UNIVERSIDAD COMPLUTENSE DE MADRID

FACULTAD DE CIENCIAS MATEMÁTICAS

TESIS DOCTORAL

Large scale Bayesian dynamic forecasting for count time series

Predicción dinámica bayesiana a gran escala para series temporales de conteo

MEMORIA PARA OPTAR AL GRADO DE DOCTOR
PRESENTADA POR

Bruno Flores Barrio

Director

David Ríos Insua

Madrid, 2022

Programa de doctorado en Ingeniería Matemática, Estadística e Investigación Operativa por la Universidad Complutense de Madrid y la Universidad Politécnica de Madrid

FACULTAD DE CIENCIAS MATEMÁTICAS (UCM)

Large scale Bayesian dynamic forecasting for count time series

Predicción dinámica bayesiana a gran escala para series temporales de conteo

Tesis Doctoral

Bruno Flores Barrio

Director

David Ríos Insua

Año 2022

UNIVERSIDAD
COMPLUTENSE
MADRID

DECLARACIÓN DE AUTORÍA Y ORIGINALIDAD DE LA TESIS PRESENTADA PARA OBTENER EL TÍTULO DE DOCTOR

D./Dña. Bruno Flores Barrio,
estudiante en el Programa de Doctorado Ingeniería Matemática, Estadística e Investigación Operativa,
de la Facultad de Ciencias Matemáticas de la Universidad Complutense de Madrid, como autor/a de la tesis presentada para la obtención del título de Doctor y titulada:

Predicción dinámica bayesiana a gran escala para series temporales de conteo (Large scale Bayesian dynamic forecasting for count time series)

y dirigida por: David Ríos Insua

DECLARO QUE:

La tesis es una obra original que no infringe los derechos de propiedad intelectual ni los derechos de propiedad industrial u otros, de acuerdo con el ordenamiento jurídico vigente, en particular, la Ley de Propiedad Intelectual (R.D. legislativo 1/1996, de 12 de abril, por el que se aprueba el texto refundido de la Ley de Propiedad Intelectual, modificado por la Ley 2/2019, de 1 de marzo, regularizando, aclarando y armonizando las disposiciones legales vigentes sobre la materia), en particular, las disposiciones referidas al derecho de cita.

Del mismo modo, asumo frente a la Universidad cualquier responsabilidad que pudiera derivarse de la autoría o falta de originalidad del contenido de la tesis presentada de conformidad con el ordenamiento jurídico vigente.

En Madrid, a 7 de febrero de 2022

Bruno Flores

Fdo.: ______________________________

Esta DECLARACIÓN DE AUTORÍA Y ORIGINALIDAD debe ser insertada en
la primera página de la tesis presentada para la obtención del título de Doctor.

Agradecimientos

En primer lugar, me gustaría agradecer a mi director David Ríos Insua su paciencia, el tiempo dedicado aun estando muy ocupado, y en general todo el apoyo que me ha dado durante la realización de esta tesis. Su capacidad de esfuerzo y trabajo constante es una fuente de inspiración.

Asimismo, quiero dar las gracias a Javier Martín Hernández y Javier Martín Rodrigo sin los cuales tampoco hubiera sido posible la realización de esta tesis: ha sido una suerte poder realizar el doctorado industrial con ellos y han contribuido enormemente a mi formación. De igual modo, ha sido muy importante el soporte del resto del grupo SPOR-DataLab del ICMAT y de mis compañeros de Aeroengy, a los que quiero dar las gracias.

También, me gustaría dar las gracias a mis padres, David y Ana, por todo el apoyo mostrado a lo largo de mi vida y su esfuerzo por facilitarme todos los recursos a su alcance; a mi hermana Clara, que siempre ha sido un ejemplo a seguir y me ha proporcionado valiosos consejos; y a mi sobrina Alba, que es capaz de alegrar el peor de los días con una simple sonrisa.

Finalmente, quiero agradecer a la Comunidad de Madrid la financiación de mi investigación a través de la beca de doctorado industrial IND2018/TIC-9901 FEDER, UE; y reconocer también el apoyo brindado por el Ministerio de Ciencia mediante el proyecto MTM2017-86875-C3-1-R AEI/FEDER UE y el AXA Research Fund a través de la AXA-ICMAT Chair.

Contents

List of Figures

List of Tables

List of Algorithms

List of Code

Notation

General comments

This industrial PhD thesis is devoted to model time series of non-negative integers (counts). Throughout the dissertation we denote a univariate (sequence of random variables) or multivariate (sequence of random vectors) time series with $\{y_t : t = 1, 2, ...\}$, $\{y_t\}_{t\geq 1}$, or just y_t for short. Binary time series (observations taking only two possible values) are usually denoted by z_t.

We also denote by D_t all available information up to the end of time period t. It will typically consist of: the new observation y_t; all additional information that becomes available at time t, $\mathcal{I}_t$ (e.g. information about a new promotion); and the previous information D_{t-1}. Thus, we have the recursive definition $D_t = \{D_{t-1}, y_t, \mathcal{I}_t\}$.

Probability distributions (p.d.f. or p.m.f.) will generally be denoted indistinctly by $\pi(y_t)$, $p(y_t)$ and $Pr(y_t)$. The use of one or the other depends only of the names of the variables in the context, and which one leads to less confusion.

Finally, we denote matrices and vectors with bold letters, e.g. $\boldsymbol{G}_t$, while scalars are left without bold emphasis (light), e.g. λ_t.

Abbreviations

ACP: Autoregressive Conditional Poisson

AESA: Agencia Estatal de Seguridad Aérea

AS: Aviation Safety

ARIMA: AutoRegressive Integrated Moving Average

ARMA: AutoRegressive Moving Average

DCMM: Dynamic Count Mixture Model

DGLM: Dynamic Generalized Linear Model

DLM: Dynamic Linear Model

EM: Expectation–Maximization

FP: Functional Programming

GARCH: Generalized AutoRegressive Conditional Heteroskedastic

GLARMA: Generalized Linear AutoRegressive Moving Average

HS: Hurdle Shifted

IATA: International Air Transport Association

ICAO: International Civil Aviation Organization

INAR: INteger-valued AutoRegressive

INGARCH: INteger-valued GARCH

MAE: Mean Absolute Error

MAPE: Mean Absolute Percentage Error

MCMC: Markov Chain Montecarlo

MLE: Maximum Likelihood Estimation

MSE: Mean Squared Error

NB: Negative Binomial

OOP: Object-Oriented Programming

OoS: Out-of-Stock

SKU: Stock Keeping Unit

SVD: Singular Value Decomposition

ZAPE: Zero Adjusted Percentage Error

ZI: Zero Inflated

Abstract

Dealing with uncertainty has been, and continues to be, an important problem to be taken into account in day-to-day activities of companies and governments. The uncertainty about some future values, whether it is the price of energy, the evolution of an epidemic, the intensity of rainfall, etc., poses difficulties for making adequate decisions. Therefore, the development of accurate forecasting models is of great importance.

On many occasions, the uncertainty is about future observations that take non-negative integer (counts) values. For the treatment of the corresponding count time series, although the use of traditional models is possible, dedicated models that assume non-negative integer observations present numerous advantages, e.g. point forecasts that are easier to interpret and prediction intervals that will not include unfeasible values. The purpose of this industrial PhD thesis, is to contribute to the state of the art in the context of time series modeling with count data. In order to achieve this, three main objectives are covered:

- Development of models for general count time series.
- Development of models for time series with large numbers of zeros and overdispersion.
- Development and implementation of efficient and scalable algorithms to apply the above models to large amounts of data.

The contributions to the field of general count series, commonly found in relatively aggregated data, are various non-homogeneous Poisson models and

their corresponding algorithms based on Bayesian analysis. These models are capable of incorporating multiple combinations of effects frequently found in time series arising in diverse applications. We illustrate the adequacy of these contributions with a real example from aviation safety risk management in which the developed models offer better performance than other well established models for time series of this type.

We also contribute to the modeling of time series with large numbers of zeros and possible overdispersion (second objective), with the development of univariate and multivariate models. These novel models are based on mixtures of Bayesian state-space models, and we show that they outperform other models in a real-world demand forecasting problem in retail inventory management.

Lastly, the fulfillment of the third objective has been achieved through the development of a library to model and obtain predictions of count time series. This library is designed to offer the greatest flexibility possible, in addition to implementing the models introduced in this thesis. It also allows the creation and use of new ones. Furthermore, it provides useful tools for exploratory analysis and evaluation of forecasting performance. The code of this library is not open-source as it is a commercial product developed within the Industrial PhD and owned by the collaborating company.

From the contents of this PhD thesis the following two papers, under diverse publishing states, have been elaborated:

> FLORES, B., RIOS INSUA, D., ALFARO, C. & GOMEZ, J. (2022). Forecasting Aviation Safety Occurrences. To appear in *Applied Stochastic Models in Business and Industry*, `https://doi.org/10.1002/asmb.2675`. **2020 ASA-TSIG Student Paper award**.
>
> FLORES, B. & RIOS INSUA, D. (2022). Demand Count Time Series Forecasting in Retail. *Submitted.*

Besides, a number of talks have also been derived from its contents:

> Flores, B. & Martin Hernandez, J. (May, 2019). Large scale dynamic forecasting for distributed inventory management. Poster presentation at the *2nd Bringing Young Mathematicians Together (BYMAT) Conference*, ICMAT, Madrid, Spain.
>
> Flores, B. & Martin Hernandez, J. (June, 2019). Large scale dynamic forecasting for distributed inventory management. Poster presentation at the *11th Bayesian Inference in Stochastic Processes (BISP) Conference*, Spanish Royal Academy of Sciences, Madrid, Spain.
>
> Flores, B., Rios Insua, D., Alfaro, C. & Gomez, J. (March, 2020). Forecasting aviation safety occurrences. Poster presentation at the *Games, Decisions, Risk and Reliability (GDRR) Transportation Workshop*, SAMSI, Durham, United States.
>
> Flores, B. (August, 2020). Forecasting aviation safety occurrences. Oral presentation at the *JSM 2020 TSIG(ASA) Meeting*, Virtual.
>
> Flores, B. (December, 2020). Modelos para predecir series temporales de conteo con un ejemplo de aplicación. Oral presentation at the *1st IMEIO-DecData: Decisión Optimización y Ciencia de Datos Workshop*, UCM, Madrid, Spain.
>
> Flores, B. (May, 2021). Forecasting count series in retail. Oral presentation at the *12th Bayesian Inference in Stochastic Processes (BISP) Conference*, Virtual.
>
> Flores, B. (September, 2021). Forecasting count time series in retail. Oral presentation at the *21st annual European Network for Business and Industrial Statistics (ENBIS) Conference*, Virtual.
>
> Flores, B. (June, 2022). Bayesian predictive forecasting for retailing. Oral presentation at the *2022 Sociedad de Estadística e Investigación Operativa (SEIO) Congress*, University of Granada, Granada, Spain.

Resumen

El tratamiento de la incertidumbre ha sido, y continúa siendo, un importante problema a tener en cuenta en el día a día de empresas y gobiernos. La incertidumbre sobre ciertos valores ya sea el precio de la energía, la evolución de una epidemia, la intensidad de precipitaciones, etc., plantea dificultades para una toma de decisiones adecuada y, por lo tanto, el desarrollo de modelos de predicción precisos es de gran utilidad.

En muchas ocasiones, la incertidumbre se refiere a observaciones futuras que toman valores enteros no negativos o de conteo. Para el tratamiento de las series temporales correspondientes, aunque sea posible el uso de modelos tradicionales, los modelos dedicados que consideran observaciones enteras y no negativas presentan numerosas ventajas, e.g. predicciones puntuales más fáciles de interpretar e intervalos de predicción que nunca incluyen valores no factibles. El objeto de esta tesis, resultado de la realización de un doctorado industrial, es contribuir al estado del arte en el contexto de la modelización de series temporales con datos de conteo. A fin de lograr esto, se cubren tres objetivos principales:

- Desarrollo de modelos para series temporales de conteo generales.
- Desarrollo de modelos para series temporales con gran cantidad de ceros y sobredispersión.
- Elaboración e implementación de algoritmos eficientes y escalables para aplicar los modelos anteriores a grandes cantidades de datos.

La contribución al ámbito de las series de conteo generales, comúnmente encontradas en datos relativamente agregados, se basa en varios modelos de Poisson no homogéneos y sus correspondientes algoritmos basados en análisis bayesiano. Estos modelos son capaces de incorporar múltiples combinaciones de efectos frecuentemente encontrados en series que surgen en diversas aplicaciones. Ilustramos la validez de estas contribuciones con un ejemplo real del campo de la seguridad aérea en el que los modelos desarrollados ofrecen mejor rendimiento que otros modelos establecidos para series temporales de este tipo.

También contribuimos al modelo de series temporales con gran cantidad de ceros y posible sobredispersión (segundo objetivo), con el desarrollo de modelos univariantes y multivariantes. Estos nuevos modelos se basan en mixturas de modelos bayesianos de espacio de estados. Mostramos que mejoran las predicciones respecto a otros modelos en un problema real de predicción de la demanda para la gestión de inventarios en venta al por menor.

Por último, el cumplimiento del tercer objetivo se ha logrado mediante el desarrollo de una librería para modelizar y obtener predicciones de series temporales de conteo. Esta librería está concebida para ofrecer la mayor flexibilidad posible, además de implementar los modelos introducidos en la tesis, permite la creación y el uso de otros nuevos. También proporciona herramientas útiles para el análisis exploratorio y evaluación de las predicciones. El código de esta librería no es de acceso libre por ser un producto comercial desarrollado dentro del Doctorado Industrial y propiedad de la empresa colaboradora.

A partir de los contenidos de esta tesis doctoral se han elaborado los siguientes dos artículos en diversos estados de publicación:

> Flores, B., Rios Insua, D., Alfaro, C. & Gomez, J. (2022). Forecasting Aviation Safety Occurrences. Por aparecer en *Applied Stochastic Models in Business and Industry*, `https://doi.org/10.1002/asmb.2675`. **Premio 2020 ASA-TSIG Student Paper**.

Flores, B. & Rios Insua, D. (2022). Demand Count Time Series Forecasting in Retail. *Enviado.*

Además, una serie de presentaciones en conferencias nacionales e internacionales también se han derivado de su contenido:

Flores, B. & Martin Hernandez, J. (May, 2019). Large scale dynamic forecasting for distributed inventory management. Presentación de póster en la *2nd Bringing Young Mathematicians Together (BYMAT) Conference*, ICMAT, Madrid, España.

Flores, B. & Martin Hernandez, J. (June, 2019). Large scale dynamic forecasting for distributed inventory management. Presentación de póster en la *11th Bayesian Inference in Stochastic Processes (BISP) Conference*, Real Academia de Ciencias, Madrid, España.

Flores, B., Rios Insua, D., Alfaro, C. & Gomez, J. (March, 2020). Forecasting aviation safety occurrences. Presentación de póster en el *Games, Decisions, Risk and Reliability (GDRR) Transportation Workshop*, SAMSI, Durham, Estados Unidos.

Flores, B. (August, 2020). Forecasting aviation safety occurrences. Presentación oral en el *JSM 2020 TSIG(ASA) Meeting*, Virtual.

Flores, B. (December, 2020). Modelos para predecir series temporales de conteo con un ejemplo de aplicación. Presentación oral en el *1st IMEIO-DecData: Decisión Optimización y Ciencia de Datos Workshop*, UCM, Madrid, España.

Flores, B. (May, 2021). Forecasting count series in retail. Presentación oral en la *12th Bayesian Inference in Stochastic Processes (BISP) Conference*, Virtual.

Flores, B. (September, 2021). Forecasting count time series in retail. Presentación oral en la *21th annual European Network for Business and Industrial Statistics (ENBIS) Conference*, Virtual.

Flores, B. (June, 2022). Bayesian predictive forecasting for retailing. Presentación oral en el *2022 Sociedad de Estadística e Investigación Operativa (SEIO) Congress*, Universidad de Granada, Granada, Spain.

Chapter 1

Introduction

1.1 Motivation

The last decade has seen a boom in the capacity of companies and governments to exploit numerous advances in information and communication technologies and statistical modeling, with the aim of collecting and processing relevant market and population data to support their decision-making processes. In many cases, a recurrent problem faced is the lack of accurate forecasts of some quantities of interest, be it the demand of a product, the number of accidents, or the cases of a new disease. This uncertainty concerning future values can be dealt with as a time series problem.

One field where time series forecasting has special relevance, and which motivated part of the research in this thesis, is aviation safety (AS). The development of a methodology for AS risk management at national level required as a major component accurate models to forecast the number of safety occurrences, i.e., safety-related events which endanger or which, if not corrected or addressed, could endanger an aircraft and its occupants; and includes, in particular, accidents or serious incidents. Despite the high safety levels of the aviation industry, occurrences continue to take place (ICAO, 2019). These may entail undesirable consequences like deaths, injured people, delays, aircraft destruction or reputation loss, among others. Countries develop national

AS plans, implemented through regulations and/or resource allocation, to try to make safety occurrences in their airspace less frequent and/or less severe should they happen. As air freight and passenger traffic is expected to increase in the forthcoming years (IATA, 2019), notwithstanding the current pandemic, the implementation of effective safety plans in air transportation is of major importance for governments, not only for the safety of its citizens, but also because economic prosperity and employment critically depend on a robust flow of goods and people. Therefore, having good quality occurrence forecasting models is paramount to properly manage risks, maintain the confidence of its users and preserve the status of aviation as a safe transportation mode. The problem is involved due to the presence of complex effects like seasonality, trends or stress that impact the rates of various occurrences and the uncertainty about future number of operations.

Another field in which we are interested in time series modeling is the other one which motivates this thesis: inventory management, and, specifically, the demand forecasting problems faced by many large retail companies. Consider a company with several hundreds of stores, each one with thousands of products for sale. Stores usually have little storage capacity beyond that available on their shelves, and, therefore, a limited number of units of each product, which makes them more susceptible to suffer out-of-stock (OoS) events when the demand is significantly higher than anticipated. Since increasing the storage capacity induces higher costs, and is generally not an option, the need for accurate demand forecasting models is crucial to avoid OoS situations entailing negative consequences, like economic or reputation losses. The problem of forecasting demand in retail has numerous difficulties, there are usually multiple relevant hierarchies (product family, store section, store, neighborhood, city, region, country) and we are interested in forecasting both aggregated and individual demand at any hierarchy level. At the most disaggregated levels, time series tend to be very diverse with sales of *intermittent demand* products showing low counts and many days of zero sales; whereas series of *high* *demand* products will rarely do so. Also, some can show overdispersion, that is, a variance much higher than the mean, while in others equidispersion is the norm.

These two motivating problems have in common that the observations of the involved time series are non-negative integers (*counts*). Time series count data are widely observed in practice and usually, unless dealing with very high counts, greatly benefit from dedicated modeling approaches. This thesis provides contributions to count time series forecasting, proposing novel models and their accompanying algorithms, as well as their efficient implementation.

Before detailing our specific research objectives (Section 1.4), let us provide an introduction and literature review of generic models for count time series in Section 1.2, and for count time series with frequent zeroes and overdispersion in Section 1.3. Finally, we end this introductory chapter with a glimpse of the structure of the dissertation.

1.2 General count time series models

A time series is a set of quantities or observations ordered in time. These can be collected at equally spaced time points or not; we denote the t-th observation as y_t $(t = 1, 2, ...)$, which can be a scalar (e.g. sales of ice-cream on day t at $shop_1$) or a k-vector (e.g. sales of ice-cream on day t at $shop_1$, ..., $shop_k$). This thesis focuses on time series of counts, with entries in $y_t \in \mathbb{Z}^{\geq 0}$.

Using the terminology in Cox (1981), time series models for count data can be divided between *observation driven* and *parameter driven*. Let y_t be an observed random variable at time t generated by a distribution that has ϕ_t as a time-varying parameter (e.g. the natural parameter η_t of an exponential family distribution), and $y_{1:t-1} = \{y_1, ..., y_{t-1}\}$ be all past observations. In an *observation driven* model, dependence between the observations in a time series is represented directly, e.g. through an autoregressive or moving average structure,

$$\phi_t = \phi(y_{1:t-1}, \epsilon_t),$$

where ϵ_t is a random innovation at time t. In a *parameter driven* model, there is an underlying latent process that induces dependence between the observations, e.g. through a state vector evolving according to a Markov process,

$$\phi_t = \phi^*(\phi_{1:t-1}, \epsilon_t^*),$$

where ϵ_t^* is a pure noise innovation process.

In the first group, we have *variants* for count data of traditional times series models including integer autoregressive (INAR) (Alzaid & Al-Osh, 1990), generalized autoregressive moving average (GLARMA) (Benjamin et al., 2003), integer-valued GARCH (INGARCH) (Ferland et al., 2006), and autoregressive conditional Poisson (ACP) (Heinen, 2003) models.

The majority of *parameter driven* models make use of a state-space formulation with the state vectors having a Markovian evolution. Some recent examples of models in this group are West (2020), Aktekin et al. (2018), Chen et al. (2018), Aktekin and Soyer (2011) and Gamerman et al. (2013). This type of models offer several advantages over the *observation-driven* ones:

- As discussed in Snyder et al. (2008), *parameter driven state-space* models tend to be more flexible and provide a closer match to the empirical properties of series.

- Non-stationary time series are routinely handled by state-space models, whereas it can be challenging for *observation-driven* models (McKenzie, 2003). As an example, for low-counts, simple differencing may radically alter the nature of the time series, since it usually results in negative values.

- Interpretability is usually much better: the components of the state vector correspond to easily explainable concepts (level, trend, seasonal differences,...), which can help the practitioner anticipate reliability or robustness problems, and facilitates communicating the forecasts to decision makers.

- Also, as pointed in Yelland (2009), in observation-driven models for count series, the determinants of the correlation structure being themselves observations, are therefore restricted to the domain of non-negative integers. This generally requires the development of specially designed mechanisms (like the *binomial thinning* operator of the INAR models) to describe the correlation structure. On the contrary, state-space models do not require the development of akin mechanisms, the entries of the state vector can take any real value, and the correlation structure of the time series is expressed with conventional algebra (almost always linear).

As a disadvantage, *parameter driven state-space* models tend to be more computationally demanding. However, with the constant improvement in computational resources (hardware and software) through the years, this is becoming less of an issue. In general, the aforementioned advantages provide the main reasons of our extensive use of *parameter driven state-space* models in this thesis.

One of the first (and still quite common) approaches to forecasting count data has been the use of relatively simple models that assume Poisson distributed observations, as in Feller (1991), who fitted a Poisson distribution to the number of flying-bomb hits in each square of a grid mapping South London during WWII, or the basic model in Section 2.3.1 that makes use of a Bayesian conjugate analysis to forecast AS incidents. While these simple approaches can be adequate for stable time series, when applying them to the complex time series encountered in practical cases, their performance is lacking in general, requiring the development of more sophisticated models, as we shall be doing.

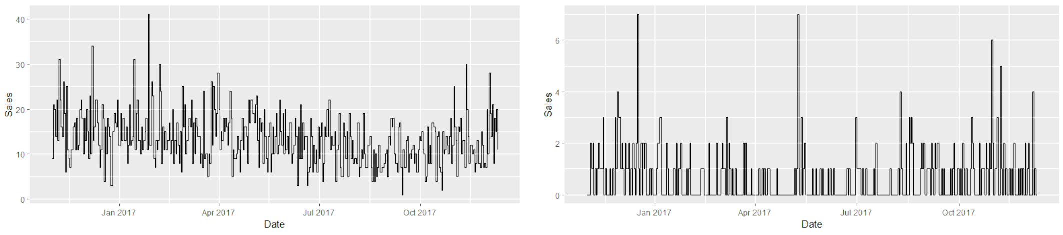
The use of Bayesian forecasting techniques in our models offer several benefits. One of the most important, as in many practical cases, will be the provision of full predictive distributions instead of only giving point forecasts. This is specially important to assess uncertainty on the one hand, and decision support, on the other, which is extremely relevant, for example, in the inventory management application for estimating the probability of OoS situations and, therefore, order planning. Also, the use of dedicated models for count data, specially when counts are relatively low, instead of traditional forecasting methods (usually assuming normally-distributed observations) yields more coherent and usable forecasts in practice (McCabe & Martin, 2005).

In many cases, multivariate data can benefit from joint models. However, the literature regarding multivariate models for integer-valued time series is somewhat scarce and has historically been that way because of the computational challenges they pose. Some recent work using *observation-driven* models can be seen in Pedeli and Karlis (2013) with the multivariate INAR approach; and *parameter-driven* models in Ravishanker et al. (2014), with a hierarchical multivariate Poisson time series model using Markov Chain Monte Carlo (MCMC). Soyer and Zhang (2021) provide an overview of recent advances in Bayesian modeling and analysis of multivariate time series of counts. As pointed out by Storvik (2002), the use of MCMC techniques present some disadvantages (for every new observation the chains need to be restarted and the simulation dimension becomes larger) over the particle filter approach we shall propose for the hierarchical multivariate models presented in Section 2.6.

The general approach adopted in this thesis for general count time series is based on dynamic Bayesian analysis of non-homogeneous Poisson models. Several models are proposed to accommodate different effects that might be encountered in time series to be forecast. Starting with a standard Poisson-Gamma model, we morph it into novel models (and novel combinations of standard models) to cover the aforementioned issues including stress, trends, seasonal, clustering and dependence effects.

1.3 Models for count time series with frequent zeros and overdispersion

With the aim of modelling time series with large amounts of zeros, Croston (1972) proposed an approach whereby non-zero values were forecast separately from the time between them. This approach is quite common when dealing with *intermittent demand* problems (Shenstone & Hyndman, 2005), although in many practical cases, such as in inventory management, there is a large quantity of time series to forecast that can be very different from one another (e.g. high and low demand products in Fig. 1.1). Thus, it is not feasible to have very specific models for each time series and there is a need for a more flexible approach.



(a) (b)

Figure 1.1: Examples of time series of sales from high (a) and low (b) or intermittent demand products.

Another more adaptable approach for dealing with series with a varying number of zero-observations are zero-inflated (ZI) and hurdle-shifted (HS) models. Both provide mixtures of a discrete probability mass function (usually a Poisson or a Negative Binomial distribution) and a Bernoulli to allow for more flexibility in modeling the probability of zero outcomes. They are becoming popular in business and other applications, e.g. Chen et al. (2016), Chen and Lee (2017), Snyder et al. (2012), McCabe and Martin (2005), Agarwal et al. (2002) or Schmidt and Pereira (2011). ZI models, as defined by Lambert (1992), add additional probability mass to the outcome of zero. As an example, for a Poisson distribution, this is represented by

$$p(y_t|\pi_t, \lambda_t) = \begin{cases} (1-\pi_t) + \pi_t \, Po(0|\lambda_t), & \text{if } y_t = 0, \\ \pi_t \, Po(y_t|\lambda_t) & \text{if } y_t > 0, \end{cases}$$

i.e. $y_t = z_t x_t$ with $z_t \sim Ber(\pi_t)$ and $x_t \sim Po(\lambda_t)$. Hurdle models, on the other hand, are formulated as pure mixtures of zero and non-zero outcomes. For the same Poisson distribution with parameter λ_t, the probability mass function for the likelihood is defined by

$$p(y_t|\pi_t, \lambda_t) = \begin{cases} (1 - \pi_t), & \text{if } y_t = 0, \\ \pi_t \, \frac{Po(y_t|\lambda_t)}{1 - Po_{\mathrm{CDF}}(0|\lambda_t)} & \text{if } y_t > 0, \end{cases}$$

where Po_{CDF} is the cumulative distribution function of the Poisson distribution. This is equivalent to $y_t = z_t(x_t + 1)$ with $z_t \sim Ber(\pi_t)$ and $x_t \sim Po(\lambda_t)$. The hurdle model is similar to the zero-inflated model, but somewhat more flexible in that the zero outcomes can be deflated as well as inflated, as necessary.

The other main obstacle to obtaining accurate forecasts is the presence of time series showing overdispersion, which can pose a problem for many count models that assume Poisson distributed observations and, thus, that the mean and variance of the observations is the same. In many practical cases, some series exhibit overdispersion and others equidispersion. Because of this, the negative binomial distribution is sometimes used as its variance is greater than the mean, while having the Poisson distribution as a limiting case (stopping-time parameter approaching infinity) and, therefore, still accommodating equidispersed series. Another common option is to maintain the Poisson introducing further randomness, e.g. by considering its parameter as a random variable that changes according to an autoregresive process (Snyder et al., 2008), or using discount factors in the predictors (Berry & West, 2020). We adopt the first approach, using negative binomial distributions (univariate or multivariate).

The models developed in this thesis to deal with forecasting count time series with varying levels of zero-observations and dispersion are Bayesian state-space models, which have proven useful in a range of count time series contexts, including dynamic network studies (Chen et al., 2018, 2019), consumer demand (Aktekin et al., 2018) or retail (Berry et al., 2020). These new models incorporate several of the approaches previously mentioned given as a result a flexible family of models. In particular, we shall use mixtures of

Dynamic Generalized Linear Models (West et al., 1985).

Multivariate models frequently use common latent factors for the individual time series and some form of decouple/recouple strategy. West (2020) gives a comprehensive review of some recent developments with multivariate count time series using DGLMs and this approach. We instead, explore the use of multivariate DGLMs.

1.4 Objectives

The aim of this industrial thesis is the development of new models and methodologies that improve the performance of current common approaches used to forecast two types of counts series: general count time series; and count time series with frequent zero-observations and overdispersion. The three main objectives of the dissertation are now detailed.

General count time series

Development of models to forecast common count time series, i.e. those with relatively *high counts* and few zero-counts, that can incorporate some combination of the following effects that are frequent in many practical cases.

- **Trends and seasonalities**, as e.g. in disease monitoring, where we can see exponential growth trends and annual seasonality peaking in winter time. It should be possible to incorporate them in the forecasting procedure.

- **Stress** effect. This is really important in safety and reliability environments: the failure rate of a piece or a process tends to increase with its use, as is the number of *human-caused* accidents with the pressure placed upon the people involved.

- **Clustering** of some kind, when dealing with a large number of time series is to be expected. It is only logical to take advantage of this to exploit cross series relationships with the objective of obtaining better forecasts.

- **Severities** or proportions of the observed quantity that belong to a finite number of groups, are a quantity of interest in many fields, e.g. the classification of accidents or infections according to how severe they are.
- **Under-reporting**, intentional or not, can appear in numerous application domains. For instance, in relation to the previous example, it is more common among the less *severe* incidents (due to the lack of a strong reporting culture) or infections (which could pass unnoticed).

Finally, the models proposed had to improve the performance over other commonly used ones in the AS application domain mentioned in Section 1.1.

This led to the paper Flores et al. (2022).

Time series with frequent zeros and overdispersion

Development of novel models to forecast time series with frequent zeros and overdispersion. These kinds of time series usually appear when dealing with highly disaggregated data, as in the retail application domain introduced in Section 1.1. Therefore, they tend to imply working with a much larger number of time series than when dealing with more aggregated data. Consequently, more emphasis was placed on the models being:

- *Automatic.* Large amount of time series to be modeled require that the intervention of modeling experts be kept to a minimum.
- *Scalable.* The size of external information in the form of covariates is usally also big. It is needed to maintain tractability, computation time and storage capacity, while scaling up.
- *Flexible.* Often, time series can be very diverse, some can have extremely low counts and others not, different dispersion levels, etc. Therefore, models must be adaptive enough.

As with the models for generic series, models here also had to be able to exploit cross series relations. Therefore, univariate and multivariate versions, with their corresponding forecasting procedure needed to be developed. The application domain in this case was inventory management.

A secondary objective, is the use of the full predictive distributions k-steps ahead to obtain the corresponding predictive cumulative distribution to develop a methodology to support informed decisions. For example, in the inventory management problem, with the stock information and demand predictions obtained from the models from the previous objective, we were interested in a methodology to make decisions that reduced the probability of future OoS events.

This led to the paper Flores & Rios Insua (2022).

Implementation

All the new models and procedures developed to achieve the three previous objectives had to be implemented in a library under a common programming language (Python in this case). The algorithms developed and their implementation had to be conceived in a general enough manner so as to be useful to forecast any count time series, no matter what the application domain was. Also, it had to be usable by anyone, regardless of the level of knowledge in modeling time series.

This led to the countTS Python library.

1.5 Dissertation structure

Chapter 2 deals with the objective of developing general models for count time series with relatively high counts. We present a framework to forecast these type of time series. It covers novel models as well as novel combinations of earlier models. They incorporate effects commonly encountered in these type of time series. This is illustrated with an application to AS data.

Chapter 3 introduces methodologies and models for large-scale dynamic forecasting of non-negative count series with frequent zeros and overdispersion. We also address the secondary objective of decision making under the results of the models. A demand forecasting problem faced by a major retail company exemplifies the developments.

In Chapter 4, the implementation of the models in the previous Chapters 2 and 3, and algorithms for the forecasting in both application domains is given, with examples.

Chapter 5 summarizes all the research carried out and raises new questions derived from the work in Chapters 2 to 4, and from where new research lines emerge.

Chapter 2

Models for general count time series

2.1 Introduction

This chapter fulfills the first objective in Section 1.4, providing new models and a methodology to forecast general count time series that can present several combinations of effects. The generic models are illustrated with the motivating problem of AS introduced in Section 1.1.

In earlier work (Rios Insua et al., 2018), a framework to support AS risk management at country level was presented. Such framework, which has been successfully applied in Spain (Elvira et al., 2020), supports a government in deciding how to allocate resources to improve AS levels. It is based on Bayesian decision analysis (Clemen & Reilly, 2013) and includes, as basic ingredients, models to: (a) forecast the numbers of various safety occurrences; (b) forecast occurrence severities; (c) forecast the consequences of safety occurrences; and (d) assess such consequences through a multi-attribute utility function. Such models are integrated to monitor safety, screen occurrences and, more importantly, allocate AS resources. Rios Insua et al. (2019) details ingredients (c) and (d). The developments in the present chapter supports parts (a) and (b), providing a methodology to forecast general count time series.

Most of the literature concerning AS occurrence forecasting has focused on predicting the circumstances that cause them, e.g. aircraft icing (McCann, 2005) or turbulence (Gill & Buchanan, 2013). While this can be useful for short horizon and operational planning, AS strategic planning at country or airline level requires focusing more on medium to long term forecasts of occurrences. Moreover, due to the many different types of AS occurrences, there is a need for flexible models capable of forecasting occurrences of very different nature, instead of models that focus on one particular occurrence, as e.g. those for runway excursions (Ayra et al., 2019), flight delays (Khanmohammadi et al., 2016) or go-around/missed approaches (Subramanian & Rao, 2018). Thus, our aim is at providing a comprehensive modeling approach that can accurately forecast diverse AS occurrences with long or short horizons, i.e. the interest lies in modeling time series of non-negative counts, taking into account the particularities of our application domain.

Note though that while the models in this Chapter are applied to the particular problem of forecasting AS occurrences, they can also be used to predict safety and reliability occurrences in other areas, like maritime transport, industry, or supply chain networks. More generally, they can be applied to any domain in which there are series with relatively high values that justify the need for specific models: not so high that they can be adequately modeled with standard models, but not so low that there are frequent zeros. We revisit this last case in Chapter 3, and Chapter 4 describes a package covering as well standard models.

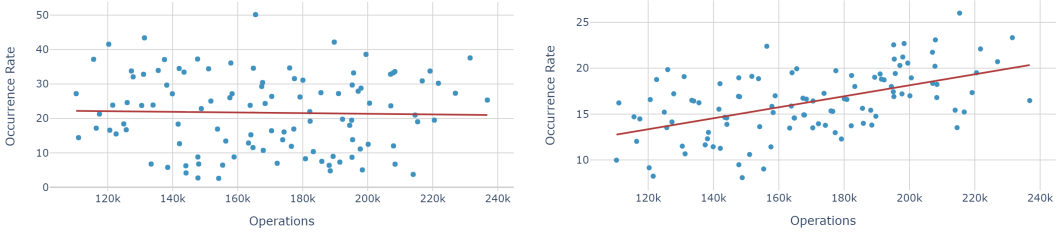
Section 2.2 introduces the problem and provides exploratory analysis illustrating key effects in the proposed forecasting domain. They are incorporated gradually in Section 2.3, stemming from a basic model to which we add one feature in turn to deal with such effects. These models are then expanded to account for the uncertainty in the number of operations, an especially important theme when interested in long-horizon AS forecasts. Then, Section 2.5 covers models that allow us to predict occurrence severity, and deal with the possible under-reporting in some of those severity groups. The models are illustrated with three cases in Section 2.6.

We use probabilistic influence diagrams throughout (Shachter, 1988) to graphically support the presentation of models. We propose novel algorithms to forecast with our models, and whose convergence follows from standard arguments that can be seen in, e.g., Gelman et al. (2013).

2.2 Exploratory data analysis of aviation safety occurrences

In our motivating scenario, airlines and national AS agencies periodically register occurrences along with the number of operations with the aim of monitoring and improving their occurrence rate as well as reducing their severity. As an example, in our case, referring to AS risk management at country level, 86 occurrence types are considered, ranging from *runway incursions* to *ground handling events* going through *low altitude operations* or *ground collisions*. Each of the occurrences is classified into one of five severity groups as proposed by the International Civil Aviation Organization (ICAO, 2018): (1) *Accident* (entailing fatalities and/or aircraft destruction); (2) *Serious Incident*; (3) *Major Incident*; (4) *Significant Incident*; and (5) *Occurrence with no safety effect*. Thus, we may talk, for example, about a *severity 2 ground handling* occurrence. For our analysis, we use daily recorded occurrences from 2010 to 2018. For each occurrence, the following information is available: date and airport code; its type and severity; the corresponding number of fatalities, and serious and minor injuries, if any; and, finally, information concerning the aircraft such as its model and maximum certificated take-off mass. Furthermore, for each airport, the recorded data include information such as its ICAO code, latitude and longitude and number of daily flight operations. In this chapter, as well as in the strategic risk management aspects of the parent project, we focus on monthly forecasts aggregating the data as required.

For a given occurrence type, e.g. *runway incursions*, the data available at the end of the k-th forecasting period is denoted with D_k and consist of the operations-occurrences pairs $\{(n_1, x_1), \ldots, (n_k, x_k)\}$, where n_i represents the number of operations and x_i, the number of occurrences during the i-th period. Hereafter, we introduce the typical effects that can be observed in AS occurrence data, with the aid of exploratory tools based on the primary data $\{(n_i, \hat{\lambda}_i)\}_{i=1}^{k}$, with $\hat{\lambda}_i = x_i/n_i$ designating the observed occurrence rate at the i-th period (month).

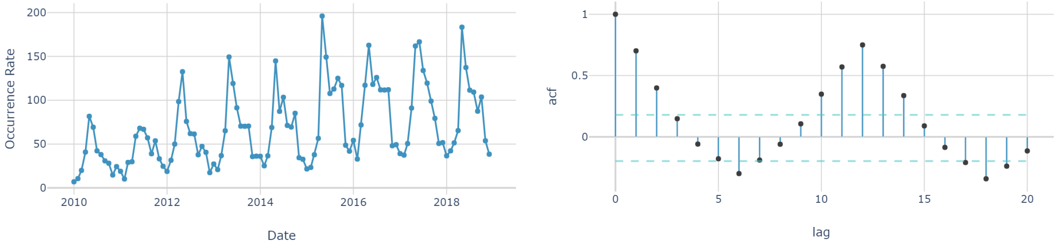


(a) (b)

Figure 2.1: Stable relation (a) and stress effect (b) in occurrence rates.

Figure 2.1(a) represents occurrence rate versus number of operations for the *Communication, navigation and surveillance failures* type, suggesting a relatively stable occurrence rate. However, several effects may appear altering such stability. The first one is showcased in Figure 2.1(b) referring to the *TCAS warning* occurrence. In it, we observe that higher numbers of operations induce higher occurrence rates, perhaps due to the increasing pressure over the involved agents (pilots, controllers). We refer to this as a *stress effect*.

Some occurrences can show a *seasonal effect*, typically when affected by regional weather patterns. This is the case of the *bird strike* occurrence, Figure 2.2(a), whose time series of rates shows a pronounced seasonal behavior produced by natural causes related to the migratory movements of birds and their passage near airports. The auto-correlation function (ACF) of occurrence rates can be used to explore and showcase the presence of this effect, Figure 2.2(b).



(a) (b)

Figure 2.2: Seasonal effect of the *bird strike* occurrence.

Sometimes, we can discern a, possibly piecewise, linear variation in the rate, expressed through a *trend*, evolving from one period to another, as Figure 2.3, displaying the annual *wind shear* occurrence rate, shows. Other times, certain *grouping effect* is appreciated over an occurrence rate, as shown by the two clusters of airports in Figure 2.4, referring to the *bird strike* occurrence. As a possible explanation, airports with akin location face analogous weather patterns that, in turn, may induce similarities in the corresponding occurrence rates. It could also be the case that a group of airports is operated by a same company with specific operational procedures which, consequently, induces specificities in their occurrence rates.

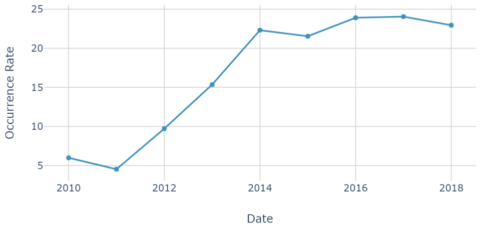


Figure 2.3: Trend in *wind shear* occurrence.

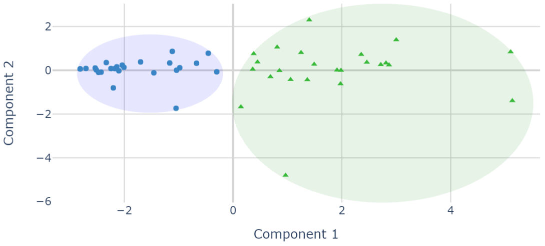


Figure 2.4: Group effect of *bird strike* occurrence.

Finally, several occurrences show relevant *correlation* due to common causes, for technical or physical reasons, or because one of them is a precursor of another. Figure 2.5 portrays correlations between eleven of them. Note, e.g., the high correlation between the *wind* and *wind shear* occurrence rates, most likely due to a common cause.

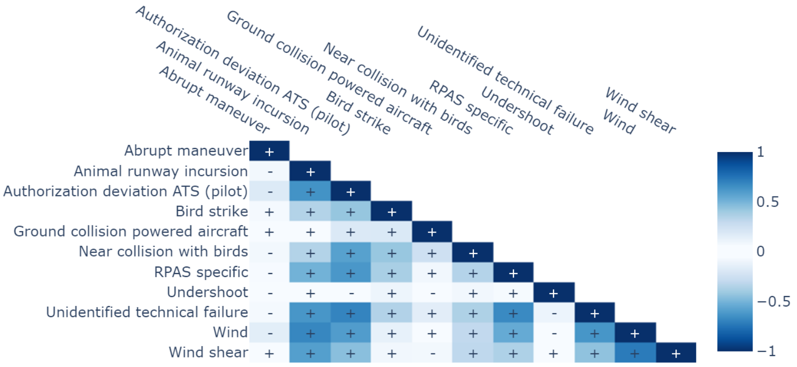


Figure 2.5: Correlation matrix for eleven occurrence types.

Several cases will include more than one effect, as illustrated in Section 2.6. Indeed, Table 2.1 summarizes the effects detected through exploratory data analysis for the 86 types of occurrences relevant in our case. As an example, twelve of the occurrence types suggested incorporating both seasonal and linear effects in the corresponding model.

Model	Types of occurrences
No Effect	25
Stress	1
Seasonal	1
Linear	44
Stress+Seasonal	3
Seasonal+Linear	12

Table 2.1: Models suggested for the 86 different types of AS occurrences.

2.3 Forecasting AS occurrences with dynamic Poisson models

Let us describe the class of models used to predict the monthly number of occurrences for the 86 relevant types. Our emphasis is on monthly forecasts, although other time granularity forecasts will be required (mainly, annual, for strategic planning, and weekly, for monitoring purposes). We start with a standard Poisson-Gamma model (Section 2.3.1) and gradually adapt it producing novel models to incorporate procedures to deal with the effects described in Section 2.2.

2.3.1 Basic model

Let us begin with the basic version of our model. In it (as in Figure 2.1(a)) the occurrence rate remains relatively stable throughout the year (and over the years), not appreciating the effects suggested in Section 2.2. We deal with it with a standard Poisson-Gamma model, taking into account the approximation of the Poisson distribution to the binomial, e.g. Rios Insua et al. (2012), and that the Gamma is conjugate for the Poisson. Thus, our basic model is

$$\begin{aligned} x_k|\lambda, n_k &\sim Po(\lambda n_k), \\ \lambda &\sim Ga(a, p), \end{aligned}$$

where x_k is the number of occurrences during the k-th period; n_k is the number of operations during such period; and, finally, λ is the occurrence rate. It is well known that the posterior distribution for the occurrence rate at the end of the k-th period, after D_k becomes available, is $\lambda|D_k \sim Ga(a_k, p_k)$, with $a_k = a_{k-1}+x_k$ and $p_k = p_{k-1}+n_k$; and $a_0 = a$, $p_0 = p$. Moreover, the posterior predictive distribution for the number of occurrences during the next period is $x_{k+1}|D_k \sim NegBin(a_k, p_k/(n_k+p_k))$, from which the predictive mean, variance and intervals follow easily (Rios Insua et al., 1999).

2.3.2 Variants over the basic model

Several non-trivial variants of the basic model need to be considered to take into account the effects in Section 2.2 leading to novel models. They are dealt with one at a time. Section 2.6.1 provides an example with an effect combination (linear trend, seasonal and group), which aggregates the proposed modeling ideas.

Stress effect

Figure 2.6 reflects an influence diagram showing how we deal with stress effects (Figure 2.1(b)). Its simplest expression would be based on a linear relationship between the rate and the number of operations as follows:

$$\begin{gathered} x_k|\lambda, n_k \sim Po(\lambda n_k) \\ \lambda = a n_k + b + \epsilon_k, \quad \epsilon_k \sim N(0, \sigma^2), \\ a \sim N(\mu_a, \sigma_a^2), \quad b \sim N(\mu_b, \sigma_b^2), \quad \sigma^2 \sim \textit{Inv-Gamma}(\alpha, \beta). \end{gathered} \tag{2.1}$$

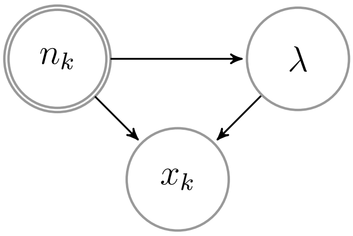


Figure 2.6: Influence diagram for stress effect (k-th period).

Given data D_k, the posterior $p(\lambda, a, b, \sigma^2|D_k)$ is easily seen to be proportional to

$$\lambda^{\sum_{i=1}^k x_i} \sigma^{-3-2\alpha} \exp\Bigg(-\lambda \sum_{i=1}^k n_i - \frac{(\lambda - a n_k - b)^2}{2\sigma^2} - \frac{(a-\mu_a)^2}{2\sigma_a^2} - \frac{(b-\mu_b)^2}{2\sigma_b^2} - \frac{\beta}{\sigma^2}\Bigg).$$

Then, the conditional posterior distributions are

$$\begin{aligned} p(a|\lambda, b, \sigma^2, D_k) &\sim N\left(a \,\middle|\, \frac{\sigma^2 \mu_a + n_k \sigma_a^2 (\lambda - b)}{\sigma^2 + n_k^2 \sigma_a^2}, \frac{\sigma^2}{n_k^2 + \sigma^2/\sigma_a^2}\right), \\ p(b|\lambda, a, \sigma^2, D_k) &\sim N\left(b \,\middle|\, \frac{\sigma^2 \mu_b + \sigma_b^2 (\lambda - a n_k)}{\sigma^2 + \sigma_b^2}, \frac{1}{1/\sigma_b^2 + 1/\sigma^2}\right), \\ p(\sigma^2|\lambda, a, b, D_k) &\sim \textit{Inv-Gamma}\left(\sigma^2 \,\middle|\, \alpha + \frac{1}{2}, \beta + \frac{1}{2}(b + a n_k - \lambda)^2\right), \\ p(\lambda|a, b, \sigma^2, D_k) &\propto \lambda^{\sum x_i} \exp\left(-\frac{1}{2\sigma^2}\left(\lambda^2 - 2\lambda\left(a n_k + b - \sigma^2 \sum_{i=1}^k n_i\right)\right)\right). \end{aligned}$$

Based on these, Algorithm 2.1 provides a hybrid scheme to sample from the posterior, using Gibbs steps to sample from the conditional posteriors of a, b and σ^2 and a Metropolis-Hastings step to sample from the conditional posterior of λ. Note that the non-negativity of λ is controlled through the support of the proposal distribution of our MCMC sampler, $q(\lambda)$. For this, we recommend using $Ga\big(1 + \sum_{i=1}^{k} x_i, \frac{\lambda}{2\sigma^2} + \sum_{i=1}^{k} n_i\big)$, which guarantees non-negativity of λ. $\pi(\lambda)$ is the expression proportional to the posterior of λ.

```
Set $a_0, b_0, \lambda_0, \sigma_0^2, j = 1$;
while convergence not detected do
    Sample $\sigma_j^2 \sim \textit{Inv-Gamma}\big(\alpha + \frac{1}{2}, \beta + \frac{1}{2}(b_{j-1} + a_{j-1}n_k - \lambda_{j-1})^2\big)$;
    Sample $b_j \sim N(\frac{\sigma_j^2\mu_b + \sigma_b^2(\lambda_{j-1} - a_{j-1}n_k)}{\sigma_j^2 + \sigma_b^2}, \frac{1}{1/\sigma_b^2 + 1/\sigma_j^2})$;
    Sample $a_j \sim N\Big(\frac{\sigma_j^2\mu_a + n_k\sigma_a^2(\lambda_{j-1} - b_j)}{\sigma_j^2 + n_k\sigma_a^2}, \frac{\sigma_j^2}{n_k^2 + \sigma_j^2/\sigma_a^2}\Big)$;
    Sample $\lambda_j^* \sim q(\lambda_{j-1})$;
    Calculate $\alpha = \min\Big(1, \frac{\pi(\lambda_j^*)}{\pi(\lambda_{j-1})}\frac{q(\lambda_{j-1})}{q(\lambda_j^*)}\Big)$;
    Do $\lambda_j = \begin{cases} \lambda_j^* & \text{with probability}\,\alpha \\ \lambda_{j-1} & \text{with probability}\,(1-\alpha) \end{cases}$;
    $j \leftarrow j + 1$;
end
```

Algorithm 2.1: MCMC sampler for *Stress Effect* model (2.1).

In particular, note that we may check whether the posterior distribution of a concentrates around 0 to eventually discard the presence of a stress effect. We do this by computing the posterior probability of an interval around 0, as illustrated in Section 2.6.2.

Regarding the predictive distribution for the number of occurrences, we use

$$\begin{aligned} Pr(x_{k+1} = z|D_k) &= \iiiint Pr(x_{k+1} = z|\lambda, n_k)\, p(\lambda|a, b, \sigma^2, D_k)\, p(a, b, \sigma^2|D_k)\, d\lambda\, da\, db\, d\sigma^2 \\ &\approx \frac{1}{N}\sum_{j=1}^{N} Pr(x_{k+1} = z|\lambda_j, n_k) = \frac{n_k^z}{N z!}\sum_{j=1}^{N} \exp(-\lambda_j n_k)\,(\lambda_j)^z, \end{aligned}$$

based on the sample $\{\lambda_j\}_{j=1}^{N}$ from Algorithm 2.1. From it, predictive means, variances and intervals follow easily.

Effects dealt with through Dynamic Linear Models

In many cases, one cannot consider the occurrence rate as constant, recall Figures 2.2 and 2.3. Trend and seasonal effects may be modeled with Dynamic Linear Models (DLM) (West & Harrison, 1997); however, as observations are considered to come from a Poisson distribution, they cannot be directly dealt with a DLM. Then, the model is described through

$$\begin{aligned}
&x_k|\lambda_k, n_k \sim Po(\lambda_k n_k),\\
&\lambda_k = \exp(u_k),\\
&u_k = \boldsymbol{F}_k\boldsymbol{\theta}_k + v_k, \qquad v_k \sim N(0, V_k),\\
&\boldsymbol{\theta}_k = \boldsymbol{G}_k\boldsymbol{\theta}_{k-1} + \boldsymbol{w}_k, \quad \boldsymbol{w}_k \sim N(\mathbf{0}, \boldsymbol{W}_k),\\
&\boldsymbol{\theta}_0 \sim N(\boldsymbol{m}_0, \boldsymbol{C}_0),
\end{aligned} \tag{2.2}$$

where $\boldsymbol{F}_k$ and $\boldsymbol{G}_k$ are known matrices; and v_k and $\boldsymbol{w}_k$ are independent sequences of normal variables with zero mean and variances V_k and $\boldsymbol{W}_k$, respectively. The exponential transformation in the second equation guarantees the positivity of the occurrence rate. The model is represented through the influence diagram in Figure 2.7, in which we obviate the deterministic relationship between λ and u.

Note that another option would be to use a Dynamic Generalized Linear Model (DGLM) (West et al., 1985) by eliminating the noise v_k above, and not assuming any particular distribution for $\boldsymbol{w}_k$, just its mean and variance. However, we prefer to use (2.2) since its dual source of error offers additional flexibility and adapts better to the motivating case. Models making use of DGLMs and its sequential updating might be beneficial in applications dealing with very large number of time series and/or time series with frequent zeros, which is not our case in this chapter. We return to this problem in Chapter 3.

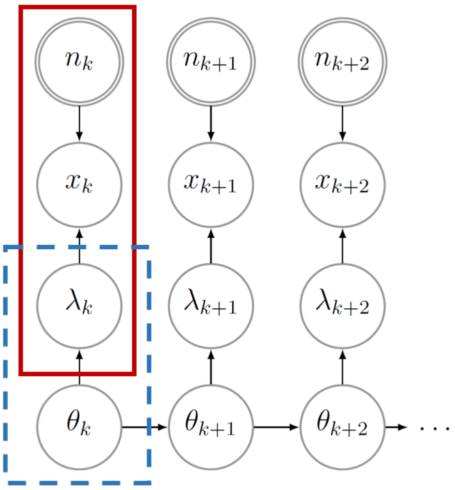


Figure 2.7: Influence diagram with log-rate as DLM. Poisson component, solid; DLM, dash.

Let us briefly discuss modeling possibilities that are convenient in our domain. The DLMs considered can be used as building blocks combined through the superposition principle (Prado & West, 2010) to form a model when both effects are deemed relevant.

Trend effect. The basic models to deal with a dynamic occurrence rate are the first order polynomial model, characterized by $\boldsymbol{F} = \boldsymbol{G} = 1$, and the second order polynomial model, or linear growth, based on

$$\boldsymbol{F}_1 = \begin{pmatrix} 1 & 0 \end{pmatrix}, \qquad \boldsymbol{G}_1 = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}.$$

We choose the later as it is more general and allows us to deal more adequately with our data sets, both for modeling and forecasting purposes.

Seasonal effect. With monthly data, when considering the presence of a seasonal effect of period 12, as in Figure 2.2, we use a DLM with the following regression vector

$$\boldsymbol{F}_2 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$

and evolution matrix

$$
\boldsymbol{G}_2 = \begin{pmatrix}
-1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0
\end{pmatrix}.
$$

Inference and prediction procedures are common in both cases (and their combination thereof) through this scheme:

Step 0. Forecast at period k. At the beginning of the k-th period, before observing x_k, we have the distributions $\pi(x_k|\lambda_k, n_k)$, $\pi(\lambda_k|u_k)$, $\pi(u_k|\boldsymbol{\theta}_k)$ and $\pi(\boldsymbol{\theta}_k)$ (typically, this last one will be characterized by a sample $\{\boldsymbol{\theta}_k^i\}_{i=1}^N$, with weights $\pi_k^i \geq 0$, and $\sum_{i=1}^N \pi_k^i = 1$). To make predictions about the number of occurrences x_k, the distribution is

$$
\begin{aligned}
\pi(x_k|n_k) &= \iiint \pi(x_k|\lambda_k, n_k)\, \pi(\lambda_k|u_k)\, \pi(u_k|\boldsymbol{\theta}_k)\, \pi(\boldsymbol{\theta}_k)\, d\lambda_k\, du_k\, d\boldsymbol{\theta}_k \\
&= \iint \pi(x_k|\exp(u_k), n_k)\, \pi(u_k|\boldsymbol{\theta}_k)\, \pi(\boldsymbol{\theta}_k)\, du_k\, d\boldsymbol{\theta}_k,
\end{aligned}
$$

estimated by simulation through

```
Sample $\{\boldsymbol{\theta}_k^i\}_{i=1}^N \sim \pi(\boldsymbol{\theta}_k)$ (possibly already available);
Do $\lambda_k^i = \exp(\boldsymbol{F}_k \boldsymbol{\theta}_k^i)$, for $i = 1, \ldots, N$;
Approximate the predictive $\pi(x_k|n_k)$ with
    $$\pi(x_k|n_k) \approx \frac{n_k^{x_k}}{N x_k!} \sum_{i=1}^{N} \exp(-\lambda_k^i n_k)(\lambda_k^i)^{x_k}.$$
```

The predictive mean and second moment are approximated through

$$\eta_k := E(X_k|n_k) = \sum x_k \pi(x_k|n_k) \approx \frac{n_k}{N}\sum_{i=1}^{N} \exp\left(\frac{1-2\boldsymbol{F}_k\boldsymbol{\theta}_k^i}{2V_k}\right),$$

$$E(X_k^2|n_k) = \sum x_k^2\ \pi(x_k|n_k) \approx \frac{n_k}{N}\sum_{i=1}^{N} \exp\left(\frac{1-2\boldsymbol{F}_k\boldsymbol{\theta}_k^i}{2V_k}\right) + \frac{n_k^2}{N}\sum_{i=1}^{N} \exp\left(\frac{2-2\boldsymbol{F}_k\boldsymbol{\theta}_k^i}{V_k}\right).$$

Therefore, the predictive variance is approximated through

$$\kappa_k^2 := V(X_k|n_k) \approx \frac{n_k}{N}\sum_{i=1}^{N} \exp\left(\frac{1-2\boldsymbol{F}_k\boldsymbol{\theta}_k^i}{2V_k}\right) + \frac{n_k^2}{N}\sum_{i=1}^{N} \exp\left(\frac{2-2\boldsymbol{F}_k\boldsymbol{\theta}_k^i}{V_k}\right) - \frac{n_k^2}{N^2}\left(\sum_{i=1}^{N} \exp\left(\frac{1-2\boldsymbol{F}_k\boldsymbol{\theta}_k^i}{2V_k}\right)\right)^2.$$

Step 1. Observation of x_k and update. Once x_k is observed, it is propagated to obtain $\pi(\theta_k|x_k)$. From step 0, preserve samples $\{\boldsymbol{\theta}_k^i\}_{i=1}^N$ and $\{\lambda_k^i\}_{i=1}^N$; for each $\boldsymbol{\theta}_k^i$, draw $\{u_k^{ih}\}_{h=1}^N \sim N(\boldsymbol{F}_k\boldsymbol{\theta}_k^i, V_k)$, do $\lambda_k^{ih} = \exp(u_k^{ih})$, $h = 1, \ldots, N$, and approximate

$$\pi(x_k|\boldsymbol{\theta}_k^i) = \iint \pi(x_k|\lambda_k, n_k)\pi(\lambda_k|u_k)\pi(u_k|\boldsymbol{\theta}_k^i)\, d\lambda_k\, du_k \approx \frac{n_k^{x_k}}{N x_k!}\sum_{h=1}^{N} \exp(-\lambda_k^{ih} n_k)(\lambda_k^{ih})^{x_k}.$$

Suppressing dependence on u_k, which is fixed, we get

$$\pi(\boldsymbol{\theta}_k^i|x_k) = \frac{\pi(x_k|\boldsymbol{\theta}_k^i)\pi(\boldsymbol{\theta}_k^i)}{\pi(x_k)} \approx \frac{\left(\frac{1}{N}\sum_{h=1}^{N}\pi(x_k|\lambda_k^{ih})\right)\pi(\boldsymbol{\theta}_k^i)}{\frac{1}{N}\sum_{i=1}^{N}\pi(x_k|\lambda_k^i)} \propto \pi(\boldsymbol{\theta}_k^i)\sum_{h=1}^{N} \exp(-\lambda_k^{ih} n_k)(\lambda_k^{ih})^{x_k}. \tag{2.3}$$

Step 2. Propagation to period $(k+1)$*.* We have

$$\pi(\boldsymbol{\theta}_{k+1}|D_k) = \int \pi(\boldsymbol{\theta}_{k+1}|\boldsymbol{\theta}_k)\pi(\boldsymbol{\theta}_k|x_k)d\boldsymbol{\theta}_k.$$

Thus, the distribution for the d-dimensional state vector $\boldsymbol{\theta}_k$ is approximated by

$$\begin{aligned}\pi(\boldsymbol{\theta}_{k+1}|D_k) &\approx \frac{1}{N}\sum_{i=1}^{N}\pi(\boldsymbol{\theta}_{k+1}|\boldsymbol{\theta}_k^i)\\ &= \frac{1}{N}\sum_{i=1}^{N}\frac{1}{\sqrt{(2\pi)^d|\boldsymbol{W}_k|}}\exp\Big(-\frac{1}{2}(\boldsymbol{\theta}_{k+1}-\boldsymbol{G}_k\boldsymbol{\theta}_k^i)'\boldsymbol{W}_k^{-1}(\boldsymbol{\theta}_{k+1}-\boldsymbol{G}_k\boldsymbol{\theta}_k^i)\Big),\end{aligned}$$

with $\{\boldsymbol{\theta}_k^i\}_{i=1}^N$ a sample of $\boldsymbol{\theta}_k|x_k$, from step 1.

After this, we would be again at step 0 and re-initiate the process. The particle filter in Algorithm 2.2 summarizes this sequence of steps, at the k-th iteration, there is a sample $\{\boldsymbol{\theta}_k^j\}_{j=1}^N$ from $\pi(\boldsymbol{\theta}_k|D_k)$ available, with weights $\{\pi_k^j\}_{j=1}^N$. Initially, $k=0$, and $\boldsymbol{\theta}_0^j \sim N(\boldsymbol{m}_0, \boldsymbol{C}_0)$, $j=1,\ldots,N$. Afterwards, the particles evolve according to the steps in (2.2), and weights are updated through (2.3) as new observations x_k are available. The effective sample size (N_{ESS}) is monitored. When it drops below a certain threshold, we resample. N is the sample size and T the number of iterations.

An MCMC approach could also be adopted for this model but, as pointed out by Storvik (2002) and Aktekin et al. (2018), this presents disadvantages over the particle filter approach.

```
Sample $\{\boldsymbol{\theta}_0^j\}_{j=1}^N \sim N(\boldsymbol{m}_0, \boldsymbol{C}_0)$;
Do $\pi_0^j = \frac{1}{N}, \quad j = 1, \ldots, N$;
for $k \leftarrow 1$ to $T$ do
    for $j \leftarrow 1$ to $N$ do
        Sample $\boldsymbol{\theta}_k^j \sim N(\boldsymbol{G}_k \boldsymbol{\theta}_{k-1}^j, \boldsymbol{W}_k)$;
        Do $\eta_k = \frac{n_k}{N} \sum_{j=1}^{N} \exp\left(\frac{1 - 2\boldsymbol{F}_k \boldsymbol{\theta}_k^j}{2V_k}\right)$;
        Do $\kappa_k^2 = \frac{n_k}{N} \sum_{j=1}^{N} \exp\left(\frac{1-2\boldsymbol{F}_k \boldsymbol{\theta}_k^j}{2V_k}\right) + \frac{{n_k}^2}{N} \sum_{j=1}^{N} \exp\left(\frac{2-2\boldsymbol{F}_k \boldsymbol{\theta}_k^j}{V_k}\right)$;
            $-\frac{{n_k}^2}{N^2} \left(\sum_{j=1}^{N} \exp\left(\frac{1-2\boldsymbol{F}_k \boldsymbol{\theta}_k^j}{2V_k}\right)\right)^2$;
        Read $x_k$;
        $\Delta_j = 0$;
        for $h \leftarrow 1$ to $N$ do
            Sample $u_k^h \sim N(\boldsymbol{F}_k \boldsymbol{\theta}_k^j, V_k)$;
            Do $\lambda_k^h = \exp(u_k^h)$;
            $\Delta_j \leftarrow \Delta_j + exp(-\lambda_k^h n_k)(\lambda_k^h)^{x_k}$
        end
        $\pi_k^j \leftarrow \pi_{k-1}^j \Delta_j$;
    end
    $\pi_k^j = \frac{\pi_k^j}{\sum_{j=1}^{N} \pi_k^j}$;
    Calculate $N_{ESS} = (\sum_{j=1}^{N} (\pi_k^j)^2)^{-1}$;
    if $N_{ESS} < N/2$ then
        Sample $\boldsymbol{\theta}_k^{j*} \sim \{\boldsymbol{\theta}_k^h, \pi_k^h\}_{h=1}^N, \quad j = 1, \ldots, N$;
        $\boldsymbol{\theta}_k^j \leftarrow \boldsymbol{\theta}_k^{j*}$;
        $\pi_k^j \leftarrow \frac{1}{N}$;
    end
end
```

Algorithm 2.2: Particle filter for Poisson DLM (2.2).

Group effect

A simple approach to dealing with *group effects* (Figure 2.4) would be to model each cluster of observations separately. A more elaborate version uses hierarchical modeling relating the cluster components through parameters coming from a same hyperdistribution. For L groups, using the basic model in Section 2.3.1, we have

$$\begin{gathered} x_k^i|\lambda^i, n_k^i \sim Po(\lambda^i n_k^i), \quad \lambda^i \sim Ga(a^i, p^i) \quad i = 1, ..., L \\ a^i \sim Ga(\alpha, \beta), \quad p^i \sim Ga(\gamma, \delta). \end{gathered} \tag{2.4}$$

The posterior distribution is

$$\begin{aligned} &\pi(\lambda^1, \ldots, \lambda^L, a^1, \ldots, a^L, p^1, \ldots, p^L|D_k) \propto \\ &\prod_{i=1}^{L} \left(\left[\prod_{j=1}^{k} \pi(x_j^i|\lambda^i, n_j^i) \right] \pi(\lambda^i|a^i, p^i)\pi(a^i|\alpha, \beta)\pi(p^i|\gamma, \delta) \right) \propto \\ &\prod_{i=1}^{L} \left[\frac{\exp\left(-p^i\lambda^i - a^i\beta - p^i\delta - \lambda^i \sum n_j^i\right) (\lambda^i)^{a^i-1+\sum x_j^i} (p^i)^{a_i+\gamma-1} \beta^\alpha \delta^\gamma (a^i)^{\alpha-1}}{\Gamma(a^i)\Gamma(\alpha)\Gamma(\gamma)} \right]. \end{aligned}$$

The conditional posterior distributions for each parameter $i = 1, ..., L$ are:

$$\begin{aligned} \pi(\lambda^i|a^i, p^i, D_k) &\propto \exp\left(-\lambda^i \left(p^i + \sum n_j^i\right)\right) (\lambda^i)^{a^i+\sum x_j^i-1} \\ &\sim Ga\big(a^i + \textstyle\sum x_j^i, p^i + \sum n_j^i\big), \\ \pi(p^i|\lambda^i, a^i, D_k) &\propto \exp(-p^i(\lambda^i + \delta))p^{i(a^i+\gamma-1)} \sim Ga(a^i + \gamma, \lambda^i + \delta), \\ \pi(a^i|\lambda^i, p^i, D_k) &\propto \frac{(\lambda^i p^i \exp(-\beta))^{a^i-1}(a^i)^{\alpha-1}}{\Gamma(a^i)}. \end{aligned}$$

The last ones lack a standard form, but can be treated through a Metropolis-Hastings step as shown in Algorithm 2.3, where $f(a^i)$ is the expression proportional to the posterior of a^i. We recommend $Ga(a^i|\alpha, -\log(\lambda^i p^i \exp(-\beta)))$ as the proposal distribution $q(a^i)$.

```
Set $\lambda_0^1, \ldots, \lambda_0^L, a_0^1, \ldots, a_0^L, p_0^1, \ldots, p_0^L, h = 1$;
while convergence not detected do
    Sample $\lambda_h^i \sim Ga(a_{h-1}^i + \sum x_j^i, p_{h-1}^i + \sum n_j^i); \quad i = 1, \ldots, L$;
    Sample $p_h^i \sim Ga(a_{h-1}^i + \gamma, \lambda_h^i + \delta); \quad i = 1, \ldots, L$;
    Sample $a_h^{*i} \sim q(a_{h-1}^i)$;
    Calculate $\psi = \min\left(1, \frac{f(a_h^{*i})}{f(a_{h-1}^i)} \frac{q(a_{h-1}^i)}{q(a_h^{*i})}\right)$;
    Do $a_h^i = \begin{cases} a_h^{*i} & \text{with probability } \psi \\ a_{h-1}^i & \text{otherwise.} \end{cases}$;
    $h \leftarrow h + 1$;
end
```

Algorithm 2.3: MCMC sampler for *Group Effect* model.

Dependence of AS occurrence types

So far, we have assumed that occurrence types were independent. However, as discussed in Section 2.2, it is reasonable to assume that some of them might be related (Fig. 2.5). Although there are other variants, a relevant representation for dependent occurrences would be as in Figure 2.8.

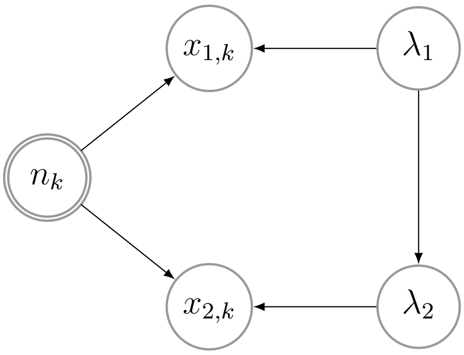


Figure 2.8: Influence diagram for the dependence effect between occurrences.

If the relation between λ_1 and λ_2 is assumed to be linear, a relevant model would be

$$
\begin{aligned}
x_{1,k}|\lambda_1, n_k \sim Po(\lambda_1 n_k), &\qquad x_{2,k}|\lambda_2, n_k \sim Po(\lambda_2 n_k),\\
\lambda_1 \sim Ga(r,p), &\qquad \lambda_2 = a\lambda_1 + b + \epsilon, \quad \epsilon \sim N(0,\sigma^2),\\
a \sim N(\mu_a, \sigma_a^2), \quad b \sim N(\mu_b, \sigma_b^2), &\qquad \sigma^2 \sim \textit{Inv-Gamma}(\alpha, \beta).
\end{aligned}
$$

Given data $D_k = \{(x_{1,i}, x_{2,i}, n_i)\}_{i=1}^k$, the joint posterior would be

$$
\begin{aligned}
\pi(\lambda_1, \lambda_2, a, b, \sigma^2|D_k) \propto \lambda_1^{r-1+\sum_{i=1}^k x_{1,i}} \lambda_2^{\sum_{i=1}^k x_{2,i}} \sigma^{-2\alpha-3} \exp\bigg(&- (\lambda_1+\lambda_2)\sum_{i=1}^k n_i - \\
&\frac{(\lambda_2 - a\lambda_1 - b)^2}{2\sigma^2} - p\lambda_1 - \frac{(a-\mu_a)^2}{2\sigma_a^2} - \frac{(b-\mu_b)^2}{2\sigma_b^2} - \frac{\beta}{\alpha^2}\bigg).
\end{aligned}
$$

The conditional posterior distributions are

$$
\begin{aligned}
\pi(\lambda_1|\lambda_2, a, b, \sigma^2, D_k) &\propto \lambda_1^{r-1+\sum x_{1,i}} \exp\Big(-\lambda_1\Big(p + \sum n_i + \frac{(\lambda_2 - a\lambda_1 - b)^2}{2\sigma^2}\Big)\Big),\\
\pi(\lambda_2|\lambda_1, a, b, \sigma^2, D_k) &\propto \lambda_2^{\sum x_{2,i}} \exp\Big(-\lambda_2\Big(\sum n_i + \frac{(\lambda_2 - a\lambda_1 - b)^2}{2\sigma^2}\Big)\Big),\\
\pi(a|\lambda_1, \lambda_2, b, \sigma^2, D_k) &\sim N\left(a \,\Big|\, \frac{\sigma^2\mu_a + \lambda_1\sigma_a^2(\lambda_2 - b)}{\sigma^2 + \lambda_1^2\sigma_a^2}, \frac{\sigma^2}{\lambda_1^2 + \sigma^2/\sigma_a^2}\right),\\
\pi(b|\lambda_1, \lambda_2, a, \sigma^2, D_k) &\sim N\left(b \,\Big|\, \frac{\sigma^2\mu_b + \sigma_b^2(\lambda_2 - a\lambda_1)}{\sigma^2 + \sigma_b^2}, \frac{1}{1/\sigma^2 + 1/\sigma_b^2}\right),\\
\pi(\sigma^2|\lambda_1, \lambda_2, a, b, D_k) &\sim \textit{Inv-Gamma}\left(\sigma^2 \,\Big|\, \alpha + \frac{1}{2}, \beta + \frac{(\lambda_2 - a\lambda_1 - b)^2}{2}\right).
\end{aligned}
$$

Based on them, we build a hybrid MCMC sampler to generate from the posterior as in Algorithm 2.4, with Gibbs steps for σ^2, β and α; and Metropolis-Hastings steps for λ_1 and λ_2.

```
Set $\lambda_{1,0}, \lambda_{2,0}, \alpha_0, \beta_0, \sigma_0^2, j = 1$;
while convergence not detected do
    Sample $\lambda^*_{1,j} \sim q_1(\lambda_{1,j-1})$;
    Calculate $\psi_1 = \min\left(1, \frac{f_1(\lambda^*_{1,j})}{f_1(\lambda_{1,j-1})} \frac{q_1(\lambda_{1,j-1})}{q_1(\lambda^*_{1,j})}\right)$;
    Do $\lambda_{1,j} = \begin{cases} \lambda^*_{1,j} & \text{with probability } \psi_1 \\ \lambda_{1,j-1} & \text{with probability } (1-\psi_1) \end{cases}$;
    Sample $\sigma_j^2 \sim \textit{Inv-Gamma}\left(\frac{1}{2}, \frac{(\lambda_{2,j-1} - \alpha_{j-1}\lambda_{1,j} - \beta_{j-1})^2}{2}\right)$;
    Sample $\beta_j \sim N(\lambda_{2,j-1} - \lambda_{1,j}\alpha_{j-1}, \sigma_j^2)$;
    Sample $\alpha_j \sim N\left(\frac{\lambda_{2,j-1} - \beta_j}{\lambda_{1,j}}, \frac{\sigma_j^2}{\lambda_{1,j}^2}\right)$;
    Sample $\lambda^*_{2,j} \sim q_2(\lambda_{2,j-1})$;
    Calculate $\psi_2 = \min\left(1, \frac{f_2(\lambda^*_{2,j})}{f_2(\lambda_{1,j-1})} \frac{q_2(\lambda_{2,j-1})}{q_1(\lambda^*_{2,j})}\right)$;
    Do $\lambda_{2,j} = \begin{cases} \lambda^*_{2,j} & \text{with probability } \psi_2 \\ \lambda_{2,j-1} & \text{with probability } (1-\psi_2) \end{cases}$;
    $j \leftarrow j + 1$;
end
```

Algorithm 2.4: MCMC sampler for *Dependence of occurrences* model.

Similarly to the stress effect model in Section 2.3.2, non-negativity of λ_1 and λ_2 is guaranteed by using the recommended candidate generating distributions: $Ga(r+\sum x_{1,i}, p+\sum n_i+(\lambda_2-a\lambda_1-b)^2/(2\sigma^2))$ for λ_1, and $Ga(1+\sum x_{2,i}, \sum n_i+(\lambda_2 - a\lambda_1 - b)^2/(2\sigma^2)$ for λ_2. We denote by $f_1(\lambda_1)$ and $f_2(\lambda_2)$, the expressions proportional to the posteriors of λ_1 and λ_2 respectively.

As a final comment, note that should the rate of one of the occurrences be higher than that of the other, which does not hold in our domain, we could alternatively consider the use of McKay's bivariate gamma model (McKay, 1934).

2.4 Forecasting AS occurrences with an uncertain number of operations

The number n_k of operations in the k-th period was assumed known in previous models. This may be realistic for short term forecasts in which there is little uncertainty about the number of operations to be held. On the other hand, for long horizons, e.g. in annual operational planning, there is uncertainty about such quantities, which should be taken into account so as to improve occurrence forecasting. Consider thus the case in which the number of operations is uncertain and both the occurrence rate and such number evolve according to DLMs. The corresponding influence diagram is reflected in Figure 2.9.

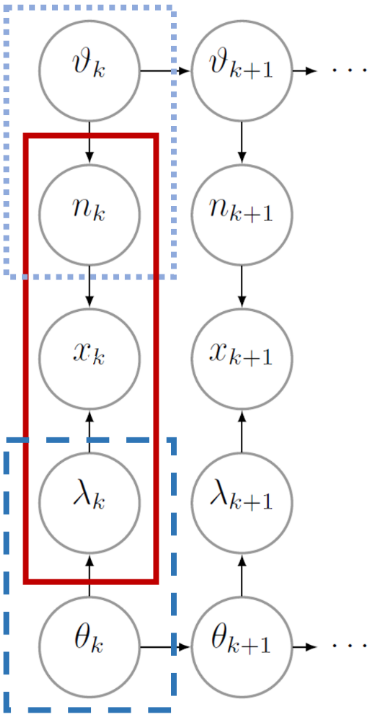


Figure 2.9: Predicting occurrences with uncertain number of operations. Occurrence rate DLM, dashed; operations DLM, dotted; Poisson, solid.

The resulting model would be

$$\begin{cases}\begin{cases}\begin{cases} n_k = \boldsymbol{H}_k\boldsymbol{\vartheta}_k + z_k, & z_k \sim N(0,\Sigma_k)\\ \boldsymbol{\vartheta}_k = \boldsymbol{J}_k\boldsymbol{\vartheta}_{i-1}+\boldsymbol{\xi}_k, & \boldsymbol{\xi}_k\sim N(\mathbf{0},\boldsymbol{S}_k)\end{cases}\\ \boldsymbol{\vartheta}_0\sim N(\boldsymbol{\eta}_0,\boldsymbol{S}_0)\end{cases}\\ x_k|\lambda_k,n_k\sim Po(\lambda_k n_k), \quad \lambda_k=\exp(u_k)\\ \begin{cases}\begin{cases} u_k = \boldsymbol{F}_k\boldsymbol{\theta}_k+v_k, & v_k\sim N(0,V_k)\\ \boldsymbol{\theta}_k=\boldsymbol{G}_k\boldsymbol{\theta}_{k-1}+\boldsymbol{w}_k, & \boldsymbol{w}_k\sim N(\mathbf{0},\boldsymbol{W}_k)\end{cases}\\ \boldsymbol{\theta}_0\sim N(\boldsymbol{m}_0,\boldsymbol{C}_0),\end{cases}\end{cases}$$

where, in addition to the features in model (2.2), $\boldsymbol{\vartheta}_k$ are the state variables for the number of operations; $\boldsymbol{H}_k$ and $\boldsymbol{J}_k$ are the regression vector and evolution matrix of the operations DLM; and, finally, z_k, ξ_k would be independent sequences of normal variables (independent of v_k and w_k) with zero mean and variances Σ_k and $\boldsymbol{S}_k$, respectively. Contrary to λ_k, the number of occurrences n_k is modeled directly with a DLM, and therefore some probability is assigned to non positive values; however, since the minimum aggregation level we are interested in is the number of operations in an airport during a month, which is consistently in the tens of thousands, this is not significant and allows to use the straightforward sequential updating of DLMs for the number of operations. The prediction procedure at the k-th step would thus be:

Step 0. Prediction of x_k and n_k at period k. We have distributions $\pi(\boldsymbol{\theta}_k)$, $\pi(u_k|\boldsymbol{\theta}_k)$, $\pi(x_k|u_k,n_k)$, $\pi(n_k|\boldsymbol{\vartheta}_k)$, $\pi(\boldsymbol{\vartheta}_k)$, and the relation $\lambda_k=\exp(u_k)$ (again, it could be the case that some of these distributions are expressed through samples; in particular, that of $\pi(\boldsymbol{\theta}_k)$ would be given by a sample $\{\boldsymbol{\theta}_k^i\}_{i=1}^N$, with weights $\pi_k^i \geq 0$, $\sum \pi_k^i = 1$). To predict n_k, use the predictive distribution $\pi(n_k) = \int \pi(n_k|\boldsymbol{\vartheta}_k)\pi(\boldsymbol{\vartheta}_k)\, d\boldsymbol{\vartheta}_k$, based on the DLM predictive formulae, Appendix B, having a normal distribution $n_k \sim N(f_k, Q_k)$. To predict x_k, use

$$\begin{aligned}\pi(x_k) &= \int\!\!\int\!\!\int\!\!\int\!\!\int \pi(x_k|\lambda_k, n_k)\pi(n_k|\boldsymbol{\vartheta}_k)\pi(\boldsymbol{\vartheta}_k)\pi(\lambda_k|u_k)\pi(u_k|\boldsymbol{\theta}_k)\pi(\boldsymbol{\theta}_k)\ du_k\ d\lambda_k\ dn_k\ d\boldsymbol{\vartheta}_k\ d\boldsymbol{\theta}_k \\ &= \iiint \pi(x_k|\exp(u_k), n_k)\pi(n_k)\pi(u_k|\boldsymbol{\theta}_k)\pi(\boldsymbol{\theta}_k)\ dn_k\ du_k\ d\boldsymbol{\theta}_k.\end{aligned}$$

We simulate it as follows:

Sample $\{\boldsymbol{\theta}_k^i\}_{i=1}^N \sim \pi(\boldsymbol{\theta}_k)$ (possibly already available);
Do $\lambda_k^i = \exp(\boldsymbol{F}_k\boldsymbol{\theta}_k^i)$, for $i = 1, \ldots, N$;
Sample $\{n_k^i\}_{i=1}^N \sim N(f_k, Q_k)$;
Approximate $\pi(x_k) \approx \frac{1}{N x_k!}\sum_{i=1}^N \exp(-\lambda_k^i n_k^i)(\lambda_k^i n_k^i)^{x_k}$.

The approximate predictive mean and second moment are

$$\begin{aligned} E(X_k) &\approx \frac{f_k}{N}\sum_{i=1}^N \exp\left(\frac{1-2\boldsymbol{F}_k\boldsymbol{\theta}_k^i}{2V_k}\right), \\ E(X_k^2) &\approx \frac{f_k}{N}\sum_{i=1}^N \exp\left(\frac{1-2\boldsymbol{F}_k\boldsymbol{\theta}_k^i}{2V_k}\right) + \frac{f_k^2+Q_k^2}{N}\sum_{i=1}^N \exp\left(\frac{2-2\boldsymbol{F}_k\boldsymbol{\theta}_k^i}{V_k}\right).\end{aligned}$$

Then, the predictive variance would be approximated by

$$\frac{f_k}{N}\sum_{i=1}^N \exp\left(\frac{1-2\boldsymbol{F}_k\boldsymbol{\theta}_k^i}{2V_k}\right) + \frac{f_k^2+Q_k^2}{N}\sum_{i=1}^N \exp\left(\frac{2-2\boldsymbol{F}_k\boldsymbol{\theta}_k^i}{V_k}\right) - \frac{f_k^2}{N^2}\left(\sum_{i=1}^N \exp\left(\frac{1-2\boldsymbol{F}_k\boldsymbol{\theta}_k^i}{2V_k}\right)\right)^2.$$

Step 1. Observation of (x_k, n_k) and update. At the end of the k-th period, observe x_k, n_k and propagate this information to obtain $\pi(\boldsymbol{\theta}_k|x_k, n_k)$ and $\pi(\boldsymbol{\vartheta}_k|n_k)$. First, invert the relation $x \to \lambda$. The new distribution at node x is $\pi(x_k|n_k, \boldsymbol{\theta}_k) = \int \pi(x_k|n_k, \lambda_k)\pi(\lambda_k|\boldsymbol{\theta}_k)\ d\lambda_k$. The posterior for λ_k is

$$\pi(\lambda_k|x_k, n_k, \boldsymbol{\theta}_k) = \frac{\pi(\lambda_k|\boldsymbol{\theta}_k)\pi(x_k|\lambda_k, n_k)}{\pi(x_k|n_k, \boldsymbol{\theta}_k)} \propto \pi(\lambda_k|\boldsymbol{\theta}_k)\pi(x_k|\lambda_k, n_k).$$

Propagate now the evidence of x_k and n_k resulting in

$$\pi(\boldsymbol{\theta}_k|x_k, n_k) \;=\; \int \pi(\lambda_k|x_k, n_k, \boldsymbol{\theta}_k)\pi(\boldsymbol{\theta}_k)\ d\lambda_k.$$

It can be approximated by

$$\pi(\boldsymbol{\theta}_k|x_k, n_k) \;\approx\; \frac{1}{N}\sum_{i=1}^{N}\pi(\lambda_k|x_k, n_k, \boldsymbol{\theta}_k^i) = \frac{1}{N}\sum_{i=1}^{N}\frac{\pi(\lambda_k|\boldsymbol{\theta}_k^i)\pi(x_k|\lambda_k, n_k)}{\int \pi(x_k|n_k, \lambda_k)\pi(\lambda_k|\boldsymbol{\theta}_k^i)\ d\lambda}.$$

The propagation of evidence n_k to $\boldsymbol{\vartheta}_k$ is done through

$$\pi(\boldsymbol{\vartheta}_k|n_k) \;=\; \frac{\pi(n_k|\boldsymbol{\vartheta}_k)\pi(\boldsymbol{\vartheta}_k)}{\pi(n_k)},$$

with $\pi(n_k) = \int \pi(n_k|\boldsymbol{\vartheta}_k)\pi(\boldsymbol{\vartheta}_k)\ d\boldsymbol{\vartheta}_k$. Thereupon, the DLM equations for sequential updating in Appendix B are used.

Step 2. Propagation to period $k+1$. The distribution of the d-dimensional state vector, $\pi(\boldsymbol{\theta}_{k+1}|D_k) = \int \pi(\boldsymbol{\theta}_{k+1}|\boldsymbol{\theta}_k)\pi(\boldsymbol{\theta}_k|x_k, n_k)\ d\boldsymbol{\theta}_k$, gets approximated through

$$\pi(\boldsymbol{\theta}_{k+1}|D_k) \approx \frac{1}{N}\sum_{i=1}^{N}\pi(\boldsymbol{\theta}_{k+1}|\boldsymbol{\theta}_k^i) = \frac{1}{N}\sum_{i=1}^{N}\frac{\exp\big(-\frac{1}{2}(\boldsymbol{\theta}_{k+1} - \boldsymbol{G}_k\boldsymbol{\theta}_k^i)'\boldsymbol{W}_k^{-1}(\boldsymbol{\theta}_{k+1} - \boldsymbol{G}\boldsymbol{\theta}_k^i)\big)}{\sqrt{(2\pi)^d|\boldsymbol{W}_k|}},$$

where $\{\boldsymbol{\theta}_k^i\}_{i=1}^N$ is a sample of $\boldsymbol{\theta}_k|x_k, n_k$ from step 1. The one step ahead predictive distribution for state $\boldsymbol{\vartheta}_k$ is obtained with the DLM equations in Appendix B.

After that, we would be back at step 0, re-starting the process. The above can be grouped into a scheme similar to Algorithm 2.2.

2.5 Forecasting severities

We also need to predict how many of the x_k occurrences in the k-th period correspond to the five severity classes. Let $\boldsymbol{p} = (p_1, p_2, p_3, p_4, p_5)$ be a vector representing their proportions, with $p_i \geq 0$, $\sum_{i=1}^{5} p_i = 1$; $\boldsymbol{s} = (s_1, s_2, s_3, s_4, s_5)$ be the vector with the number of occurrences of each severity, with $s_i \geq 0$ and $\sum_{i=1}^{5} s_i = x_k$; and $D_{k-1} = \{(s_1^j, s_2^j, s_3^j, s_4^j, s_5^j)\}_{j=1}^{k-1}$ the data at the beginning of the k-th period, where s_i^j is the number of occurrences of severity i in period j.

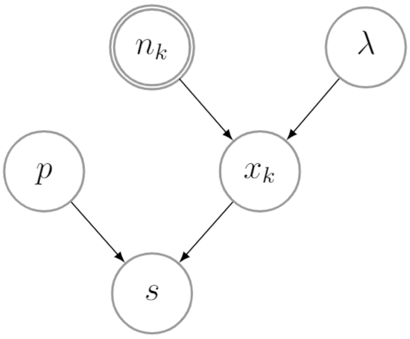


Figure 2.10: Influence diagram to forecast aviation occurrences severity.

In our problem, the number x_k of occurrences in the k-th period is unknown, and predicted as in Sections 2.3 and 2.4. For example, if we consider the initial basic model for x_k and a Multinomial-Dirichlet model for the severity, Figure 2.10, we have

$$\begin{aligned} x_k \sim Po(\lambda n_k), &\qquad \lambda \sim Ga(a, b), \\ \boldsymbol{s}|\boldsymbol{p}, x_k &\sim \mathcal{M}(x_k; p_1, p_2, p_3, p_4, p_5), \\ \boldsymbol{p} &\sim \mathcal{D}ir(\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5). \end{aligned}$$

The predictions would be

$$\begin{aligned} Pr(s_i|D_{k-1}) &= \sum_{r=s_i}^{\infty} \binom{r}{s_i} \frac{\mathrm{B}(s_i + \alpha_i', A + r - s_i - \alpha_i')}{\mathrm{B}(\alpha_i', A - \alpha_i')} \frac{b_k^{a_k}}{(b_k+1)^{a_k+r}} \frac{\Gamma(a_k + r)}{r!\,\Gamma(a_k)} \\ &\approx \sum_{r=s_i}^{h} \binom{r}{s_i} \frac{\mathrm{B}(s_i + \alpha_i', A + r - s_i - \alpha_i')}{\mathrm{B}(\alpha_i', A - \alpha_i')} \frac{b_k^{a_k}}{(b_k+1)^{a_k+r}} \frac{\Gamma(a_k + r)}{r!\,\Gamma(a_k)}, \end{aligned}$$

for big enough h, with $\alpha_i' = \alpha_i + \sum_{j=1}^{k-1} s_i^j$, $A = \sum_{i=1}^{5} \alpha_i'$, $a_k = a + \sum_{j=1}^{k-1} x_j$, and $b_k = b + \sum_{j=1}^{k-1} n_j$. The predictive expected number of occurrences is $E(s_i|D_{k-1}) = E(x_k)E(p_i)$. As an example, with the basic model, Section 2.3.1, $E(s_i|D_{k-1}) = (x_k a_k \alpha_i')/(b_k A)$, and the predictive variance is $E(p_i(1-p_i))E(x_k^2) + (E(p_i))^2 Var(x_k)$. Notice that although shown here in combination with the most basic model, extensions to other cases follow a similar path, as illustrated e.g. in Section 2.6.1.

2.5.1 The problem of underreporting

A major obstacle to forecast occurrences, would be the unavailability of accurate data, which could hide latent conditions that end up causing more severe ones. Indeed, in the absence of a strong reporting culture among the agents involved (pilots, air controllers,...) it would be common not to report low severity occurrences (Haslbeck et al., 2015).

As an example, Figure 2.11 displays the occurrence rate for *animal runway incursion*, which appears to have increased significantly from 2010 to 2016. If we do not detect a technical or socio-economic explanation of this rising rate, this is likely a case of underreporting in the early years. In this particular case, recall however that the 2008 financial crisis, among many other effects, reduced the possibilities of maintaining airport fences, with the consequent increase in the entry of animals into airports, which might have eventually increased runway incursions.

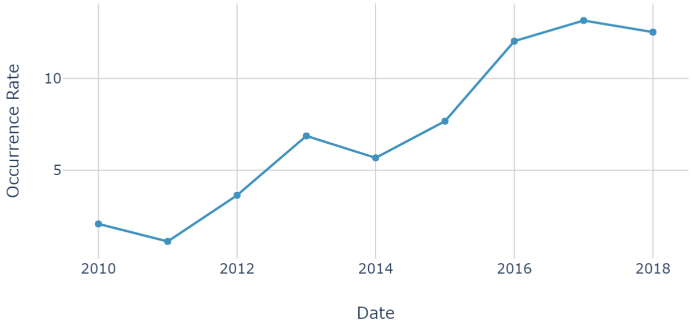


Figure 2.11: Possible underreporting in *animal runway incursion* occurrences.

In a case with suspected underreporting, we can apply the logic of the influence diagram in Figure 2.12, based on our basic model from Section 2.3.1. Again, extensions to the other models can be developed. Introduce the vector of reported occurrences for each severity class, $\boldsymbol{z} = (z_1, z_2, z_3, z_4, z_5)$, and the vector with the proportion of reported occurrences $\boldsymbol{\varrho} = (\varrho_1, \varrho_2, \varrho_3, \varrho_4, \varrho_5)$. For example, ϱ_3=0.75 would mean that 75% of severity class 3 occurrences are reported (and 25% are not).

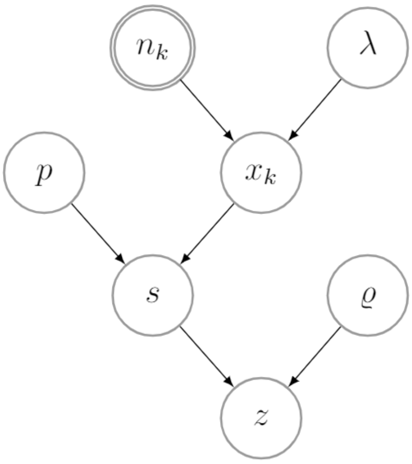


Figure 2.12: Model for the underreporting problem.

Partial information about ϱ_i is typically available, in particular, $\varrho_1 = 1$, $\varrho_2 \approx 1$ and $\varrho_2 > \varrho_3 > \varrho_4 > \varrho_5$, i.e. as occurrences get less severe they are less likely to be reported, which would represent the fact that, as occurrences get less severe, they are less likely to be reported. Thus, additional features to model in Figure 2.10, are $\boldsymbol{\varrho}$, with $\varrho_i \sim Be(\gamma_i, \beta_i)$, $i = 1, \ldots, 5$; and $\boldsymbol{z}$, with $z_i | s_i, \varrho_i \sim Bin(s_i, \varrho_i)$, $i = 1, \ldots, 5$.

At the beginning of each period, our goal is to predict $\boldsymbol{s}$, x and $\boldsymbol{z}$; then, after observing $\boldsymbol{z}$, to propagate that information to the different levels of the graph in Figure 2.12. To simplify the problem, assume $\boldsymbol{p}$ known[1]. Due to the Poisson process partition property, $s_i | \lambda \sim Po(p_i \lambda)$ (Rios Insua et al., 2012).

---

[1] Extensions to unknown p_i's follow in a straightforward manner.

Furthermore,

$$Pr(Z_i = z_i|\lambda, \varrho_i) = \sum_{s_i=z_i}^{\infty} Pr(Z_i = z_i|s_i, \varrho_i)Pr(S_i = s_i|\lambda) = \frac{(p_i\lambda\varrho_i)^{z_i}}{z_i!}\exp(-p_i\lambda\varrho_i).$$

Hence, $z_i|\lambda, \varrho_i \sim Po(p_i\lambda\varrho_i)$, and the likelihood of the observed data $\boldsymbol{z}$ is

$$\pi(\boldsymbol{z}|\lambda, \boldsymbol{\varrho}) \propto \lambda^{\sum z_i}\exp\left(-\lambda\sum_{i=1}^{5} p_i\varrho_i\right)\prod_{i=1}^{5}\varrho_i^{z_i}.$$

Assuming all parameters independent, the posterior distribution would be

$$\pi(\lambda, \boldsymbol{\varrho}|\boldsymbol{z}) \propto \pi(\lambda)\lambda^{\sum z_i}\exp\left(-\lambda\sum_{i=1}^{5} p_i\varrho_i\right)\prod_{i=1}^{5}\varrho_i^{z_i}\pi(\varrho_i).$$

Under the non-informative prior $\pi(\lambda) \propto \lambda^{-1}$, the posterior is

$$\pi(\lambda, \boldsymbol{\varrho}|\boldsymbol{z}) \propto \lambda^{\sum z_i - 1}\exp\left(-\lambda\sum_{i=1}^{5} p_i\varrho_i\right)\prod_{i=1}^{5}\varrho_i^{z_i+\gamma_i-1}(1-\varrho_i)^{\beta_i-1},$$

and the posterior conditionals are

$$\begin{aligned}\pi(\lambda|\boldsymbol{\varrho}, \boldsymbol{z}) &\propto \lambda^{\sum z_i-1}\exp(-\lambda\sum_{i=1}^{5} p_i\varrho_i) \sim Ga\left(\sum_{i=1}^{5} p_i\varrho_i, \sum_{i=1}^{5} z_i\right)\\ \pi(\varrho_i|\lambda, \boldsymbol{z}) &\propto \varrho_i^{z_i+\gamma_i-1}(1-\varrho_i)^{\beta_i-1}\exp(-\lambda p_i\varrho_i), \quad i = 1, \ldots, 5.\end{aligned}$$

The last distributions are not standard, but are unimodal and log-concave, so it is easy to sample from them as in Algorithm 2.5, obtaining samples $\{(\lambda^j, \varrho_1^j, \ldots, \varrho_5^j)\}_{j=1}^{N}$.

```
Set $\lambda^0, \varrho_1^0, \ldots, \varrho_5^0, j = 1.$;
while convergence not detected do
    Sample $\lambda^j \sim Ga(\sum_{i=1}^5 p_i\varrho_i^{j-1}, \sum_{i=1}^5 z_i)$;
    Sample $\varrho_i^j \propto (\varrho_i^j)^{z_i+\alpha_i-1}(1-\varrho_i^j)^{\beta_i-1}\exp(-\lambda^j p_i\varrho_i^j), \quad i = 1, \ldots, 5$;
    $j \leftarrow j + 1$;
end
```

Algorithm 2.5: Sampler for *Underreporting model*.

Our interest lies in sampling from the distributions $s_i|\boldsymbol{z}$. Observe that

$$Pr(s_i|\boldsymbol{z}) = \iint Pr(s_i|\lambda, \boldsymbol{\varrho}, \boldsymbol{z})\pi(\lambda, \boldsymbol{\varrho}|z)\, d\lambda\, d\boldsymbol{\varrho}.$$

Then $s_i|\lambda, \boldsymbol{\varrho}, \boldsymbol{z} \sim z_i + Po(p_i\lambda(1-\varrho_i))$, and $s_i^j \sim z_i + Po(p_i\lambda^j(1-\varrho_i^j))$ constitutes a sample from $s_i|\boldsymbol{z}$. We summarize it with $\frac{1}{N}\sum_{j=1}^{N} s_i^j$ which approximates $E(s_i|\boldsymbol{z})$.

2.6 Cases

As application examples, we present the models used for the *wind shear* and *TCAS warning* occurrences, and to a simulated occurrence type showing dependence. More emphasis is placed on the *wind shear* model, because it is the most versatile in general, and the most used in this particular application domain of AS. Core ideas are given for the other two cases.

2.6.1 Wind Shear

Wind shear consists of a change in wind speed and/or direction over a short distance (FAA, 2008). It can occur either horizontally or vertically, at high or low altitude, most often associated with strong temperature inversions or density gradients. It may significantly affect the airspeed and trajectory of a plane, being more dangerous the closer to the ground and the slower the aircraft is. Therefore, AS occurrences reported in relation to wind shear usually happen during take-off or landing.

Exploratory analysis. Table 2.2 displays the evolution of the number of occurrences from 2010 to 2018, the number of operations (in blocks of 100,000), the occurrence rate (number of occurrences per 100,000 operations) as well as the evolution for the five severities. As we see, the occurrence rate has been growing annually, especially during the first five years, then stabilizing. Note

	Sev. 1	Sev. 2	Sev. 3	Sev. 4	Sev. 5	Total Occ.	Ops.	Occ. rate
2010	0	0	5	113	9	127	21.20	5.99
2011	1	0	0	91	5	97	21.40	4.53
2012	0	0	4	160	23	187	19.25	9.71
2013	0	1	2	265	7	275	17.91	15.35
2014	0	1	5	357	46	409	18.33	22.31
2015	0	0	1	385	24	410	19.03	21.55
2016	0	0	2	474	13	489	20.45	23.91
2017	0	0	2	511	10	523	21.74	24.05
2018	0	0	2	518	8	528	23.00	22.95

Table 2.2: Number of occurrences and operations for *wind shear*, 2010-2018.

that in 2011 there were 23% less occurrences compared to the previous year, while the number of operations increased slightly.

Regarding occurrence severity, Figure 2.13, observe that every year, severity 4 occurrences were the most reported, followed by those of severity 5. Finally, note that there has been only two severity 2 occurrences, and one severity 1 during the considered period, suggesting that this event is not very severe, impact-wise.

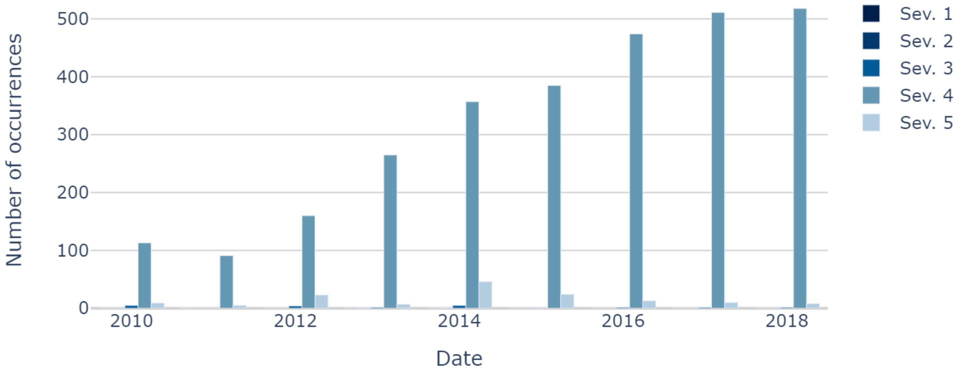


Figure 2.13: *Wind shear* occurrences, period 2010-2018, by severity.

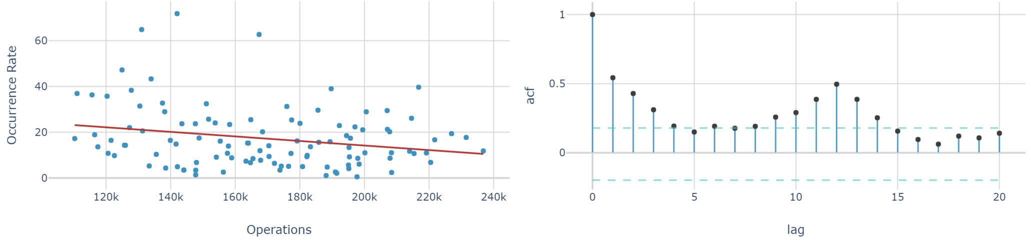
Effects. Graphical and numerical analyses used to identify the relevant effects follow.

Stress effect. Figure 2.14(a) shows the scatter plot for the number of operations versus occurrence rates, as well as the regression line relating both variables. The correlation coefficient is -0.23 and no *stress effect* is included.

Seasonal effect. The monthly ACF is in Figure 2.14(b) suggests a seasonal effect, through the relevance of the lag 12 autocorrelation, due to weather relevance over this phenomenon. In addition, the first ones, although not strong, are relevant, suggesting a relationship between rates at consecutive months.

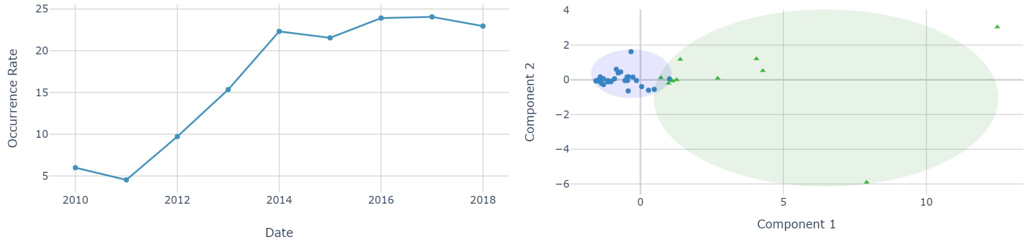
Linear effect. Figure 2.14(c) represents the annual evolution of occurrence rates. The annual time series suggests a linear increase of wind shear occurrence rates during the first five years. The effect is considerable because, except for year 2011, in which it was slightly less than the previous one, the rate has grown annually.

Group effect. A cluster analysis allows us to identify two groups of airports with similar *wind shear* occurrence rate, Figure 2.14(d). The first one (triangles) includes the ten airports in temperate coastal areas. The second group (circles) would be formed by the remaining airports. Because of the climate differences, we deal with both groups hierarchically and aggregate the forecasts. Note that this would also allow for a certain *lag* between the seasonalities in groups, which for example would be relevant when dealing with occurrences related to migratory birds that arrive at airports at different times of the year.



(a) Stress effect

(b) Seasonal effect



(c) Linear effect

(d) Group effect

Figure 2.14: Effect analysis for *wind shear*.

Model. We thus have detected a seasonal effect of period 12, a (possible) linear growth effect and two groups of airports. Hence, we consider a hierarchical model for the occurrences x_k^i at group i of airports, based on the Poisson DLM (2.2) and the hierarchical model (2.4) of Section 2.3.2, with a linear growth component, a seasonal component of period 12, and a common prior,

$$\begin{aligned} x_k^i|\lambda_k^i, n_k^i &\sim Po(\lambda_k^i n_k^i), \quad \lambda_k^i = \exp(u_k^i), \qquad i = 1, 2 \\ u_k^i &= \boldsymbol{F}\boldsymbol{\theta}_k^i + v^i, \quad v^i \sim N(0, V^i), \\ \boldsymbol{\theta}_k^i &= \boldsymbol{G}\boldsymbol{\theta}_{k-1}^i + \boldsymbol{w}^i, \quad \boldsymbol{w}^i \sim N(\boldsymbol{0}, \boldsymbol{W}^i), \\ \boldsymbol{\theta}_0^i &\sim N(\boldsymbol{m}_0, \boldsymbol{C}_0), \end{aligned} \tag{2.5}$$

where $\boldsymbol{F} = \big(\boldsymbol{F}_1 \;\; \boldsymbol{F}_2\big)$ and $\boldsymbol{G} = \text{blockdiag}\big(\boldsymbol{G}_1 \;\; \boldsymbol{G}_2\big)$. Matrices $\boldsymbol{W}^i$ and V^i are initialised based on the observations, using maximum likelihood (Petris et al., 2009).

To complete model specification, we need the prior moments $\boldsymbol{m}_0$ and $\boldsymbol{C}_0$. These priors aim to be flexible enough and somewhat informative. The prior mean vector, $\boldsymbol{m}_0$, is initialized[2] using the the first year of data (12 observations). Parameter m_0^{L2} describes the expected growth and it is initialised with $m_0^{L2} = (y_{12} - y_1)/11$, where $y_k = \log(\sum x_k^i / \sum n_k^i)$; m_0^{L1} describes the expected level and is initialised with $m_0^{L1} = (\sum_{k=1}^{12} y_k - 78\, m_0^{L2})/12$; m_0^{Sj} describes the j-th seasonal component and, to assess it, we use $m_0^{Sj} = y_{13-j} - m_0^{L1} + (j-13)\, m_0^{L2}$. Hence, for both groups of airports, we have

$$\boldsymbol{m}_0 = (\overbrace{1.6}^{m_0^{L1}}, \overbrace{0.0}^{m_0^{L2}}, \overbrace{-0.3}^{m_0^{S1}}, \overbrace{-0.7}^{m_0^{S2}}, \overbrace{0.0}^{m_0^{S3}}, \overbrace{-0.9}^{m_0^{S4}}, \overbrace{0.6}^{m_0^{S5}}, \overbrace{0.8}^{m_0^{S6}}, \overbrace{0.1}^{m_0^{S7}}, \overbrace{-0.6}^{m_0^{S8}}, \overbrace{-0.4}^{m_0^{S9}}, \overbrace{0.6}^{m_0^{S10}}, \overbrace{1.1}^{m_0^{S11}})'$$

For more accuracy, we could repeat the calculations as many times as years of data are available, and take the average value for each parameter. Since we are confident about the suitability of our $\boldsymbol{m}_0$, we use a relatively small prior variance $\boldsymbol{C}_0 = \boldsymbol{I}_{13}/10$. Also, for both groups of airports, we include the prior parameters α_j ($j = 1, \ldots, 5$) for the different severities and, based on expert judgement, set at 1, 2, 3, 7 and 5, respectively. They are chosen not very high to facilitate learning.

We then adjust the previous models using approximations analogous to those described in Section 2.3.2 for Algorithm 2.2, which in turn result in Algorithm 2.6. Since there are two groups of airports, $L = 2$, and we obtain two samples, one for the predictive distribution of each group of airports. The aggregation of both samples facilitates a predictive sample for the total number of wind shear occurrences.

---

[2] m_0^{Lj} and m_0^{Sj} indicate the j-th parameters of the linear growth and seasonal blocks respectively.

```
Sample $\{\boldsymbol{\theta}_0^{i,j}\}_{j=1}^N \sim N(\boldsymbol{m}_0, \boldsymbol{C}_0), \quad i = 1, \ldots, L;$
Do $\pi_0^{i,j} = \frac{1}{N}, \quad j = 1, \ldots, N, \quad i = 1, \ldots, L;$
for $k \leftarrow 1$ to $T$ do
    for $i \leftarrow 1$ to $L$ do
        for $j \leftarrow 1$ to $N$ do
            Sample $\boldsymbol{\theta}_k^{i,j} \sim N(\boldsymbol{G}_k \boldsymbol{\theta}_{k-1}^{i,j}, \boldsymbol{W}_k^i);$
            Do $\eta_{k,i} = \frac{n_k^i}{N} \sum_{j=1}^{N} \exp\left(\frac{1 - 2\boldsymbol{F}_k \boldsymbol{\theta}_k^{i,j}}{2V_k^i}\right);$
            Do
              $\kappa_{k,i}^2 = \frac{n_k^i}{N} \sum_{j=1}^N \exp\left(\frac{1-2\boldsymbol{F}_k\boldsymbol{\theta}_k^{i,j}}{2V_k^i}\right) + \frac{{n_k^i}^2}{N} \sum_{j=1}^N \exp\left(\frac{2-2\boldsymbol{F}_k\boldsymbol{\theta}_k^{i,j}}{V_k^i}\right);$
                  $-\frac{{n_k^i}^2}{N^2} \left(\sum_{j=1}^N \exp\left(\frac{1-2\boldsymbol{F}_k\boldsymbol{\theta}_k^{i,j}}{2V_k^i}\right)\right)^2;$
            Read $x_k^i;$
            $\Delta_j = 0;$
            for $h \leftarrow 1$ to $N$ do
                Sample $u_k^h \sim N(\boldsymbol{F}_k \boldsymbol{\theta}_k^{i,j}, V_k^i);$
                Do $\lambda_k^h = \exp(u_k^h);$
                $\Delta_j \leftarrow \Delta_j + exp(-\lambda_k^h n_k^i)(\lambda_k^h)^{x_k^i}$
            end
            $\pi_k^{i,j} \leftarrow \pi_{k-1}^{i,j} \Delta_j;$
        end
        $\pi_k^{i,j} = \frac{\pi_k^{i,j}}{\sum_{j=1}^N \pi_k^{i,j}};$
        Calculate $N_{ESS} = (\sum_{j=1}^N (\pi_k^{i,j})^2)^{-1};$
        if $N_{ESS} < N/2$ then
            Sample $\boldsymbol{\theta}_k^{i,j*} \sim \{\boldsymbol{\theta}_k^{i,h}, \pi_k^{i,h}\}_{h=1}^N, \quad j = 1, \ldots, N;$
            $\boldsymbol{\theta}_k^{i,j} \leftarrow \boldsymbol{\theta}_k^{i,j*};$
            $\pi_k^{i,j} \leftarrow \frac{1}{N};$
        end
    end
end
```

Algorithm 2.6: Particle filter for hierarchical model (2.5).

Figure 2.15 shows one-month ahead predictions for 2010-2018 observations (black dots), the predictive mean (solid line) and the 90% probability band (dashed lines). We also show the 12 step ahead forecast for year 2019 (without actual observations, since they were not available), for which the uncertainty in the future number of operations is modeled as in Section 2.4.

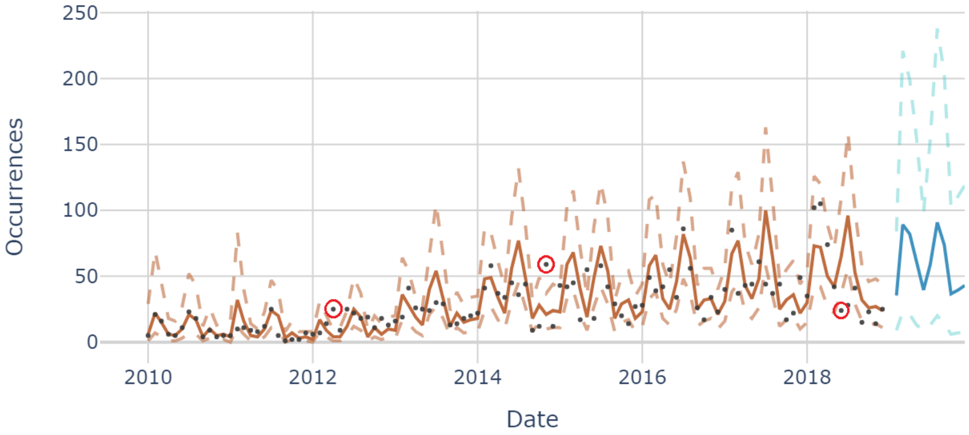


Figure 2.15: Prediction of number of occurrences.

Routine forecasting and monitoring will be responsible for checking the stability of the model and suggesting anomalies, sudden instabilities, and deterioration in forecast performance that have not been anticipated through expert intervention. We would therefore raise alarms whenever observed values lay outside predictive intervals, like the observations marked with a circumference in Figure 2.15. As Figure 2.16 shows, the credible intervals for one-step ahead predictions (solid blue) adequately capture the observations, showing only slight under-coverage for credible intervals between 50% and 95%.

Compared to other popular models used with non-negative integer time series, like GLARMA (Benjamin et al., 2003), and INGARCH (Ferland et al., 2006), Table 2.3 shows that our proposed model (2.5) offers significantly better point forecasts than any of them, using either a Poisson or Negative Binomial distribution for the observations, or a standard DLM. Additionally, as seen in Figure 2.16, the credible intervals (solid blue line) of our model closely match

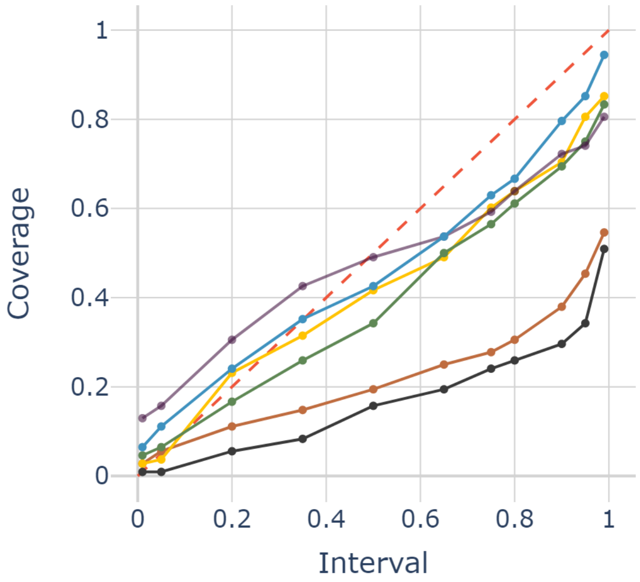


Figure 2.16: Empirical (solid) versus nominal (dashed) coverage of credible intervals for *wind shear*.

the nominal coverage probability (45 degree dashed line) and perform nicely when compared with the analysed competitors: Poisson GLARMA (brown), NB GLARMA (yellow), Poisson INGARCH (black), NB INGARCH (green) and DLM (purple).

	Our Model	GLARMA (Poi)	GLARMA (NB)	INGARCH (Poi)	INGARCH (NB)	DLM
MSE	296.74	350.33	392.35	425.51	471.22	379.7
MAE	11.96	13.35	14.03	14.81	15.37	14.53
MAPE	0.53	0.63	0.65	0.61	0.59	0.8
Theil's U	0.91	0.94	0.99	1.03	1.09	1.03

Table 2.3: Error metrics for the predictive median of models for *wind shear*.

The analysis is completed with the prediction of the number of occurrences of each severity class. Table 2.4 shows a summary of the predictions, with μ' and σ' designating the predictive mean and standard deviation of the number of occurrences for the next period to forecast (in this case, the next month January 2019), and α'_j, the parameter of the posterior distribution of the j-th severity class of such event. Thus, the expected number of occurrences for each severity would be $\mu'\alpha'_j \,/\, \sum_{h=1}^{5} \alpha'_h$, e.g. 33.27 severity 4 occurrences.

μ'	σ'	α'_1	α'_2	α'_3	α'_4	α'_5
35.38	26.55	2	4	26	2881	150

Table 2.4: Prediction summary.

Regarding sensitivity to the hyper-parameters $\boldsymbol{m}_0$, $\boldsymbol{C}_0$ for the prior of the initial state; the election of a different $\boldsymbol{m}_0$, e.g. the usual vector of zeros which does not use prior information, has little effect on forecast performance beyond the first few observations (algorithm particles still arrive relatively quickly to a zone with high probability) unless we deviate a lot from these values for the states (with prior values for all states outside the interval $[-2, 2]$). Moving from the proposed variance $\boldsymbol{C}_0 = \boldsymbol{I}_{13}/10$ up to $\boldsymbol{I}_{13}$ results in too much dispersion and very high predictive intervals during the first observations, more resamples and overall worse forecasting performance; using lower values for the diagonal down to $\boldsymbol{I}_{13}/100$ also worsens the point forecast metrics and coverage of the predictive distributions, although less dramatically.

2.6.2 TCAS warnings

Traffic Collision Avoidance Systems (TCAS) warn pilots of the presence of other aircrafts which may present a threat of mid-air collisions. This type of occurrences, unlike *wind shear*, presents a *stress effect*, Figure 2.1(b). Since it does not show any other of the effects mentioned in Section 2.2, we model it through *stress effect* model in Section 2.3.2.

Using Algorithm 2.1 we obtain the one-month ahead forecasts in Figure 2.17, with predictive mean (solid line) and 99% probability band (dashed lines), and we check that it adequately predicts time series with this effect.

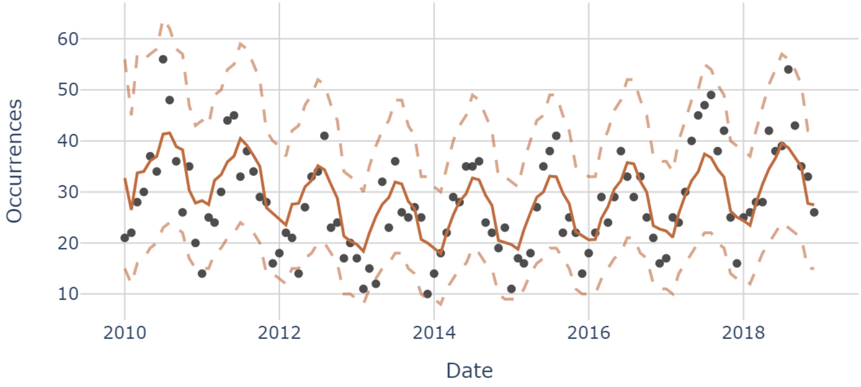


Figure 2.17: Observations and forecasts for *TCAS warnings*.

Figure 2.18 shows that the predictive distributions adequately cover the observations at different credible intervals (blue line) and that there is an improvement in coverage over the *basic* model (yellow) from Section 2.3.1.

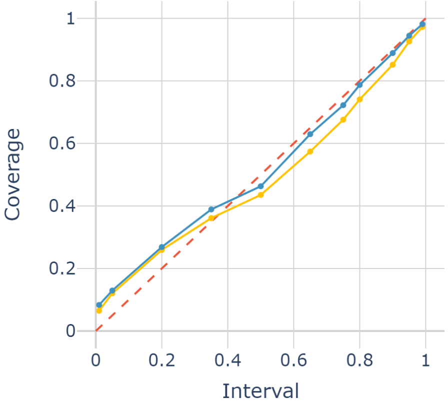


Figure 2.18: Coverage plot of *TCAS* occurrence.

This is also the case for the error metrics of the point forecasts (Table 2.5).

	Stress model	Basic model
MSE	36.57	45.67
MAE	4.78	5.28
MAPE	0.2	0.22
Theil's U	0.77	0.82

Table 2.5: Error metrics *TCAS* occurrence.

Furthermore, as Figure 2.19 shows, the posterior distribution of parameter a in the model concentrates around 12, away from 0, which is consistent with the relevance of the *stress effect*.

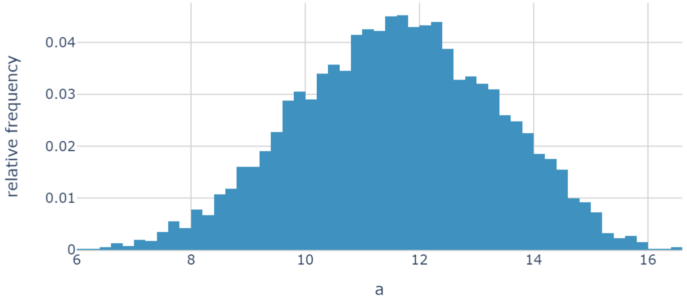


Figure 2.19: Posterior distribution of a.

A sensitivity analysis concerning the hyper-parameters of the priors over a, b and σ^2 suggests that, as long as plausible values are chosen, the performance of the *stress effect* model (2.1) is robust. For example, for *TCAS warnings*, even if we would to select negative values for the means of a and b ($\mu_a = \mu_b = -5$), indicating a negative stress effect, with variances $\sigma_a^2 = \sigma_b^2 = 10$ we arrive to similar posterior distributions. Hence we recommend the election of means consistent with the available data, and relatively high variances that give more leeaway for misspecification of the means.

2.6.3 Dependence of occurrence types.

The relevance of the *dependence* model in Section 2.3.2 can be readily exemplified with simulated data of a new occurrence type (Figure 2.20) that shows

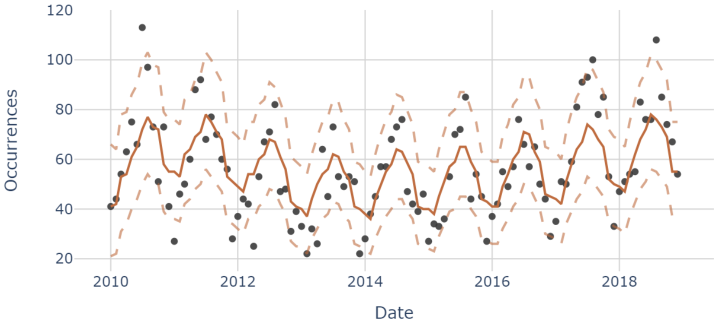


Figure 2.20: Observations and forecasts of *dependent* occurrence.

dependence with *TCAS warnings*.

The use of the *dependence* model and Algorithm 2.4 to sample from it, improves the forecast performance over the basic model from Section 2.3.1 that assumes independence as shown in Table 2.6. Similarly to Algorithm 2.1, the proposed MCMC sampler is quite robust against reasonable choices of the values in the hyper-parameters of the priors.

	Dependence model	Basic model
MSE	159.54	245.25
MAE	9.89	11.51
MAPE	0.21	0.22
Theil's U	0.78	0.94

Table 2.6: Error metrics *dependent* occurrence.

2.7 Discussion

We have provided a methodology to forecast general count time series that can present several combinations of effects based on an initial standard model, suitable for situations in which the Poisson rate remains relatively stable over the period of interest.

In most practical cases, several effects impact the rate evolution. Therefore, the initial model was adapted by adding specific components (stress effect, seasonal and trend effect, group effect and dependence), and accompanying algorithms to forecast with these new models were provided. Also, since time series tend to show more than one effect, we have illustrated how the models can be combined with a case study in Section 2.6.1.

Additionally, we have described a model to predict the proportion of future observations that belong to different classes, and which can be used in combination with any of the above models. We have also suggested a model to address the problem of underreporting, which can be a problem in many domains, specially in relation to the classification of observations into severity levels, where less severe ones tend to be underreported.

The models developed in this chapter are illustrated with an application to AS occurrence data. In fact, the proposed models are fundamental in the risk management methodology in Rios Insua et al. (2018), feeding its AS resource allocation models. They are also important in predicting and monitoring events that allow identifying anomalies related to an unexpected increase (or decrease) in the number of occurrences. The methodology emphasizes a *management by exception principle* (West & Harrison, 1997) with our models used for routine inference, prediction (and decision support) under standard circumstances until exceptional ones arise in which case an intervention is requested.

The performance of our model was compared to other popular ones like dynamic linear models (DLM), generalized linear ARMA (GLARMA), and integer-valued GARCH (INGARCH) models, showing better forecasting performance with the AS time series studied. However, some of these models assuming negative binomially distributed observations might be more relevant when exploring approaches at smaller time (weeks) and spatial (airport) frames, which might present more overdispersion. Also, given the high safety levels in the aviation system we should expect numerous zero counts. All of which motivates the development of new models in the following chapter to address these problems.

Chapter 3

Models for count time series with frequent zeros

3.1 Introduction

In this chapter, we focus on meeting the second objective presented in Section 1.4: the development of models for count time series with frequent zeros and possible overdispersion that improve currently established approaches. We also present a methodology for using the predictive distributions of said models to make informed decisions that reduce the risk of reaching critical situations, therefore addressing the corresponding secondary objective as well.

We propose Bayesian state-space models that are flexible enough to adequately forecast high and low count series and exploit cross-series relationships within a hierarchical multivariate approach. This methodology is illustrated with the demand forecasting problem faced by a major retail company, integrated within its inventory management planning methodology, introduced in Section 1.1. The company has hundreds of stores, each one with thousands of products whose demand has to be accurately predicted in order to efficiently manage its stock.

It is worth reiterating that, as in most practical cases, we are interested in forecasting both aggregated and individual demand at any hierarchy level (product family, store section, store, neighborhood, city, region, country). And, therefore, the time series faced as part of the inventory management planning system tend to be very diverse (some with low counts and, even, many days of zero sales, while others have few zero observations).

To better forecast demand, models must be able also to take into account additional relevant information in the form of regression variables, like promotions or prices, which can significantly improve model performance (Ali et al., 2009). Another important aspect to consider is the relation among series at a given hierarchy level or between between stores with similar location (in socioeconomic terms, climate,...). We are also interested in giving full distribution predictions instead of point forecasts of the demand: this is specially useful in our motivating case as it offers a way of calculating the probability of OoS events, used to make informed decisions about when to place an order. These forecast distributions are one of the main inputs to any order planning module within a decision support system (DSS). In our case study, when looking at the most disaggregated level, daily sales of a product in a store, we are mainly interested in forecast demand for the days remaining until the next resupply. Although we consider as well different forecast horizons for other applications and aggregation levels.

The frequency of low counts and many zeros in our time series means that traditional models like ARIMA (Box et al., 2015), exponential smoothing (Hyndman et al., 2008) and Gaussian DLMs (West & Harrison, 1997) are not the most adequate. Also, time series of sales are usually non stationary. All of which encourages the development of new models.

In this chapter we build upon Dynamic Generalized Linear Models (West et al., 1985) to improve the forecast performance with count time series with the aforementioned characteristics: many zeroes and possible overdispersion. Our univariate model, like Berry and West (2020), uses a mixture of two DGLMs, one for the binary outcome *sales/no sales* and another for the *number of sales*, but the later is modeled through a negative binomial distribution instead of a Poisson, which offers more flexibility and more adequately models the series in our use case, specially those with significant overdispersion.

We begin by introducing our motivating problem and illustrating some key issues to be considered in forecasting demand count series in Section 3.2. Section 3.3 introduces a univariate model to forecast those count series, extended to the multivariate case in Section 3.4, taking advantage of cross-series information (among products within a store, among a product at different stores, among stores). Some criteria to deal with OoS events are proposed in Section 3.5. A case study with real data is presented in Section 3.6. We end up with some discussion.

3.2 Exploratory Analysis

In our motivating problem, we need to forecast the demand of several thousands of products for a retail company, each one defining a count time series. Our raw data consists of daily sales of the products at several stores over 194 days, along with relevant information like price, promotions, available stock, etc. Although the main interest is in forecasting the daily time series, we also consider other aggregated series (monthly, weekly, product in a region, etc).

Table 3.1 provides summary statistics of examples of the types of time series to forecast, along with their corresponding *Stock Keeping Unit* (SKU), a number that uniquely identifies each product. These are representative of the type of count time series commonly encountered in retail. Observe two peculiarities common in time series in our application domain: some products like shaving gel (SKU '24144') or shampoo (SKU '216880') show many days with zero sales (84.1% and 82.6%, respectively); others exhibit significant overdispersion, as with beers (SKU '182', '14752', '29352' and '29358'), which have a variance much higher than its mean. In some occasions, sales time series might also display long right tail distributions, like SKU '29352', with a mean higher than the median.

SKU	Description	Mean	Variance	Median	% 0 sales
182	Beer 1	30.6	331.2	29.0	1.0
14752	Beer 2	30.9	1024.0	24.0	1.0
24144	Shaving gel	0.2	0.2	0.0	84.1
29352	Non-alcoholic beer 1	28.6	1797.8	16.0	2.6
29358	Non-alcoholic beer 2	6.7	139.2	0.0	51.3
33057	Liquid Yogurt 1	0.6	1.0	0.0	59.5
70598	Bathroom cleaner	1.2	2.2	1.0	41.0
117866	Yogurt 1	1.9	2.0	2.0	14.4
123683	Liquid Yogurt 2	0.6	0.6	0.0	56.4
130111	Detergent	0.7	1.2	0.0	59.5
131735	Liquid Yogurt 3	0.6	0.8	0.0	56.9
151114	Yogurt 2	1.6	2.6	1.0	29.7
177427	Diaper	0.0	0.1	0.0	98.5
216880	Shampoo	0.3	1.0	0.0	82.6

Table 3.1: Summary statistics for some time series (daily sales in a store).

It is also common to observe the presence of *seasonalities* in sales time series. With daily data, as in our case, we tend to observe weekly (period 7) and/or yearly (period 365) seasonalities. For example, daily sales of SKU '182' beer show seasonality of period 7, Figure 3.1. In some cases, series also exhibit some local trend, for example when a new product is introduced.

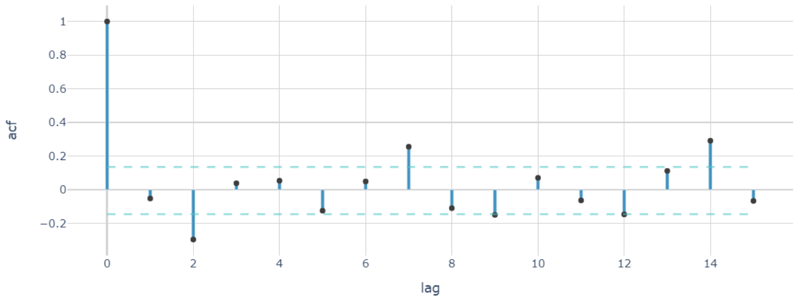


Figure 3.1: Weekly seasonality in the sales of beer (SKU '182').

Besides these, two effects are common in count series in this specific domain.

Promotions The introduction of promotions can cause spikes in product demand. Retail experts frequently observe that the effect of promotions on sales is not the same through all its duration; instead, the increase in demand tends to be lower at the beginning and the end of the promotion. To deal with this effect, instead of using a simple binary regression variable (with 1 indicating active promotion and 0, no promotion) a categorical variable with 3 levels, beginning, middle and end of promotion can be used (using dummy coding).

Substitute goods The presence of families of substitute goods, and the availability of information about the current stock of products in a store, means that when there is an OoS event with some products, we could anticipate an increase in demand for substitute ones. These products can also cannibalize sales from others when their price becomes significantly lower than the alternative.

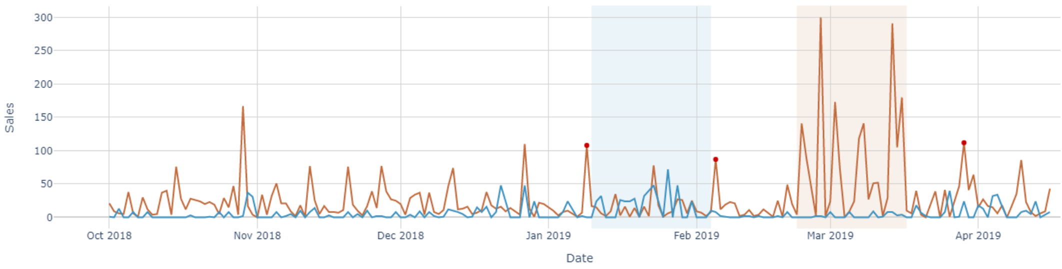


Figure 3.2: Daily sales of two beer brands (SKU '29352', brown; '29358', blue).

Figure 3.2 shows the number of sales of two brands of alcohol-free beer in a store, generally the beer with SKU '29352' in brown is a bigger seller, despite having a slightly higher price on average than the beer with SKU '29358' in blue. Two periods might indicate that both beers can be considered substitute products: the first one during the second half of Jan '19, in light blue, corresponds to an interval in which the price of '29358' is almost half the price of

’29352’ and is the only one in which ’29358’ sales are on the same level than ’29352’ sales; the second one, during May ’19, light brown, to a period with the price of ’29352’ significantly lower than the other brand. Additionally there are three spikes in ’29352’ sales (red points) that coincide with sudden increases in ’29358’ price, leading to equal or similar prices for both beer brands.

This effect can be taken into account in several ways, for example by introducing the price difference with the substitute product as a covariate in the univariate model, or both prices in the multivariate model.

Additionally, sales time series can also show positive correlations, for example between Beer 1 and Beer 2 (SKUs ’182’ and ’14752’) in Figure 3.3.

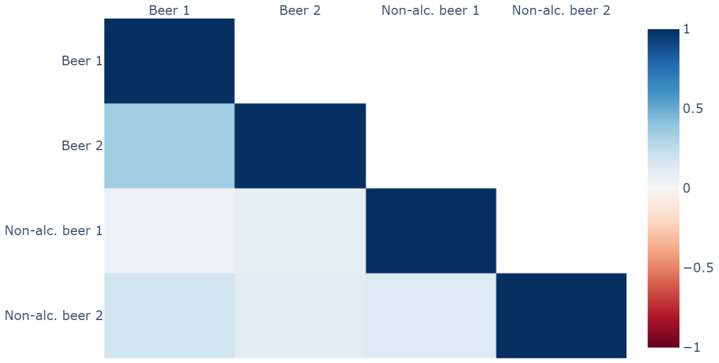


Figure 3.3: Correlation coefficient of sales in the same store of 4 products.

In summary, the presence of these effects justify the introduction of covariates and the development of multivariate models.

3.3 Model

Time series of counts with many zero-valued observations are commonly encountered while analyzing data coming from natural disasters, inventory management or disease surveillance, to name but a few relevant domains.

State space models consider a time series y_t as the output of a dynamic system perturbed by stochastic disruptions. They offer a flexible framework for a wide range of applications and lend themselves quite naturally to be treated within a Bayesian paradigm. One such class are Dynamic Linear Models (DLM) extensively treated in West and Harrison (1997). A shortcoming of DLMs for its use with time series of counts is that they assume a normal distribution for the observations. This might be appropriate for modelling time series with large counts, where the probability assigned to negative outcomes is very low. However, the abundance of series with low counts in retail makes necessary the use of different distributions for the observations. Dynamic Generalized Linear Models (DGLM) (West et al., 1985) extend the observational distributions of DLMs to any probability density function (or p.m.f. in the discrete case) of the exponential family,

$$p(y_t|\eta_t, V_t) = \exp\{V_t^{-1}[T(y_t)\eta_t - a(\eta_t)]\}b(y_t, V_t), \tag{3.1}$$

for some defining quantities η_t and V_t, and known functions $T(y_t)$, $a(\eta_t)$ and $b(y_t, V_t)$. The DGLM for the series y_t is defined through the components:

$$\begin{array}{ll} \text{Observation model:} & p(y_t|\eta_t) \quad \text{and} \quad g(\eta_t) = \lambda_t = \boldsymbol{F}_t\boldsymbol{\theta}_t, \\ \text{State equation:} & \boldsymbol{\theta}_t = \boldsymbol{G}_t\boldsymbol{\theta}_{t-1} + \boldsymbol{\omega}_t \quad \text{with} \quad \boldsymbol{\omega}_t \sim (\boldsymbol{0}, \boldsymbol{W}_t), \\ \text{Prior information:} & \boldsymbol{\theta}_0 \sim (\boldsymbol{m}_0, \boldsymbol{C}_0), \end{array} \tag{3.2}$$

where g is the link function relating to linear predictor λ_t, and $\boldsymbol{\omega}_t \sim (\boldsymbol{0}, \boldsymbol{W}_t)$ denotes that we do not assume any particular distribution for the evolution errors, only its mean $\boldsymbol{0}$, and variance $\boldsymbol{W}_t$. Matrices $\boldsymbol{F}_t$ and $\boldsymbol{W}_t$ contain (as diagonal blocks) different factors deemed important to predict series behavior, like trend, seasonality, or regression variables (Prado & West, 2010).

3.3.1 DGLM mixture

As mentioned in Section 3.1, the model proposed to forecast time series of counts with many zeros, as those encountered in the current application domain, analogously to DCMM (Berry & West, 2020), is a mixture of two DGLMs: a Bernoulli for *zero/non-zero sales*, and a negative binomial for the *number of sales*. From the series of counts y_t, we define the binary time series $z_t = \mathbb{1}_{(y_t > 0)}$. The global model would be defined through

$$z_t \sim Ber(\pi_t) \quad \text{and} \quad y_t | z_t = \begin{cases} 0, & \text{if } z_t = 0, \\ 1 + x_t, \quad x_t \sim NB(r_t, p_t) & \text{if } z_t = 1, \end{cases} \tag{3.3}$$

i.e. $y_t = z_t(x_t + 1)$ with $z_t \sim Ber(\pi_t)$, $x_t \sim NB(r_t, p_t)$. The link functions relating to the linear predictors for the Bernoulli (Ber) and Negative Binomial (NB) components respectively are

$$\text{logit}(\pi_t) = \boldsymbol{F}_t^0 \boldsymbol{\theta}_t^0 \quad \text{and} \quad \log(p_t) = \boldsymbol{F}_t^+ \boldsymbol{\theta}_t^+. \tag{3.4}$$

We consider a fixed r_t for all t, so that the NB belongs to the exponential family of distributions. The state equations are

$$\boldsymbol{\theta}_t^0 = \boldsymbol{G}_t^0 \boldsymbol{\theta}_{t-1}^0 + \boldsymbol{\omega}_t^0 \quad \text{and} \quad \boldsymbol{\theta}_t^+ = \boldsymbol{G}_t^+ \boldsymbol{\theta}_{t-1}^+ + \boldsymbol{\omega}_t^+,$$

and the prior moments for the states of each DGLM are $\boldsymbol{m}_0^0, \boldsymbol{C}_0^0, \boldsymbol{m}_0^+, \boldsymbol{C}_0^+$.

The conditional model for the positive counts, $y_t | (z_t = 1)$, is a *shifted* negative binomial DGLM. This component is only updated when sales are observed, $z_t = 1$; otherwise, the value y_t is treated as missing. This allows for a range of applications with significant probability of zeros over time. In those cases, the NB part will play a limited role. Also, the use of a NB instead of a Poisson (which is a special case, with $r_t \to \infty$) allows to better deal with cases of overdispersion, as will be showed later.

Due to treating zero observations as *missing* in the NB part, in the case of time series with sudden and long intervals with no sales, this part might take some time to adapt, with forecast performance deteriorating. Although, as later discussed, this can be mitigated through the adequate use of discount factors, another option is the use of a *zero inflated* version of model (3.3), i.e. without the shifting ($y_t = z_t x_t$), through

$$p(y_t|\pi_t, r_t, p_t) \sim \begin{cases} (1-\pi_t) + \pi_t \, NB(0|r_t, p_t), & \text{if } y_t = 0, \\ \pi_t \, NB(y_t|r_t, p_t) & \text{if } y_t > 0. \end{cases} \tag{3.5}$$

3.3.2 Sequential Learning and Forecast

For the NB term, we use the parametrization in Appendix A.1, which with a fixed r_t, can be expressed in the canonical exponential family form (3.1) with

$$\begin{aligned} &T(y_t) = y_t, \qquad V_t = 1 \implies \Phi_t := V_t^{-1} = 1, \qquad \eta_t = \log(p_t), \\ &a(\eta_t) = -r_t \log(1 - \exp \eta_t) \quad \text{and} \quad b(y_t, V_t) = \frac{\Gamma(r_t + y_t)}{y_t! \Gamma(r_t)}. \end{aligned}$$

The update and forecast procedure of this DGLM with distribution $y_t \mid \eta_t, V_t \sim NB(r_t, p_t)$ for the observations, follows the one in (West & Harrison, 1997) with the following prior, predictive and posterior distributions:

Conjugate Prior for η_t,

$$\begin{aligned} \pi(\eta_t|D_{t-1}) &= c(\alpha_t, \beta_t) \exp(\alpha_t \eta_t - \beta_t a(\eta_t)) \qquad (3.6) \\ &= c(\alpha_t, \beta_t) \exp(\alpha_t \eta_t + \beta_t r_t \log(1 - \exp \eta_t)). \end{aligned}$$

Since it must integrate to one,

$$c(\alpha_t, \beta_t) = \Big(\int_{\log 0}^{\log 1} \exp(\alpha_t \eta_t + \beta_t r_t \log(1 - \exp \eta_t)) \, d\eta_t \Big)^{-1} = B(\alpha_t, \beta_t r_t + 1)^{-1}.$$

Therefore $\pi(\eta_t|D_{t-1}) = c(\alpha_t, \beta_t) B(\alpha_t, \beta_t r_t + 1)^{-1}$.

Conjugate Prior for p_t. With the change of variable $p_t = \exp \eta_t$ in (3.6) we have

$$\pi(p_t|D_{t-1}) = B(\alpha_t, \beta_t r_t + 1)^{-1} p_t^{\alpha_t - 1}(1-p_t)^{\beta_t r_t} \sim Be(\alpha_t, \beta_t r_t + 1).$$

Predictive distribution for Y_t,

$$\begin{aligned}\pi(Y_t|D_{t-1}) &= \frac{c(\alpha_t, \beta_t) b(Y_t, V_t)}{c(\alpha_t + \Phi_t Y_t, \beta_t + \Phi_t)} \\ &= \frac{\Gamma(r_t + Y_t)}{Y_t! \Gamma(r_t)} \frac{B(\beta_t r_t + 1 + r_t, \alpha_t + Y_t)}{B(\beta_t r_t + 1, \alpha_t)} \sim BNB(\beta_t r_t + 1, \alpha_t, r_t).\end{aligned}$$

Posterior distribution for η_t. After observing Y_t, the posterior is the prior with updated parameters $\alpha_t' = \alpha_t + Y_t$, $\beta_t' = \beta_t + 1$

$$\pi(\eta_t|D_t) = B(\alpha_t + Y_t, \beta_t r_t + r_t + 1)^{-1} (e^{\eta_t})^{\alpha_t + Y_t}(1 - e^{\eta_t})^{\beta_t r_t + r_t}.$$

Posterior distribution for p_t. Equivalently, we have

$$\pi(p_t|D_t) = Be(p_t|\, \alpha_t + Y_t, \beta_t r_t + r_t + 1).$$

Note that in the conjugate updating method, the scale parameter V_t is assumed known for all t, which is the case with the models in the present thesis. Recent work by Souza et al. (2018) introduces an extension to the update procedure that allows for an unknown scale parameter that varies over time.

The update for the Bernoulli DGLM can be obtained in an analogous manner, and the resulting procedure for joint the mixture model (3.3), using the moment matching technique in West et al. (1985), can thus be summarized, for each $t > 0$, as :

- One step ahead prior moments for the states given $\mathcal{D}_{t-1}$ ($y_{1:t-1}$ and other relevant information), $\boldsymbol{\theta}_t|\mathcal{D}_{t-1} \sim (\boldsymbol{a}_t, \boldsymbol{R}_t)$, with

$$\boldsymbol{a}_t = \boldsymbol{G}_t \boldsymbol{m}_{t-1} \qquad \boldsymbol{R}_t = \boldsymbol{G}_t \boldsymbol{C}_{t-1} \boldsymbol{G}_t' + \boldsymbol{W}_t$$

- One step ahead forecasts are based on the Bernoulli and the Beta-Negative Binomial (BNB) distributions

  $$z_t|\mathcal{D}_{t-1} \sim Ber\Big(\frac{\alpha_t^0}{\alpha_t^0+\beta_t^0}\Big) \quad \text{and} \quad x_t|\mathcal{D}_{t-1} \sim BNB(\beta_t^+ r_t + 1, \alpha_t^+, r_t),$$

  with hyper-parameters $\alpha_t^0, \beta_t^0, \alpha_t^+, \beta_t^+$ satisfying

  $$\begin{aligned} f^0 &= \gamma(\alpha_t^0) - \gamma(\beta_t^0), & q^0 &= \dot{\gamma}(\alpha_t^0) + \dot{\gamma}(\beta_t^0), \\ f^+ &= \gamma(\alpha_t^+) - \gamma(\alpha_t^+ + \beta_t^+ r_t + 1), & q^+ &= \dot{\gamma}(\alpha_t^+) - \dot{\gamma}(\alpha_t^+ + \beta_t^+ r_t + 1), \end{aligned}$$

  where $f^0 = f^+ = \boldsymbol{F}_t \boldsymbol{a}_t$ and $q^0 = q^+ = \boldsymbol{F}_t \boldsymbol{R}_t \boldsymbol{F}_t{}'$ are the predictive mean and variance of the corresponding linear predictor λ_t in (3.4); and γ, $\dot{\gamma}$ are the digamma and trigamma functions, respectively.

- Posterior moments for the states after observing y_t, $\boldsymbol{\theta}_t|\mathcal{D}_t \sim (\boldsymbol{m}_t, \boldsymbol{C}_t)$,

  $$\boldsymbol{m}_t = \boldsymbol{a}_t + \boldsymbol{R}_t \boldsymbol{F}_t{}'(\widehat{f}_t - f_t)/q_t, \qquad \boldsymbol{C}_t = \boldsymbol{R}_t - \boldsymbol{R}_t \boldsymbol{F}_t{}' \boldsymbol{F}_t \boldsymbol{R}_t (1 - \widehat{q}_t/q_t)/q_t,$$

  with

  $$\begin{aligned} \widehat{f}_t^0 &= \gamma(\alpha_t^0 + z_t) - \gamma(\beta_t^0 + 1 - z_t), & \widehat{f}_t^+ &= \gamma(\alpha_t^+ + x_t) - \gamma(\alpha_t^+ + x_t + \beta_t^+ r_t + r_t + 1), \\ \widehat{q}_t^0 &= \dot{\gamma}(\alpha_t^0 + z_t) + \dot{\gamma}(\beta_t^0 + 1 - z_t), & \widehat{q}_t^+ &= \dot{\gamma}(\alpha_t^+ + x_t) - \dot{\gamma}(\alpha_t^+ + x_t + \beta_t^+ r_t + r_t + 1). \end{aligned}$$

As mentioned earlier, with the shifted NB in model (3.3) the last step is only performed after observing a sale, $y_t > 0$. Otherwise, $x_t = y_t - 1$ is treated as missing and $\boldsymbol{m}_t^+ = \boldsymbol{a}_t^+$ and $\boldsymbol{C}_t^+ = \boldsymbol{R}_t^+$. This is not the case when using model (3.5), where the number of sales ($x_t = y_t$) is modeled directly.

If we denote with $\mathbb{1}_{HS}$ a binary variable that is 1 when using model (3.3), and 0, when using (3.5), the p.m.f of the resulting predictive distribution for the observations is given by

$$p(y_t|D_{t-1}, \pi_t) = (1 - \pi_t)\delta_0(y_t) + \pi_t BNB(y_t - \mathbb{1}_{HS}|\beta_t^+ r_t + 1, \alpha_t^+, r_t), \tag{3.7}$$

with $(\pi_t|D_{t-1}) \sim Be(\alpha_t^0, \beta_t^0)$ and δ_0 the Kronecker delta function. Equivalently, we have

$$y_t|D_{t-1} = z_t(x_t + \mathbb{1}_{HS}), \quad z_t \sim Ber\left(\frac{\alpha_t^0}{\alpha_t^0 + \beta_t^0}\right), \, x_t \sim BNB(\beta_t^+ r_t + 1, \alpha_t^+, r_t).$$

We, then, have:

- Mean

  $$\mu_{z_t}(\mu_{x_t} + \mathbb{1}_{HS}) = \frac{\alpha_t^0}{\alpha_t^0 + \beta_t^0}\left(\frac{\alpha_t^+}{\beta_t^+} + \mathbb{1}_{HS}\right),$$

  where μ_{z_t} and μ_{x_t} are, respectively, the mean of the predictive distributions of the Bernoulli and the NB DGLMs.

- Variance

  $$\begin{aligned}&(\sigma_{z_t}^2 + \mu_{z_t}^2)(\sigma_{x_t}^2 + (\mu_{x_t} + \mathbb{1}_{HS})^2) - \mu_{z_t}^2(\mu_{x_t} + \mathbb{1}_{HS})^2\\ &= \frac{\alpha_t^0}{\alpha_t^0 + \beta_t^0}\left(\frac{\alpha_t^+(1 + \beta_t^+)(\alpha_t^+ + r_t\beta_t^+)}{{\beta_t^+}^2(\beta_t^+ r_t - 1)} + \left(\frac{\alpha_t^+}{\beta_t^+} + \mathbb{1}_{HS}\right)^2\right) - \left(\frac{\alpha_t^0}{\alpha_t^0 + \beta_t^0}\right)^2\left(\frac{\alpha_t^+}{\beta_t^+} + \mathbb{1}_{HS}\right)^2,\end{aligned}$$

  where $\sigma_{z_t}^2$ and $\sigma_{y_t}^2$ are, respectively, the variance of the predictive distributions of the Bernoulli and the NB DGLMs.

- Median

  $$\begin{cases} 0, & \text{if } \mu_{z_t} < 0.5, \\ Q_{BNB}((\mu_{z_t} - 0.5)/\mu_{z_t}; \beta_t^+ r_t + 1, \alpha_t^+, r_t) + \mathbb{1}_{HS} & \text{otherwise,} \end{cases}$$

  where $Q_{BNB}(x; \beta, \alpha, r)$ denotes the x-th quantile of the beta negative binomial with parameters β, α and r. Similarly, we obtain the desired credible intervals (Figure 3.4).

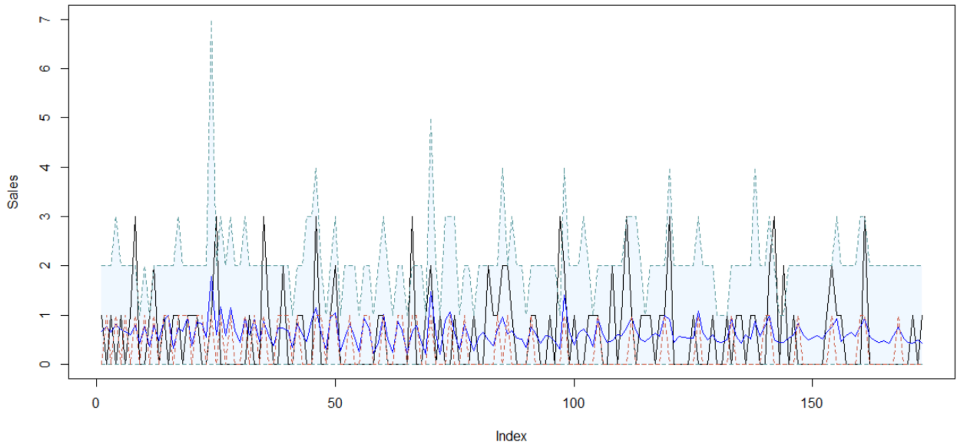


Figure 3.4: One step ahead mean (blue), median (red), 90% credible intervals (light blue) and observations (black) for yogurt sales.

For more than one step ahead forecast distributions, we use Monte Carlo simulations to generate N random projections up to the desired forecast horizon, as reflected in Algorithm 3.1.

```
$T \equiv$ "last index of the time series"; $N \equiv$ "number of projections";
$\boldsymbol{P} \equiv$ "matrix for storing the simulations";
for $p$ in $1:N$ do
    for $t$ in $T+1:T+k$ do
        Calculate prior moments for states, $\boldsymbol{\theta}_t|\mathcal{D}_{t-1} \sim (\boldsymbol{a}_t, \boldsymbol{R}_t)$;
        Calculate hyper-parameters $\alpha_t^0, \beta_t^0, \alpha_t^+, \beta_t^+$, use them to draw $y_t^*$
          from (3.7);
        Consider draw as observed value, $y_t = y_t^*$;
        Save the draw, $\boldsymbol{P}_{pt} = y_t$;
        Update state moments with $y_t$ and other relevant information
          $\boldsymbol{\theta}_t|\mathcal{D}_t \sim (\boldsymbol{m}_t, \boldsymbol{C}_t)$;
    end
end
```

Algorithm 3.1: Forecast k-step ahead via simulations.

At each projection p, and for each time t of the k future observations to predict, a random draw from (3.7) is performed and used as the observed value for the previously explained update procedure. This gives full forecast distributions and facilitates analyzing and performing inference on the projections, predictive intervals (Figure 3.5) and cumulative outcomes. A useful direct application of the sample is the estimation of the probability that the cumulative demand up to a time point reaches a certain threshold. This is particularly useful for knowing the probability of an OoS event, raise alarms and plan orders, as we will show later.

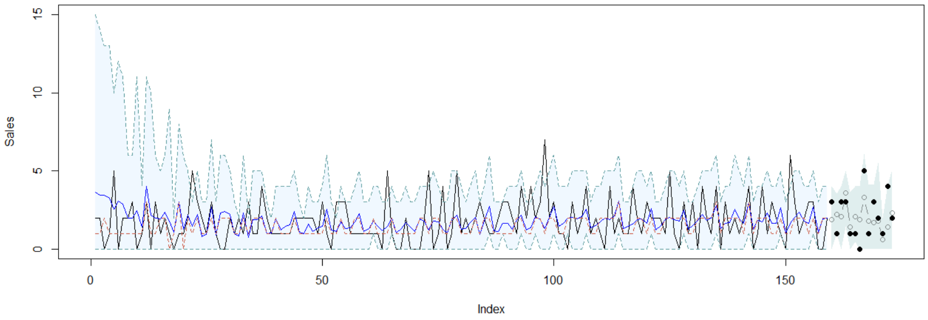


Figure 3.5: Two weeks ahead (14 days) forecast (grey) with intervals and real values (black points) for SKU ’117866’.

3.3.3 Discount factors

The specification of the unknown state evolution variance matrix $\boldsymbol{W}_t$ is crucially important for successful forecasting. Its values control the stochastic variation in the evolution of the model and, hence, determine stability over time. In the system equation, $\boldsymbol{W}_t$ leads to an increase in uncertainty or, equivalently, a loss of information about the state vector between times t and $t+1$. Due to the difficulty of correctly specifying this variance matrix, a common alternative, adopted here, is the use of discount factors (West & Harrison, 1997) which are easier to elicit. The discount factor δ takes values in $(0, 1]$, with 1 being the case of a stable state vector with no stochastic changes

($\boldsymbol{W}_t = \mathbf{0}$). In practice, discount factors are usually assigned values between 0.8 and 0.99. The prior variance $\boldsymbol{R}_t$ for the state in the sequential update procedure changes from $\boldsymbol{G}_t\boldsymbol{C}_{t-1}\boldsymbol{G}_t' + \boldsymbol{W}_t$ to $\boldsymbol{G}_t\boldsymbol{C}_{t-1}\boldsymbol{G}_t'/\delta$.

While it is possible to use a single discount factor for all model components, this might not be always adequate. For example, the trend and seasonal components often require different discount factors: usually, the seasonal characterization is more durable in time and, hence, more accurately represented through higher values of δ. When using several discount factors for the different components we divide the corresponding block of matrix $\boldsymbol{R}_t$ in the updating procedure, by the the discount factor chosen for that component. Additionally, there are occasions when a discount factor that is not constant, and varies through time (δ_t) might prove beneficial.

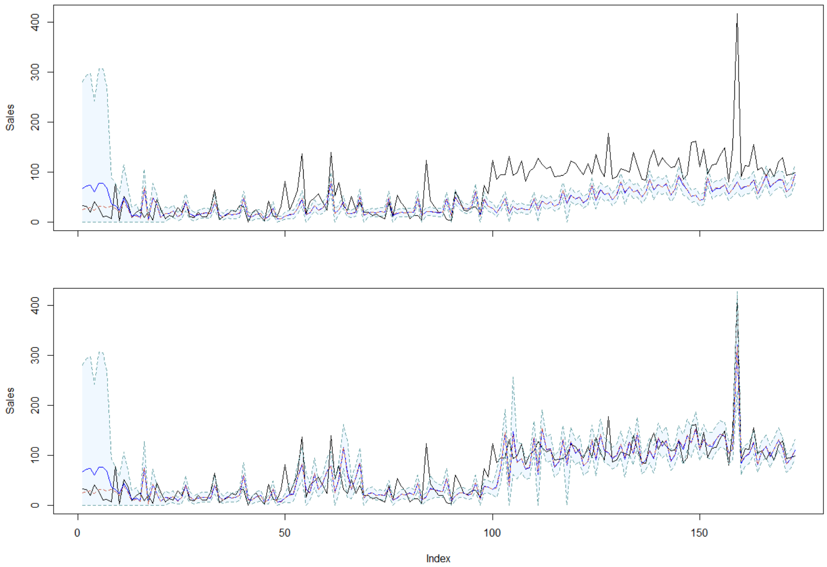


Figure 3.6: One step ahead plots with mean (blue), median (red), 90% credible intervals (light blue) and observations (black).

Due to the high number of time series to be monitored in the retail domain, an automatic detection system for unusual bad model performance was implemented, which automatically lowers the discount factor for a predetermined number of time instants. This *adaptive* discount factor reflects our state of greater uncertainty about the state parameter and enables the model to adapt faster to the changes in the underlying process, as shown in Figure 3.6: model with constant discount factor for all components (level, seasonal) above, and model with the level component discount factor being *adaptive* below.

The resulting automatic exception detection and handling routine (and its integration in the updating process) is sketched in Algorithm 3.2. When several days with zero sales are detected, which could signal the beginning of an interval without sales like those discussed earlier, the discount factor of the Bernoulli term momentarily changes from its *usual* value of 0.95 to 0.8. Also, when the observations fall outside the 99% credible interval, we reduce the the NB term discount factor from 0.9 to 0.8, to facilitate faster adaptation.

```
$T \equiv$ "last index of the time series";
$k \equiv$ "number of time instants to add uncertainty after an exception";
for $t$ in $1 : T$ do
    Calculate predictive moments for states $\boldsymbol{\theta}_t^0 | D_{t-1}$, $\boldsymbol{\theta}_t^+ | D_{t-1}$ ;
    Calculate one step ahead predictive distribution $y_t | D_t$ ;
    Observe $y_t$ ;
    if $y_t = 0$ and $y_{t-1} = 0$ then
        $\delta^0_{t+1:t+k} = 0.8$ ;
    else
        if $y_t \in$ 99% credible interval of $p(y_t | D_{t-1})$ then
            $\delta^+_{t+1:t+k} = 0.8$;
    end
    Calculate posterior moments for states $\boldsymbol{\theta}_t^0 | D_t$, $\boldsymbol{\theta}_t^+ | D_t$;
end
```

Algorithm 3.2: Exception detection and handling routine.

3.3.4 Overdispersion

As previously mentioned, one problem with using Poisson distributions for modeling count time series is that in many instances these present overdispersion. This is appreciated in Figure 3.7 through the inability of the model to adequately forecast infrequent values, specially higher ones, resulting in under-coverage of the predictive distributions (Figure 3.8). One way to alleviate this, while assuming Poisson distributed observations, is proposed for DCMM, Berry and West (2020) and consists of using *random effects*, a discount factor $\rho \in (0, 1]$ in the variance of the linear predictor λ_t, so that its previous variance q_t is changed by $q_t^* \equiv q_t/\rho$.

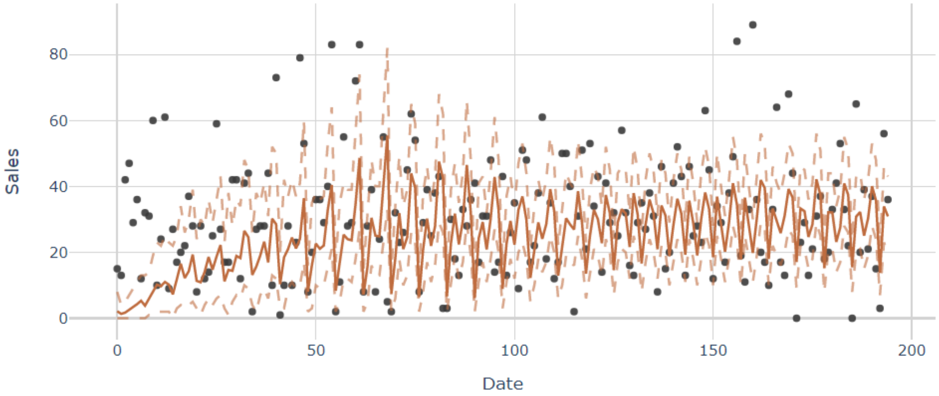


Figure 3.7: One day ahead predictions (solid) with 95% credible intervals (dashed) and real values (dots) for SKU '182' with DCMM.

This works particularly well for high values of q_t, for which $\gamma(\alpha_t) \approx \log(\alpha_t)$ and $\dot{\gamma}(\alpha_t)$ is a good approximation (West and Harrison (1997) ch. 14) and the mean of the forecast distribution remains the same while the variance is increased. However, this is not always a good approximation in practice and can affect the predictive mean. Moreover, even low random effects values might continue to inadequately forecast infrequently high values in our domain, e.g., Figure 3.8 shows under-coverage with a fairly low *rho* value of 0.3 at virtually all intervals; and Figure 3.7 the particular under-coverage of the 95% credible intervals (0.7 empirical coverage).

The use of a NB distribution for the positive counts in models (3.3, 3.5) with the estimation of parameter r_t indicating the grade of dispersion improves the forecasts over the same model (i.e. same $\boldsymbol{F}_t$, $\boldsymbol{G}_t$) with a Poisson in overdispersed products without compromising model performance for equidispersed products. This approach (Figure 3.9) also improves over the use of the Poisson with *random effects* (DCMM), which offers similar performance for point forecasts but is sometimes unable to completely remove the under-coverage of the predictive distributions shown in Figure 3.8.

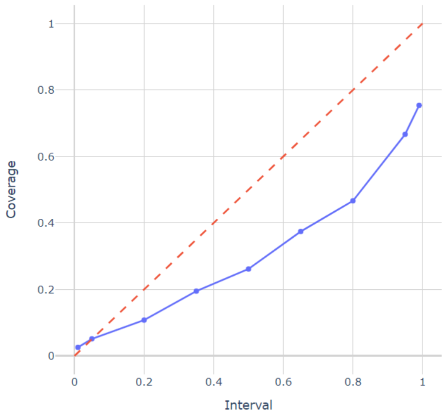


Figure 3.8: Coverage plot for SKU '182' using DCMM with a low *random effects*.

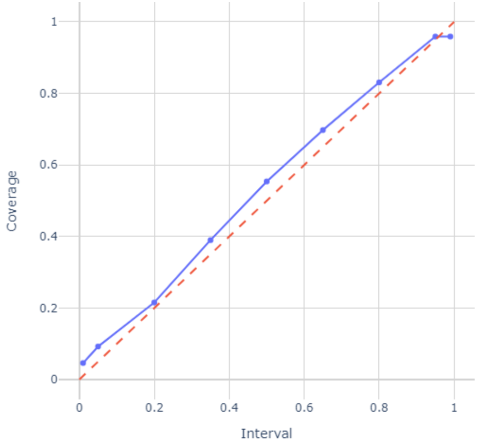


Figure 3.9: Coverage plot for SKU '182' using model (3.3).

3.4 The multivariate case

Figure 3.3 illustrated that there could be dependence between demand time series, suggesting relevant *correlation* due to common causes (environment, location,...) or substitute goods, for example. We expand now the model introduced in Section 3.3 to the multivariate case, so that we take advantage of cross-series information to improve the forecasts of individual or aggregated time series. As an example, demand forecasts of a product like a given beer can benefit from information on sales from other beers with different SKUs:

it is common for them to exhibit significant positive correlation as Figure 3.3 shows (and even negative in case of products *cannibalizing* sales). Indeed, product sales usually show marked seasonalities which can pass unnoticed for the univariate model in case of low demand individual series, but might show on related series with higher demand. Additionally, in many cases we are also interested in a subset of time series from a hierarchy level that share similar characteristics, like cake sales in stores situated in neighborhoods with analogous socio-economic indicators. The multivariate modeling of these time series can improve the forecast performance as the case study in Section 3.6 will show.

In the multivariate framework, let us denote by y_{it} the observation of the i-th time series ($i = 1, \ldots, m$) at time t. Defining $z_{it} = \mathbb{1}_{(y_{it}>0)}$, it is possible to extend model (3.3) to the multivariate case with

$$z_{it} \sim Ber(\pi_{it}) \quad \text{and} \quad y_{it}|z_{it} = \begin{cases} 0, & \text{if } z_{it} = 0, \\ 1 + x_{it}, \quad x_{it} \sim \textit{Neg-Bin}(r_t, \mu_{it}) & \text{if } z_{it} = 1, \end{cases} \tag{3.8}$$

which correspond to the marginals of the model $\boldsymbol{z}_t = \boldsymbol{x}_t \circ (\boldsymbol{y}_t + \mathbf{1}_m)$ with $\boldsymbol{z}_t \sim MBer(\boldsymbol{p}_t)$ and $\boldsymbol{x}_t \sim MNB(r_t, \boldsymbol{\mu}_t)$. The parameter of the multi-Bernoulli (MBer) is a vector $\boldsymbol{p}_t = (p_{00..00,t}, p_{00..01,t}, ..., p_{11..11,t})$ of dimension 2^m indicating the probabilities of each possible outcome (mutually exclusive events adding to one, and $\pi_{it} = \sum_{x\in\{0,1\}} p_{xx..1..xx}$, where 1 is in the i-th position). The Multi-Negative Binomial (MNB) has as parameters, a scalar r_t indicating the dispersion of all marginals, and vector $\boldsymbol{\mu}_t = (\mu_{1t}, ..., \mu_{mt})$ with the means of each time series. Alternatively, in matrix notation, $\boldsymbol{Y} = \boldsymbol{Z} \circ (\boldsymbol{X} + \mathbf{1}_{m\mathrm{x}T})$, where $\circ$ is the Hadamard product, that is,

$$\boldsymbol{Y} = \begin{pmatrix} z_{11} & \dots & \overbrace{z_{1t}}^{\boldsymbol{z}_t} & \dots & z_{1T} \\ \vdots & & \vdots & & \vdots \\ z_{i1} & \dots & z_{it} & \dots & z_{iT} \\ \vdots & & \vdots & & \vdots \\ z_{m1} & \dots & z_{mt} & \dots & z_{mT} \end{pmatrix} \circ \left(\begin{pmatrix} x_{11} & \dots & \overbrace{x_{1t}}^{\boldsymbol{x}_t} & \dots & x_{1T} \\ \vdots & & \vdots & & \vdots \\ x_{i1} & \dots & x_{it} & \dots & x_{iT} \\ \vdots & & \vdots & & \vdots \\ x_{m1} & \dots & x_{mt} & \dots & x_{mT} \end{pmatrix} + \boldsymbol{1}_{m\,\mathrm{x}\,T} \right).$$

We have therefore, as in the univariate case, two DGLMs

$$\begin{aligned} &\boldsymbol{z}_t \sim MBer(\boldsymbol{p}_t), && \boldsymbol{x}_t \sim MNB(r_t;\boldsymbol{\mu}_t), \\ &softmax^{-1}(\boldsymbol{p}_t) = \boldsymbol{F}_t^0\boldsymbol{\theta}_t^0, && \log(\boldsymbol{\mu}_t) = \boldsymbol{F}_t^+\boldsymbol{\theta}_t^+, \\ &\boldsymbol{\theta}_t^0 = \boldsymbol{G}_t^0\boldsymbol{\theta}_{t-1}^0 + \boldsymbol{\omega}_t^0\,,\ \ \boldsymbol{\omega}_t^0 \sim (\boldsymbol{0},\boldsymbol{W}_t^0), && \boldsymbol{\theta}_t^+ = \boldsymbol{G}_t^+\boldsymbol{\theta}_{t-1}^+ + \boldsymbol{\omega}_t^+\,,\ \ \boldsymbol{\omega}_t^+ \sim (\boldsymbol{0},\boldsymbol{W}_t^+), \\ &\boldsymbol{\theta}_0^0 \sim (\boldsymbol{m}_0^0,\boldsymbol{C}_0^0), && \boldsymbol{\theta}_0^+ \sim (\boldsymbol{m}_0^+,\boldsymbol{C}_0^+), \end{aligned} \tag{3.9}$$

softmax^{-1} being the inverse of the *softmax* function[1]. This joint model introduces dependence across the states of time series y_{1t},..., y_{mt}, improving forecast performance by *borrowing strength*, i.e. for each y_{it} we exploit the information provided by the other $m-1$ similar time series. Since both distributions, *MBer* for the *sale/no sale* part, and *MNB* (with fixed r_t) for the *number of sales* part belong to the exponential family we can use conjugate analysis for the sequential updating and forecast procedure of the model. This can be done in parallel for each DGLM as detailed below.

***Sale/no-sale* part. Multi-Bernoulli DGLM** For the the *sale/no-sale* part of model (3.8), we have the binary observation vector $\boldsymbol{z}_t$ generated from the original time series, which is modeled through the multivariate Bernoulli (MBer) DGLM with *softmax*$^{-1}$ link function,

---

[1] The *softmax* function is a generalization of the logistic function. Its inverse *softmax*$^{-1}$ generalizes the logit function, $softmax^{-1}(\boldsymbol{p}_t) = (\log(p_{00..00,t}/p_{11..11,t})$, ..., $\log(p_{01..11,t}/p_{11..11,t}), 0)$.

$$
\begin{aligned}
&\boldsymbol{z}_t \sim MBer(\boldsymbol{p}_t),\\
&softmax^{-1}(\boldsymbol{p}_t) = \boldsymbol{F}_t\boldsymbol{\theta}_t,\\
&\boldsymbol{\theta}_t = \boldsymbol{G}_t\boldsymbol{\theta}_{t-1} + \boldsymbol{\omega}_t \quad \text{with} \quad \boldsymbol{\omega}_t \sim (\boldsymbol{0}, \boldsymbol{W}_t),\\
&\boldsymbol{\theta}_0 \sim (\boldsymbol{m}_0, \boldsymbol{C}_0),
\end{aligned}
$$

with MBer following the definition in Dai et al. (2013), and $\boldsymbol{p}_t = (p_{00..00,t}, p_{00..01,t}, ..., p_{11..11,t})$ is a vector of dimension 2^m indicating the probabilities of each possible outcome, with $\sum_{k\in\{00..00,...,11..11\}} p_{k,t} = 1$. Note that, although 2^m can grow quickly, we are usually interested in modeling a limited number of time series together (and with a similar dispersion due to the particularities of the MNB), and therefore, the computations remain tractable. Using the exponential family notation in (3.1) we have

$$
T(\boldsymbol{z}_t) = \begin{pmatrix} (1-z_{1,t})...(1-z_{m-1,t})(1-z_{m,t}) \\ (1-z_{1,t})...(1-z_{m-1,t})z_{m,t} \\ \vdots \\ z_{1,t}...z_{m-1,t}z_{m,t} \end{pmatrix}, \qquad V_t = 1 \implies \Phi_t := V_t^{-1} = 1,
$$

$$
\boldsymbol{\eta}_t = softmax^{-1}(\boldsymbol{p}_t), \qquad a(\boldsymbol{\eta}_t) = \log\Big(\sum_k e^{\eta_{k,t}}\Big), \qquad b(\boldsymbol{z}_t, V_t) = 1.
$$

Then, the conjugate prior for the linear predictor, $softmax^{-1}(\boldsymbol{p}_t)$, is

$$
\begin{aligned}
CP_{\boldsymbol{\eta}_t}(\boldsymbol{\alpha}_t, \beta_t) = \pi(\boldsymbol{\eta}_t|D_{t-1}) &= c(\boldsymbol{\alpha}_t, \beta_t)\exp(\boldsymbol{\alpha}_t\boldsymbol{\eta}_t - \beta_t a(\boldsymbol{\eta}_t)) \qquad (3.10)\\
&= c(\boldsymbol{\alpha}_t, \beta_t)\exp\Big(\boldsymbol{\alpha}_t\boldsymbol{\eta}_t - \beta_t\log(\sum_k e^{\eta_{k,t}})\Big)\\
&= c(\boldsymbol{\alpha}_t, \beta_t)\,\frac{e^{\boldsymbol{\alpha}_t\boldsymbol{\eta}_t}}{(\sum_k e^{\eta_{k,t}})^{\beta_t}} = c(\boldsymbol{\alpha}_t, \beta_t)\,\frac{e^{\sum_k \alpha_{k,t}\eta_{k,t}}}{(\sum_k e^{\eta_{k,t}})^{\beta_t}},
\end{aligned}
$$

for some normalizing constant $c(\boldsymbol{\alpha}_t, \beta_t)$. As the distribution $CP_{\boldsymbol{\eta}_t}(\boldsymbol{\alpha}_t, \beta_t)$ must integrate to 1, the normalizing constant is

$$c(\boldsymbol{\alpha}_t, \beta_t) = \left(\int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} \frac{e^{\sum_k \alpha_{k,t}\eta_{k,t}}}{(\sum_k e^{\eta_{k,t}})^{\beta_t}} \, d\eta_{00..00,t} \dots d\eta_{01..11,t} \right)^{-1},$$

which, with the change of variable $\boldsymbol{p}_t = \mathit{softmax}(\boldsymbol{\eta}_t)$, is equivalent to

$$\left(\int_0^1 \dots \int_0^{\varphi} \frac{\left(\frac{p_{00..00,t}}{p_{11..11,t}}\right)^{\alpha_{00..00,t}} \dots \left(\frac{p_{01..11,t}}{p_{11..11,t}}\right)^{\alpha_{01..11,t}} \left(\frac{1}{p_{11..11,t}}\right)^{1-\beta_t}}{p_{00..00,t} \dots p_{01..11,t}} \, dp_{00..00,t} \dots dp_{01..11,t} \right)^{-1}$$

$$= \frac{\Gamma(\beta_t)}{\Gamma(\alpha_{00..00,t}) \dots \Gamma(\alpha_{01..11,t}) \Gamma(b - \alpha_{00..00,t} - \dots - \alpha_{01..11,t})}.$$

where $\varphi = 1 - p_{00..01,t} - \dots - p_{01..11,t}$ and $p_{11..11,t} = 1 - p_{00..00,t} - \dots - p_{01..11,t}$. Therefore, from (3.10), we get that the conjugate prior $\boldsymbol{p}_t$ is

$$CP_{\boldsymbol{p}_t}(\boldsymbol{\alpha}_t, \beta_t) = \mathit{Dirichlet}(\alpha_{00..00,t}, \dots, \alpha_{01..11,t}, \beta_t - \alpha_{00..00,t} - \dots - \alpha_{01..11,t}).$$

The hyperparameters $\alpha_{00..00,t}$,..., $\alpha_{01..11,t}, \beta_t$ are estimated with the moment method, equalizing the mean and variance of the linear predictor $\mathit{softmax}^{-1}(\boldsymbol{p}_t)$ to $\boldsymbol{f}_t = \boldsymbol{F}_t \boldsymbol{a}_t$ and $\boldsymbol{Q}_t = \boldsymbol{F}_t \boldsymbol{R}_t \boldsymbol{F}_t{}'$ respectively:

$$E[\mathit{softmax}^{-1}(\boldsymbol{p}_t)|D_{t-1}] = \begin{pmatrix} \gamma(\alpha_{00..00,t}) - \gamma(b) \\ \vdots \\ \gamma(\alpha_{01..11,t}) - \gamma(b) \\ 0 \end{pmatrix},$$

$$Var[\mathit{softmax}^{-1}(\boldsymbol{p}_t)|D_{t-1}] =$$

$$\begin{pmatrix} \dot\gamma(\alpha_{00..00,t}) + \dot\gamma(b) & \dot\gamma(b) & \dot\gamma(b) & \cdots & \dot\gamma(b) & 0 \\ \dot\gamma(b) & \dot\gamma(\alpha_{00..01,t}) + \dot\gamma(b) & \dot\gamma(b) & \cdots & \dot\gamma(b) & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \dot\gamma(b) & \dot\gamma(b) & \dot\gamma(b) & \cdots & \dot\gamma(\alpha_{01..11,t}) + \dot\gamma(b) & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 \end{pmatrix},$$

where $b = \beta_t - \alpha_{00..00,t} - ... - \alpha_{01..11,t}$. Using the approximation $\dot{\gamma}(x) = \log(x) - \frac{1}{2x}$ we obtain good approximations for the hyperparameters which we can use as such, or as starting points for a Newton-Raphson algorithm to obtain even more accurate values.

The predictive distribution for the observation vector $\boldsymbol{z}_t$ is

$$\begin{aligned}\pi(\boldsymbol{z}_t|\boldsymbol{\alpha}_t, \beta_t) &= \frac{c(\boldsymbol{\alpha}_t, \beta_t)b(z_t, V_t)}{c(\boldsymbol{\alpha}_t + \phi_t T(\boldsymbol{z}_t), \beta_t + \phi_t)} \qquad (3.11)\\ &= MBer\Big(\frac{\alpha_{00..00,t}}{\beta_t}, ..., \frac{\alpha_{01..11,t}}{\beta_t}, \frac{\beta_t - \alpha_{00..00,t} \cdots - \alpha_{01..11,t}}{\beta_t}\Big).\end{aligned}$$

After observing the realization $\boldsymbol{z}_t$, the posterior distributions for $\boldsymbol{\eta}_t$ and $\boldsymbol{p}_t$ have the same form than the priors with updated hyperparameters $\widehat{\boldsymbol{\alpha}}_t = \boldsymbol{\alpha}_t + \phi_t T(\boldsymbol{z_t})$, $\widehat{\beta}_t = \beta_t + \phi_t$.

***Number of sales* part. Multi-Negative Binomial DGLM** For the the number of sales part in model (3.8), we have the observation vector $\boldsymbol{x}_t$ modeled through a DGLM with the multivariate negative binomial (MNB) in Arbous and Kerrich (1951) (Appendix A)

$$\begin{aligned}&\boldsymbol{x}_t \sim MNB(r_t; \boldsymbol{\mu}_t)\\ &\log(\boldsymbol{\mu}_t) = \boldsymbol{F}_t\boldsymbol{\theta}_t,\\ &\boldsymbol{\theta}_t = \boldsymbol{G}_t\boldsymbol{\theta}_{t-1} + \boldsymbol{\omega}_t \quad \text{with} \quad \boldsymbol{\omega}_t \sim (\boldsymbol{0}, \boldsymbol{W}_t),\\ &\boldsymbol{\theta}_0 \sim (\boldsymbol{m}_0, \boldsymbol{C}_0),\end{aligned}$$

with $\boldsymbol{\mu}_t = (\mu_{1,t}, ..., \mu_{m,t})$ being the vector of means of each time series. Using the exponential family notation in (3.1) we have

$$T(\boldsymbol{x}_t) = \boldsymbol{x}_t, \qquad V_t = 1 \implies \Phi_t := V_t^{-1} = 1, \qquad \boldsymbol{\eta}_t = \begin{pmatrix}\log(\frac{\mu_{1,t}}{r_t + \sum_k \mu_{k,t}})\\ \vdots \\ \log(\frac{\mu_{m,t}}{r_t + \sum_k \mu_{k,t}})\end{pmatrix},$$

$$a(\boldsymbol{\eta}_t) = -r_t \log\Big(1 - \sum_k e^{\eta_{k,t}}\Big), \qquad b(\boldsymbol{x}_t, V_t) = \frac{\Gamma(r + \sum_k y_{k,t})}{\Gamma(r_t)\prod_k y_{k,t}!}.$$

The conjugate prior for the natural parameter $\boldsymbol{\eta}_t$ is

$$\begin{aligned} CP_{\boldsymbol{\eta}_t}(\boldsymbol{\alpha}_t, \beta_t) &= \pi(\boldsymbol{\eta}_t|D_{t-1}) = c(\boldsymbol{\alpha}_t, \beta_t)\exp(\boldsymbol{\alpha}_t\boldsymbol{\eta}_t - \beta_t a(\boldsymbol{\eta}_t)) \\ &= c(\boldsymbol{\alpha}_t, \beta_t)\exp\Big(\boldsymbol{\alpha}_t\boldsymbol{\eta}_t + r_t\beta_t \log\Big(1 - \sum_k e^{\eta_{k,t}}\Big)\Big), \end{aligned} \tag{3.12}$$

for some normalizing constant $c(\boldsymbol{\alpha}_t, \beta_t)$. As the distribution $CP_{\boldsymbol{\eta}_t}(\boldsymbol{\alpha}_t, \beta_t)$ must integrate to 1, the normalizing constant is

$$c(\boldsymbol{\alpha}_t, \beta_t) = \left(\int_{-\infty}^{0} .. \int_{-\infty}^{0} \frac{\exp(\eta_{1,t}\alpha_{1,t} .. + \eta_{m,t}\alpha_{m,t})(1 - \exp(\eta_{1,t}) .. - \exp(\eta_{1,t})}{(\sum_k e^{\eta_{k,t}})^{\beta_t}} \, d\eta_{1,t} .. d\eta_{m,t}\right)^{-1},$$

which, with the change of variable $p_{i,t} = \frac{\mu_{i,t}}{r+\mu_{i,t}}$, is the multivariate beta function $(B(\alpha_{1t}, ..., \alpha_{mt}, \beta_t r_t + 1))^{-1}$.

Then, from (3.12) we get that the conjugate prior for the parameter $\boldsymbol{\mu}_t$ is

$$CP_{\boldsymbol{\mu}_t}(\boldsymbol{\alpha}_t, \beta_t) = \frac{r_t\left(\frac{\mu_{1t}}{r_t+\sum_k \mu_{kt}}\right)^{\alpha_1 - 1} ... \left(\frac{\mu_{mt}}{r_t+\sum_k \mu_{kt}}\right)^{\alpha_m - 1}\left(\frac{r_t}{r_t+\sum_k \mu_{kt}}\right)^{\beta_t + 1 - 1}}{B(\alpha_{1t}, ..., \alpha_{mt}, \beta_t r_t + 1)(r_t + \sum_k \mu_{kt})^{m+1}}.$$

The hyperparameters α_{1t},..., α_{mt}, β_t are estimated through the moment method, equalizing the mean and variance of the linear predictor $\log \boldsymbol{\mu}_t$ to $\boldsymbol{f}_t = \boldsymbol{F}_t\boldsymbol{a}_t$ and $\boldsymbol{Q}_t = \boldsymbol{F}_t\boldsymbol{R}_t\boldsymbol{F}_t'$, respectively:

$$E[\log(\boldsymbol{\mu}_t)|D_{t-1}] = \begin{pmatrix} \gamma(\alpha_{1t}) - \gamma(\beta_t r + 1) + \log(r) \\ \vdots \\ \gamma(\alpha_{mt}) - \gamma(\beta_t r + 1) + \log(r) \end{pmatrix},$$

$$Var[\log(\boldsymbol{\mu}_t)|D_{t-1}] = \begin{pmatrix} \dot{\gamma}(\alpha_{1t}) + \dot{\gamma}(\beta r + 1) & \cdots & \dot{\gamma}(\beta r + 1) \\ \vdots & \ddots & \vdots \\ \dot{\gamma}(\beta r + 1) & \cdots & \dot{\gamma}(\alpha_{mt}) + \dot{\gamma}(\beta r + 1) \end{pmatrix}.$$

The predictive distribution for the observation vector $\boldsymbol{x}_t$ is

$$
\begin{aligned}
\pi(\boldsymbol{x}_t|\boldsymbol{\alpha}_t,\beta_t) &= \frac{c(\boldsymbol{\alpha}_t,\beta_t)b(\boldsymbol{x}_t,V_t)}{c(\boldsymbol{\alpha}_t+\phi_t T(\boldsymbol{x}_t),\beta_t+\phi_t)} \qquad (3.13)\\
&= \frac{B(\alpha_{1t}+y_{1t},...,\alpha_{mt}+y_{mt},(\beta_t+1)r_t+1)}{B(\alpha_{1t},...,\alpha_{mt},\beta_t r_t+1)}\frac{\Gamma(r+\sum_k y_{kt})}{\Gamma(r_t)\prod_k y_{kt}!}\\
&\equiv MBNB(\boldsymbol{x}_t|\boldsymbol{\alpha}_t,\beta_t,r_t),
\end{aligned}
$$

which we denote as Multivariate Beta Negative Binomial (MBNB) due to its marginals being Beta Negative Binomial (BNB) distributions. After observing the realization $\boldsymbol{x}_t$, the posterior distributions for $\boldsymbol{\eta}_t$ and $\boldsymbol{\mu}_t$ have the same form than the priors but with updated hyperparameters $\widehat{\boldsymbol{\alpha}}_t = \boldsymbol{\alpha}_t + \phi_t T(\boldsymbol{x}_t)$, $\widehat{\beta}_t = \beta_t + \phi_t$.

Therefore, the resulting predictive distribution of the joint mixture model (3.8) for the observations vector $\boldsymbol{y}_t$ is

$$
\boldsymbol{y}_t|D_{t-1}{=}\boldsymbol{z}_t{\circ}(\boldsymbol{x}_t{+}\mathbf{1}_m) \text{ with } \begin{cases} \boldsymbol{z}_t \sim MBer\Big(\frac{\alpha^0_{00..00,t}}{\beta^0_t},...,\frac{\alpha^0_{01..11,t}}{\beta^0_t},\frac{\beta^0_t-\alpha^0_{00..00,t}\cdot\cdot-\alpha^0_{01..11,t}}{\beta^0_t}\Big),\\ \boldsymbol{x}_t \sim MBNB(\boldsymbol{\alpha}^+_t,\beta^+_t,r_t), \end{cases}
$$

which correspond to the predictive distribution for the marginals $y_{it}|D_{t-1} = z_{it}(x_{it}{+}1_m)$, with z_{it} a Bernoulli and x_{it} a beta negative binomial, the marginals of (3.11) and (3.13) respectively. This can be written analogously to (3.7) since

$$
p(y_{it}|D_{t-1},\pi_{it}) = (1-\pi_{it})\delta_0(y_{it}) + \pi_{it}BNB(y_{it}-1|\beta^+_t r_t+1,\alpha^+_{it},r_t),
$$

with $(\pi_{it}|D_{t-1}) \sim Be((\beta^0_t - \sum_{x\in\{0,1\}}\alpha^0_{xx..0..xx})/\beta^0_t)$ and δ_0 the Kronecker delta function. From this expression, we can obtain the median to use as a point forecast and calculate credible intervals, computing the corresponding percentiles ρ as

$$\begin{cases} 0, & \text{if } (1-\mu_{z_{it}}) \geq \rho, \\ Q_{BNB}((\rho - (1-\mu_{z_{it}}))/\mu_{z_{it}}; \beta_t^+ r_t + 1, \alpha_{it}^+, r_t) + 1, & \text{otherwise,} \end{cases}$$

where $\mu_{z_{it}}$ is the mean of the i-th marginal of the predictive distribution (3.11), and Q_{BNB} denotes the quantile of the i-th marginal of the predictive distribution (3.13).

Although the equations in this section correspond to the multivariate version of *hurdle shifted* model (3.3), they can easily be modified as in Section 3.3 to obtain the ones for the multivariate *zero inflated* model (3.5). We explore the performance of both versions in Section 3.6.

3.5 Out-of-stock events

The information about the likelihood of OoS events is essential in inventory management to support decisions concerning when to place resupply orders. It facilitates devising a real time monitoring algorithm to raise alarms that prompts (or automatically places) orders whenever a potentially *critical* situation is predicted.

With the information about current stock and arrival dates of replenishment orders, and the demand forecast up to the $(t+k)$-th time, it is possible to obtain the probability that a product becomes out of stock over the next k periods. Indeed, the stock at the end of period t for any given product is

$$\mathit{stock}_t = \mathit{stock}_{t-1} - \mathit{sales}_t + \mathit{resupply}_t.$$

Observe that in our specific application domain, the resupplies arrive to the store at the beginning of the period (day); hence, those units can be used to satisfy demand in that same period or day t. We consider $\mathit{resupply}_t$ a time series of non-negative integers (zero when there is no resupply expected) that is updated whenever a new order is made. If we consider $\mathit{resupply}_t$ as known, which is reasonable as it is based on our requests to own or external suppliers with firm lead times, we can use the predictive distribution of demand to estimate the probability of an OoS event at time t, $P(stock_t = 0)$, as

$$P(demand_t \geq stock_{t-1} + resupply_t),$$

where we note that $sales_t = \min(stock_t, demand_t)$.

For estimating the probability of an OoS event, from the end of the current time period t in which we have already observed D_t up to time k, we use the predictive distributions of the demand obtained via simulation through Algorithm 3.1: we have approximations for $p(y_{t+1}|D_t), ..., p(y_{t+k}|D_t)$, and, for any $i \in \{1, ..., k\}$, the cumulative demand up to that instant ahead, $d_i|D_t := (y_{t+1} + ... + y_{t+i}) \,|\, D_t$, can be approximated with the previous samples. Thus, we can use it to estimate the probability of an OoS event at or before time $t + i$, until the next resupply,

$$\begin{aligned} p(OoS_i) &:= 1 - \sum_{j=0}^{stock_t} p(y_{t+1} + ... + y_{t+i} = j|D_t), \\ &= 1 - \sum_{j=0}^{stock_t} p(d_i = j|D_t). \end{aligned}$$

Now, at the current time period t, the previous estimations of the probabilities of OoS events up to some horizon $t + k$ can be used to make an informed decision on whether to place an order or not (and when we want that order to arrive). This can be done according to several criteria, including:

- We place an order before the forecast of the probability of an OoS event reaches a certain threshold α, that is, if we have the set with the time periods that surpass the threshold, $A = \{i \in [1, ..., k] : P(OoS_i) \geq \alpha\}$, and the criterion is,

- If the set A is empty, then, with the chosen threshold α we are not worried about OoS events up to $t+k$
- Otherwise, we are prompted to make an order that arrives before or at time period $t+\min(A)$

In case the retailer wants to completely avoid OoS events, the threshold value α should be low.

- If there is more information that allows calculating the expected benefits (sell prices, storage costs, etc), we place an order if that monetary profit is greater than the alternative decision (no order). For that, if the profit per unit sold is c, and the reputation loss (expressed in monetary terms) per unit not sold is f, we define the profit at time $t+i$ ($i = 1, ..., k$) as

  $$(1 - P(OoS_i)) \times (d_i\, c) - P(OoS_i) \times (d_i - stock_t) f.$$

  Now, since we have the predictive distribution of the demand $d_i|D_t$ up to time $t+i$, if we denote $\widehat{d_i} := E[d_i|D_t]$ and the expected profit as

  $$EProfit_i := (1 - P(OoS_i)) \times (\widehat{d_i}\, c) - P(OoS_i) \times (\widehat{d_i} - stock_t) f,$$

  then, the set with the instants with negative expected profit is $A = \{i \in [1, ..., k] : EProfit_i < 0\}$. Analogously to the previous criteria, if the set A is not empty, we would be prompted to make an order that arrives at time $t+\min(A)$ or before.

- Via the use of utility functions that take into account risk aversion. In this case, we would have the utility at time $t+i$,

  $$(1 - P(OoS_i)) \times u(d_i\, c) - P(OoS_i) \times u((d_i - stock_t) f),$$

where u is the corresponding utility function. Note that if the utility function u is the identity, we would be in the previous case. The procedure is analogous, we would have a expected utility

$$EUtil_i := (1 - P(OoS_i)) \times E[u(d_i\,c)] - P(OoS_i) \times E[u((d_i - stock_t)f)],$$

and a corresponding set of instants for which the expected utility is negative, $A = \{i \in [1, ..., k] : EUtil_i < 0\}$. The user would be prompted to make an order if that set A is not empty.

In any case, with any of the above criteria applied to the time series of a product, there are two possible results: if the set A is empty, the user is not recommended to place an order; otherwise, the user is prompted to place an order arriving at or before the time period $t + \min(A)$.

As mentioned, the recommendation to make an order is done generating an alarm, which can adopt two levels, warning and critical. This is done based on the number of time periods left (window of opportunity) to take an action to prevent the OoS situation, i.e., until $t + \min(A)$ time period. As an example, in the proposed application of forecasting daily demand: we would issue a *warning level* when there are 3 or more days left until $t + \min(A)$, and a *critical level* when there are only 1 or 2 days left until $t + \min(A)$.

If the user is prompted to make an order now (i.e., at time period t), he can chose to do it immediately, or wait (specially if the alarm is of the *warning level* type) for more products requiring an order. If the order for the product is performed, the corresponding resupply time series would be updated and used when facing the ordering decision process at the next time period $t + 1$ (tomorrow in this case).

Section 3.6 illustrates this methodology, with the threshold technique, applied to the stock of beer during a week into the future.

3.6 Case study. Supermarket sales forecasting

The data in our case study consists of the fourteen time series of daily sales introduced in Section 3.2. We use the models in Sections 3.3 and 3.4 to obtain forecasts and compare its performance with other models for count time series.

3.6.1 Model specification

To apply the models in Sections 3.3 and 3.4, we first specify the $\boldsymbol{F}_t$ and $\boldsymbol{G}_t$ matrices. For the *sale/no sale* part of the univariate models (3.3, 3.5), Bernoulli DGLM, we include a second order polynomial, or linear growth, component with

$$\boldsymbol{F}_t^0 = \begin{pmatrix} 1 & 0 \end{pmatrix}, \qquad \boldsymbol{G}_t^0 = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}. \tag{3.14}$$

The *number of sales* part of the model includes also a linear growth component, plus a seasonal one of period 7, and two covariates referring to the price logarithm and a three level promotion variable (as discussed in Section 3.2),

$$\boldsymbol{F}_t^+ = \Big(\overbrace{1 \quad 0}^{\substack{\text{Linear}\\ \text{growth}}} \quad \overbrace{\log(price_t) \quad promo_t}^{\text{Covariates}} \quad \overbrace{1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0}^{\text{Sesaonal}} \Big),$$

$$\boldsymbol{G}_t^+ = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & -1 & -1 & -1 & -1 & -1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}. \tag{3.15}$$

For the multivariate version (3.9), the same components are used for $\boldsymbol{F}_t$ and $\boldsymbol{G}_t$ matrices, but with the dependence structure among state vectors given by Seemingly Unrelated Time Series Equations (SUTSE) as in Fernández and Harvey (1990). This joint model introduces dependence across the states of time series $y_{1t} \dots y_{mt}$, via the evolution errors $\boldsymbol{\omega}_t^0$ and $\boldsymbol{\omega}_t^+$. The new matrices are therefore the result of implementing the Kronecker product[2], $\otimes$, between those in (3.14) and (3.15), and the identity matrix of dimension m, where m is the number of time series incorporated into the multivariate model

$$\boldsymbol{F}_t^0 \otimes \boldsymbol{I}_m\,, \quad \boldsymbol{F}_t^+ \otimes \boldsymbol{I}_m\,, \quad \boldsymbol{G}_t^0 \otimes \boldsymbol{I}_m\,, \quad \boldsymbol{G}_t^+ \otimes \boldsymbol{I}_m.$$

State vectors will also change from dimension q to dimension qm, with $q = 2$ for the *sale/no sale* part (3.14) and $q = 10$ for the *number of sales* part (3.15),

$$\begin{pmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_q \end{pmatrix} \longrightarrow \begin{pmatrix} \theta_{1,1} \\ \vdots \\ \theta_{m,1} \\ \theta_{1,2} \\ \vdots \\ \theta_{m,2} \\ \vdots \\ \theta_{m,q} \end{pmatrix}. \tag{3.16}$$

As mentioned in Section 3.3, discount factors are used due to the advantages they present. After analyzing and modeling several representative data sets, we decided that the most adequate component discount factors were $\delta^0 = 0.95$ for the *sale/no sale* part (whether it is Bernoulli or MultiBernoulli), and $\boldsymbol{\delta}^+ = (0.99, 0.995, 0.995, 0.995)$ for the level, regression and seasonal components of the *number of sales* term (NB or MNB). Finally, the dispersion parameter r_t

---

[2] If $\boldsymbol{A}$ is an $m \times n$ matrix and $\boldsymbol{B}$ is a $p \times q$ matrix, then the Kronecker product $\boldsymbol{A} \otimes \boldsymbol{B}$ is the $pm \times qn$ block matrix, $\begin{pmatrix} a_{11}\boldsymbol{B} & \dots & a_{11}\boldsymbol{B} \\ \vdots & \ddots & \vdots \\ a_{m1}\boldsymbol{B} & \dots & a_{mn}\boldsymbol{B} \end{pmatrix}$.

is estimated using the EM algorithm, as in Adamidis (1999), with the first 14 observations (two weeks).

3.6.2 Prior information

To complete model specification, we require the prior moments for the state vector $\boldsymbol{\theta}$. Our priors aim to be flexible enough and somewhat informative. They are estimated using the first week of data. For the Bernoulli, the prior moments adopted are:

$$\boldsymbol{m}_0^0 = \begin{pmatrix} \log(\widehat{p}/(1-\widehat{p})) & 0 \end{pmatrix}' \quad \text{and} \quad \boldsymbol{C}_0^0 = \boldsymbol{I}_2,$$

where $\widehat{p}$ is the proportion of the first 7 days in which there were sales. For the NB DGLM, we use

$$\boldsymbol{m}_0^+ = \begin{pmatrix} m_0^{L1} & m_0^{L2} & 0 & 0 & m_0^{S1} & m_0^{S2} & m_0^{S3} & m_0^{S4} & m_0^{S5} \end{pmatrix}' \quad \text{and} \quad \boldsymbol{C}_0^+ = \boldsymbol{I}_9,$$

where parameter m_0^{L2} describes the expected growth and it is initialized with $m_0^{L2} = (u_7 - u_1)/6$, with $u_t = \log(y_t)$; m_0^{L1} describes the expected level and is initialized with $m_0^{L1} = (\sum_{t=1}^{7} u_t - 28\, m_0^{L2})/7$; m_0^{Sj} describes the j-th seasonal component and, to assess it, we use $m_0^{Sj} = y_{8-j} - m_0^{L1} + (j-8)\, m_0^{L2}$. For the multivariate version (3.9), the prior mean vector is constructed to be consistent with the state structure in 3.16, i.e., first assess m_0^{L1} for each of the m time series, then the m_0^{L2}'s, etc.

We use the identity matrices, $\boldsymbol{I}_2$ and $\boldsymbol{I}_9$, as prior matrix variances for each of the model blocks, which allows the algorithm to adapt quickly in very different contexts. This is due to the fact that we try to build a general and automatic algorithm. However, in each individual case, performance could improve should there be more precise prior information.

3.6.3 Results

Univariate model

The proposed univariate model (3.3) in Section 3.3 significantly improves forecast performance over DCMM in Berry and West (2020), as can be seen by comparing Figure 3.7 with Figure 3.10. Note specially that the 95% credible intervals cover more adequately days with high demand.

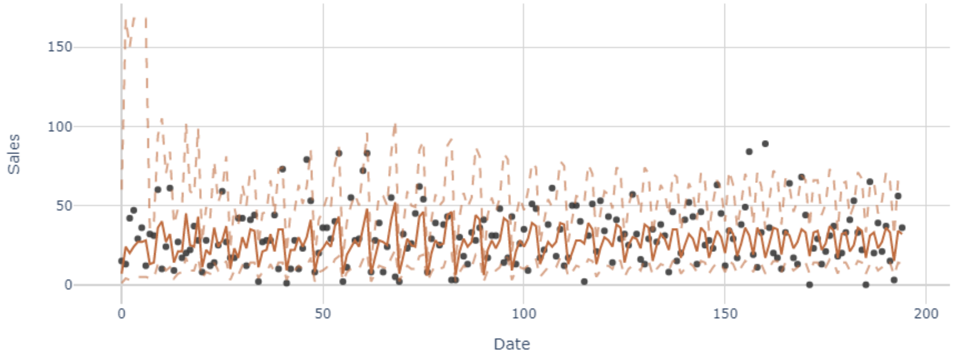


Figure 3.10: One day ahead predictions (solid) with 95% credible intervals (dashed) and real values (dots) for SKU '182' with model (3.3).

In fact, the predictive distributions of this model cover quite adequately the observations of the studied series at most intervals, as shown in Figure 3.11 with sales for alcoholic beers (SKU 182 and 14752).

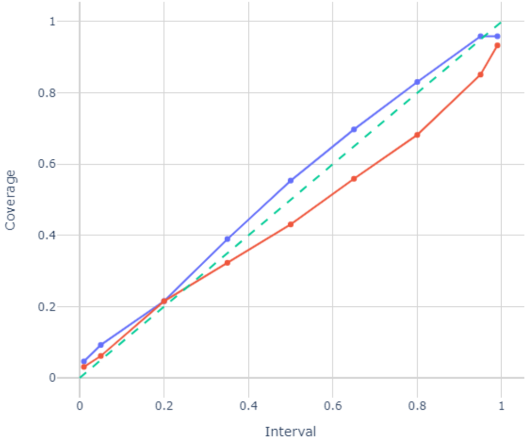


Figure 3.11: Coverage plots for SKU 182 beer (bue) and SKU 14752 beer (red).

Using the samples from the 7-step ahead predictive distributions obtained with Algorithm 3.1, we estimate the probability of observing OoS events for those 7 days ahead as explained in Section 3.5. Figure 3.12 shows that the OoS probability increases significantly the fourth day. Thus, using the first criteria to avoid OoS situations with threshold $\alpha = 0.75$, it is recommended to place a restock order such that it arrives before the end of the 4-th day. Additionally, conforming to that criteria and the *two level* notification types, a *warning level* notification is issued since there are still four days left to reach the critical threshold, $t + \min(A)$ in Section 3.5.

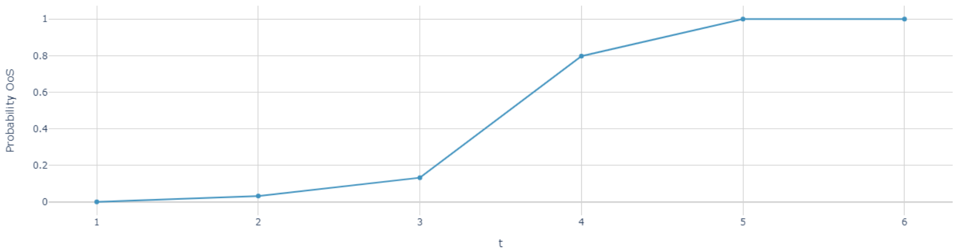


Figure 3.12: Out of Stock probability for beer (SKU '182') for next week.

Multivariate model

The multivariate model (3.4), that uses cross-series dependencies tends to improve the performance over the univariate versions of our model, whether we use the *zero inflated* or the *hurdle shifted* variant. It is worth noting though that when modeling series where almost all observations are zero, like cleaning products (bathroom cleaner with SKU '70598' and detergent with SKU '130111' in Figure 3.13), we usually observe that the credible intervals cover the observation way better in the *zero inflated* case.

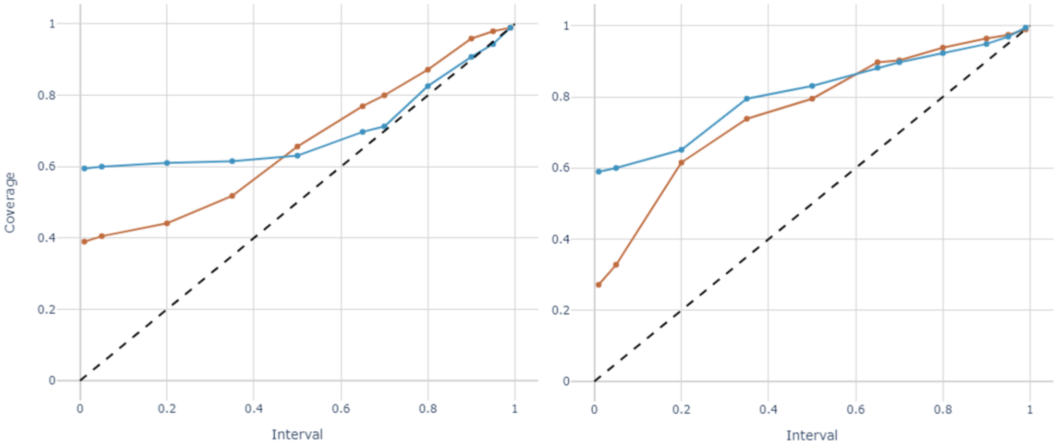


Figure 3.13: Coverage plots for bathroom cleaner in blue, and detergent in red. ZI version of (3.3) on the left, HS on the right.

Additionally a summary of the performance of point forecasts with different error metrics is shown in Table 3.2 for the same cleaning products. There, it can be seen that with these metrics, the multivariate model proposed in Section 3.4 outperforms a DCMM.

Error Metric	Zero Inflated Multiv.	Hurdle Shifted Multiv.	DCMM
MSE	(3.58 ,1.75)	(2.65,1.65)	(3.16,1.82)
MAE	(1.14 ,0.70)	(0.99,0.73)	(1.19,0.77)
ZAPE	(0.59, 0.41)	(0.56,0.46)	(0.69,0.45)
Theil's U	(0.87, 0.83)	(0.75,0.81)	(0.82,0.85)

Table 3.2: Point forecast error metrics for *zero inflated*, *hurdle shifted*, and DCMM applied to cleaning products (SKUs '70598', '130111').

3.7 Discussion

We have provided a family of models to forecast individual time series with frequent zeros and possible overdispersion, and a multivariate extension of the proposed model that takes advantage of cross-series dependencies, making better use of the available information (among products within a store, among a product at different stores, among stores, etc.). The models and methodology introduced are illustrated with a real demand forecasting problem, and shown to improve the performance of models commonly used in this application domain. In fact, are an essential part for any DSS in retail to support optimal decisions for the company.

Indeed, we present a methodology that with the forecasts of the models plus additional information (stock, keeping costs, etc in our application domain) aids in decision making, indicating when actions should be taken to avoid potentially critical situations.

Finally, though the architecture and methodology was inspired by a massive scale stock management problem, and developed as part of a Decision Support System (DSS) for a large retail company, it could equally well be used to support predictive stock control at SMEs, or any other domain where time series with frequent zeros and overdispersion arise.

Chapter 4

countTS. A Python library to support time series forecasting

4.1 Introduction

In the previous chapters we have introduced several models to forecast count time series and the corresponding algorithms to obtain forecasts. These algorithms are complex and are not straightforward to implement in any common programming language using existing libraries. Therefore, because of its relevance for the industrial sponsor of this thesis, we have developed a library implementing them, which is proprietary.

The package adopts the Object-Oriented Programming (OOP) paradigm for implementing DLMs and DGLMs on the one hand; and the Functional Programming (FP) paradigm for the novel models proposed in this thesis and the auxiliary functions on the other. The general structure of the package is shown in Figure 4.1 . The syntax for defining the models is similar to that in the popular `dlm` package in `R` (Petris et al., 2009); for the practitioner familiar with it, this allows a faster transition to our `Python` package (while providing additional and enhanced models and functionalities).

The main advantage of this package over `dlm` is that it implements DGLMs (West et al., 1985), i.e. extends DLMs to observations from any distribution of the exponential family, and, via *wrapper functions*, the novel models proposed in this thesis. Additionally, it is implemented in `Python`, which is a general-purpose programming language, and currently more popular than `R` (Carbonnelle, 2022). Other packages implementing this type of models in `Python` are: `pyDLM`, which only implements DLMs, i.e., only considers normally distributed observations; and `pyBATS`, with respect to which, our approach offers additional features, like more exponential family distributions for the observations and useful additional tools for modeling and verifying results.

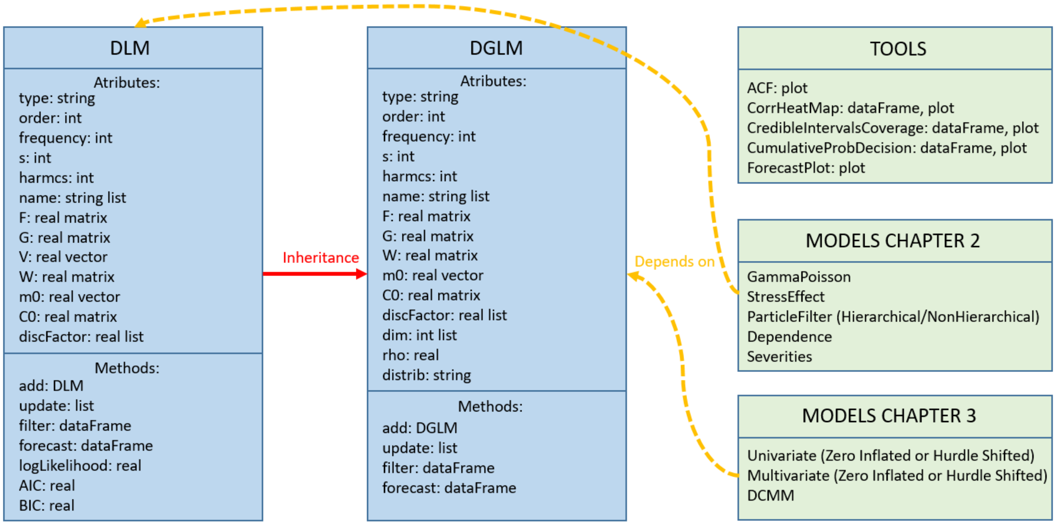


Figure 4.1: UML diagram of the two classes in the package (DLM and DGLM) in blue, and groups of additional functions in green.

4.2 Model definition

In this section we detail the definition of DLMs and DGLMs, as well as the different common types of these (linear trend, seasonal, etc) which can be combined to form new models.

4.2.1 Definition of DLMs

The basic models implemented in the package are DLMs, extensively treated in West and Harrison (1997), a class of state space models that lend themselves quite naturally to be treated from a Bayesian approach. Furthermore, they allow for a natural interpretation of a time series as the combination of several components, such as trend, seasonal or regression components (Prado & West, 2010). The standard DLM is defined by equations (B.1) in Appendix B. And, therefore, observations y_t and states $\boldsymbol{\theta}_t$ are normal with

$$\begin{aligned} y_t \,|\, \boldsymbol{\theta}_t &\sim N(\boldsymbol{F}_t\boldsymbol{\theta}_t, V_t), \\ \boldsymbol{\theta}_t \,|\, \boldsymbol{\theta}_{t-1} &\sim N(\boldsymbol{G}_t\boldsymbol{\theta}_{t-1}, \boldsymbol{W}_t). \end{aligned}$$

Any DLM is thus completely defined by the quadruplet $\{\boldsymbol{F}_t, \boldsymbol{G}_t, V_t, \boldsymbol{W}_t\}$. Matrices $\boldsymbol{F}_t$ and $\boldsymbol{G}_t$ contain (as diagonal blocks) the different factors deemed important to predict series behavior (trend, seasonality,...). A simple example on how to *buid* a simple DLM object called `modDLM` with a single order 2 polynomial component, corresponding to a linear growth trend, is given by Listing 4.1, and characterized by the quadruplet

$$\boldsymbol{F}_t = \begin{pmatrix} 1 & 0 \end{pmatrix}, \quad \boldsymbol{G}_t = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}, \quad V_t = 1, \quad \boldsymbol{W}_t = \boldsymbol{I}_2.$$

```python
import dlm # File with class DLM
import numpy as np
modDLM=dlm.DLM(type='Poly',order=2,V=1,W=np.identity(2))
```

Listing 4.1: Specification of a simple object of a DLM.

It is also possible to specify directly all the defining matrices of the model to create a *custom* model, as `modDLMm` in Listing 4.2. Note that unlike the model in Listing 4.1, `modDLMm` is multivariate, specifically a bivariate model. Multivariate DLMs can be defined in this fashion, manually specifying matrices of adequate dimensions, $\boldsymbol{F}_t$ would now be of dimension $n \times m$, where n is the number of states of the model and m the number of time series.

```
F_t = np.array([[1, 0],[0, 1], [0, 0], [0, 0]])
G_t = np.array([[1, 0, 1, 0], [0, 1, 0, 1],
                [0, 0, 1, 0], [0, 0, 0, 1]])
V_t = np.identity(2); W_t = np.identity(4)
modDLMm=dlm.DLM(F=F_t,G=G_t,V=V_t,W=W_t)
```

Listing 4.2: Creation of a DLM specifying all matrices.

4.2.2 Definition of DGLMs

DGLMs (West et al., 1985) extend the observational distributions of DLMs to any probability density function (or p.m.f. in the discrete case) within the exponential family, $p(y_t|\eta_t)$, and are defined by equations (3.2), which are equivalent to

$$\begin{aligned} g(\eta_t) &= \lambda_t = \boldsymbol{F}_t\boldsymbol{\theta}_t, \\ \boldsymbol{\theta}_t \,|\, \boldsymbol{\theta}_{t-1} &\sim (\boldsymbol{G}_t\boldsymbol{\theta}_{t-1}, \boldsymbol{W}_t). \end{aligned}$$

Note that, unlike a DLM, no particular distribution is assumed for the states $\boldsymbol{\theta}_t$ or the linear predictor λ_t relating to the observations y_t, only their means and variances. Also, there is no evolution variance V_t and thus a DGLM is defined by the triplet $\{\boldsymbol{F}_t, \boldsymbol{G}_t, \boldsymbol{W}_t\}$.

In the current version of the code, the supported observational distributions from the exponential family are: Poisson, Bernoulli, Multivariate Bernoulli, Negative Binomial and Multivariate Negative Binomial. The last two require fixed dispersion parameters so as to belong to the exponential family of distributions. Listing 4.3 shows a Poisson DGLM, `modDGLM`, with the same polynomial component as Listing 4.1.

```
import dglm # File with class DGLM
modDGLM=dglm.DGLM(type='Poly',order=2,distrib='Poi',
                  W=np.identity(2))
```

Listing 4.3: Specification of a simple instance of a DGLM.

As with standard DLMs, the triplet $\{\boldsymbol{F}_t, \boldsymbol{G}_t, \boldsymbol{W}_t\}$ can be manually defined when creating a new object (model) of class DGLM; and matrices $\boldsymbol{F}_t$ and $\boldsymbol{G}_t$ can contain different factors or components.

4.2.3 Components

As mentioned, both DLMs and DGLMs can incorporate different components as building blocks (which by themselves also constitute standalone models):

- **Polynomial trend.** Polynomial components are used to model the trends in the time series. The most commonly used are: the first order polynomial model, or local level, characterized by $\boldsymbol{F}_t = \boldsymbol{G}_t = 1$; and, the second order polynomial model, or linear growth. The n^{th}-order polynomial DLM is defined through

  $$\boldsymbol{F}_t = \begin{pmatrix} 1 & 0 & \dots & 0 \end{pmatrix}, \quad \boldsymbol{G}_t = \begin{pmatrix} 1 & 1 & & \mathbf{0} \\ & 1 & \ddots & \\ & & \ddots & 1 \\ \mathbf{0} & & & 1 \end{pmatrix},$$

  with $\boldsymbol{F}_t$, $\boldsymbol{G}_t$ of dimensions $n \times 1$ and $n \times n$ respectively. The syntax for creating a DLM (DGLM is analogous) object with this structure is `dlm.DLM(type= 'Poly', order=n)`.

- **Seasonal factors.** When considering the presence of a seasonal effect of period s, one of the approaches available to model seasonality is through a seasonal factor component,

  $$\boldsymbol{F}_t = \begin{pmatrix} 1 & 0 & \dots & 0 \end{pmatrix}, \quad \boldsymbol{G}_t = \begin{pmatrix} -1 & -1 & \dots & -1 \\ & 1 & & \mathbf{0} \\ & & \ddots & \\ \mathbf{0} & & & 1 \end{pmatrix},$$

  with $\boldsymbol{F}_t$, $\boldsymbol{G}_t$ of dimensions $(s-1) \times 1$ and $(s-1) \times (s-1)$ respectively. The proposed syntax is `dlm.DLM(type='Seas', frequency=s)`.

- **Fourier Form Seasonal.** The other approach available to model a period s seasonality is through a Fourier component, which can include all harmonics $\boldsymbol{H}_j$ ($j = 1, .., \lfloor s/2 \rfloor$), or only some of them,

$$\boldsymbol{F}_t' = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 1 \\ 0 \end{pmatrix}, \quad \boldsymbol{G}_t = \text{diag}\big(\boldsymbol{H}_1 \ \dots \ \boldsymbol{H}_{\lfloor s/2 \rfloor}\big), \quad \boldsymbol{H}_j = \begin{pmatrix} cos(2\pi j/s) & sin(2\pi j/s) \\ -sin(2\pi j/s) & cos(2\pi j/s) \end{pmatrix},$$

  with $\boldsymbol{F}_t$, $\boldsymbol{G}_t$ of dimensions $(s-1)\times 1$ and $(s-1)\times(s-1)$ respectively. The proposed syntax for a Fourier component with period s and q harmonics is `dlm.DLM(type='Trig', s=p, harmcs=q)`. The use of only a few of the lower-frequency harmonics reduces the size of the involved matrices and usually gives a more parsimonious representation of the seasonality, which is less sensible to noise (useful for example with yearly seasonality in daily data, s=365).

- **Regression.** Covariates $\{x_t\}_{t=1}^T$ are also straightforward to include through a model with $\boldsymbol{F}_t = x_t$ and $\boldsymbol{G}_t = 1$. This is done with `dlm.DLM(type='Reg', name='x')`, where $'x'$ is the name given to the particular covariate.

Thanks to the superposition principle (Prado & West, 2010) components can be combined at will to form new models that more adequately reflect the underlying process driving the time series under study. Given $\boldsymbol{F}_1$, $\boldsymbol{G}_1$ from one component/model; and $\boldsymbol{F}_2$, $\boldsymbol{G}_2$ from another, their superposition results in a new model with $\boldsymbol{F} = \big(\boldsymbol{F}_1 \ \ \boldsymbol{F}_2\big)$ and $\boldsymbol{G} = \text{diag}\big(\boldsymbol{G}_1 \ \ \boldsymbol{G}_2\big)$. This is done with the operator `+`, as shown in Listing 4.4.

```
modDLM=dlm.DLM(type='Poly',order=2) + dlm.DLM(type='Seas',
    frequency=7)
```

Listing 4.4: Specification of a DLM with linear growth and period 7 seasonality.

4.3 Prior information

To complete specification, note that it is still necessary to specify the initial prior moments for the state vector, $\boldsymbol{m}_0$ and $\boldsymbol{C}_0$. If these are not specified through the arguments `m0, C0`, the default choice is a vector of zeros for $\boldsymbol{m}_0$, and the identity matrix for $\boldsymbol{C}_0$ (of conformable dimensions both). However, if previous observations of the time series to model are available, the moments can be initialized using MLE. Another alternative, specially when the number of observations is low, is to directly solve the system of equations without evolution errors given by (B.1) or (3.2), as illustrated for $\boldsymbol{m}_0$ in the *wind shear* example in Section 2.6.1.

When defining a Negative Binomial DGLM (univariate o multivariate), the fixed dispersion parameter r_t, can be specified manually using the argument `rt` in the call to the builder `dglm.DGLM()`. If the parameter is not specified, it is estimated using an EM algorithm (Adamidis, 1999) when using the *wrapper function* `NegBinDGLM`, which applies a NB DGLM to a series and returns a `DataFrame` with the forecast distributions at the desired credible intervals. The number of observations used by `NegBinDGLM` to estimate r_t is controlled via the argument `obsForR` (Listing 4.5); by default, it uses the first 21 observations.

```
DF_NB = NegBinDGLM(series=x, model=modDGLM_NB,
                   credInterv=CredIntervals, obsForR=21)
```

Listing 4.5: Forecast of series `x` with model `modNB` and r_t estimated using EM with the first 21 observations.

4.4 Discount factors

The specification of the unknown state evolution variance matrix $\boldsymbol{W}_t$ is crucially important for obtaining accurate forecasts. However, its elicitation can be difficult; a common alternative, implemented in this package, is the use of *discount factors* (West & Harrison, 1997) which are easier to elicit. The discount factor δ take values in $(0,1]$, with 1 being the case of a stable state vector with no stochastic changes ($\boldsymbol{W}_t = \mathbf{0}$). In practice, discount factors are usually assigned a value between 0.8 and 0.99. This is done with the the argument `discFactor` of the corresponding builder, as Listing 4.6 shows for a DLM. In this case, the resulting model `modDLM` in Listing 4.6 would therefore have different discount factors for each component. This is quite useful as trend and seasonal components often require different discounts: usually, the seasonal characterization is more stable in time and, hence, more accurately represented with higher values.

```
modDLM=dlm.DLM(type='Poly',order=2,discFactor=[0.8]) +
        dlm.DLM(type='Seas',frequency=7,discFactor=[0.9])
```

Listing 4.6: DLM with two components with different discount factors.

4.5 The DLM class

As seen in Sections 4.2 through 4.4, the instances of the `DLM` class are essential for using Dynamic Linear Models to forecast time series. This section gives a brief overview of the class, its attributes and methods (Figure 4.1).

4.5.1 Attributes

An instance/object of class DLM has thirteen attributes. Nine of them are used in every DLM:

- `type`: list of strings that indicates the components of the model, e.g. `['Poly','Seas']`.
- `name`: list of strings that can assign different names to each component of the model, e.g. `['Price','Promotion']`.
- `F`: Observation evolution matrix ($\boldsymbol{F}_t$ in equations (B.1)).

- `G`: State evolution matrix ($\boldsymbol{G}_t$ in equations (B.1)).
- `V`: Evolution variance for the observation equation (V_t in equations (B.1)).
- `W`: Evolution variance matrix for the state equation ($\boldsymbol{W}_t$ in equations (B.1)).
- `m0`: Prior mean for the state ($\boldsymbol{m}_0$ in equations (B.1)).
- `C0`: Prior variance for the state ($\boldsymbol{C}_0$ in equations (B.1)).
- `discFactor`: list of discount factors for each component of the model, e.g. `[0.8,0.9]`.

Four additional attributes are specific to the different `types`:

- `order`: order of the polynomial (`'Poly'`) component.
- `frequency`: period of the seasonal factor (`'Seas'`) component.
- `s`: period of the Fourier form seasonal (`'Trig'`) component.
- `harmcs`: harmonics included in the Fourier form seasonal (`'Trig'`) component (e.g. `harmcs=2` indicates that only the two lower harmonics are to be included).

4.5.2 Update and forecast methods

The learning and one-step ahead forecast procedure is completely defined by the equations in Appendix B. The package implements this in Python, leaning on *numpy* for matrix operations. To avoid possible numerical instabilities, equivalent equations using singular value decomposition (SVD) (Wang et al., 1992), replacing those in Appendix B, are used by default. If we denote the SVD of a matrix $\boldsymbol{A}$ as $\boldsymbol{U}_A\boldsymbol{D}_A\boldsymbol{V}'_A$, the predictive variance $\boldsymbol{R}_t$ of the states, is calculated as

$$\boldsymbol{M} = \begin{pmatrix} \sqrt{\boldsymbol{D}_{C_{t-1}}}\,\boldsymbol{U}'_{C_{t-1}}\,\boldsymbol{G}'_t \\ \sqrt{\boldsymbol{D}_{W_t}}\,\boldsymbol{U}'_{W_t} \end{pmatrix},$$

$$\boldsymbol{R}_t = \boldsymbol{V}_M\,\boldsymbol{D}_M^2\,\boldsymbol{V}'_M,$$

and the posterior variance, $\boldsymbol{C}_t$, with

$$\boldsymbol{N} = \begin{pmatrix} \sqrt{\boldsymbol{V}_t^{-1}}\,\boldsymbol{F}_t\,\boldsymbol{U}_{R_t} \\ (\sqrt{\boldsymbol{D}_{R_t}})^{-1} \end{pmatrix},$$

$$\boldsymbol{C}_t = \boldsymbol{U}_{R_t}\,(\boldsymbol{D}_N^{-1})^2\,\boldsymbol{U}'_{R_t}.$$

The method `update` implements this and can be used to evolve the model, one observation at a time (Listing 4.7).

```
def update(self, obs, discFactor=[], SVD=True, warnings=False):
```

Listing 4.7: Arguments of `update` method. In case of not wanting to use SVD, just set `SVD=False`.

Whenever there is a missing observation `obs=np.nan` the posterior moments of the state distribution are not updated ($\boldsymbol{m}_t = \boldsymbol{a}_t$ and $\boldsymbol{C}_t = \boldsymbol{R}_t$) since there is no new observation. This method facilitates the incorporation of the update procedure into other functions. However, most practitioners will find much more useful the `filter` method that takes as argument the whole series and calls to `update` (Listing 4.8).

```
def filter(self, series, discFactor=[], dfReg=None, SVD=True,
    warnings=False):
```

Listing 4.8: Arguments of `filter` method.

The `dfReg` attribute of `filter` is optional and takes the regressor variables of the model. The format is a `DataFrame` with each covariate in a column with the same name used when defining the regression component (Section 4.2.3).

This corresponds to a time-varying $\boldsymbol{F}_t$. Until now we have only considered constant entries for $\boldsymbol{G}_t$, $\boldsymbol{V}_t$ and $\boldsymbol{W}_t$; in case these are also time-varying, the update must be done with the `update` method after modifying the corresponding time-varying entry directly in the model attribute. Finally, the method `forecast`, returns a summary `DataFrame` of `nAhead` observations into the future (Listing 4.9).

```
def forecast(self, nAhead=None, discFactor=[], dfReg=None):
```

Listing 4.9: Arguments of `forecast` method.

4.5.3 Other methods

The rest of the methods of class `DLM` are related to the loglikelihood. They are used for estimating model parameters via MLE, and computing the AIC and BIC of different models for comparison purposes. Method `LogLikelihood`, computes the logarithm of the likelihood of observations $y_1, ..., y_n$ (which are passed to the method with argument `obs`), which is,

$$-\frac{1}{2}\log(2\pi) - \frac{1}{2}\sum_{t=1}^{n}\log(Q_t) - \frac{1}{2}\sum_{t=1}^{n}(y_t - f_t)'Q_t^{-1}(y_t - f_t).$$

The AIC and BIC methods use this to compute and return the corresponding information criterion.

4.6 The DGLM class

The `DGLM` class implements the generalized version of DLMs. It currently supports five distributions for the observations ($p(y_t|\eta_t)$ in (3.2)): Bernoulli, Multivariate Bernoulli, Poisson, Negative Binomial, and Multivariate Negative Binomial. Its structure is similar to that of the `DLM` class (Figure 4.1).

4.6.1 Attributes

An object of class `DGLM` has fifteen attributes (Figure 4.1): the ones from a `DLM` object (Section 4.5.1), except for the evolution variance of the observation equation `V` which is not present in a DGLM (3.2), and two new ones:

- `rho`: real number (`float`) indicating the discount factor for the *random effects*, as explained in Section 3.3. It is only used by the `update` method in conjunction with Poisson distributed observations to introduce further variability in the predictions. This results in wider predictive distributions which can more adequately model overdispersed time series. It takes values in $(0, 1]$; by default, it is assigned a value of 1 when initializing a DGLM object, i.e. no discount factor or random effects are applied to the linear predictor λ_t in (3.2); lower values induce higher variability.

- `distrib`: string that indicates the distribution from the exponential family assumed for the observations in the DGLM. The acceptable values are: `'Ber'` (Bernoulli), `'MBer'` (Multivariate Bernoulli), `'Poi'` (Poisson), `'NB'` (Negative Binomial), and `'MNB'` (Multivariate Negative Binomial).

4.6.2 Update and forecast

The learning and one-step ahead forecast procedure for a DGLM, see West et al. (1985), is similar to that defined for a standard DLM by the equations in Appendix B, and using the moment matching technique. For each $t > 0$, it results in:

- One step ahead prior moments for the states given $\mathcal{D}_{t-1}$ ($\boldsymbol{y}_{1:t-1}$ and other relevant information), $\boldsymbol{\theta}_t|\mathcal{D}_{t-1} \sim (\boldsymbol{a}_t, \boldsymbol{R}_t)$, with

$$\boldsymbol{a}_t = \boldsymbol{G}_t \boldsymbol{m}_{t-1}, \qquad \boldsymbol{R}_t = \boldsymbol{G}_t \boldsymbol{C}_{t-1} \boldsymbol{G}_t{}' + \boldsymbol{W}_t.$$

- One step ahead forecasts are based on the conjugacy-induced predictive distribution with pdf

  $$p(\boldsymbol{y}_t|\boldsymbol{\alpha}_t, \beta_t) = b(\boldsymbol{y}_t, \phi_t)c(\boldsymbol{\alpha}_t, \beta_t)/c(\boldsymbol{\alpha}_t + \phi_t T(\boldsymbol{y}_t), \beta_t + \phi_t),$$

  where the precision parameter ϕ_t is the inverse of the variance V_t in (3.1), and the hyper-parameters $\{\boldsymbol{\alpha}_t,\ \beta_t\}$ are estimated using moment matching,

  $$E[g(\boldsymbol{\eta}_t) = \lambda_t|D_{t-1}] = f_t,$$
  $$V[g(\boldsymbol{\eta}_t) = \lambda_t|D_{t-1}] = \boldsymbol{Q}_t,$$

  where $f_t = \boldsymbol{F}_t\boldsymbol{a}_t$ and $\boldsymbol{Q}_t = \boldsymbol{F}_t\boldsymbol{R}_t\boldsymbol{F}'_t$.

- Posterior moments for the states after observing $\boldsymbol{y}_t$, $\boldsymbol{\theta}_t|\mathcal{D}_t \sim (\boldsymbol{m}_t, \boldsymbol{C}_t)$,

  $$\boldsymbol{m}_t = \boldsymbol{a}_t + \boldsymbol{R}_t\boldsymbol{F}'_t\boldsymbol{Q}_t^{-1}(f_t^* - f_t), \qquad \boldsymbol{C}_t = \boldsymbol{R}_t - \boldsymbol{R}_t\boldsymbol{F}'_t\boldsymbol{Q}_t^{-1}(\boldsymbol{I}_m - \boldsymbol{Q}_t^*\boldsymbol{Q}_t^{-1})\boldsymbol{F}_t\boldsymbol{R}_t,$$

  with $f_t^* = E[g(\eta_t)|D_t]$ and $\boldsymbol{Q}_t^* = V[g(\boldsymbol{\eta}_t)|D_t]$, given that the posterior of the natural parameter is

  $$p(\boldsymbol{\eta}_t|D_t) = c\big(\boldsymbol{\alpha}_t + \phi_t T(\boldsymbol{y}_t), \beta_t + \phi_t\big) \exp\Big(\big(\boldsymbol{\alpha}_t + \phi_t T(\boldsymbol{y}_t)\big)\boldsymbol{\eta}_t - \big(\beta_t + \phi_t\big)a(\boldsymbol{\eta}_t)\Big).$$

As with DLMs, when an observation is missing, moments $\boldsymbol{m}_t$ and $\boldsymbol{C}_t$ are not updated with new information and we have $\boldsymbol{m}_t = \boldsymbol{a}_t$ and $\boldsymbol{C}_t = \boldsymbol{R}_t$. The particular conjugate analysis needed for using the multivariate Bernoulli or multivariate Negative Binomial distributions was detailed in Section 3.4. This update procedure is incorporated in the method `update` of the class, which automatically uses the appropriate conjugate distribution for the distribution assumed for the observations (`distrib`). As shown in Listing 4.10, the method returns a list with seven elements: the predictive moments $\boldsymbol{a}_t$, $\boldsymbol{R}_t$ for the state vector; the predictive mean, variance and median for the observations; and the posterior values for the hyperparameters $\boldsymbol{\alpha}_t$, β_t.

```
series = data['x']
predMeanStates = list()
predVarStates = list()
for t in range(len(series)):
    resultFilt = modDGLM.update(series[t])
    predMeanStates=predMeanStates+[resultFilt[0]]
    predVarStates=predVarStates+[resultFilt[1]]
    data.loc[t,'mean_t']=resultFilt[2]
    data.loc[t,'var_t']=resultFilt[3]
    data.loc[t,'median_t']=resultFilt[4]
    data.loc[t,'alpha_t']=resultFilt[5]
    data.loc[t,'beta_t']=resultFilt[6]
    # 90% credible intervals:
    data.loc[t,'5%']=nbinom.ppf(0.05,
                        n=data.loc[t,'alpha_t'],
                        p=data.loc[t,'beta_t']/(1+data.loc[t,
                                                  'beta_t']))
    data.loc[t,'95%']=nbinom.ppf(0.95,
                        n=data.loc[t,'alpha_t'],
                        p=data.loc[t,'beta_t']/(1+data.loc[t,
                                                  'beta_t']))
```

Listing 4.10: Example of forecasting with `update` and Poisson DGLM.

There is also the useful `filter` method. After receiving as arguments the whole time series (`obs`), and optionally the possible regression variables (`dfReg`) and credible intervals to compute (`CredIntervals`), it returns the summary `DataFrame` with the mean, variance, median and, optionally, the credible intervals of the predictive distribution for each relevant time instant.

The `forecast` method has analogous behavior to that of the DLM class (Section 4.5.2), but with the new argument `obsForPreFit` that indicates the number of observations to be used for calculating the dispersion parameter for the Negative Binomial or Multivariate Negative Binomial cases.

```
modDGLM.filter(obs=series,dfReg=data['price','promo'],
               CredIntervals=[0.2,0.5,0.8,0.9],
               obsForPreFit=21)
modDGLM.forecast(nAhead=14,
                 dfNReg=dataNew['price','promotion'])
```

Listing 4.11: Example of use of the `filter` and `forecast` methods.

4.7 Complex Models

Besides the standard DLM, and the five variants of DGLMs considered, it is also possible to use these to create new models that rely on them.

4.7.1 Univariate mixtures

One model that can be implemented leaning on the classes and methods of our package is the univariate model proposed in Section 3.3 to deal with demand time series with frequent zeros and varying levels of dispersion. This model uses Poisson and Negative Binomial DGLMs. As it is one of the central models in this thesis, it has its own auxiliary functions predefined in the file `alg.py`. As commented in Section 3.3, and discussed in the example in 3.6, we can consider two version of the univariate model, one that follows a hurdle shifted scheme like DCMMs (3.3) and another that follows a zero inflated one (3.5). Listing 4.12 shows how to do a filtering of both versions.

```
CredibleIntervals = [0.2,0.35,0.5,0.65,0.7,0.8,0.9,0.95]
y = data['sales']
x_ZI = y                                                 # ZI
x_HS = np.array([np.nan if yy==0 else yy-1 for yy in y])  # HS

modDGLM_Ber = dglm.DGLM(type='Poly',order=1,distrib='Ber') +
    dglm.DGLM(type='Trig',s=7,distrib='Ber')
modDGLM_NegBin = dglm.DGLM(type='Poly',order=1,distrib='NB') +
    dglm.DGLM(type='Seas',frequency=7,distrib='NB')
```

```
DF_NegBin_ZI = alg.NegBinDGLM(x_ZI,modDGLM_NegBin,
    CredibleIntervals,daysforR=21)
DF_NegBin_HS = alg.NegBinDGLM(x_HS,modDGLM_NegBin,
    CredibleIntervals,daysforR=21)
DF_Ber = alg.BernoulliDGLM(z,modDGLM_Ber)
```

Listing 4.12: Implementation of univariate models (3.3) and (3.5) in Section 3.3.

Then, it is possible to use the function `UnivariateModel` implementing the algorithms to obtain point forecasts and credible intervals of the predictive distribution (3.7), as in Listing 4.13.

```
DF_UniM_ZI = alg.UnivariateModel(DF_Ber,DF_NegBin_ZI)
DF_UniM_HS = alg.UnivariateModel(DF_Ber,DF_NegBin_HS)
```

Listing 4.13: Obtaining predictive distribution of univariate models (3.3) and (3.5).

By default, `alg.NegBinDGLM` and `alg.BernoulliDGLM` forecasts one step ahead, but it is also possible to obtain forecasts k-steps ahead into the future by changing the default argument `kAhead` from 1 to the desired value. This is done using Monte Carlo samples, drawing from the corresponding predictive distribution and, then, using the drawn value as the observed, update, and repeating the process until reaching the desired prediction horizon k. As Algorithm 3.1 reflects, this is done for N different chains or paths.

4.7.2 Multivariate mixture

The multivariate version of the previous model, which consists of the two DGLMs in 3.9, also has the corresponding functions for its implementation in the file `alg.py`, as shown in Listing 4.14.

```python
y1 = data1['sales']; y2 = data2['sales']
x1_ZI = y1; x2_ZI = y2                                    # ZERO INFLATED
x1_HS = np.array([np.nan if y==0 else y-1 for y in y1])  # HURDLE SHIFTED
x2_HS = np.array([np.nan if y==0 else y-1 for y in y2])  # HURDLE SHIFTED

modDGLM_MBer = dglm.DGLM(type='Poly',order=1,distrib='MBer') + dglm.DGLM(type='Trig',s=7,distrib='MBer')
modDGLM_MNB = dglm.DGLM(type='Poly',order=1,distrib='MNB') + dglm.DGLM(type='Seas',frequency=7,distrib='MNB')

DF_MBer = alg.MultiBerDGLM([z1,z2],modMBerDGLM)
DF_MNegBin_ZI = alg.MultiNegBinDGLM([y1,y2],[x1_ZI,x2_ZI],modDGLM_MNB,CredibleIntervals,daysforR=21)
DF_MNegBin_HS = alg.MultiNegBinDGLM([y1,y2],[x1_HS,x2_HS],modDGLM_MNB,CredibleIntervals,daysforR=21)

DF_MultiM_ZI = alg.UnivariateModel(DF_MBer,DF_MNegBin_ZI)
DF_MultiM_HS = alg.UnivariateModel(DF_MBer,DF_MNegBin_HS)
```

Listing 4.14: Multivariate mixture of two DGLMs.

4.7.3 DCMM

Finally, another family of models implemented in the package are the Dynamic Count Mixture Models (DCMM) of Berry and West (2020), a mixture of two DGLMs: a Bernoulli for *zero/non-zero sales*, and a Poisson for the *number of sales*. If we define $z_t = \mathbb{1}_{(y_t>0)}$, the DCMM is defined through $y_t = z_t(x_t + 1)$ with $z_t \sim Ber(\pi_t)$, $x_t \sim Po(r_t, p_t)$, with link functions relating to the linear predictors for the Bernoulli and Poisson components being *logit* and *log* respectively.

This type of model could be implemented using a pair of instances of the DGLM class, with the caveat that since we are modeling the shifted series

$x_t = y_t - 1$ with a Poisson DGLM, for zero-valued observations of the original series (y_t=0) we would need to pass a negative value (x_t= $-$ 1) to the `update` method; this can be dealt with treating those observation as missing (`obs=None`). However, as aforementioned, the DCMM family of models is already implemented and the practitioner can obtain forecasts using the function `DCMM` (Listing 4.15).

```
CredibleIntervals = [0.2,0.35,0.5,0.65,0.7,0.8,0.9,0.95]
y = data['sales']
x = np.array([np.nan if yy==0 else yy-1 for yy in y])
modDGLM_Ber=dglm.DGLM(type='Poly',order=1,distrib='Ber') + dglm
    .DGLM(type='Trig',s=7,distrib='Ber')
modDGLM_Pois=dglm.DGLM(type='Poly',order=1,distrib='Poi') +
    dglm.DGLM(type='Seas',frequency=7,distrib='Poi')
DF_Ber = alg.BernoulliDGLM(z,modDGLM_Ber)
DF_Pois = alg.PoissonDGLM(x_HS,modDGLM_Pois,CredibleIntervals)
Res = alg.DCMM(DF_Ber,DF_Pois)
```

Listing 4.15: Implementation of a DCMM.

4.7.4 Models for general count time series

The library `countTS` also implements the models in Chapter 2. They are preprogramed through several functions in `alg.py`:

- `GammaPois`. The basic Gamma-Poisson model introduced in Section 2.3.1, has as syntax `alg.GammaPois(x,n,a_0,p_0)`, where `x` and `n` refer to the series x_k, n_k; and `(a_0,p_0)` are the initial prior values for parameters a and p of the Gamma distribution.

- `StressEffect`. Contains the code to implement Algorithm 2.1 for the *stress effect* model (2.1) in Section 2.3.2. Its syntax is `alg.StressEffect(x,n,proposalD,l_j,a_j,b_j,s2_j,mu_a,sigma2_a,mu_b,sigma2_b,alpha,beta,numSamples,burnP)`, where: `x` and `n` are the series x_k, n_k;

`proposalD` is a string indicating the proposal distribution to use, `'Gamma'` for the recommended one, and `'Normal'` for the usual symmetric normal distribution used in MCMC algorithms; `a_j`, `b_j`, `l_j`, `s2_j` are the initial values $a_0, b_0, \lambda_0, \sigma_0^2$; `mu_a`, `mu_b` are the prior values for the means of a and b respectively, and `sigma2_a`, `sigma2_b` for the variances; `alpha`, `beta` are the prior values for the distribution of σ^2; lastly, `numSamples` is the number of samples that the algorithm will return, and `burnP`, the burn-in period.

- `ParticleFilter`. Contains the code to implement the particle filter in Algorithms 2.2 of Model (2.2) in Section 2.3.2, and its hierarchical version, Algorithm 2.6 of Model (2.5) in Section 2.6.1. The first Algorithm is used when the function receives as arguments a single pair of series (x_k, n_k) corresponding to a single group; and the hierarchical one is used when it receives more than one (corresponding to different groups, i.e., $L > 1$). The syntax is `alg.ParticleFilter(x,n,modDLM,N,Ness,verbose)`, where: `x` is a list with series x_k^i (as many as different groups) and `n` is a list with series n_k^i; `modDLM` is an object of class `DLM` defining the evolution matrices of (2.2) or (2.5); `N` is the number of chains for the MCMC; `Ness` a proportion of `N` indicating the minimum effective sample size to accept; finally, `verbose` is a Boolean that indicates whether the function displays on the screen information (`True`) or not (`Not`) about each step during the execution.

- `Dependence`. Contains the code to implement Algorithm 2.4 for the *dependence* model in Section 2.3.2. Its syntax is `alg.Dependence(x1,x2,n-,proposalD,l1_j,l2_j,a_j,b_j,s2_j,r,p,mu_a,sigma2_a,mu_b,sigma2_b,alpha,beta,numSamples,burnP)`, where: `r` and `p` are the prior values for the Gamma distribution of λ_1; and the rest of arguments are analogous to the ones of function `StressEffect`.

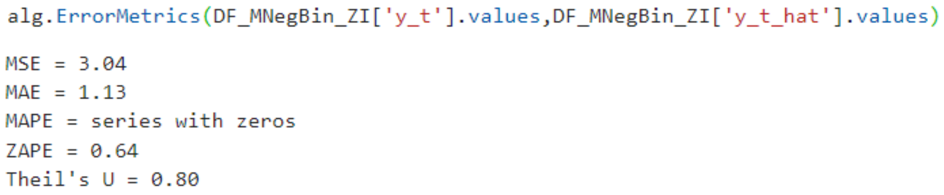
- `Severities`. Contains the code to implement the *severity* model in

Section 2.5, and can be applied in conjunction with any of the previous models. Its syntax is `alg.Severities(samples,dirParam)`, where: `samples` are the samples returned from any of the previous models; and `dirParam` a list with the initial prior parameters for the Dirichlet distribution.

4.8 Auxiliary functions

Other important functions and algorithms in the package, essential for example to effectively use it in the inventory management domain, are detailed in this subsection. All of them are included in the file `alg.py`. In the Listings in the section we assume, such file has already been imported with the command `import alg`.

Error metrics for Point Forecast. The function `ErrorMetrics` returns five error metrics quite useful to asses the performance of count time series: Mean Squared Error (MSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), Zero Adjusted Percentage Error (ZAPE) and Theil's U. Figure 4.2 shows its syntax and exemplifies its use.



```
alg.ErrorMetrics(DF_MNegBin_ZI['y_t'].values,DF_MNegBin_ZI['y_t_hat'].values)
```

```
MSE = 3.04
MAE = 1.13
MAPE = series with zeros
ZAPE = 0.64
Theil's U = 0.80
```

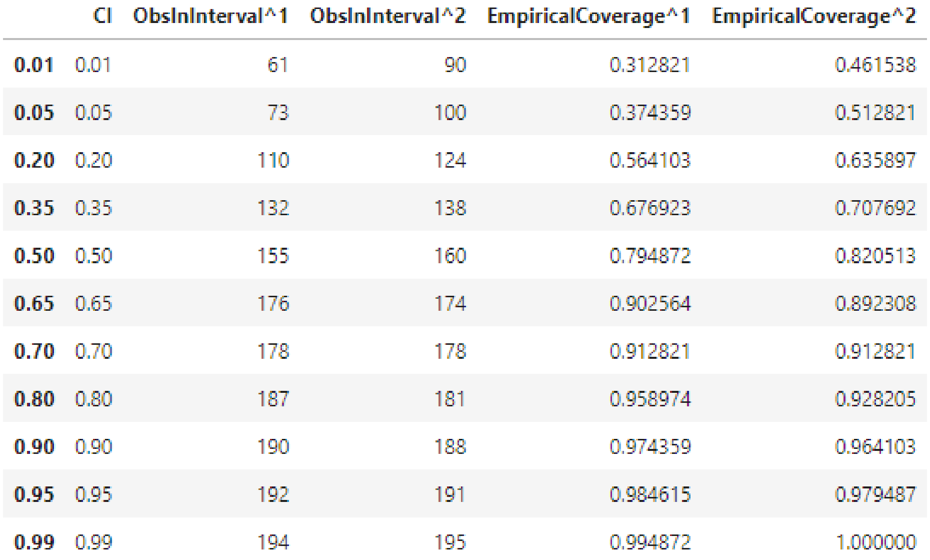
Figure 4.2: Metrics returned by function `ErrorMetrics`.

Error metrics for Predictive Distributions. In order to evaluate if the credible intervals at different percentages adequately cover the observations, we provide the function `CredibleIntervalsCoverage`. It can be called as in Listing 4.16.

```
CredibleIntervals
   =[0.01,0.05,0.2,0.35,0.5,0.65,0.7,0.8,0.9,0.95,0.99]
CIdataMNB=alg.CredibleIntervalsCoverage(DF=DF_MNegBin_ZI,
   CredibleIntervals=CredibleIntervals,title='Multi NegBin')
```

Listing 4.16: Calculation of the coverage of credible intervals.

The call returns the `DataFrame CIdataMNB` in Figure 4.3, which indicates the number of observations that fall in each of the credible intervals selected, and the empirical coverage ('observations that fall inside'/'number of observations').



	CI	ObsInInterval^1	ObsInInterval^2	EmpiricalCoverage^1	EmpiricalCoverage^2
0.01	0.01	61	90	0.312821	0.461538
0.05	0.05	73	100	0.374359	0.512821
0.20	0.20	110	124	0.564103	0.635897
0.35	0.35	132	138	0.676923	0.707692
0.50	0.50	155	160	0.794872	0.820513
0.65	0.65	176	174	0.902564	0.892308
0.70	0.70	178	178	0.912821	0.912821
0.80	0.80	187	181	0.958974	0.928205
0.90	0.90	190	188	0.974359	0.964103
0.95	0.95	192	191	0.984615	0.979487
0.99	0.99	194	195	0.994872	1.000000

Figure 4.3: Summary returned by `CredibleIntervalsCoverage`, for a multivariate model with two series, denoted by suffixes **'^1'** and **'^2'**.

Additionally the function plots the coverage in Figure 4.4.

Cumulative forecasts and decision making. As mentioned in Chapter 1, in many applications there is an interest in using the predictive distributions k-steps ahead to make informed decisions that avoid critical situations. In inventory management, for example, one of the main reasons why there is interest in obtaining demand forecasts is to avoid OoS situations. For this

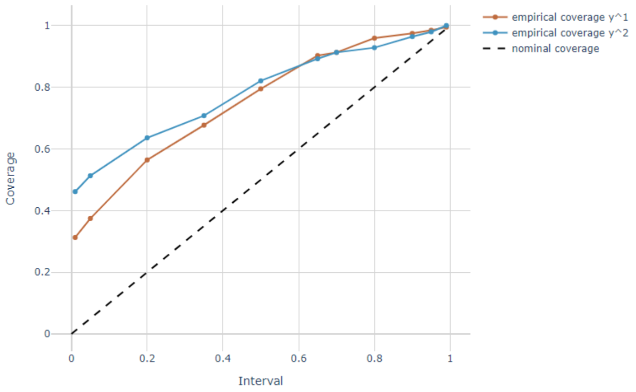


Figure 4.4: Coverage plot returned by `CredibleIntervalsCoverage`.

purpose, the package offers the function `CumulativeProbDecision`, which via the argument `criteria` (with possible values `alpha`, `cost` or `utility`) implements the three approaches for decision making mentioned in Section 3.5. It returns a summary `DataFrame` and a plot (Figure 3.12), which, if according to the selected criteria the user is prompted to take an action, also include a mark in the time period before which an action must be taken (and its critical or warning level classification). The resulting `DataFrame` can then be used, for example, to feed a Decision Support System software that automatically recommends a purchase order of a set of products.

Other ancillary functions. The package also offers several useful functions for undertaking the preliminary exploratory analysis of the time series to forecast, and determine which distributions, components, etc are adequate for modeling purposes:

- `plotACF` returns the Auto Correlation Function (ACF) plot with the number of lags specified by argument `lags`.
- `ForecastPlot` (given a data frame and passing as arguments the column name for the observations (`obs`), point forecasts (`pred`), and lower and upper bounds of credible intervals (`lb`,`ub`)) returns a plot (see Figure 3.10).

- `CorrHeatMap` computes the correlation coefficient between all the columns in the `DataFrame` that takes as argument and returns a heat-map plot with the name of the columns, as in Figure 3.3.

4.9 Example

In this section we show an example of how the library `countTS` can be used to model and forecast time series, specifically, we reproduce here the procedure to replicate the univariate example in Section 3.6. We want to fit the univariate model (3.3) in Section 3.3.1, to the sales time series of beer with SKU ’182’ in store with id ’173’ (Figure 4.5), and compare its performance with DCMM.

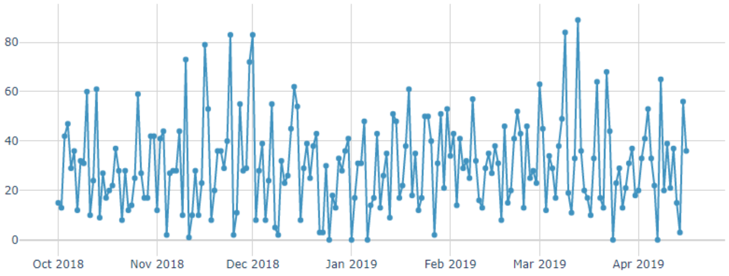


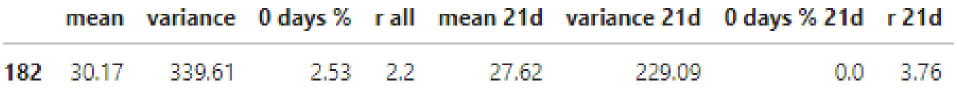
Figure 4.5: Daily sales of beer with SKU ’182’.

First, the code in Listing 4.17 returns a `DataFrame` with a summary of some statistics of the time series (Figure 4.6), and the ACF (Figure 3.1).

```
y=data[’173’][’182’]
pd.DataFrame(data={’mean’: [np.mean(y)],
  ’variance’: [np.var(y)],
  ’0 days %’: [np.sum(y==0)/len(y)*100],
  ’r all’:[fit_nbinom.fit(y)[’size’]],
  ’mean 21d’: [np.mean(y[:21])],
  ’variance 21d’: [np.var(y[:21])],
  ’0 days % 21d’: [np.sum(y[:21]==0)/len(y[:21])*100],
  ’r 21d’:[fit_nbinom.fit(y[:21])[’size’]]},
  index=[’182’])
alg.plotACF(data=data[’173’][’182’], var=’sales’, lags=15)
```

Listing 4.17: Exploratory analysis of beer ’182’

Note that all the information about the time series is in `data[’173’][’182’]`, which has a column named `sales`, with the daily sales of beer ’182’ in store ’173’. Figure 3.1, suggests the use of a model with weekly seasonality (period 7). Figure 4.6 suggests a dispersion parameter for the NB around 3.



	mean	variance	0 days %	r all	mean 21d	variance 21d	0 days % 21d	r 21d
182	30.17	339.61	2.53	2.2	27.62	229.09	0.0	3.76

Figure 4.6: Summary statistics of beer with SKU ’182’.

We then apply the univariate model specified in Section 3.6.1, that is, with a linear growth component for the Bernoulli part; and a linear growth component, plus a seasonal one of period 7, and two covariates referring to the price logarithm and a three level promotion variable for the NB part. Listing 4.18 shows the code for implementing this with `countTS`. Note that we do not specify r_t when defining `modNegBinDLM`, and let the algorithm automatically estimate it.

```
z = np.array([0 if yy==0 else 1 for yy in y])
CredibleIntervals = [0.05,0.2,0.35,0.5,0.65,0.7,0.8,0.9]

# Sale/NoSale part (Bernoulli):
modBer = dlm.DLM(type=’Poly’,order=2,distrib=’Ber’,
                 discFactor=[0.995])
DF_Ber = modBer.Filter(z,CredibleIntervals,dfReg=None)

# NumberSales part (Negative Binomial):
modNB = dglm.DGLM(type=’Poly’,order=2,distrib=’NB’) +
        dglm.DGLM(type=’Seas’,frequency=7,distrib=’NB’) +
        dglm.DGLM(type=’Reg’,name=[’Promo’],distrib=’NB’) +
        dglm.DGLM(type=’Reg’,name=[’LogPrice’],distrib=’NB’)
modNB.discFactor=[0.95] # same for the 4 components
DF_NB = modNB.Filter(y,CredibleIntervals,dfReg=None)

# Univariate model (3.3)
DF_UniMod = alg.UnivariateModel(DF_Ber,DF_NB)
```

Listing 4.18: Fitting of univariate model (3.3) to beer ’182’ data.

Likewise, we can apply a DCMM with the same components to the time series of daily sales of beer '182', as shown in Listing 4.19. We use a low *random effects* (0.3) discount value to better deal with the overdispersion of the time series.

```
# NumberSales part (Poisson):
modPo = dglm.DGLM(type='Poly',order=2,distrib='Poi') +
        dglm.DGLM(type='Seas',frequency=7,distrib='Poi') +
        dglm.DGLM(type='Reg',name=['Promo'],distrib='Poi') +
        dglm.DGLM(type='Reg',name=['LogPrice'],distrib='Poi')
modPo.discFactor=[0.95] # same for the 4 components
modPo.rho=0.3 # 'random effects' discount factor
DF_Po = modPo.Filter(y,CredibleIntervals,dfReg=None)

# DCMM
DF_DCMM = alg.DCMM(DF_Ber,DF_Po)
```

Listing 4.19: Fitting of a DCMM to beer '182' data.

Finally, comparing both models (Listing 4.20) we can see that the univariate model (3.3), offers better point forecasts, as shown in Figure 4.9.

```
alg.ErrorMetrics(y,DF_UniMod['y_t_hat'].values)
alg.ErrorMetrics(y,DF_DCMM['y_t_hat'].values)
alg.plotResults(DF=[DF_UniMod,DF_DCMM],interval=0.9)
CIdataPoisson=alg.CredibleIntervalsCoverage(DF=[DF_UniMod,
             DF_DCMM],CredibleIntervals=CredibleIntervals)
```

Listing 4.20: Comparison of forecasts from model (3.3) and DCMM.

This seems to prove that the proposed univariate model is more flexible than DCMM, and better adapts to time series with high dispersion like the current one (Figure 4.7).

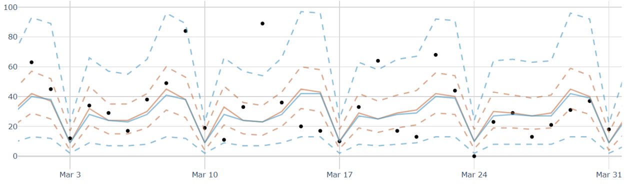


Figure 4.7: Forecasts from DCMM in brown, and model (3.3) in blue.

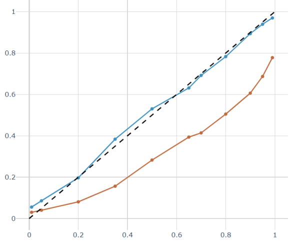
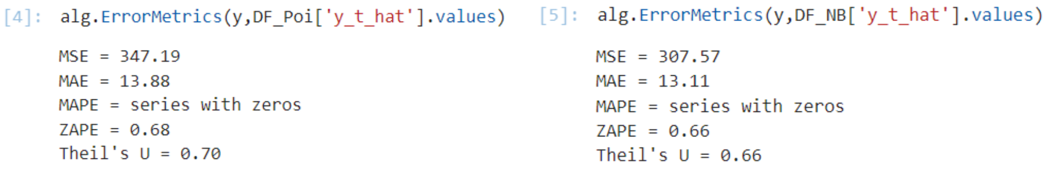


Figure 4.8: Coverage plot of DCMM and model (3.3).

As seen in Figure 4.8, the predictive intervals from our univariate model (blue), at all values, capture extremely well the observations; while the ones from the DCMM (brown) show significant underdispersion.



```
[4]: alg.ErrorMetrics(y,DF_Poi['y_t_hat'].values)

     MSE = 347.19
     MAE = 13.88
     MAPE = series with zeros
     ZAPE = 0.68
     Theil's U = 0.70
```

(a)

```
[5]: alg.ErrorMetrics(y,DF_NB['y_t_hat'].values)

     MSE = 307.57
     MAE = 13.11
     MAPE = series with zeros
     ZAPE = 0.66
     Theil's U = 0.66
```

(b)

Figure 4.9: Error metrics for DCMM, (a), and model (3.3), (b).

4.10 Discussion

We have developed a versatile library that, besides implementing the new models proposed in the thesis, it also covers DLMs and DGLMs in Python and offers further functionality over existing libraries like pyDLM and pyBATS. It allows fitting any model with custom evolution matrices or a combination of commonly used ones (polynomial trend, seasonal, regression, etc.). We offer multiple distributions of the exponential family for the observations (Normal, Bernoulli, Multivariate Bernoulli, Poisson, Negative Binomial and Multivariate Negative Binomial in the current version) that adapt to many different time series, not only the ones with non-negative integer observations which have been central in this thesis. This has primarily been done through the use of an OOP paradigm that implements a class for each family and all the methods necessary for its effortless application to any suitable time series. The practitioner can access to methods like `filter` and `forecast` that provide summary dataframes and plot the results of the corresponding fitting and forecast procedure; and also intermediate ones like `update` that allow using complex time-varying models.

Additionally, we also provide other important functions and algorithms in the library which are essential for its effective use in practical cases: to aid in the initial exploratory analysis and elicit possible models, functions to asses the performance of point forecasts and predictive distributions and performing comparisons, etc. The package developed, also aids in the use of forecast distributions for decision support, through the implementation of the methodology described in Section 3.5. Finally, there are functions that implement several complex models introduced in Chapters 2 and 3, and some brief instructions on how to build new ones that make use of the library are given.

Chapter 5

Conclusions and future work

5.1 Introduction

Time series of counts arise in many different areas like finance, epidemiology, transport or inventory management. Having models that adequately capture and represent the time series and provide accurate forecasts is essential in those domains.

This thesis has contributed to this topic by presenting novel models or novel combinations of previous models for general count time series that could be affected by several effects, and also for count series with frequent zeros and overdispersion. Additionally, we also contribute with algorithms for their effective implementation in a powerful package.

The current chapter provides a synopsis of the developments in Chapters 2, 3 and 4; and proposes new research lines related to the developments in those chapters.

5.2 Summary

General count time series models. Chapter 2 focused on the problem of forecasting general time series of counts, that is, those with relatively high counts and few zero observations, that can incorporate some combination of effects commonly encountered in practical cases.

We provided a methodology to forecast future values based on an initial standard Poisson-Gamma model, suitable for situations in which the Poisson rate remains relatively stable over the period of interest. In most cases, various effects impact the rate evolution. Thus, we adapted the original model by adding specific components (stress effect, seasonal and trend effect, group effect and dependence) and proposed algorithms to forecast with these new models. Several of them need to be combined for certain applications, as shown with the case in Section 2.6.1.

In addition, we have described a model to predict the proportion of future observations that belong to different classes, illustrated with a problem of classifying AS occurrences into the five severity levels classification proposed by the ICAO.

The above models are suitable when all the information about different types of occurrences is available. However, in some cases there is underreporting and part of the observed values are not recorded, which usually affects in different intensities to the classes (e.g a higher proportion of mild cases of a disease go unnoticed than more severe ones). A model is suggested to address these reporting problem.

In the AS application domain, the proposed models have been fundamental in a risk management methodology feeding resource allocation models. They are also important in predicting and monitoring events that allow identifying anomalies related to an unexpected increase (or decrease) in the number of occurrences. In particular, the methodology emphasizes a *management by exception principle* (West & Harrison, 1997) with our models used for routine inference, prediction (and decision support) under standard circumstances until exceptional ones arise in which case an intervention is requested.

The forecasting performance of our models was compared to other popular ones like dynamic linear models (DLM), generalized linear ARMA (GLARMA), and integer-valued GARCH (INGARCH) models, showing better forecasting performance with the AS time series studied. However, some of the models assuming negative binomially distributed observations might be more relevant when exploring approaches at smaller time (weeks) and spatial (airport) resolutions, which might present more overdispersion. Also, given the high safety levels in the aviation system we should expect numerous zero counts and, in particular, models such as those in Berry and West (2020) or the ones we propose in Chapter 4 would be relevant.

Models for count time series with frequent zeros. Chapter 3 explored count time series with frequent zeros and possible overdispersion; as those commonly encountered in highly disaggregated data. We provided a family of models to forecast individual time series of this type, while also discussing how to deal with some peculiarities in series in many application domains.

Moreover, we introduced a multivariate extension of the proposed model that takes advantage of cross-series dependencies, making better use of the available information (in the retail domain for example among products within a shop, among a product at different shops, among shops, etc.). This is done through an extension of multivariate DGLMs, for which we developed the necessary algorithms and methodology to incorporate the new information and obtain forecasts.

Another contribution is the development of a methodology that uses the full predictive distributions several steps ahead of the previous models, to obtain the corresponding predictive cumulative distribution to make informed decisions that avoid critical situations (OoS in inventory management, ICU overrun in epidemic monitoring, etc).

The contributions in the chapter are illustrated with a large demand forecasting problem in the retail industry, and the associated problem of using the predictive distributions to avoid OoS events. The models and methodology introduced improve the performance of models commonly used in this type of count data, as shown in the in-depth comparison in Section 3.6.

countTS. A Python library to support time series forecasting. Chapter 4 focused on the industrial library developed to implement the models presented in this thesis. In addition to the novel models, the library also implements DLMs and DGLMs in Python, and offers further functionality over existing libraries like pyDLM and pyBATS. Models can assume observations following several distributions of the exponential family: Normal, Bernoulli, Multivariate Bernoulli, Poisson, Negative Binomial and Multivariate Negative Binomial. Thus, the package can be used with many different time series, not only the ones with non-negative integer observations which have been central in this thesis. Also we allow to fit any model with custom evolution matrices or with a combination of commonly used ones (polynomial trend, seasonal, regression, etc.), even time-varying.

The implementation in Python has primarily been done through the use of an OOP paradigm, which facilitates future changes and the addition of new functionalities. There is a class for each family and all the methods necessary for its straightforward application to any suitable time series. The practitioner can access methods like `filter` and `forecast` that provide summary tables and plot the results of the corresponding fit and forecasting procedure; and also intermediate ones like `update` that allow using complex time-varying models.

Functions and algorithms to aid in the initial exploratory analysis and elicit possible models in any application domain were also provided. Additionally, the package developed provides functions to asses the performance of point forecasts and predictive distributions and perform comparisons, aid in decision making, etc. Lastly, we illustrated the package and its functionalities with an example concerning a retail time series.

5.3 Future research lines

In this section, we provide a general overview of future research lines within the area of forecasting count time series that seem specially interesting.

Common factors and decouple/recouple strategies. The use of a decouple/recouple strategy that combines models for disaggregated time series and the aggregated one can be quite powerful when modeling multivariate time series (West, 2020) in general, and to incorporate common factors that affect a set of series (Berry & West, 2020) in particular.

In the Bayesian state-space context, this strategy consists of a separate univariate model (DLM or DGLM) with an independent prior for the states for each decoupled time series (therefore its learning and forecast procedure can be in parallel) and a common model for the factors that affect all the series. A *top-down* philosophy is used, that is, the sequential analysis of the model for the common factor of the coupled time series is done first, and the resulting posterior predictive distribution is forwarded/fed to the decoupled univariate models.

It thus seems interesting to apply the common factor ideas to the two types of time series discussed in the thesis and illustrated by the application cases of AS and retail in Sections 2.6 and 3.6, respectively, in conjunction with the proposed multivariate models. We would therefore have a set of multivariate models $\{\mathcal{M}_i\}_{i=1}^n$, each one for a group of similar time series e.g. sales of a family of products like time series of ice cream, and, above them, a model $\mathcal{M}_0$ for the common factor.

Further future work that is worth considering refers to the shared latent factor processes to be multivariate, with dimensions reflecting different ways in which series are conceptually related. This, in addition to the possibility for more than one common factor model that affect the decoupled time series, might present some challenges on their own, specially to maintain scalability and reasonable computing times for the application.

Modeling and computational enhancements. The use of Linear Bayes Estimation and moment matching for updating the moments of the states in DGLMs is well developed and gives satisfactory results (Section 3.6). However, specially for multivariate distributed observations, the use of a different approach might be worth exploring to see if it proves to be more accurate, faster, or more robust and less sensitive to initial prior parameters. Ferreira and Gamerman (2010) provide a brief overview of some alternatives for univariate DGLMs which can be used as a basis for this investigation.

Also, it might be interesting to explore variations of the Particle Filter algorithms proposed for the models in Sections 2.3.2 and 2.4 and how those change the sensitivity to initial prior parameters and computing time.

The forecast methodology proposed in Chapter 3 only accounts for possible overdispersion. In case of also detecting underdispersion in the time series of counts, it might be worth considering the use of Conway–Maxwell–Poisson (CMP) distributions, although the lack of a known closed form conjugate distribution (Kadane et al., 2006) impedes the direct use of the current update procedure, requiring the development of alternative and possibly more computationally expensive methods. The required algorithm should evidently also be included in the `countTS` library. Beside the CMP, another interesting addition would be to include new distributions from the exponential family in the DGLM class implementation.

Also, although in the example in Section 3.6 we did not have the item sales grouped by transaction (*shopping basket*), it could be interesting to combine the proposed models with the ideas in (Berry et al., 2020), and see if it results in an improvement in the forecasts.

Another possible future work to improve the package includes exploring the use of SVD (or other 'square root filtering') for avoiding the calculation of inverse matrices (and possibly obtain even better numerical stability) in the DGLM updating procedure detailed in Section 4.6.2, similar to what is done in this library for DLMs. Also, although it is possible to use DLMs with time-varying evolution matrices with the current version of the library, as explained in Section 4.5.2, these are not as straightforward to implement as other DLMs; future versions would benefit from a modification of the `DLM` class offering a simpler implementation for the user, without the need to program a loop that modifies the corresponding matrix at each time period and then uses the `update` method.

Applications. As we have mentioned throughout the thesis, the proposed model can be applied to other domains where count time series appear, besides Aviation Safety or Retail. Some of the relevant areas that might be interest to explore how well can be forecast with the models introduced, and what changes or additions could be introduced to improve the performance, are:

- **Network Flow Monitoring.** With the explosive growth in the use of internet and social networks over the last two decades this topic is expanding significantly, and with increasingly large-scale data. It is easy to encounter count time series in this context, for example when considering the number of visits to a given web-page, or the number of likes or interactions with a publication (which can also be classified into groups).

  The forecast of this type of data can benefit from the models introduced in Chapters 2 and 3, including the severity forecasting in Section 2.5. Some recent examples of this count series, and the application of Bayesian state space models in this domain can be seen in Chen et al. (2018) and Chen et al. (2019).

- **Epidemiological monitoring** is a topic that has been of special relevance during the ongoing COVID-19 pandemic. There is great interest in having good forecasts for the evolution in the number of cases of an epidemic and, in many occasions, also in the severity of each observation. The infections that might result in ICU admission, hospitalization, or no special treatment. Zhang and Ma (2021) provide a recent example of modeling COVID-19 deaths.

- **Natural disasters**, like earthquakes or hurricanes (Livsey et al., 2018), is another field where some of the models and ideas introduced in this PhD thesis might prove useful. These type are occurrences are quite rare and, therefore, would probably benefit from the models in Chapter 3, and the inclusion of ideas from extreme value theory.

- **Safety and reliability occurrences in other areas** like maritime or road transport (number of highway accidents), industry (number of defective pieces), or supply chain networks, give rise to count time series with many similarities to the ones observed in aviation safety and are an obvious candidate for the models proposed in Chapter 3.

Appendices

Appendix A

Distributions

A.1 Negative Binomial

The variable y_t follows a negative binomial distribution with parameters r_t, p_t, denoted by $y_t \sim NB(r_t, p_t)$, if its probability mass function is

$$NB(y_t|r_t, p_t) = \frac{\Gamma(r_t + y_t)}{y_t!\Gamma(r_t)}(1 - p_t)^{r_t} p_t^{y_t},$$

with support $y_t = \{0, 1, 2, \dots\}$ and parameters $r_t > 0$ and $p_t \in [0, 1]$. The Poisson distribution is a special case, with $r_t \to \infty$.

A.2 Multivariate Negative Binomial

The variable $\boldsymbol{y}_t = (y_{1t}, ..., y_{mt})$ follows a multivariate negative binomial distribution (Arbous & Kerrich, 1951) with parameters r_t, $\boldsymbol{\mu}_t = (\mu_{1t}, ..., \mu_{mt})$, denoted by $\boldsymbol{y}_t \sim MNB(r_t, \boldsymbol{\mu}_t)$, if its probability mass function is

$$MNB(\boldsymbol{y}_t|r_t, \boldsymbol{\mu}_t) = \frac{\Gamma(r_t + \sum_k y_{kt})}{\Gamma(r_t)\prod_k y_{kt}!}\left(\frac{\mu_{1t}}{r_t + \sum_k \mu_{kt}}\right)^{y_{1t}} \cdots \left(\frac{\mu_{mt}}{r_t + \sum_k \mu_{kt}}\right)^{y_{mt}} \left(\frac{r_t}{r_t + \sum_k \mu_{kt}}\right)^{r_t},$$

with support $y_{it} = \{0, 1, 2, \dots\}$ and parameters $r_t, \mu_{it} > 0$. The marginal distribution for y_{it} is $NB(r_t, \mu_{it})$.

Appendix B

Dynamic linear models

A normal dynamic linear model (DLM) for univariate observations X_t, specified by the quadruple $\{\boldsymbol{F}_t, \boldsymbol{G}_t, V_t, \boldsymbol{W}_t\}$, is defined through

$$\begin{aligned} x_t &= \boldsymbol{F}_t\boldsymbol{\theta}_t + v_t, & v_t &\sim N(0, V_t), \\ \boldsymbol{\theta}_t &= \boldsymbol{G}_t\boldsymbol{\theta}_{t-1} + \boldsymbol{w}_t, & \boldsymbol{w}_t &\sim N(\boldsymbol{0}, \boldsymbol{W}_t), \\ \boldsymbol{\theta}_0 &\sim N(\boldsymbol{m}_0, \boldsymbol{C}_0), \end{aligned} \tag{B.1}$$

with v_t and w_t internally and mutually independent (West & Harrison, 1997).

For the univariate DLM, if we denote the available information at the beginning of period t as $D_t = \{D_{t-1}, x_{t-1}\}$, the sequential update and forecast procedure is given by the recursion:

- One-step ahead predictive distribution of $\boldsymbol{\theta}_t$, given D_t. It is $N(\boldsymbol{a}_t, \boldsymbol{R}_t)$, with $\boldsymbol{a}_t = \boldsymbol{G}_t\boldsymbol{m}_{t-1}$ and $\boldsymbol{R}_t = \boldsymbol{G}_t\boldsymbol{C}_{t-1}\boldsymbol{G}_t' + \boldsymbol{W}_t$.

- One-step ahead predictive distribution of x_t, given D_t. It is $N(f_t, Q_t)$, with $f_t = \boldsymbol{F}_t\boldsymbol{a}_t$ and $Q_t = \boldsymbol{F}_t\boldsymbol{R}_t\boldsymbol{F}_t' + V_t$.

- Filtering or posterior distribution of $\boldsymbol{\theta}_t$, given D_t and x_t. It is $N(\boldsymbol{m}_t, \boldsymbol{C}_t)$, with $\boldsymbol{m}_t = \boldsymbol{a}_t + \boldsymbol{R}_t\boldsymbol{F}_t'Q_t^{-1}(x_t - f_t)$ and $\boldsymbol{C}_t = \boldsymbol{R}_t - \boldsymbol{R}_t\boldsymbol{F}_t'Q_t^{-1}\boldsymbol{F}_t\boldsymbol{R}_t$.

References

Adamidis, K. (1999). An EM algorithm for estimating negative binomial parameters. *Australian & New Zealand Journal of Statistics*, *41*, 213–221.

Agarwal, D. K., Gelfand, A. E., & Citron-Pousty, S. (2002). Zero inflated models with application to spatial count data. *Environmental and Ecological Statistics*, *9*, 341—355.

Aktekin, T., Polson, N. G., & Soyer, R. (2018). Sequential Bayesian analysis of multivariate count data. *Bayesian Analysis*, *13*(2), 385–409.

Aktekin, T., & Soyer, R. (2011). Call center arrival modeling: A Bayesian state-space approach. *Naval Research Logistics (NRL)*, *58*(1), 28–42.

Ali, O. G., Sayin, S., Van Woensel, T., & Fransoo, J. C. (2009). SKU demand forecasting in the presence of promotions. *Expert Systems with Applications*, *36*(10), 12340–12348.

Alzaid, A. A., & Al-Osh, M. (1990). An integer-valued pth-order autoregressive structure (INAR(p)) process. *Journal of Applied Probability*, *27*(2), 314–324.

Arbous, A. G., & Kerrich, J. (1951). Accident statistics and the concept of accident-proneness. *Biometrics*, *7*(4), 340–432.

Ayra, E. S., Rios Insua, D., & Cano, J. (2019). Bayesian network for managing runway overruns in aviation safety. *Journal of Aerospace Information Systems*, *16*(2).

Benjamin, M. A., Rigby, R. A., & Stasinopoulos, D. M. (2003). Generalized autoregressive moving average models. *Journal of the American Statis-*

tical Association, *98*(461), 214–223.

Berry, L. R., Helman, P., & West, M. (2020). Probabilistic forecasting of heterogeneous consumer transaction-sales time series. *International Journal of Forecasting*, *36*, 552–569.

Berry, L. R., & West, M. (2020). Bayesian forecasting of many count-valued time series. *Journal of Business and Economic Statistics*, *38*, 872–887.

Box, G. E. P., Jenkins, G. M., Reinsel, G. C., & Ljung, G. M. (2015). *Time series analysis: Forecasting and control.* Wiley.

Carbonnelle, P. (2022). *PYPL popularity of programming language.* Retrieved 14/01/2022, from `https://pypl.github.io/PYPL.html`

Chen, C. W. S., & Lee, S. (2017). Bayesian causality test for integer-valued time series models with applications to climate and crime data. *Journal of the Royal Statistical Society*, *66*(4), 797–814.

Chen, C. W. S., So, M. K. P., Li, J. C., & Sriboonchitta, S. (2016). Autoregressive conditional negative binomial model applied to over-dispersed time series of counts. *Statistical Methodology*, *31*, 73–90.

Chen, X., Banks, D., & West, M. (2019). Bayesian dynamic modeling and monitoring of network flows. *Network Science*, *7*, 292–318.

Chen, X., Irie, K., Banks, D., Haslinger, R., Thomas, J., & West, M. (2018). Scalable Bayesian modeling, monitoring and analysis of dynamic network flow data. *Journal of the American Statistical Association*, *113*, 519–533.

Clemen, R. T., & Reilly, T. (2013). *Making hard decisions with DecisionTools.* Cengage Learning.

Cox, D. R. (1981). Statistical analysis of time series: Some recent developments. *Scandinavian Journal of Statistics*, *8*(2), 93–115.

Croston, J. D. (1972). Forecasting and stock control for intermittent demands. *Journal of the Operational Research Society*, *23*(3), 289–303.

Dai, B., Ding, S., & Wahba, G. (2013). Multivariate Bernoulli distribution. *Bernoulli Society for Mathematical Statistics and Probability*, *19*, 1465–

1483.

Elvira, V., Bernal, F., Hernandez-Coronado, P., Herraiz, E., Alfaro, C., Gomez, J., & Rios Insua, D. (2020). Safer skies over spain. *INFORMS Journal on Applied Analytics*, *50*(1), 21–36.

FAA. (2008). *Wind shear* (Tech. Rep.). Federal Aviation Administration.

Feller, W. (1991). *An introduction to probability theory and its applications.* Wiley.

Ferland, R., Latour, A., & Oraichi, D. (2006). Integer-valued GARCH process. *Journal of Time Series Analysis*, *27*(6), 923–942.

Fernández, F. J., & Harvey, A. C. (1990). Seemingly unrelated time series equations and a test for homogeneity. *Journal of Business & Economic Statistics*, *8*, 71–81.

Ferreira, M. A. R., & Gamerman, D. (2010). *Dynamic generalized linear models.* Retrieved 14/01/2022, from `http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.47.3423`

Gamerman, D., dos Santos, T. R., & Franco, G. C. (2013). A non-gaussian family of state-space models with exact marginal likelihood. *Journal of Time Series Analysis*, *34*(6), 625–645.

Gelman, A., Carlin, J., Stern, H., Dunson, D. B., Vehtari, A., & Rubin, D. (2013). *Bayesian data analysis.* Chapman & Hall/CRC Press.

Gill, P. G., & Buchanan, P. (2013). An ensemble based turbulence forecasting system. *Royal Meteorological Society. Meteorological Applications*, *21*(1), 12–19.

Haslbeck, A., Schmidt-Moll, C., & Schubert, E. (2015). Pilot's willingness to report aviation incidents. In *International symposium on aviation psychology.*

Heinen, A. (2003). Modeling time series count data: An autoregressive conditional poisson model. *SSRN*. Retrieved from `https://ssrn.com/abstract=1117187`

Hyndman, R. J., Koehler, A. B., Ord, J. K., & Snyder, R. D. (2008). *Fore-*

casting with exponential smoothing: The state space approach. Springer.

IATA. (2019). *Economic performance of the airline industry* (Tech. Rep.). International Air Transport Association.

ICAO. (2018). *Safety management manual 4th ed. (Doc 9859)* (Tech. Rep.). International Civil Aviation Organization.

ICAO. (2019). *State of global aviation safety* (Tech. Rep.). International Civil Aviation Organization.

Kadane, J., Shmueli, G., T.P., M., S., B., & P., B. (2006). Conjugate analysis of the Conway-Maxwell-Poisson distribution. *Bayesian Analysis*, *1*(2), 363–374.

Khanmohammadi, S., Tutun, S., & Kucuk, Y. (2016). A new multilevel input layer artificial neural network for predicting flight delays at JFK airport. *Procedia Computer Science*, *95*, 237–244.

Lambert, D. (1992). Zero-inflated Poisson regression, with an application to defects in manufacturing. *Technometrics*, *34*, 1–14.

Livsey, J., Lund, R., Kechagias, S., & Pipiras, V. (2018). Multivariate integer-valued time series with flexible autocovariances and their application to major hurricane counts. *Annals of applied statistics*, *12*, 408–431.

McCabe, B. P. M., & Martin, G. M. (2005). Bayesian predictions of low count time series. *International Journal of Forecasting*, *21*, 315–330.

McCann, D. W. (2005). NNICE – a neural network aircraft icing algorithm. *Environmental Modelling & Software*, *20*(10), 1335–1342.

McKay, A. T. (1934). Sampling from batches. *Supplement to the Journal of the Royal Statistical Society*, *1*(2), 207–216.

McKenzie, E. (2003). Discrete variate time series. *Handbook of Statistics*, *21*, 573–606.

Pedeli, X., & Karlis, D. (2013). Some properties of multivariate INAR(1) processes. *Computational Statistics and Data Analysis*, *67*, 213–215.

Petris, G., Petrone, S., & Campagnoli, P. (2009). *Dynamic linear models with R*. Springer.

Prado, R., & West, M. (2010). *Time series: Modelling, computation and inference.* Chapman & Hall/CRC Press.

Ravishanker, N., Serhiyenko, V., & Willig, M. R. (2014). Hierarchical dynamic models for multivariate times series of counts. *Statistics and Its Interface*, *7*, 559–570.

Rios Insua, D., Alfaro, C., Gomez, J., Hernandez-Coronado, P., & Bernal, F. (2018). A framework for risk management decisions in aviation safety at state level. *Reliability Engineering & System Safety*, *179*, 74–82.

Rios Insua, D., Alfaro, C., Gomez, J., Hernandez-Coronado, P., & Bernal, F. (2019). Forecasting and assessing consequences of aviation safety occurrences. *Safety Science*, *111*, 243–252.

Rios Insua, D., Ruggeri, F., & Wiper, M. (2012). *Bayesian analysis of stochastic process models.* Wiley.

Rios Insua, S., Martin, J., Rios Insua, D., & Ruggeri, F. (1999). Bayesian forecasting for accident proneness evaluation. *Scandinavian Actuarial Journal*, *1999*(2), 134–156.

Schmidt, A. M., & Pereira, J. B. M. (2011). Modelling time series of counts in epidemiology. *International Statistical Review*, *79*(1), 48–69.

Shachter, R. (1988). Probabilistic inference and influence diagrams. *Operations Research*, *36*(4), 589–604.

Shenstone, L., & Hyndman, R. (2005). Hierarchical dynamic models for multivariate times series of counts. *Journal of Forecasting*, *24*, 389–402.

Snyder, R. D., Martin, G. M., Gould, P., & Feigin, P. D. (2008). *An assessment of alternative state space models for count time series* (Tech. Rep.). Monash University, Department of Econometrics and Business Statistics.

Snyder, R. D., Ord, J. K., & Beaumont, A. (2012). Forecasting the intermittent demand for slow-moving inventories: A modelling approach. *International Journal of Forecasting*, *28*(2), 485–496.

Souza, M. A. O., H.S., M., & J.B.M, P. (2018). Extended dynamic generalized linear models: The two-parameter exponential family. *Computational Statistics and Data Analysis*, *121*, 164–179.

Soyer, R., & Zhang, D. (2021). Bayesian modeling of multivariate time series of counts. *Wiley Interdisciplinary Reviews: Computational Statistics*.

Storvik, G. (2002). Particle filters for state-space models with the presence of unknown static parameters. *IEEE Transactions on Signal Processing*, *50*(2), 281–289.

Subramanian, S. V., & Rao, A. H. (2018). Deep-learning based time series forecasting of go-around incidents in the national airspace system. In *Aiaa modeling and simulation technologies conference.*

Wang, L., Liber, G., & Manneback, P. (1992). Kalman filter algorithm based on singular value decomposition. In *31st ieee conference on decision and control.*

West, M. (2020). Bayesian forecasting of multivariate time series: Scalability, structure uncertainty and decisions (with discussion). *Annals of the Institute of Statistical Mathematics*, *72*, 1–44.

West, M., & Harrison, J. (1997). *Bayesian forecasting and dynamic models.* Springer.

West, M., Harrison, P. J., & Migon, H. S. (1985). Dynamic generalised linear models and Bayesian forecasting (with discussion). *Journal of the American Statistical Association*, *80*, 73–97.

Yelland, P. M. (2009). Bayesian forecasting for low-count time series using state-space models: An empirical evaluation for inventory management. *International Journal of Production Economics*, *118*(1), 95–103.

Zhang, X., & Ma, R. (2021). Forecasting waved daily covid-19 death count series with a novel combination of segmented poisson model and arima models. *Journal of Applied Statistics*.