

Sequential Bayesian Analysis of Multivariate Count Data

Tevfik Aktekin^{*}, Nick Polson[†], and Refik Soyer[‡]

Abstract. We develop a new class of dynamic multivariate Poisson count models that allow for fast online updating. We refer to this class as multivariate Poisson-scaled beta (MPSB) models. The MPSB model allows for serial dependence in count data as well as dependence with a random common environment across time series. Notable features of our model are analytic forms for state propagation, predictive likelihood densities, and sequential updating via sufficient statistics for the static model parameters. Our approach leads to a fully adapted particle learning algorithm and a new class of predictive likelihoods and marginal distributions which we refer to as the (dynamic) multivariate confluent hyper-geometric negative binomial distribution (MCHG-NB) and the dynamic multivariate negative binomial (DMNB) distribution, respectively. To illustrate our methodology, we use a simulation study and empirical data on weekly consumer non-durable goods demand.

Keywords: state space, count time series, multivariate poisson, scaled beta prior, particle learning.

1 Introduction

Discrete-valued count data poses a number of statistical modeling challenges with widespread applications in web analytics, epidemiology, economics, finance, operations, and other fields. Amazon, Facebook, and Google are often interested in predicting the number of online customer arrivals during a specific time period. Policy makers are interested in predicting the number of individuals who possess a common trait for resource allocation decisions. Online web data present the challenge of fast and efficient prediction of counts from multiple web pages over time. Moreover, the total number of clicks over time may be dependent across pages requiring dynamic multivariate count data models. In this paper, we develop a class of dynamic (state space) multivariate Poisson model together with a particle filtering (PF) and learning (PL) algorithm for sequential online updating (Gordon et al., 1993; Carvalho et al., 2010a). A scaled beta state evolution with a random common environment is proposed to account for dependence in counts. We term our model the multivariate Poisson-scaled beta (MPSB) and as a by-product, we introduce two new multivariate distributions, the dynamic multivariate negative binomial (DMNB) and the multivariate confluent hyper-geometric negative binomial (MCHG-NB) distributions which correspond to marginal and predictive distributions, respectively.

^{*}University of New Hampshire, Peter T. Paul College of Business and Economics, Durham, NH, tevfik.aktekin@unh.edu

[†]University of Chicago, Booth School of Business, Chicago, IL, ngp@chicagobooth.edu

[‡]The George Washington University, School of Business, Washington, DC, soyer@gwu.edu

Recent advances in discrete-valued time series can be found in Davis et al. (2015). However, there is little work on count data models which allow for serial dependence. The dependence between time series of counts is modeled either using traditional stationary time series models as in Al-Osh and Alzaid (1987); Zeger (1988); Freeland and McCabe (2004) which are known as *observation driven* models in the terminology of Cox (1981), or state space models as in Harvey and Fernandes (1989); Durbin and Koopman (2000); Fruhwirth-Schnatter and Wagner (2006); Aktekin and Soyer (2011); Aktekin et al. (2013); Gamerman et al. (2013) that are called *parameter driven* models. In a state space model, the dependence between the counts is captured via latent factors which follow a stochastic process. These models assume that the counts are conditional independent given the latent factors as opposed to stationary models where counts are unconditionally dependent.

Analysis of discrete-valued multivariate time series has also been limited due to computational challenges. In particular, little attention has been given to multivariate models and our methodology fills this gap. Recent work include Pedeli and Karlis (2011, 2012) who use observation driven multivariate INAR(1) models. Ravishanker et al. (2014) introduce a Bayesian hierarchical multivariate Poisson time series model. Markov chain Monte Carlo (MCMC) methods are used for computation where the evaluation of the multivariate Poisson likelihood requires a significant computational effort. Serhiyenko et al. (2015) develop zero-inflated Poisson models for multivariate time series of counts and Ravishanker et al. (2015) study finite mixtures of multivariate Poisson time series. State space models of multivariate count data were presented in Ord et al. (1993) and were implemented in Jorgensen et al. (1999) using the EM algorithm. Other closely related models of correlated Poisson counts in a temporal setting include research on marked Poisson processes as in Taddy (2010), Taddy and Kottas (2012), and Ding et al. (2012).

One advantage of parameter driven models is that the previous correlations are captured by the time evolution of the state parameter, which we refer to as the *random common environment*. Correlations among multiple series are induced by this random common environment which follows a Markovian evolution; see Smith and Miller (1986); Aktekin et al. (2013); Gamerman et al. (2013). The idea of a random common environment is widely used in risk analysis (Arbous and Kerrich, 1951) and reliability (Lindley and Singpurwalla, 1986) literatures to model dependence. Our strategy provides a new class of dynamic extensions of such models; see also Arbous and Kerrich (1951).

Sequential Bayesian analysis (Polson et al., 2008; Carvalho et al., 2010a) and forecasting requires the use of sequential Monte Carlo techniques. The forward filtering backward sampling (FFBS) algorithm of Carter and Kohn (1994) and Fruhwirth-Schnatter (1994) can be computationally expensive as it requires rerunning of chains to obtain filtering distributions with each additional observation. Particle learning (PL) avoids this burden by providing dynamic state and static posterior distributions in an efficient manner. Carvalho et al. (2010a) observe that estimating static parameters is notoriously difficult especially in higher dimensions. Given the specific structure of our proposed state space model (as the conditional filtering densities of all the static parameters can be obtained in closed form with known conditional sufficient statistic recursions), it is

possible to develop such a filtering scheme that can be used for both on-line updating and forecasting.

The rest of the paper is organized as follows. Section 2 introduces our multivariate time series model for modeling counts and develops its properties. Section 3 introduces the PL algorithm for the MPSB model. Our proposed model and estimation algorithms are illustrated in Section 4 using calibration studies and an actual data set on weekly time series of consumer demand for non-durable goods. Section 5 provides concluding remarks, discussion of limitations and future work.

2 Multivariate Poisson-Scaled Beta (MPSB) Model

Suppose that we observe a sequence of evenly spaced counts observed up until time T for J series denoted by $\{(Y_{11}, \dots, Y_{1T}), \dots, (Y_{J1}, \dots, Y_{JT})\}$. We assume that these J series are exposed to the same external environment similar to the common operational conditions for the components of a system as considered by Lindley and Singpurwalla (1986) in reliability analysis. The analysis of financial and economic time series also includes several series that are affected by the same economic swings in the market. To account for such dependence, we assume a Bayesian hierarchical model of the form

$$(Y_{jt} | \lambda_j, \theta_t) \sim \text{Pois}(\lambda_j \theta_t), \text{ for } j = 1, \dots, J \text{ and } t = 1, \dots, T, \quad (1)$$

where λ_j is the rate specific to the j th series and θ_t represents the effects of the random common environment modulating λ_j . Following Smith and Miller (1986), a Markovian evolution is assumed for θ_t as

$$\theta_t = \frac{\theta_{t-1}}{\gamma} \epsilon_t, \quad (2)$$

where the error term follows a Beta distribution given by

$$(\epsilon_t | D^{t-1}, \lambda_1, \dots, \lambda_J) \sim \text{Beta}[\gamma \alpha_{t-1}, (1 - \gamma) \alpha_{t-1}],$$

where $\alpha_{t-1} > 0, 0 < \gamma < 1$ and $D^{t-1} = \{D^{t-2}, Y_{1,t-1}, \dots, Y_{J,t-1}\}$ represents the sequential arrival of data. We refer to this class of models as multivariate Poisson-scaled beta (MPSB) models due to the relationship between the observation and state equations. Following Smith and Miller (1986), the state equation is defined conditional on previous counts unlike the state equations in traditional dynamic linear models.

2.1 Dynamic Online Bayesian Updating

The observation model (1) is a function of both the dynamic environment θ_t and the static parameters, λ_j 's. In the case where Y_{jt} represents the weekly consumer demand for household j at time t , λ_j accounts for the effects of the household specific rate and θ_t for the effects of the random common economic environment that all households are exposed to at time t . When $\theta_t > 1$, the environment is said to be more favorable than usual which leads to a higher overall Poisson rate and vice versa. In the evolution (2), the term γ acts like a discount factor common for all j series. For notational convenience, we

suppress the dependence of all conditional distributions on γ in our discussion below. Having the state evolution as (2) also implies the following scaled beta density for $(\theta_t|\theta_{t-1}, D^{t-1}, \boldsymbol{\lambda})$

$$\frac{\Gamma(\alpha_{t-1})}{\Gamma(\gamma\alpha_{t-1})\Gamma((1-\gamma)\alpha_{t-1})} \left(\frac{\gamma}{\theta_{t-1}}\right)^{\gamma\alpha_{t-1}} \theta_t^{\gamma\alpha_{t-1}-1} \left(1 - \frac{\gamma}{\theta_{t-1}}\theta_t\right)^{(1-\gamma)\alpha_{t-1}}, \quad (3)$$

where $(\theta_t|\theta_{t-1}, D^{t-1}, \boldsymbol{\lambda})$ is defined over $(0; \theta_{t-1}/\gamma)$ and the vector of static parameters is defined as $\boldsymbol{\lambda} = \{\lambda_1, \dots, \lambda_J\}$. Here we assume given θ_t 's and λ_j , Y_{jt} 's are conditionally independent over time. Furthermore, we assume that at time t , given θ_t and λ_j 's, Y_{jt} 's are conditionally independent of each other.

Conditional on the static parameters, it is possible to obtain an analytically tractable filtering of the states. At time 0, prior to observing any count data, we assume that $(\theta_0|D^0) \sim \text{Gamma}(\alpha_0, \beta_0)$, then by induction we have

$$(\theta_{t-1}|D^{t-1}, \boldsymbol{\lambda}) \sim \text{Gamma}(\alpha_{t-1}, \beta_{t-1}), \quad (4)$$

and using (3) and (4), $(\theta_t|D^{t-1}, \boldsymbol{\lambda})$ is

$$p(\theta_t|D^{t-1}, \boldsymbol{\lambda}) = \int p(\theta_t|\theta_{t-1}, D^{t-1}, \boldsymbol{\lambda})p(\theta_{t-1}|D^{t-1}, \boldsymbol{\lambda})d\theta_{t-1} \quad (5)$$

$$\sim \text{Gamma}(\gamma\alpha_{t-1}, \gamma\beta_{t-1}). \quad (6)$$

The conditional filtering density at time t can be obtained using (1) and (6) as

$$\begin{aligned} p(\theta_t|D^t, \boldsymbol{\lambda}) &\propto p(Y_{1t}, \dots, Y_{Jt}|\theta_t, \boldsymbol{\lambda})p(\theta_t|D^{t-1}, \boldsymbol{\lambda}) \\ &\propto \left(\prod_j (\theta_t \lambda_j)^{Y_{jt}} e^{-\lambda_j \theta_t}\right) \left(\theta_t^{\gamma\alpha_{t-1}-1} e^{-\gamma\beta_{t-1}\theta_t}\right), \end{aligned}$$

which is

$$(\theta_t|D^t, \boldsymbol{\lambda}) \sim \text{Gamma}(\alpha_t, \beta_t), \quad (7)$$

where $\alpha_t = \gamma\alpha_{t-1} + (Y_{1t} + \dots + Y_{Jt})$ and $\beta_t = \gamma\beta_{t-1} + (\lambda_1 + \dots + \lambda_J)$. As a consequence, both the effects of all counts as well as the individual effects of each series are used in updating the random common environment. The filtering density conditional on the static parameters (7) is central for the implementation of our FFBS algorithm which is mainly used for smoothing.

2.2 Dynamic Multivariate Negative Binomial (DMNB) Distribution

An important feature of our model is the availability of the marginal distribution of Y_{jt} conditional on λ_j 's for $j = 1, \dots, J$. This is given by

$$p(Y_{jt}|\boldsymbol{\lambda}, D^{t-1}) = \int p(Y_{jt}|\theta_t, \lambda_j)p(\theta_t|D^{t-1}, \boldsymbol{\lambda})d\theta_t \quad (8)$$

$$= \binom{\gamma\alpha_{t-1} + Y_{jt} - 1}{Y_{jt}} \left(1 - \frac{\lambda_j}{\gamma\beta_{t-1} + \lambda_j}\right)^{\gamma\alpha_{t-1}} \left(\frac{\lambda_j}{\gamma\beta_{t-1} + \lambda_j}\right)^{Y_{jt}}, \quad (9)$$

which is a negative binomial model denoted as $NB[\gamma\alpha_{t-1}, \lambda_j/(\gamma\beta_{t-1} + \lambda_j)]$, where $\lambda_j/(\gamma\beta_{t-1} + \lambda_j)$ is the probability of success. From the conditional independence assumptions, we can obtain the multivariate distribution of $\mathbf{Y}_t = \{Y_{1t}, \dots, Y_{Jt}\}$ conditional on λ_j 's as

$$p(\mathbf{Y}_t | \boldsymbol{\lambda}, D^{t-1}) = \frac{\Gamma(\gamma\alpha_{t-1} + \sum_j Y_{jt})}{\Gamma(\gamma\alpha_{t-1}) \prod_j \Gamma(Y_{jt} + 1)} \prod_j \left(\frac{\lambda_j}{\gamma\beta_{t-1} + \sum_j \lambda_j} \right)^{Y_{jt}} \left(\frac{\gamma\beta_{t-1}}{\gamma\beta_{t-1} + \sum_j \lambda_j} \right)^{\gamma\alpha_{t-1}}. \tag{10}$$

This is a generalization of the negative binomial distribution. We refer to this distribution as the dynamic multivariate negative binomial (DMNB) distribution which will play an important role in learning about the discount parameter, γ . The joint distribution, $p(Y_{it}, Y_{jt} | \boldsymbol{\lambda}, D^{t-1})$, for series i and j , is given by

$$p(Y_{it}, Y_{jt} | \boldsymbol{\lambda}, D^{t-1}) = \frac{\Gamma(\gamma\alpha_{t-1} + Y_{it} + Y_{jt})}{\Gamma(\gamma\alpha_{t-1})\Gamma(Y_{it} + 1)\Gamma(Y_{jt} + 1)} \left(\frac{\gamma\beta_{t-1}}{\lambda_i + \lambda_j + \gamma\beta_{t-1}} \right)^{\gamma\alpha_{t-1}} \left(\frac{\lambda_i}{\lambda_i + \lambda_j + \gamma\beta_{t-1}} \right)^{Y_{it}} \left(\frac{\lambda_j}{\lambda_i + \lambda_j + \gamma\beta_{t-1}} \right)^{Y_{jt}}.$$

We remark that this is a dynamic version of the multivariate negative binomial distribution from Arbous and Kerrich (1951) who modeled the number of industrial accidents in a workplace such as a production facility. The conditional distributions of Y_{jt} 's are also negative binomial distributions. The conditional mean regression of Y_{jt} on Y_{it} is a linear function of Y_{it} , given by

$$E[Y_{jt} | Y_{it}, \boldsymbol{\lambda}, D^{t-1}] = \frac{\lambda_j(\gamma\alpha_{t-1} + Y_{it})}{(\lambda_i + \gamma\beta_{t-1})}. \tag{11}$$

The bivariate counts have positive correlation given by

$$Cor(Y_{it}, Y_{jt} | \boldsymbol{\lambda}, D^{t-1}) = \sqrt{\frac{\lambda_i \lambda_j}{(\lambda_i + \gamma\beta_{t-1})(\lambda_j + \gamma\beta_{t-1})}}. \tag{12}$$

Our proposed model is suitable for positively correlated series. In our examples, we consider counts of weekly demand for consumer non-durable goods of several households that are positively correlated with each other. The structure (12) suggests that as γ approaches zero holding the values of λ_j 's the same, the correlation between two series increases. A similar argument can be made by observing the state (2) where γ was introduced as a common discount parameter. In our simulations and analysis of real count data, we only consider series that are positively correlated and discuss its implications. Even though this is a limitation of our model, it is possible to find positively correlated time series of counts in many fields when the series are assumed to be exposed to the same environment.

2.3 Forward Filtering and Backward Sampling (FFBS)

In what follows, we introduce the forward filtering and backward sampling (FFBS) algorithm for estimating the dynamic state parameters, θ_t 's, the static parameters, λ_j 's, and the discount factor γ . We first assume that γ is known.

We assume that apriori λ_j 's are independent of each other as well as θ_0 and having gamma priors as

$$\lambda_j \sim \text{Gamma}(a_j, b_j), \text{ for } j = 1, \dots, J. \quad (13)$$

The model can be estimated either using MCMC techniques or PF methods. For MCMC, one needs to generate samples from the joint posterior of all parameters as in $p(\boldsymbol{\theta}_t, \boldsymbol{\lambda} | D^t)$ where $\boldsymbol{\theta}_t = \{\theta_1, \dots, \theta_t\}$ using a Gibbs sampling scheme via the following steps

1. Generate θ_t 's via $p(\theta_1, \dots, \theta_t | \lambda_1, \dots, \lambda_j, D^t)$
2. Generate λ_j 's via $p(\lambda_1, \dots, \lambda_j | \theta_1, \dots, \theta_t, D^t)$

In step 1, the FFBS algorithm can be used to estimate the conditional joint distribution of the state parameters where the joint density $p(\theta_1, \dots, \theta_t | \boldsymbol{\lambda}, D^t)$ can be factored as

$$p(\theta_t | \boldsymbol{\lambda}, D^t) p(\theta_{t-1} | \theta_t, \boldsymbol{\lambda}, D^{t-1}) \cdots p(\theta_1 | \theta_2, \boldsymbol{\lambda}, D^1).$$

The implementation of FFBS would be straightforward in our model as we have the following shifted gamma densities where $\gamma\theta_t < \theta_{t-1}$

$$(\theta_{t-1} | \theta_t, \boldsymbol{\lambda}, D^{t-1}) \sim \text{Gamma}[(1 - \gamma)\alpha_{t-1}, \beta_{t-1}].$$

In Step 2, we can use the Poisson-Gamma conjugacy,

$$\begin{aligned} p(\lambda_j | \boldsymbol{\theta}_t, D^t) &\propto p(Y_{j1}, \dots, Y_{jt} | \boldsymbol{\theta}_t, \lambda_j) p(\lambda_j) \\ &\propto \left(\prod_t (\theta_t \lambda_j)^{Y_{jt}} e^{-\lambda_j \theta_t} \right) \left(\lambda_j^{a_j - 1} e^{-b_j \lambda_j} \right), \end{aligned}$$

which is a gamma density as

$$(\lambda_j | \boldsymbol{\theta}_t, D^t) \sim \text{Gamma}(a_{jt}, b_{jt}), \quad (14)$$

where $a_{jt} = a_j + (Y_{j1} + \dots + Y_{jt})$ and $b_{jt} = b_j + (\theta_1 + \dots + \theta_t)$. It is important to observe that given the state parameters, $\boldsymbol{\theta}_t$ and data, λ_j 's are conditionally independent. However, unconditionally they will not necessarily be independent. The availability of (14) and more importantly the sequential updating of its parameters using sufficient statistics is important in developing PL methods which we discuss in detail in the sequel.

As pointed out by Storvik (2002) and Carvalho et al. (2010a), the issue with MCMC methods in state space models is that the chains need to be restarted for every data point

observed and the simulation dimension becomes larger as we observe more data over time. Furthermore, MCMC methods require convergence of chains via the calibration of thinning intervals (to reduce autocorrelation of the samples) and the determination of the burn-in period's size, both of which would increase the computational burden. Therefore, using MCMC methods would not be ideal for sequential updating whose implications we investigate in our numerical example section. However, the FFBS algorithm can be used to obtain smoothing estimates in a very straightforward manner since, unlike filtering, smoothing does not require sequentially restarting the chains. In a single block run of the above FFBS algorithm, one can obtain estimates of $(\theta_1, \dots, \theta_t | D^t)$ by collecting the associated samples generated from $p(\theta_1, \dots, \theta_t, \boldsymbol{\lambda} | D^t)$. When fast sequential estimation is of interest, an alternative approach is the use of PF techniques that are based on the idea of re-balancing a finite number of particles of the posterior states given the next data point with weights that are proportional to its likelihood.

3 Particle Learning of the MPSB Model

For sequential state filtering and parameter learning, we make use of the PL method of Carvalho et al. (2010a) to update both the dynamic and the static parameters. To summarize, the PL approach starts with the resampling of all particles at time t using weights proportional to the predictive likelihood which ensures that the highly likely particles are moved forward. The resampling step is followed by the propagation of the current state (t) to the future state ($t+1$). Note that in both the resampling and propagation steps, one-step-ahead observations are used. The last step involves updating the static parameters by computing the conditional sufficient statistics. Even though there has been several applications of the PL methods in the literature, almost none of them focus on the analysis of Poisson count data with the exception of Taddy (2010). Among many other successful applications, some recent work of the PL algorithm include Carvalho et al. (2010b) for estimating general mixtures, Gramacy and Polson (2011) for estimating Gaussian process models in sequential design and optimization, and Lopes and Polson (2016) for estimating fat-tailed distributions.

Let us first assume that γ is known and define z_t as the essential vector of parameters that we need to keep track of at each t . The essential vector will consist of the dynamic state parameter (θ_t), static parameters ($\boldsymbol{\lambda}$) and conditional sufficient statistics $s_t = f(s_{t-1}, \theta_t, \mathbf{Y}_t)$ for updating the static parameters. The fully adapted version of PL can be summarized as follows using the traditional notation of PF methods

1. (Resample) $\{z_t\}_{i=1}^N$ from $z_t^{(i)} = \{s_t, \boldsymbol{\lambda}\}^{(i)}$ using weights $w_t^{(i)} \propto p(\mathbf{Y}_{t+1} | z_t^{(i)})$
2. (Propagate) $\{\theta_t^{(i)}\}$ to $\{\theta_{t+1}^{(i)}\}$ via $p(\theta_{t+1} | z_t^{(i)}, \mathbf{Y}_{t+1})$
3. (Update) $s_{t+1}^{(i)} = f(s_t^{(i)}, \theta_{t+1}^{(i)}, \mathbf{Y}_{t+1})$
4. (Sample) $(\boldsymbol{\lambda})^{(i)}$ from $p(\boldsymbol{\lambda} | s_{t+1}^{(i)})$

In step 1, z_t will be stored at each point in time and only includes one state parameter (θ_t), hence eliminating the need to update all state parameters ($\boldsymbol{\theta}_t$) jointly for each time new data is observed. In step 1, s_t represents the ensemble of conditional sufficient statistics for updating the static parameters. From (14), it is easy to see that s_t should only consist of \mathbf{Y}_t and θ_t if we rewrite $a_{jt} = a_{j,t-1} + Y_{jt}$ and $b_{jt} = b_{j,t-1} + \theta_t$ for each j . In step 3, $f(\cdot)$ represents this deterministic updating of the conditional sufficient statistic based on the a_{jt} and b_{jt} recursions.

In order to implement the above PL scheme, we need $p(\mathbf{Y}_{t+1}|z_t^{(i)})$, the predictive likelihood, for computing the weights in step 1 and $p(\theta_{t+1}|z_t^{(i)}, \mathbf{Y}_{t+1})$, the propagation density, for step 2. Note that this propagation density is not the same state evolution equation from (3) due to the inclusion of \mathbf{Y}_{t+1} in the conditioning argument, which ensures that the most current data is considered in propagating the states. Next, we present the summary of how these two quantities can be obtained. A detailed version can be found in Supplementary Appendix A (Aktekin et al., 2017). We also show in detail the conjugate nature of our model in Supplementary Appendix B with the detailed steps of how the dynamic multivariate version was obtained starting with the static univariate model.

Step 1: Obtaining the Resampling Weights

The predictive likelihood is denoted by $p(\mathbf{Y}_{t+1}|z_t) = p(\mathbf{Y}_{t+1}|\theta_t, \boldsymbol{\lambda}, D^t)$ and is required to compute the resampling weights in step 1 of the above PL algorithm. Specifically, we need to compute

$$w_t = p(\mathbf{Y}_{t+1}|\theta_t, \boldsymbol{\lambda}, D^t) = \int p(\mathbf{Y}_{t+1}|\theta_{t+1}, \boldsymbol{\lambda})p(\theta_{t+1}|\theta_t, \boldsymbol{\lambda}, D^t)d\theta_{t+1},$$

where $p(\mathbf{Y}_{t+1}|\theta_{t+1}, \boldsymbol{\lambda})$ is the product of the Poisson likelihoods (1) and $p(\theta_{t+1}|\theta_t, \boldsymbol{\lambda}, D^t)$ is the state (3). Thus, w_t would be

$$w_t = \left(\prod_j \frac{\lambda_j^{Y_{j,t+1}}}{Y_{j,t+1}!} \right) \left(\frac{\theta_t}{\gamma} \right)^{\sum_j Y_{j,t+1}} \left(\frac{\Gamma(\sum_j Y_{j,t+1} + \gamma\alpha_t)\Gamma(\alpha_t)}{\Gamma(\sum_j Y_{j,t+1} + \alpha_t)\Gamma(\gamma\alpha_t)} \right) CHF(a; a + b; -c), \quad (15)$$

where $a = \sum_j Y_{j,t+1} + \gamma\alpha_t$, $a + b = \sum_j Y_{j,t+1} + \alpha_t$, $c = \frac{\theta_t}{\gamma} \sum_j \lambda_j$. Here, CHF represents the confluent hyper-geometric function of Abramowitz and Stegun (1968). For evaluating the CHF function, fast computation methods exist; see for instance the *gsl* package in R by Hankin (2006). The resampling weights (15) also represent the predictive likelihood (marginal) for the proposed class of dynamic multivariate Poisson models. To the best of our knowledge, (15) represents the form of a new multivariate distribution which we refer to as (dynamic) multivariate confluent hyper-geometric negative binomial distribution (MCHG-NB); see the Supplementary Appendix B for the details.

Step 2: Obtaining the Propagation Density

The propagation density in step 2 of the PL algorithm can be shown to be

$$p(\theta_{t+1}|\theta_t, \boldsymbol{\lambda}, \mathbf{Y}_{t+1}, D^t) \propto \theta_{t+1}^{(\sum_j Y_{j,t+1}) + \gamma\alpha_t - 1} \left(1 - \frac{\gamma}{\theta_t}\theta_{t+1}\right)^{(1-\gamma)\alpha_t - 1} e^{-(\sum_j \lambda_j)\theta_{t+1}}.$$

The above form is proportional to a scaled hyper-geometric beta density as in Gordy (1998) and is defined over the range $(0; \frac{\theta_t}{\gamma})$, as $HGB(a, b, c)$, with parameters

$$a = (\sum_j Y_{j,t+1}) + \gamma\alpha_t, b = (1 - \gamma)\alpha_t \text{ and } c = \frac{\theta_t}{\gamma} \sum_j \lambda_j.$$

To generate samples from the HGB density, it is possible to use a rejection sampling based approach. First, we can numerically evaluate the maximum of the HGB density over $(0,1)$ using a non-linear numerical search technique and use the maximum as an enveloping constant for developing a rejection sampling algorithm. We comment on the performance of the sampling method in our numerical section and also provide an alternative below.

Now that we have both the predictive likelihood for computing the resampling weights and the propagation density, the PL algorithm can be summarized as

1. (Resample) $\{z_t\}_{i=1}^N$ from $z_t^{(i)} = \{s_t, \boldsymbol{\lambda}\}^{(i)}$ using weights

$$w_t^{(i)} \propto \left(\prod_j \frac{\lambda_j^{Y_{j,t+1}}}{Y_{j,t+1}!}\right) \left(\frac{\theta_t}{\gamma}\right)^{\sum_j Y_{j,t+1}} \left(\frac{\Gamma(\sum_j Y_{j,t+1} + \gamma\alpha_t)\Gamma(\alpha_t)}{\Gamma(\sum_j Y_{j,t+1} + \alpha_t)\Gamma(\gamma\alpha_t)}\right) CHF(a; a+b; -c)$$
2. (Propagate) $\{\theta_t^{(i)}\}$ to $\{\theta_{t+1}^{(i)}\}$ via $HGB[(\sum_j Y_{j,t+1}) + \gamma\alpha_t, (1 - \gamma)\alpha_t, \sum_j \lambda_j]$ defined over $(0; \frac{\theta_t}{\gamma})$
3. (Update) $s_{t+1}^{(i)} = f(s_t^{(i)}, \theta_{t+1}^{(i)}, \mathbf{Y}_{t+1})$
4. (Sample) $(\boldsymbol{\lambda})^{(i)}$ from $p(\lambda_j | s_{t+1}^{(i)}) \sim \text{Gamma}(a_{j,t+1}, b_{j,t+1})$ for $j = 1, \dots, J$.

The availability of the recursive updating for the sufficient statistics of the static parameters makes our model an ideal candidate for applying the PL method. Note that in step 4, the conditional distributions of the static parameters are coming from (14). Alternatively, if generating from the HGB distribution in step 2 is not computationally efficient, then one can use another step in the vein of sequential importance sampling by resampling the θ_{t+1} 's using weights proportional to the likelihood. For instance, we can replace step 2 in the above with

- (Propagate) $\{\tilde{\theta}_{t+1}\}^{(i)}$ from $p(\theta_{t+1}|\theta_t^{(i)}, \boldsymbol{\lambda}^{(i)}, D^t)$
- (Resample) $\{\theta_{t+1}\}^{(i)}$ using weights $w_{t+1} \propto p(\mathbf{Y}_t | \tilde{\theta}_{t+1}^{(i)}, \boldsymbol{\lambda}^{(i)})$

We comment on the performance of the above approach in our numerical example.

Updating the Discount Factor γ

For the sequential estimation of the γ posterior at each point in time, we make use of the availability of the marginal likelihood conditional on the λ_j 's which is a dynamic multivariate negative binomial density. Estimation of a static parameter that does not evolve over time is surprisingly challenging in a PL context. It is not possible to incorporate the estimation of γ in step 5 of the above algorithm using an importance sampling step as it will lead to the well known particle degeneracy issue. Unlike the λ_j 's, the conditional posterior distribution of γ is not a known density with deterministic conditional recursive updating. Therefore, for models where γ is treated as an unknown quantity, we suggest the use of the marginal likelihood conditional on the λ_j 's from (9). Therefore, we can write the conditional posterior distribution of γ as

$$p(\gamma = k | \boldsymbol{\lambda}, D^{t+1}) \propto \prod_{i=1}^{t+1} p(\mathbf{Y}_i | \boldsymbol{\lambda}, D^{i-1}, \gamma = k) p(\gamma = k), \quad (16)$$

where $p(\gamma = k)$ is a discrete uniform prior defined over $(0.001, 0.999)$ with K categories (we comment on determining the dimension of K in our simulation studies). To incorporate the learning of (16) at the end of step 4 of our PL algorithm above, we first estimate the discrete posterior distribution of γ using the Monte Carlo average of the updated samples of $\lambda_1, \dots, \lambda_J$ at time $t + 1$. Then, we resample particles from this distribution to update $f(\cdot)$ in step 3 at time $t + 2$.

4 Numerical Examples

To illustrate our MPSB model and the associated estimation algorithms, we consider several simulation studies and an actual data on consumer demand for two households. The consumer demand data we were given access to is a subset of a large set used in Kim (2013). The data as well as the R code are available upon request via email from the authors.

4.1 Example: Calibration Study

First, we present the results of several simulated studies. We constructed 10 simulated sets from the data generating process of the MPSB given by (1) and (2). Each simulated set contains a total of 200 observations that consist of 40 data points for 5 series. Each sequence of counts sampled from the model are realizations from the underlying time series model with varying pairwise sample correlations among individual series. The parameter values are unchanged but each simulated set behaves differently as the random common environment differs drastically across simulations even for the same values of the static parameters.

To initialize the simulations, we set $\theta_0 \sim G(\alpha_0 = 10, \beta_0 = 10)$ representing the initial status of the random common environment. We explicitly assume that the random common environment is initialized around the unit scale (with mean $\alpha_0/\beta_0 = 1$). In

doing so, one obtains a better understanding of the scale of the static parameters, λ_j 's, as a function of actual count data. This is especially important when dealing with real count data when specifying the hyper-parameters of priors for θ_0 and the λ_j 's which we discuss in the sequel. We assumed that $J = 5$, and the static parameters, λ_j 's, were 2, 2.5, 3, 3.5, and 4, respectively. The values are close to each other to investigate if the model can distinguish these static parameters. Finally, the common discount parameter, γ was set at 0.30.

Our PL algorithm uses $N=1,000$ particles. Since all simulated counts are roughly between 0 and 40 with initial values up to 5-6, we set $\theta_0 \sim G(10, 10)$ and $\lambda_j \sim G(2, 1)$ for all j (reflecting the fact that very high values of the parameter space does not make practical sense). Our numerical experiments revealed that having tighter priors especially on λ_j 's help identifying the true value of the parameters. Varying the hyper-parameters of the priors (within reasonable bounds with respect to the scale of the counts) does not have a significant effect on the overall fit of the models. When the priors are vague and uninformative (e.g. $G(0.001, 0.001)$), our algorithm has difficulty identifying regions close to the real values of the parameters at the outset. However, in such cases the mean filtered estimates, $E(\theta_t \lambda_j | D^t)$'s, are found to be in the near proximity of the real counts. When dealing with real data, this is not a major drawback as long as the model is able to provide reasonable filtering estimates since the true value of the static parameters will always be unknown. For practical reasons, we suggest that the initial state prior be set around the unit scale as in $\theta_0 \sim G(10, 10)$. We note here that the results were not sensitive to changes in the hyper-parameters of θ_0 as long as its mean stayed around the region of unit scale such as those in $G(1, 1), G(10, 10)$ or $G(100, 100)$.

	Sim #1	Sim #2	Sim #3	Sim #4	Sim #5
λ_1	2.06 (1.83;2.29)	2.05 (1.83;2.25)	1.70 (1.40;2.05)	1.66 (1.42;1.93)	1.91 (1.73;2.10)
λ_2	2.32 (2.09;2.57)	2.64 (2.39;2.87)	2.22 (1.85;2.61)	2.20 (1.92;2.51)	2.52 (2.31;2.75)
λ_3	2.69 (2.43;2.97)	2.79 (2.54;3.05)	2.75 (2.37;3.19)	2.38 (2.11;2.70)	2.99 (2.76;3.24)
λ_4	2.97 (2.70;3.24)	3.54 (3.27;3.86)	2.95 (2.53;3.39)	2.58 (2.27;2.93)	3.42 (3.17;3.67)
λ_5	3.54 (3.24;3.85)	3.80 (3.52;4.08)	3.57 (3.08;4.02)	3.19 (2.82;3.55)	3.72 (3.47;3.99)
γ	0.19 (0.14;0.32)	0.20 (0.12;0.32)	0.37 (0.22;0.50)	0.25 (0.15;0.43)	0.26 (0.15;0.43)
	Sim #6	Sim #7	Sim #8	Sim #9	Sim #10
λ_1	2.13 (1.83;2.46)	1.96 (1.67;2.26)	2.25 (1.91;2.59)	2.17 (1.94;2.39)	2.01 (1.81;2.21)
λ_2	2.81 (2.47;3.19)	2.40 (2.10;2.71)	2.50 (2.17;2.87)	2.48 (2.25;2.77)	2.67 (2.43;2.92)
λ_3	3.01 (2.64;3.37)	2.81 (2.47;3.17)	2.97 (2.59;3.35)	2.72 (2.47;2.98)	3.05 (2.76;3.30)
λ_4	3.57 (3.18;3.98)	3.26 (2.88;3.65)	3.38 (3.00;3.83)	3.36 (3.08;3.66)	3.25 (2.99;3.53)
λ_5	4.29 (3.87;4.75)	3.75 (3.35;4.19)	3.96 (3.53;4.44)	3.52 (3.23;3.84)	3.77 (2.50;4.05)
γ	0.28 (0.18;0.45)	0.28 (0.17;0.45)	0.26 (0.16;0.41)	0.22 (0.14;0.33)	0.22 (0.14;0.33)

Table 1: Posterior means and 95% credibility intervals (in parenthesis) for static parameters across 10 simulated examples.

Table 1 shows the means and 95% credibility intervals (in parenthesis) for the estimated static parameters for 10 different simulations. For each case, the PL algorithm is able identify posterior distributions that are close to the true values of the parameters

($\lambda_1 = 2, \lambda_2 = 2.5, \lambda_3 = 3, \lambda_4 = 3.5, \lambda_5 = 4$ and $\gamma = 0.3$). In addition, we also computed posterior coverage probabilities across 10 simulations by investigating if the true value of the parameter was within the 95% credibility bounds. (i.e. the number of times the true values of the parameter was within a given credibility interval across 10 simulations). These coverage probabilities were estimated to be 0.9, 1.0, 0.7, 0.7 and 0.7 for the λ_j 's and 1.00 for γ , showing support in favor of the algorithm being able to provide coverage of the true values most of the time.

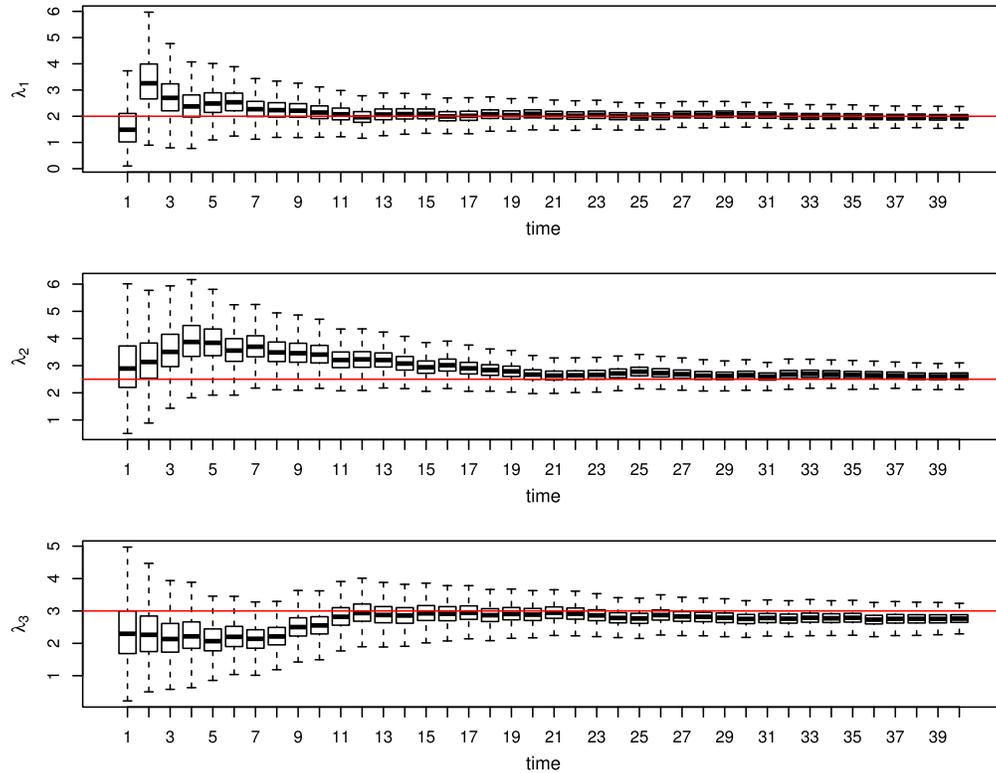


Figure 1: Estimation paths of the static parameters, $\lambda_1, \lambda_2, \lambda_3$ over time for a given simulation (red straight line represents the real value of the parameter).

Figures 1 and 2 show the boxplots of the estimation paths of the static parameters for one of the simulations where the straight line represents the true value of the parameter. As can be observed from the size of the boxplots, for the first few observations the posterior distributions exhibit more uncertainty. As we observe more data, the uncertainty tapers off and the posterior distributions converge to regions close to the true value of the parameters (similar plots were obtained for all 10 simulations). After observing up to 9–10 points in time, our algorithm is able to learn about the λ_j 's very easily, however learning of the γ takes a few more observations. The dip in the value of γ around time period 10 may be attributed to the jump we observe in the simulated

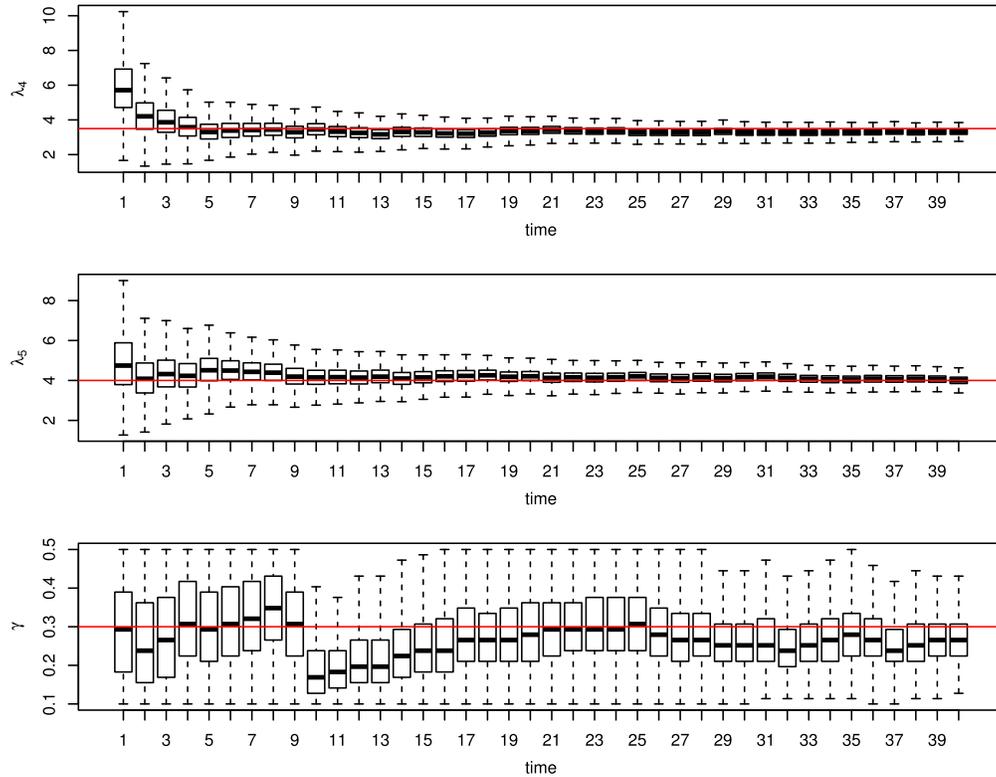


Figure 2: Estimation paths of the static parameters, $\lambda_4, \lambda_5, \gamma$ over time for a given simulation (red straight line represents the real value of the parameter).

counts in 4 out of 5 series that can be observed in Figure 4 (from time period 9 to 10) since a lower value of γ implies a higher correlation in our model. After a few more observations, the posterior γ goes back to exploring regions around its true value.

The final posterior density plots of $\lambda_1, \dots, \lambda_5$ after observing all the data are shown in the top panel of Figure 3 for one of the simulations. All of the density plots cover the true value of the parameter as indicated by the vertical straight lines. The posterior distribution of γ from Figure 3 also shows that most of its support is close to the region of 0.30 which is the actual value of γ . The posterior mode was between 0.25 and 0.30 and the mean was estimated to be 0.27 (as there is more support on the left side of the true value in the posterior distribution). In our proposed algorithm, the estimation of γ discussed in (16) requires that we put a reasonably large value for K which is the number of discrete categories for γ . For a discrete uniform prior defined over the region (0.001; 0.999), we experimented with different values for K and explored cases when $K = 5, 10, 30, 50, 100$ and 500. For all 10 simulations, the posterior distributions were almost identical when K was 30 or larger. For relatively smaller values of K as in 5 and 10, the posterior distribution did not mix well and did not explore regions wide enough

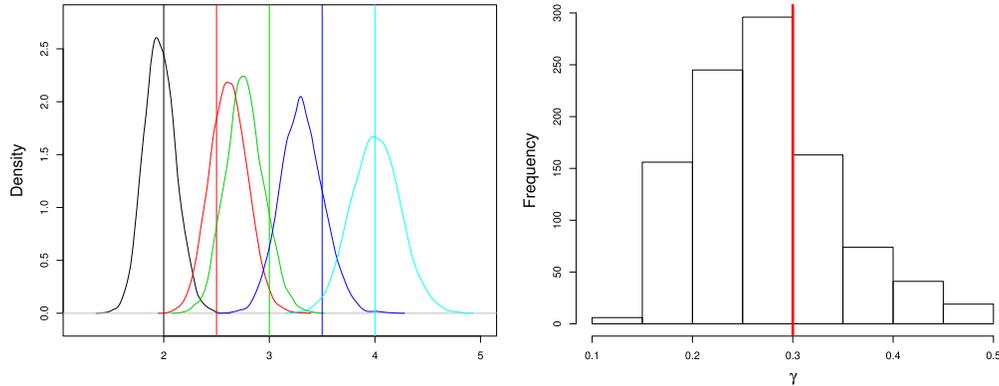


Figure 3: Posterior distributions of the static parameters, $\lambda_1, \dots, \lambda_5$ (left) and γ (right), for a given simulation. Vertical lines represent the real value of the parameter.

	Sim #1	Sim #2	Sim #3	Sim #4	Sim #5
MAPE	0.19	0.14	0.31	0.22	0.18
	Sim #6	Sim #7	Sim #8	Sim #9	Sim #10
MAPE	0.21	0.25	0.25	0.20	0.23

Table 2: Summary of MAPEs for all simulations.

for converging to the right distribution. In cases when fast estimation is of interest, we suggest that K is kept in the region of 30–40 since increasing its dimension leads to losses in estimation speed due to the fact that the negative binomial likelihood needs to be evaluated for each point in time equal to “ $K \times$ number of particles”.

Another noteworthy investigation is how good our estimated filters are with respect to actual data across simulations. To assess the model fit, we first computed the absolute percentage error (APE) for each simulation (a total of 200 observations for each simulation where each one of the 5 series consists of 40 observations) and computed the median of these APEs. The results are shown in Table 2 where the estimates range between 14% and 25%. The reason we report the median instead of the mean APEs is the presence of some outliers which skew the results immensely. Typically the APE estimates range between 0 and 0.30 and some outliers are in the range of 3–4, which when we take the average of, show very misleading results. When we plotted the histograms of APEs for each simulation, we were able to observe that the median and the mode of the distributions were very close to each other with the means located away from these two measures due to one or two very highly influential percentage error values in the right tail of the distributions. We did not report the mean squared errors (MSE) as they would not be comparable across simulations since the scale of the counts vary from one simulation to another even for the same values of the static parameters.

Figure 4 shows the posterior means of the filtered rates, $E(\theta_t \lambda_j)$'s, at each point in time versus the actual counts for a given simulated example. In this example, the

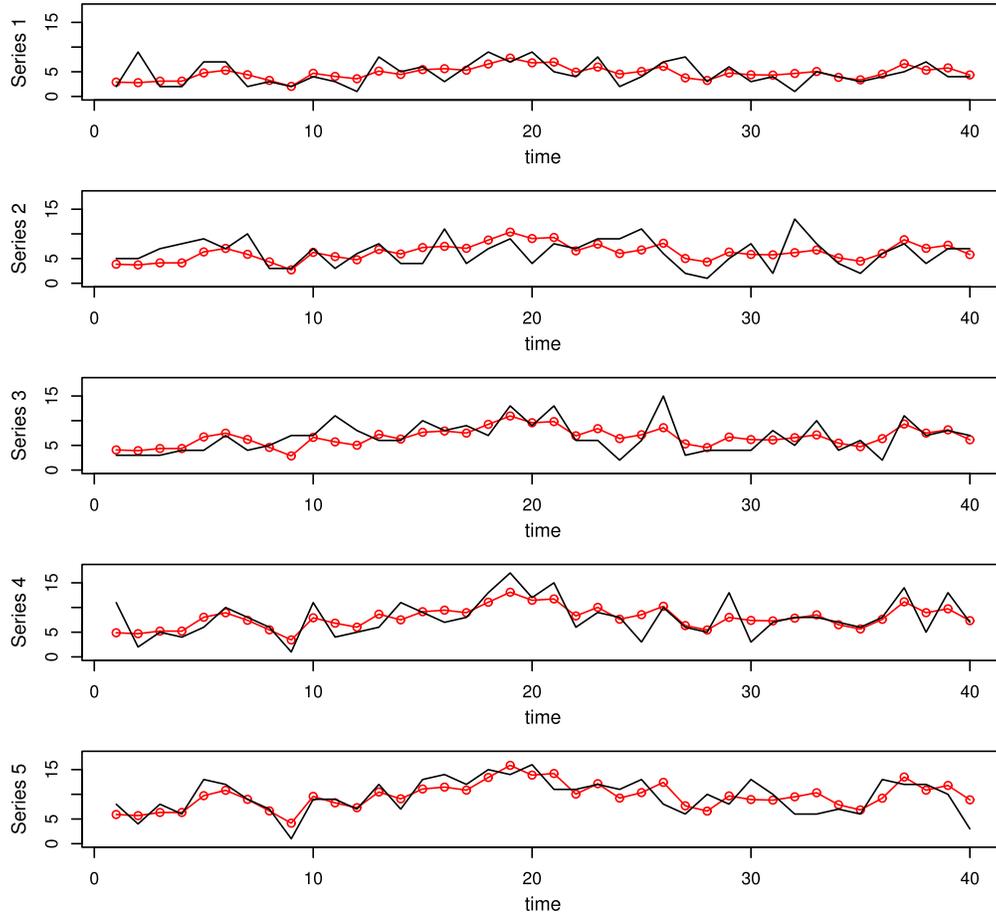


Figure 4: Mean filtered states, $E(\theta_t \lambda_j | D^t)$'s (red), versus the actual counts (straight line) for one of the simulated examples.

series were moderately correlated with sample pairwise correlations ranging between 0.59 and 0.69. The model is able to capture most swings except for rare cases when all five series do not exhibit similar (upward/downward) patterns at a given point in time. For instance, around roughly time period 9, the counts for series 1, 2, 4 and 5 exhibit a drop whereas series 3 shows an increase. As the dependency across series is based on the random common environment idea, the filtered states around time period 9 exhibit a decay for all 5 series (not only for series 1, 2, 4 and 5). Such disagreements lead to extremely large APE estimates as discussed before but are usually no more than 1–2 times in a given simulated set.

Figure 5 shows the stochastic evolution of the state of the random common environment over time that all five series have been exposed to (i.e. $p(\theta_t | D^t)$) which is free

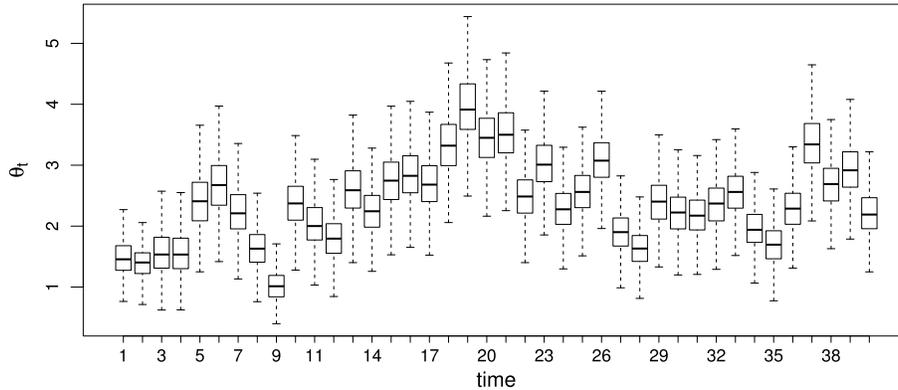


Figure 5: Filtered stochastic evolution for the state of the environment, $(\theta_t|D^t)$'s, over time for one of the simulated examples.

of the static parameters) for a given simulation study. For instance, such a common environment could represent the economic environment financial and economic series are exposed to with swings representing local sudden changes in the market place. In our model, θ_t s dictate the autocorrelation structure of the underlying state evolution and they induce correlations among the 5 series. The sample partial autocorrelation estimate at lag 1 for the mean of these posterior state parameters was between 0.80 and 0.90 indicating a strong first order Markovian behavior in the random common environment.

As a final exercise, we also used the FFBS algorithm introduced in Section 2.3 to generate the full posterior joint distribution of the model parameters for each time period t as in, $p(\theta_1, \dots, \theta_t, \lambda_1, \dots, \lambda_j | D^t)$. As pointed out by Storvik (2002), for any MCMC based sampling method dealing with sequential estimation, the chains would need to be restarted at each point in time. In addition, issues of convergence, thinning and the size of the burn-in periods would need to be investigated. Therefore, using the FFBS algorithm would not be preferred over the PL algorithm when fast sequential estimation would be of interest as in the analysis of streaming data in web applications. To show the differences in computing speed, we estimated one of the simulated examples using both algorithms. The models were estimated on a PC running Windows 7 Professional OS with an Intel Xeon @3.2GHz CPU and 6GBs of RAM. The PL algorithm takes about 17.25 (or 58.7) seconds with 1,000 (or 5,000) particles and the FFBS algorithm takes about 270.74 seconds for 5,000 collected samples (with a thinning interval of 4) where the first 1,000 are treated as the burn-in period. In both cases, we kept γ fixed at 0.30 even though the computational burden for its estimation with the FFBS algorithm would have been higher with “ $K \times$ Number of Samples generated=5,000” versus “ $K \times$ Number of particles=1,000”. We also note that the posterior means of the static parameters using the FFBS model were close to those estimated with the PL algorithm from Table 1. A summary of posterior statistics for one of the simulated sets is shown in Table 3 for the FFBS algorithm. We view the FFBS algorithm as an

	λ_1	λ_2	λ_3	λ_4	λ_5
Mean	1.92	2.64	3.11	3.51	3.70
95% CI	(1.56;2.32)	(2.18;3.15)	(2.57;3.71)	(2.91;4.16)	(3.05;4.28)

Table 3: Posterior means and 95% credibility intervals (in parenthesis) for static parameters for simulation no: 5 using the FFBS algorithm.

alternative when smoothing is of interest which can be handled in a straightforward manner as discussed in Section 2.3. For sequential filtering and prediction, we would prefer the PL algorithm due to its computational efficiency. We would like to note that the results summarized above are based on the version of our algorithm which uses the sequential importance sampling step for the state propagation instead of the rejection sampling method discussed in Step 2 of our PL algorithm. Even though the results were identical in both cases, the computational burden for the rejection sampling algorithm was very high in some cases. Our numerical experiments revealed that the acceptance rate of the sampler became extremely small for certain values of the HGB density parameters. Therefore, unless a very efficient way of generating samples from the HGB density can be developed, we suggest the use of the extra importance sampling step in implementing our PL algorithm.

Sequential Predictive Likelihood Computation

Another important feature of the PL approach is the availability of sequential predictive likelihood computations that can be used for model assessment in obtaining estimates for (sequential) Bayes factors. To show the implementation of this feature, we considered a simple model comparison for one of our simulated examples where the competing model is a series of static univariate Poisson densities with Gamma priors on the rates as in $(Y_{jt}|\lambda_j) \sim Pois(\lambda_j)$ with flat priors set as $\lambda_j \sim G(0.001, 0.001)$. The marginal likelihood of the static univariate model will be straightforward to evaluate due to the Poisson-Gamma-Negative Binomial relation. In our model the sequential predictive densities can be evaluated using

$$p(\mathbf{Y}_t|D^{t-1}) = \frac{1}{N} \sum_{i=1}^N p(\mathbf{Y}_t|\boldsymbol{\lambda}^{(i)}, D^{t-1}),$$

where N is the size of the particle set and $p(\mathbf{Y}_t|\boldsymbol{\lambda}, D^{t-1})$ is given by (9). Thus, the marginal likelihood can be evaluated as

$$p(\mathbf{Y}_1, \dots, \mathbf{Y}_T) = \prod_{j=1}^T p(\mathbf{Y}_j|D^{j-1}).$$

Using the counts of one of the simulated sets, we computed the marginal likelihoods in the log-scale for both our model and the univariate static Poisson models as $-10,784$ and $-267,462$, respectively. As expected, accounting for temporal variations leads to better model fit as evidenced by the estimated marginal likelihoods. PL allows the

predictive likelihoods to be evaluated directly without the need for repeating the same process by T times unlike MCMC based methods. Johannes et al. (2009) remark the importance of sequential monitoring of model fit using particle methods and considers an example of modeling stochastic volatility where a simpler model significantly under performs their model when a large shock is observed in the system.

4.2 Example: Weekly Consumer Demand Data

To show the application of our model with actual data, we used the weekly demand for consumer non-durable goods (measured by the total number of trips to the super market) of two households in the Chicago region over a period of 104 weeks (an example for a bivariate model). Therefore, in this illustration, Y_{jt} for $t = 1, \dots, 104$ and $j = 1, 2$ are the demand of household j during the time period t , θ_t represents the common economic environment that the households are exposed to at time t and λ_j represents the individual random effect for household j . The example is suitable for our proposed model since a quick empirical study of the data revealed that weekly demand of these households exhibit correlated behavior over time (temporal dependence) as well as across households (dependence from the random common environment). The sample correlation between the two series was estimated to be 0.41 which is in line with our model structure that requires positively correlated counts. In addition, the partial auto-correlation functions of both series also show significant correlations at lag 1, justifying our use of the first order Markovian evolution equation for the states. As before, we estimated the model using 1,000 particles and used similar priors. Specifically, we assumed that $\theta_0 \sim (10, 10)$ so that the initial state distribution is around the unit scale and assumed that $\lambda_j \sim G(2, 1)$. Figure 6 shows the time series plot of these two series (straight red line represents household 1 and the dashed black line represents household 2) for 104 consecutive weeks.

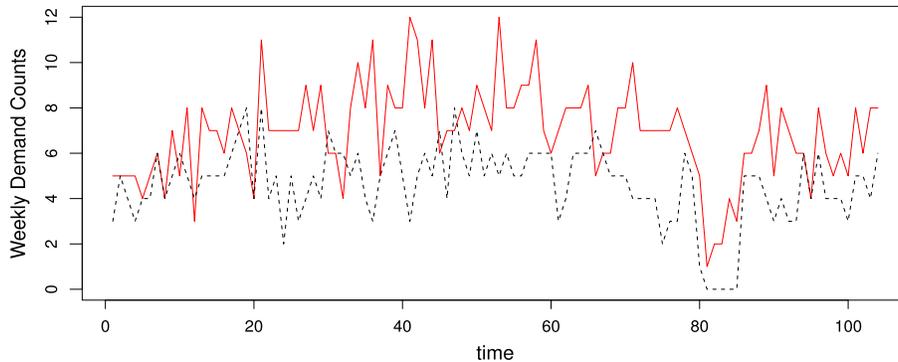


Figure 6: Time series plot of weekly demand for households 1 (top straight red line) and 2 (bottom dashed black line) for 104 weeks.

Figure 7 shows the mean posterior (filtered) estimates (red circles) and the 95% credibility intervals (straight lines) versus the actual data (black dots). We can observe

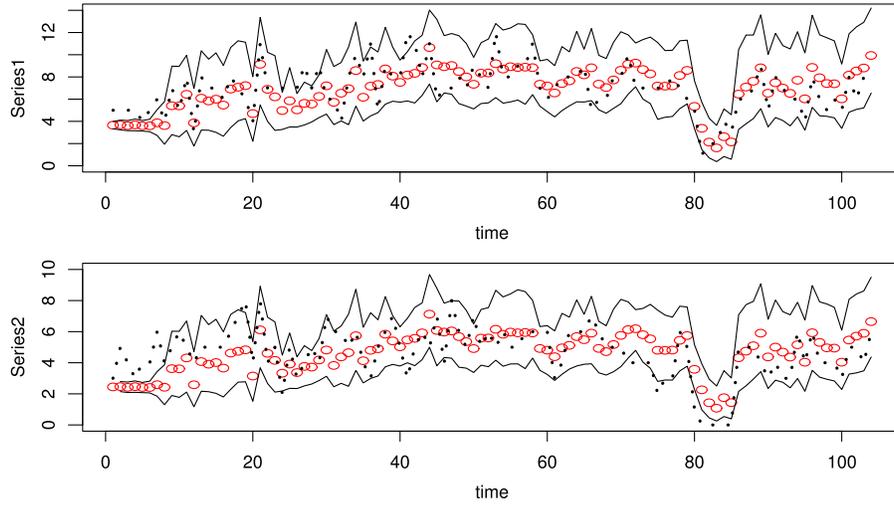


Figure 7: Mean posterior estimates (red circles) and the 95% credibility intervals (straight lines) versus the actual data (black dots) for the consumer demand data.

that in most cases the counts are within the credibility intervals except for the beginning first roughly ten time periods. This may be attributed to the fact that the counts for these two households were relatively lower and closer to each other initially, resulting with less global uncertainty in the counts and tighter intervals. However, visually the plots suggest that the model is able to account for sudden changes in the environment (for instance there is a sudden drop around weeks 80–85) while providing an overall reasonable fit for the counts of both households. Since the sample correlation between the two series was 0.41, suggesting a relatively low correlation, there were certain time periods when the intervals do not cover the actual data. For instance, the first 10 observations especially for series 2, look problematic and the model is slow to adapt to the sudden drop between weeks 80–85. However, approximately more than 90% of the real counts are within the credibility interval bounds of the filtered states. Even though we do not know the data generating process unlike the simulated examples, MAPE obtained for this example was 0.18 which is reasonably low.

The posterior distributions of γ as well as those of λ_1 and λ_2 are given in Figure 8. A higher value of λ indicates a higher order of spending habit for household 1 as opposed to household 2 given that both are exposed to the same economic environment. The mean estimates were 3.05 and 2.04, respectively for the two static parameters. We also note that the posterior correlation between λ_1 and λ_2 was estimated to be 0.21, as expected a positive correlation a posteriori. Furthermore, the posterior mean of γ was around 0.29. In our experience with both simulated and demand data, we observed that the posterior distribution of the static parameter γ did not vary significantly as we observe more data points (say beyond 20–30 observations as argued previously based on Figure 2). Therefore, a practical approach for cases where on-line learning and forecasting is of highest importance, would be to treat γ as fixed (either at the posterior

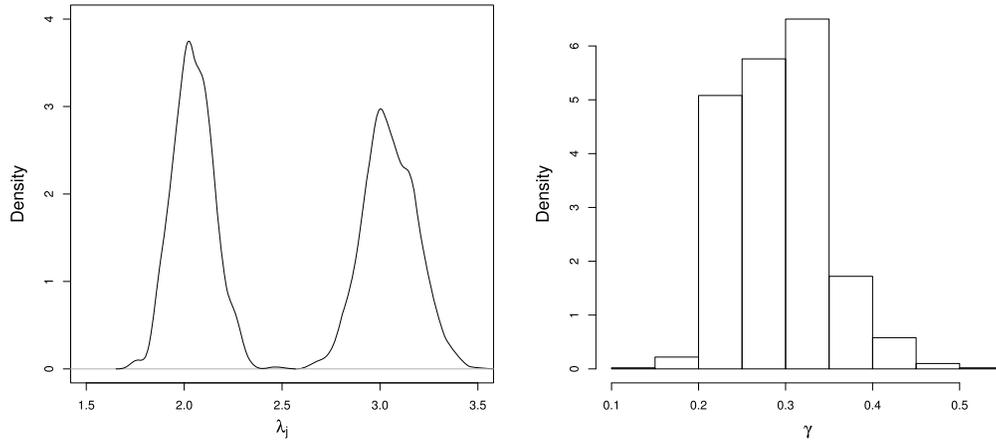


Figure 8: Posterior distributions of the static parameters, λ_1, λ_2 (left) and γ (right) for the customer demand data. λ_1 and λ_2 posterior estimates indicate distinct customer demand behavior for each household.

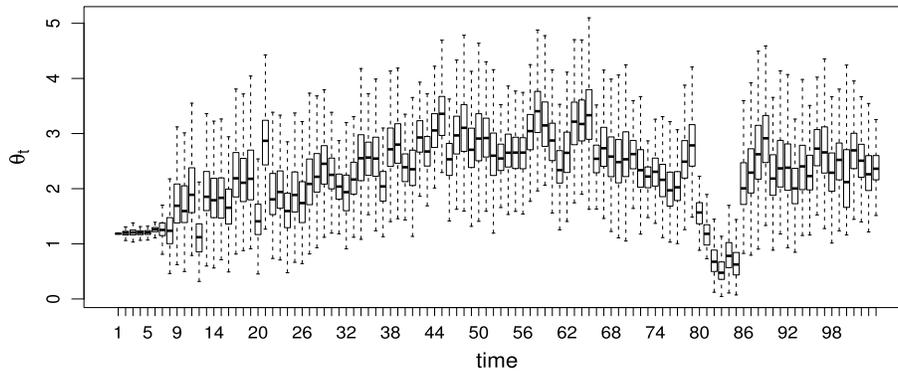


Figure 9: Boxplot of the dynamic state parameters, θ_t 's for the customer demand example, representing the random common economic environment that the two households are exposed to.

mean or the mode) which can significantly reduce the computational burden by making filtering very fast.

Figure 9 shows the boxplot of the posterior state parameters, in other words how the common environment that both households are exposed to changes over time. We can observe that the uncertainty about the environment is relatively lower at the beginning (in the first 1–5 time periods) with respect to the following time periods. This is the same observation we had drawn from the credibility intervals and could be due to the small difference between the counts. Also, the environment is said to be less favorable during roughly weeks 80–85 as there is a steep drop in the state estimates. We believe that

being able to model and predict household demand would be of interest to operations managers for long term as well as short term staffing purposes. For instance, related work in queuing systems require the modeling of the time varying arrival rates that are used as inputs of a stochastic optimization formulation to determine the optimal staffing levels (see Weinberg et al. (2007) and Aktekin and Soyer (2012) and the references therein for recent work using Bayesian methods for modeling Poisson arrivals in queuing models). In addition, the marketers may use these models for optimally timing the placements of advertisements and promotions. For instance, a steep drop in the state parameters (as in the weeks of 80–85 in our illustration) might lead to reductions in staffing for cutting operational costs (employees may be diverted to other tasks) or the company may decide to launch a more aggressive advertisement/promotion campaign to cope with undesirable market conditions.

5 Conclusion

In summary, we introduced a new class of dynamic multivariate Poisson models (which we call the MPSB model) that are assumed to be exposed to the same random common environment. We considered their Bayesian sequential inference using PL methods for fast online updating. One of the attractive features of the PL approach as opposed to MCMC counterparts, is how fast it generates particles sequentially in the face of new data, a feature not shared with MCMC methods where the whole chain needs to be restarted when new data is observed. The model allowed us to obtain analytic forms of the propagation density and as well as the predictive likelihood that are essential for the implementation of PL methods which is a property that not many state space models possess in the literature outside of Gaussian models. In addition, our model allowed us to obtain sequential updating of sufficient statistics in learning our static parameters that is another crucial and desirable feature of the PL method. Further, we showed how the proposed model leads to a new class of predictive likelihoods (marginals) for dynamic multivariate Poisson time series, which we refer to as the (dynamic) multivariate confluent hyper-geometric negative binomial distribution (MCHG-NB) and a new multivariate distribution which we call the dynamic multivariate negative binomial (DMNB) distribution. To show the implementation of our model, we considered various simulations and one actual data on weekly consumer demand for non-durable goods and discussed implications of learning both the dynamic state and static parameters.

To conclude, we believe that it is worth noting limitations of our model. The first one is the positive correlation requirement among series as induced by (12). As the series are assumed to be exposed to the same random common environment, our model requires them to be positively correlated. We investigated the implications of this requirement in the estimation paths of our static parameters in Figures 1 and 2 and the empirical example in Figure 7. Based on these plots, it is possible to infer that initially there maybe a few observations that do not follow this requirement where the static parameter estimation paths and the filtered means are not in line with their respective real values. However, if the data is overall positively correlated, our model converges to regions around the true values of the parameters (Figures 1 and 2) and the mean filtered estimates are within the 95% credibility intervals of the real counts (Figure 7) after

8–10 time periods. Another noteworthy limitation is the identifiability issue when the priors for the static parameters are uninformative. Even though, the model keeps the product of the Poisson mean, $\theta_t \lambda_j$, close to the observed counts, it takes a very long time for the learning algorithm to explore regions close to the real values of the static parameters. To mitigate this issue, we suggest to use a prior centered around unity for θ_0 and to use slightly tighter priors on λ_j 's as discussed in our numerical example. When dealing with real count data, we believe this approach to be reasonable as long as the posterior filtered estimates provide coverage for the true counts, as the true values of the static parameters will never be known.

In addition, we believe that the proposed class of models can be a fertile future area of research in developing models that can account for sparsity typically observed in multivariate count data. Our current model does not have a suitable mechanism for dealing with sparsity, however modifying the state equation to account for a transition equation that can account for sparsity may be possible and is currently being investigated by the authors. Another possible extension would be to introduce a similar approach for a general class of multivariate non-Gaussian state space models. This is also currently being considered by the authors with encouraging results.

Supplementary Material

Supplementary Appendices for “Sequential Bayesian Analysis of Multivariate Count Data” (DOI: [10.1214/17-BA1054SUPP](https://doi.org/10.1214/17-BA1054SUPP); .pdf).

References

- Abramowitz, M. and Stegun, I. (1968). *Handbook of Mathematical Functions Number 55*. Applied Mathematical Series. 392
- Aktekin, T. and Soyer, R. (2011). “Call center arrival modeling: A Bayesian state-space approach.” *Naval Research Logistics*, 58(1): 28–42. [MR2796402](https://doi.org/10.1002/nav.20436). doi: <https://doi.org/10.1002/nav.20436>. 386
- Aktekin, T. and Soyer, R. (2012). “Bayesian analysis of queues with impatient customers: Applications to call centers.” *Naval Research Logistics*, 59(2): 441–456. [MR2969100](https://doi.org/10.1002/nav.21499). doi: <https://doi.org/10.1002/nav.21499>. 405
- Aktekin, T., Soyer, R., and Xu, F. (2013). “Assessment of Mortgage Default Risk via Bayesian State Space Models.” *Annals of Applied Statistics*, 7(3): 1450–1473. 386
- Aktekin, T., Polson, N., and Soyer, R. (2017). “Supplementary Appendices for “Sequential Bayesian Analysis of Multivariate Count Data”.” *Bayesian Analysis*. doi: <https://doi.org/10.1214/17-BA1054SUPP>. 392
- Al-Osh, M. A. and Alzaid, A. A. (1987). “First-order integer valued autoregressive (INAR(1)) Process.” *Journal of Time Series Analysis*, 8(3): 261–275. 386
- Arbous, A. G. and Kerrich, J. (1951). “Accident statistics and the concept of accident proneness.” *Biometrics*, 7: 340–432. 386, 389

- Carter, C. K. and Kohn, R. (1994). “On Gibbs sampling for state space models.” *Biometrika*, 81(3): 541–553. [386](#)
- Carvalho, C., Johannes, M. S., Lopes, H. F., and Polson, N. (2010a). “Particle learning and smoothing.” *Statistical Science*, 25(1): 88–106. [MR2741816](#). doi: <https://doi.org/10.1214/10-STS325>. [385](#), [386](#), [390](#), [391](#)
- Carvalho, C. M., Lopes, H. F., Polson, N. G., and Taddy, M. A. (2010b). “Particle learning for general mixtures.” *Bayesian Analysis*, 5(4): 709–740. [391](#)
- Cox, D. R. (1981). “Statistical Analysis of Time Series: Some Recent Developments.” *Scandinavian Journal of Statistics*, 8: 93–115. [MR0623586](#). [386](#)
- Davis, R., Holan, S., Lund, R., and Ravishanker, N. (2015). *Handbook of Discrete-Valued Time Series*. Chapman and Hall/CRC. [386](#)
- Ding, M., He, L., Dunson, D., and Carin, L. (2012). “Nonparametric Bayesian segmentation of a multivariate inhomogeneous space-time Poisson process.” *Bayesian Analysis*, 7(4): 813. [MR3000015](#). doi: <https://doi.org/10.1214/12-BA727>. [386](#)
- Durbin, J. and Koopman, S. (2000). “Time Series Analysis of Non-Gaussian Observations based on state space models from both classical and Bayesian perspectives.” *Journal of the Royal Statistical Society, Series B*, 62(1): 3–56. [386](#)
- Freeland, R. K. and McCabe, B. P. M. (2004). “Analysis of Low Count Time Series Data by Poisson Autocorrelation.” *Journal of Time Series Analysis*, 25(5): 701–722. [386](#)
- Fruhwirth-Schnatter, S. (1994). “Data Augmentation and Dynamic Linear Models.” *Journal of Time Series Analysis*, 15(2): 183–202. [386](#)
- Fruhwirth-Schnatter, S. and Wagner, H. (2006). “Auxiliary mixture sampling for parameter-driven models of time series of counts with applications to state space modelling.” *Biometrika*, 93(4): 827–841. [386](#)
- Gamerman, D., Dos-Santos, T. R., and 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. [386](#)
- Gordon, N. J., Salmon, D. J., and Smith, A. F. (1993). “Novel approach to nonlinear/non-Gaussian Bayesian state estimation.” *IEE Proceedings F (Radar and Signal Processing)*, 140(2): 107–113. [385](#)
- Gordy, M. B. (1998). “Computationally convenient distributional assumptions for common-value auctions.” *Computational Economics*, 12(1): 61–78. [393](#)
- Gramacy, R. B. and Polson, N. G. (2011). “Particle learning of Gaussian process models for sequential design and optimization.” *Journal of Computational and Graphical Statistics*, 20(1). [391](#)
- Hankin, R. K. S. (2006). “Special functions in R: Introducing the gsl package.” *R News*, 6. [392](#)

- Harvey, A. C. and Fernandes, C. (1989). “Time Series Models for Count or Qualitative Observations.” *Journal of Business and Economic Statistics*, 7(4): 407–417. [386](#)
- Johannes, M. S., Polson, N. G., and Stroud, J. R. (2009). “Optimal filtering of jump diffusions: Extracting latent states from asset prices.” *Review of Financial Studies*, 22(7): 2759–2799. [402](#)
- Jorgensen, B., Lundbye-Christensen, S., Song, P., and Sun, L. (1999). “A State Space Model for Multivariate Longitudinal Count Data.” *Biometrika*, 86: 169–181. [386](#)
- Kim, B. (2013). “Essays in Dynamic Bayesian Models in Marketing.” Ph.D. thesis, The George Washington University. [394](#)
- Lindley, D. V. and Singpurwalla, N. D. (1986). “Multivariate Distributions for the Life Lengths of Components of a System Sharing a Common Environment.” *Journal of Applied Probability*, 23: 418–431. [386](#), [387](#)
- Lopes, H. F. and Polson, N. G. (2016). “Particle Learning for Fat-Tailed Distributions.” *Econometric Reviews*, 1–26. [391](#)
- Ord, K., Fernandes, C., and Harvey, A. C. (1993). “Time series models for multivariate series of count data.” In *Developments in Time Series Analysis: In honour of Maurice B. Priestley, T. Subba Rao (ed.)*, 295–309. [MR1292272](#). [386](#)
- Pedeli, X. and Karlis, D. (2011). “A bivariate INAR(1) model with application.” *Statistical Modelling*, 11: 325–349. [386](#)
- Pedeli, X. and Karlis, D. (2012). “On composite likelihood estimation of a multivariate INAR(1) model.” *Journal of Time Series Analysis*, 33: 903–915. [386](#)
- Polson, N. G., Stroud, J. R., and Müller, P. (2008). “Practical filtering with sequential parameter learning.” *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70(2): 413–428. [386](#)
- Ravishanker, N., Serhiyenko, V., and Willig, M. R. (2014). “Hierarchical dynamic models for multivariate times series of counts.” *Statistics and Its Interface*, 7: 559–570. [386](#)
- Ravishanker, N., Venkatesan, R., and Hu, S. (2015). “Dynamic models for time series of counts with a marketing application.” In *Handbook of discrete-valued time series*, R. A. Davis, S. H. Holan, R. Lund and N. Ravishanker (eds.), 423–445. [386](#)
- Serhiyenko, V., Ravishanker, N., and Venkatesan, R. (2015). “Approximate Bayesian estimation for multivariate count time series models.” In *Ordered Data Analysis, Modeling and Health Research Methods*, P. K. Choudhary et al. (eds.), 155–167. [MR3478370](#). doi: https://doi.org/10.1007/978-3-319-25433-3_10. [386](#)
- Smith, R. and Miller, J. E. (1986). “A Non-Gaussian State Space Model and Application to Prediction of Records.” *Journal of the Royal Statistical Society, Series B*, 48(1): 79–88. [386](#), [387](#)
- 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. [390](#), [400](#)

- Taddy, M. A. (2010). “Autoregressive mixture models for dynamic spatial Poisson processes: Application to tracking intensity of violent crime.” *Journal of the American Statistical Association*, 105(492): 1403–1417. 386, 391
- Taddy, M. A. and Kottas, A. (2012). “Mixture modeling for marked Poisson processes.” *Bayesian Analysis*, 7(2): 335–362. 386
- Weinberg, J., Brown, L. D., and Stroud, J. R. (2007). “Bayesian forecasting of an inhomogeneous Poisson process with applications to call center data.” *Journal of the American Statistical Association*, 102(480): 1185–1198. MR2412542. doi: <https://doi.org/10.1198/016214506000001455>. 405
- Zeger, S. L. (1988). “A regression model for time series of counts.” *Biometrika*, 75(4): 621–629. 386

Acknowledgments

The authors would like to thank the anonymous referee, the associate editor, and the editor whose comments helped us significantly improve the manuscript.