Journal of Statistical Software

November 2017, Volume 82, Issue 5. *doi: 10.18637/jss.v082.i05*

tscount: An R Package for Analysis of Count Time Series Following Generalized Linear Models

Tobias Liboschik
TU Dortmund University

Konstantinos Fokianos
University of Cyprus

Roland Fried
TU Dortmund University

Abstract

The R package **tscount** provides likelihood-based estimation methods for analysis and modeling of count time series following generalized linear models. This is a flexible class of models which can describe serial correlation in a parsimonious way. The conditional mean of the process is linked to its past values, to past observations and to potential covariate effects. The package allows for models with the identity and with the logarithmic link function. The conditional distribution can be Poisson or negative binomial. An important special case of this class is the so-called INGARCH model and its log-linear extension. The package includes methods for model fitting and assessment, prediction and intervention analysis. This paper summarizes the theoretical background of these methods. It gives details on the implementation of the package and provides simulation results for models which have not been studied theoretically before. The usage of the package is illustrated by two data examples. Additionally, we provide a review of R packages which can be used for count time series analysis. This includes a detailed comparison of **tscount** to those packages.

Keywords: aberration detection, autoregressive models, intervention analysis, likelihood, mixed Poisson, model selection, prediction, R, regression model, serial correlation.

1. Introduction

Recently, there has been an increasing interest in regression models for time series of counts and a considerable number of publications on this subject has appeared in the literature. However, most of the proposed methods are not yet available in a statistical software package and hence they cannot be applied easily. We aim at filling this gap and publish a package, named **tscount** (Liboschik, Fried, Fokianos, and Probst 2017), for the popular free and open source software environment R (R Core Team 2017). In fact, our main goal is to develop software for models whose conditional mean depends on previous observations and on its

own previous values. These models are quite analogous to the generalized autoregressive conditional heteroscedasticity (GARCH) models (Bollerslev 1986) which were proposed for describing the conditional variance.

Count time series appear naturally in various areas whenever a number of events per time period is observed over time. Examples showing the wide range of applications are the daily number of hospital admissions, from public health, the number of stock market transactions per minute, from finance or the hourly number of defect items, from industrial quality control.

Models for count time series should take into account that the observations are nonnegative integers and they should capture suitably the dependence among observations. A convenient and flexible approach is to employ the generalized linear model (GLM) methodology (Nelder and Wedderburn 1972) for modeling the observations conditionally on the past information. This methodology is implemented by choosing a suitable distribution for count data and an appropriate link function. Such an approach is pursued by Fahrmeir and Tutz (2001, Chapter 6) and Kedem and Fokianos (2002, Chapters 1–4), among others. Another important class of models for time series of counts is based on the thinning operator, like the integer autoregressive moving average (INARMA) models, which, in a way, imitate the structure of the common autoregressive moving average (ARMA) models (see the review article by Weiß 2008). A different type of count time series models are the so-called state space models. We refer to the reviews of Fokianos (2011), Jung and Tremayne (2011), Fokianos (2012), Tjøstheim (2012) and Fokianos (2015) for an in-depth overview of models for count time series. Advantages of GLM-based models compared to the models which are based on the thinning operator are the following:

(a) They can describe covariate effects and negative correlations in a straightforward way.
(b) There is a rich toolkit available for this class of models.

State space models allow to describe even more flexible data generating processes than GLM models but at the cost of a more complicated model specification. On the other hand, GLM-based models yield predictions in a convenient way due to their explicit formulation.

In the presented version of the **tscount** package we provide likelihood-based methods for the framework of count time series following GLMs. Some simple autoregressive models can be fitted with standard software by treating the observations as if they were independent (see Section 8 and Appendix A.3), for example, using the R function `glm`. However, these procedures are in general not tailored for dependent data and may yield invalid model fits. The implementation in the package **tscount** allows for a more general dependence structure which is specified conveniently by the user. We consider general time series models whose conditional mean may depend on time-varying covariates, previous observations and, similar to the conditional variance of a GARCH model, on its own previous values. The usage and output of our functions are in parts inspired by the R functions `arima` and `glm` in order to provide a familiar user experience. Furthermore **tscount** is object-oriented and provides many standard S3 methods for well-known generic functions. There are several other R functions available which can be employed for analyzing count time series. Many of those are related to GLMs and have been developed for independent observations but are, with some limitations, also capable to describe simple forms of serial dependence. There are also some functions available for extending such models to time series. Another group of functions handles state space models for count time series. We briefly review these functions and the corresponding model classes in Section 8 and compare them to **tscount**. As it turns out, there

are special cases for which our model corresponds to existing ones. In these cases we obtain quite similar results with functions from some other packages, thus confirming the reliability of our package. However, many features of **tscount**, like the flexible dependence structure, outreach the capability of other packages. Admittedly, some packages provide features like zero-inflation or more general forms of the linear predictor which cannot be accommodated yet by **tscount** but could possibly be included in future versions. As a conclusion, this package is a valuable addition to the R environment which fills some significant gaps associated with time series fitting.

The functionality of **tscount** partly goes beyond the theory available in the literature since theoretical investigation of these models is still an ongoing research theme. For instance the problem of accommodating covariates in such GLM-type count time series models or fitting a mixed Poisson log-linear model have not been studied theoretically. We have checked their appropriateness by simulations reported in Appendix B. However, some care should be taken when applying the package's programs to situations which are not covered by available theory.

This paper is organized as follows. First the theoretical background of the methods included in the package is briefly summarized with references to the literature for more details. Section 2 introduces the models we consider. Section 3 describes quasi maximum likelihood estimation of the unknown model parameters and gives some details regarding its implementation. Section 4 treats prediction with such models. Section 5 sums up tools for model assessment. Section 6 discusses procedures for the detection of interventions. Section 7 demonstrates the usage of the package with two data examples. Section 8 reviews other R packages which are capable to model count time series and compares them with our package. Finally, Section 9 gives an outlook on possible future extensions of our package. In the Appendix we give further details and we confirm empirically some of the new methods that we discuss but which have not been studied theoretically, as of yet.

2. Models

Denote a count time series by $\{Y_t : t \in \mathbb{N}\}$. We will denote by $\{\boldsymbol{X}_t : t \in \mathbb{N}\}$ a time-varying r-dimensional covariate vector, say $\boldsymbol{X}_t = (X_{t,1}, \ldots, X_{t,r})^\top$. We model the conditional mean $\mathsf{E}(Y_t|\mathcal{F}_{t-1})$ of the count time series by a process, say $\{\lambda_t : t \in \mathbb{N}\}$, such that $\mathsf{E}(Y_t|\mathcal{F}_{t-1}) = \lambda_t$. Denote by $\mathcal{F}_t$ the history of the joint process $\{Y_t, \lambda_t, \boldsymbol{X}_{t+1} : t \in \mathbb{N}\}$ up to time t including the covariate information at time $t+1$. The distributional assumption for Y_t given $\mathcal{F}_{t-1}$ is discussed later. We are interested in models of the general form

$$g(\lambda_t) = \beta_0 + \sum_{k=1}^{p} \beta_k \, \widetilde{g}(Y_{t-i_k}) + \sum_{\ell=1}^{q} \alpha_\ell g(\lambda_{t-j_\ell}) + \boldsymbol{\eta}^\top \boldsymbol{X}_t, \tag{1}$$

where $g : \mathbb{R}^+ \to \mathbb{R}$ is a link function and $\widetilde{g} : \mathbb{N}_0 \to \mathbb{R}$ is a transformation function. The parameter vector $\boldsymbol{\eta} = (\eta_1, \ldots, \eta_r)^\top$ corresponds to the effects of covariates. In the terminology of GLMs we call $\nu_t = g(\lambda_t)$ the linear predictor. To allow for regression on arbitrary past observations of the response, define a set $P = \{i_1, i_2, \ldots, i_p\}$ and integers $0 < i_1 < i_2 \ldots < i_p < \infty$, with $p \in \mathbb{N}_0$. This enables us to regress on the lagged observations $Y_{t-i_1}, Y_{t-i_2}, \ldots, Y_{t-i_p}$. Analogously, define a set $Q = \{j_1, j_2, \ldots, j_q\}$, $q \in \mathbb{N}_0$ and integers $0 < j_1 < j_2 \ldots < j_q < \infty$, for regression on lagged conditional means $\lambda_{t-j_1}, \lambda_{t-j_2}, \ldots, \lambda_{t-j_q}$. This case is covered by the theory for models with $P = \{1, \ldots, p\}$ and $Q = \{1, \ldots, q\}$ by choosing p and q suitably and

setting some model parameters to zero. Our formulation is particularly useful when dealing with modeling stochastic seasonality (see Section 7.1, for an example). Specification of the model order, i.e., of the sets P and Q, are guided by considering the empirical autocorrelation functions of the observed data. This approach is described for ARMA models in many time series analysis textbooks and transfers to the above model by employing its ARMA representation (see (20) in Appendix A.3). Parameter constraints which ensure stationarity and ergodicity of two important special cases of (1) are given in Section 3.

We give several examples of model (1). Consider the situation where g and $\widetilde{g}$ equal the identity, i.e., $g(x) = \widetilde{g}(x) = x$. Furthermore, let $P = \{1, \ldots, p\}$, $Q = \{1, \ldots, q\}$ and $\boldsymbol{\eta} = \mathbf{0}$. Then model (1) becomes

$$\lambda_t = \beta_0 + \sum_{k=1}^{p} \beta_k \, Y_{t-k} + \sum_{\ell=1}^{q} \alpha_\ell \lambda_{t-\ell}. \tag{2}$$

Assuming further that Y_t given the past is Poisson distributed, then we obtain an *integer-valued GARCH model* of order p and q, abbreviated as INGARCH(p, q). These models are also known as *autoregressive conditional Poisson (ACP) models*. They have been discussed by Heinen (2003), Ferland, Latour, and Oraichi (2006) and Fokianos, Rahbek, and Tjøstheim (2009), among others. When $\boldsymbol{\eta} \neq \mathbf{0}$, then our package fits INGARCH models with nonnegative covariates; this is so because we need to ensure that the resulting mean process is positive. An example of an INGARCH model with covariates is given in Section 6, where we fit a count time series model which includes intervention effects.

Consider again model (1) but now with the logarithmic link function $g(x) = \log(x)$, $\widetilde{g}(x) = \log(x + 1)$ and P, Q as before. Then, we obtain a *log-linear model* of order p and q for the analysis of count time series. Indeed, set $\nu_t = \log(\lambda_t)$ to obtain from (1) that

$$\nu_t = \beta_0 + \sum_{k=1}^{p} \beta_k \, \log(Y_{t-k} + 1) + \sum_{\ell=1}^{q} \alpha_\ell \nu_{t-\ell}. \tag{3}$$

This log-linear model has been studied by Fokianos and Tjøstheim (2011), Woodard, Matteson, and Henderson (2011) and Douc, Doukhan, and Moulines (2013). We follow Fokianos and Tjøstheim (2011) in transforming past observations by employing the function $\widetilde{g}(x) = \log(x+1)$, such that they are on the same scale as the linear predictor ν_t. These authors show that the addition of a constant c to each observation for avoiding zero values does not affect inference; in addition they argue that a reasonable choice for c is 1. Note that model (3) allows modeling of negative serial correlation, whereas (2) accommodates positive serial correlation only. Additionally, (3) accommodates covariates easier than (2) since the log-linear model implies positivity of the conditional mean process $\{\lambda_t\}$. The linear model (2) with covariates should be fitted with some care because it is limited to positive effects on $\{\lambda_t\}$. The effects of covariates on the response is multiplicative for model (3); it is additive though for model (2). For a discussion on the inclusion of time-dependent covariates see Fokianos and Tjøstheim (2011, Section 4.3).

Model (1) together with the *Poisson* assumption, i.e., $Y_t|\mathcal{F}_{t-1} \sim \text{Poisson}(\lambda_t)$, implies that

$$\mathsf{P}\left(Y_t = y|\mathcal{F}_{t-1}\right) = \frac{\lambda_t^y \exp(-\lambda_t)}{y!}, \quad y = 0, 1, \ldots. \tag{4}$$

It holds $\mathsf{VAR}(Y_t|\mathcal{F}_{t-1}) = \mathsf{E}(Y_t|\mathcal{F}_{t-1}) = \lambda_t$. Hence in the case of a conditional Poisson response model the conditional mean is identical to the conditional variance of the observed process.

The *negative binomial* distribution allows for a conditional variance to be larger than the mean λ_t, which is often referred to as overdispersion. Following Christou and Fokianos (2014), it is assumed that $Y_t|\mathcal{F}_{t-1} \sim \text{NegBin}(\lambda_t, \phi)$, where the negative binomial distribution is parametrized in terms of its mean with an additional dispersion parameter $\phi \in (0, \infty)$, i.e.,

$$\mathsf{P}(Y_t = y|\mathcal{F}_{t-1}) = \frac{\Gamma(\phi + y)}{\Gamma(y+1)\Gamma(\phi)} \left(\frac{\phi}{\phi + \lambda_t}\right)^{\phi} \left(\frac{\lambda_t}{\phi + \lambda_t}\right)^{y}, \quad y = 0, 1, \ldots. \tag{5}$$

In this case, $\mathsf{VAR}(Y_t|\mathcal{F}_{t-1}) = \lambda_t + \lambda_t^2/\phi$, i.e., the conditional variance increases quadratically with λ_t. The Poisson distribution is a limiting case of the negative binomial when $\phi \to \infty$.

Note that the negative binomial distribution belongs to the class of mixed Poisson processes. A mixed Poisson process is specified by setting $Y_t = N_t(0, Z_t\lambda_t]$, where $\{N_t\}$ are i.i.d. Poisson processes with unit intensity and $\{Z_t\}$ are i.i.d. random variables with mean 1 and variance σ^2, independent of $\{Y_t\}$ (Christou and Fokianos 2014). When $\{Z_t\}$ is an i.i.d. process of Gamma random variables, then we obtain the negative binomial process with $\sigma^2 = 1/\phi$. We refer to σ^2 as the overdispersion coefficient because it is proportional to the extent of overdispersion of the conditional distribution. The limiting case of $\sigma^2 = 0$ corresponds to the Poisson distribution, i.e., no overdispersion. The estimation procedure we study is not confined to the negative binomial case but to any mixed Poisson distribution. However, the negative binomial assumption is required for prediction intervals and model assessment; these topics are discussed in Sections 4 and 5.

In model (1) the effect of a covariate fully enters the dynamics of the process and propagates to future observations both by the regression on past observations and by the regression on past conditional means. The effect of such covariates can be seen as an internal influence on the data-generating process, which is why we refer to it as an *internal* covariate effect. We also allow to include covariates in a way that their effect only propagates to future observations by the regression on past observations but not directly by the regression on past conditional means. Following Liboschik, Kerschke, Fokianos, and Fried (2016), who make this distinction for the case of intervention effects described by deterministic covariates, we refer to the effect of such covariates as an *external* covariate effect. Let $\boldsymbol{e} = (e_1, \ldots, e_r)^\top$ be a vector specified by the user with $e_i = 1$ if the ith component of the covariate vector has an external effect and $e_i = 0$ otherwise, $i = 1, \ldots, r$. Denote by $\text{diag}(\boldsymbol{e})$ a diagonal matrix with diagonal elements given by $\boldsymbol{e}$. The generalization of (1) allowing for both internal and external covariate effects is given by

$$g(\lambda_t) = \beta_0 + \sum_{k=1}^{p} \beta_k \, \widetilde{g}(Y_{t-i_k}) + \sum_{\ell=1}^{q} \alpha_\ell \left(g(\lambda_{t-j_\ell}) - \boldsymbol{\eta}^\top \text{diag}(\boldsymbol{e}) \boldsymbol{X}_{t-j_\ell}\right) + \boldsymbol{\eta}^\top \boldsymbol{X}_t. \tag{6}$$

Basically, the effect of all covariates with an external effect is subtracted in the feedback terms such that their effect enters the dynamics of the process only via the observations. We refer to Liboschik *et al.* (2016) for an extensive discussion and comparison of internal and external effects. It is our experience with these models that an empirical discrimination between internal and external covariate effects is difficult and that it is not crucial which type of covariate effect is chosen for applications.

3. Estimation and inference

The **tscount** package fits models of the form (1) by quasi conditional maximum likelihood (ML) estimation (function `tsglm`). If the Poisson assumption holds true, then we obtain an ordinary ML estimator. However, under the mixed Poisson assumption we obtain a quasi-ML estimator. Denote by $\boldsymbol{\theta} = (\beta_0, \beta_1, \ldots, \beta_p, \alpha_1, \ldots, \alpha_q, \eta_1, \ldots, \eta_r)^\top$ the vector of regression parameters. Regardless of the distributional assumption, the parameter space for the INGARCH model (2) with covariates is given by

$$\Theta = \left\{ \boldsymbol{\theta} \in \mathbb{R}^{p+q+r+1} : \ \beta_0 > 0, \ \beta_1, \ldots, \beta_p, \alpha_1, \ldots, \alpha_q, \eta_1, \ldots, \eta_r \geq 0, \ \sum_{k=1}^{p} \beta_k + \sum_{\ell=1}^{q} \alpha_\ell < 1 \right\}.$$

The intercept β_0 must be positive and all other parameters must be nonnegative to ensure positivity of the conditional mean λ_t. The further condition ensures that the fitted model has a stationary and ergodic solution with moments of any order (Ferland *et al.* 2006; Fokianos *et al.* 2009; Doukhan, Fokianos, and Tjøstheim 2012); see also Tjøstheim (2015) for a recent review. For the log-linear model (3) with covariates the parameter space is taken to be

$$\Theta = \left\{ \boldsymbol{\theta} \in \mathbb{R}^{p+q+r+1} : \ |\beta_1|, \ldots, |\beta_p|, |\alpha_1|, \ldots, |\alpha_q| < 1, \ \left| \sum_{k=1}^{p} \beta_k + \sum_{\ell=1}^{q} \alpha_\ell \right| < 1 \right\},$$

see Appendix A.1 for a discussion. Christou and Fokianos (2014) point out that with the parametrization (5) of the negative binomial distribution the estimation of the regression parameters $\boldsymbol{\theta}$ does not depend on the additional dispersion parameter ϕ. This allows to employ a quasi maximum likelihood approach based on the Poisson likelihood to estimate the regression parameters $\boldsymbol{\theta}$, which is described below. The nuisance parameter ϕ is then estimated separately in a second step. This approach is different from a full maximum likelihood estimation based on the negative binomial distribution, which for example has been implemented in the function `glm.nb` in the R package **MASS** (Venables and Ripley 2002). In that algorithm, maximization of the negative binomial likelihood for an estimated dispersion parameter ϕ and estimation of ϕ given the estimated regression parameters $\boldsymbol{\theta}$ are iterated until convergence. The quasi negative binomial approach has been chosen for simplicity and its usefulness on deriving consistent estimators when the model for λ_t has been correctly specified (see also Ahmad and Francq 2016).

The log-likelihood, score vector and information matrix are derived conditionally on presample values of the time series and the conditional mean process $\{\lambda_t\}$, precisely on $\mathcal{F}_0$. An appropriate initialization is needed for their evaluation, which is discussed in Section 3.1. For a vector of observations $\boldsymbol{y} = (y_1, \ldots, y_n)^\top$, the conditional quasi log-likelihood function, up to a constant, is given by

$$\ell(\boldsymbol{\theta}) = \sum_{t=1}^{n} \log p_t(y_t; \boldsymbol{\theta}) = \sum_{t=1}^{n} \Big(y_t \ln(\lambda_t(\boldsymbol{\theta})) - \lambda_t(\boldsymbol{\theta}) \Big), \tag{7}$$

where $p_t(y; \boldsymbol{\theta}) = \mathsf{P}(Y_t = y | \mathcal{F}_{t-1})$ is the probability density function of a Poisson distribution as defined in (4). The conditional mean is regarded as a function $\lambda_t : \Theta \to \mathbb{R}^+$ and thus it is denoted by $\lambda_t(\boldsymbol{\theta})$ for all t. The conditional score function is the $(p+q+r+1)$-dimensional vector given by

$$S_n(\boldsymbol{\theta}) = \frac{\partial \ell(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \sum_{t=1}^{n} \left(\frac{y_t}{\lambda_t(\boldsymbol{\theta})} - 1 \right) \frac{\partial \lambda_t(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}. \tag{8}$$

The vector of partial derivatives $\partial\lambda_t(\boldsymbol{\theta})/\partial\boldsymbol{\theta}$ can be computed recursively by the recursions given in Appendix A.2. Finally, the conditional information matrix is given by

$$G_n(\boldsymbol{\theta};\sigma^2) = \sum_{t=1}^{n} \mathsf{COV}\left(\frac{\partial\ell(\boldsymbol{\theta};Y_t)}{\partial\boldsymbol{\theta}}\middle|\mathcal{F}_{t-1}\right) = \sum_{t=1}^{n}\left(\frac{1}{\lambda_t(\boldsymbol{\theta})} + \sigma^2\right)\left(\frac{\partial\lambda_t(\boldsymbol{\theta})}{\partial\boldsymbol{\theta}}\right)\left(\frac{\partial\lambda_t(\boldsymbol{\theta})}{\partial\boldsymbol{\theta}}\right)^{\top}.$$

In the case of the Poisson assumption it holds $\sigma^2 = 0$ and in the case of the negative binomial assumption $\sigma^2 = 1/\phi$. For the ease of notation let $G_n^*(\boldsymbol{\theta}) = G_n(\boldsymbol{\theta};0)$, which is the conditional information matrix in case of a Poisson distribution.

The quasi maximum likelihood estimator (QMLE) $\widehat{\boldsymbol{\theta}}_n$ of $\boldsymbol{\theta}$, assuming that it exists, is the solution of the non-linear constrained optimization problem

$$\widehat{\boldsymbol{\theta}} := \widehat{\boldsymbol{\theta}}_n = \arg\max_{\boldsymbol{\theta}\in\Theta}\ell(\boldsymbol{\theta}). \tag{9}$$

Denote the fitted values by $\widehat{\lambda}_t = \lambda_t(\widehat{\theta})$. Following Christou and Fokianos (2014), the dispersion parameter ϕ of the negative binomial distribution is estimated by solving the equation

$$\sum_{t=1}^{n}\frac{(Y_t-\widehat{\lambda}_t)^2}{\widehat{\lambda}_t+\widehat{\lambda}_t^2/\widehat{\phi})} = n-(p+q+r+1), \tag{10}$$

which is based on Pearson's χ^2-statistic. The variance parameter σ^2 is estimated by $\widehat{\sigma}^2 = 1/\widehat{\phi}$. For the Poisson distribution we set $\widehat{\sigma}^2 = 0$. Strictly speaking, the log-linear model (3) does not fall into the class of models considered by Christou and Fokianos (2014). However, results obtained by Douc *et al.* (2013) (for $p = q = 1$) and Sim (2016) (for $p = q$) allow us to use this estimator also for the log-linear model. This issue is addressed by simulations in Appendix B.2, which support that the estimator obtained by (10) provides good results also for models with the logarithmic link function.

Inference for the regression parameters is based on the asymptotic normality of the QMLE, which has been studied by Fokianos *et al.* (2009) and Christou and Fokianos (2014) for models without covariates. For a well behaved covariate process $\{\boldsymbol{X}_t\}$ we conjecture that

$$\sqrt{n}\left(\widehat{\boldsymbol{\theta}}_n-\boldsymbol{\theta}_0\right) \xrightarrow{d} N_{p+q+r+1}\left(\mathbf{0}, G_n^{-1}(\widehat{\boldsymbol{\theta}}_n;\widehat{\sigma}^2)G_n^*(\widehat{\boldsymbol{\theta}}_n)G_n^{-1}(\widehat{\boldsymbol{\theta}}_n;\widehat{\sigma}^2)\right), \tag{11}$$

as $n\to\infty$, where $\boldsymbol{\theta}_0$ denotes the true parameter value and $\widehat{\sigma}^2$ is a consistent estimator of σ^2. We suppose that this applies under the same assumptions usually made for the ordinary linear regression model (see for example Demidenko 2013, p. 140 ff.). For deterministic covariates these assumptions are $\|\boldsymbol{X}_t\| < c$, where $\|\cdot\|$ denotes the usual Euclidean norm, i.e., the covariate process is bounded, and $\lim_{n\to\infty} n^{-1}\sum_{t=1}^{n}\boldsymbol{X}_t\boldsymbol{X}_t^{\top} = A$, where c is a constant and A is a nonsingular matrix. For stochastic covariates it is assumed that the expectations $\mathsf{E}(\boldsymbol{X}_t)$ and $\mathsf{E}\left(\boldsymbol{X}_t\boldsymbol{X}_t^{\top}\right)$ exist and that $\mathsf{E}\left(\boldsymbol{X}_t\boldsymbol{X}_t^{\top}\right)$ is nonsingular. These assumptions imply that the information on each covariate grows linearly with the sample size and that the covariates are not linearly dependent. Fuller (1996, Theorem 9.1.1) shows asymptotic normality of the least squares estimator for a regression model with time series errors under even more general conditions which allow the presence of certain types of trends in the covariates. For the special case of a Poisson model with the identity link, Agosto, Cavaliere, Kristensen, and Rahbek (2015) show asymptotic normality of the MLE for a model with covariates that are functions of Markov processes with finite second moments and that are not collinearly related

to the response. The asymptotic normality of the QMLE in our context is supported by the simulations presented in Appendix B.1. A formal proof requires further research. To avoid numerical instabilities when inverting $G_n(\widehat{\boldsymbol{\theta}}_n; \widehat{\sigma}^2)$ we apply an algorithm which makes use of the fact that it is a real symmetric and positive definite matrix; see Appendix A.4.

As an alternative method to the normal approximation (11) for obtaining standard errors and confidence intervals (function `se`) we include a parametric bootstrap procedure (argument `B`), for which computation time is many times higher. Accordingly, B time series are simulated from the model fitted to the original data. The empirical standard errors of the parameter estimates for these B time series are the bootstrap standard errors. Confidence intervals are based on quantiles of the bootstrap sample, see Efron and Tibshirani (1993, Chapter 13). This procedure can compute standard errors and confidence intervals both for $\widehat{\boldsymbol{\theta}}$ and $\widehat{\sigma}^2$. In our experience $B = 500$ yields stable results.

3.1. Implementation

This section and Appendix A provide some details on the implementation of the function `tsglm` and explain its technical arguments. The default settings of these arguments are chosen carefully based on plenty of experiments and should be sufficient for most situations.

The parameter restrictions which are imposed by the condition $\boldsymbol{\theta} \in \Theta$ can be formulated as d linear inequalities. This means that there exists a matrix $\boldsymbol{U}$ of dimension $d \times (p+q+r+1)$ and a vector $\boldsymbol{c}$ of length d, such that $\Theta = \{\boldsymbol{\theta} \mid \boldsymbol{U}\boldsymbol{\theta} \geq \boldsymbol{c}\}$. For the linear model (2) one needs $d = p+q+r+2$ constraints to ensure nonnegativity of the conditional mean λ_t and stationarity of the resulting process. For the log-linear model (3) there are not any constraints on the intercept term and on the covariate coefficients; hence $d = 2(p+q+1)$. In order to enforce strict inequalities the respective constraints are tightened by an arbitrarily small constant $\xi > 0$; this constant is set to $\xi = 10^{-6}$ by default (argument `slackvar`).

For numerically solving the maximization problem (9) we employ by default the function `constrOptim`. This function applies an algorithm described by Lange (1999, Chapter 14), which essentially enforces the constraints by adding a barrier value to the objective function and then employs an algorithm for unconstrained optimization of this new objective function, iterating these two steps if necessary. By default the quasi-Newton Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm is employed for the latter task of unconstrained optimization, which additionally makes use of the score vector (8). It is possible to tune the optimization algorithm and even to employ an unconstrained optimization (argument `final.control`).

Note that the log-likelihood (7) and the score (8) are given conditional on unobserved presample values. They depend on the linear predictor and its partial derivatives, which can be computed recursively using any initialization. We give the recursions and present several strategies for their initialization in Appendix A.2 (arguments `init.method` and `init.drop`). Christou and Fokianos (2014, Remark 3.1) show that the effect of the initialization vanishes asymptotically. Nevertheless, from a practical point of view the initialization of the recursions is crucial. Especially in the presence of strong serial dependence, the resulting estimates can differ substantially even for long time series with, for example, 1000 observations; see the simulated example in Table 3 in Appendix A.2.

Solving the non-linear optimization problem (9) requires a starting value for the parameter vector $\boldsymbol{\theta}$. This starting value can be obtained from fitting a simpler model for which an estima-

tion procedure is readily available. We consider either to fit a GLM or to fit an ARMA model. A third possibility is to fit a naive i.i.d. model without covariates. Furthermore, the user can assign fixed values. All these possibilities are available by the argument `start.control`. It turns out that the optimization algorithm converges very reliably even if the starting values are not close to the global optimum of the likelihood. A starting value which is closer to the global optimum usually requires fewer iterations until convergence. However, we have encountered some examples where starting values close to a local optimum, obtained by one of the first two methods mentioned before, do not yield the global optimum. Consequently, we recommend fitting the naive i.i.d. model without covariates to obtain starting values. More details on these approaches are given in Appendix A.3.

4. Prediction

In terms of the mean square error, the optimal 1-step-ahead predictor $\widehat{Y}_{n+1}$ for Y_{n+1}, given $\mathcal{F}_n$, i.e., the past of the process up to time n and potential covariates at time $n+1$, is the conditional expectation λ_{n+1} given in (1) (S3 method of function `predict`). By construction of the model the conditional distribution of $\widehat{Y}_{n+1}$ is a Poisson (4) respectively negative binomial (5) distribution with mean λ_{n+1}. An h-step-ahead prediction $\widehat{Y}_{n+h}$ for Y_{n+h} is obtained by recursive 1-step-ahead predictions, where unobserved values $Y_{n+1}, \ldots, Y_{n+h-1}$ are replaced by their respective 1-step-ahead prediction, $h \in \mathbb{N}$. The distribution of this h-step-ahead prediction $\widehat{Y}_{n+h}$ is not known analytically but can be approximated numerically by a parametric bootstrap procedure, which is described below.

In applications, λ_{n+1} is substituted by its estimator $\widehat{\lambda}_{n+1} = \lambda_{n+1}(\widehat{\boldsymbol{\theta}})$, which depends on the estimated regression parameters $\widehat{\boldsymbol{\theta}}$. The dispersion parameter ϕ of the negative binomial distribution is replaced by its estimator $\widehat{\phi}$. Note that plugging in the estimated parameters induces additional uncertainty to the predictive distribution. This estimation uncertainty is not taken into account for the construction of prediction intervals described in the following paragraphs.

Prediction intervals for Y_{n+h} with a given coverage rate $1-\alpha$ (argument `level`) are designed to cover the true observation Y_{n+h} with a probability of $1-\alpha$. Simultaneous prediction intervals achieving a global coverage rate for $Y_{n+1}, \ldots, Y_{n+h}$ can be obtained by a Bonferroni adjustment of the individual coverage rates to $1-\alpha/h$ each (argument `global = TRUE`).

There are two different principles for constructing prediction intervals available which in practice often yield identical intervals. Firstly, the limits can be the $(\alpha/2)$- and $(1-\alpha/2)$-quantile of the (approximated) predictive distribution (argument `type = "quantiles"`). Secondly, the limits can be chosen such that the interval has minimal length given that, according to the (approximated) predictive distribution, the probability that a value falls into this interval is at least as large as the desired coverage rate $1-\alpha$ (argument `type = "shortest"`).

One-step-ahead prediction intervals can be straightforwardly obtained from the conditional distribution (argument `method = "conddistr"`). Prediction intervals obtained by a parametric bootstrap procedure (argument `method = "bootstrap"`) are based on B simulations of realizations $y_{n+1}^{(b)}, \ldots, y_{n+h}^{(b)}$ from the fitted model, $b = 1, \ldots, B$ (argument `B`). To obtain an approximate prediction interval for Y_{n+h} one can either use the empirical $(\alpha/2)$- and $(1-\alpha/2)$-quantile of $y_{n+h}^{(1)}, \ldots, y_{n+h}^{(B)}$ (if `type = "quantiles"`) or find the shortest interval which contains at least $\lceil (1-\alpha) \cdot B \rceil$ of these observations (if `type = "shortest"`). This

bootstrap procedure can be accelerated by distributing it to multiple cores simultaneously (argument `parallel = TRUE`), which requires a computing cluster registered by the R package **parallel** (see the help page of the function `setDefaultCluster`).

5. Model assessment

Tools originally developed for generalized linear models as well as for time series can be utilized to assess the model fit and its predictive performance. Within the class of count time series following generalized linear models it is desirable to assess the specification of the linear predictor as well as the choice of the link function and of the conditional distribution. The tools presented in this section facilitate the selection of an adequate model for a given data set. Note that all tools are introduced as in-sample versions, meaning that the observations $y_1 \ldots, y_n$ are used for fitting the model as well as for assessing the obtained fit. However, it is straightforward to apply such tools as out-of-sample criteria.

Recall that the fitted values are denoted by $\widehat{\lambda}_t = \lambda_t(\widehat{\theta})$. Note that these do not depend on the chosen distribution, because the mean is the same regardless of the response distribution. There are various types of *residuals* available (S3 method of function `residuals`). Response (or raw) residuals (argument `type = "response"`) are given by

$$r_t = y_t - \widehat{\lambda}_t, \tag{12}$$

whereas a standardized alternative are Pearson residuals (argument `type = "pearson"`)

$$r_t^P = (y_t - \widehat{\lambda}_t)/\sqrt{\widehat{\lambda}_t + \widehat{\lambda}_t^2\widehat{\sigma}^2}, \tag{13}$$

or the more symmetrically distributed standardized Anscombe residuals (argument `type = "anscombe"`)

$$r_t^A = \frac{3/\widehat{\sigma}^2\big((1+y_t\widehat{\sigma}^2)^{2/3} - (1+\widehat{\lambda}_t\widehat{\sigma}^2)^{2/3}\big) + 3(y_t^{2/3} - \widehat{\lambda}_t^{2/3})}{2(\widehat{\lambda}_t + \widehat{\lambda}_t^2\widehat{\sigma}^2)^{1/6}}, \tag{14}$$

for $t = 1, \ldots, n$ (see for example Hilbe 2011, Section 5.1). The empirical autocorrelation function of these residuals is useful for diagnosing serial dependence which has not been explained by the fitted model. A plot of the residuals against time can reveal changes of the data generating process over time. Furthermore, a plot of squared residuals r_t^2 against the corresponding fitted values $\widehat{\lambda}_t$ exhibits the relation of mean and variance and might point to the Poisson distribution if the points scatter around the identity function or to the negative binomial distribution if there exists a quadratic relation (see Ver Hoef and Boveng 2007).

Christou and Fokianos (2015b) and Jung and Tremayne (2011) extend tools for assessing the predictive performance to count time series, which were originally proposed by Gneiting, Balabdaoui, and Raftery (2007) and others for continuous data and transferred to independent but not identically distributed count data by Czado, Gneiting, and Held (2009). These tools follow the *prequential principle* formulated by Dawid (1984), depending only on the realized observations and their respective forecast distributions. Denote by $P_t(y) = \mathsf{P}(Y_t \leq y|\mathcal{F}_{t-1})$ the cumulative distribution function (c.d.f.), by $p_t(y) = \mathsf{P}(Y_t = y|\mathcal{F}_{t-1})$ the probability density function, $y \in \mathbb{N}_0$, and by $\upsilon_t = \sqrt{\mathsf{VAR}\,(Y_t|\mathcal{F}_{t-1})}$ the standard deviation of the predictive

distribution, which is either a Poisson distribution with mean $\widehat{\lambda}_t$ or a negative binomial distribution with mean $\widehat{\lambda}_t$ and overdispersion coefficient $\widehat{\sigma}^2$ (recall Section 4 on 1-step-ahead prediction).

A tool for assessing the probabilistic calibration of the predictive distribution (see Gneiting *et al.* 2007) is the *probability integral transform* (PIT), which will follow a uniform distribution if the predictive distribution is correct. For count data Czado *et al.* (2009) define a non-randomized PIT value for the observed value y_t and the predictive distribution $P_t(y)$ by

$$F_t(u|y) = \begin{cases} 0, & u \leq P_t(y-1) \\ \dfrac{u - P_t(y-1)}{P_t(y) - P_t(y-1)}, & P_t(y-1) < u < P_t(y) \\ 1, & u \geq P_t(y) \end{cases}.$$

The mean PIT is then given by

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

To check whether $\overline{F}(u)$ is the c.d.f. of a uniform distribution Czado *et al.* (2009) propose plotting a histogram with H bins, where bin h has the height $f_j = \overline{F}(h/H) - \overline{F}((h-1)/H)$, $h = 1, \ldots, H$ (function `pit`). By default H is chosen to be 10. A U-shape indicates underdispersion of the predictive distribution, whereas an upside down U-shape indicates overdispersion. Gneiting *et al.* (2007) point out that the empirical coverage of central, e.g., 90% prediction intervals can be read off the PIT histogram as the area under the 90% central bins.

Marginal calibration is defined as the difference of the average predictive c.d.f. and the empirical c.d.f. of the observations, i.e.,

$$\frac{1}{n} \sum_{t=1}^{n} P_t(y) - \frac{1}{n} \sum_{t=1}^{n} \mathbb{1}(y_t \leq y) \tag{15}$$

for all $y \in \mathbb{R}$. In practice we plot the marginal calibration for values y in the range of the original observations (Christou and Fokianos 2015b) (function `marcal`). If the predictions from a model are appropriate the marginal distribution of the predictions resembles the marginal distribution of the observations and (15) should be close to zero. Major deviations from zero point to model deficiencies.

Gneiting *et al.* (2007) show that the calibration assessed by a PIT histogram or a marginal calibration plot is a necessary but not sufficient condition for a forecaster to be ideal. They advocate to favor the model with the maximal sharpness among all sufficiently calibrated models. Sharpness is the concentration of the predictive distribution and can be measured by the width of prediction intervals. A simultaneous assessment of calibration and sharpness summarized in a single numerical score can be accomplished by *proper scoring rules* (Gneiting *et al.* 2007). Denote a score for the predictive distribution P_t and the observation y_t by $s(P_t, y_t)$. A number of possible proper scoring rules is given in Table 1. The mean score for each corresponding model is given by $\sum_{t=1}^{n} s(P_t, y_t)/n$. Each of the different proper scoring rules captures different characteristics of the predictive distribution and its distance to the observed data (function `scoring`). Except for the normalized error score, the model with the lowest score is preferable. The mean squared error score is the only one which does not

Scoring rule	Abbreviation	Definition
squared error score	`sqerror`	$(y_t - \lambda_t)^2$
normalized squared error score	`normsq`	$(y_t - \lambda_t)^2/v_t^2$
Dawid-Sebastiani score	`dawseb`	$(y_t - \lambda_t)^2/v_t^2 + 2\log(v_t)$
logarithmic score	`logarithmic`	$-\log(p_t(y_t))$
quadratic (or Brier) score	`quadratic`	$-2p_t(y_t) + \|p_t\|^2$
spherical score	`spherical`	$-p_t(y_t)/\|p_t\|$
ranked probability score	`rankprob`	$\sum_{y=0}^{\infty}(P_t(y) - \mathbb{1}(y_t \leq y))^2$

Table 1: Definitions of proper scoring rules $s(P_t, y_t)$ (cf. Czado *et al.* 2009) and their abbreviations in the package; $\|p_t\|^2 = \sum_{y=0}^{\infty} p_t^2(y)$.

depend on the distribution and is also known as mean squared prediction error. The mean normalized squared error score measures the variance of the Pearson residuals and is close to one if the model is adequate. The Dawid-Sebastini score is a variant of this with an extra term to penalize overestimation of the standard deviation.

Other popular tools are model selection criteria like Akaike's information criterion (AIC) and the Bayesian information criterion (BIC) (functions `AIC` and `BIC`). The model with the lowest value of the respective information criterion is preferable. Denote the log-likelihood by $\widetilde{\ell}(\widehat{\boldsymbol{\theta}}, \widehat{\sigma}^2) = \sum_{t=1}^{n} \log\left(p_t(y_t)\right)$. Note that this is the true and not the quasi log-likelihood given in (7). Furthermore, $\ell(\widehat{\boldsymbol{\theta}}, \widehat{\sigma}^2)$ includes all constant terms which have been omitted on the right hand side of (7). The AIC and BIC are given by AIC $= -2\widetilde{\ell}(\widehat{\boldsymbol{\theta}}, \widehat{\sigma}^2) + 2df$ and BIC $= -2\widetilde{\ell}(\widehat{\boldsymbol{\theta}}, \widehat{\sigma}^2) + \log(n_{\text{eff}})df$, respectively. Here df is the total number of parameters (including the dispersion coefficient) and n_{eff} the number of effective observations (excluding those only used for initialization when argument `init.drop = TRUE`). The BIC generally yields more parsimonious models than the AIC. Note that for other distributions than the Poisson, $\widehat{\boldsymbol{\theta}}$ maximizes the quasi log-likelihood (7) but not $\widetilde{\ell}(\boldsymbol{\theta}, \sigma^2)$. In such cases the quasi information criterion (QIC), proposed by Pan (2001) for regression analysis based on the generalized estimating equations, is a properly adjusted alternative to the AIC (function `QIC`). We have verified by a simulation reported in Appendix B.3 that in case of a Poisson distribution the QIC approximates the AIC quite satisfactorily.

6. Intervention analysis

In many applications sudden changes or extraordinary events occur. Box and Tiao (1975) refer to such special events as interventions. This could be for example the outbreak of an epidemic in a time series which counts the weekly number of patients infected with a particular disease. It is of interest to examine the effect of known interventions, for example to judge whether a policy change had the intended impact, or to search for unknown intervention effects and find explanations for them *a posteriori*.

Fokianos and Fried (2010, 2012) model interventions affecting the location by including a deterministic covariate of the form $\delta^{t-\tau}\mathbb{1}(t \geq \tau)$, where τ is the time of occurrence and the decay rate δ is a known constant (function `interv_covariate`). This covers various types of interventions for different choices of the constant δ: a singular effect for $\delta = 0$ (spiky outlier), an exponentially decaying change in location for $\delta \in (0, 1)$ (transient shift) and a permanent

change of location for $\delta = 1$ (level shift). Similar to the case of covariates, the effect of an intervention is essentially additive for the linear model and multiplicative for the log-linear model. However, the intervention enters the dynamics of the process and therefore its effect on the linear predictor is not purely additive. Our package includes methods to test for such intervention effects developed by Fokianos and Fried (2010, 2012), suitably adapted to the more general model class described in Section 2. The linear predictor of a model with s types of interventions according to parameters $\delta_1, \ldots, \delta_s$ occurring at time points $\tau_1, \ldots, \tau_s$ reads

$$g(\lambda_t) = \beta_0 + \sum_{k=1}^{p} \beta_k \, \widetilde{g}(Y_{t-i_k}) + \sum_{\ell=1}^{q} \alpha_\ell g(\lambda_{t-j_\ell}) + \boldsymbol{\eta}^\top \boldsymbol{X}_t + \sum_{m=1}^{s} \omega_m \delta_m^{t-\tau_m} \mathbb{1}(t \geq \tau_m), \tag{16}$$

where ω_m, $m = 1, \ldots, s$ are the intervention sizes. At the time of its occurrence an intervention changes the level of the time series by adding the magnitude ω_m, for a linear model like (2), or by multiplying the factor $\exp(\omega_m)$, for a log-linear model like (3). In the following paragraphs we briefly outline the proposed intervention detection procedures and refer to the original articles for details.

Our package allows to test whether s interventions of certain types occurring at given time points, according to model (16), have an effect on the observed time series, i.e., to test the hypothesis $H_0 : \omega_1 = \ldots = \omega_s = 0$ against the alternative $H_1 : \omega_\ell \neq 0$ for some $\ell \in \{1, \ldots, s\}$. This is accomplished by employing an approximate score test (function `interv_test`). Under the null hypothesis the score test statistic $T_n(\tau_1, \ldots, \tau_s)$ has asymptotically a χ^2-distribution with s degrees of freedom, assuming some regularity conditions (Fokianos and Fried 2010, Lemma 1).

For testing whether a single intervention of a certain type occurring at an unknown time point τ has an effect, the package employs the maximum of the score test statistics $T_n(\tau)$ and determines a p value by a parametric bootstrap procedure (function `interv_detect`). If we consider a set D of time points at which the intervention might occur, e.g., $D = \{2, \ldots, n\}$, this test statistic is given by $\widetilde{T}_n = \max_{\tau \in D} T_n(\tau)$. The bootstrap procedure can be computed on multiple cores simultaneously (argument `parallel = TRUE`). The time point of the intervention is estimated to be the value τ which maximizes this test statistic. Our empirical observation is that such an estimator usually has a large variability. It is possible to speed up the computation of the bootstrap test statistics by using the model parameters used for generation of the bootstrap samples instead of estimating them for each bootstrap sample (argument `final.control_bootstrap = NULL`). This results in a conservative procedure, as noted by Fokianos and Fried (2012).

If more than one intervention is suspected in the data, but neither their types nor the time points of their occurrences are known, an iterative detection procedure is used (function `interv_multiple`). Consider the set of possible intervention times D as before and a set of possible intervention types Δ, e.g., $\Delta = \{0, 0.8, 1\}$. In a first step the time series is tested for an intervention of each type $\delta \in \Delta$ as described in the previous paragraph and the p values are corrected to account for multiple testing by the Bonferroni method. If none of the p values is below a previously specified significance level, the procedure stops and does not identify an intervention effect. Otherwise the procedure detects an intervention of the type corresponding to the lowest p value. In case of equal p values preference is given to interventions with $\delta = 1$, that is level shifts, and then to those with the largest test statistic. In a second step, the effect of the detected intervention is eliminated from the time series and the procedures starts anew and continues until no further intervention effects are detected. Finally, model (16) with all

detected intervention effects can be fitted to the data to estimate the intervention sizes and the other parameters jointly (which are in general different than when estimated in separate steps). Note that statistical inference for this final model fit has to be done with care.

In practical applications, the decay rate δ of a particular intervention effect is often unknown and needs to be estimated. Since the parameter δ is not identifiable when the corresponding intervention size ω is zero, its estimation is nonstandard. As suggested by a reviewer, estimation could be carried out by profiling the likelihood over this parameter. For a single intervention effect this could be done by computing the (quasi) ML estimator of all other parameters for a given decay rate δ. This is repeated for all $\delta \in \Delta$, where Δ is a set of possible decay rates, and the value which results in the maximum value of the log-likelihood is chosen (apply the function `tsglm` repeatedly). Note that this approach affects the validity of the usual statistical inference for the other parameters.

Liboschik *et al.* (2016) study a model for external intervention effects (modeled by external covariate effects, recall (6) and the related discussion) and compare it to internal intervention effects studied in the two aforementioned publications (argument `external`).

7. Usage of the package

The version of the **tscount** package used for the computations in this paper is provided in the supplementary material. The most recent stable version is distributed via the Comprehensive R Archive Network (CRAN) and is available at `https://CRAN.R-project.org/package=tscount`. A current development version is available from the project's website `https://tscount.R-Forge.R-project.org` on the development platform R-Forge. After installation of the package it can be loaded in R by typing `library("tscount")`.

The central function for fitting a GLM for count time series is `tsglm`, whose help page (accessible by `?tsglm`) is a good starting point to become familiar with the usage of the package. The most relevant functions of the package are summarized in Table 2. There are many standard S3 methods available for well-known generic functions. A detailed description of the functions' usage including examples can be found on the accompanying help pages. The package provides some data sets which are also listed in Table 2.

In the following sections we demonstrate typical applications of the package by two data examples.

7.1. Campylobacter infections in Canada

We first analyze the number of campylobacterosis cases (reported every 28 days) in the North of Québec in Canada. The data are shown in Figure 1 and were first reported by Ferland *et al.* (2006). These data are made available in the package (object `campy`). We fit a model to this time series using the function `tsglm`. Following the analysis of Ferland *et al.* (2006) we fit model (2) with the identity link function, defined by the argument `link`. For taking into account serial dependence we include a regression on the previous observation. Seasonality is captured by regressing on λ_{t-13}, the unobserved conditional mean 13 time units (which is about one year) back in time. The aforementioned specification of the model for the linear predictor is assigned by the argument `model`, which has to be a list. We also include the two intervention effects detected by Fokianos and Fried (2010) in the model by suitably chosen covariates provided by the argument `xreg`. We compare a fit of a Poisson with that of a

	Name	Description
Functions	`tsglm`	Fitting a model to given data (class '`tsglm`')
	`tsglm.sim`	Simulating from the model
	Generic functions with methods for class '`tsglm`':	
	`plot`	Diagnostic plots
	`se`	Standard errors and confidence intervals
	`summary`	Summary of the fitted model
	`fitted`	Fitted values
	`residuals`	Residuals
	`AIC`	Akaike's information criterion
	`BIC`	Bayesian information criterion
	`QIC`	Quasi information criterion
	`pit`	Probability integral transform histogram
	`marcal`	Marginal calibration plot
	`scoring`	Proper scoring rules
	`predict`	Prediction
	`interv_test`	Test for intervention effects
	`interv_detect`	Detection of single intervention effects
	`interv_multiple`	Iterative detection of multiple intervention effects
Data sets	`campy`	Campylobacter infections in Québec
	`ecoli`	E. coli infections in North Rhine-Westphalia (NRW)
	`ehec`	EHEC/HUS infections in NRW
	`influenza`	Influenza infections in NRW
	`measles`	Measles infections in NRW

Table 2: Most important functions of the R package **tscount** and the included data sets.

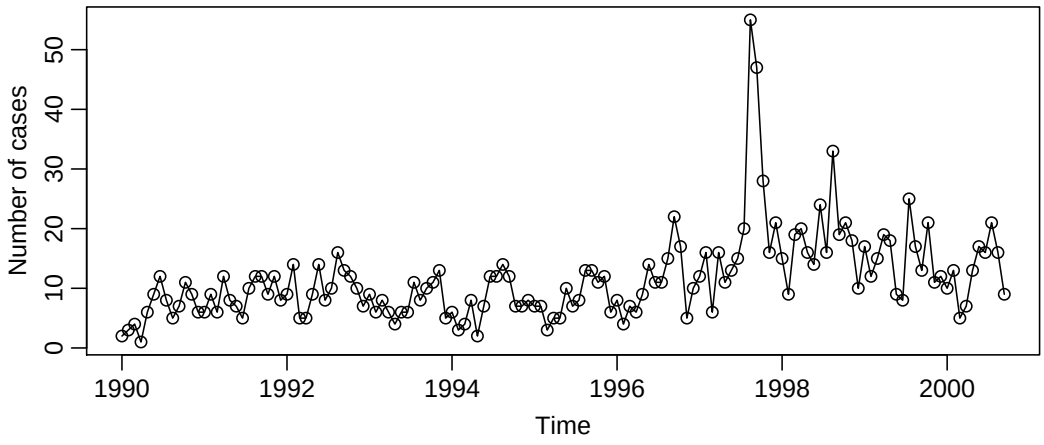

Figure 1: Number of campylobacterosis cases (reported every 28 days) in the North of Québec in Canada.

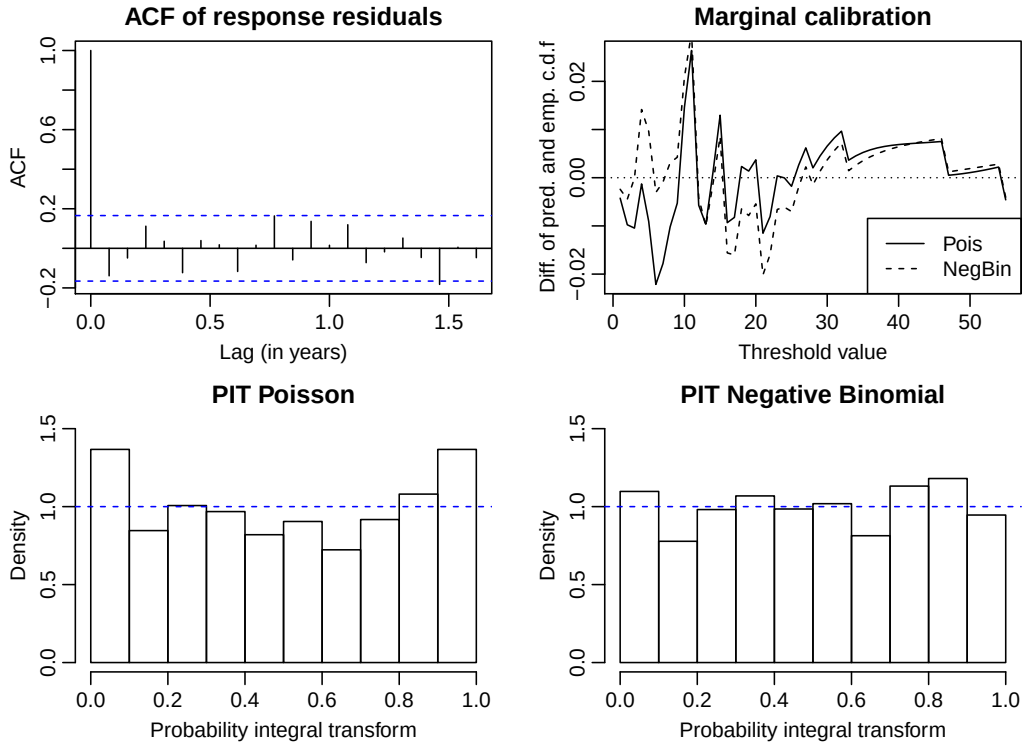

Figure 2: Diagnostic plots after model fitting to the campylobacterosis data.

negative binomial conditional distribution, specified by the argument `distr`. The call for both model fits is then given by:

```
R> interventions <- interv_covariate(n = length(campy), tau = c(84, 100),
+    delta = c(1, 0))
R> campyfit_pois <- tsglm(campy, model = list(past_obs = 1, past_mean = 13),
+    xreg = interventions, distr = "poisson")
R> campyfit_nbin <- tsglm(campy, model = list(past_obs = 1, past_mean = 13),
+    xreg = interventions, distr = "nbinom")
```

The resulting fitted models `campyfit_pois` and `campyfit_nbin` have class '`tsglm`', for which a number of methods is provided (see help page), including `summary` for a detailed model summary and `plot` for diagnostic plots. The diagnostic plots like in Figure 2 can be produced by:

```
R> acf(residuals(campyfit_pois), main = "ACF of response residuals")
R> marcal(campyfit_pois, main = "Marginal calibration")
R> lines(marcal(campyfit_nbin, plot = FALSE), lty = "dashed")
R> legend("bottomright", legend = c("Pois", "NegBin"), lwd = 1,
+    lty = c("solid", "dashed"))
R> pit(campyfit_pois, ylim = c(0, 1.5), main = "PIT Poisson")
R> pit(campyfit_nbin, ylim = c(0, 1.5), main = "PIT Negative Binomial")
```

The response residuals are identical for the two conditional distributions. Their empirical

autocorrelation function, shown in Figure 2 (top left), does not exhibit any serial correlation or seasonality which has not been taken into account by the models. Figure 2 (bottom left) points to an approximately U-shaped PIT histogram indicating that the Poisson distribution is not adequate for model fitting. As opposed to this, the PIT histogram which corresponds to the negative binomial distribution appears to approach uniformity better. Hence the probabilistic calibration of the negative binomial model is satisfactory. The marginal calibration plot, shown in Figure 2 (top right), is inconclusive. As a last tool we consider the scoring rules for the two distributions:

```
R> rbind(Poisson = scoring(campyfit_pois), NegBin = scoring(campyfit_nbin))

        logarithmic quadratic spherical rankprob dawseb normsq sqerror
Poisson       2.750  -0.07669   -0.2751    2.200  3.662 1.3081   16.51
NegBin        2.722  -0.07800   -0.2766    2.185  3.606 0.9643   16.51
```

All considered scoring rules are in favor of the negative binomial distribution. Based on the PIT histograms and the results obtained by the scoring rules, we decide for the negative binomial model. The degree of overdispersion seems to be small, as the estimated overdispersion coefficient `sigmasq` of 0.0297 given in the output below is close to zero.

```
R> summary(campyfit_nbin)

Call:
tsglm(ts = campy, model = list(past_obs = 1, past_mean = 13),
    xreg = interventions, distr = "nbinom")

Coefficients:
             Estimate  Std.Error  CI(lower)  CI(upper)
(Intercept)    3.3184     0.7851     1.7797      4.857
beta_1         0.3690     0.0696     0.2326      0.505
alpha_13       0.2198     0.0942     0.0352      0.404
interv_1       3.0810     0.8560     1.4032      4.759
interv_2      41.9541    12.0914    18.2554     65.653
sigmasq        0.0297         NA         NA         NA
Standard errors and confidence intervals (level =  95 %) obtained
by normal approximation.

Link function: identity
Distribution family: nbinom (with overdispersion coefficient 'sigmasq')
Number of coefficients: 6
Log-likelihood: -381.1
AIC: 774.2
BIC: 791.8
QIC: 787.6
```

The coefficient `beta_1` corresponds to regression on the previous observation, `alpha_13` corresponds to regression on values of the conditional mean thirteen units back in time. The output

reports the estimation of the overdispersion coefficient σ^2, which is related to the dispersion parameter ϕ of the negative binomial distribution by $\phi = 1/\sigma^2$. Accordingly, the fitted model for the number of new infections Y_t in time period t is given by $Y_t|\mathcal{F}_{t-1} \sim \text{NegBin}(\lambda_t, 33.61)$ with

$$\lambda_t = 3.32 + 0.37Y_{t-1} + 0.22\lambda_{t-13} + 3.08\mathbb{1}(t = 84) + 41.95\mathbb{1}(t \geq 100), \quad t = 1, \ldots, 140.$$

The standard errors of the estimated regression parameters and the corresponding confidence intervals in the summary above are based on the normal approximation given in (11). For the additional overdispersion coefficient `sigmasq` of the negative binomial distribution there is no analytical approximation available for its standard error. Alternatively, standard errors (and confidence intervals, not shown here) of the regression parameters and the overdispersion coefficient can be obtained by a parametric bootstrap (which takes about 15 minutes computation time on a single 3.2 GHz processor for 500 replications):

```
R> se(campyfit_nbin, B = 500)$se

(Intercept)      beta_1    alpha_13    interv_1    interv_2     sigmasq
    0.89850     0.06941     0.10136     0.93836    11.16856     0.01460

Warning message:
In se.tsglm(campyfit_nbin, B = 500) :
  The overdispersion coefficient 'sigmasq' could not be estimated
in 5 of the 500 replications. It is set to zero for these
replications. This might to some extent result in a biased estimation
of its true variability.
```

Estimation problems for the dispersion parameter (see warning message) occur occasionally for models where the true overdispersion coefficient σ^2 is small, i.e., which are close to a Poisson model; see Appendix B.2. The bootstrap standard errors of the regression parameters are slightly larger than those based on the normal approximation. Note that neither of the approaches reflects the additional uncertainty induced by the model selection.

7.2. Road casualties in Great Britain

Next we study the monthly number of killed drivers of light goods vehicles in Great Britain between January 1969 and December 1984 shown in Figure 3. This time series is part of a dataset which was first considered by Harvey and Durbin (1986) for studying the effect of compulsory wearing of seat belts introduced on 31 January 1983. The dataset, including additional covariates, is available in R in the object `Seatbelts`. In their paper Harvey and Durbin (1986) analyze the numbers of casualties for drivers and passengers of cars, which are so large that they can be treated with methods for continuous-valued data. The monthly number of killed drivers of vans analyzed here is much smaller (its minimum is 2 and its maximum 17) and therefore methods for count data are to be preferred.

For model selection we only use the data until December 1981. We choose the log-linear model with the logarithmic link because it allows for negative covariate effects. We aim at capturing the short range serial dependence by a first order autoregressive term and the yearly

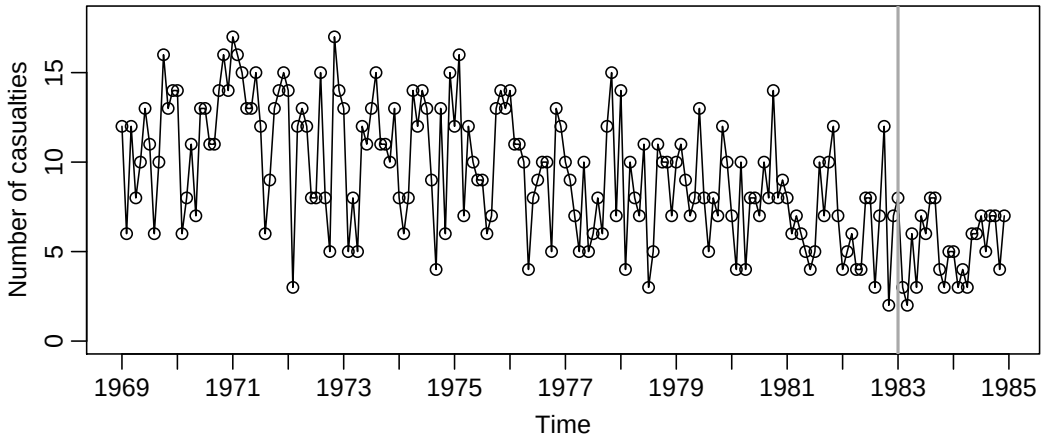

Figure 3: Monthly number of killed van drivers in Great Britain. The introduction of compulsory wearing of seat belts on 31 January 1983 is marked by a vertical line.

seasonality by a 12th order autoregressive term. Both of these terms are declared by the list element named `past_obs` of the argument `model`. Following Harvey and Durbin (1986) we use the real price of petrol as an explanatory variable. We also include a deterministic covariate describing a linear trend. Both covariates are provided by the argument `xreg`. Based on PIT histograms, a marginal calibration plot and the scoring rules (not shown here) we find that the Poisson distribution is sufficient for modeling. The model is fitted by the call:

```
R> timeseries <- Seatbelts[, "VanKilled"]
R> regressors <- cbind(PetrolPrice = Seatbelts[, c("PetrolPrice")],
+    linearTrend = seq(along = timeseries)/12)
R> timeseries_until1981 <- window(timeseries, end = 1981 + 11/12)
R> regressors_until1981 <- window(regressors, end = 1981 + 11/12)
R> seatbeltsfit <- tsglm(timeseries_until1981,
+    model = list(past_obs = c(1, 12)), link = "log", distr = "poisson",
+    xreg = regressors_until1981)
R> summary(seatbeltsfit, B = 500)

Call:
tsglm(ts = timeseries_until1981, model = list(past_obs = c(1,
    12)), xreg = regressors_until1981, link = "log", distr = "pois")

Coefficients:
             Estimate  Std.Error  CI(lower)  CI(upper)
(Intercept)    1.8347    0.38343     1.2817     2.7490
beta_1         0.0866    0.08312    -0.0902     0.2267
beta_12        0.1535    0.09009    -0.0488     0.2947
PetrolPrice    0.7787    2.46641    -4.1364     5.5425
linearTrend   -0.0303    0.00855    -0.0475    -0.0161
Standard errors and confidence intervals (level =  95 %) obtained
by parametric bootstrap with 500 replications.
```

```
Link function: log
Distribution family: poisson
Number of coefficients: 5
Log-likelihood: -396.2
AIC: 802.4
BIC: 817.6
QIC: 802.4
```

Accordingly, the fitted model for the number of van drivers Y_t killed in month t is given by $Y_t|\mathcal{F}_{t-1} \sim \text{Poisson}(\lambda_t)$ with

$$\log(\lambda_t) = 1.83 + 0.09Y_{t-1} + 0.15Y_{t-12} + 0.78X_t - 0.03t/12, \quad t = 1, \ldots, 156,$$

where X_t denotes the real price of petrol at time t.

The estimated coefficient `beta_1` corresponding to the first order autocorrelation is very small and even slightly below the size of its approximate standard error, indicating that there is no notable dependence on the number of killed van drivers of the preceding week. We find a seasonal effect captured by the twelfth order autocorrelation coefficient `beta_12`. Unlike in the model for the car drivers by Harvey and Durbin (1986), the petrol price does not seem to influence the number of killed van drivers. An explanation might be that vans are much more often used for commercial purposes than cars and that commercial traffic is less influenced by the price of fuel. The linear trend can be interpreted as a yearly reduction of the number of casualties by a factor of 0.97 (obtained by exponentiating the corresponding estimated coefficient), i.e., on average we expect 3% fewer killed van drivers per year (which is below one in absolute numbers).

Based on the model fitted to the training data until December 1981, we can predict the number of road casualties in 1982 given the respective petrol price. Coherent, i.e., integer-valued, forecasts could be obtained by rounding the predictions. A graphical representation of the following predictions is given in Figure 4.

```
R> timeseries_1982 <- window(timeseries, start = 1982, end = 1982 + 11/12)
R> regressors_1982 <- window(regressors, start = 1982, end = 1982 + 11/12)
R> predict(seatbeltsfit, n.ahead = 12, level = 0.9, global = TRUE,
+    B = 2000, newxreg = regressors_1982)$pred

      Jan  Feb  Mar  Apr  May  Jun  Jul  Aug  Sep  Oct  Nov  Dec
1982 7.72 7.44 7.56 7.41 7.20 7.00 7.16 7.86 7.53 7.86 8.06 7.48
```

Finally, we test whether there was an abrupt shift in the number of casualties occurring when the compulsory wearing of seat belts is introduced on 31 January 1983. The approximate score test described in Section 6 is applied:

```
R> seatbeltsfit_alldata <- tsglm(timeseries, link = "log",
+    model = list(past_obs = c(1, 12)), xreg = regressors, distr = "poisson")
R> interv_test(seatbeltsfit_alldata, tau = 170, delta = 1, est_interv = TRUE)

        Score test on intervention(s) of given type at given time
```

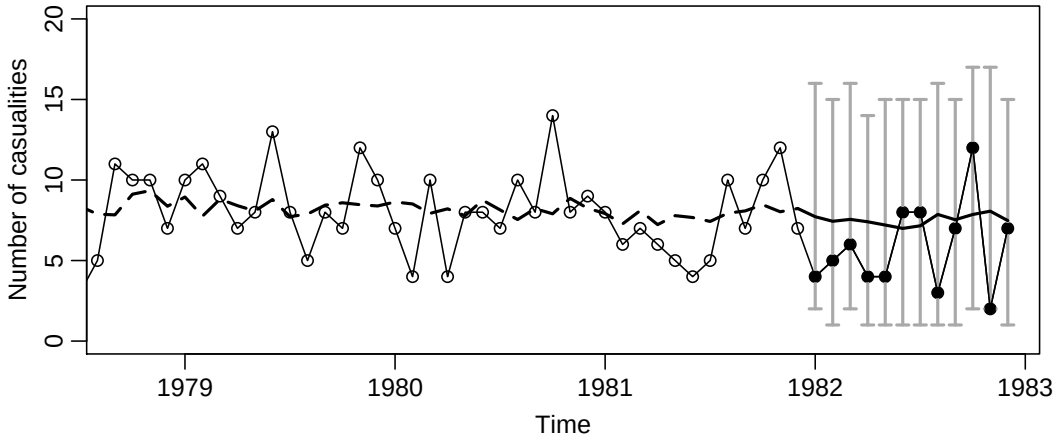

Figure 4: Fitted values (dashed line) and predicted values (solid line) according to the model with the Poisson distribution. Prediction intervals (gray bars) are designed to ensure a global coverage rate of 90%. They are chosen to have minimal length and are based on a simulation with 2000 replications.

```
Chisq-Statistic: 1.109 on 1 degree(s) of freedom, p-value: 0.2923

Fitted model with the specified intervention:

Call:
tsglm(ts = fit$ts, model = model_extended, xreg = xreg_extended,
    link = fit$link, distr = fit$distr)

Coefficients:
(Intercept)       beta_1      beta_12  PetrolPrice  linearTrend
    1.93298      0.08178      0.13943      0.41863     -0.03466
   interv_1
   -0.21683
```

With a p value of 0.29 the null hypothesis of no intervention cannot be rejected at a 5% significance level. Note that this result does not rule out that there is an effect of the seatbelts law which is either too small for being significant or of a different type than it is tested for. For illustration we fit the model under the alternative of a level shift after the introduction of the seatbelts law (see the output above). The multiplicative effect size of the intervention is found to be 0.805. This indicates that according to this model fit -19.5% less van drivers are killed after the law enforcement. For comparison, Harvey and Durbin (1986) estimate a reduction of 18% for the number of killed car drivers.

8. Comparison with other software packages

In this section we review functions (and the corresponding models) from other R packages which can be employed for count time series analysis. Many of them have been published only very recently, a fact that demonstrates the raising interest in count time series analysis. We

discuss how these packages differ from our package **tscount**. For illustration we use the time series of campylobacter infections analyzed in Section 7.1 ignoring the intervention effects. For the presentation of other models we use a notation parallel to the one used in the previous sections to highlight similarities. Interpretation of the final model should be done carefully, though.

We consider a large number of somehow related packages which makes this comparison quite extensive yet interesting for those readers who want guidance on choosing the most appropriate package for their data. In the first subsection we present packages for independent data and in the second subsection we discuss packages for dependent data.

8.1. Packages for independent data

We start reviewing functions which have been introduced for independent observations but can, with certain limitations, be employed for time series whose temporal dependence is rather simple. This is exemplarily discussed in the following paragraph.

The function `glm` in package **stats** and, for the negative binomial distribution, the function `glm.nb` in package **MASS** (Venables and Ripley 2002) can fit standard GLMs to count time series with the iteratively reweighted least squares (IRLS) algorithm. Just like with our `tsglm` function, one can choose the identity or logarithmic link in combination with a Poisson or negative binomial conditional distribution. Standard GLMs have been introduced to model independent but not identically distributed observations. In principle, one could also fit simple models for time series by including lagged values of the time series, i.e., $Y_{t-i_1}, \ldots, Y_{t-i_p}$, as covariates. However, the `glm` function has several limitations; the most important being that it does not allow for regression on past values of the conditional mean. For example, the `glm` function cannot be used to fit the model which included stochastic seasonality; recall Section 7.1. Furthermore, the `glm` function does not induce the constraints on the vector of parameters given in Section 3, which are necessary to ensure stationarity of the fitted process. Models which are violating these parameter constraints are generally not suitable for prediction. We have also experienced that `glm` occasionally does not find good starting values for its optimization procedure such that it returns an error and requests the user to provide starting values. At least for the very simple case of a Poisson INGARCH$(1, 0)$ model fitted to the campylobacterosis data the `glm` function performs well and we obtain very similar parameter estimates to those obtained with the `tsglm` function:

```
R> campydata <- data.frame(ts = campy[-1], lag1 = campy[-length(campy)])
R> coef(glm(ts ~ lag1, family = poisson(link = "identity"),
+    data = campydata))

(Intercept)        lag1
     4.0322      0.6556

R> coef(tsglm(campy, model = list(past_obs = 1), link = "identity"))

(Intercept)      beta_1
     4.0083      0.6501
```

As described in more detail in Appendix A.3, a fit by the `glm` function can be used as a starting value to the function `tsglm`.

The class of generalized additive models for location, scale and shape (GAMLSS) has been introduced by Rigby and Stasinopoulos (2005) as an extension of a GLM and a generalized additive model (GAM). In addition to the location parameter further parameters of the conditional distribution can be modeled as functions of the explanatory variables. In the following example we use the package **gamlss** (authored by Rigby and Stasinopoulos 2005; Stasinopoulos and Rigby 2007) to fit an INGARCH$(1, 0)$ model to the campylobacterosis data. The overdispersion coefficient σ_t^2 of the negative binomial distribution is not constant but changes with time according to the equation

$$\sigma_t^2 = \exp\left(\beta_0^* + \beta_1^* \log(Y_{t-1} + 1)\right).$$

```
R> library("gamlss")
R> gamlss(ts ~ lag1, sigma.formula = ~ log(lag1 + 1), data = campydata,
+    family = NBI(mu.link = "identity", sigma.link = "log"))[c(25, 43)]

GAMLSS-RS iteration 1: Global Deviance = 803.7
GAMLSS-RS iteration 2: Global Deviance = 803.7
GAMLSS-RS iteration 3: Global Deviance = 803.7
$mu.coefficients
(Intercept)        lag1
     3.8409      0.6768

$sigma.coefficients
  (Intercept) log(lag1 + 1)
      -4.2986        0.7167
```

The possibility of a time dependent dispersion coefficient does not improve the fit for this data example (according to the AIC, which is 811.72 compared to 811.64 for a model with constant overdispersion coefficient) but might be quite useful for other data examples. However, it is clear that such a complex model yields more uncertainty of the parameter estimations (i.e., larger standard errors, which are not shown here).

The package **ZIM** (Yang, Zamba, and Cavanaugh 2017) fits zero-inflated models (ZIM) for count time series with excess zeros. These models are suitable for data where the value zero occurs more frequently than it would be expected when assuming other count time series models. The main idea of these models is to replace the ordinary Poisson or negative binomial distribution by its respective zero-inflated version, which is a mixture of a singular distribution in zero (with probability ω_t) and a Poisson or negative binomial distribution (with probability $1 - \omega_t$), respectively. The model proposed by Yang, Zamba, and Cavanaugh (2013) allows both, the probability ω_t and the conditional mean λ_t of the ordinary count data distribution, to vary over time. The conditional mean λ_t is modeled by using a logistic regression model. The probability ω_t is modeled by a GLM with the logistic link. Other methods for count data with excess zeros, which also have these limitations, are provided by the well-established functions `zeroinfl` and `hurdle` from the package **pscl** (Zeileis, Kleiber, and Jackman 2008). However, the package **ZIM** includes an extension of ZIM to state space models, which is treated in the next section. The parameters of a ZIM are fitted by the function `zim` employing an EM algorithm. Zero-inflation models are definitely appealing for count time series which occasionally exhibit excess zeros. For our data example of campylobacter infections, which does not include any zero observations, ZIMs are not applicable.

The current version of the **tscount** package considered in this paper is limited to modeling univariate data. A possible extension to models for vectors of counts is provided by the package **VGAM** (Yee 2017) introduced by Yee (2015). The function `vglm` in this package fits a vector GLM (VGLM) (see Yee and Wild 1996) where the conditional density function of a d-dimensional response vector $\boldsymbol{Y}_t$ given an r-dimensional covariate vector $\boldsymbol{X}_t$ is assumed to be of the form

$$f(\boldsymbol{Y}_t|\boldsymbol{X}_t;\boldsymbol{H}) = h(\boldsymbol{Y}_t, \nu_1, \ldots, \nu_s),$$

where $\nu_j = \boldsymbol{\eta}_j^\top \boldsymbol{X}_t$, $j = 1, \ldots, s$ and $h(\cdot)$ is a suitably defined function. The model parameters are given by the $(r \times s)$-dimensional parameter matrix $\boldsymbol{H} = (\boldsymbol{\eta}_1^\top, \ldots, \boldsymbol{\eta}_s^\top)^\top$. Choosing $d = s = 1$ results in the special case of an ordinary, univariate GLM. We demonstrate a fit of an INGARCH$(1, 0)$ model to the campylobacterosis data by the following code:

```
R> library("VGAM")
R> coef(vglm(ts ~ lag1, family = poissonff(link = "identitylink"),
+    data = campydata))

(Intercept)        lag1
     4.0322      0.6556
```

We note that the function `vglm` produces exactly the same output as the function `glm` for this special case. The function `vgam` from the same package would allow to fit an even more general vector generalized additive model (VGAM), which is a multivariate generalization of a generalized additive model (GAM), see Yee (2015) for more details.

Due to the aforementioned limitations of the procedures developed for independent data we would generally suggest the use of the function `tsglm` for modeling count time series. However, in certain situations, where features of the data are currently not supported by `tsglm`, the aforementioned packages can be employed with care; recall the second paragraph of this section. For count time series with many zeros one might want to consider using, for example, the package **ZIM**. If there are reasons to assume a time-varying overdispersion coefficient, the package **gamlss** is a good choice. Multivariate count time series could be analyzed with the package **VGAM**.

8.2. Packages for time series data

In this section we present R packages developed for count time series data.

The package **acp** (Siakoulis 2015) has been published recently and provides maximum likelihood fitting of autoregressive conditional Poisson (ACP) regression models. These are the INGARCH models given by (2); see Section 2. The **acp** package also allows to include covariate effects. In its latest version 2.1, which has been published in December 2015, the package has been extended to fit models of general order p and q. The `tsglm` function of our package includes these models as special cases and is more general in the following aspects:

- The **acp** package is different in many technical details. Notably, it does not allow to incorporate the parameter constraints given in Section 3.
- Quasi maximum likelihood fitting allows to choose a more flexible negative binomial model instead of a Poisson model (argument `distr = "nbinom"`).

- The `tsglm` function additionally comprises a log-linear model (argument `link = "log"`), which is more adequate for many count time series.
- The `tsglm` function allows for more flexible dependence modeling by allowing arbitrary specification of dynamics. This flexibility is missing by the `acp` function for model fitting because it requires all variables up to a given order to be included (e.g., $\lambda_{t-1}, \ldots, \lambda_{t-12}$ and not just λ_{t-12}). For instance, `tsglm` allows for stochastic seasonality (see Section 7.1).
- The `tsglm` function differentiates between covariates with so-called external and internal effect (see Equation 6 and the accompanying discussion).

In the following example, an INGARCH$(1, 1)$ model (ignoring the seasonal effect) is fitted to the campylobacterosis data analyzed in Section 7.1:

```
R> library("acp")
R> coef(acp(campy ~ -1, p = 1, q = 1))

[1] 2.5320 0.5562 0.2295

R> coef(tsglm(campy, model = list(past_obs = 1, past_mean = 1)))

(Intercept)      beta_1     alpha_1
     2.3890      0.5183      0.2693
```

The parameter estimations obtained by the `acp` function are very similar to those obtained by the `tsglm` function when fitting the same model.

The class of generalized linear autoregressive moving average (GLARMA) models combines GLM with ARMA processes. A software implementation is available in the package **glarma** (Dunsmuir and Scott 2015). The GLARMA model assumes the conditional distribution of Y_t given the past $\mathcal{F}_{t-1}$ to be Poisson or negative binomial with mean λ_t and density $f(Y_t|\lambda_t)$, with λ_t given by

$$g(\lambda_t) = \boldsymbol{\eta}^\top \boldsymbol{X}_t + O_t + Z_t,$$

where O_t is an offset term. An intercept is included by choosing the first column of the time-varying covariate matrix $\boldsymbol{X}_t$ to be the vector $(1, \ldots, 1)^\top$. Serial correlation is induced by an autoregressive moving average (ARMA) structure of Z_t, which is given by

$$Z_t = \sum_{k=1}^{p} \phi_k (Z_{t-i_k} + e_{t-i_k}) + \sum_{\ell=1}^{q} \psi_\ell e_{t-j_\ell}.$$

Hereby the process $\{Z_t : t \in \mathbb{N}\}$ is defined by means of residuals e_t which can be possibly rescaled, see (12) and (13). In the example below we choose Pearson residuals. For the link function $g(\cdot)$, the **glarma** package currently supports only the logarithm but not the identity, which is available in our function `tsglm`. Like in our package, the user can specify the model order by considering the sets $P = \{i_1, \ldots, i_p\}$ and $Q = \{j_1, \ldots, j_q\}$. The formulation of the GLARMA model we consider describes the modeling possibilities provided by the **glarma** package. In fact, this formulation is more general than the accompanying article by Dunsmuir and Scott (2015), where the authors consider the case $Q = \{1, \ldots, q\}$ and $P = \{1, \ldots, p\}$.

Choosing P and Q, in the context of GLARMA modeling, should be done cautiously (see Dunsmuir and Scott 2015, Section 3.4). Our limited experience shows that the minimum element of the set Q should be chosen in such a way that it is larger than the maximum element of P for avoiding errors. Unlike ordinary ARMA models, GLARMA models are not driven by random innovations but by residuals e_t. Note that the model fitted by the function `tsglm` is also related to ARMA processes, see (20) in the Appendix. The function `glarma` implements maximum likelihood fitting of a GLARMA model. We compare the following model fitted to the campylobacterosis data by the function `glarma` with a fit by `tsglm` (see Section 7.1, but without the intervention effects):

```
R> library("glarma")
R> glarmaModelEstimates(glarma(campy, phiLags = 1:3, thetaLags = 13,
+    residuals = "Pearson", X = cbind(intercept = rep(1, length(campy))),
+    type = "NegBin"))[c("Estimate", "Std.Error")]

          Estimate Std.Error
intercept  2.34110   0.10757
phi_1      0.22101   0.03940
phi_2      0.05978   0.04555
phi_3      0.09784   0.04298
theta_13   0.08602   0.03736
alpha     10.50823   1.91232
```

With the notation introduced above the fitted model for the number of new infections Y_t in time period t is given by $Y_t|\mathcal{F}_{t-1} \sim \text{NegBin}(\lambda_t, 10.51)$ with $\log(\lambda_t) = 2.34 + Z_t$ and

$$Z_t = 0.22(Z_{t-1} + e_{t-1}) + 0.06(Z_{t-2} + e_{t-2}) + 0.1(Z_{t-3} + e_{t-3}) + 0.09e_{t-13}.$$

We focus on the models' capability to explain the serial correlation which is present in the data. Considering the GLARMA model, a choice of $P = \{1, 2, 3\}$ and $Q = \{13\}$ leaves approximately uncorrelated residuals (see the autocorrelation function in Figure 5 (top right)). For the model class fitted by our function `tsglm` we have chosen the more parsimonious model with $P = \{1\}$ and $Q = \{13\}$ to obtain a fit with approximately uncorrelated residuals. Figure 6 shows that both models seem to provide an adequate fit to the given data. The package **glarma** provides a collection of functions which can be applied to a fitted GLARMA model. For example it provides a function for testing whether there exists serial dependence and it offers tools for model diagnostics. To conclude, both models are able to explain quite general forms of serial correlation but the role of the dependence parameters is quite different and any results should be interpreted carefully. A more detailed comparison would be interesting but is beyond the scope of this paper.

Another class of models, which is closely related to the GLARMA models, are the so-called generalized autoregressive moving average (GARMA) models developed by Benjamin, Rigby, and Stasinopoulos (2003). Dunsmuir and Scott (2015, Section 3) remark that both model classes are similar in their structure but they have some important differences. The GARMA model is formulated by

$$g(\lambda_t) = \boldsymbol{\eta}^\top \boldsymbol{X}_t + \sum_{k=1}^{p} \phi_k \left(g(Y_{t-k}) - \boldsymbol{\eta}^\top \boldsymbol{X}_{t-k} \right) + \sum_{\ell=1}^{q} \psi_\ell \left(g(Y_{t-k}) - g(\lambda_{t-\ell}) \right),$$

where the notation follows the GLARMA notation. Compared to the GLARMA model, the GARMA model does not include an offset and the ARMA structure applies to values which are transformed by the link function g, i.e., on the scale of the linear predictor. In case of a logarithmic link, the observations Y_t are replaced by $\max(Y_t, c)$ for a threshold $c \in (0, 1)$, such that $g(Y_t)$ is well-defined. In our package this problem is handled by replacing Y_t with $Y_t + 1$. The package **gamlss.util** (Stasinopoulos, Rigby, and Eilers 2016) contains the function `garmaFit` for fitting such GARMA models. Like ordinary GLMs, these models are fitted by maximum likelihood employing the IRLS algorithm. As pointed out on the accompanying help page, the function `garmaFit` does not guarantee stationarity of the fitted model. Additionally, the function `garmaFit` does not allow to specify serial dependence of higher order without including all lower orders, which would be necessary for parsimoniously describing stochastic seasonality. The following example shows a fit of a negative binomial GARMA model of order $p = 1$ and $q = 1$ with link $g(\cdot) = \log(\cdot)$ to the campylobacterosis data:

```
R> library("gamlss.util")
R> coef(garmaFit(campy ~ 1, order = c(1, 1), family = NBI(mu.link = "log")))

deviance of linear model=  891.1
deviance of  garma model=  803.3
beta.(Intercept)              phi            theta
          2.6216           0.7763          -0.2917
```

In the above output the AR coefficient ϕ_1 is named `phi` and the MA coefficient ψ_1 `theta`. The function `garma` from the package **VGAM** (Yee 2017) is an alternative implementation for fitting GARMA models. However, the accompanying help page warns that this function is still in premature stage and points to potential problems with the initialization (in version 1.0-1 of the package). In addition, `garma` allows only for autoregressive modeling (i.e., $q = 0$) and the negative binomial distribution is not supported. Hence our example can only show a fit of a Poisson GARMA model of order $p = 1$ and $q = 0$ to the campylobacterosis data:

```
R> coef(vglm(campy ~ 1, family = garma(link = "loge", p.ar.lag = 1,
+    q.ma.lag = 0, coefstart = c(0.1, 0.1)),
+    control = vglm.control(stepsize = 0.01, maxit = 700)))

(Intercept)       (lag1)
      2.561        0.646
```

In this example the estimated coefficient for the autoregressive term is larger than one, which suggests that the fitted process is not stationary. We could not find settings for which the functions `garmaFit` and `garma` fit the same model and give identical or at least similar results. Due to the close relationship of GARMA and GLARMA models we refrain from presenting a comparison to a fit with our package and refer to the comparison in the previous paragraph made for GLARMA models.

The models presented so far are determined by a single source of randomness, i.e., given all past observations, uncertainty is only induced by the Poisson or negative binomial distribution from which the observations are assumed to be drawn. These models belong to the class of observation-driven models according to the classification of Cox (1981). In the following

paragraphs we present parameter-driven models. These models are determined by multiple sources of randomness introduced by one or more innovation processes. Helske (2017b) comments on the merits of both approaches. He argues that parameter-driven models are appealing because they allow to introduce even multiple latent structures in a flexible way. On the other hand, he observes that observation-driven models like the ones we consider are of advantage for prediction because of their explicit dependence on past observations and covariates.

The package **surveillance** (Salmon, Schumacher, and Höhle 2016) includes methods for online change point detection in count time series. The included function `hhh4` fits the model proposed by Held, Höhle, and Hofmann (2005), where it is assumed that $Y_t \sim \text{NegBin}(\lambda_t, \phi)$. The conditional mean λ_t is given by

$$\lambda_t = \exp\left(\beta_0 + \delta t + \gamma_t\right) + \beta_1 Y_{t-1},$$

where γ_t is a periodic function describing a seasonal effect. The exponential function is applied to the linear trend and the seasonal effect but not to the autoregressive component such that this model is not a GLM (since the linear predictor is not linear in the parameters) but could instead be regarded as a generalized additive model (GAM). Another type of model considered by the **surveillance** package is the hierarchical time series (HTS) model, as proposed by Manitz and Höhle (2013) based on the work by Heisterkamp, Dekkers, and Heijne (2006). This particular state space model accounts for serial dependence by a time-varying intercept. More precisely, it is assumed that the conditional mean λ_t is given by

$$\lambda_t = \exp\left(\beta_{0,t} + \delta t + \gamma_t + \boldsymbol{\eta}^\top \boldsymbol{X}_t\right).$$

The time-varying intercept $\beta_{0,t}$ is assumed to depend on its previous values according to

$$\Delta_d \beta_{0,t} | \beta_{0,t-1}, \ldots, \beta_{0,t-d} \sim \text{N}(0, \kappa_{\beta_0}^{-1}),$$

for $d = 1, 2, 3$, respectively. For $d > 0$ this induces dependence between successive observations. The other parameters, δ for the linear trend, γ_t for a seasonal effect, and the vector $\boldsymbol{\eta}$ for the effect of a covariate vector $\boldsymbol{X}_t$, are also assumed to be normally distributed with certain priors. Inference is done in a Bayesian framework and utilizes an efficient integrated nested Laplace approximation (INLA) provided by the package **INLA** (Lindgren and Rue 2015, available from http://www.R-INLA.org/). In the following example we fit a negative binomial model without trend, seasonality or covariate effects but with a time-varying intercept of order $d = 1$ to the campylobacterosis data:

```
R> library("INLA")
R> campyfit_INLA <- inla(ts ~ f(time, model = "rw1", cyclic = FALSE),
+    data = data.frame(time = seq(along = campy), ts = campy),
+    family = "nbinomial", E = mean(campy),
+    control.predictor = list(compute = TRUE, link = 1),
+    control.compute = list(cpo = FALSE, config = TRUE),
+    control.inla = list(int.strategy = "grid", dz = 1, diff.logdens = 10))
R> posterior <- inla.posterior.sample(1000, campyfit_INLA)
R> rowMeans(sapply(posterior, function(x) (unname(x$hyperpar))))

[1] 317.5  33.3
```

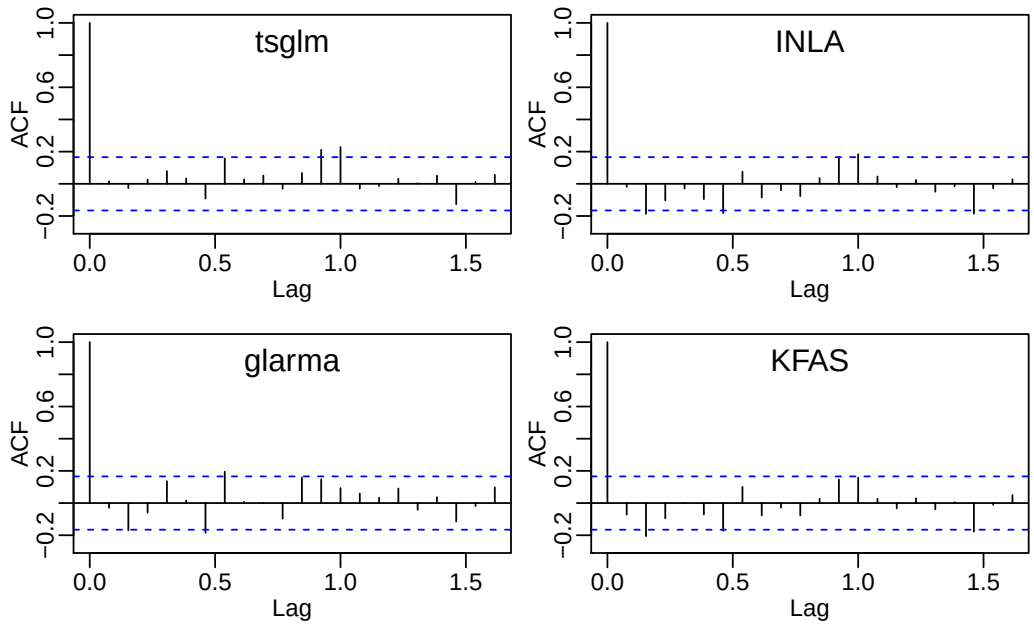

Figure 5: Empirical autocorrelation function of the response residuals for a model fit to the campylobacterosis data by our function `tsglm` (see Section 7.1, but without the intervention effects) and the packages **glarma**, **INLA** and **KFAS** (see Section 8).

The estimates for the parameters ϕ (the former) and κ_{β_0} (the latter) in the output above are based on means of a sample of size 1000 from the posterior distribution. Fitted values $\widehat{\lambda}_1, \ldots, \widehat{\lambda}_{140}$ are obtained in the same way (not shown in the code above). With the Bayesian approach it is very natural to obtain prediction intervals for future observations from the posterior distribution which account for the estimation and observation uncertainties. This is a clear advantage over the classical likelihood-based approach pursued in our package (cf. Section 4). A disadvantage of the Bayesian approach is its much higher computational effort which could be an obstacle for real-time applications and simultaneous analysis of several time series. The above example needs more than eight seconds to complete on a standard office computer (Intel Xeon CPU with 2.83 GHz); this is seven times longer than `tsglm` takes to fit the model. An additional difference between our approach and that taken by INLA is the specification of temporal dependence. The comparison of the final fitted values, shown in Figure 6, illustrates that the model with a time-varying intercept fitted by `inla` gives a much smoother line through the observed values when compared to the model fitted by `tsglm`. For this example, the empirical autocorrelation function of the response residuals in Figure 5 (bottom left) is significantly different from zero at lag one; hence short term temporal correlation is not explained sufficiently by the hierarchical model. It also becomes clear by this plot that we should have included seasonality by employing the term γ_t. The residuals of the GLM-based fit by the function `tsglm` do not exhibit any serial correlation which has not been explained by the model (see Figure 5 (top left)). In general, the GLM-based model is expected to provide more accurate 1-step-ahead predictions whilst the hierarchical model prediction obtained by `inla` is more stable. Either of these two features could be preferable depending upon the specific application. It would be interesting to study these two ways of modeling temporal dependence in a future work.

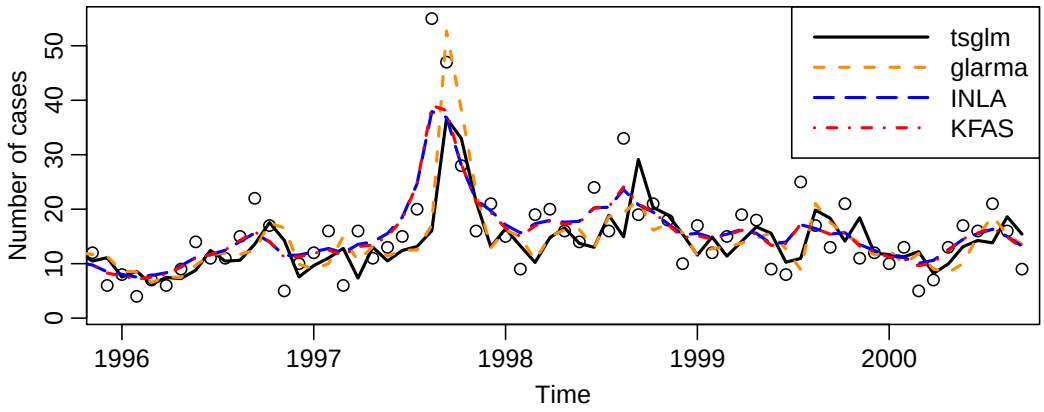

Figure 6: Comparison of a model fit to the campylobacterosis data by our function `tsglm` (see Section 7.1, but without the intervention effects) and the packages **glarma**, **INLA** and **KFAS** (see Section 8).

The package **KFAS** (Helske 2017a) treats state space models for multivariate time series where the distribution of the observations belongs to the exponential family (and also includes the negative binomial distribution). Its name refers to Kalman filtering and smoothing, which are the two key algorithms employed by the package. This package is able to cope with very general state space models at the cost of a rather big effort for its correct specification. However, some auxiliary functions and the use of symbolic model description reduce this effort. In contrast to the package **INLA**, which is also capable of fitting state space models, **KFAS** implements maximum likelihood estimation (for a comparison of these two packages see Helske 2017b). One possible univariate model for the campylobacterosis data could be the state space model $Y_t|\mathcal{F}_{t-1} \sim \text{NegBin}(\lambda_t, \phi)$ where $\lambda_t = \exp(\nu_t)$ and the state equation is

$$\nu_t = \nu_{t-1} + \varepsilon_t.$$

The initialization ν_1 is specified by assuming $\nu_1 \sim \text{N}(\lambda, \sigma_\nu^2)$. The degree of serial dependence is induced by independently distributed innovations ε_t for which it is assumed that $\varepsilon_t \sim \text{N}(0, \sigma_\varepsilon^2)$. This model has unknown parameters $\lambda \in \mathbb{R}$, and $\phi, \sigma_{\nu_1}^2, \sigma_\varepsilon^2 \in [0, \infty)$ and can be fitted as follows:

```
R> library("KFAS")
R> model <- SSModel(campy ~ SSMcustom(Z = 1, T = 1, R = 1, Q = 0,
+    a1 = NA, P1 = NA) - 1, distribution = "negative binomial", u = NA)
R> updatefn <- function(pars, model, ...) {
+    model$a1[1, 1] <- pars[1]
+    model$u[, 1] <- exp(pars[2])
+    model$P1[1, 1] <- exp(pars[3])
+    model$Q[1, 1, 1] <- exp(pars[4])
+    return(model)
+  }
R> campyfit_KFAS <- fitSSM(model = model, inits = c(mean(campy), 0, 0, 0),
+    updatefn = updatefn)
```

```
R> exp(campyfit_KFAS$optim.out$par)

[1] 3.427e+00 9.148e+01 2.775e-16 4.334e-02
```

The output above corresponds to the estimated parameters λ, ϕ, $\sigma^2_{\nu_1}$ and σ^2_ε, respectively. We observe that the estimation procedure is quite sensitive to the starting values given; this fact has been pointed out by Helske (2017b). As shown by the empirical autocorrelation function of the residuals in Figure 5 (bottom right), the fitted model explains the temporal dependence of the data quite adequately. The values fitted by this model (see Figure 6) do not show any delay when compared to the fit obtained by `tsglm`. The algorithm used by `tsglm` yields fitted values by 1-step-ahead forecasts based on previous observations; note that only the model parameters are fitted using all available observations. The algorithm of the **KFAS** package for obtaining fitted values includes future observations which naturally leads to a more accurate fit. However, this methodology does not guarantee better out-of-sample forecasting performance since future observations will not be available in general. Further empirical comparison between **KFAS** and **tscount** is required to compare the accuracy of predictions obtained by both models.

Another state space model which could be used to describe count time series is a partially observed Markov process (POMP). The package **pomp** (King, Nguyen, and Ionides 2016) provides a general and abstract representation of such models. One example by King *et al.* (2016, Sections 4.5 and 4.6) is the so-called Ricker model for describing the size N_t of a population which is assumed to fulfill

$$N_{t+1} = rN_t \exp(-N_t + \varepsilon_t)$$

with innovations $\varepsilon_t \sim \mathrm{N}(0, \sigma^2_\varepsilon)$. The actual observations Y_t are noisy measurements of the population size N_t and it is assumed to hold $Y_t \sim \text{Poisson}(\phi N_t)$, where ϕ is an unknown dispersion parameter. Specification and fitting of this rather simple model with the package **pomp** require more than thirty lines of code. This complexity is an obstacle for using the package in standard situations but might prove beneficial in very special scenarios.

The package **gcmr** provides methodology for Gaussian copula marginal regression (Masarotto and Varin 2012, 2017), a framework which is also capable to model count time series. The marginal distribution of a time series Y_t given a covariate vector $\boldsymbol{X}_t$ can be modeled by a Poisson or negative binomial distribution with mean λ_t using that $g(\lambda_t) = \beta_0 + \boldsymbol{\eta}^\top \boldsymbol{X}_t$. This is similar to model (1) but it does not include the terms for regression on past values of Y_t and λ_t. Furthermore, randomness is introduced through an unobserved error process $\{\varepsilon_t : t \in \mathbb{N}\}$ by assuming that $Y_t = F_t^{-1}(\Phi(\varepsilon_t))$, where F_t is the cumulative distribution function of the Poisson or negative binomial distribution with mean λ_t and Φ is the cumulative distribution function of the standard normal distribution. Hence the actual value of Y_t is the $\Phi(\varepsilon_t)$-quantile of the Poisson or negative binomial distribution with mean λ_t. As pointed out by Masarotto and Varin (2012), copulas with discrete marginals might not be unique (see also Genest and Nešlehová 2007). Temporal dependence of $\{Y_t\}$ is modeled through the error process $\{\varepsilon_t\}$ by assuming an autoregressive moving average (ARMA) model of order p and q. Note that although this model accounts for serial dependence it does not model the conditional distribution of Y_t given the past but only its time-varying marginal distribution. The mean λ_t of this marginal distribution is not influenced by the actual (unobserved) value of the error ε_t. Hence this model is not suitable for accurate 1-step-ahead predictions but rather for quantifying the additional uncertainty induced by the serial dependence. The following

example presents a fit of a negative binomial model with an ARMA error process of order $p = 1$ and $q = 1$:

```
R> library("gcmr")
R> gcmr(ts ~ 1, marginal = negbin.marg(link = "identity"),
+    cormat = arma.cormat(p = 1, q = 1), data = data.frame(ts = campy))

Call:
gcmr(formula = ts ~ 1, data = data.frame(ts = campy),
    marginal = negbin.marg(link = "identity"),
    cormat = arma.cormat(p = 1, q = 1))

Marginal model parameters:
(Intercept)   dispersion
     11.321        0.255

Gaussian copula parameters:
   ar1     ma1
 0.824  -0.271
```

The estimated ARMA parameters of the error process indicate that there is a considerable amount of serial correlation in the data. Some of the observed variability is explained by this serial dependence. As discussed above, the fitted values are based solely on the estimated marginal distribution and are therefore constant over time (equal to the estimated intercept in the above output).

An extension of the zero-inflated models of the package **ZIM**, which was presented in the previous section, are the so-called dynamic zero-inflated models (DZIM) as proposed by Yang, Cavanaugh, and Zamba (2015). These fall within the framework of state space models and introduce serial dependence by an unobserved autoregressive process of order p. Fitting is based on the EM algorithm and MCMC simulation.

The package **tsintermittent** (Kourentzes and Petropoulos 2016) provides methods for so-called intermittent demand time series (see for example Kourentzes 2014). These are time series containing the number of requested items of a particular product (e.g., rarely needed spare parts) which are demanded in a sporadic fashion with periods of zero demand. Reliable forecasts of such time series are important for companies to efficiently plan stocking the respective items. In principle, these are count time series which could be analyzed with the methods in our package. However, it is known that classical forecasting methods perform unsatisfactorily for this kind of data (Kourentzes 2014). More successful approaches, which are included in the **tsintermittent** package, are based on the idea of separately modeling the non-zero demand size and the inter-demand intervals. These methods do not take into account that the observations are integers and do not include covariate effects, as the methods in our package do. Temporal dependence is considered implicitly, by assuming that there are periods where subsequent observations are zero. Our functions consider explicit time dependence. The methods in the **tsintermittent** package are possibly more appropriate for the specific context they are tailored for (which would need further examination), but not suitable for count time series in general. They compete with other types of models for zero excess time series data, for example with DZIM.

9. Outlook

In its current version, the R package **tscount** allows for the analysis of count time series with a quite broad class of models. It will hopefully prove to be useful for a wide range of applications. There is a number of desirable extensions of the package which could be included in future releases. We invite other researchers and developers to contribute to this package.

As an alternative to the negative binomial distribution, one could consider the so-called Quasi-Poisson model. It allows for a conditional variance of $\phi\lambda_t$ (instead of $\lambda_t + \phi\lambda_t^2$, as for the negative binomial distribution), which is linearly and not quadratically increasing in the conditional mean λ_t (for the case of independent data see Ver Hoef and Boveng 2007). A scatterplot of the squared residuals against the fitted values could reveal whether a linear relation between conditional mean and variance is more adequate for a given time series. A generalization of the test for overdispersion in INGARCH$(1, 0)$ processes proposed by Weiß and Schweer (2015) could provide guidance for choosing an appropriate conditional distribution.

The common regression models for count data are often not capable to describe data with an exceptionally large number of observations with the value zero. In the literature so-called zero-inflated and hurdle regression models have become popular for zero excess count data (for an introduction and comparison see Loeys, Moerkerke, De Smet, and Buysse 2012). A first attempt to utilize zero-inflation for INGARCH time series models is made by Zhu (2012).

In some applications the variable of interest is not the number of events but the rate, which expresses the number of events per unit. For example the number of infected people per 10 000 inhabitants, where the population size is a so-called exposure variable which varies over time. For models with a logarithmic link function such a rate could be described by a model where the number of events is the response variable and the logarithm of the exposure variable is a so-called offset. An offset is supported by many standard functions for GLMs and could be part of a future release of our package.

Alternative nonlinear models are for example the threshold model suggested by Woodard *et al.* (2011) or the models studied by Fokianos and Tjøstheim (2012). Fokianos and Neumann (2013) propose a class of goodness-of-fit tests for the specification of the linear predictor, which are based on the smoothed empirical process of Pearson residuals. Christou and Fokianos (2015a) develop suitably adjusted score tests for parameters which are identifiable as well as non-identifiable under the null hypothesis. These tests can be employed to test for linearity of an assumed model.

In practical applications one is often faced with outliers. Elsaied and Fried (2014) and Kitromilidou and Fokianos (2016) develop M-estimators for the linear and the log-linear model, respectively. Fried, Liboschik, Elsaied, Kitromilidou, and Fokianos (2014) compare robust estimators of the (partial) autocorrelation (see also Dürre, Fried, and Liboschik 2015) for time series of counts, which can be useful for identifying the correct model order.

In the long term, related models for binary or categorical time series (Moysiadis and Fokianos 2014) or potential multivariate extensions of count time series following GLMs could be included as well.

The models which are so far included in the package or mentioned above fall into the class of time series following GLMs. There is also quite a lot of literature on thinning-based time series models but we are not aware of any publicly available software implementations. To

name just a few of many publications, Weiß (2008) reviews univariate time series models based on the thinning operation, Pedeli and Karlis (2013) study a multivariate extension and Scotto, Weiß, Silva, and Pereira (2014) consider models for time series with a finite range of counts. For the wide class of state space models there are the R packages **INLA**, **KFAS** and **pomp** available, although it is quite complex to apply these to count time series. A future version of our package could provide simple interfaces to such packages specifically for fitting certain count time series models.

Acknowledgments

Part of this work was done while K. Fokianos was a Gambrinus Fellow at TU Dortmund University. The research of R. Fried and T. Liboschik was supported by the German Research Foundation (DFG, SFB 823 "Statistical modelling of nonlinear dynamic processes"). The authors thank Philipp Probst for his considerable contribution to the development of the package and Jonathan Rathjens for carefully checking the package. The fruitful discussions with Maëlle Salmon (formerly at Robert Koch Institute, Berlin, Germany; currently at Center for Research in Environmental Epidemiology, Barcelona, Spain) improved both the package and this manuscript. The authors also thank the handling editor and two referees for their helpful and constructive comments which have improved the final version of this manuscript.

References

Agosto A, Cavaliere G, Kristensen D, Rahbek A (2015). "Modeling Corporate Defaults: Poisson Autoregressions with Exogenous Covariates (PARX)." *CREATES Research Paper*, School of Economics and Management, University of Aarhus.

Ahmad A, Francq C (2016). "Poisson QMLE of Count Time Series Models." *Journal of Time Series Analysis*, **37**(3), 291–314. doi:10.1111/jtsa.12167.

Benjamin MA, Rigby RA, Stasinopoulos DM (2003). "Generalized Autoregressive Moving Average Models." *Journal of the American Statistical Association*, **98**(461), 214–223. doi:10.1198/016214503388619238.

Bollerslev T (1986). "Generalized Autoregressive Conditional Heteroskedasticity." *Journal of Econometrics*, **31**(3), 307–327. doi:10.1016/0304-4076(86)90063-1.

Box GEP, Tiao GC (1975). "Intervention Analysis with Applications to Economic and Environmental Problems." *Journal of the American Statistical Association*, **70**(349), 70–79. doi:10.2307/2285379.

Christou V, Fokianos K (2014). "Quasi-Likelihood Inference for Negative Binomial Time Series Models." *Journal of Time Series Analysis*, **35**(1), 55–78. doi:10.1111/jtsa.12050.

Christou V, Fokianos K (2015a). "Estimation and Testing Linearity for Non-Linear Mixed Poisson Autoregressions." *Electronic Journal of Statistics*, **9**(1), 1357–1377. doi:10.1214/15-ejs1044.

Christou V, Fokianos K (2015b). "On Count Time Series Prediction." *Journal of Statistical Computation and Simulation*, **85**(2), 357–373. doi:10.1080/00949655.2013.823612.

Cox DR (1981). "Statistical Analysis of Time Series: Some Recent Developments." *Scandinavian Journal of Statistics*, **8**(2), 93–115.

Czado C, Gneiting T, Held L (2009). "Predictive Model Assessment for Count Data." *Biometrics*, **65**(4), 1254–1261. doi:10.1111/j.1541-0420.2009.01191.x.

Dawid AP (1984). "Statistical Theory: The Prequential Approach." *Journal of the Royal Statistical Society A*, **147**(2), 278–292. doi:10.2307/2981683.

Demidenko E (2013). *Mixed Models: Theory and Applications with R*. Wiley Series in Probability and Statistics, 2nd edition. John Wiley & Sons, Hoboken. doi:10.1002/0471728438.

Douc R, Doukhan P, Moulines E (2013). "Ergodicity of Observation-Driven Time Series Models and Consistency of the Maximum Likelihood Estimator." *Stochastic Processes and their Applications*, **123**(7), 2620–2647. doi:10.1016/j.spa.2013.04.010.

Doukhan P, Fokianos K, Tjøstheim D (2012). "On Weak Dependence Conditions for Poisson Autoregressions." *Statistics & Probability Letters*, **82**(5), 942–948. doi:10.1016/j.spl.2012.01.015.

Dunsmuir WTM, Scott DJ (2015). "The **glarma** Package for Observation-Driven Time Series Regression of Counts." *Journal of Statistical Software*, **67**(7), 1–36. doi:10.18637/jss.v067.i07.

Dürre A, Fried R, Liboschik T (2015). "Robust Estimation of (Partial) Autocorrelation." *WIREs Computational Statistics*, **7**(3), 205–222. doi:10.1002/wics.1351.

Efron B, Tibshirani R (1993). *An Introduction to the Bootstrap*. Monographs on Statistics and Applied Probability. Chapman & Hall, New York.

Elsaied H, Fried R (2014). "Robust Fitting of INARCH Models." *Journal of Time Series Analysis*, **35**(6), 517–535. doi:10.1111/jtsa.12079.

Fahrmeir L, Tutz G (2001). *Multivariate Statistical Modelling Based on Generalized Linear Models*. Springer-Verlag, New York. doi:10.1007/978-1-4757-3454-6.

Ferland R, Latour A, Oraichi D (2006). "Integer-Valued GARCH Process." *Journal of Time Series Analysis*, **27**(6), 923–942. doi:10.1111/j.1467-9892.2006.00496.x.

Fokianos K (2011). "Some Recent Progress in Count Time Series." *Statistics*, **45**(1), 49–58. doi:10.1080/02331888.2010.541250.

Fokianos K (2012). "Count Time Series Models." In T Subba Rao, S Subba Rao, CR Rao (eds.), *Time Series – Methods and Applications*, Handbook of Statistics, pp. 315–347. Elsevier, Amsterdam.

Fokianos K (2015). "Statistical Analysis of Count Time Series Models: A GLM Perspective." In R Davis, S Holan, R Lund, N Ravishanker (eds.), *Handbook of Discrete-Valued Time Series*, Handbooks of Modern Statistical Methods, pp. 3–28. Chapman & Hall, London.

Fokianos K, Fried R (2010). "Interventions in INGARCH Processes." *Journal of Time Series Analysis*, **31**(3), 210–225. doi:10.1111/j.1467-9892.2010.00657.x.

Fokianos K, Fried R (2012). "Interventions in Log-Linear Poisson Autoregression." *Statistical Modelling*, **12**(4), 299–322. doi:10.1177/1471082x1201200401.

Fokianos K, Neumann MH (2013). "A Goodness-of-Fit Test for Poisson Count Processes." *Electronic Journal of Statistics*, **7**, 793–819. doi:10.1214/13-ejs790.

Fokianos K, Rahbek A, Tjøstheim D (2009). "Poisson Autoregression." *Journal of the American Statistical Association*, **104**(488), 1430–1439. doi:10.1198/jasa.2009.tm08270.

Fokianos K, Tjøstheim D (2011). "Log-Linear Poisson Autoregression." *Journal of Multivariate Analysis*, **102**(3), 563–578. doi:10.1016/j.jmva.2010.11.002.

Fokianos K, Tjøstheim D (2012). "Nonlinear Poisson Autoregression." *The Annals of the Institute of Statistical Mathematics*, **64**(6), 1205–1225. doi:10.1007/s10463-012-0351-3.

Fried R, Liboschik T, Elsaied H, Kitromilidou S, Fokianos K (2014). "On Outliers and Interventions in Count Time Series Following GLMs." *Austrian Journal of Statistics*, **43**(3), 181–193. doi:10.17713/ajs.v43i3.30.

Fuller WA (1996). *Introduction to Statistical Time Series*. 2nd edition. John Wiley & Sons, New York.

Genest C, Nešlehová J (2007). "A Primer on Copulas for Count Data." *ASTIN Bulletin*, **37**(2), 475–515. doi:10.1017/s0515036100014963.

Gneiting T, Balabdaoui F, Raftery AE (2007). "Probabilistic Forecasts, Calibration and Sharpness." *Journal of the Royal Statistical Society B*, **69**(2), 243–268. doi:10.1111/j.1467-9868.2007.00587.x.

Harvey AC, Durbin J (1986). "The Effects of Seat Belt Legislation on British Road Casualties: A Case Study in Structural Time Series Modelling." *Journal of the Royal Statistical Society A*, **149**(3), 187–227. doi:10.2307/2981553.

Heinen A (2003). "Modelling Time Series Count Data: An Autoregressive Conditional Poisson Model." *CORE Discussion Paper*, **62**. doi:10.2139/ssrn.1117187.

Heisterkamp SH, Dekkers ALM, Heijne JCM (2006). "Automated Detection of Infectious Disease Outbreaks: Hierarchical Time Series Models." *Statistics in Medicine*, **25**(24), 4179–4196. doi:10.1002/sim.2674.

Held L, Höhle M, Hofmann M (2005). "A Statistical Framework for the Analysis of Multivariate Infectious Disease Surveillance Counts." *Statistical Modelling*, **5**(3), 187–199. doi:10.1191/1471082x05st098oa.

Helske J (2017a). "**KFAS**: Exponential Family State Space Models in R." *Journal of Statistical Software*, **78**(10), 1–39. doi:10.18637/jss.v078.i10.

Helske J (2017b). "**KFAS**: Kalman Filter and Smoother for Exponential Family State Space Models." R package version 1.2.8, URL https://CRAN.R-project.org/package=KFAS.

Hilbe JM (2011). *Negative Binomial Regression*. 2nd edition. Cambridge University Press, Cambridge.

Jung R, Tremayne A (2011). "Useful Models for Time Series of Counts or Simply Wrong Ones?" *AStA Advances in Statistical Analysis*, **95**(1), 59–91. doi:10.1007/s10182-010-0139-9.

Kedem B, Fokianos K (2002). *Regression Models for Time Series Analysis*. Wiley Series in Probability and Statistics. John Wiley & Sons, Hoboken. doi:10.1002/0471266981.

King AA, Nguyen D, Ionides EL (2016). "Statistical Inference for Partially Observed Markov Processes via the R Package **pomp**." *Journal of Statistical Software*, **69**(12), 1–43. doi:10.18637/jss.v069.i12.

Kitromilidou S, Fokianos K (2016). "Robust Estimation Methods for a Class of Log-Linear Count Time Series Models." *Journal of Statistical Computation and Simulation*, **86**(4), 740–755. doi:10.1080/00949655.2015.1035271.

Kourentzes N (2014). "On Intermittent Demand Model Optimisation and Selection." *International Journal of Production Economics*, **156**, 180–190. doi:10.1016/j.ijpe.2014.06.007.

Kourentzes N, Petropoulos F (2016). "**tsintermittent**: Intermittent Time Series Forecasting." R package version 1.9, URL https://CRAN.R-project.org/package=tsintermittent.

Lange K (1999). *Numerical Analysis for Statisticians*. Statistics and Computing. Springer-Verlag, New York.

Liboschik T, Fried R, Fokianos K, Probst P (2017). ***tscount**: Analysis of Count Time Series*. R package version 1.4.1, URL https://CRAN.R-project.org/package=tscount.

Liboschik T, Kerschke P, Fokianos K, Fried R (2016). "Modelling Interventions in INGARCH Processes." *International Journal of Computer Mathematics*, **93**(4), 640–657. doi:10.1080/00207160.2014.949250.

Lindgren F, Rue H (2015). "Bayesian Spatial Modelling with R-**INLA**." *Journal of Statistical Software*, **63**(19), 1–25. doi:10.18637/jss.v063.i19.

Loeys T, Moerkerke B, De Smet O, Buysse A (2012). "The Analysis of Zero-Inflated Count Data: Beyond Zero-Inflated Poisson Regression." *British Journal of Mathematical and Statistical Psychology*, **65**(1), 163–180. doi:10.1111/j.2044-8317.2011.02031.x.

Manitz J, Höhle M (2013). "Bayesian Outbreak Detection Algorithm for Monitoring Reported Cases of Campylobacteriosis in Germany." *Biometrical Journal*, **55**(4), 509–526. doi:10.1002/bimj.201200141.

Masarotto G, Varin C (2012). "Gaussian Copula Marginal Regression." *Electronic Journal of Statistics*, **6**, 1517–1549. doi:10.1214/12-ejs721.

Masarotto G, Varin C (2017). "Gaussian Copula Regression in R." *Journal of Statistical Software*, **77**(8), 1–26. doi:10.18637/jss.v077.i08.

Moysiadis T, Fokianos K (2014). "On Binary and Categorical Time Series Models with Feedback." *Journal of Multivariate Analysis*, **131**, 209–228. doi:10.1016/j.jmva.2014.07.004.

Nelder JA, Wedderburn RWM (1972). "Generalized Linear Models." *Journal of the Royal Statistical Society A*, **135**(3), 370–384. doi:10.2307/2344614.

Pan W (2001). "Akaike's Information Criterion in Generalized Estimating Equations." *Biometrics*, **57**(1), 120–125. doi:10.1111/j.0006-341x.2001.00120.x.

Pedeli X, Karlis D (2013). "On Composite Likelihood Estimation of a Multivariate INAR(1) Model." *Journal of Time Series Analysis*, **34**(2), 206–220. doi:10.1111/jtsa.12003.

R Core Team (2017). *R: A Language and Environment for Statistical Computing*. Vienna, Austria. URL https://www.R-project.org.

Rigby RA, Stasinopoulos DM (2005). "Generalized Additive Models for Location, Scale and Shape." *Journal of the Royal Statistical Society C*, **54**(3), 507–554. doi:10.1111/j.1467-9876.2005.00510.x.

Salmon M, Schumacher D, Höhle M (2016). "Monitoring Count Time Series in R: Aberration Detection in Public Health Surveillance." *Journal of Statistical Software*, **70**(10), 1–35. doi:10.18637/jss.v070.i10.

Scotto MG, Weiß CH, Silva ME, Pereira I (2014). "Bivariate Binomial Autoregressive Models." *Journal of Multivariate Analysis*, **125**, 233–251. doi:10.1016/j.jmva.2013.12.014.

Siakoulis V (2015). "**acp**: Autoregressive Conditional Poisson." R package version 2.1, URL https://CRAN.R-project.org/package=acp.

Sim T (2016). *Maximum Likelihood Estimation in Partially Observed Markov Models with Applications to Time Series of Counts*. Ph.D. thesis, Télécom ParisTech, Paris.

Stasinopoulos DM, Rigby RA (2007). "Generalized Additive Models for Location Scale and Shape (GAMLSS) in R." *Journal of Statistical Software*, **23**(7), 1–46. doi:10.18637/jss.v023.i07.

Stasinopoulos DM, Rigby RA, Eilers P (2016). "**gamlss.util**: GAMLSS Utilities." R package version 4.3-4, URL https://CRAN.R-project.org/package=gamlss.util.

Tjøstheim D (2012). "Some Recent Theory for Autoregressive Count Time Series." *TEST*, **21**(3), 413–438. doi:10.1007/s11749-012-0296-0.

Tjøstheim D (2015). "Count Time Series with Observation-Driven Autoregressive Parameter Dynamics." In R Davis, S Holan, R Lund, N Ravishanker (eds.), *Handbook of Discrete-Valued Time Series*, Handbooks of Modern Statistical Methods, pp. 77–100. Chapman & Hall, London.

Venables WN, Ripley BD (2002). *Modern Applied Statistics with S*. Statistics and Computing, 4th edition. Springer-Verlag, New York. doi:10.1007/978-0-387-21706-2.

Ver Hoef JM, Boveng PL (2007). "Quasi-Poisson vs. Negative Binomial Regression: How Should We Model Overdispersed Count Data?" *Ecology*, **88**(11), 2766–2772. doi:10.1890/07-0043.1.

Weiß CH (2008). "Thinning Operations for Modeling Time Series of Counts – A Survey." *AStA Advances in Statistical Analysis*, **92**(3), 319–341. doi:10.1007/s10182-008-0072-3.

Weiß CH, Schweer S (2015). "Detecting Overdispersion in INARCH(1) Processes." *Statistica Neerlandica*, **69**(3), 281–297. doi:10.1111/stan.12059.

Woodard DB, Matteson DS, Henderson SG (2011). "Stationarity of Generalized Autoregressive Moving Average Models." *Electronic Journal of Statistics*, **5**, 800–828. doi:10.1214/11-ejs627.

Yang M, Cavanaugh JE, Zamba GKD (2015). "State-Space Models for Count Time Series with Excess Zeros." *Statistical Modelling*, **15**(1), 70–90. doi:10.1177/1471082x14535530.

Yang M, Zamba GKD, Cavanaugh JE (2013). "Markov Regression Models for Count Time Series with Excess Zeros: A Partial Likelihood Approach." *Statistical Methodology*, **14**, 26–38. doi:10.1016/j.stamet.2013.02.001.

Yang M, Zamba GKD, Cavanaugh JE (2017). "**ZIM**: Zero-Inflated Models for Count Time Series with Excess Zeros." R package version 1.0.3, URL https://CRAN.R-project.org/package=ZIM.

Yee TW (2015). *Vector Generalized Linear and Additive Models: With an Implementation in R*. Springer-Verlag, New York.

Yee TW (2017). "**VGAM**: Vector Generalized Linear and Additive Models." R package version 1.0-3, URL https://CRAN.R-project.org/package=VGAM.

Yee TW, Wild CJ (1996). "Vector Generalized Additive Models." *Journal of the Royal Statistical Society B*, **58**(3), 481–493.

Zeileis A, Kleiber C, Jackman S (2008). "Regression Models for Count Data in R." *Journal of Statistical Software*, **27**(8), 1–25. doi:10.18637/jss.v027.i08.

Zhu F (2012). "Zero-Inflated Poisson and Negative Binomial Integer-Valued GARCH Models." *Journal of Statistical Planning and Inference*, **142**(4), 826–839. doi:10.1016/j.jspi.2011.10.002.

A. Implementation details

A.1. Parameter space for the log-linear model

The parameter space Θ for the log-linear model (3) which guarantees a stationary and ergodic solution of the process is subject of current research. In the current implementation of `tsglm` the parameters need to fulfill the condition

$$\max\left\{|\beta_1|,\ldots,|\beta_p|,|\alpha_1|,\ldots,|\alpha_q|,\left|\sum_{k=1}^{p}\beta_k+\sum_{\ell=1}^{q}\alpha_\ell\right|\right\}<1. \tag{17}$$

At the time we started developing **tscount**, (17) appeared as a reasonable extension of the condition

$$\max\left\{|\beta_1|,|\alpha_1|,|\beta_1+\alpha_1|\right\}<1,$$

which Douc *et al.* (2013, Lemma 14) derive for $p=q=1$. However, in a recent work, Sim (2016, Proposition 5.4.7) derives sufficient conditions for a model of order $p=q$. For the first order model he obtains the weaker condition

$$\max\left\{|\alpha_1|,|\beta_1+\alpha_1|\right\}<1.$$

For $p=q=2$ the required condition is

$$\begin{aligned}\max\{\ & |\alpha_1|+|\alpha_2|+|\beta_2|\,,|\alpha_1\alpha_2|+\left|\alpha_2\alpha_1^2\right|+|\alpha_1+\beta_2|\,,|\alpha_2|+|\alpha_1+\beta_1|+|\beta_2|\,,\\ & |\alpha_2(\alpha_1+\beta_1)|+|\alpha_2+\alpha_1(\alpha_1+\beta_1)|+|(\alpha_1+\beta_1)\beta_2|\,,\\ & |\alpha_1\alpha_2|+|\alpha_2+\alpha_1(\alpha_1+\beta_1)+\beta_2|+|\alpha_1\beta_2|\,,\\ & |(\alpha_1+\beta_1)\alpha_2|+\left|\alpha_2+(\alpha_1+\beta_1)^2+\beta_2\right|+|(\alpha_1+\beta_1)\beta_2|\ \}\ <1.\end{aligned} \tag{18}$$

There are parameters which fulfill (17) but not (18) (e.g., $\beta_1=-0.9$, $\beta_2=0.9$, $\alpha_1=0$, $\alpha_2=0$) and vice versa (e.g., $\beta_1=-1.8$, $\beta_2=0$, $\alpha_1=0.9$, $\alpha_2=0$). However, there exists a large intersection of values which fulfill (17) and (18). For the general case $p=q$ the condition can be obtained by considering the maximum among p elements of the norms of matrix products with p factors, where each factor corresponds to a $(2p-1)\times(2p-1)$ matrix. The implementation of this condition is a challenging problem and therefore we have decided in favor of (17). Alternatively, we can obtain unconstrained estimates (argument `final.control = list(constrained = NULL)`), which should be examined carefully.

A.2. Recursions for inference and their initialization

Let h be the inverse of the link function g and let $h'(x)=\partial h(x)/\partial x$ be its derivative. In the case of the identity link $g(x)=x$ it holds $h(x)=x$ and $h'(x)=1$ and in the case of the logarithmic link $g(x)=\log(x)$ it holds $h(x)=h'(x)=\exp(x)$. The partial derivative of the conditional mean $\lambda_t(\boldsymbol{\theta})$ is given by

$$\frac{\partial\lambda_t(\boldsymbol{\theta})}{\partial\boldsymbol{\theta}}=h'\left(\nu_t(\boldsymbol{\theta})\right)\frac{\partial\nu_t(\boldsymbol{\theta})}{\partial\boldsymbol{\theta}},$$

where the vector of partial derivatives of the linear predictor $\nu_t(\boldsymbol{\theta})$,

$$\frac{\partial \nu_t(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \left(\frac{\partial \nu_t(\boldsymbol{\theta})}{\partial \beta_0}, \frac{\partial \nu_t(\boldsymbol{\theta})}{\partial \beta_1}, \ldots, \frac{\partial \nu_t(\boldsymbol{\theta})}{\partial \beta_p}, \frac{\partial \nu_t(\boldsymbol{\theta})}{\partial \alpha_1}, \ldots, \frac{\partial \nu_t(\boldsymbol{\theta})}{\partial \alpha_q}, \frac{\partial \nu_t(\boldsymbol{\theta})}{\partial \eta_1}, \ldots, \frac{\partial \nu_t(\boldsymbol{\theta})}{\partial \eta_r} \right)^\top,$$

can be computed recursively. The recursions are given by

$$\begin{aligned}
\frac{\partial \nu_t(\boldsymbol{\theta})}{\partial \beta_0} &= 1 + \sum_{\ell=1}^{q} \alpha_\ell \frac{\partial \nu_{t-j_\ell}(\boldsymbol{\theta})}{\partial \beta_0}, \\
\frac{\partial \nu_t(\boldsymbol{\theta})}{\partial \beta_s} &= \widetilde{g}(Y_{t-i_s}) + \sum_{\ell=1}^{q} \alpha_\ell \frac{\partial \nu_{t-j_\ell}(\boldsymbol{\theta})}{\partial \beta_s}, \quad s = 1, \ldots, p, \\
\frac{\partial \nu_t(\boldsymbol{\theta})}{\partial \alpha_s} &= \sum_{\ell=1}^{q} \alpha_\ell \frac{\partial \nu_{t-j_\ell}(\boldsymbol{\theta})}{\partial \alpha_s} + \nu_{t-j_s}(\boldsymbol{\theta}), \quad s = 1, \ldots, q, \\
\frac{\partial \nu_t(\boldsymbol{\theta})}{\partial \eta_s} &= \sum_{\ell=1}^{q} \alpha_\ell \frac{\partial \nu_{t-j_\ell}(\boldsymbol{\theta})}{\partial \eta_s} + X_{t,s}, \quad s = 1, \ldots, r.
\end{aligned}$$

The recursions for the linear predictor $\nu_t = g(\lambda_t)$ and its partial derivatives depend on past values of the linear predictor and its derivatives, which are generally not observable. We implemented three possibilities for initialization of these values. The default and preferable choice is to initialize by the respective marginal expectations, assuming a model without covariate effects, such that the process is stationary (argument `init.method = "marginal"`). For the linear model (2) it holds (Ferland *et al.* 2006)

$$\mathsf{E}(Y_t) = \mathsf{E}(\nu_t) = \frac{\beta_0}{1 - \sum_{k=1}^{p} \beta_k - \sum_{\ell=1}^{q} \alpha_\ell} =: \mu(\boldsymbol{\theta}). \tag{19}$$

For the log-linear model (3) we instead consider the transformed time series $Z_t := \log(Y_t + 1)$, which has approximately the same second order properties as a time series from the linear model (2). It approximately holds $\mathsf{E}(Z_t) \approx \mathsf{E}(\nu_t) \approx \mu(\boldsymbol{\theta})$. Specifically, we initialize past values of ν_t by $\mu(\boldsymbol{\theta})$ and past values of $\partial \nu_t(\boldsymbol{\theta}) / \partial \boldsymbol{\theta}$ by

$$\frac{\partial \mu(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \left(\frac{\partial \mu(\boldsymbol{\theta})}{\partial \beta_0}, \frac{\partial \mu(\boldsymbol{\theta})}{\partial \beta_1}, \ldots, \frac{\partial \mu(\boldsymbol{\theta})}{\partial \beta_p}, \frac{\partial \mu(\boldsymbol{\theta})}{\partial \alpha_1}, \ldots, \frac{\partial \mu(\boldsymbol{\theta})}{\partial \alpha_q}, \frac{\partial \mu(\boldsymbol{\theta})}{\partial \eta_1}, \ldots, \frac{\partial \mu(\boldsymbol{\theta})}{\partial \eta_r} \right)^\top,$$

which is explicitly given by

$$\begin{aligned}
\frac{\partial \mu(\boldsymbol{\theta})}{\partial \beta_0} &= \frac{1}{1 - \sum_{k=1}^{p} \beta_k - \sum_{\ell=1}^{q} \alpha_\ell}, \\
\frac{\partial \mu(\boldsymbol{\theta})}{\partial \beta_k} &= \frac{\partial \mu(\boldsymbol{\theta})}{\partial \alpha_\ell} = \frac{\beta_0}{\left(1 - \sum_{k=1}^{p} \beta_k - \sum_{\ell=1}^{q} \alpha_\ell\right)^2}, \quad k = 1, \ldots, p, \ \ell = 1, \ldots, q, \quad \text{and} \\
\frac{\partial \mu(\boldsymbol{\theta})}{\partial \eta_m} &= 0, \quad m = 1, \ldots, r.
\end{aligned}$$

Another possibility is to initialize ν_t by β_0 and $\partial \nu_t(\boldsymbol{\theta}) / \partial \boldsymbol{\theta}$ by zero. In this case the model corresponds to standard i.i.d. Poisson random variables (argument `init.method = "iid"`). A third possibility would be a data-dependent initialization of ν_t, for example by $\widetilde{g}(y_1)$.

	$\widehat{\beta}_0$	$\widehat{\beta}_1$	$\widehat{\alpha}_1$	$\ell(\widehat{\boldsymbol{\theta}})$
`init.method = "marginal", init.drop = FALSE`	0.500	0.733	0.249	−3024.7
`init.method = "marginal", init.drop = TRUE`	0.567	0.746	0.236	−2568.0
`init.method = "iid", init.drop = FALSE`	0.867	0.757	0.218	−3037.2
`init.method = "iid", init.drop = TRUE`	0.563	0.738	0.246	−2587.8
`init.method = "firstobs", init.drop = FALSE`	0.559	0.739	0.246	−3018.7
`init.method = "firstobs", init.drop = TRUE`	0.559	0.739	0.246	−2578.1

Table 3: Estimated parameters and log-likelihood of a time series of length 1000 simulated from model (2) for different initialization strategies. The true parameters are $\beta_0 = 0.5$, $\beta_1 = 0.77$ and $\alpha_1 = 0.22$. Likelihood values are included for completeness of the presentation. They are not comparable as they are based on a different number of observations.

In this case, the partial derivatives of ν_t are initialized by zero (argument `init.method = "firstobs"`).

The recursions also depend on unavailable past observations of the time series, prior to the sample which is used for the likelihood computation. The package allows to choose between two strategies to cope with that. The default choice is to replace these pre-sample observations by the same initializations as used for the linear predictor ν_t (see above), transformed by the inverse link function h (argument `init.drop = FALSE`). An alternative is to use the first i_p observations for initialization and to compute the log-likelihood on the remaining observations $y_{i_p+1}, \ldots, y_n$ (argument `init.drop = TRUE`). Recall that i_p is the highest order for regression on past observations.

Particularly in the presence of strong serial dependence, the different methods for initialization can affect the estimation substantially even for quite long time series with 1000 observations. We illustrate this by the simulated example presented in Table 3.

A.3. Starting values for optimization

The numerical optimization of the log-likelihood function requires a starting value for the parameter vector $\boldsymbol{\theta}$, which can be obtained by initial estimation based on a simpler model. Different strategies for this (controlled by the argument `start.control`) are discussed in this section. We call this start estimation (and not initial estimation) to avoid confusion with the initialization of the recursions described in the previous section.

The start estimation by the R function `glm` utilizes the fact that a time series following a GLM without feedback (as in Kedem and Fokianos 2002) can be fitted by employing standard software. Neglecting the feedback mechanism, the parameters of the GLM

$$\begin{aligned} &Y_t|\mathcal{F}^*_{t-1} \sim \text{Poi}(\lambda^*_t), \text{ with } \nu^*_t = g(\lambda^*_t) \text{ and} \\ &\nu^*_t = \beta^*_0 + \beta^*_1 \widetilde{g}(Y_{t-i_1}) + \ldots + \beta^*_p \widetilde{g}(Y_{t-i_p}) + \eta^*_1 X_{t,1} + \ldots + \eta^*_r X_{t,r}, \ t = i_p + 1, \ldots, n, \end{aligned}$$

with $\mathcal{F}^*_t$ the history of the joint process $\{Y_t, \boldsymbol{X}_t\}$, are estimated using the R function `glm`. Denote the estimated parameters by $\widehat{\beta}^*_0$, $\widehat{\beta}^*_1, \ldots, \widehat{\beta}^*_p$, $\widehat{\eta}^*_1, \ldots, \widehat{\eta}^*_r$ and set $\widehat{\alpha}^*_1, \ldots, \widehat{\alpha}^*_q$ to zero (argument `start.control$method = "GLM"`).

Fokianos *et al.* (2009) suggest start estimation of $\boldsymbol{\theta}$, for the first order linear model (2) without covariates, by employing its representation as an ARMA$(1,1)$ process with identical second-order properties, see Ferland *et al.* (2006). For arbitrary orders P and Q with $s := \max(P, Q)$

and the general model from Section 2 this representation, after straightforward calculations, is given by

$$(\widetilde{g}(Y_t) - \underbrace{\mu(\boldsymbol{\theta})}_{=:\zeta}) - \sum_{i=1}^{s} \underbrace{(\beta_i + \alpha_i)}_{=:\varphi_i} (\widetilde{g}(Y_{t-i}) - \mu(\boldsymbol{\theta})) = \varepsilon_t + \sum_{i=1}^{q} \underbrace{(-\alpha_i)}_{=:\psi_i} \varepsilon_{t-i}, \tag{20}$$

where $\beta_i := 0$ for $i \notin P$, $\alpha_i := 0$ for $i \notin Q$ and $\{\varepsilon_t\}$ is a white noise process. Recall that $\widetilde{g}$ is defined by $\widetilde{g}(x) = x$ for the linear model and $\widetilde{g}(x) = \log(x + 1)$ for the log-linear model. Given the autoregressive parameters φ_i and the moving average parameters ψ_i of the ARMA representation of $\{Y_t\}$, the parameters of the original process are obtained by $\alpha_i = -\psi_i$ and $\beta_i = \varphi_i + \psi_i$. We get β_0 from $\beta_0 = \zeta(1 - \sum_{k=1}^{p} \beta_k - \sum_{\ell=1}^{q} \alpha_\ell)$ using the formula for the marginal mean of $\{Y_t\}$. With these formulas estimates $\widehat{\beta}_0^*$, $\widehat{\beta}_i^*$ and $\widehat{\alpha}_i^*$ are obtained from the ARMA estimates $\widehat{\zeta}$, $\widehat{\varphi}_i$ and $\widehat{\psi}_i$. Estimation of the ARMA parameters is implemented by conditional least squares (argument `start.control$method = "CSS"`), maximum likelihood assuming normally distributed errors (argument `start.control$method = "ML"`), or, for models up to first order, the method of moments (argument `start.control$method = "MM"`). If covariates are included, a linear regression is fitted to $\widetilde{g}(Y_t)$, whose errors follow an ARMA model like (20). Consequently, the covariate effects do not enter the dynamics of the process, as it is the case in the actual model (1). It would be preferable to fit an ARMAX model, in which covariate effects are included on the right hand side of (20), but this is currently not readily available in R.

We compare both approaches to obtain start estimates. The GLM approach apparently disregards the feedback mechanism, i.e., the dependence on past values of the conditional mean. As opposed to this, the ARMA approach does not treat covariate effects in an appropriate way. From extensive simulations we note that the final estimation results are almost equally good for both approaches.

However, we also found out that in some situations (especially in the presence of certain types of covariates) both approaches occasionally provoke the likelihood optimization algorithms to run into a local optimum. This happens more often for increasing sample size. To overcome this problem we recommend a naive start estimation assuming an i.i.d. model without covariates, which only estimates the intercept and sets all other parameters to zero (argument `start.control$method = "iid"`). This starting value is usually not close to any local optimum of the likelihood function. Hence we expect, possibly, a larger number of steps for the optimization algorithm to converge. This is the default method of start estimation as we do not guarantee a global optimum with the other two methods, in some special cases.

Particularly for the linear model, some of the aforementioned approaches do not yield a starting value $\widehat{\boldsymbol{\theta}}^* = (\widehat{\beta}_0^*, \widehat{\beta}_1^*, \ldots, \widehat{\beta}_p^*, \widehat{\alpha}_1^*, \ldots, \widehat{\alpha}_q^*, \widehat{\eta}_1^*, \ldots, \widehat{\eta}_r^*)^\top$ for $\boldsymbol{\theta}$ which lays in the interior of the parameter space Θ. To overcome this problem, $\widehat{\boldsymbol{\theta}}^*$ is suitably transformed to be used as a starting value. For the linear model (2) this transformation is done according to the following procedure (Liboschik *et al.* 2016):

1a. Set $\widehat{\beta}_k^* := \min\{\widehat{\beta}_k^*, \varepsilon\}$ and $\widehat{\alpha}_\ell^* := \min\{\widehat{\alpha}_\ell^*, \varepsilon\}$.

1b. If $c := \sum_{k=1}^{p} \widehat{\beta}_k^* + \sum_{\ell=1}^{q} \widehat{\alpha}_\ell^* > 1 - \xi - \varepsilon$, then shrink each $\widehat{\beta}_k^*$ and $\widehat{\alpha}_\ell^*$ by multiplication with the factor $(1 - \xi - \varepsilon)/c$.

2a. Set $\widehat{\beta}_0^* := \widehat{\beta}_0^* \cdot \left(1 - \sum_{k=1}^{p} \widehat{\beta}_k^* + \sum_{\ell=1}^{q} \widehat{\alpha}_\ell^*\right)/c$.

2b. Set $\widehat{\beta}_0^* := \max\{\widehat{\beta}_0^*, \xi + \varepsilon\}$.

3. Set $\widehat{\eta}_m^* := \max\{\widehat{\eta}_m^*, \varepsilon\}$.

A small constant $\varepsilon > 0$ ensures that the initial value lies inside the parameter space Θ and not on its boundaries. It is chosen to be $\varepsilon = 10^{-6}$ by default (argument `epsilon`). Another small constant $\xi > 0$ enforces the inequalities to be strict (i.e., $<$ instead of $\leq$). This constant is set to $\xi = 10^{-6}$ by default (argument `slackvar`); recall Section 3. The shrinkage factor in Step 1b is chosen such that the sum of the parameters equals $1-\xi-\varepsilon$ after possible shrinkage in this step. The choice of $\widehat{\beta}_0^*$ in Step 2a ensures that the marginal mean remains unchanged after possible shrinkage in Step 1b. For the log-linear model (3) it is not necessary to ensure positivity of the parameters. A valid starting value $\widehat{\boldsymbol{\theta}}^*$ is transformed with the following procedure:

1a. Set $\widehat{\beta}_k^* := \text{sign}(\widehat{\beta}_k^*) \cdot \min\{|\widehat{\beta}_k^*|, \varepsilon\}$ and $\widehat{\alpha}_\ell^* := \text{sign}(\widehat{\alpha}_\ell^*) \cdot \min\{|\widehat{\alpha}_\ell^*|, \varepsilon\}$.

1b. If $c := \left|\sum_{k=1}^p \widehat{\beta}_k^* + \sum_{\ell=1}^q \widehat{\alpha}_\ell^*\right| > 1 - \xi - \varepsilon$, then shrink each $\widehat{\beta}_k^*$ and $\widehat{\alpha}_\ell^*$ by multiplication with the factor $(1 - \xi - \varepsilon)/c$.

A.4. Stable inversion of the information matrix

In order to obtain standard errors from the normal approximation (11) one needs to invert the information matrix $G_n(\widehat{\boldsymbol{\theta}}; \widehat{\sigma}^2)$. To avoid numerical instabilities we make use of the fact that an information matrix is a real symmetric and positive definite matrix. We first compute a Choleski factorization of the information matrix. Then we apply an efficient algorithm to invert the matrix employing the upper triangular factor of the Choleski decomposition (see R functions `chol` and `chol2inv`). This procedure is implemented in the function `invertinfo` in our package.

B. Simulations

In this section we present simulations where the results indicate that the methods that have not yet been treated thoroughly in the literature work reliably.

B.1. Covariates

We present some limited simulation results for the problem of including covariates, in both linear and log-linear models. For simplicity we employ first order models with one covariate and a conditional Poisson distribution, that is, we consider the linear model with the identity link function

$$Y_t|\mathcal{F}_{t-1} \sim \text{Poisson}(\lambda_t), \quad \lambda_t = \beta_0 + \beta_1 Y_{t-1} + \alpha_1 \lambda_{t-1} + \eta_1 X_t,$$

$t = 1, \ldots, n$, and the log-linear model with the logarithmic link function

$$Y_t|\mathcal{F}_{t-1} \sim \text{Poisson}(\lambda_t), \quad \log(\lambda_t) = \beta_0 + \beta_1 \log(Y_{t-1} + 1) + \alpha_1 \log(\lambda_{t-1}) + \eta_1 X_t,$$

Abbreviation	Definition
Linear	t/n
Sine	$(\sin(2\pi \cdot 5 \cdot t/n) + 1)/2$
Spiky outlier	$\mathbb{1}(t = \tau)$
Transient shift	$0.8^{t-\tau}\mathbb{1}(t \geq \tau)$
Level shift	$\mathbb{1}(t \geq \tau)$
GARCH(1, 1)	$\sqrt{h_t}\varepsilon_t$ with $\varepsilon_t \sim \mathrm{N}(0.5, 1)$ and $h_t = 0.002 + 0.1X_{t-1}^2 + 0.8h_{t-1}$
Exponential	i.i.d. exponential distributed with mean 0.5
Normal	i.i.d. normally distributed with mean 0.5 and variance 0.04

Table 4: Covariates $\{X_t : t = 1, \ldots, n\}$ considered in the simulation study. The interventions occur at time $\tau = n/2$. The GARCH model is defined recursively (see Bollerslev 1986).

$t = 1, \ldots, n$. The parameters are chosen to be $\beta_1 = 0.3$ and $\alpha_1 = 0.2$. The intercept parameter is $\beta_0 = 4 \cdot 0.5$ for the linear and $\beta_0 = \log(4) \cdot 0.5$ for the log-linear model in order to obtain a marginal mean (without the covariate effect) of about 4 in both cases. We consider the covariates listed in Table 4, covering a simple linear trend, seasonality, intervention effects, i.i.d. observations from different distributions and a stochastic process. The covariates are chosen to be nonnegative, which is necessary for the linear model but not for the log-linear model. All covariates have a mean of about 0.5, such that their effect sizes are somewhat comparable. The regression coefficient is chosen to be $\eta_1 = 2 \cdot \beta_0$ for the linear and $\eta_1 = 1.5 \cdot \beta_0$ for the log-linear model.

Apparently, certain types of covariates can to some extent be confused with serial dependence. This is the case for the linear trend and the level shift, but also for the sinusoidal term, since these lead to data patterns which resemble positive serial correlation; see Figure 7.

A second finding is that the effect of covariates, like a transient shift or a spiky outlier, is hard to be estimated precisely. Note that both covariates have most of their values different from zero only at very few time points (especially the spiky outlier) which explains this behavior of the estimation procedure. The estimators for the coefficients of such covariates have a large variance which decreases only very slowly with growing sample size; see the bottom right plots in Figures 8 and 9 for the linear and the log-linear model, respectively. This does not affect the estimation of the other parameters, see the other three plots in the same figures. For all other types of covariates the variance of the estimator for the regression parameter decreases with growing sample size, which indicates consistency of the estimator.

The conjectured approximate normality of the model parameters stated in (11) seems to hold for most of the covariates considered here even in case of a rather moderate sample size of 100, as indicated by the QQ plots shown in Figure 10. The only serious deviation from normality happens for the spiky outlier in the linear model, a case where many estimates of the covariate coefficient η_1 lie close to zero. This value is the lower boundary of the parameter space for this model. Due to the consistency problem for this covariate (discussed in the previous paragraph) the observed deviation from normality is still present even for a much larger sample size of 2000 (not shown here). Note that for the spiky outlier the conditions for asymptotic normality in linear regression models stated in Section 3 are not fulfilled. QQ plots for the other model parameters β_0, β_1 and α_1 look satisfactory for all types of covariates and are not shown here.

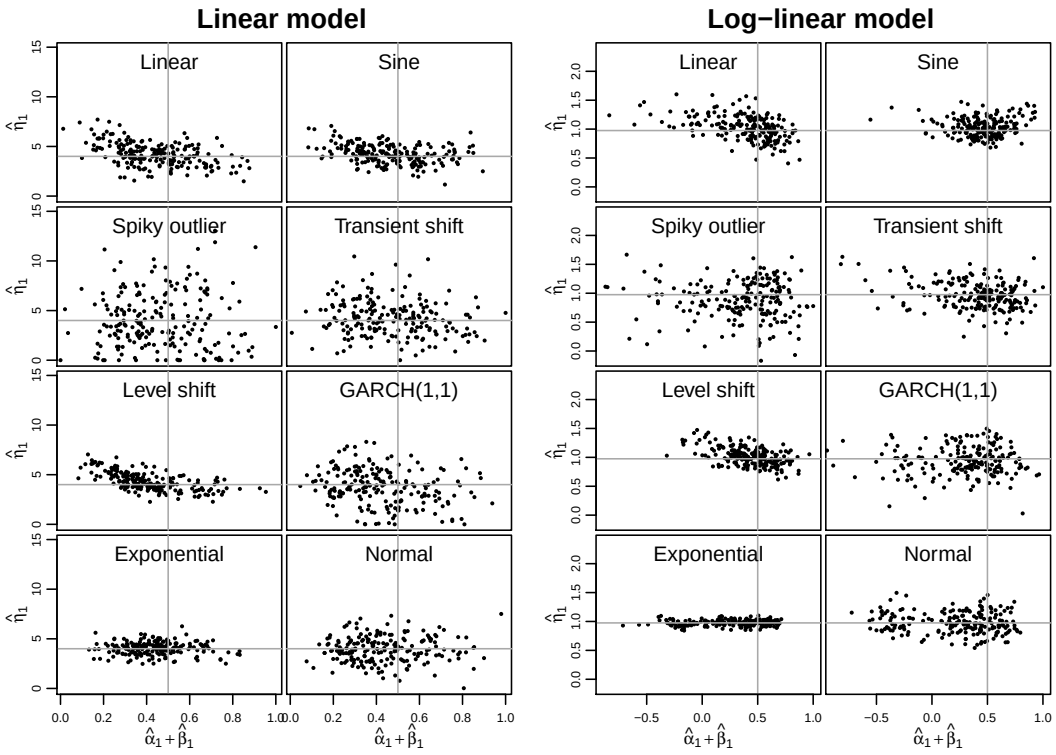

Figure 7: Scatterplots of the estimated covariate parameter $\widehat{\eta}_1$ against the sum $\widehat{\beta}_1+\widehat{\alpha}_1$ of the estimated dependence parameters in a linear (left) respectively log-linear (right) model with an additional covariate of the given type. The time series of length $n = 100$ are simulated from the respective model with the true values marked by gray lines. Each dot represents one of 200 replications.

B.2. Negative binomial distribution

As mentioned before, the model with the logarithmic link function is not covered by the theory derived by Christou and Fokianos (2014). Consequently, we confirm by simulations that estimating the additional dispersion parameter ϕ of the negative binomial distribution by Equation 10 yields good results. We consider both the linear model with the identity link

$$Y_t|\mathcal{F}_{t-1} \sim \text{NegBin}(\lambda_t, \phi), \quad \lambda_t = \beta_0 + \beta_1 Y_{t-1} + \alpha_1 \lambda_{t-1},$$

$t = 1, \ldots, n$, and the log-linear model with the logarithmic link

$$Y_t|\mathcal{F}_{t-1} \sim \text{NegBin}(\lambda_t, \phi), \quad \log(\lambda_t) = \beta_0 + \beta_1 \log(Y_{t-1} + 1) + \alpha_1 \log(\lambda_{t-1}),$$

$t = 1, \ldots, n$. The parameters β_0, β_1 and α_1 are chosen like in Appendix B.1. For the dispersion parameter ϕ we employ the values 1, 5, 10, 20 and ∞, which are corresponding to overdispersion coefficients σ^2 of 1, 0.2, 0.1, 0.05 and 0, respectively.

The estimator of the dispersion parameter ϕ has a positively skewed distribution. It is thus preferable to consider the distribution of its inverse $\widehat{\sigma}^2 = 1/\widehat{\phi}$, which is only slightly negatively skewed; see Table 5. In certain cases it is numerically not possible to solve (10) and the estimation fails. This happens when the true value of ϕ is large and we are close to the

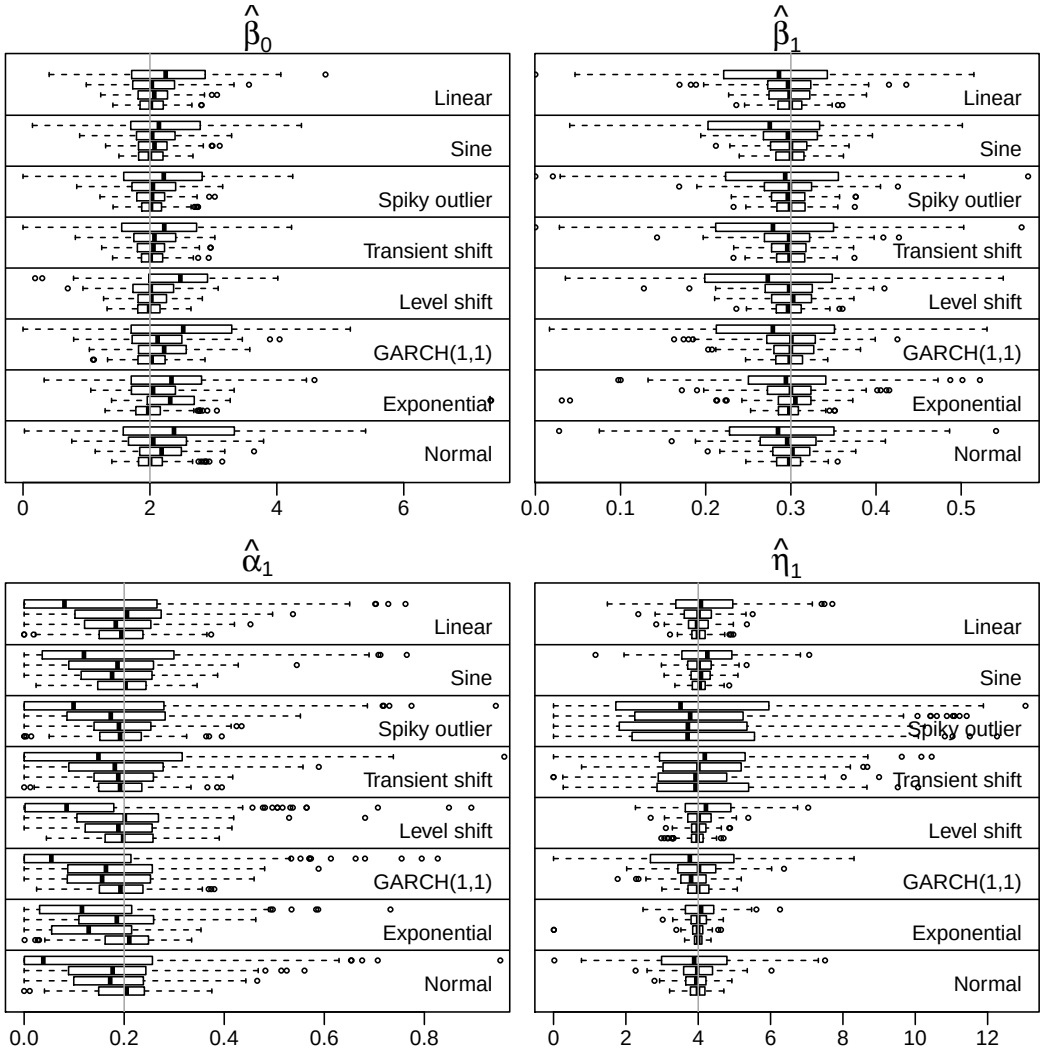

Figure 8: Estimated coefficients for a linear model of order $p = q = 1$ with an additional covariate of the given type. The time series of length $n = 100, 500, 1000, 2000$ (from top to bottom in each panel) are simulated from the respective model with the true coefficients marked by a gray vertical line. Each boxplot is based on 200 replications.

σ^2	Mean	Median	Std.dev.	MAD	Failures (in %)
1.00	0.99	0.97	0.18	0.17	0.00
0.20	0.20	0.19	0.05	0.05	0.00
0.10	0.10	0.10	0.04	0.03	0.00
0.05	0.05	0.05	0.03	0.03	2.80
0.00	0.02	0.02	0.02	0.01	51.70

Table 5: Summary statistics for the estimated overdispersion coefficient $\widehat{\sigma}^2$ of the negative binomial distribution. The time series are simulated from a log-linear model with the true overdispersion coefficient given in the rows. Each statistic is based on 200 replications.

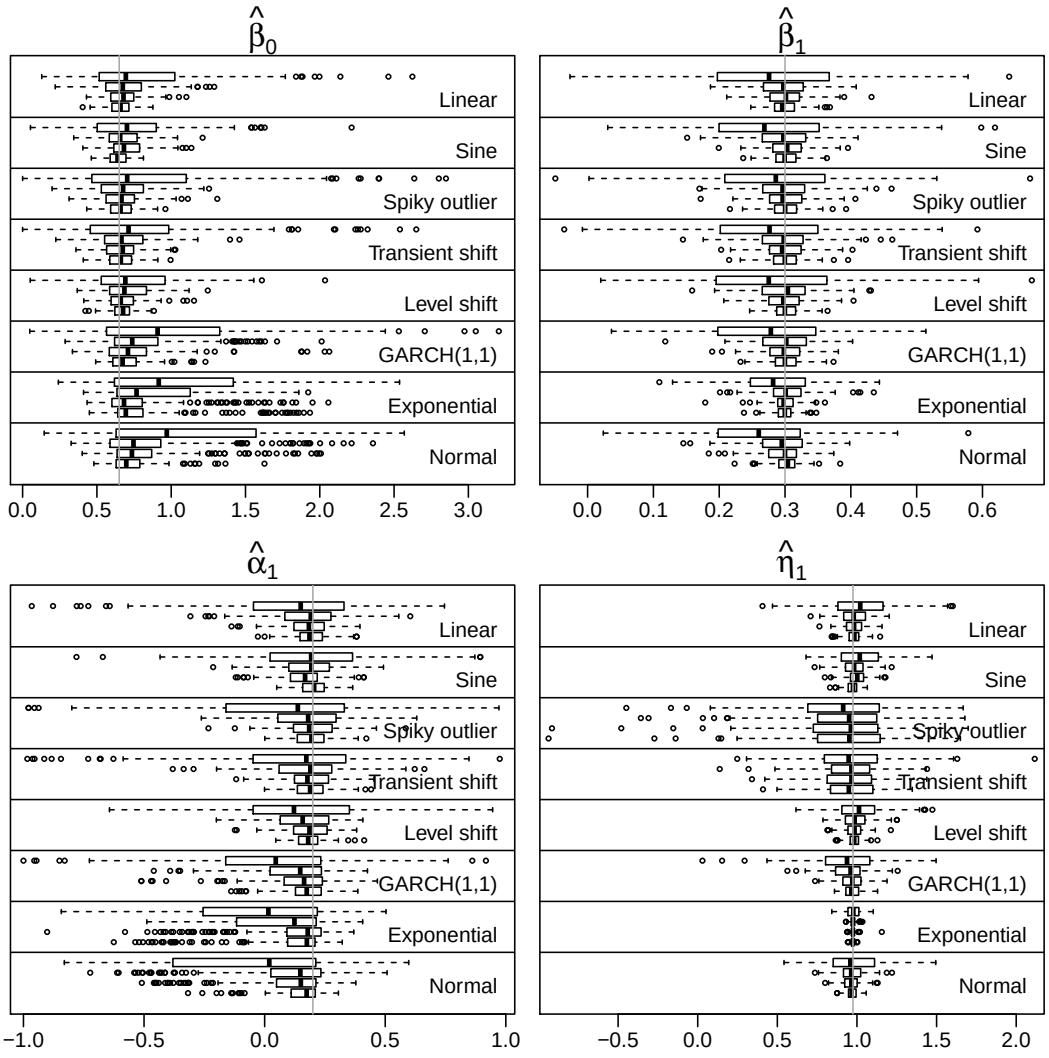

Figure 9: Identical simulation results as those shown in Figure 8 but for the log-linear model.

limiting case of a Poisson distribution (see the proportion of failures in the last column of the table). In such a case our fitting function gives an error and recommends fitting a model with a Poisson distribution instead. These results are very similar for the linear model and thus not shown here.

We check the consistency of the estimator by a simulation for a true value of $\sigma^2 = 1/\phi = 1$. Our results shown in Figure 11 indicate that on average the deviation of the estimation from the true value decreases with increasing sample size for both, the linear and the log-linear model. The boxplots also confirm our above finding that the estimator has a clearly asymmetric distribution for sample sizes up to several hundred.

B.3. Quasi information criterion

We confirm by simulation that the quasi information criterion (QIC) approximates Akaike's information criterion (AIC) in case of a Poisson distribution. Like in Appendix B.2, we

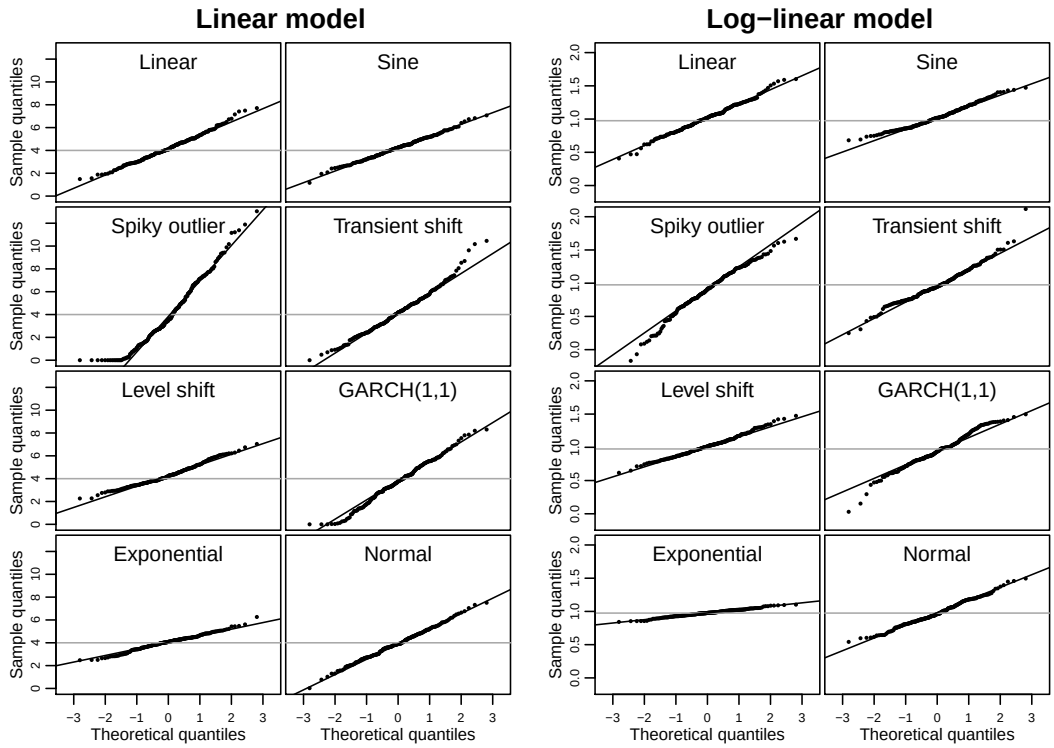

Figure 10: Normal QQ plots for the estimated covariate coefficient $\widehat{\eta}_1$ in a linear (left) respectively log-linear (right) model of order $p = q = 1$ with an additional covariate of the given type. The time series of length $n = 100$ are simulated from the respective model with the true coefficient marked by a gray horizontal line. Each plot is based on 200 replications.

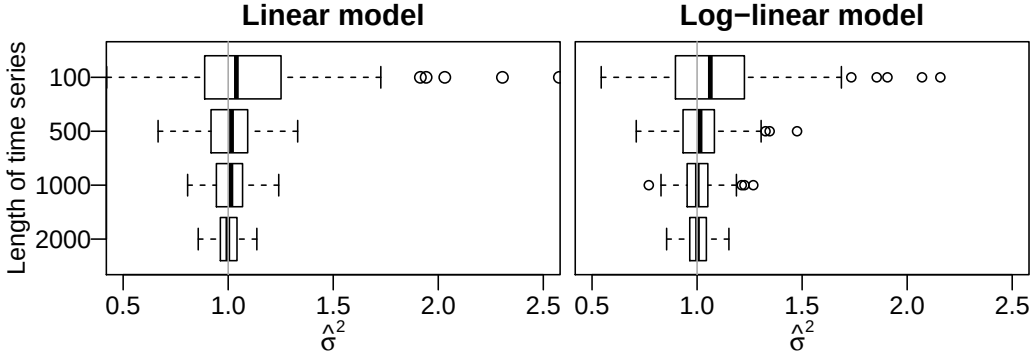

Figure 11: Estimated overdispersion coefficient $\widehat{\sigma}^2$ of the negative binomial distribution for a linear (left) respectively log-linear (right) model of order $p = q = 1$. The time series are simulated from the respective model with the true overdispersion coefficient marked by a gray vertical line. Each boxplot is based on 200 replications.

consider both the linear model with the identity link

$$Y_t|\mathcal{F}_{t-1} \sim \text{Poisson}(\lambda_t), \quad \lambda_t = \beta_0 + \beta_1 Y_{t-1} + \alpha_1 \lambda_{t-1}, \quad t = 1, \ldots, n,$$

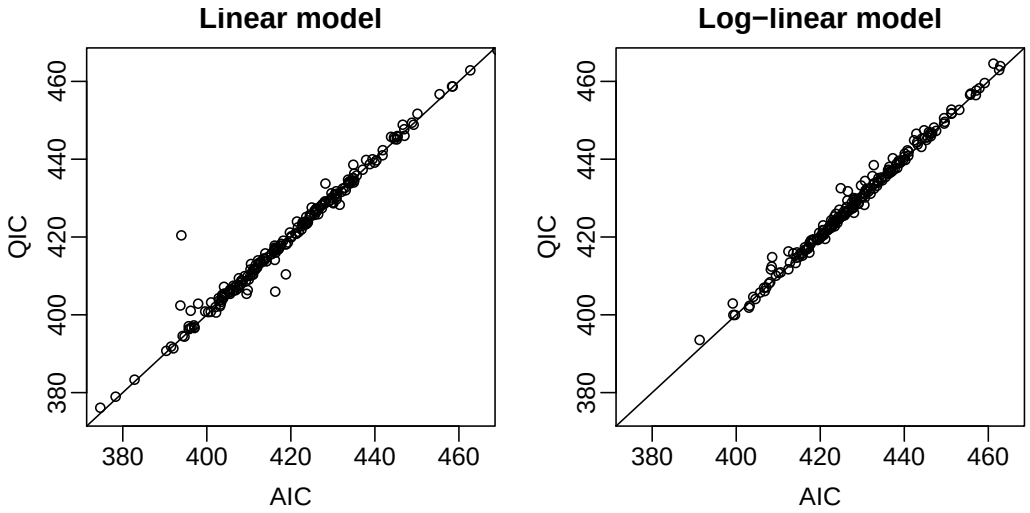

Figure 12: Relationship of QIC and AIC for a linear (left) respectively log-linear (right) model of order $p = q = 1$. Each of the 200 points represents the QIC and AIC of a fit to a time series of length $n = 100$ simulated from the respective model. The diagonal line is the identity, i.e., it represents values for which the QIC equals the AIC.

and the log-linear model with the logarithmic link

$$Y_t|\mathcal{F}_{t-1} \sim \text{Poisson}(\lambda_t), \quad \log(\lambda_t) = \beta_0 + \beta_1 \log(Y_{t-1} + 1) + \alpha_1 \log(\lambda_{t-1}), \quad t = 1, \ldots, n,$$

but now with a Poisson distribution. Again, the parameters β_0, β_1 and α_1 are chosen like in Appendix B.1.

From each of the two models we simulate 200 time series of length $n = 100$ and compute the QIC and AIC of the fitted model. Figure 12 shows that the relationship between QIC and AIC is very close to the identity, i.e., the QIC is approximately equal to the AIC. There is only one out of 200 cases (for the linear model) where the QIC deviates largely from the AIC.

Affiliation:

Tobias Liboschik, Roland Fried *(corresponding author)*
Department of Statistics
TU Dortmund University
44221 Dortmund, Germany
E-mail: liboschik@statistik.tu-dortmund.de, fried@statistik.tu-dortmund.de
URL: http://www.statistik.tu-dortmund.de/liboschik-en.html
http://www.statistik.tu-dortmund.de/fried-en.html

Konstantinos Fokianos
Department of Mathematics & Statistics
University of Cyprus
Nicosia 1678, Cyprus
E-mail: fokianos@ucy.ac.cy
URL: http://www.mas.ucy.ac.cy/~fokianos

Journal of Statistical Software http://www.jstatsoft.org/
published by the Foundation for Open Access Statistics http://www.foastat.org/
November 2017, Volume 82, Issue 5 *Submitted:* 2015-02-19
doi:10.18637/jss.v082.i05 *Accepted:* 2016-08-18