.925	-0.014	-0.189	11.482	6.257	3.697	79.349
	NegBin	**3.255**	**-0.032**	**-0.196**	**11.213**	**4.854**	**0.993**	**79.349**

Table 4.6: Scoring rules calculated for the measles data after fitting the linear model (3.6) and the nonlinear model (3.11) for $\gamma = 0.5$. The two forecasters are compared by the mean logarithmic, quadratic, spherical, ranked probability, Dawid-Sebastiani, normalized squared error and squared error scores. Bold face numbers in each column indicate the minimum value obtained between the two forecasters.

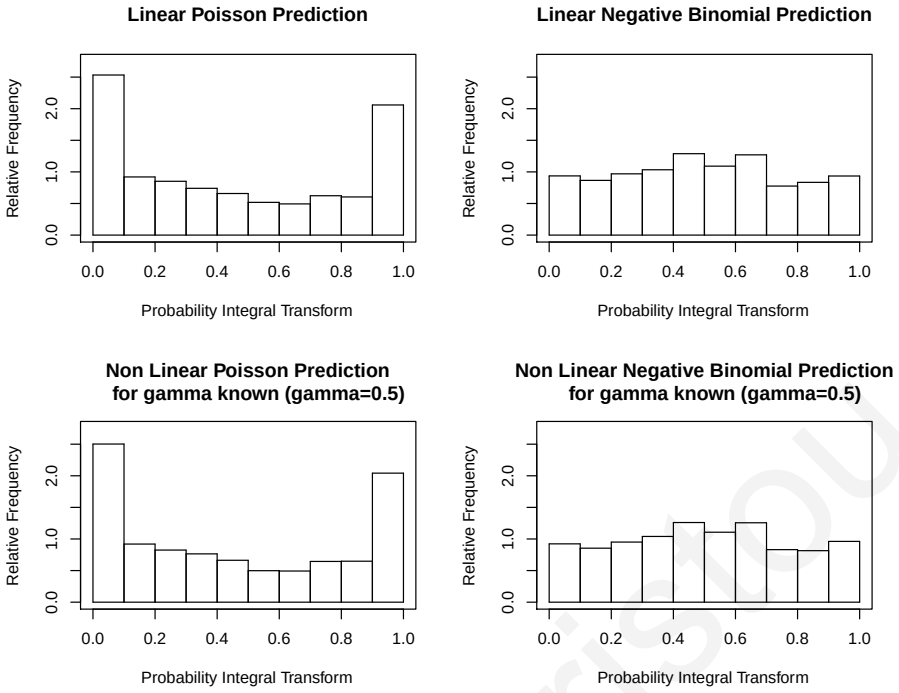

Figure 4.6: PIT histograms applied to the total number of measles in Sheffield for the time period between September 8th, 1978 and April 17th, 1987. From top to bottom: PIT histograms for the linear model (3.6) and the nonlinear model (3.11) for $\gamma = 0.5$. Left Plots: The conditional distribution is the Poisson. Right Plots: The conditional distribution is the negative binomial.

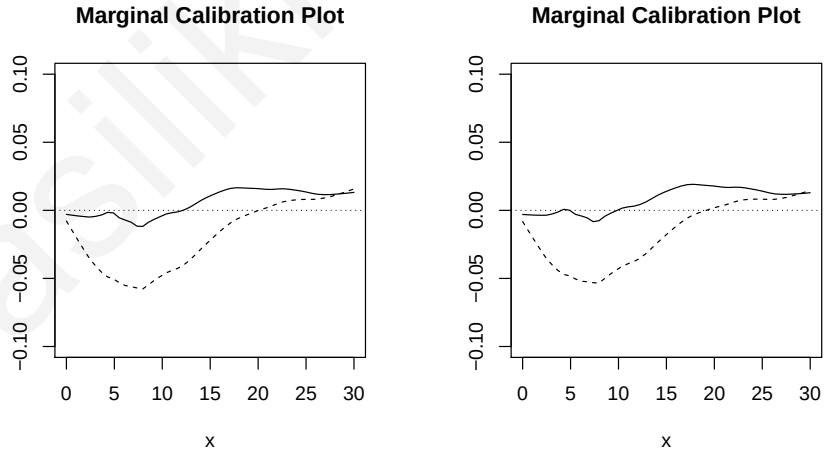

Figure 4.7: Left Plot: Marginal calibration plot for the measles data if we fit the linear model (3.6). Right Plot: Marginal calibration plot for the measles data if we fit the nonlinear model (3.11) for $\gamma = 0.5$. Solid line corresponds to the negative binomial prediction, while dashed line is for the Poisson forecast.

4.5.3 Breech Births Data

Consider now the third real data collection introduced in the previous chapter which corresponds to the number of monthly breech births in a hospital of South Africa. Following the same methodology as before, we obtain Figures 4.8 and 4.9 and Table 4.7. Once again, the results shows the superiority of the negative binomial forecast against the Poisson prediction.

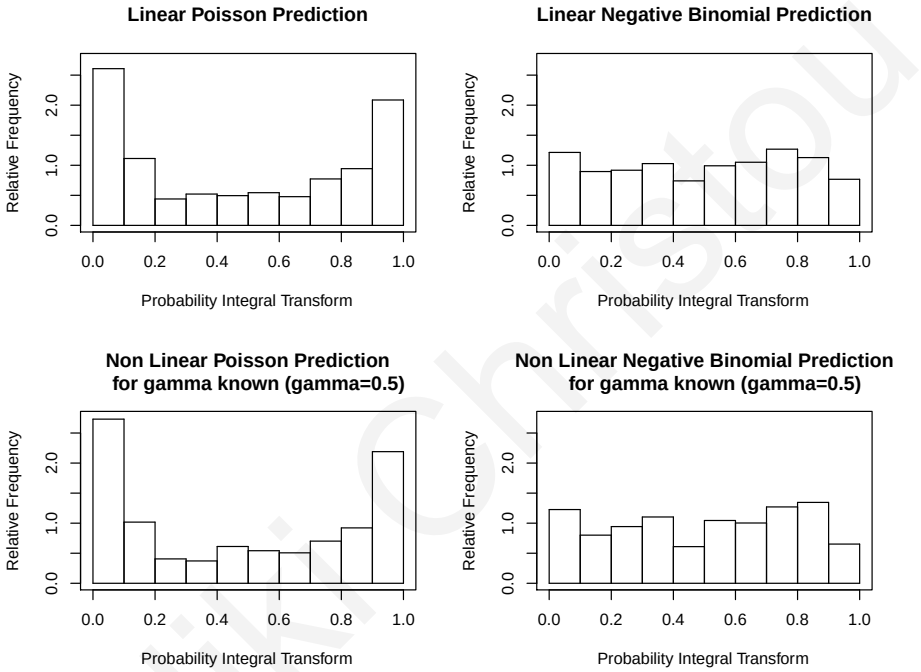

Figure 4.8: PIT histograms applied to the number of breech births in Edendale hospital from February 1977 to January 1986. From top to bottom: PIT histograms for the linear model (3.6) and the nonlinear model (3.11) for $\gamma = 0.5$. Left Plots: The conditional distribution is the Poisson. Right Plots: The conditional distribution is the negative binomial.

		Scoring Rules						
	Forecaster	*logs*	*qs*	*sphs*	*rps*	*dss*	*nses*	*ses*
Linear Model (3.6)	Poisson	3.970	-0.001	-0.151	4.754	6.109	3.208	57.232
	NegBin	**3.447**	**-0.030**	**-0.178**	**4.621**	**5.088**	**0.972**	**57.232**
Nonlinear Model (3.11)	Poisson	3.984	-0.001	-0.151	4.674	6.102	3.199	57.647
	NegBin	**3.452**	**-0.030**	**-0.177**	**4.579**	**5.080**	**0.972**	**57.647**

Table 4.7: Scoring rules calculated for the breech births data after fitting the linear model (3.6) and the nonlinear model (3.11) for $\gamma = 0.5$. The two forecasters are compared by the mean logarithmic, quadratic, spherical, ranked probability, Dawid-Sebastiani, normalized squared error and squared error scores. Bold face numbers in each column indicate the minimum value obtained between the two forecasters.

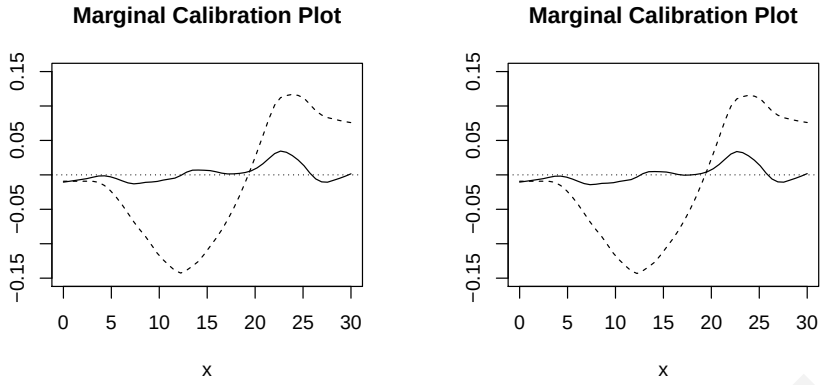

Figure 4.9: Left Plot: Marginal calibration plot for the breech births data if we fit the linear model (3.6). Right Plot: Marginal calibration plot for the births data if we fit the nonlinear model (3.11) for $\gamma = 0.5$. Solid line corresponds to the negative binomial prediction, while dashed line is for the Poisson forecast.

4.5.4 Homicides Data

Consider now the fourth real data collection introduced in Section 3.5 of Chapter 3 which corresponds to the number of deaths recorded at the Salt River state mortuary at Cape Town, for the period time 1986 - 1991. Following the same methodology as before, we obtain Figures 4.10 and 4.11 and Table 4.8. Once again, the results show the superiority of the negative binomial forecast against the Poisson prediction.

		Scoring Rules						
	Forecaster	*logs*	*qs*	*sphs*	*rps*	*dss*	*nses*	*ses*
Linear Model (3.6)	Poisson	2.026	-0.178	-0.422	1.297	2.547	1.649	4.671
	NegBin	**1.956**	**-0.188**	**-0.436**	**1.272**	**2.401**	**0.990**	**4.671**
Nonlinear Model (3.11)	Poisson	2.023	-0.180	-0.424	1.297	2.539	1.644	4.636
	NegBin	**1.955**	**-0.189**	**-0.438**	**1.271**	**2.395**	**0.990**	**4.636**

Table 4.8: Scoring rules calculated for the deaths data after fitting the linear model (3.6) and the nonlinear model (3.11) for $\gamma = 0.5$. The two forecasters are compared by the mean logarithmic, quadratic, spherical, ranked probability, Dawid-Sebastiani, normalized squared error and squared error scores. Bold face numbers in each column indicate the minimum value obtained between the two forecasters.

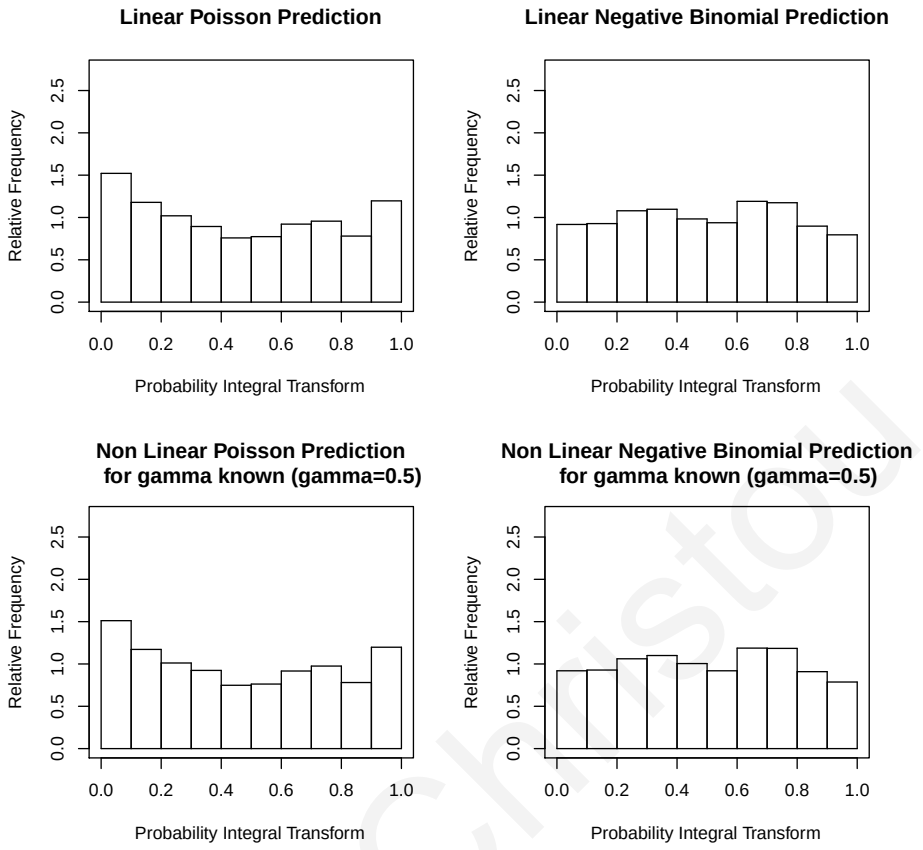

Figure 4.10: PIT histograms applied to the number of deaths recorded at the Salt River state mortuary at Cape Town, for the period time 1986 - 1991. From top to bottom: PIT histograms for the linear model (3.6) and the nonlinear model (3.11) for $\gamma = 0.5$. Left Plots: The conditional distribution is the Poisson. Right Plots: The conditional distribution is the negative binomial.

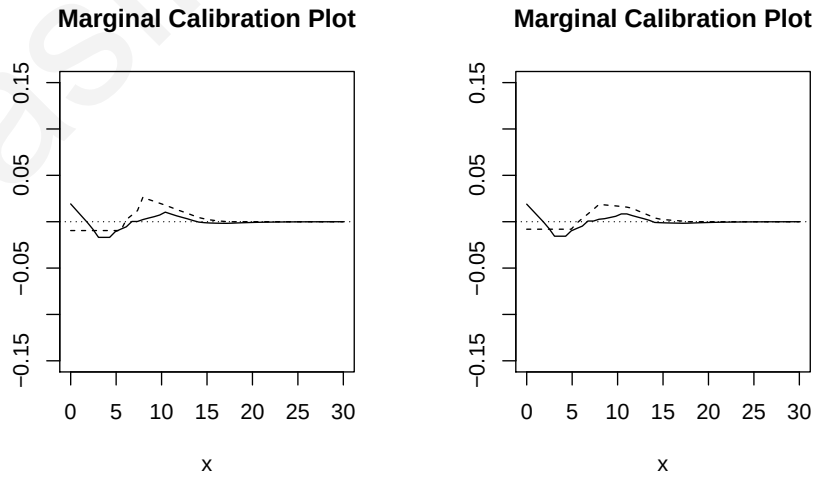

Figure 4.11: Left Plot: Marginal calibration plot for the deaths data if we fit the linear model (3.6). Right Plot: Marginal calibration plot for the births data if we fit the nonlinear model (3.11) for $\gamma = 0.5$. Solid line corresponds to the negative binomial prediction, while dashed line is for the Poisson forecast.

Appendix – R codes

R code for the construction of tables and figures of Chapter 4. Some basic functions are already reported in the Appendix of Chapter 3 and thus they are omitted.

```
####################
## Split the data ##
####################
q=c(0.65,0.75,0.85,1)


scoring.rules=function(d,a,b,size,sim,nu)
{
out=array(NA,dim=c(7,2*length(q),sim))
out1=array(NA,dim=c(7,length(q),sim))
out.data1=matrix(NA,sim,2*length(q))
out.data2=matrix(NA,sim,2*length(q))
out.data3=matrix(NA,sim,2*length(q))
out.data4=matrix(NA,sim,2*length(q))
out.data5=matrix(NA,sim,2*length(q))
out.data6=matrix(NA,sim,2*length(q))
out.data7=matrix(NA,sim,2*length(q))
x=seq(0,30,length=50)
G=rep(NA,length(x))
F1=rep(NA,length(x))
F2=rep(NA,length(x))
dif1=array(NA,dim=c(length(x),length(q),sim))
dif2=array(NA,dim=c(length(x),length(q),sim))
for (i in 1:sim)
{
ydata=linearnegbin.ts(d,a,b,size,nu)[501:size,1]
for(j in 1:length(q))
{
if (q[j]!=1)
```

```
{
# Training and test set
ydata.training=ydata[1:floor(length(ydata)*q[j])]
ydata.test=ydata[(length(ydata.training)+1):length(ydata)]
}
else
{
ydata.training=ydata
ydata.test=ydata
}
# Fit the linear model
# Estimation of d, a, b
epsilon=0.001
arma_fit=arima(ydata.training,order=c(1,0,1),method="CSS")
ma_1=min(max(arma_fit$coef["ma1"],-1+epsilon),0-epsilon)
ar_1=min(max(arma_fit$coef["ar1"],0+epsilon-ma_1),1-epsilon)
sigma_sq =max(arma_fit$sigma2,epsilon)
theta_init=c(sigma_sq*(1-ar_1),-ma_1,ar_1+ma_1)
results=constrOptim(theta=theta_init,f=liklinear.poisson,
        grad=scorelinear.poisson,data=ydata.training,ui=uilinear,
        ci=cilinear,outer.iterations=100,outer.eps=1e-05,method="BFGS")
results1=results$par
hat.d=results1[1]
hat.a=results1[2]
hat.b=results1[3]
# Calculation of \hat{\lambda}
lambda=rep(NA, length(ydata))
lambda[1]=mean(ydata)
for (t in 2:length(ydata))
{
lambda[t]=hat.d+hat.a*lambda[t-1]+hat.b*ydata[t-1]
```

```
}
if (q[j]!=1)
{
lambda.training=lambda[1:length(ydata.training)]
lambda.test=lambda[(length(ydata.training)+1):length(ydata)]
}
else
{
lambda.training=lambda
lambda.test=lambda
}
# Estimation of \nu
library(MASS)
est.nu1=(mean(((ydata.training-lambda.training)^{2}-lambda.training)/
        (lambda.training^{2})))^{-1}
est.nu2=theta.mm(ydata.training,lambda.training,length(ydata.training)
       -length(results1))
# Calculate scoring rules
# Parameter settings for computing scores
kk =100000     #cut-off for summations
my.k=(0:kk)-1  #to handle ranked probability score
n=length(ydata.test)
# Poisson prediction
pois.Px=ppois(ydata.test,lambda.test)
pois.Px1=ppois(ydata.test-1,lambda.test)
pois.px=dpois(ydata.test,lambda.test)
# NB prediction
nb.Px=pnbinom(ydata.test,size=est.nu2,mu=lambda.test)
nb.Px1=pnbinom(ydata.test-1,size=est.nu2,mu=lambda.test)
nb.px=dnbinom(ydata.test,size=est.nu2,mu=lambda.test)
# Compute scores
```

```
pois.logs= -log(pois.px)
 pois.norm=sum(dpois(my.k,lambda.test)^2)
pois.qs= -2*pois.px+pois.norm
pois.sphs= -pois.px/sqrt(pois.norm)
 i.cumsum=cumsum(ppois(my.k,lambda.test)^2)
 ii.sum=sum((ppois(my.k,lambda.test)-1)^2)
 ii.cumsum=cumsum((ppois(my.k,lambda.test)-1)^2)
pois.rps=(i.cumsum[ydata.test+1]+ii.sum-ii.cumsum[ydata.test+1])
pois.dss=(ydata.test-lambda.test)^2/lambda.test+log(lambda.test)
pois.nses=(ydata.test-lambda.test)^2/lambda.test
pois.ses=(ydata.test-lambda.test)^2
nb.logs= -log(nb.px)
 nb.norm=sum(dnbinom(my.k,mu=lambda.test,size=est.nu2)^2)
nb.qs= -2*nb.px+nb.norm
nb.sphs= -nb.px/sqrt(nb.norm)
 i.cumsum=cumsum(pnbinom(my.k,mu=lambda.test,size=est.nu2)^2)
 ii.sum=sum((pnbinom(my.k,mu=lambda.test,size=est.nu2)-1)^2)
 ii.cumsum=cumsum((pnbinom(my.k,mu=lambda.test,size=est.nu2)-1)^2)
nb.rps=(i.cumsum[ydata.test+1]+ii.sum-ii.cumsum[ydata.test+1])
nb.dss=(ydata.test-lambda.test)^2/(lambda.test*(1+lambda.test/est.nu2))
      +log(lambda.test*(1+lambda.test/est.nu2))
nb.nses=(ydata.test-lambda.test)^2/(lambda.test*(1+lambda.test/est.nu2))
nb.ses=(ydata.test-lambda.test)^2
# Scoring rules
out[1,(2*j-1):(2*j),i]=round(c(mean(pois.logs),mean(nb.logs)),3)
out[2,(2*j-1):(2*j),i]=round(c(mean(pois.qs),mean(nb.qs)),3)
out[3,(2*j-1):(2*j),i]=round(c(mean(pois.sphs),mean(nb.sphs)),3)
out[4,(2*j-1):(2*j),i]=round(c(mean(pois.rps),mean(nb.rps)),3)
out[5,(2*j-1):(2*j),i]=round(c(mean(pois.dss),mean(nb.dss)),3)
out[6,(2*j-1):(2*j),i]=round(c(mean(pois.nses),mean(nb.nses)),3)
out[7,(2*j-1):(2*j),i]=round(c(mean(pois.ses),mean(nb.ses)),3)
```

```
out1[1,j,i]=ifelse(out[1,(2*j-1),i]-out[1,(2*j),i]>0,1,0)
out1[2,j,i]=ifelse(out[2,(2*j-1),i]-out[2,(2*j),i]>0,1,0)
out1[3,j,i]=ifelse(out[3,(2*j-1),i]-out[3,(2*j),i]>0,1,0)
out1[4,j,i]=ifelse(out[4,(2*j-1),i]-out[4,(2*j),i]>0,1,0)
out1[5,j,i]=ifelse(out[5,(2*j-1),i]-out[5,(2*j),i]>0,1,0)
out1[6,j,i]=ifelse(out[6,(2*j-1),i]-out[6,(2*j),i]>0,1,0)
out1[7,j,i]=out[7,(2*j-1),i]-out[7,(2*j),i]
for(k in 1:(2*length(q)))
{
out.data1[,k]=out[1,k,]
out.data2[,k]=out[2,k,]
out.data3[,k]=out[3,k,]
out.data4[,k]=out[4,k,]
out.data5[,k]=out[5,k,]
out.data6[,k]=out[6,k,]
out.data7[,k]=out[7,k,]
}
indic=rep(NA,length(ydata.test))
# Obtain the marginal calibration plot
for(l in 1:length(x))
{
indic=as.numeric(ydata.test<=x[l])
G[l]=mean(indic)
pois.px.new=ppois(x[l],lambda.test)                    #Poisson prediction
nb.px.new=pnbinom(x[l],size=est.nu2,mu=lambda.test)  #NegBin prediction
F1[l]=mean(pois.px.new)
F2[l]=mean(nb.px.new)
}
dif1[,j,i]=F1-G
dif2[,j,i]=F2-G
}
```

```
}
mean.scoring.rules=round(apply(out1,1:2,mean),2)
mean.dif1=apply(dif1,1:2,mean)
mean.dif2=apply(dif2,1:2,mean)
return(list(out,out1,mean.scoring.rules,out.data1,out.data2,out.data3,
       out.data4,out.data5,out.data6,out.data7,mean.dif1,mean.dif2))
}

test=scoring.rules(0.5,0.4,0.5,1500,1000,2)
```

Construction of the left panel of Table 4.2, first and third panels of Table 4.3, Figure 4.1 and left plot of Figure 4.2.

```
test[[3]]

#######################################################################
## Compare the scoring rules for Poisson and the negative binomial ##
## prediction using the full data                                    ##
#######################################################################
mat=matrix(NA,2,7)
for(l in 1:7)
{
mat[,l]=apply(test[[(l+3)]][,7:8],2,mean)
}
mat

##############
## Boxplots ##
##############
par(mfrow=c(2,3))
library(fields)
at=c(1.5,4.5,7.5,10.5)
```

```
pos=c(1:2,4:5,7:8,10:11)
labels=c("65 %","75 %","85 %","full data")
bplot(test[[4]],pos=pos,width=c(0.45,0.5,0.5,0.5,0.5,0.5,0.5,0.45),
      col=c("white","grey"),outline=FALSE)
axis(3,at=at,labels=labels)
title(main="Logarithmic Score (logs)",line=3)
bplot(test[[5]],pos=pos,width=c(0.45,0.5,0.5,0.5,0.5,0.5,0.5,0.45),
      col=c("white","grey"),outline=FALSE)
axis(3,at=at,labels=labels)
title(main="Quadratic Score (qs)",line=3)
bplot(test[[6]],pos=pos,width=c(0.45,0.5,0.5,0.5,0.5,0.5,0.5,0.45),
      col=c("white","grey"),outline=FALSE)
axis(3,at=at,labels=labels)
title(main="Spherical Score (sphs)",line=3)
bplot(test[[7]],pos=pos,width=c(0.45,0.5,0.5,0.5,0.5,0.5,0.5,0.45),
      col=c("white","grey"),outline=FALSE)
axis(3,at=at,labels=labels)
title(main="Ranked Probability Score (rps)",line=3)
bplot(test[[8]],pos=pos,width=c(0.45,0.5,0.5,0.5,0.5,0.5,0.5,0.45),
      col=c("white","grey"),outline=FALSE)
axis(3,at=at,labels=labels)
title(main="Dawid-Sebastiani Score (dss)",line=3)
bplot(test[[9]],pos=pos,width=c(0.45,0.5,0.5,0.5,0.5,0.5,0.5,0.45),
      col=c("white","grey"),outline=FALSE)
axis(3,at=at,labels=labels)
title(main="Normalized Squared Error Score (nses)",line=3)

################################################
## Marginal calibration plot for the full data ##
################################################
x=seq(0,30,length=50)
```

```
plot(x,test[[11]][,4],ylim=c(-0.015,0.015),type="n",
     main="Marginal Calibration Plot",ylab="")
lines(lowess(x,test[[11]][,4],f=0.2),lty=2)
lines(lowess(x,test[[12]][,4],f=0.2))
abline(h=0,lty=3)
```

For the construction of Table 4.4 we use the function `linearpoisson.ts` shown below, in order to generate data from the linear model (3.6) using the Poisson distribution. We calculate the scoring rules using the function `scoring.rules` shown above, but now we delete the argument nu, since we do not generate data from the negative binomial distribution.

```
############################################################
## Simulate the linear model using Poisson distribution ##
############################################################
linearpoisson.ts=function(d,a,b,size)
{
y=rep(NA,size)
mu=rep(NA,size)
mu[1]=1
y[1]=rpois(1,mu[1])
for (t in 2:size)
{
mu[t]=d+a*mu[t-1]+b*y[t-1]
y[t]=rpois(1,mu[t])
}
return(cbind(y,mu))
}
```

For the construction of Table 4.1 we generate data from the linear model (3.10) using the negative binomial distribution. The following functions are used in order to calculate the scoring rules.

```
##############################################
## Simulate the linear model (3.10)          ##
## using the negative binomial distribution ##
##############################################
linearnegbin1.ts =function(d,b1,b2,b3,b4,b5,nu,size)
{
y=rep(NA,size)
mu=rep(NA,size)
theta=rep(NA,size)
mu[1:5]=1 #initial value
y[1:5]=rnbinom(1,size=nu,mu=mu[1:5])
for (t in 6:size)
{
mu[t]=d+b1*y[t-1]+b2*y[t-2]+b3*y[t-3]+b4*y[t-4]+b5*y[t-5]
y[t]=rnbinom(1,size=nu,mu=mu[t])
}
return(cbind(y,mu))
}


############################################################
## (Poisson) Quasi-likelihood for the linear model (3.10) ##
############################################################
liklinear.poisson1=function(theta, data)
{
lambda=rep(NA,length(data))
loglik=rep(NA,length(data))
lambda[1:5]=0
loglik[1:5]=0
for (t in 6:length(data))
{
lambda[t]=theta[1]+theta[2]*data[t-1]+theta[3]*data[t-2]
```

```
        +theta[4]*data[t-3]+theta[5]*data[t-4]+theta[6]*data[t-5]
if (lambda[t] <=0) loglik[t]= 0 else
if (lambda[t] >0)  loglik[t]= -data[t]*log(lambda[t])+lambda[t]
}
final=sum(loglik)
}

#############################################################
## (Poisson) Score function for the linear model (3.10) ##
#############################################################
scorelinear.poisson1=function(theta,data)
{
lambda=rep(NA,length(data))
lambda[1]=0
first=rep(NA,length(data))
first[1]=0
second=rep(NA,length(data))
second[1]=0
third=rep(NA,length(data))
third[1]=0
fourth=rep(NA,length(data))
fourth[1]=0
fifth=rep(NA,length(data))
fifth[1]=0
sixth=rep(NA,length(data))
sixth[1]=0
s1=rep(NA,length(data))
s2=rep(NA,length(data))
s3=rep(NA,length(data))
s4=rep(NA,length(data))
s5=rep(NA,length(data))
```

```
s6=rep(NA,length(data))
for (t in 6:length(data))
{
lambda[t]=theta[1]+theta[2]*data[t-1]+theta[3]*data[t-2]
         +theta[4]*data[t-3]+theta[5]*data[t-4]+theta[6]*data[t-5]
first[t]= 1
second[t]=data[t-1]
third[t]=data[t-2]
fourth[t]=data[t-3]
fifth[t]=data[t-4]
sixth[t]=data[t-5]
s1[t]=-((data[t]/lambda[t])-1)*first[t]
s2[t]=-((data[t]/lambda[t])-1)*second[t]
s3[t]=-((data[t]/lambda[t])-1)*third[t]
s4[t]=-((data[t]/lambda[t])-1)*fourth[t]
s5[t]=-((data[t]/lambda[t])-1)*fifth[t]
s6[t]=-((data[t]/lambda[t])-1)*sixth[t]
}
ss1=sum(s1[-1])
ss2=sum(s2[-1])
ss3=sum(s3[-1])
ss4=sum(s4[-1])
ss5=sum(s5[-1])
ss6=sum(s6[-1])
score=c(ss1,ss2,ss3,ss4,ss5,ss6)
}

####################################
## Constrains to obtain the QMLE ##
## for model (3.10)               ##
## d,b_1,b_2,b_3,b_4,b_5>0,       ##
```

```
## 0<b_1+b_2+b_3+b_4+b_5<1       ##
###################################
uilinear1=matrix(0,nrow=7,ncol=6) #matrix for the linear constraints
uilinear1[1,1]=1
uilinear1[2,2]=1
uilinear1[3,3]=1
uilinear1[4,4]=1
uilinear1[5,5]=1
uilinear1[6,6]=1
uilinear1[7,2]=-1
uilinear1[7,3]=-1
uilinear1[7,4]=-1
uilinear1[7,5]=-1
uilinear1[7,6]=-1
cilinear1=rep(0,7) #constant vector for the linear constraints
cilinear1[7]=-1

#############################################################
## (Poisson) Quasi-likelihood for the linear model (3.6) ##
#############################################################
liklinear.poisson2=function(theta,data)
{
lambda=rep(NA,length(data))
loglik=rep(NA,length(data))
lambda[1]=0
loglik[1]=0
for (t in 2:length(data))
{
lambda[t]=theta[1]+theta[2]*lambda[t-1]+theta[3]*data[t-1]
if (lambda[t]<=0) loglik[t]= 0 else
if (lambda[t]>0)  loglik[t]= -data[t]*log(lambda[t])+lambda[t]
```

```
}
final=sum(loglik)
}

##########################################################
## (Poisson) Score function for the linear model (3.6) ##
##########################################################
scorelinear.poisson2=function(theta,data)
{
lambda=rep(NA,length(data))
lambda[1]=0
first=rep(NA,length(data))
first[1]=0
second=rep(NA,length(data))
second[1]=0
third=rep(NA,length(data))
third[1]=0
s1=rep(NA,length(data))
s2=rep(NA,length(data))
s3=rep(NA,length(data))
for (t in 2:length(data))
{
lambda[t]=theta[1]+theta[2]*lambda[t-1]+theta[3]*data[t-1]
first[t]=(1+theta[2]*first[t-1])
second[t]=(lambda[t-1]+theta[2]*second[t-1])
third[t]=(data[t-1]+theta[2]*third[t-1])
s1[t]=-((data[t]/lambda[t])-1)*first[t]
s2[t]=-((data[t]/lambda[t])-1)*second[t]
s3[t]=-((data[t]/lambda[t])-1)*third[t]
}
ss1=sum(s1[-1])
```

```
ss2=sum(s2[-1])
ss3=sum(s3[-1])
score=c(ss1,ss2,ss3)
}

###################################
## Constrains to obtain the QMLE ##
## for model (3.6)               ##
## d>0, a>0, b>0, 0<a+b<1        ##
###################################
uilinear2=matrix(0,nrow=4,ncol=3) #matrix for the linear constraints
uilinear2[1,1]=1
uilinear2[2,2]=1
uilinear2[3,3]=1
uilinear2[4,2]=-1
uilinear2[4,3]=-1
cilinear2=rep(0,4) #constant vector for the linear constraints
cilinear2[4]=-1

parsimonious.model=function(d,b1,b2,b3,b4,b5,nu,size,sim,epsilon=0.001)
{
results1=matrix(NA,nrow=sim,ncol=6)
results2=matrix(NA,nrow=sim,ncol=3)
out=array(NA, dim=c(7,2,sim))
for (i in 1:sim)
{
ydata1=linearnegbin1.ts(d,b1,b2,b3,b4,b5,nu,size)[501:size,1]
# Fit the linear model (3.10)
# Estimation of d, b1, b2, b3, b4, b5
arma_fit1=arima(ydata1,order=c(0,0,5),method="CSS")
b.coef1=arma_fit1$coef[1:5]
```

```
mean=arma_fit1$coef[6]
d.coef=mean*(1-sum(b.coef1))
theta_in=c(max(epsilon,d.coef),max(epsilon,b.coef1[1]),
         max(epsilon,b.coef1[2]),max(epsilon,b.coef1[3]),
         max(epsilon,b.coef1[4]),max(epsilon,b.coef1[5]))
results=constrOptim(theta=theta_in,f=liklinear.poisson1,
        grad=scorelinear.poisson1,data=ydata1,ui=uilinear1,
        ci=cilinear1,outer.iterations=100,outer.eps=1e-05,method="BFGS")
results1[i,]=results$par
hat.d1=results1[i,1]
hat.b1=results1[i,2]
hat.b2=results1[i,3]
hat.b3=results1[i,4]
hat.b4=results1[i,5]
hat.b5=results1[i,6]
# Calculation of \hat{\lambda}
lambda1=rep(NA, length(ydata1))
lambda1[1:5]=mean(ydata1)
for (t in 6:length(ydata1))
{
 lambda1[t]=hat.d1+hat.b1*ydata1[t-1]+hat.b2*ydata1[t-2]
          +hat.b3*ydata1[t-3]+hat.b4*ydata1[t-4]+hat.b5*ydata1[t-5]
}
# Estimation of \nu
est1.nu1=(mean(((ydata1-lambda1)^{2}-lambda1)/(lambda1^{2})))^{-1}
est1.nu2=theta.mm(ydata1,lambda1,length(ydata1)-ncol(results1))
# NB prediction
nb1.Px=pnbinom(ydata1,size=est1.nu1,mu=lambda1)
nb1.Px1=pnbinom(ydata1-1,size=est1.nu1,mu=lambda1)
nb1.px=dnbinom(ydata1,size=est1.nu1,mu=lambda1)
```

```
# Calculate scoring rules
# Parameter settings for computing scores
kk=100000
my.k=(0:kk)-1
n=length(ydata1)
# Compute scores
nb1.logs= -log(nb1.px)
 nb1.norm=sum(dnbinom(my.k,mu=lambda1,size=est1.nu1)^2)
nb1.qs= -2*nb1.px+nb1.norm
nb1.sphs= -nb1.px/sqrt(nb1.norm)
 i.cumsum1=cumsum(pnbinom(my.k,mu=lambda1,size=est1.nu1)^2)
 ii.sum1=sum((pnbinom(my.k,mu=lambda1,size=est1.nu1)-1)^2)
 ii.cumsum1=cumsum((pnbinom(my.k,mu=lambda1,size=est1.nu1)-1)^2)
nb1.rps=(i.cumsum1[ydata1+1]+ii.sum1-ii.cumsum1[ydata1+1])
nb1.dss=(ydata1-lambda1)^2/(lambda1*(1+lambda1/est1.nu1))
       +log(lambda1*(1+lambda1/est1.nu1))
nb1.nses=(ydata1-lambda1)^2/(lambda1*(1+lambda1/est1.nu1))
nb1.ses=(ydata1-lambda1)^2
# Fit the linear model (3.6)
# using the data we generated from the linear model (3.10)
# Estimation of d, a, b
arma_fit2=arima(ydata1,order=c(1,0,1),method="CSS")
ma_1=min(max(arma_fit2$coef["ma1"],-1+epsilon),0-epsilon)
ar_1=min(max(arma_fit2$coef["ar1"],0+epsilon-ma_1),1-epsilon)
sigma_sq=max(arma_fit2$sigma2,epsilon)
theta_init=c(sigma_sq*(1-ar_1),-ma_1,ar_1+ma_1)
results=constrOptim(theta=theta_init,f=liklinear.poisson2,
        grad=scorelinear.poisson2,data=ydata1,ui=uilinear2,
        ci=cilinear2,outer.iterations=100,outer.eps=1e-05,method="BFGS")
results2[i,]=results$par
hat.d2=results2[i,1]
```

```
hat.a=results2[i,2]
hat.b=results2[i,3]
# Calculation of \hat{\lambda}
lambda2=rep(NA, length(ydata1))
lambda2[1]=mean(ydata1)
for (t in 2:length(ydata1))
{
lambda2[t]=hat.d2+hat.a*lambda2[t-1]+hat.b*ydata1[t-1]
}
# Estimation of \nu
library(MASS)
est2.nu1=(mean(((ydata1-lambda2)^{2}-lambda2)/(lambda2^{2})))^{-1}
est2.nu2=theta.mm(ydata1,lambda2,length(ydata1)-ncol(results2))
# NB prediction
nb2.Px=pnbinom(ydata1,size=est2.nu1,mu=lambda2)
nb2.Px1=pnbinom(ydata1-1,size=est2.nu1,mu=lambda2)
nb2.px=dnbinom(ydata1,size=est2.nu1,mu=lambda2)
# Calculate scoring rules
# Parameter settings for computing scores
kk=100000
my.k=(0:kk)-1
n=length(ydata1)
# Compute scores
nb2.logs= -log(nb2.px)
 nb2.norm=sum(dnbinom(my.k,mu=lambda2,size=est2.nu1)^2)
nb2.qs= -2*nb2.px+nb2.norm
nb2.sphs= -nb2.px/sqrt(nb2.norm)
 i.cumsum2=cumsum(pnbinom(my.k,mu=lambda2,size=est2.nu1)^2)
 ii.sum2=sum((pnbinom(my.k,mu=lambda2,size=est2.nu1)-1)^2)
 ii.cumsum2=cumsum((pnbinom(my.k,mu=lambda2,size=est2.nu1)-1)^2)
nb2.rps=(i.cumsum2[ydata1+1]+ii.sum2-ii.cumsum2[ydata1+1])
```

```
nb2.dss=(ydata1-lambda2)^2/(lambda2*(1+lambda2/est2.nu1))
       +log(lambda2*(1+lambda2/est2.nu1))
nb2.nses=(ydata1-lambda2)^2/(lambda2*(1+lambda2/est2.nu1))
nb2.ses=(ydata1-lambda2)^2
out[1,1:2,i]=round(c(mean(nb1.logs),mean(nb2.logs)),3)
out[2,1:2,i]=round(c(mean(nb1.qs),mean(nb2.qs)),3)
out[3,1:2,i]=round(c(mean(nb1.sphs),mean(nb2.sphs)),3)
out[4,1:2,i]=round(c(mean(nb1.rps),mean(nb2.rps)),3)
out[5,1:2,i]=round(c(mean(nb1.dss),mean(nb2.dss)),3)
out[6,1:2,i]=round(c(mean(nb1.nses),mean(nb2.nses)),3)
out[7,1:2,i]=round(c(mean(nb1.ses),mean(nb2.ses)),3)
}
mean.scoring.rules=round(apply(out,1:2,mean),2)
return(list(out,mean.scoring.rules))
}

test=parsimonious.model(1,0.01,0.05,0.01,0.2,0.3,2,1500,1000)
test[[2]]
```

Construction of the PIT histogram.

```
############################################################
## Probability Integral Plot function-Czado,Gneiting,Held ##
############################################################
## function for nonrandomized PIT histogram
##
## input:
##   x    observed data
##   Px   CDF at x
##   Px1  CDF at x-1

pit=function(x,Px,Px1,n.bins=10,y.max=2.75,my.title="PIT Histogram")
```

```
{
a.mat=matrix(0,n.bins,length(x))
k.vec=pmax(ceiling(n.bins*Px1),1)
m.vec=ceiling(n.bins*Px)
d.vec=Px-Px1
for (i in 1:length(x))
{
if (k.vec[i]==m.vec[i]) {a.mat[k.vec[i],i]=1}
else
{
a.mat[k.vec[i],i]=((k.vec[i]/n.bins)-Px1[i])/d.vec[i]
if ((k.vec[i]+1)<=(m.vec[i]-1))
{for (j in ((k.vec[i]+1):(m.vec[i]-1))) {a.mat[j,i]=(1/(n.bins*d.vec[i]))}}
a.mat[m.vec[i],i]=(Px[i]-((m.vec[i]-1)/n.bins))/d.vec[i]
}
}
a=apply(a.mat,1,sum)
a=(n.bins*a)/(length(x))
p=(0:n.bins)/n.bins
PIT="Probability Integral Transform"
RF="Relative Frequency"
plot(p,p,ylim=c(0,y.max),type="n",xlab=PIT,ylab=RF,main=my.title)
temp1=((1:n.bins)-1)/n.bins
temp2=((1:n.bins)/n.bins)
o.vec=rep(0,n.bins)
segments(temp1,o.vec,temp1,a)
segments(temp1,a,temp2,a)
segments(temp2,o.vec,temp2,a)
segments(0,0,1,0)
}
```

Considering that `ydata` denotes the vector of observations of the real data collections and after obtaining $\hat{\lambda}_t$ and $\hat{\nu}_2$ using the appropriate functions shown in the Appendix of Chapter 3, we plot the PIT histograms using the `R` code given below.

```
################################################
## PIT Histogram  using the negative binomial ##
################################################
negbin.Px=pnbinom(ydata,size=est.nu2,mu=lambda)
negbin.Px1=pnbinom(ydata-1,size=est.nu2,mu=lambda)
pit(ydata,negbin.Px,negbin.Px1,n.bins=10,y.max=2.75,
    my.title="Linear Negative Binomial Prediction")

####################################
## PIT Histogram using the Poisson ##
####################################
pois.Px=ppois(ydata,lambda)
pois.Px1=ppois(ydata-1,lambda)
pit(ydata,pois.Px,pois.Px1,n.bins=10,y.max=2.75,
    my.title="Linear Poisson Prediction")
```

For the results of the simulation experiments based on the nonlinear model, we use the functions which are properly defined according to model 3.11.

In addition, the functions which are used for the case studies in order to construct Tables 4.5, 4.6, 4.7, 4.8 and Figures 4.5, 4.7, 4.9, 4.11, are properly defined as in the simulation study.

Chapter 5

Testing Linearity for Nonlinear Count Time Series Models

5.1 Introduction

In this chapter we discuss the problem of testing for linearity against two special classes of nonlinear alternatives for count time series data. Evaluating the significance of added variables in a regression equation can be carried out using the likelihood ratio test, Wald test or score (Lagrange Multiplier) test under quasi-likelihood theory. The score test is often a very convenient tool because it does not require the estimation of the nonlinear model and only requires the estimation of the constrained model under the hypothesis. However, careful application of the methodology requires suitable adjustment of the score test, since it is based on quasi-likelihood methodology. Note that, the aforementioned types of test are asymptotically equivalent (cf. Francq and Zakoïan [40, Ch. 8]).

Two major classes of nonlinear models are considered, which both of them nest the linear model. The first class consists of models which do not face the problem of non identifiability, that is all the parameters of the model are identified under the null hypothesis. For this class of models and under the null hypothesis of linearity, the score test statistic possesses an asymptotic $\mathcal{X}^2$ distribution, even if the parameters lie at the boundary of the parameter space; see Francq and Zakoïan [40]. The second class of nonlinear models consists of models in which a nonnegative nuisance parameter exists under the alternative hypothesis but not when linearity holds. In this particular case the testing problem is

nonstandard and the classical asymptotic theory for the score test does not apply. Thus, to resolve this problem, we employ a supremum (sup) type test; see for instance Davies [14].

5.2 Autoregressive Modeling

The proposed modeling approach that we take, has been already introduced in (3.5) of Chapter 3, that is

$$Y_t = \tilde{N}_t(0, Z_t\lambda_t] = N_t(0, \lambda_t], \qquad \lambda_t = f(Y_{t-1}, \lambda_{t-1}), \qquad t \in \mathbb{Z},$$

where N_t, Z_t and f have been described in the aforementioned chapter. The nonlinear specifications of f that we take in this chapter, are slightly different from those used in terms of estimation in Chapter 3. In particular, they are given by

$$\lambda_t = \frac{d}{(1+Y_{t-1})^{\gamma}} + a_1\lambda_{t-1} + b_1 Y_{t-1}, \tag{5.1}$$

and

$$\lambda_t = d + a_1\lambda_{t-1} + (b_1 + c_1\exp(-\gamma Y_{t-1}^2))Y_{t-1}. \tag{5.2}$$

Models (5.1) and (5.2) are modifications of analogous models studied in Chapters 3 and 4. Model (5.1) is a nonlinear model deviation of (3.6) in the sense that small values of γ make (5.1) to approach (3.6). Following the approach taken in Chapter 3, Section 3.2, when $\max\{b_1, d\gamma - b_1\} + a_1 < 1$, model (5.1) is ergodic and stationary whose moments are finite. Similarly, model (5.2) can be viewed as a smooth transition autoregressive (STAR) model (see Teräsvirta [70]). It turns out that when $0 < a_1 + b_1 + c_1 < 1$, model (5.2) is ergodic, stationary and it has moments of all orders.

Note that both of them nest the linear model (3.6).

5.3 Testing Linearity

Among many important issues in practical model building of a nonlinear regression model, is to test for linearity before actually employ a more elaborate nonlinear model. Testing

can be carried out using the likelihood ratio test, Wald test or score (Lagrange Multiplier) test under likelihood theory. Both likelihood ratio and Wald test require estimation for the full model, which in some cases can be very challenging to be implemented. The main advantage of the score test is that it requires estimation only for the constrained model. In other words, testing for linearity requires estimation of the simple linear model, which is the model under the null. In addition, the asymptotic distribution of the score statistic is not affected when parameters lie at the boundary of the hypothesis, (see Francq and Zakoïan [40, Ch. 8]). Score test statistic seems to be the most prominent for testing linearity and the above arguments show that it is an appropriate approach to our testing problem.

Generally speaking, let us denote by $\boldsymbol{\theta} = (\boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}^{(2)})$ the unknown parameter, where $\boldsymbol{\theta}^{(1)}$ and $\boldsymbol{\theta}^{(2)}$ are vectors of dimension m_1 and m_2, respectively, such that $m_1 + m_2 = m$. The hypotheses of interest are

$$H_0 : \boldsymbol{\theta}^{(2)} = \boldsymbol{\theta}_0^{(2)} \quad \text{vs.} \quad H_1 : \boldsymbol{\theta}^{(2)} > \boldsymbol{\theta}_0^{(2)}, \quad \text{componentwise.}$$

Let $\tilde{\boldsymbol{\theta}}_n = (\tilde{\boldsymbol{\theta}}_n^{(1)}, \tilde{\boldsymbol{\theta}}_n^{(2)})$ be the constrained quasi-likelihood estimator of $\boldsymbol{\theta} = (\boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}^{(2)})$, that is the consistent estimator for the parameters of the linear model. Let $\boldsymbol{S}_n = (\boldsymbol{S}_n^{(1)}, \boldsymbol{S}_n^{(2)})$ be the corresponding partition for the score function. The general form of the score statistic is given by (see Rao [69])

$$LM_n = \boldsymbol{S}_n'(\tilde{\boldsymbol{\theta}}_n)\boldsymbol{I}^{-1}(\tilde{\boldsymbol{\theta}}_n)\boldsymbol{S}_n(\tilde{\boldsymbol{\theta}}_n),$$

where $\boldsymbol{I}(\cdot)$ denotes the information matrix. However, it can be shown that this can be reduced to the following formula (see Breslow [7] and Harvey [50, Ch. 5] for instance),

$$LM_n = \boldsymbol{S}_n^{(2)'}(\tilde{\boldsymbol{\theta}}_n)\widetilde{\boldsymbol{\Sigma}}^{-1}(\tilde{\boldsymbol{\theta}}_n)\boldsymbol{S}_n^{(2)}(\tilde{\boldsymbol{\theta}}_n). \tag{5.3}$$

In the above, $\widetilde{\boldsymbol{\Sigma}}$ is an appropriate estimator for the covariance matrix $\boldsymbol{\Sigma} = \mathrm{Var}(\frac{1}{\sqrt{n}}\boldsymbol{S}_n^{(2)}(\tilde{\boldsymbol{\theta}}_n))$.

Because our approach is based on quasi-score instead of the true score, certain adjustments should be made for obtaining its asymptotic distribution. Recall (3.16). Then

consider the following partition

$$\boldsymbol{G} = \begin{pmatrix} \boldsymbol{G}_{11} & \boldsymbol{G}_{12} \\ \boldsymbol{G}_{21} & \boldsymbol{G}_{22} \end{pmatrix},$$

and similarly for $\boldsymbol{G}_1$ defined by means of (3.17). Here $\boldsymbol{G}_{11}, \boldsymbol{G}_{12}, \boldsymbol{G}_{21}, \boldsymbol{G}_{22}$ are partitions of dimension $m_1 \times m_1, m_1 \times m_2, m_2 \times m_1$ and $m_2 \times m_2$ respectively. Partitions of the $\boldsymbol{G}_1$ matrix are defined analogously. Then it can be shown that

$$\boldsymbol{\Sigma} \equiv \boldsymbol{\Sigma}_{MP} = \boldsymbol{G}_{1,22} - \boldsymbol{G}_{21}\boldsymbol{G}_{11}^{-1}\boldsymbol{G}_{1,12} - \boldsymbol{G}_{1,21}\boldsymbol{G}_{11}^{-1}\boldsymbol{G}_{12} + \boldsymbol{G}_{21}\boldsymbol{G}_{11}^{-1}\boldsymbol{G}_{1,11}\boldsymbol{G}_{11}^{-1}\boldsymbol{G}_{12}. \tag{5.4}$$

In the particular case of the Poisson distribution, the matrices $\boldsymbol{G}$ and $\boldsymbol{G}_1$ coincide and therefore

$$\boldsymbol{\Sigma} \equiv \boldsymbol{\Sigma}_P = \boldsymbol{G}_{22} - \boldsymbol{G}_{21}\boldsymbol{G}_{11}^{-1}\boldsymbol{G}_{12},$$

which is the usual form of the covariance matrix for applying the score test under the correct model.

Remark 5.3.1 Consider the nonlinear model (5.1) and let $\boldsymbol{\theta}^{(1)} = (d, a_1, b_1)$ and $\boldsymbol{\theta}^{(2)} = \gamma$. We will test the hypothesis $H_0 : \gamma = 0$. Then the vector $\partial\lambda_t(\boldsymbol{\theta})/\partial\boldsymbol{\theta}$, which appears in the calculation of the score function and $\boldsymbol{\Sigma}$ matrix, is a four-dimensional vector with components given by

$$\begin{aligned}
\frac{\partial\lambda_t}{\partial d} &= \frac{1}{(1+Y_{t-1})^{\gamma}} + a_1\frac{\partial\lambda_{t-1}}{\partial d}, \\
\frac{\partial\lambda_t}{\partial a_1} &= \lambda_{t-1} + a_1\frac{\partial\lambda_{t-1}}{\partial a_1}, \\
\frac{\partial\lambda_t}{\partial b_1} &= Y_{t-1} + a_1\frac{\partial\lambda_{t-1}}{\partial b_1} \quad \text{and} \\
\frac{\partial\lambda_t}{\partial \gamma} &= -\frac{d\log(1+Y_{t-1})}{(1+Y_{t-1})^{\gamma}} + a_1\frac{\partial\lambda_{t-1}}{\partial \gamma}.
\end{aligned}$$

Similarly, for the exponential model (5.2) we want to test $H_0 : c_1 = 0$. Then for γ known, $\partial\lambda_t(\boldsymbol{\theta})/\partial\boldsymbol{\theta}$ is a four-dimensional vector with components given by

$$\begin{aligned}
\frac{\partial \lambda_t}{\partial d} &= 1 + a_1 \frac{\partial \lambda_{t-1}}{\partial d}, \\
\frac{\partial \lambda_t}{\partial a_1} &= \lambda_{t-1} + a_1 \frac{\partial \lambda_{t-1}}{\partial a_1}, \\
\frac{\partial \lambda_t}{\partial b_1} &= Y_{t-1} + a_1 \frac{\partial \lambda_{t-1}}{\partial b_1} \quad \text{and} \\
\frac{\partial \lambda_t}{\partial c_1} &= Y_{t-1} \exp(-\gamma Y_{t-1}^2) + a_1 \frac{\partial \lambda_{t-1}}{\partial c_1}.
\end{aligned}$$

In both models, $\partial \lambda_t(\boldsymbol{\theta})/\partial \boldsymbol{\theta}$ is evaluated at $\tilde{\boldsymbol{\theta}} = (\tilde{\boldsymbol{\theta}}^{(1)}, 0) = (\tilde{d}, \tilde{a}_1, \tilde{b}_1, 0)$, that is at the estimator under the null.

5.3.1 Standard Implementation

If all the parameters are identified under the null hypothesis, then the standard asymptotic theory holds. In particular, we have the following result.

Proposition 5.3.1 Suppose that $\{Y_t, t = 1, \ldots, n\}$ is an observed count time series specified by (3.1) with mean process $\{\lambda_t\}$ defined by (3.4). Suppose that the function $f(\cdot)$ depends upon a vector $(\boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}^{(2)})$, where $\boldsymbol{\theta}^{(i)}$ is of dimension $m_i, i = 1, 2$. Consider the problem

$$H_0 : \boldsymbol{\theta}^{(2)} = \boldsymbol{\theta}_0^{(2)} \quad \text{vs.} \quad H_1 : \boldsymbol{\theta}^{(2)} > \boldsymbol{\theta}_0^{(2)}, \quad \text{componentwise.} \tag{5.5}$$

Then the score test converges to a chi-square random variable, i.e.

$$LM_n = \boldsymbol{S}_n^{(2)'}(\tilde{\boldsymbol{\theta}}_n) \widetilde{\boldsymbol{\Sigma}}^{-1}(\tilde{\boldsymbol{\theta}}_n) \boldsymbol{S}_n^{(2)}(\tilde{\boldsymbol{\theta}}_n) \xrightarrow{D} \mathcal{X}^2_{m_2},$$

as $n \to \infty$, when H_0 is true.

In addition, consider testing

$$H_0 : \boldsymbol{\theta}^{(2)} = \boldsymbol{\theta}_0^{(2)} \quad \text{vs.} \quad H_1 : \boldsymbol{\theta}^{(2)} = \boldsymbol{\theta}_0^{(2)} + n^{-1/2}\boldsymbol{\delta},$$

where $\boldsymbol{\delta}$ is a fixed vector in $\mathbb{R}_+^{m_2}$, then

$$LM_n = \boldsymbol{S}_n^{(2)'}(\tilde{\boldsymbol{\theta}}_n) \widetilde{\boldsymbol{\Sigma}}^{-1}(\tilde{\boldsymbol{\theta}}_n) \boldsymbol{S}_n^{(2)}(\tilde{\boldsymbol{\theta}}_n) \xrightarrow{D} \mathcal{X}^2_{m_2}(\boldsymbol{\delta}' \boldsymbol{\Delta} \boldsymbol{\delta}),$$

with $\boldsymbol{\Delta} = \widetilde{\boldsymbol{\Sigma}}_P \widetilde{\boldsymbol{\Sigma}}_{MP}^{-1} \widetilde{\boldsymbol{\Sigma}}_P$.

The proof is postponed to the appendix. In particular, note that if the data are generated by the usual Poisson model, then the non centrality parameter $\boldsymbol{\Delta}$ equals to $\widetilde{\boldsymbol{\Sigma}}_P$.

For example, recall the nonlinear model (5.1) and let $\boldsymbol{\theta}^{(1)} = (d, a_1, b_1)$ and $\boldsymbol{\theta}^{(2)} = \gamma$. Testing the hypothesis $H_0 : \gamma = 0$, we face the problem that under the null the parameter γ is at the boundary of the parameter space. According to Francq and Zakoïan [40, pp. 196], the score statistic is not affected by this fact and the asymptotic distribution is the chi-square with one degree of freedom.

5.3.2 Non-Standard Implementation

In many cases we have to test linearity assumption for nonlinear models that contain nuisance parameters that are not identified under the null. As a consequence of this, the quasi maximum likelihood estimators have nonstandard behavior. The lack of identification affects also the score test and the classical asymptotic theory does not apply. A comprehensive discussion of this identification problem can be found in the work of Davies [14], Andrews and Ploberger [2], Teräsvirta [70], Andrews [1] and Luukkonen et al. [62] for the case of smooth transition autoregressive (STAR) modeling.

An illustrative example of this case is given by model (5.2). When $c_1 = 0$, the parameter γ is not identified under the null. Consequently, the test of

$$H_0 : c_1 = 0 \quad \text{vs.} \quad H_1 : c_1 > 0, \tag{5.6}$$

is not standard and the traditional large sample theory is not applicable.

To deal with this problem a reasonable strategy is to consider a fixed arbitrary value of γ. Then the model is still linear in the parameters and the score statistic LM_n is asymptotically distributed as $\mathcal{X}_1^2$ under H_0. Despite the fact that this test is in general consistent, even for alternatives where $\gamma_0 \neq \gamma$, it may lacks of power for alternatives where γ_0 is far from γ.

To avoid low values of power regarding the above approach, we turned to a supremum test proposed by Davies [14]. Consider Γ is a grid of values for the parameter γ, which is a meaningless parameter under the null hypothesis. Then the sup-score test statistic is

given by

$$LM_n = \sup_{\gamma \in \Gamma} LM_n(\gamma),$$

where $LM_n(\gamma)$ is given by (5.3). We reject hypothesis (5.6) for large values of LM_n which can be calculated by bootstrapping the test statistic.

In the following, we examine the finite sample behavior of the score test under both cases of identifiable and non identifiable parameters.

5.4 Simulation Study

The main goal of the test procedure is to obtain test statistics which have correct size and high power. However, in the vast majority of interesting cases, the distributions of the test statistic we use are known only asymptotically. A reasonable approach is to simulate the finite sample behavior by employing parametric bootstrap. To obtain the size of the test, the basic idea is to bootstrap the test statistic drawing a large number of bootstrap samples, which obey the null hypothesis and resemble the real sample, and then compare the observed test statistic to the ones calculated from the bootstrap samples (see Hill [52] for instance). More specifically, testing hypotheses (5.5) when the parameter of interest is identifiable can be implemented by the following steps, for the case of Poisson or negative binomial process.

1. Generate data $(Y_1, Y_2, \ldots, Y_n)$ from the linear model.

2. Estimate the unknown parameters under the null hypothesis H_0. The estimator is denoted by $\tilde{\boldsymbol{\theta}}$.

3. Compute the observed score statistic value LM given by (5.3), using the correct estimator for the covariance matrix $\boldsymbol{\Sigma}$ according to the distributional assumption.

4. Generate B time series of length n using the estimated model under H_0.

5. For each time series, $\mathrm{b} = 1, \ldots, \mathrm{B}$, estimate the model under H_0. The estimator is denoted by $\tilde{\boldsymbol{\theta}}^*_{\mathrm{b}}$.

6. For each time series, $\mathrm{b} = 1, \ldots, \mathrm{B}$, compute the value of the score statistic denoted by LM^*_{b}.

7. Compute the p-value by the following ratio

$$\text{p-value} = \frac{\#\{\text{b} : LM_{\text{b}}^{*} \geq LM\} + 1}{\text{B} + 1}.$$

In the case of a non identifiable parameter, modify the algorithm by generating a grid for the values of the non identifiable parameter and consider the supremum of the score test over this grid.

5.4.1 Results for Testing Linearity for Model (5.1)

To obtain the size of the test we generate data $Y_1, \ldots, Y_n$ using either the Poisson distribution or the negative binomial where the mean values are given by model (3.6). We choose $(d, a_1, b_1) = (0.5, 0.4, 0.5)$ or $(1.5, 0.05, 0.6)$. We consider two different sample sizes, $n = 500$ and $n = 1000$. For all of the cases, we use $\text{B} = 499$ bootstrap replicates and the simulations run 200 times. For the case of the negative binomial assumption we use $\nu = 2$ and $\nu = 4$. Calculation of the maximum likelihood estimators is carried out by optimizing the log-likelihood function (3.13) by a quasi-Newton method using the `constrOptim()` function of `R`. Estimation is implemented by discarding the first 500 observations so that the process is within its stationary region. For selected nominal significance level $\alpha = 1\%, \alpha = 5\%$ or $\alpha = 10\%$ we obtain the results shown in Tables 5.1 and 5.2. Note that for purposes of comparison, the last three lines of the tables list the size of the test derived from the $\mathcal{X}_1^2$ distribution. We note that in all cases the bootstrap approximation works reasonably well.

For the power simulations we work in an analogous way but now the mean process is given by

$$\lambda_t = \frac{1.5}{(1 + Y_{t-1})^{\gamma}} + 0.05\lambda_{t-1} + 0.6Y_{t-1},$$

for values of $\gamma = 0.3, 0.5$ and 1. Table 5.3 reports the results and it shows that as γ assumes larger values, the power of the test statistic (5.3) approaches unity; see also Figure 5.1.

Remark 5.4.1 We have also tried to examine the sensitivity of the power of the score test for model (5.1) when γ assumes negative values but close to zero. Table 5.4 reports power results for $\gamma = -0.5, -0.3, -0.1$. For values of $\gamma < -0.5$, we do not obtain a stationary

	Bootstrap test for $n = 500$			Bootstrap test for $n = 1000$		
Nominal significance level	Poisson	NegBin ($\nu = 2$)	NegBin ($\nu = 4$)	Poisson	NegBin ($\nu = 2$)	NegBin ($\nu = 4$)
$\alpha = 1\%$	0.005	0.005	0.000	0.000	0.016	0.015
$\alpha = 5\%$	0.050	0.048	0.025	0.055	0.069	0.060
$\alpha = 10\%$	0.130	0.134	0.080	0.100	0.139	0.095
	Approximation test for $n = 500$			Approximation test for $n = 1000$		
$\alpha = 1\%$	0.020	0.010	0.005	0.015	0.015	0.005
$\alpha = 5\%$	0.075	0.035	0.030	0.070	0.030	0.035
$\alpha = 10\%$	0.165	0.090	0.080	0.145	0.055	0.085

Table 5.1: Empirical size for sample sizes $n = 500$ and $n = 1000$. Data are generated from the linear model (3.6) with true values $(d, a_1, b_1) = (0.5, 0.4, 0.5)$. Results are based on B = 499 bootstrap replicates and 200 simulations.

	Bootstrap test for $n = 500$			Bootstrap test for $n = 1000$		
Nominal significance level	Poisson	NegBin ($\nu = 2$)	NegBin ($\nu = 4$)	Poisson	NegBin ($\nu = 2$)	NegBin ($\nu = 4$)
$\alpha = 1\%$	0.010	0.005	0.005	0.010	0.010	0.026
$\alpha = 5\%$	0.041	0.068	0.037	0.057	0.037	0.051
$\alpha = 10\%$	0.096	0.094	0.084	0.114	0.115	0.097
	Approximation test for $n = 500$			Approximation test for $n = 1000$		
$\alpha = 1\%$	0.020	0.010	0.015	0.010	0.005	0.010
$\alpha = 5\%$	0.075	0.030	0.055	0.060	0.020	0.035
$\alpha = 10\%$	0.150	0.075	0.110	0.115	0.055	0.090

Table 5.2: Empirical size for sample sizes $n = 500$ and $n = 1000$. Data are generated from the linear model (3.6) with true values $(d, a_1, b_1) = (1.5, 0.05, 0.6)$. Results are based on B = 499 bootstrap replicates and 200 simulations.

sequence. We note from Table 5.4 that in the case of Poisson the power of the test is close to its nominal significance level when $n = 500$. For larger sample sizes and under the Poisson, the power is rather poor. For the negative binomial case, we note that the power behaves better than the Poisson case, specifically when $\gamma = -0.5$, that is γ far from 0.

This simple exercise shows that it might be worthwhile to investigate power of the test statistic for parameter values which do not belong to the stationary region.

Nonlinear model (5.1)	Bootstrap test for $n = 500$			Bootstrap test for $n = 1000$		
γ	Poisson	NegBin ($\nu = 2$)	NegBin ($\nu = 4$)	Poisson	NegBin ($\nu = 2$)	NegBin ($\nu = 4$)
$\gamma = 0.3$	0.207	0.061	0.157	0.271	0.144	0.212
$\gamma = 0.5$	0.450	0.251	0.424	0.740	0.548	0.688
$\gamma = 1$	0.924	0.756	0.837	1.000	0.974	0.995
	Approximation test for n=500			Approximation test for n=1000		
$\gamma = 0.3$	0.210	0.045	0.130	0.360	0.185	0.250
$\gamma = 0.5$	0.485	0.240	0.300	0.745	0.455	0.635
$\gamma = 1$	0.920	0.710	0.870	0.995	0.955	0.990

Table 5.3: Empirical power for sample sizes $n = 500$ and $n = 1000$. Data are generated from the nonlinear model (5.1) with true values $(d, a_1, b_1) = (1.5, 0.05, 0.6)$ and $\gamma \in \{0.3, 0.5, 1\}$. Results are based on B = 499 bootstrap replicates and 200 simulations. The nominal significance level is $\alpha = 5\%$.

Nonlinear model (5.1)	Bootstrap test for $n = 500$			Bootstrap test for $n = 1000$		
γ	Poisson	NegBin ($\nu = 2$)	NegBin ($\nu = 4$)	Poisson	NegBin ($\nu = 2$)	NegBin ($\nu = 4$)
$\gamma = -0.5$	0.040	0.587	0.361	0.105	0.650	0.661
$\gamma = -0.3$	0.070	0.269	0.194	0.165	0.305	0.297
$\gamma = -0.1$	0.055	0.071	0.093	0.055	0.115	0.062

Table 5.4: Empirical power for sample sizes $n = 500$ and $n = 1000$. Data are generated from the nonlinear model (5.1) with true values $(d, a_1, b_1) = (1.5, 0.05, 0.6)$ and $\gamma \in \{-0.5, -0.3, -0.1\}$. Results are based on B = 499 bootstrap replicates and 200 simulations. The nominal significance level is $\alpha = 5\%$.

5.4.2 Results for Testing Linearity for Model (5.2)

To obtain the size of the test we work exactly in the same way as for the linearity test of nonlinear model (5.1), using the true values $(d, a_1, b_1) = (0.5, 0.4, 0.5)$ or $(d, a_1, b_1) = (0.5, 0.3, 0.2)$. As far as the power of the test is concerned, the mean process is given by

$$\lambda_t = 0.5 + 0.3\lambda_{t-1} + (0.2 + c_1 \exp(-\gamma Y_{t-1}^2))Y_{t-1},$$

where $c_1 \in \{0.2, 0.4\}$ and $\gamma \in \{0.05, 0.2, 0.5, 1\}$.

Note that the grid for γ to obtain the sup-score statistic, is a sequence of 30 equidistant values in the interval $[0.01, 2]$. Results are given in Tables 5.5, 5.6 and 5.7. Tables 5.5 and 5.6 show that for small samples the nominal level of the test statistic is usually

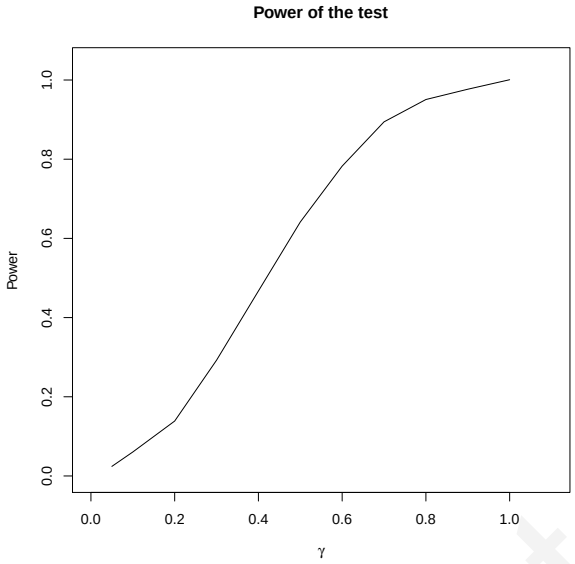

Figure 5.1: Plot for the power of the test based on $\mathcal{X}_1^2$ distribution, when the data are generated from the nonlinear model (5.1) with true values $(d, a_1, b_1) = (1.5, 0.05, 0.6)$, $\nu = 4$ and for different values of γ. The results are based on 1000 data points and 200 simulations.

underestimated. However, larger sample sizes yield to more accurate approximation. The power of the supremum test statistic is relatively large for larger sample sizes and for values of γ close to zero provided that the parameter c_1 is also of large magnitude in model (5.2).

	Bootstrap test for $n = 500$			Bootstrap test for $n = 1000$		
Nominal significance level	Poisson	NegBin ($\nu = 2$)	NegBin ($\nu = 4$)	Poisson	NegBin ($\nu = 2$)	NegBin ($\nu = 4$)
$\alpha = 1\%$	0.015	0.005	0.005	0.005	0.010	0.015
$\alpha = 5\%$	0.070	0.054	0.035	0.030	0.068	0.065
$\alpha = 10\%$	0.105	0.097	0.080	0.075	0.099	0.140

Table 5.5: Empirical size for sample sizes $n = 500$ and $n = 1000$. Data are generated from the linear model (3.6) with true values $(d, a_1, b_1) = (0.5, 0.4, 0.5)$. Results are based on B = 499 bootstrap replicates and 200 simulations.

5.5 Examples

In order to examine whether or not the linear model (3.6) is adequate to fit a real data collection, we employ testing for linearity against two different nonlinear specifications; the nonlinear model (5.1) and the exponential model (5.2), following the Poisson or the negative binomial distributional assumption as discussed above.

	Bootstrap test for $n = 500$			Bootstrap test for $n = 1000$		
Nominal significance level	Poisson	NegBin ($\nu = 2$)	NegBin ($\nu = 4$)	Poisson	NegBin ($\nu = 2$)	NegBin ($\nu = 4$)
$\alpha = 1\%$	0.000	0.000	0.010	0.000	0.000	0.005
$\alpha = 5\%$	0.048	0.043	0.037	0.020	0.045	0.060
$\alpha = 10\%$	0.122	0.102	0.099	0.046	0.075	0.105

Table 5.6: Empirical size for sample sizes $n = 500$ and $n = 1000$. Data are generated from the linear model (3.6) with true values $(d, a_1, b_1) = (0.5, 0.3, 0.2)$. Results are based on B = 499 bootstrap replicates and 200 simulations.

Nonlinear model (5.2)		Bootstrap test for $n = 500$			Bootstrap test for $n = 1000$		
c_1	γ	Poisson	NegBin ($\nu = 2$)	NegBin ($\nu = 4$)	Poisson	NegBin ($\nu = 2$)	NegBin ($\nu = 4$)
$c_1 = 0.2$	$\gamma = 0.05$	0.140	0.178	0.220	0.315	0.390	0.355
$c_1 = 0.2$	$\gamma = 0.2$	0.199	0.175	0.231	0.485	0.335	0.372
$c_1 = 0.2$	$\gamma = 0.5$	0.111	0.156	0.122	0.312	0.190	0.265
$c_1 = 0.2$	$\gamma = 1$	0.077	0.070	0.084	0.105	0.122	0.135
$c_1 = 0.4$	$\gamma = 0.05$	0.755	0.690	0.739	0.985	0.950	0.985
$c_1 = 0.4$	$\gamma = 0.2$	0.746	0.609	0.628	0.985	0.905	0.925
$c_1 = 0.4$	$\gamma = 0.5$	0.420	0.358	0.469	0.855	0.650	0.775
$c_1 = 0.4$	$\gamma = 1$	0.174	0.154	0.235	0.465	0.255	0.340

Table 5.7: Empirical power for sample sizes $n = 500$ and $n = 1000$. Data are generated from the exponential model (5.2) with true values $(d, a_1, b_1) = (0.5, 0.3, 0.2)$, $c_1 \in \{0.2, 0.4\}$ and $\gamma \in \{0.05, 0.2, 0.5, 1\}$. Results are based on B = 499 bootstrap replicates and 200 simulations. The nominal significance level is $\alpha = 5\%$.

Firstly, we consider the data collection of transactions reported in Section 3.5 of Chapter 3. Following the steps proposed by the bootstrap procedure, for B = 499 bootstrap replications, we calculate the p-values for each model and for each distributional assumption. Table 5.8 summarizes the results, which in most of the cases, indicate that the linear model (3.6) is accepted. If we consider testing linearity for the exponential model (5.2) regarding the negative binomial distributional assumption, then the linear model is rejected.

The second data example that we consider, corresponds to the monthly number of measles at Sheffield, studied in Section 3.5 of Chapter 3. The results of the p-values calculated by the proposed bootstrap procedure for B = 499 bootstrap replications, are shown in Table 5.9, which indicate that the linear model (3.6) is always rejected.

Another data example that we consider, corresponds to the numbers of deaths recorded

	Distributional assumption	
	Poisson	Negative binomial
Nonlinear Model (5.1)	100%	99%
Exponential Model (5.2)	100%	0%

Table 5.8: p-values of the linearity test for the total number of transactions per minute for the stock Ericsson B for the time period between July 2nd and July 22nd, 2002. The total number of observations is 460.

	Distributional assumption	
	Poisson	Negative binomial
Nonlinear Model (5.1)	0%	0%
Exponential Model (5.2)	0%	0%

Table 5.9: p-values of the linearity test for the monthly number of measles at Sheffield for the period between September 8th, 1978 and April 17th, 1987. The total number of observations is 450.

at the Salt River state mortuary at Cape Town, for the period time 1986 - 1991. The results of the p-values calculated by the proposed bootstrap procedure for B = 499 bootstrap replications, are shown in Table 5.10. The results indicate that when the exponential model (5.2) and the Poisson distribution is assumed, then the linear model (3.6) is adequate to fit the homicides data, while in all the other cases the linear model (3.6) is rejected.

	Distributional assumption	
	Poisson	Negative binomial
Nonlinear Model (5.1)	0%	0%
Exponential Model (5.2)	46%	0%

Table 5.10: p-values of the linearity test for the numbers of deaths recorded at the Salt River state mortuary at Cape Town, for the period time 1986 - 1991. The total number of observations is 312.

Appendix A – Proofs

Proof of Proposition 5.3.1

Recall that $\hat{\sigma}_Z^2$ is defined by the solution of (3.20) and is consistent. Consider first the case where the data are generated from the Poisson distribution and recall again that $\tilde{\boldsymbol{\theta}}_n = (\tilde{\boldsymbol{\theta}}_n^{(1)}, \tilde{\boldsymbol{\theta}}_n^{(2)})$ is the consistent estimator of $\boldsymbol{\theta} = (\boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}^{(2)})$ under the null hypothesis. Since $\tilde{\boldsymbol{\theta}}_n^{(1)}$ is a consistent estimator of $\boldsymbol{\theta}^{(1)}$ and $\boldsymbol{\theta}^{(1)} > \mathbf{0}$, then for n large enough we have that $\tilde{\boldsymbol{\theta}}_n^{(1)} > \mathbf{0}$ and $S_{n,i}(\tilde{\boldsymbol{\theta}}_n) := \partial l_n(\tilde{\boldsymbol{\theta}}_n)/\partial\theta_i = 0, \quad \forall\ i = 1, \ldots, m_1$. That is, $\tilde{\boldsymbol{\theta}}_n^{(1)} > \mathbf{0}$ and $\boldsymbol{S}_n^{(1)}(\tilde{\boldsymbol{\theta}}_n) = \mathbf{0}$.

If we define the matrices $\boldsymbol{K} = (\boldsymbol{O}_{m_2 \times m_1}, \boldsymbol{I}_{m_2})$ and $\widetilde{\boldsymbol{K}} = (\boldsymbol{I}_{m_1}, \boldsymbol{O}_{m_1 \times m_2})$, we have that

$$\boldsymbol{S}_n^{(1)}(\tilde{\boldsymbol{\theta}}_n) = \widetilde{\boldsymbol{K}}\boldsymbol{S}_n(\tilde{\boldsymbol{\theta}}_n) = \mathbf{0} \quad \text{and} \quad \boldsymbol{S}_n(\tilde{\boldsymbol{\theta}}_n) = \boldsymbol{K}'\boldsymbol{S}_n^{(2)}(\tilde{\boldsymbol{\theta}}_n). \tag{5.7}$$

Since

$$\frac{1}{n}\boldsymbol{H}_n(\boldsymbol{\theta}_0) = -\frac{1}{n}\sum_{t=1}^{n} \frac{\partial^2 l_t(\boldsymbol{\theta}_0)}{\partial\boldsymbol{\theta}\partial\boldsymbol{\theta}'} \xrightarrow{p} \boldsymbol{G}(\boldsymbol{\theta}_0),$$

a Taylor expansion shows that

$$\boldsymbol{S}_n(\tilde{\boldsymbol{\theta}}_n) \overset{o_p(1)}{=} \boldsymbol{S}_n(\boldsymbol{\theta}_0) - \boldsymbol{G}(\tilde{\boldsymbol{\theta}}_n - \boldsymbol{\theta}_0).$$

Therefore, the last m_2 components of the above relation give

$$\boldsymbol{S}_n^{(2)}(\tilde{\boldsymbol{\theta}}_n) \overset{o_p(1)}{=} \boldsymbol{S}_n^{(2)}(\boldsymbol{\theta}_0) - \boldsymbol{K}\boldsymbol{G}(\tilde{\boldsymbol{\theta}}_n - \boldsymbol{\theta}_0), \tag{5.8}$$

and the first m_1 components yield

$$\mathbf{0} = \boldsymbol{S}_n^{(1)}(\tilde{\boldsymbol{\theta}}_n) \overset{o_p(1)}{=} \boldsymbol{S}_n^{(1)}(\boldsymbol{\theta}_0) - \widetilde{\boldsymbol{K}}\boldsymbol{G}(\tilde{\boldsymbol{\theta}}_n - \boldsymbol{\theta}_0).$$

In addition,

$$(\tilde{\boldsymbol{\theta}}_n - \boldsymbol{\theta}_0) = \widetilde{\boldsymbol{K}}'(\tilde{\boldsymbol{\theta}}_n^{(1)} - \boldsymbol{\theta}_0^{(1)}). \tag{5.9}$$

Thus, we have that

$$\mathbf{0} \overset{o_p(1)}{=} \boldsymbol{S}_n^{(1)}(\boldsymbol{\theta}_0) - \widetilde{\boldsymbol{K}}\boldsymbol{G}\widetilde{\boldsymbol{K}}'(\tilde{\boldsymbol{\theta}}_n^{(1)} - \boldsymbol{\theta}_0^{(1)}),$$

or

$$(\tilde{\boldsymbol{\theta}}_n^{(1)} - \boldsymbol{\theta}_0^{(1)}) \overset{o_p(1)}{=} (\widetilde{\boldsymbol{K}}\boldsymbol{G}\widetilde{\boldsymbol{K}}')^{-1}\boldsymbol{S}_n^{(1)}(\boldsymbol{\theta}_0). \tag{5.10}$$

Substituting (5.7), (5.8) and (5.9) in the general expression of the score test, we have that

$$\begin{aligned}
LM_n &= \boldsymbol{S}_n'(\tilde{\boldsymbol{\theta}}_n)\boldsymbol{I}^{-1}(\tilde{\boldsymbol{\theta}}_n)\boldsymbol{S}_n(\tilde{\boldsymbol{\theta}}_n) \\
&= \boldsymbol{S}_n'(\tilde{\boldsymbol{\theta}}_n)\boldsymbol{G}^{-1}(\tilde{\boldsymbol{\theta}}_n)\boldsymbol{S}_n(\tilde{\boldsymbol{\theta}}_n) \\
&= \boldsymbol{S}_n^{(2)'}(\tilde{\boldsymbol{\theta}}_n)\boldsymbol{K}\boldsymbol{G}^{-1}\boldsymbol{K}'\boldsymbol{S}_n^{(2)}(\tilde{\boldsymbol{\theta}}_n) \\
&\overset{o_p(1)}{=} (\boldsymbol{S}_n^{(2)}(\boldsymbol{\theta}_0) - \boldsymbol{K}\boldsymbol{G}(\tilde{\boldsymbol{\theta}}_n - \boldsymbol{\theta}_0))'\boldsymbol{K}\boldsymbol{G}^{-1}\boldsymbol{K}'(\boldsymbol{S}_n^{(2)}(\boldsymbol{\theta}_0) - \boldsymbol{K}\boldsymbol{G}(\tilde{\boldsymbol{\theta}}_n - \boldsymbol{\theta}_0)) \\
&= (\boldsymbol{S}_n^{(2)}(\boldsymbol{\theta}_0) - \boldsymbol{K}\boldsymbol{G}\widetilde{\boldsymbol{K}}'(\tilde{\boldsymbol{\theta}}_n^{(1)} - \boldsymbol{\theta}_0^{(1)}))'\boldsymbol{K}\boldsymbol{G}^{-1}\boldsymbol{K}'(\boldsymbol{S}_n^{(2)}(\boldsymbol{\theta}_0) - \boldsymbol{K}\boldsymbol{G}\widetilde{\boldsymbol{K}}'(\tilde{\boldsymbol{\theta}}_n^{(1)} - \boldsymbol{\theta}_0^{(1)})).
\end{aligned}$$

Let

$$\boldsymbol{W} = \begin{pmatrix}\boldsymbol{W}_1 \\ \boldsymbol{W}_2\end{pmatrix} = \frac{1}{\sqrt{n}}\begin{pmatrix}\boldsymbol{S}_n^{(1)}(\boldsymbol{\theta}_0) \\ \boldsymbol{S}_n^{(2)}(\boldsymbol{\theta}_0)\end{pmatrix} \xrightarrow{D} \mathcal{N}(\boldsymbol{0}, \boldsymbol{G}) \equiv \mathcal{N}\Big(\boldsymbol{0}, \begin{pmatrix}\boldsymbol{G}_{11} & \boldsymbol{G}_{12} \\ \boldsymbol{G}_{21} & \boldsymbol{G}_{22}\end{pmatrix}\Big).$$

It holds that

$$\boldsymbol{K}\boldsymbol{G}\widetilde{\boldsymbol{K}}' = \boldsymbol{G}_{21},\ \widetilde{\boldsymbol{K}}\boldsymbol{G}\boldsymbol{K}' = \boldsymbol{G}_{12},\ \widetilde{\boldsymbol{K}}\boldsymbol{G}\widetilde{\boldsymbol{K}}' = \boldsymbol{G}_{11},\ \boldsymbol{K}\boldsymbol{G}^{-1}\boldsymbol{K}' = (\boldsymbol{G}_{22} - \boldsymbol{G}_{21}\boldsymbol{G}_{11}^{-1}\boldsymbol{G}_{12})^{-1},$$

where the last equality comes from the inversion of the block matrix $\boldsymbol{G}$ and denotes the matrix $\widetilde{\boldsymbol{\Sigma}}_P^{-1}$.

Thus, the score statistic is

$$LM_n = (\boldsymbol{W}_2 - \boldsymbol{G}_{21}(\tilde{\boldsymbol{\theta}}_n^{(1)} - \boldsymbol{\theta}_0^{(1)}))'(\boldsymbol{G}_{22} - \boldsymbol{G}_{21}\boldsymbol{G}_{11}^{-1}\boldsymbol{G}_{12})^{-1}(\boldsymbol{W}_2 - \boldsymbol{G}_{21}(\tilde{\boldsymbol{\theta}}_n^{(1)} - \boldsymbol{\theta}_0^{(1)})),$$

and using (5.10) we finally have that

$$LM_n = (\boldsymbol{W}_2 - \boldsymbol{G}_{21}\boldsymbol{G}_{11}^{-1}\boldsymbol{W}_1)'\widetilde{\boldsymbol{\Sigma}}_P^{-1}(\boldsymbol{W}_2 - \boldsymbol{G}_{21}\boldsymbol{G}_{11}^{-1}\boldsymbol{W}_1). \tag{5.11}$$

Then $LM_n \xrightarrow{D} LM \overset{H_0}{\approx} \mathcal{X}^2_{m_2}$, because of the fact that

$$\boldsymbol{W}_2 - \boldsymbol{G}_{21}\boldsymbol{G}_{11}^{-1}\boldsymbol{W}_1 \overset{H_0}{\approx} \mathcal{N}(\boldsymbol{0}, \widetilde{\boldsymbol{\Sigma}}_P).$$

Indeed, we have that

$$\boldsymbol{W}_2 - \boldsymbol{G}_{21}\boldsymbol{G}_{11}^{-1}\boldsymbol{W}_1 = \begin{pmatrix} -\boldsymbol{G}_{21}\boldsymbol{G}_{11}^{-1} & \boldsymbol{I} \end{pmatrix} \begin{pmatrix} \boldsymbol{W}_1 \\ \boldsymbol{W}_2 \end{pmatrix} = \boldsymbol{G}^*\boldsymbol{W}$$

and therefore,

$$\begin{aligned} \mathrm{E}(\boldsymbol{W}_2 - \boldsymbol{G}_{21}\boldsymbol{G}_{11}^{-1}\boldsymbol{W}_1) &= \boldsymbol{G}^*\mathrm{E}(\boldsymbol{W}) = \boldsymbol{0} \quad \text{and} \\ \mathrm{Var}(\boldsymbol{W}_2 - \boldsymbol{G}_{21}\boldsymbol{G}_{11}^{-1}\boldsymbol{W}_1) &= \boldsymbol{G}^*\mathrm{Var}(\boldsymbol{W})\boldsymbol{G}^{*'} \\ &= \begin{pmatrix} -\boldsymbol{G}_{21}\boldsymbol{G}_{11}^{-1} & \boldsymbol{I} \end{pmatrix} \begin{pmatrix} \boldsymbol{G}_{11} & \boldsymbol{G}_{12} \\ \boldsymbol{G}_{21} & \boldsymbol{G}_{22} \end{pmatrix} \begin{pmatrix} -\boldsymbol{G}_{11}^{-1}\boldsymbol{G}_{12} \\ \boldsymbol{I} \end{pmatrix} \\ &= \begin{pmatrix} \boldsymbol{0} & -\boldsymbol{G}_{21}\boldsymbol{G}_{11}^{-1}\boldsymbol{G}_{12} + \boldsymbol{G}_{22} \end{pmatrix} \begin{pmatrix} -\boldsymbol{G}_{11}^{-1}\boldsymbol{G}_{12} \\ \boldsymbol{I} \end{pmatrix} \\ &= \boldsymbol{G}_{22} - \boldsymbol{G}_{21}\boldsymbol{G}_{11}^{-1}\boldsymbol{G}_{12} = \widetilde{\boldsymbol{\Sigma}}_P. \end{aligned}$$

Consider now the misspecified model where the data are generated from the mixed Poisson model. Following Francq and Zakoïan [40] and Gallant [41, pp. 126], (see also Boos [6], Kent [57] and Engle [28]), the score statistic is given by

$$\begin{aligned} LM_n &= \boldsymbol{S}_n'(\tilde{\boldsymbol{\theta}}_n)\boldsymbol{G}^{-1}\boldsymbol{K}'(\boldsymbol{K}\boldsymbol{G}^{-1}\boldsymbol{G}_1\boldsymbol{G}^{-1}\boldsymbol{K}')^{-1}\boldsymbol{K}\boldsymbol{G}^{-1}\boldsymbol{S}_n(\tilde{\boldsymbol{\theta}}_n) \\ &= \boldsymbol{S}_n^{(2)'}(\tilde{\boldsymbol{\theta}}_n)\boldsymbol{K}\boldsymbol{G}^{-1}\boldsymbol{K}'(\boldsymbol{K}\boldsymbol{G}^{-1}\boldsymbol{G}_1\boldsymbol{G}^{-1}\boldsymbol{K}')^{-1}\boldsymbol{K}\boldsymbol{G}^{-1}\boldsymbol{K}'\boldsymbol{S}_n^{(2)}(\tilde{\boldsymbol{\theta}}_n). \end{aligned}$$

Some calculations yield that $\boldsymbol{K}\boldsymbol{G}^{-1}\boldsymbol{K}' = \boldsymbol{G}^{22} = (\boldsymbol{G}_{22} - \boldsymbol{G}_{21}\boldsymbol{G}_{11}^{-1}\boldsymbol{G}_{12})^{-1}$ and $(\boldsymbol{K}\boldsymbol{G}^{-1}\boldsymbol{G}_1\boldsymbol{G}^{-1}\boldsymbol{K}')^{-1} = (\boldsymbol{G}^{22})^{-1}\widetilde{\boldsymbol{\Sigma}}_{MP}^{-1}(\boldsymbol{G}^{22})^{-1}$, where $\widetilde{\boldsymbol{\Sigma}}_{MP}$ is given by (5.4), that is

$$\widetilde{\boldsymbol{\Sigma}}_{MP} = \boldsymbol{G}_{1,22} - \boldsymbol{G}_{21}\boldsymbol{G}_{11}^{-1}\boldsymbol{G}_{1,12} - \boldsymbol{G}_{1,21}\boldsymbol{G}_{11}^{-1}\boldsymbol{G}_{12} + \boldsymbol{G}_{21}\boldsymbol{G}_{11}^{-1}\boldsymbol{G}_{1,11}\boldsymbol{G}_{11}^{-1}\boldsymbol{G}_{12}$$

and thus, the score test statistic is given by

$$LM_n = \boldsymbol{S}_n^{(2)'}(\tilde{\boldsymbol{\theta}}_n)\widetilde{\boldsymbol{\Sigma}}_{MP}^{-1}\boldsymbol{S}_n^{(2)}(\tilde{\boldsymbol{\theta}}_n). \tag{5.12}$$

It can be shown that

$$\boldsymbol{S}_n^{(2)}(\tilde{\boldsymbol{\theta}}_n) \overset{o_p(1)}{=} \boldsymbol{W}_2 - \boldsymbol{G}_{21}\boldsymbol{G}_{11}^{-1}\boldsymbol{W}_1 \overset{H_0}{\sim} \mathcal{N}(\boldsymbol{0}, \widetilde{\boldsymbol{\Sigma}}_{MP}).$$

and therefore, $LM_n \xrightarrow{D} LM \overset{H_0}{\sim} \mathcal{X}^2_{m_2}$.

Assume now that the data are generated again from a Poisson model under the local Pitman-type alternatives $H_1 : \boldsymbol{\theta}^{(2)} = \boldsymbol{\theta}_0^{(2)} + n^{-1/2}\boldsymbol{\delta}$. In order to obtain the limiting distribution of the score test statistic (5.3) under the alternative, we only need the asymptotic distribution of the score function under H_1. By a Taylor expansion of $\boldsymbol{S}_n(\boldsymbol{\theta}_0)$ about $\boldsymbol{\theta}_n = \boldsymbol{\theta}_0 + n^{-1/2}\boldsymbol{\delta}^*$, where now $\boldsymbol{\delta}^*$ is a fixed vector in $\mathbb{R}^m_+$ of the form $\boldsymbol{\delta}^* = (\boldsymbol{\delta_1}, \boldsymbol{\delta})$, we have that

$$n^{-1/2}\boldsymbol{S}_n(\boldsymbol{\theta}_0) \overset{o_p(1)}{=} n^{-1/2}\boldsymbol{S}_n(\boldsymbol{\theta}_n) - n^{-1}\boldsymbol{H}_n(\boldsymbol{\theta}_n)\boldsymbol{\delta}^*.$$

Since $n^{-1/2}\boldsymbol{S}_n(\boldsymbol{\theta}_n) \xrightarrow{D} \mathcal{N}(\mathbf{0}, \boldsymbol{G}(\boldsymbol{\theta}_0))$ and $n^{-1}\boldsymbol{H}_n(\boldsymbol{\theta}_n)\boldsymbol{\delta}^* \xrightarrow{p} \boldsymbol{G}(\boldsymbol{\theta}_0)\boldsymbol{\delta}^*$ under the alternative, then $\boldsymbol{W} = n^{-1/2}\boldsymbol{S}_n(\boldsymbol{\theta}_0) \xrightarrow{D} \mathcal{N}(-\boldsymbol{G}(\boldsymbol{\theta}_0)\boldsymbol{\delta}^*, \boldsymbol{G}(\boldsymbol{\theta}_0))$. Therefore,

$$\boldsymbol{S}_n^{(2)}(\tilde{\boldsymbol{\theta}}_n) \overset{o_p(1)}{=} \boldsymbol{W}_2 - \boldsymbol{G}_{21}\boldsymbol{G}_{11}^{-1}\boldsymbol{W}_1 \overset{H_1}{\sim} \mathcal{N}(-\widetilde{\boldsymbol{\Sigma}}_P\boldsymbol{\delta}, \widetilde{\boldsymbol{\Sigma}}_P).$$

and considering again the expression of the score statistic given by (5.11), we have that $LM_n \xrightarrow{D} LM \overset{H_1}{\sim} \mathcal{X}^2_{m_2}(\boldsymbol{\delta}'\boldsymbol{\Delta}\boldsymbol{\delta})$, where for the case of the Poisson assumption $\boldsymbol{\Delta} = \widetilde{\boldsymbol{\Sigma}}_P = \boldsymbol{G}_{22} - \boldsymbol{G}_{21}\boldsymbol{G}_{11}^{-1}\boldsymbol{G}_{12}$ evaluated at $\tilde{\boldsymbol{\theta}}_n$.

If the data are generated from the mixed Poisson model, then following the same steps as for the Poisson and since now it holds that $n^{-1/2}\boldsymbol{S}_n(\boldsymbol{\theta}_n) \xrightarrow{D} \mathcal{N}(\mathbf{0}, \boldsymbol{G}_1(\boldsymbol{\theta}_0))$ and $n^{-1}\boldsymbol{H}_n(\boldsymbol{\theta}_n)\boldsymbol{\delta}^* \xrightarrow{p} \boldsymbol{G}(\boldsymbol{\theta}_0)\boldsymbol{\delta}^*$ under the alternative, we have that

$$\boldsymbol{W} = n^{-1/2}\boldsymbol{S}_n(\boldsymbol{\theta}_0) \xrightarrow{D} \mathcal{N}(-\boldsymbol{G}(\boldsymbol{\theta}_0)\boldsymbol{\delta}^*, \boldsymbol{G}_1(\boldsymbol{\theta}_0)).$$

Thus,

$$\boldsymbol{S}_n^{(2)}(\tilde{\boldsymbol{\theta}}_n) \overset{o_p(1)}{=} \boldsymbol{W}_2 - \boldsymbol{G}_{21}\boldsymbol{G}_{11}^{-1}\boldsymbol{W}_1 \overset{H_1}{\sim} \mathcal{N}(-\widetilde{\boldsymbol{\Sigma}}_P\boldsymbol{\delta}, \widetilde{\boldsymbol{\Sigma}}_{MP})$$

and considering again the expression of the score statistic given by (5.12), we have that $LM_n \xrightarrow{D} LM \overset{H_1}{\sim} \mathcal{X}^2_{m_2}(\boldsymbol{\delta}'\boldsymbol{\Delta}\boldsymbol{\delta})$, where for this case $\boldsymbol{\Delta} = \widetilde{\boldsymbol{\Sigma}}_P'\widetilde{\boldsymbol{\Sigma}}_{MP}^{-1}\widetilde{\boldsymbol{\Sigma}}_P$ evaluated at $\tilde{\boldsymbol{\theta}}_n$. □

Appendix B – R codes

Below we give the R code for the construction of Tables 5.1 and 5.2. Note that for reasons of brevity, we do not report again the functions which where defined in the previous chapters.

Construction of the top panel of Tables 5.1 and 5.2 for the results which are based on the Poisson distribution.

```
####################################
## Simulate the nonlinear model   ##
## using the Poisson distribution ##
####################################
nonlinpoisson.gamma.ts=function(d,a,b,gamma,size)
{
y=rep(NA,size)
mu=rep(NA,size)
mu[1]=1 #initial value
y[1]=rpois(1,mu[1])
for (t in 2:size)
{
mu[t]=d/((1+y[t-1])^(gamma))+a*mu[t-1]+b*y[t-1]
y[t]=rpois(1,mu[t])
}
return(cbind(y,mu))
}


#################################################
## (Poisson) Quasi-likelihood for the nonlinear  ##
##  model for gamma unknown                     ##
#################################################
liknonlin.poisson=function(theta,data)
{
lambda=rep(NA,length(data))
```

```
loglik=rep(NA,length(data))
lambda[1]=1
loglik[1]=0
for (t in 2:length(data))
{
lambda[t]=theta[1]/((1+data[t-1])^(theta[4]))+theta[2]*lambda[t-1]
         +theta[3]*data[t-1]
if (lambda[t]==0) loglik[t]= 0 else
if (lambda[t] >0) loglik[t]= -data[t]*log(lambda[t])+lambda[t]
}
final=sum(loglik)
}

##############################################
## Score function for the nonlinear model ##
## for gamma unknown                      ##
##############################################
scorenonlin.poisson=function(theta,data)
{
lambda=rep(NA,length(data))
lambda[1]=1
first=rep(NA,length(data))
first[1]=1
second=rep(NA,length(data))
second[1]=1
third=rep(NA,length(data))
third[1]=1
fourth=rep(NA,length(data))
fourth[1]=1
s1=rep(NA,length(data))
s2=rep(NA,length(data))
```

```
s3=rep(NA,length(data))
s4=rep(NA,length(data))
for (t in 2:length(data))
{
lambda[t]=theta[1]/((1+data[t-1])^(theta[4]))+theta[2]*lambda[t-1]
         +theta[3]*data[t-1]
first[t]=1/((1+data[t-1])^(theta[4]))+theta[2]*first[t-1]
second[t]=lambda[t-1]+theta[2]*second[t-1]
third[t]=theta[2]*third[t-1]+data[t-1]
fourth[t]=theta[2]*fourth[t-1]-theta[1]*log(1+data[t-1])
         *((1+data[t-1])^(-theta[4]))
s1[t]=-((data[t]/lambda[t])-1)*first[t]
s2[t]=-((data[t]/lambda[t])-1)*second[t]
s3[t]=-((data[t]/lambda[t])-1)*third[t]
s4[t]=-((data[t]/lambda[t])-1)*fourth[t]
}
ss1=sum(s1[-1])
ss2=sum(s2[-1])
ss3=sum(s3[-1])
ss4=sum(s4[-1])
score=c(ss1,ss2,ss3,ss4)
}

#####################################################################
## Information matrix based on the Poisson distribution-G matrix ##
#####################################################################
information.nonlinear.poisson=function(theta,data)
{
lambda=rep(NA,length(data))
lambda[1]=1
first=rep(NA,length(data))
```

```
first[1]=1
second=rep(NA,length(data))
second[1]=1
third=rep(NA,length(data))
third[1]=1
fourth=rep(NA,length(data))
fourth[1]=1
Information=matrix(0,nrow=4,ncol=4)
s1=rep(NA,length(data))
s2=rep(NA,length(data))
s3=rep(NA,length(data))
s4=rep(NA,length(data))
for (t in 2:length(data))
{
lambda[t]=theta[1]/((1+data[t-1])^(theta[4]))+theta[2]*lambda[t-1]
         +theta[3]*data[t-1]
first[t]=1/((1+data[t-1])^(theta[4]))+theta[2]*first[t-1]
second[t]=lambda[t-1]+theta[2]*second[t-1]
third[t]=theta[2]*third[t-1]+data[t-1]
fourth[t]=theta[2]*fourth[t-1]-theta[1]*log(1+data[t-1])
         *((1+data[t-1])^(-theta[4]))
s1[t]=first[t]
s2[t]=second[t]
s3[t]=third[t]
s4[t]=fourth[t]
var.comp=(1/sqrt(lambda[t]))*c(s1[t],s2[t],s3[t],s4[t])
Information=Information+var.comp%*%t(var.comp)
}
return(Information)
}
```

```
#########################
## QMLE of d, a and b  ##
#########################
library(MASS)
calculate.par=function(d,a,b,size,epsilon=0.001)
{
# Estimation
data.test=linearpoisson.ts(d,a,b,size)[501:size,1]
arma_fit=arima(data.test,order=c(1,0,1),method="CSS")
ma_1=min(max(arma_fit$coef["ma1"],-1+epsilon),0-epsilon)
ar_1=min(max(arma_fit$coef["ar1"],0+epsilon-ma_1),1-epsilon)
sigma_sq=max(arma_fit$sigma2,epsilon)
theta_init=c(sigma_sq*(1-ar_1),-ma_1,ar_1+ma_1)
# Estimation in H_0
results.linear=constrOptim(theta=theta_init,f=liklinear.poisson,
               grad=scorelinear.poisson,data=data.test,ui=uilinear,
               ci=cilinear,outer.iterations=100,outer.eps=1e-05,
               method="BFGS")$par
linear.hat.d=results.linear[1]
linear.hat.a=results.linear[2]
linear.hat.b=results.linear[3]
linear.est=c(linear.hat.d,linear.hat.a,linear.hat.b)
linear.hat.theta=c(linear.est,0)
# Score statistic
inf=information.nonlinear.poisson(theta=linear.hat.theta,data=data.test)
sco=scorenonlin.poisson(theta=linear.hat.theta,data=data.test)
score.statistic=sco[4]^{2}/(inf[4,4]-t(inf[4,1:3])%*%
                solve(inf[1:3,1:3])%*%inf[1:3,4])
return(list(linear.est,score.statistic))
}
```

```
##########################################################
## Now, we bootstrap the value of the test statistic. ##
## We need the previous function to get the QMLE and  ##
## the value of the test statistic.                   ##
## B is the number of bootstrap replications.         ##
##########################################################
boot.calculate.test=function(d,a,b,size,B)
{
step1=calculate.par(d,a,b,size)
param=step1[[1]]
LM.observed=step1[[2]]
LM.boot=rep(NA,B)
pval=rep(NA,B)
for (b1 in 1:B)
{
out1=calculate.par(param[1],param[2],param[3],size)
LM.boot[b1]=out1[[2]]
pval[b1]=ifelse(LM.boot[b1]>=LM.observed,1,0)
}
pvalue1=sum(pval)/(1+B)
pvalue2=(1+sum(pval))/(1+B)
result1=(pvalue1<c(0.10, 0.05, 0.01))
result2=(pvalue2<c(0.10, 0.05, 0.01))
return(list(result1,result2))
}

library(foreach)
library(doMC)
registerDoMC(cores=12)
sim.test=function(d,a,b,size,B,sim)
{
```

```
simulation.test=foreach(i=1:sim,.inorder=FALSE,.errorhandling="pass")%dopar%
{
cat("\n\n********** Now doing iteration",i,"of",sim,"**********\n\n")
pvalue.sim=boot.calculate.test(d,a,b,size,B)
write(pvalue.sim[[1]],file="test1.txt",ncolumns=3,append=TRUE)
write(pvalue.sim[[2]],file="test2.txt",ncolumns=3,append=TRUE)
return(pvalue.sim)
}
return(simulation.test)
}

output1=sim.test(1.5,0.05,0.6,1500,499,200)

rejectout1=matrix(NA,nrow=200,ncol=3)
rejectout2=matrix(NA,nrow=200,ncol=3)
for(i in 1:200)
{
rejectout1[i,]=output1[[i]][[1]]
rejectout2[i,]=output1[[i]][[2]]
}
round(100*apply(rejectout1,2,mean),3)
round(100*apply(rejectout2,2,mean),3)
```

Construction of the bottom panel of Tables 5.1 and 5.2 for the results which are based on the Poisson distribution.

```
calculate.par=function(d,a,b,size,sim,epsilon=0.001)
{
score.statistic=rep(NA,sim)
for (i in 1:sim)
{
# Estimation
```

```
data.test=linearpoisson.ts(d,a,b,size)[501:size,1]
arma_fit=arima(data.test,order=c(1,0,1),method="CSS")
ma_1=min(max(arma_fit$coef["ma1"],-1+epsilon),0-epsilon)
ar_1=min(max(arma_fit$coef["ar1"],0+epsilon-ma_1),1-epsilon)
sigma_sq=max(arma_fit$sigma2,epsilon)
theta_init=c(sigma_sq*(1-ar_1),-ma_1,ar_1+ma_1)
# Estimation in H_0
results.linear=constrOptim(theta=theta_init,f=liklinear.poisson,
               grad=scorelinear.poisson,data=data.test,ui=uilinear,
               ci=cilinear,outer.iterations=100,outer.eps=1e-05,
               method="BFGS")$par
linear.hat.d=results.linear[1]
linear.hat.a=results.linear[2]
linear.hat.b=results.linear[3]
linear.est=c(linear.hat.d,linear.hat.a,linear.hat.b)
linear.hat.theta=c(linear.est,0)
# Score statistic
inf=information.nonlinear.poisson(theta=linear.hat.theta,data=data.test)
sco=scorenonlin.poisson(theta=linear.hat.theta,data=data.test)
score.statistic[i]=sco[4]^{2}/(inf[4,4]-t(inf[4,1:3])%*%
                solve(inf[1:3,1:3])%*%inf[1:3,4])
}
return(score.statistic)
}

testnew1=calculate.par(1.5,0.05,0.6,1500,200)
mean(testnew1>qchisq(0.95,1))
```

Construction of the top panel of Tables 5.1 and 5.2 for the results which are based on the negative binomial distribution.

```
#####################################################################
```

```
## Information matrix based on the negative binomial distribution ##
## G_{1} matrix                                                    ##
#####################################################################
information.nonlinear.negbin=function(theta,data,nu)
{
lambda=rep(NA,length(data))
lambda[1]=1
first=rep(NA,length(data))
first[1]=1
second=rep(NA,length(data))
second[1]=1
third=rep(NA,length(data))
third[1]=1
fourth=rep(NA,length(data))
fourth[1]=1
Information=matrix(0,nrow=4,ncol=4)
s1=rep(NA,length(data))
s2=rep(NA,length(data))
s3=rep(NA,length(data))
s4=rep(NA,length(data))
for (t in 2:length(data))
{
lambda[t]=theta[1]/((1+data[t-1])^(theta[4]))+theta[2]*lambda[t-1]
        +theta[3]*data[t-1]
first[t]=1/((1+data[t-1])^(theta[4]))+theta[2]*first[t-1]
second[t]=lambda[t-1]+theta[2]*second[t-1]
third[t]=theta[2]*third[t-1]+data[t-1]
fourth[t]=theta[2]*fourth[t-1]-theta[1]*log(1+data[t-1])
         *((1+data[t-1])^(-theta[4]))
s1[t]= first[t]
s2[t]= second[t]
```

```
s3[t]= third[t]
s4[t]= fourth[t]
var.comp=(sqrt(1/lambda[t]+1/nu))*c(s1[t],s2[t],s3[t],s4[t])
Information=Information+var.comp%*%t(var.comp)
}
return(Information)
}

########################
## QMLE of d, a and b ##
########################
library(MASS)
calculate.par=function(d,a,b,size,nu,epsilon=0.001)
{
# Estimation
data.test=linearnegbin.ts(d,a,b,size,nu)[501:size,1]
arma_fit=arima(data.test,order=c(1,0,1),method="CSS")
ma_1=min(max(arma_fit$coef["ma1"],-1+epsilon),0-epsilon)
ar_1=min(max(arma_fit$coef["ar1"],0+epsilon-ma_1),1-epsilon)
sigma_sq=max(arma_fit$sigma2,epsilon)
theta_init=c(sigma_sq*(1-ar_1),-ma_1,ar_1+ma_1)
# Estimation in H_0
results.linear=constrOptim(theta=theta_init,f=liklinear.poisson,
                grad=scorelinear.poisson,data=data.test,ui=uilinear,
                ci=cilinear,outer.iterations=100,outer.eps=1e-05,
                method="BFGS")$par
linear.hat.d=results.linear[1]
linear.hat.a=results.linear[2]
linear.hat.b=results.linear[3]
linear.est=c(linear.hat.d,linear.hat.a,linear.hat.b)
linear.hat.theta=c(linear.est,0)
```

```
# Calculation of \hat{\lambda}
lambda=rep(NA,length(data.test))
lambda[1]=mean(data.test)
for (t in 2:length(data.test))
{
lambda[t]=linear.hat.d+linear.hat.a*lambda[t-1]
         +linear.hat.b*data.test[t-1]
}
# Estimation of \nu
est.nu2=theta.mm(data.test,lambda,length(data.test)-3)
# Score statistic-Breslow
A=information.nonlinear.poisson(theta=linear.hat.theta,data=data.test)
B=information.nonlinear.negbin(theta=linear.hat.theta,data=data.test,
  nu=est.nu2)
S=(B[4,4]-A[4,1:3]%*%solve(A[1:3,1:3])%*%B[1:3,4]-B[4,1:3]%*%
   solve(A[1:3,1:3])%*%A[1:3,4]+A[4,1:3]%*%solve(A[1:3,1:3])%*%
   B[1:3,1:3]%*%solve(A[1:3,1:3])%*%A[1:3,4])
U=scorenonlin.poisson(theta=linear.hat.theta,data=data.test)
score.statistic=((U[4])^2)/S
return(list(linear.est,score.statistic,est.nu2))
}

boot.calculate.test=function(d,a,b,size,nu,B)
{
step1=calculate.par(d,a,b,size,nu)
param=step1[[1]]
LM.observed=step1[[2]]
LM.boot=rep(NA,B)
pval=rep(NA,B)
for (b1 in 1:B)
{
```

```
out1=calculate.par(param[1],param[2],param[3],size,step1[[3]])
LM.boot[b1]=out1[[2]]
pval[b1]=ifelse(LM.boot[b1]>=LM.observed,1,0)
}
pvalue1=sum(pval)/(1+B)
pvalue2=(1+sum(pval))/(1+B)
result1=(pvalue1<c(0.10, 0.05, 0.01))
result2=(pvalue2<c(0.10, 0.05, 0.01))
return(list(result1,result2))
}

library(foreach)
library(doMC)
registerDoMC(cores=12)
sim.test=function(d,a,b,size,nu,B,sim)
{
simulation.test=foreach(i=1:sim,.inorder=FALSE,.errorhandling="pass")%dopar%
{
cat("\n\n********** Now doing iteration",i,"of",sim,"**********\n\n")
pvalue.sim=boot.calculate.test(d,a,b,size,nu,B)
write(pvalue.sim[[1]],file="test1.txt",ncolumns=3,append=TRUE)
write(pvalue.sim[[2]],file="test2.txt",ncolumns=3,append=TRUE)
return(pvalue.sim)
}
return(simulation.test)
}

output1=sim.test(1.5,0.05,0.6,1500,2,499,200)

rejectout1=matrix(NA,nrow=200,ncol=3)
rejectout2=matrix(NA,nrow=200,ncol=3)
```

```
for(i in 1:200)
{
rejectout1[i,]=output1[[i]][[1]]
rejectout2[i,]=output1[[i]][[2]]
}
round(100*apply(rejectout1,2,mean),3)
round(100*apply(rejectout2,2,mean),3)
```

Construction of the bottom panel of Tables 5.1 and 5.2 for the results which are based on the negative binomial distribution.

```
calculate.par=function(d,a,b,size,nu,sim,epsilon=0.001)
{
score.statistic=rep(NA,sim)
for (i in 1:sim)
{
# Estimation
data.test=linearnegbin.ts(d,a,b,size,nu)[501:size,1]
arma_fit=arima(data.test,order=c(1,0,1),method="CSS")
ma_1=min(max(arma_fit$coef["ma1"],-1+epsilon),0-epsilon)
ar_1=min(max(arma_fit$coef["ar1"],0+epsilon-ma_1),1-epsilon)
sigma_sq=max(arma_fit$sigma2,epsilon)
theta_init=c(sigma_sq*(1-ar_1),-ma_1,ar_1+ma_1)
# Estimation in H_0
results.linear=constrOptim(theta=theta_init,f=liklinear.poisson,
               grad=scorelinear.poisson,data=data.test,ui=uilinear,
               ci=cilinear,outer.iterations=100,
               outer.eps=1e-05,method="BFGS")$par
linear.hat.d=results.linear[1]
linear.hat.a=results.linear[2]
linear.hat.b=results.linear[3]
```

```
linear.est=c(linear.hat.d,linear.hat.a,linear.hat.b)
linear.hat.theta=c(linear.est,0)
# Calculation of \hat{\lambda}
lambda=rep(NA,length(data.test))
lambda[1]=mean(data.test)
for (t in 2:length(data.test))
{
lambda[t]=linear.hat.d+linear.hat.a*lambda[t-1]
         +linear.hat.b*data.test[t-1]
}
# Estimation of \nu
est.nu2=theta.mm(data.test,lambda,length(data.test)-3)
# Score statistic-Breslow
A=information.nonlinear.poisson(theta=linear.hat.theta,data=data.test)
B=information.nonlinear.negbin(theta=linear.hat.theta,data=data.test,
  nu=est.nu2)
S=(B[4,4]-A[4,1:3]%*%solve(A[1:3,1:3])%*%B[1:3,4]-B[4,1:3]%*%
   solve(A[1:3,1:3])%*%A[1:3,4]+A[4,1:3]%*%solve(A[1:3,1:3])%*%
   B[1:3,1:3]%*%solve(A[1:3,1:3])%*%A[1:3,4])
U=scorenonlin.poisson(theta=linear.hat.theta,data=data.test)
score.statistic[i]=t(U[4])%*%solve(S)%*%U[4]
}
return(score.statistic)
}


testnew1=calculate.par(1.5,0.05,0.6,1500,2,200)
mean(testnew1>qchisq(0.95,1))
```

For the construction of Table 5.3 we follow the same methodology as above, but now we generate data from the nonlinear model (5.1), that is

```
data.test=nonlinpoisson.gamma.ts(d,a,b,gamma,size)[501:size,1]
```

for the Poisson distributional assumption and

`data.test=nonlinnegbin.gamma.ts(d,a,b,gamma,size,nu)[501:size,1]`

for the negative binomial assumption.

Construction of Tables 5.5 and 5.6 for the results which are based on the Poisson distribution.

```
#####################################################
## Simulate the exponential autoregressive model ##
## using the Poisson distribution                ##
#####################################################
exparpoisson.ts=function(d,a,b,c,gamma,size)
{
y=rep(NA,size)
mu=rep(NA,size)
mu[1]=1 #initial value
y[1]=rpois(1,mu[1])
for (t in 2:size)
{
mu[t]=d+a*mu[t-1]+(b+c*exp(-gamma*(y[t-1])^2))*y[t-1]
y[t]=rpois(1,mu[t])
}
return(cbind(y,mu))
}

#############################################################
## (Poisson) Quasi-likelihood for the exponential model  ##
## for gamma known                                       ##
#############################################################
liknonlin.gamma.poisson=function(theta,gamma,data)
{
lambda=rep(NA,length(data))
```

```
loglik=rep(NA,length(data))
lambda[1]=1
loglik[1]=0
for (t in 2:length(data))
{
lambda[t]=theta[1]+theta[2]*lambda[t-1]+(theta[3]+theta[4]*
          exp(-gamma*(data[t-1])^{2}))*data[t-1]
if (lambda[t]==0) loglik[t]= 0 else
if (lambda[t] >0) loglik[t]= -data[t]*log(lambda[t])+lambda[t]
}
final=sum(loglik)
}

################################################
## Score function for the exponential model ##
## for gamma known                          ##
################################################
scorenonlin.gamma.poisson=function(theta,gamma,data)
{
lambda=rep(NA,length(data))
lambda[1]=1
first=rep(NA,length(data))
first[1]=1
second=rep(NA,length(data))
second[1]=1
third=rep(NA,length(data))
third[1]=1
fourth=rep(NA,length(data))
fourth[1]=1
s1=rep(NA,length(data))
s2=rep(NA,length(data))
```

```
s3=rep(NA,length(data))
s4=rep(NA,length(data))
for (t in 2:length(data))
{
lambda[t]=theta[1]+theta[2]*lambda[t-1]+(theta[3]+theta[4]*
          exp(-gamma*(data[t-1])^{2}))*data[t-1]
first[t]=1+theta[2]*first[t-1]
second[t]=lambda[t-1]+theta[2]*second[t-1]
third[t]=theta[2]*third[t-1]+data[t-1]
fourth[t]=theta[2]*fourth[t-1]+data[t-1]*exp(-gamma*(data[t-1])^2)
s1[t]=-((data[t]/lambda[t])-1)*first[t]
s2[t]=-((data[t]/lambda[t])-1)*second[t]
s3[t]=-((data[t]/lambda[t])-1)*third[t]
s4[t]=-((data[t]/lambda[t])-1)*fourth[t]
}
ss1=sum(s1[-1])
ss2=sum(s2[-1])
ss3=sum(s3[-1])
ss4=sum(s4[-1])
score=c(ss1,ss2,ss3,ss4)
}

#####################################################################
## Information matrix based on the Poisson distribution-G matrix ##
#####################################################################
information.gamma.poisson=function(theta,gamma,data)
{
lambda=rep(NA,length(data))
lambda[1]=1
first=rep(NA,length(data))
first[1]=1
```

```
second=rep(NA,length(data))
second[1]=1
third=rep(NA,length(data))
third[1]=1
fourth=rep(NA,length(data))
fourth[1]=1
Information=matrix(0,nrow=4,ncol=4)
s1=rep(NA,length(data))
s2=rep(NA,length(data))
s3=rep(NA,length(data))
s4=rep(NA,length(data))
for (t in 2:length(data))
{
lambda[t]=theta[1]+theta[2]*lambda[t-1]+(theta[3]+theta[4]*
          exp(-gamma*(data[t-1])^{2}))*data[t-1]
first[t]=1+theta[2]*first[t-1]
second[t]=lambda[t-1]+theta[2]*second[t-1]
third[t]=theta[2]*third[t-1]+data[t-1]
fourth[t]=theta[2]*fourth[t-1]+data[t-1]*exp(-gamma*(data[t-1])^2)
s1[t]=first[t]
s2[t]=second[t]
s3[t]=third[t]
s4[t]=fourth[t]
var.comp=(1/sqrt(lambda[t]))*c(s1[t],s2[t],s3[t],s4[t])
Information=Information+var.comp%*%t(var.comp)
}
return(Information)
}

library(MASS)
calculate.par=function(d,a,b,size,epsilon=0.001)
```

```
{
# Estimation
data.test=linearpoisson.ts(d,a,b,size)[501:size,1]
arma_fit=arima(data.test,order=c(1,0,1),method="CSS")
ma_1=min(max(arma_fit$coef["ma1"],-1+epsilon),0-epsilon)
ar_1=min(max(arma_fit$coef["ar1"],0+epsilon-ma_1),1-epsilon)
sigma_sq=max(arma_fit$sigma2,epsilon)
theta_init=c(sigma_sq*(1-ar_1),-ma_1,ar_1+ma_1)
# Estimation in H_0
results.linear=constrOptim(theta=theta_init,f=liklinear.poisson,
               grad=scorelinear.poisson,data=data.test,ui=uilinear,
               ci=cilinear,outer.iterations=100,
               outer.eps=1e-05,method="BFGS")$par
linear.hat.d=results.linear[1]
linear.hat.a=results.linear[2]
linear.hat.b=results.linear[3]
linear.est=c(linear.hat.d,linear.hat.a,linear.hat.b)
linear.hat.theta=c(linear.est,0)
# Grid for gamma
gam=seq(0.01,2,length=30)
score.stat=rep(NA,length(gam))
for(j in 1:length(gam))
{
# Score statistic
inf=information.gamma.poisson(theta=linear.hat.theta,gamma=gam[j],
    data=data.test)
sco=scorenonlin.gamma.poisson(theta=linear.hat.theta,gamma=gam[j],
    data=data.test)
score.stat[j]=sco[4]^{2}/(inf[4,4]-t(inf[4,1:3])%*%solve(inf[1:3,1:3])%*%
              inf[1:3,4])
}
```

```
score.statistic=max(score.stat)
return(list(linear.est,score.statistic))
}

boot.calculate.test=function(d,a,b,size,B)
{
step1=calculate.par(d,a,b,size)
param=step1[[1]]
LM.observed=step1[[2]]
LM.boot=rep(NA,B)
pval=rep(NA,B)
for (b1 in 1:B)
{
out1=calculate.par(param[1],param[2],param[3],size)
LM.boot[b1]=out1[[2]]
pval[b1]=ifelse(LM.boot[b1]>=LM.observed,1,0)
}
pvalue1=sum(pval)/(1+B)
pvalue2=(1+sum(pval))/(1+B)
result1=(pvalue1<c(0.10, 0.05, 0.01))
result2=(pvalue2<c(0.10, 0.05, 0.01))
return(list(result1,result2))
}

library(foreach)
library(doMC)
registerDoMC(cores=12)
sim.test=function(d,a,b,size,B,sim)
{
simulation.test=foreach(i=1:sim,.inorder=FALSE,.errorhandling="pass")%dopar%
{
```

```
cat("\n\n********** Now doing iteration",i,"of",sim,"**********\n\n")
pvalue.sim=boot.calculate.test(d,a,b,size,B)
write(pvalue.sim[[1]],file="test1.txt",ncolumns=3,append=TRUE)
write(pvalue.sim[[2]],file="test2.txt",ncolumns=3,append=TRUE)
return(pvalue.sim)
}
return(simulation.test)
}

result1=sim.test(0.5,0.3,0.2,1500,499,200)

rejectout1=matrix(NA,nrow=200,ncol=3)
rejectout2=matrix(NA,nrow=200,ncol=3)
for(i in 1:200)
{
rejectout1[i,]=result1[[i]][[1]]
rejectout2[i,]=result1[[i]][[2]]
}
round(100*apply(rejectout1,2,mean),3)
round(100*apply(rejectout2,2,mean),3)
```

Construction of Tables 5.5 and 5.6 for the results which are based on the negative binomial distribution.

```
#####################################################
## Simulate the exponential autoregressive model ##
## using the negative binomial distribution     ##
#####################################################
exparnegbin.ts=function(d,a,b,c,gamma,size,nu)
{
y=rep(NA,size)
mu=rep(NA,size)
```

```
mu[1]=1 #initial value
y[1]=rnbinom(1,size=nu,mu=mu[1])
for (t in 2:size)
{
mu[t]=d+a*mu[t-1]+(b+c*exp(-gamma*(y[t-1])^2))*y[t-1]
y[t]=rnbinom(1,size=nu,mu=mu[t])
}
return(cbind(y,mu))
}

#########################################################################
## Information matrix based on the negative binomial distribution ##
## G_{1} matrix                                                        ##
#########################################################################
information.gamma.negbin=function(theta,gamma,data,nu)
{
lambda=rep(NA,length(data))
lambda[1]=1
first=rep(NA,length(data))
first[1]=1
second=rep(NA,length(data))
second[1]=1
third=rep(NA,length(data))
third[1]=1
fourth=rep(NA,length(data))
fourth[1]=1
Information=matrix(0,nrow=4,ncol=4)
s1=rep(NA,length(data))
s2=rep(NA,length(data))
s3=rep(NA,length(data))
s4=rep(NA,length(data))
```

```
for (t in 2:length(data))
{
lambda[t]=theta[1]+theta[2]*lambda[t-1]+(theta[3]+theta[4]*
          exp(-gamma*(data[t-1])^{2}))*data[t-1]
first[t]=1+theta[2]*first[t-1]
second[t]=lambda[t-1]+theta[2]*second[t-1]
third[t]=theta[2]*third[t-1]+data[t-1]
fourth[t]=theta[2]*fourth[t-1]+data[t-1]*exp(-gamma*(data[t-1])^2)
s1[t]=first[t]
s2[t]=second[t]
s3[t]=third[t]
s4[t]=fourth[t]
var.comp=(sqrt(1/lambda[t]+1/nu))*c(s1[t],s2[t],s3[t],s4[t])
Information=Information+var.comp%*%t(var.comp)
}
return(Information)
}

library(MASS)
calculate.par=function(d,a,b,size,nu,epsilon=0.001)
{
# Estimation
data.test=linearnegbin.ts(d,a,b,size,nu)[501:size,1]
arma_fit=arima(data.test,order=c(1,0,1),method="CSS")
ma_1=min(max(arma_fit$coef["ma1"],-1+epsilon),0-epsilon)
ar_1=min(max(arma_fit$coef["ar1"],0+epsilon-ma_1),1-epsilon)
sigma_sq=max(arma_fit$sigma2,epsilon)
theta_init=c(sigma_sq*(1-ar_1),-ma_1,ar_1+ma_1)
# Estimation in H_0
results.linear=constrOptim(theta=theta_init,f=liklinear.poisson,
               grad=scorelinear.poisson,data=data.test,ui=uilinear,
```

```
                ci=cilinear,outer.iterations=100,
                outer.eps=1e-05,method="BFGS")$par
linear.hat.d=results.linear[1]
linear.hat.a=results.linear[2]
linear.hat.b=results.linear[3]
linear.est=c(linear.hat.d,linear.hat.a,linear.hat.b)
linear.hat.theta=c(linear.est,0)
# Calculation of \hat{\lambda}
lambda=rep(NA, length(data.test))
lambda[1]=mean(data.test)
for (t in 2:length(data.test))
{
lambda[t]=linear.hat.d+linear.hat.a*lambda[t-1]+linear.hat.b*data.test[t-1]
}
# Estimation of \nu
est.nu2=theta.mm(data.test,lambda,length(data.test)-3)
# Grid for gamma
gam=seq(0.01,2,length=30)
score.stat=rep(NA,length(gam))
for(j in 1:length(gam))
{
# Score statistic-Breslow
A=information.gamma.poisson(theta=linear.hat.theta,gamma=gam[j],
  data=data.test)
B=information.gamma.negbin(theta=linear.hat.theta,gamma=gam[j],
  data=data.test,nu=est.nu2)
S=(B[4,4]-A[4,1:3]%*%solve(A[1:3,1:3])%*%B[1:3,4]-B[4,1:3]%*%
   solve(A[1:3,1:3])%*%A[1:3,4]+A[4,1:3]%*%solve(A[1:3,1:3])%*%
   B[1:3,1:3]%*%solve(A[1:3,1:3])%*%A[1:3,4])
U=scorenonlin.gamma.poisson(theta=linear.hat.theta,gamma=gam[j],
  data=data.test)
```

```
score.stat[j]=((U[4])^2)/S
}
score.statistic=max(score.stat)
return(list(linear.est,score.statistic,est.nu2))
}

boot.calculate.test=function(d,a,b,size,nu,B)
{
step1=calculate.par(d,a,b,size,nu)
param=step1[[1]]
LM.observed=step1[[2]]
LM.boot=rep(NA,B)
pval=rep(NA,B)
for (b1 in 1:B)
{
out1=calculate.par(param[1],param[2],param[3],size,step1[[3]])
LM.boot[b1]=out1[[2]]
pval[b1]=ifelse(LM.boot[b1]>=LM.observed,1,0)
}
pvalue1=sum(pval)/(1+B)
pvalue2=(1+sum(pval))/(1+B)
result1=(pvalue1<c(0.10, 0.05, 0.01))
result2=(pvalue2<c(0.10, 0.05, 0.01))
return(list(result1,result2))
}

library(foreach)
library(doMC)
registerDoMC(cores=12)
sim.test=function(d,a,b,size,nu,B,sim)
{
```

```
simulation.test=foreach(i=1:sim,.inorder=FALSE,.errorhandling="pass")%dopar%
{
cat("\n\n********** Now doing iteration",i,"of",sim,"**********\n\n")
pvalue.sim=boot.calculate.test(d,a,b,size,nu,B)
write(pvalue.sim[[1]],file="test1.txt",ncolumns=3,append=TRUE)
write(pvalue.sim[[2]],file="test2.txt",ncolumns=3,append=TRUE)
return(pvalue.sim)
}
return(simulation.test)
}

result1=sim.test(0.5,0.3,0.2,1500,2,499,200)

rejectout1=matrix(NA,nrow=200,ncol=3)
rejectout2=matrix(NA,nrow=200,ncol=3)
for(i in 1:200)
{
rejectout1[i,]=result1[[i]][[1]]
rejectout2[i,]=result1[[i]][[2]]
}
round(100*apply(rejectout1,2,mean),3)
round(100*apply(rejectout2,2,mean),3)
```

For the construction of Table 5.7 we follow the same methodology as above, but now we generate data from the exponential model (5.2), that is

`data.test=exparpoisson.ts(d,a,b,c,gamma,size)[501:size,1]`

for the Poisson distributional assumption and

`data.test=exparnegbin.ts(d,a,b,c,gamma,size,nu)[501:size,1]`

for the negative binomial assumption.

In addition, the linear constrains that we take into account are shown below.

```
################################################
```

```
## Constrains to obtain the QMLE (gamma known)  ##
## d>0, a>0, b>0, c>0, 0<a+b+c<1                 ##
################################################
uiexpar.gamma=matrix(0,nrow=5,ncol=4) #matrix for the linear constraints
uiexpar.gamma[1,1]=1
uiexpar.gamma[2,2]=1
uiexpar.gamma[3,3]=1
uiexpar.gamma[4,4]=1
uiexpar.gamma[5,2]=-1
uiexpar.gamma[5,3]=-1
uiexpar.gamma[5,4]=-1
ciexpar.gamma=rep(0,5) #constant vector for the linear constraints
ciexpar.gamma[5]=-1
```

Construction of the top panel of Tables 5.8, 5.9 and 5.10 regarding the Poisson distributional assumption.

```
########################
## Transactions data ##
########################
ericsson=read.table("ericssonfirstday.txt",header=T,dec=",",as.is=F)
names(ericsson)
transactions=ericsson$trans
ydata=transactions

##################
## Measles data ##
##################
data=read.table("ewcitmeas.txt",header=T,dec=",",as.is=F)
names(data)
ydata=data$Sheffield[1601:2050]
```

```
####################
## Homicides data ##
####################
data=read.table("homicides.txt",header=T,dec=",",as.is=F)
names(data)
ydata=data$X0

########################
## QMLE of d, a and b ##
########################
library(MASS)
epsilon=0.001
# Estimation
data.test=ydata
arma_fit=arima(data.test,order=c(1,0,1),method="CSS")
ma_1=min(max(arma_fit$coef["ma1"],-1+epsilon),0-epsilon)
ar_1=min(max(arma_fit$coef["ar1"],0+epsilon-ma_1),1-epsilon)
sigma_sq=max(arma_fit$sigma2,epsilon)
theta_init=c(sigma_sq*(1-ar_1),-ma_1,ar_1+ma_1)
# Estimation in H_0
results.linear=constrOptim(theta=theta_init,f=liklinear.poisson,
            grad=scorelinear.poisson,data=data.test,ui=uilinear,
            ci=cilinear,outer.iterations=100,
            outer.eps=1e-05,method="BFGS")$par
linear.hat.d=results.linear[1]
linear.hat.a=results.linear[2]
linear.hat.b=results.linear[3]
linear.est=c(linear.hat.d,linear.hat.a,linear.hat.b)
linear.hat.theta=c(linear.est,0)
# Score statistic
inf=information.nonlinear.poisson(theta=linear.hat.theta,data=data.test)
```

```
sco=scorenonlin.poisson(theta=linear.hat.theta,data=data.test)
score.statistic=sco[4]^{2}/(inf[4,4]-t(inf[4,1:3])%*%solve(inf[1:3,1:3])
                %*%inf[1:3,4])
score.statistic


calculate.par=function(d,a,b,size,epsilon=0.001)
{
# Estimation
data.test=linearpoisson.ts(d,a,b,size)[501:size,1]
arma_fit=arima(data.test,order=c(1,0,1),method="CSS")
ma_1=min(max(arma_fit$coef["ma1"],-1+epsilon),0-epsilon)
ar_1=min(max(arma_fit$coef["ar1"],0+epsilon-ma_1),1-epsilon)
sigma_sq=max(arma_fit$sigma2,epsilon)
theta_init=c(sigma_sq*(1-ar_1),-ma_1,ar_1+ma_1)
# Estimation in H_0
results.linear=constrOptim(theta=theta_init,f=liklinear.poisson,
               grad=scorelinear.poisson,data=data.test,ui=uilinear,
               ci=cilinear,outer.iterations=100,
               outer.eps=1e-05,method="BFGS")$par
linear.hat.d=results.linear[1]
linear.hat.a=results.linear[2]
linear.hat.b=results.linear[3]
linear.est=c(linear.hat.d,linear.hat.a,linear.hat.b)
linear.hat.theta=c(linear.est,0)
# Score statistic
inf=information.nonlinear.poisson(theta=linear.hat.theta,data=data.test)
sco=scorenonlin.poisson(theta=linear.hat.theta,data=data.test)
score.statistic=sco[4]^{2}/(inf[4,4]-t(inf[4,1:3])%*%solve(inf[1:3,1:3])%*%
               inf[1:3,4])
return(list(linear.est,score.statistic))
}
```

```
boot.calculate.test=function(size,B)
{
param=linear.est
LM.observed=score.statistic
LM.boot=rep(NA,B)
pval=rep(NA,B)
for (b1 in 1:B)
{
out1=calculate.par(param[1],param[2],param[3],size)
LM.boot[b1]=out1[[2]]
pval[b1]=ifelse(LM.boot[b1]>=LM.observed,1,0)
}
pvalue1=sum(pval)/(1+B)
pvalue2=(1+sum(pval))/(1+B)
result1=(pvalue1<c(0.10, 0.05, 0.01))
result2=(pvalue2<c(0.10, 0.05, 0.01))
return(list(result1,result2))
}

library(foreach)
library(doMC)
registerDoMC(cores=12)
sim.test=function(size,B,sim)
{
simulation.test=foreach(i=1:sim,.inorder=FALSE,.errorhandling="pass")%dopar%
{
cat("\n\n********** Now doing iteration",i,"of",sim,"**********\n\n")
pvalue.sim=boot.calculate.test(size,B)
write(pvalue.sim[[1]],file="test1.txt",ncolumns=3,append=TRUE)
write(pvalue.sim[[2]],file="test2.txt",ncolumns=3,append=TRUE)
```

```
return(pvalue.sim)
}
return(simulation.test)
}

out=sim.test(960,499,200)

rejectout1=matrix(NA,nrow=200,ncol=3)
rejectout2=matrix(NA,nrow=200,ncol=3)
for(i in 1:200)
{
rejectout1[i,]=out[[i]][[1]]
rejectout2[i,]=out[[i]][[2]]
}
round(100*apply(rejectout1,2,mean),3)
round(100*apply(rejectout2,2,mean),3)
```

Construction of the top panel of Tables 5.8, 5.9 and 5.10 regarding the negative binomial distributional assumption.

```
########################
## QMLE of d, a and b ##
########################
library(MASS)
epsilon=0.001
# Estimation
data.test=ydata
arma_fit=arima(data.test,order=c(1,0,1),method="CSS")
ma_1=min(max(arma_fit$coef["ma1"],-1+epsilon),0-epsilon)
ar_1=min(max(arma_fit$coef["ar1"],0+epsilon-ma_1),1-epsilon)
sigma_sq=max(arma_fit$sigma2,epsilon)
theta_init=c(sigma_sq*(1-ar_1),-ma_1,ar_1+ma_1)
```

```
# Estimation in H_0
results.linear=constrOptim(theta=theta_init,f=liklinear.poisson,
               grad=scorelinear.poisson,data=data.test,ui=uilinear,
               ci=cilinear,outer.iterations=100,
               outer.eps=1e-05,method="BFGS")$par
linear.hat.d=results.linear[1]
linear.hat.a=results.linear[2]
linear.hat.b=results.linear[3]
linear.est=c(linear.hat.d,linear.hat.a,linear.hat.b)
linear.hat.theta=c(linear.est,0)
# Calculation of \hat{\lambda}
lambda=rep(NA, length(data.test))
lambda[1]=mean(data.test)
for (t in 2:length(data.test))
{
lambda[t]=linear.hat.d+linear.hat.a*lambda[t-1]+linear.hat.b*data.test[t-1]
}
# Estimation of \nu
est.nu2=theta.mm(data.test,lambda,length(data.test)-3)
# Score statistic-Breslow
A=information.nonlinear.poisson(theta=linear.hat.theta,data=data.test)
B=information.nonlinear.negbin(theta=linear.hat.theta,data=data.test,
  nu=est.nu2)
S=(B[4,4]-A[4,1:3]%*%solve(A[1:3,1:3])%*%B[1:3,4]-B[4,1:3]%*%
   solve(A[1:3,1:3])%*%A[1:3,4]+A[4,1:3]%*%solve(A[1:3,1:3])%*%
   B[1:3,1:3]%*%solve(A[1:3,1:3])%*%A[1:3,4])
U=scorenonlin.poisson(theta=linear.hat.theta,data=data.test)
score.statistic=((U[4])^2)/S
score.statistic

calculate.par=function(d,a,b,size,nu,epsilon=0.001)
```

```
{
# Estimation
data.test=linearnegbin.ts(d,a,b,size,nu)[501:size,1]
arma_fit=arima(data.test,order=c(1,0,1),method="CSS")
ma_1=min(max(arma_fit$coef["ma1"],-1+epsilon),0-epsilon)
ar_1=min(max(arma_fit$coef["ar1"],0+epsilon-ma_1),1-epsilon)
sigma_sq=max(arma_fit$sigma2,epsilon)
theta_init=c(sigma_sq*(1-ar_1),-ma_1,ar_1+ma_1)
# Estimation in H_0
results.linear=constrOptim(theta=theta_init,f=liklinear.poisson,
               grad=scorelinear.poisson,data=data.test,ui=uilinear,
               ci=cilinear,outer.iterations=100,
               outer.eps=1e-05,method="BFGS")$par
linear.hat.d=results.linear[1]
linear.hat.a=results.linear[2]
linear.hat.b=results.linear[3]
linear.est=c(linear.hat.d,linear.hat.a,linear.hat.b)
linear.hat.theta=c(linear.est,0)
# Calculation of \hat{\lambda}
lambda=rep(NA, length(data.test))
lambda[1]=mean(data.test)
for (t in 2:length(data.test))
{
lambda[t]=linear.hat.d+linear.hat.a*lambda[t-1]+linear.hat.b*data.test[t-1]
}
# Estimation of \nu
est.nu2=theta.mm(data.test,lambda,length(data.test)-3)
# Score statistic-Breslow
A=information.nonlinear.poisson(theta=linear.hat.theta,data=data.test)
B=information.nonlinear.negbin(theta=linear.hat.theta,data=data.test,
  nu=est.nu2)
```

```
S=(B[4,4]-A[4,1:3]%*%solve(A[1:3,1:3])%*%B[1:3,4]-B[4,1:3]%*%
   solve(A[1:3,1:3])%*%A[1:3,4]+A[4,1:3]%*%solve(A[1:3,1:3])%*%
   B[1:3,1:3]%*%solve(A[1:3,1:3])%*%A[1:3,4])
U=scorenonlin.poisson(theta=linear.hat.theta,data=data.test)
score.statistic=t(U[4])%*%solve(S)%*%U[4]
return(list(linear.est,score.statistic,est.nu2))
}

boot.calculate.test=function(size,B)
{
param=linear.est
LM.observed=score.statistic
LM.boot=rep(NA,B)
pval=rep(NA,B)
for (b1 in 1:B)
{
out1=calculate.par(param[1],param[2],param[3],size,est.nu2)
LM.boot[b1]=out1[[2]]
pval[b1]=ifelse(LM.boot[b1]>=LM.observed,1,0)
}
pvalue1=sum(pval)/(1+B)
pvalue2=(1+sum(pval))/(1+B)
result1=(pvalue1<c(0.10, 0.05, 0.01))
result2=(pvalue2<c(0.10, 0.05, 0.01))
return(list(result1,result2))
}

library(foreach)
library(doMC)
registerDoMC(cores=12)
sim.test=function(size,B,sim)
```

```
{
simulation.test=foreach(i=1:sim,.inorder=FALSE,.errorhandling="pass")%dopar%
{
cat("\n\n********** Now doing iteration",i,"of",sim,"**********\n\n")
pvalue.sim=boot.calculate.test(size,B)
write(pvalue.sim[[1]],file="test1.txt",ncolumns=3,append=TRUE)
write(pvalue.sim[[2]],file="test2.txt",ncolumns=3,append=TRUE)
return(pvalue.sim)
}
return(simulation.test)
}

out=sim.test(960,499,200)
```

For the construction of the bottom panel of Tables 5.8, 5.9 and 5.10 we work analogously using all the appropriate functions regarding the exponential model (5.2).
For the construction of Figure 5.1 we use the following R code.

```
library(MASS)
calculate.par=function(d,a,b,gamma,size,nu,sim,epsilon=0.001)
{
score.statistic=rep(NA,sim)
for (i in 1:sim)
{
# Estimation
data.test=nonlinnegbin.gamma.ts(d,a,b,gamma,size,nu)[501:size,1]
arma_fit=arima(data.test,order=c(1,0,1),method="CSS")
ma_1=min(max(arma_fit$coef["ma1"],-1+epsilon),0-epsilon)
ar_1=min(max(arma_fit$coef["ar1"],0+epsilon-ma_1),1-epsilon)
sigma_sq=max(arma_fit$sigma2,epsilon)
theta_init=c(sigma_sq*(1-ar_1),-ma_1,ar_1+ma_1)
# Estimation in H_0
```

```
results.linear=constrOptim(theta=theta_init,f=liklinear.poisson,
               grad=scorelinear.poisson,data=data.test,ui=uilinear,
               ci=cilinear,outer.iterations=100,
               outer.eps=1e-05,method="BFGS")$par
linear.hat.d=results.linear[1]
linear.hat.a=results.linear[2]
linear.hat.b=results.linear[3]
linear.est=c(linear.hat.d,linear.hat.a,linear.hat.b)
linear.hat.theta=c(linear.est,0)
# Calculation of \hat{\lambda}
lambda=rep(NA,length(data.test))
lambda[1]=mean(data.test)
for (t in 2:length(data.test))
{
lambda[t]=linear.hat.d+linear.hat.a*lambda[t-1]+linear.hat.b*data.test[t-1]
}
# Estimation of \nu
est.nu2=theta.mm(data.test,lambda,length(data.test)-3)
# Score statistic-Breslow
A=information.nonlinear.poisson(theta=linear.hat.theta,data=data.test)
B=information.nonlinear.negbin(theta=linear.hat.theta,data=data.test,
  nu=est.nu2)
S=(B[4,4]-A[4,1:3]%*%solve(A[1:3,1:3])%*%B[1:3,4]-B[4,1:3]%*%
   solve(A[1:3,1:3])%*%A[1:3,4]+A[4,1:3]%*%solve(A[1:3,1:3])%*%
   B[1:3,1:3]%*%solve(A[1:3,1:3])%*%A[1:3,4])
U=scorenonlin.poisson(theta=linear.hat.theta,data=data.test)
score.statistic[i]=((U[4])^2)/S
}
return(score.statistic)
}
```

```
gamma=c(0.05,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1)

f=function(d,a,b,size,nu,sim)
{
gam=gamma
results=rep(NA,length(gam))
res=rep(NA,sim)
for(i in 1:length(gam))
{
test1=calculate.par(d,a,b,gam[i],size,nu,sim)
res=ifelse(test1>qchisq(0.95,1),1,0)
results[i]=mean(res)
}
return(results)
}

power2=f(1.5,0.05,0.6,1500,4,200)

plot(gamma,power2,type="n",xlim=c(0,1.1),ylim=c(0,1.04),
     xlab=expression(gamma),ylab="Power",main="Power of the test")
lines(lowess(gamma,power2,f=0.5))
```

Chapter 6

Conclusions and Further Research

6.1 Conclusions

The first part of this thesis suggests a quite general approach for modeling and inference about count time series models. We have introduced mixed Poisson processes for modeling count time series. The main observation is that, if the mean of the process is correctly specified, then existing results about Poisson count time series can be transferred to this case without any major obstacles. In particular, concerning mixed Poisson count time series, we have given ways for data modeling and we have proved ergodicity and existence of moments. These results are instrumental in studying large sample behavior of maximum likelihood based estimators. In principle, existence of moments guarantees application of a central limit theorem for martingales and assures stability of the Hessian matrix. We have chosen to work with the Poisson log-likelihood as a quasi-likelihood for the mixed Poisson process, provided that the expectation of the mixing variable Z is equal to 1 (recall (3.1)). This is so because the main ingredient of asymptotic theory is the correct mean specification form. If the mean process has been correctly specified and moments of the joint process $\{(Y_t, \lambda_t)\}$ exist, then we can advance the theory further by considering GARCH type models for compound or mixed Poisson processes. Regardless of the chosen distribution, we have shown that this class of models and maximum likelihood theory provide a systematic framework for the analysis of integer-valued time series data. This theory can be extended to the class of exponential dispersion model by employing a suitable variance function; see Kokonendji et al. [58]. Estimation, diagnostics, model assessment,

and forecasting are implemented in a straightforward manner, where the computation is carried out by standard software. These issues are addressed in the list of desiderata suggested by Zeger and Qaqish [76] and Davis et al. [18].

The second part of this work studies the prediction problem in the context of count time series. More specifically, we follow the recent methodology of Czado et al. [13], where various tools for predictive model assessment are developed for independent but not identically distributed data. We show that these methods can also be applied for count dependent data. We focus on Poisson and negative binomial distributions since these are occurred in applications more frequently (and they fall under the mixed Poisson class); however the methods can be applied to other discrete distributions provided that they are suitably parameterized, as discussed in Chapter 3. We assess the predictive performance of the proposed models by extending the tools developed by Czado et al. [13]. More specifically, we address the problem of examining probabilistic calibration, marginal calibration and sharpness of the proposed models. We propose the use of probability integral transformation histogram (PIT), marginal calibration plot and scoring rules to assess the predictive performance and rank the competing forecast models.

The third part of this thesis investigates testing for the class of the mixed Poisson autoregressions. Our main focus is the development of testing procedures for testing linearity of the assumed model. Towards this goal, we study the behavior of the score test, suitably adjusted to take into account the misspecification of the log-likelihood function. Other techniques (Wald test, likelihood ratio test) might be developed and compared with our approach. However, the score test provides a convenient framework for testing hypotheses for this type of models.

In particular, we consider testing linearity against two special classes of nonlinear alternatives for count time series data. The first class contains of models which do not face the problem of non identifiability, that is all the parameters of the model are identified under the null hypothesis. For this class of models and under the null hypothesis of linearity, the score test statistic possesses an asymptotic $\mathcal{X}^2$ distribution. The second class of nonlinear models consists of models in which a nonnegative nuisance parameter exists under the alternative hypothesis but not when linearity holds. In this particular case the testing problem is nonstandard and the classical asymptotic theory for the score

test does not apply.

6.2 Further Research

Several extensions of this work can be discussed.

In this thesis, we focus exclusively on models of order $p = q = 1$, recall again (3.5). An obvious generalization of the mean specification in (3.5) is given by (3.4), where now the $f(\cdot, \cdot)$ function is such that $f : \mathbb{N}_0^p \times \mathbb{R}_+^q \to (0, \infty)$. It is clear that models (3.11), (3.12), (5.1) and (5.2) can be extended according to (3.4). Such examples are provided by the class of smooth transition autoregressive models of which the exponential autoregressive model is a special case (cf. Teräsvirta [70] and Teräsvirta et al. [71]). Further examples of nonlinear time series models can be found in Tong [72] and Fan and Yao [30]. These models have not been considered earlier in the literature in the context of generalized linear models for count time series, and they provide a flexible framework for studying dependent count data.

Some alternative nonlinear specifications for model (3.5) that can be potentially considered are given below,

$$\begin{aligned}
\lambda_t &= d + a_1\lambda_{t-1} + b_1 Y_{t-1} + c_1\lambda_{t-1}Y_{t-1}, \\
\lambda_t &= d + \Big(a_1 + \frac{c_1}{1+\exp(-\lambda_{t-1})}\Big)\lambda_{t-1} + b_1 Y_{t-1}, \\
\lambda_t &= d + a_1\lambda_{t-1} + (b_1 + c_1(1-\exp(-\gamma Y_{t-1}^2))Y_{t-1}, \\
\lambda_t &= d + a_1\lambda_{t-1} + \Big(b_1 + c_1\Big(\frac{1}{1+\exp(-\gamma Y_{t-1})} - \frac{1}{2}\Big)\Big)Y_{t-1}.
\end{aligned}$$

As empirical experience has shown, models like those we discuss in this work, can successfully accommodate count dependent data, especially when there exists positive autocorrelation and there are no covariates. However, in applications we observe time series data that might be negatively correlated, and occasionally, with possibly additional covariates. In this case, the choice of the logarithmic function is the most popular among the link functions for modeling count data. In fact, this choice corresponds to the canonical link model according to the terminology of generalized linear models.

Recall that $\{Y_t\}$ denotes a count time series and let $\nu_t = \log\lambda_t$ following the notation

introduced in (3.1). A log-linear model of order (1,1) with feedback for the analysis of count time series is defined as

$$\nu_t = d + a_1\nu_{t-1} + b_1 \log(Y_{t-1} + 1).$$

In general, the parameters d, a_1, b_1 can be positive or negative but they need to satisfy certain conditions so that we obtain a stationary time series. For more details, see Fokianos and Tjøstheim [36] and Fokianos [32].

Another interesting topic of research is the analysis of multivariate count time series models; see Liu [61], Pedeli and Karlis [67] and the fourth part of this volume which contains many interesting results. The main issue for attacking the problem of multivariate count time series is that multivariate count distributions are quite complex to be analyzed by maximum likelihood methods.

Assume that $\{\boldsymbol{Y}_t = (Y_{i,t}), t = 1, 2, \ldots, n\}$ denotes a $p-$dimensional count time series and suppose further that $\{\boldsymbol{\lambda}_t = (\lambda_{i,t}), t = 1, 2, \ldots, n\}$ is a corresponding $p-$dimensional intensity process. Here the notation p denotes dimension but not order as in (3.4). Then, a natural generalization of (3.6) is given by

$$Y_{i,t} = N_{i,t}(0, \lambda_{i,t}],\ i = 1, \cdots, p, \quad \boldsymbol{\lambda}_t = \boldsymbol{d} + \boldsymbol{A}\boldsymbol{\lambda}_{t-1} + \boldsymbol{B}\boldsymbol{Y}_{t-1}, \tag{6.1}$$

where $\boldsymbol{d}$ is a $p-$dimensional vector and $\boldsymbol{A}, \boldsymbol{B}$ are $p \times p$ matrices, all of them unknowns to be estimated. Model (6.1) is a direct extension of the linear autoregressive model (3.6) and assumes that marginally the count process is Poisson distributed. However the statistical problem is dealing with the joint distribution of the vector process $\{\boldsymbol{Y}_t\}$ and this requires further research; some preliminary results about ergodicity and stationarity of (6.1) have been obtained by Liu [61]. This work can be extended to the mixed Poisson framework.

Furthermore, we note that the problem of obtaining the asymptotic distribution of the supremum test is worth further investigation, like in Francq et al. [38, Theorem 4.1]. In addition, the parametric bootstrap procedure which was suggested to approximate its large sample behavior need to be studied thoroughly. Towards this goal, we note the recent work by Fokianos and Neumann [33] who proved the validity of parametric bootstrap for testing goodness of fit under the standard Poisson model.

Last but not least, further research in the field of constrained optimization problem can be discussed. Optimization algorithms should be improved in order to give us more accurate results for the estimators of the unknown parameters, specifically for complicated nonlinear formulas.

Bibliography

[1] Andrews, D. W. K. (2001). Testing when a parameter is on the boundary of the maintained hypothesis. *Econometrica*, 69:683–734.

[2] Andrews, D. W. K. and Ploberger, W. (1994). Optimal tests when a nuisance parameter is present only under the alternative. *Econometrica*, 62:1383–1414.

[3] Bardet, J.-M. and Wintenberger, O. (2009). Asymptotic normality of the quasi-maximum likelihood estimator for multidimensional causal processes. *The Annals of Statistics*, 37:2730–2759.

[4] Berkes, I., Horváth, L., and Kokoszka, P. (2003). GARCH processes: structure and estimation. *Bernoulli*, 9:201–227.

[5] Bollerslev, T. (1986). Generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics*, 31:307–327.

[6] Boos, D. D. (1992). On generalized score tests. *The American Statistician*, 46:327–333.

[7] Breslow, N. (1990). Tests of hypotheses in overdispersed poisson regression and other quasi-likelihood models. *Journal of the American Statistical Association*, 85:565–571.

[8] Brockwell, P. J. and Davis, R. A. (1991). *Time Series: Theory and Methods*. Springer-Verlag, New York. Second Edition.

[9] Brumback, B. A., Ryan, L. M., Schwartz, J. D., Neas, L. M., Stark, P. C., and Burge, H. A. (2000). Transitional regression models with application to environmental time series. *Journal of the American Statistical Association*, 85:16–27.

[10] Cameron, A. C. and Trivedi, P. K. (1998). *Regression Analysis of Count Data*. Cambridge University Press, Cambridge. First Edition.

[11] Christou, V. and Fokianos, K. (2013). Quasi–likelihood inference for negative binomial time series models. *Journal of Time Series Analysis*. To appear.

[12] Cox, D. R. (1981). Statistical analysis of time series: Some recent developments. *Scandinavian Journal of Statistics*, 8:93–115.

[13] Czado, C., Gneiting, T., and Held, L. (2009). Predictive model assessment for count data. *Biometrics*, 65:1254–1261.

[14] Davies, R. B. (1987). Hypothesis testing when a nuisance parameter is present only under the alternative. *Biometrika*, 74:33–43.

[15] Davis, R. and Wu, R. (2009). A negative binomial model for time series of counts. *Biometrika*, 96:735–749.

[16] Davis, R. A., Dunsmuir, W. T. M., and Streett, S. B. (2003). Observation-driven models for Poisson counts. *Biometrika*, 90:777–790.

[17] Davis, R. A. and Liu, H. (2012). Theory and Inference for a Class of Observation-driven Models with Application to Time Series of Counts. *ArXiv e-prints*. http://adsabs.harvard.edu/abs/2012arXiv1204.3915D.

[18] Davis, R. A., Wang, Y., and Dunsmuir, W. T. M. (1999). Modelling time series of count data. In Ghosh, S., editor, *Asymptotics, Nonparametric & Time Series*, pages 63–114. Marcel Dekker, New York.

[19] Dawid, A. P. (1984). Statistical theory: The prequential approach. *Journal of the Royal Statistical Society, Series A*, 147:278–292.

[20] Dean, C. B., Eaves, D. M., and Martinez, C. J. (1995). A comment on the use of empirical covariance matrices in the analysis of count data. *Journal of Statistical Planning and Inference*, 48:197–205.

[21] Dedecker, J., Doukhan, P., Lang, G., León, J. R., Louhichi, S., and Prieur, C. (2007). *Weak dependence: with examples and applications*, volume 190 of *Lecture Notes in Statistics*. Springer, New York.

[22] Douc, R., Doukhan, P., and Moulines, E. (2013). Ergodicity of observation-driven time series models and consistency of the maximum likelihood estimator. *Stochastic Processes and their Applications*, 123:2620–2647.

[23] Doukhan, P., Fokianos, K., and Tjøstheim, D. (2012). On weak dependence conditions for Poisson autoregressions. *Statistics & Probability Letters*, 82:942–948.

[24] Doukhan, P., Fokianos, K., and Tjøstheim, D. (2013). Corrigendum to "On weak dependence conditions for Poisson autoregression" [Statist. Probab. Lett. 82 (2012) 942–48]. *Statistics & Probability Letters*, 83:1926–1927.

[25] Doukhan, P. and Kengne, W. (2013). Inference and testing for structural change in time series of counts model. *Submitted for publication*.

[26] Doukhan, P. and Louhichi, S. (1999). A new weak dependence condition and applications to moment inequalities. *Stochastic Processes and their Applications*, 84:313–342.

[27] Doukhan, P. and Wintenberger, O. (2008). Weakly dependent chains with infinite memory. *Stochastic Processes and Their Applications*, 118:1997–2013.

[28] Engle, R. F. (1984). Wald, likelihood ratio, and lagrange multiplier tests in econometrics. volume 2 of *Handbook of Econometrics*, pages 775 – 826. Elsevier, Amsterdam: North Holland.

[29] Fahrmeir, L. and Tutz, G. (2001). *Multivariate Statistical Modelling Based on Generalized Linear Models*. Springer, New York. Second Edition.

[30] Fan, J. and Yao, Q. (2003). *Nonlinear time series: Nonparametric and parametric methods*. Springer-Verlag, New York.

[31] Ferland, R., Latour, A., and Oraichi, D. (2006). Integer–valued GARCH processes. *Journal of Time Series Analysis*, 27:923–942.

[32] Fokianos, K. (2012). Count time series. In Rao, T. S., Rao, S. S., and Rao, C. R., editors, *Handbook of Statistics: Time Series Analysis–Methods and Applications*, volume 30, pages 315–347. Elsevier B. V., Amsterdam.

[33] Fokianos, K. and Neumann, M. H. (2013). A goodness-of-fit test for Poisson count processes. *Electronic Journal of Statistics*, 7:793–819.

[34] Fokianos, K., Rahbek, A., and Tjøstheim, D. (2009a). Poisson autoregression. *Journal of the American Statistical Association*, 104:1430–1439.

[35] Fokianos, K., Rahbek, A., and Tjøstheim, D. (2009b). Poisson autoregression (complete version). available at `http://pubs.amstat.org/toc/jasa/104/488.`

[36] Fokianos, K. and Tjøstheim, D. (2011). Log–linear Poisson autoregression. *Journal of Multivariate Analysis*, 102:563–578.

[37] Fokianos, K. and Tjøstheim, D. (2012). Nonlinear Poisson autoregression. *Annals of the Institute of Statistical Mathematics*, 64:1205–1225.

[38] Francq, C., Horvath, L., and Zakoian, J.-M. (2008). Sup-tests for linearity in a general nonlinear AR(1) model when the supremum is taken over the full parameter space. MPRA Paper 16669, University Library of Munich, Germany.

[39] Francq, C. and Zakoïan, J.-M. (2004). Maximum likelihood estimation of pure GARCH and ARMA-GARCH processes. *Bernoulli*, 10:605–637.

[40] Francq, C. and Zakoïan, J.-M. (2010). *GARCH models: Stracture, Statistical Inference and Financial Applications*. John Wiley, United Kingdom.

[41] Gallant, A. R. (1987). *Nonlinear statistical models*. Wiley Series in Probability and Mathematical Statistics: Applied Probability and Statistics. John Wiley, New York.

[42] Gao, J., King, M., Lu, Z., and Tjøstheim, D. (2009). Specification testing in nonlinear and nonstationary time series regression. *Annals of Statistics*, 37:3893–3928.

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

[44] Godambe, V. P. and Heyde, C. C. (1987). Quasi–likelihood and optimal estimation. *International Statistical Review*, 55:231–244.

[45] Gourieroux, C., Monfort, A., and Trognon, A. (1984a). Pseudo maximum likelihood methods: Applications to Poisson models. *Econometrica*, 52:701–720.

[46] Gourieroux, C., Monfort, A., and Trognon, A. (1984b). Pseudo maximum likelihood methods: Theory. *Econometrica*, 52:681–700.

[47] Haggan, V. and Ozaki, T. (1981). Modelling nonlinear random vibrations using an amplitude-dependent autoregressive time series model. *Biometrika*, 68:189–196.

[48] Hall, P. and Heyde, C. C. (1980). *Martingale Limit Theory and its Applications*. Academic Press, New York.

[49] Handcock, M. S. and Morris, M. (1999). *Relative distribution methods in the social sciences*. Springer-Verlag, New York.

[50] Harvey, A. C. (1990). *The econometric analysis of time series*. LSE Handbooks in Economics. MIT Press, Cambridge, MA. Second Edition.

[51] Heyde, C. C. (1997). *Quasi-Likelihood and its Applications: A General Approach to Optimal Parameter Estimation*. Springer, New York.

[52] Hill, J. B. (2004). Lm-tests for linearity against smooth transition alternatives: A bootstrap simulation study. Working Papers 0412, Florida International University, Department of Economics.

[53] Jain, G. C. and Consul, P. C. (1971). A generalized negative binomial distribution. *SIAM Journal on Applied Mathematics*, 21:501–513.

[54] Jung, R. and Tremayne, A. (2011). Useful models for time series of counts or simply wrong ones? *AStA Advances in Statistical Analysis*, 95:59–91.

[55] Jung, R. C., Kukuk, M., and Liesenfeld, R. (2006). Time series of count data: modeling, estimation and diagnostics. *Computational Statistics & Data Analysis*, 51:2350–2364.

[56] Kedem, B. and Fokianos, K. (2002). *Regression Models for Time Series Analysis*. John Wiley, Hoboken, NJ.

[57] Kent, J. T. (1982). Robust properties of likelihood ratio tests. *Biometrika*, 69:19–27.

[58] Kokonendji, C. C., Dossou-Gbété, S., and Demétrio, C. G. B. (2004). Some discrete exponential dispersion models: Poisson-Tweedie and Hinde-Demétrio classes. *Statistics and Operations Research Transactions*, 28:201–213.

[59] Krengel, U. (1985). *Ergodic theorems*. Walter de Gruyter & Co., Berlin.

[60] Lawless, J. F. (1987). Negative binomial and mixed Poisson regression. *The Canadian Journal of Statistics*, 15:209–225.

[61] Liu, H. (2012). *Some Models for Time Series of Counts*. Ph.D. Thesis–Columbia University, USA.

[62] Luukkonen, R., Saikkonen, P., and Teräsvirta, T. (1988). Testing linearity against smooth transition autoregressive models. *Biometrika*, 75:491–499.

[63] Meitz, M. and Saikkonen, P. (2011). Parameter estimation in nonlinear AR-GARCH models. *Econometric Theory*, 27:1236–1278.

[64] Mikosch, T. (2009). *Non-life insurance mathematics, An introduction with the Poisson process*. Springer-Verlag, Berlin. Second Edition.

[65] Mikosch, T. and Straumann, D. (2006). Quasi-maximum-likelihood estimation in conditionally heteroscedastic time series: a stochastic recurrence equations approach. *The Annals of Statistics*, 34:2449–2495.

[66] Neumann, M. (2011). Absolute regularity and ergodicity of Poisson count processes. *Bernoulli*, 17:1268–1284.

[67] Pedeli, X. and Karlis, D. (2013). Some properties of multivariate INAR(1) processes. *Computational Statistics & Data Analysis*, 67:213–225.

[68] R Development Core Team (2012). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. http://www.R-project.org/.

[69] Rao, R. C. (1948). Large sample tests of statistical hypotheses concerning several parameters with applications to problems of estimation. *Mathematical Proceedings of the Cambridge Philosophical Society*, 44:50–57.

[70] Teräsvirta, T. (1994). Specification, estimation, and evaluation of smooth transition autoregressive models. *Journal of the American Statistical Association*, 89:208–218.

[71] Teräsvirta, T., Tjøstheim, D., and Granger, C. W. J. (2010). *Modelling Nonlinear Economic Time Series*. Oxford University Press, Oxford.

[72] Tong, H. (1990). *Nonlinear time series: A dynamical system approach*. Oxford University Press, New York.

[73] White, H. (1982). Maximum likelihood estimation of misspecified models. *Econometrica*, 50:1–25.

[74] Woodard, D. W., Matteson, D. S., and Henderson, S. G. (2011). Stationarity of count-valued and nonlinear time series models. *Electronic Journal of Statistics*, 5:800–828.

[75] Zeger, S. L. and Liang, K.-Y. (1986). Longitudinal data analysis for discrete and continuous outcomes. *Biometrics*, 42:121–130.

[76] Zeger, S. L. and Qaqish, B. (1988). Markov regression models for time series: a quasi-likelihood approach. *Biometrics*, 44:1019–1031.

[77] Zhu, F. (2011). A negative binomial integer-valued GARCH model. *Journal of Time Series Analysis*, 32:54–67.

[78] Zucchini, W. and MacDonald, L. I. (2009). *Hidden Markov models for time series*. CRC Press, Boca Raton, FL.