State Space Modeling of Autocorrelated Multivariate Poisson Counts

Chen Zhang†, Nan Chen‡, and Zhiguo Li‡

†Department of Industrial and Systems Engineering, National University of Singapore
‡IBM T.J. Watson Research Center

Abstract

Though many applications involve autocorrelated multivariate counts, there is a scarcity of research on statistical modeling of them. To fill this research gap, this paper proposes a state space model to describe autocorrelated multivariate counts. The model builds upon the multivariate log-normal mixture Poisson distribution and allows for serial correlations by considering the Poisson mean vector as a latent process driven by a nonlinear autoregressive model. In this way the model allows for flexible cross-correlation and autocorrelation structures of count data, and can capture the overdispersion as well. The Monte Carlo Expectation Maximization algorithm together with particle filtering & smoothing methods provide satisfactory estimators for the model parameters and the latent process variables. Numerical studies show that, compared with other state-of-the-art models, the proposed model has superiority and more generality with respect to well describing count data generated from different mechanisms of the process of counts. Finally we use this model to analyze counts of different types of damages collected from a power utility system as a case study. Supplementary materials are available for this article. Go to the publisher’s online edition of IIE Transactions for additional tables and figures.

Keywords: Multivariate time series; Multivariate Poisson distribution; Overdispersion; State space model; Particle filtering & smoothing; Monte Carlo expectation maximization

1 Introduction

Count data arise in many areas, such as ecology, epidemiology, economics, manufacturing, etc, where usually multiple counts are observed together. because they usually exhibit certain correlations with

*Corresponding author: ise.cn@nus.edu.sg
each other, i.e., cross-correlations, we call them multivariate counts and want to analyze them together. For examples, in ecology, counts of different species interact due to the common environmental features (Billheimer et al., 2001). In epidemiology, counts of patients with related diseases may be correlated with each other (Paul et al., 2008). In marketing, sales volumes of different products or brands may influence each other (Chen et al., 2015). In quality control, counts of different types of defects may be caused or influenced by some common factors.

In addition to cross-correlation, in many applications multivariate counts evolve over time and have serial correlations with their previous observations, i.e., autocorrelations. For example, the count of infected persons with an infectious disease in this month is influenced by that in the last month. The weekly sales volume of a product often fluctuates with seasonable or economic variations. The number of defects in neighbour samples in manufacturing may be driven by certain common inertial elements when the sampling interval is small. Usually, the serial dependence can be either positive or negative.

On top of these correlation features, another common feature of autocorrelated multivariate counts is overdispersion, which means, with respect to a model, the variance of the count data is greater than the expectation. Overdispersion is often an affiliation property of cross-correlations and autocorrelations, and called much attention to in Poisson regression models and time series models. Usually overdispersion is caused by some unobserved heterogeneities across the count data. Suppose that the multivariate counts depend on an unobserved or omitted covariate \( z_t \). Then the change of \( z_t \) over time will introduce additional variance into the count data. A detailed discussion of these can be found in Cox and Isham (1980).

Though autocorrelated multivariate counts are quite common in our daily lives, so far to our best knowledge, a general model to describe them is yet to be addressed. A reasonable model should not only be able to describe cross-correlation and autocorrelation structures of count data flexibly, but also accommodate overdispersion. Although statistical models of univariate time series of count data are thoroughly explored in previous studies (see Davis et al. (1999) and the references therein for more background knowledge), the extensions to multivariate cases are underdeveloped. A brief literature review of these extensions is discussed below, with their model properties summarized in Table 1.

### 1.1 Multivariate Poisson Distribution

As we know, the Poisson distribution or its variants are often used to model count data. So far there are three main types of methods to construct multivariate Poisson distributions for multivariate counts \( Y = [Y_1, \cdots, Y_d] \) with dimension \( d \). We will discuss them in detail as below. The first directly extends
the bivariate Poisson distribution (Holgate, 1964) to $d$ dimensions by retaining the expression of each dimension as a sum of two independent variables, i.e.,

$$Y_i = Z_i + Z_0, \quad i = 1, \ldots, d.$$  \hspace{1cm} (1)

where $Z_i$ and $Z_0$ follow Poisson distributions with rate $\lambda_i$ and $\lambda_0$, separately. Given the parameters $\Lambda = \{\lambda_1, \ldots, \lambda_d, \lambda_0\}$, the joint density of the multivariate Poisson distribution is defined as

$$p(Y|\Lambda) = \prod_{i=1}^{d} p(Y_i|\lambda_i, \lambda_0),$$

whose marginal distribution of every dimension is the Poisson distribution. Clearly, it is $Z_0$ that introduces the same cross-correlation between different dimensions. Later extensions to allow different cross-correlations are also discussed in Kocherlakota and Kocherlakota (1992), etc. The second is to construct the multivariate Poisson distribution using copula (Song, 2000). Copula is a general way to introduce dependence between variables when their marginal distributions are given. Its idea is that a $d$-dimensional distribution can be written in terms of $d$ marginal distributions and a copula which describes the dependence structure of these dimensions. However, though copula modeling provides useful tools for analyzing the cross-correlations between multiple variables and has been vastly used for continuous distributed data, its primary difficulty in the discrete case is the lack of uniqueness of Sklar’s representation and the unidentifiability of the copula. This difficulty indicates that many of the convenient properties of copula cannot carry over from the continuous case to the discrete case. Therefore modeling and interpreting dependence for count data through copulas is still underdeveloped and subjects to caution (See Genest and Nešlehová (2007) for a comprehensive discussion).

Furthermore, unfortunately, most models in the above two categories can only support limited positive cross-correlations of multivariate counts. Furthermore, they have limited flexibilities in accounting for overdispersion.

A more flexible method is based on the mixture model by placing a distribution on the mean vector $\lambda = [\lambda_1, \ldots, \lambda_d]$ of $d$-dimensional Poisson counts. In this way the method is able to allow cross-correlations of either sign as well as overdispersion in a large range. This method can be further divided into two groups. The first considers a finite mixture, implying that the mean vector is chosen from finite $K$ components (Karlis and Meligkotsidou, 2007) with the corresponding probabilities $\pi_k, k = 1, \ldots, K$. Every component is a $d$-dimensional Poisson distribution defined in (1) with the parameters $\Lambda_k = \{\lambda_{k1}, \ldots, \lambda_{kd}, \lambda_{k0}\}$. Then the joint probability mass function (pmf) of $Y$ is given by

$$p(Y|\Lambda_{1:K}, \pi_{1:K}) = \sum_{k=1}^{K} \pi_k p(Y|\Lambda_k).$$  \hspace{1cm} (2)
However the assessment of the unknown \( K \) often requires much work. The second imposes a continuous distribution \( g(\lambda|\Theta) \) on the mean vector \( \lambda \). Then the unconditional multivariate Poisson distribution is a marginalization, integrating out the mean vector distribution as

\[
p(\mathbf{Y}|\Theta) = \int_{\mathbb{R}^+_d} \prod_{i=1}^{d} p(Y_i|\lambda_i) g(\lambda|\Theta) d\lambda, \tag{3}
\]

where \( p(Y_i|\lambda_i) \) is the Poisson pmf with rate \( \lambda_i \) for \( i = 1, \cdots, d \). This idea is generally adopted by many models with different forms of \( g(\lambda|\Theta) \), such as the gamma distribution (Arbous and Kerrich, 1951; Nelson, 1985), the normal distribution (Steyn, 1976), the log-normal distribution (Aitchison and Ho, 1989), and the beta distribution (Sarabia and Gómez-Déniz, 2011). Among them, the one using the log-normal mixture is the most powerful one. It on one hand brings the rich vein of cross-correlation structures of the multivariate normal distribution into the multivariate Poisson distribution, and on the other ensures that the Poisson mean vector is always positive according to the logarithmic transformation.

### 1.2 Multivariate Time Series of Counts

Though many multivariate Poisson distributions have been proposed in the literature as discussed above, extensions accounting for autocorrelation are still in infancy. One notable study is the multivariate autoregressive conditional Poisson model (MACP) (Heinen and Rengifo, 2007). This model extends the univariate integer-valued autoregressive conditional heteroskedasticity time series model (INGARCH) (Ferland et al., 2006) to multivariate cases. Specifically, for a \( d \)-dimensional count \( \mathbf{Y}_t = [Y_{t1}, \cdots, Y_{td}] \) at time \( t \) with the mean vector \( \mathbf{\lambda}_t = [\lambda_{t1}, \cdots, \lambda_{td}] \), MACP assumes that \( \mathbf{\lambda}_t \) follows a vector autoregressive moving average (VARMA)-type model with order \( p \) and \( q \) as

\[
\mathbf{\lambda}_t = \mathbf{\lambda}_0 + \sum_{j=1}^{p} \mathbf{A}_j \mathbf{Y}_{t-j} + \sum_{j=1}^{q} \mathbf{B}_j \mathbf{\lambda}_{t-j}. \tag{4}
\]

Then the joint pmf of \( \mathbf{Y}_t \) given \( \mathbf{\lambda}_t \) is \( p(\mathbf{Y}_t|\mathbf{\lambda}_t) = \prod_{i=1}^{d} p(Y_{ti}|\lambda_{ti}) \) where \( p(Y_{ti}|\lambda_{ti}) \) is the Poisson pmf with rate \( \lambda_{ti} \). To further allow for overdispersion, MACP suggests replacing the Poisson distribution \( p(Y_{ti}|\lambda_{ti}) \) with the double-Poisson distribution \( p(Y_{ti}|\lambda_{ti}, \phi) \) where \( \phi \) is the common overdispersion parameter. In addition, to conquer the limitation that INGARCH can only support positive cross-correlation structures, MACP further imposes a multivariate normal copula on \( \mathbf{Y}_t \) to allow for negative ones. Then the joint cumulative distribution function (cdf) of \( \mathbf{Y}_t \) is defined as a copula function \( C \) of the
cdfs of \( d \) marginal double Poisson distributions \( F_i(Y_i), i = 1, \cdots, d \), i.e.,

\[
F(Y_t) = C(F_1(Y_1), \cdots, F_d(Y_d)).
\]  

However, MACP still suffers from the inherited limitation of INGARCH that it can only describe positive autocorrelated count data (Jung et al., 2006). In particular, as defined in Heinen and Rengifo (2007), MACP(1,1) is stationary only if the eigenvalues of \( I - A_1 - B_1 \) lie within the unit circle. Then according to Proposition 2.1 and Proposition 2.2 of (Heinen and Rengifo, 2007), the autocovariance matrix of MACP(1,1) can only take positive values. Furthermore, as mentioned early, since the copula for discrete data is no longer identifiable, further theoretical properties and assumptions of MACP are still to be carefully discussed. In particular, inference (and particularly rank-based inference) for the copula parameters is fraught with difficulties. Another type of notable work is the extension of univariate integer-value autoregressive (INAR) models to multivariate (MINAR) cases by generalizing the binomial thinning operator in the INAR models to a thinning matrix (Pedeli and Karlis, 2013a,b), i.e.,

\[
Y_t = A \circ Y_{t-1} + R_t.
\]

The \( d \times d \) matrix \( A = \{a_{ij}, i, j = 1, \cdots, d\} \) acts as the usual matrix multiplication but keep the properties of the binomial thinning operator (Weiß, 2008). Specifically, the operators \( a_{ij}\circ \) are mutually independent. Each operator is defined as \( a_{ij} \circ Y_j = \sum_{k=1}^{Y_j} X_k \), where \( \{X_k\}_{k=1}^{Y_j} \) is a sequence of i.i.d Bernoulli random variables such that \( p(X_k = 1) = a_{ij} = 1 - p(X_k = 0) \) and \( a_{ij} \in [0,1] \). Currently most work in this field focus on first order MINAR models, denoted as MINAR(1), for bivariate counts with \( R_t \) defined as a bivariate Poisson distribution in (1). This is because, as analyzed earlier, defining \( R_t \) for higher dimensions with flexible cross-correlation structures is not easy. Currently the only MINAR(1) model considering more than two dimensions is Pedeli and Karlis (2013a), which considers \( R_t \) with flexible cross-correlations between different dimensions. However this model assumes \( A \) only has diagonal components. Another limitation of the MINAR models is that they can only support positive cross-correlations and autocorrelations of count data, and do not allow for large overdispersion (Pedeli and Karlis, 2013b). This is because that the autocovariance matrix of MINAR models can be written as

\[
\gamma(h) = A \gamma(h - 1) = A^h \gamma(0), \quad h \geq 1
\]  

5
where $\gamma(0)$ is the cross-correlation matrix. Since both $A$ and $\gamma(0)$ can only have positive values according to Equation (6) of Pedeli and Karlis (2013b), (7) can only achieve positive autocorrelations as well.

In time series analysis, the two models mentioned above belong to the class of observation-driven models. As mentioned in Davis et al. (1999), while the observation-driven model is advantageous for easily calculating the forecasting density function, it is not good at characterizing the evolutionary properties of time series. An alternative is the parameter-driven model, which assumes the serial correlation is induced by a latent variable. Then the evolutionary properties can be typically inherited by those assumed for this latent variable. Usually we call this latent variable as the hidden state and can resort to state space approaches for analysis (Durbin and Koopman, 2000).

In the case of univariate count series, state space approaches have been widely used in Zeger (1988), Harvey and Fernandes (1989), and Chan and Ledolter (1995). For more discussions about univariate count series modeling, please refer to Fokianos (2012). For multivariate cases, Jørgensen et al. (1999) and Jung et al. (2011) propose two factor models to describe autocorrelated multivariate Poisson counts. Both models assume that the count data of each dimension follow a Poisson distribution whose mean value is driven by some common latent factors following gamma Markov processes (Jørgensen et al., 1999) or Gaussian autoregressive processes (Jung et al., 2011). The former allows for mere positive autocorrelations, while the later allows for autocorrelations of either sign. Because these factor models explain the interactions of different counts by regression models, they avoid directly analyzing their cross-correlation structure. However, the choice of latent factors usually requires domain-specific knowledge, and the criteria about how many factors are needed are not always clear. As a result, it is difficult to generalize these models to fields where no factor or ambiguous factors exist.

Motivated by the wide application of autocorrelated multivariate counts and the infancy of reasonable models to describe them, this paper further explores this field with a twofold contribution. First, this paper proposes an easy-to-interpret state space model to describe autocorrelated multivariate counts. This model allows for flexible cross-correlation and autocorrelation structures of count data, and can handle a large range of overdispersion. Specifically, this model builds upon the log-normal mixture Poisson distribution of Aitchison and Ho (1989), and allows for serial dependence by considering the Poisson mean vector as a latent variable evolving according to a state space model. In this way the model can describe the cross-correlations and autocorrelations of count data flexibly. By integrating out the latent variable distribution, this model can generate itself an over-dispersed unconditional distribution, and hence can capture the overdispersion of count data. Second, this paper presents an efficient estimation
Table 1: Summary of some state-of-the-art Poisson models of multivariate counts. “+”: allow for positive correlations; “-”: allow for negative correlations; ×: cannot describe the feature; √: can describe the feature.

<table>
<thead>
<tr>
<th>Literature</th>
<th>Method</th>
<th>Cross-correlation</th>
<th>Autocorrelation</th>
<th>Over-dispersion</th>
</tr>
</thead>
<tbody>
<tr>
<td>Karlis (2003)</td>
<td>Sum of Poisson</td>
<td>+</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>Karlis and Meligkotsidou (2005)</td>
<td>Sum of Poisson</td>
<td>+</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>Song (2000)</td>
<td>Normal copulas</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>Karlis and Meligkotsidou (2007)</td>
<td>Finite mixture</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>Arbous and Kerrich (1951)</td>
<td>Poisson-Gamma mixture</td>
<td>+</td>
<td>×</td>
<td>√</td>
</tr>
<tr>
<td>Steyn (1976)</td>
<td>Poisson-normal mixture</td>
<td>+</td>
<td>×</td>
<td>√</td>
</tr>
<tr>
<td>Aitchison and Ho (1989)</td>
<td>Poisson-lognormal mixture</td>
<td>+</td>
<td>×</td>
<td>√</td>
</tr>
<tr>
<td>Sarabia and Gómez-Déniz (2011)</td>
<td>Poisson-Beta mixture</td>
<td>+</td>
<td>×</td>
<td>√</td>
</tr>
<tr>
<td>Heinen and Rengifo (2007)</td>
<td>Multivariate INGARCH, copulas</td>
<td>×</td>
<td>+</td>
<td>√</td>
</tr>
<tr>
<td>Latour (1997)</td>
<td>Multivariate GINAR</td>
<td>+</td>
<td>+</td>
<td>√</td>
</tr>
<tr>
<td>Pedeli and Karlis (2013b)</td>
<td>Multivariate GINAR</td>
<td>+</td>
<td>+</td>
<td>√</td>
</tr>
</tbody>
</table>

The algorithm for the proposed model. The challenge is that the marginal unconditional likelihood function of the model has no close form, so the model needs numerical integration methods for estimation. Here the Monte Carlo Expectation Maximization (MCEM) algorithm is used, where the MC part is done by particle filtering & smoothing methods. Numerical studies show that the MCEM algorithm presents accurate estimation results for the model parameters. Particle filtering also provides an asymptotically unbiased estimator for the latent variable in a sequential way with small computational complexity.

The remainder of the paper is organized as follows. Section 2 introduces our proposed state space multivariate Poisson model in detail. Section 3 discusses the model estimation procedure. Section 4 stresses the advantages of the proposed model by comparing it with some other state-of-the-art Poisson models of multivariate counts, and demonstrates the proposal using a real-data example from the power utility industry. Finally Section 5 concludes this paper with remarks. Some technical details are provided in the Appendix.

2 A State Space Model for Autocorrelated Multivariate Poisson Counts

In this section, we introduce a state space model to describe autocorrelated multivariate counts, which can not only flexibly describe the cross-correlation and autocorrelation structure of count data, but also take overdispersion into consideration.

Consider $d$-dimensional count variables $Y_t = [Y_{t1}, \ldots, Y_{td}]$ for time $t = 1, \ldots, T$. For each dimension at time $t$, conditional on its mean $\lambda_{ti}$, we assume $Y_{ti}$ follows an independent Poisson distri-
bution with the pmf as

\[ p(Y_{ti} | \lambda_{ti}) = \frac{\exp(-\lambda_{ti}) \lambda_{ti}^{Y_{ti}}}{Y_{ti}!}, \quad i = 1, \ldots, d, \ t = 1, \ldots, T. \]  

(8)

Then the joint conditional distribution of \( Y_t \) is

\[ p(Y_t | \lambda_t) = \prod_{i=1}^{d} p(Y_{ti} | \lambda_{ti}), \quad t = 1, \ldots, T. \]  

(9)

where \( \lambda_t = [\lambda_{t1}, \ldots, \lambda_{td}] \). We assume \( X_t = \log(\lambda_t) = [\log(\lambda_{t1}), \ldots, \log(\lambda_{td})] \) as a latent random variable following multivariate normal distribution. It is \( X_t \) that introduces both cross-correlations and autocorrelations into \( Y_t \). Specifically, we consider \( X_t \) evolves according to a state space model as

\[ p_{\Theta}(X_t | X_{t-1}) : X_t - \mu = \Phi \cdot (X_{t-1} - \mu) + \epsilon_t, \]  

(10)

where \( \epsilon_t \) is the white noise following a \( d \)-dimensional multivariate normal distribution with mean vector \( 0 \) and covariance matrix \( \Sigma \), i.e., \( N(0, \Sigma) \). So far we have introduced all the model parameters \( \Theta = \{\mu, \Phi, \Sigma\} \).

As long as \( \Phi \) satisfies

\[ \det(\mathbf{I} - z \Phi) \neq 0, \quad \text{for all } |z| \leq 1, \ z \in \mathbb{C}, \]

\( X_t \) is stationary, and the marginal distribution of \( X_t \) is multivariate normal with mean vector \( \mu \) and covariance matrix \( \Gamma \), where \( \Gamma \) is the solution of \( \Gamma = \Phi \Gamma \Phi' + \Sigma \) according to the Yule-Walker relationship.

Marginalizing out \( X_t \), the unconditional distribution of \( Y_t \) can be expressed as

\[ p_{\Theta}(Y_t) = \int_{\mathbb{R}^d_+} \prod_{i=1}^{d} \frac{\exp(-\exp(X_{ti})) \exp(X_{ti})^{Y_{ti}}}{Y_{ti}!} p_{\Theta}(X_t) dX_t, \]  

(11)

where \( p_{\Theta}(X_t) \) is the pdf of \( N(\mu, \Gamma) \).

Figure 1 illustrates the evolution process of \( \{X_t, Y_t\} \), which can be viewed as a nonlinear state space model. This hierarchical model allows for flexible cross-correlations and autocorrelations of multivariate counts. It ties multiple counts together but allows for individual stochastic components through the term \( \epsilon_t \). Higher-order autocorrelations of \( X_t \) can also be accommodated in the model. Hereafter we call the proposed model as the state space multivariate Poisson model (SSMP). It is to be noted that when
\[ d = 1, \text{SSMP degenerates to the univariate parameter-driven count series models of Chan and Ledolter (1995); Kuk and Cheng (1997); Jung and Liesenfeld (2001). Though the unconditional distribution of } Y_t \text{ in (11) has no close form, its moment properties can be obtained through conditional expectations and the properties of Poisson and normal distributions as shown in the following propositions. Detailed derivations are given in the Appendix.}

**Proposition 1 (Mean of SSMP)** Provided that \( X_t \) is stationary following (10), \( Y_t \) is stationary with its unconditional mean as

\[
E(Y_{ti}) \equiv \alpha_i = \exp(\mu_i + \frac{1}{2} \Gamma_{ii})
\]

for \( i = 1, \ldots, d \), where \( \Gamma_{ii} \) is the \( i \)th diagonal component of \( \Gamma \).

This proposition shows that, as long as the latent variable is stationary, the process of counts is stationary whose mean vector is jointly decided by \( \Theta = \{ \mu, \Phi, \Sigma \} \).

**Proposition 2 (Variance of SSMP)** Provided that \( X_t \) is stationary following (10), the unconditional covariance matrix of \( Y_t \) can be expressed as

\[
\text{Var}(Y_{ti}) = \alpha_i [1 + \alpha_i (\exp(\Gamma_{ii}) - 1)], \quad \text{for} \quad i = 1, \ldots, d,
\]

\[
\text{Cov}(Y_{ti}, Y_{tj}) = \alpha_i \alpha_j (\exp(\Gamma_{ij}) - 1), \quad \text{for} \quad i \neq j, i, j = 1, \ldots, d.
\]

As a result,

\[
\text{Corr}(Y_{ti}, Y_{tj}) = \frac{\exp(\Gamma_{ij}) - 1}{\sqrt{\alpha_i^{-1} + \exp(\Gamma_{ii}) - 1} \sqrt{\alpha_j^{-1} + \exp(\Gamma_{jj}) - 1}},
\]

where \( \Gamma_{ij} \) is the \( (i, j) \) component of \( \Gamma \).

(13) shows that the unconditional variance of \( Y_{ti} \) for every dimension of SSMP exhibits overdispersion. The amount of overdispersion increases with \( \Gamma_{ii} \) and \( \alpha_i \). Only if \( \Gamma_{ii} = 0 \) indicating that \( X_{ti} \) degenerates
to a fixed value, the overdispersion disappears, and consequently $Y_t$ follows the traditional Poisson distribution with no serial correlations. The cross-correlation structure of $Y_t$ depends on $\Gamma_{ij}$ as well, and can take either positive or negative values. However, as mentioned in Aitchison and Ho (1989), because

$$|\text{Corr}(Y_{ti}, Y_{tj})| < |\text{Corr}(X_{ti}, X_{tj})|,$$

(16)

the range of possible cross-correlations of $Y_t$ is not as wide as those of $X_t$. However their gap becomes smaller when $\alpha_i$ and $\alpha_j$ become larger. Figure 2a provides the descriptive power of SSMP for bivariate counts with respect to attainable cross-correlations and overdispersion. Here we only consider a special case with $\mu_1 = \mu_2 = \mu$, $\Gamma_{11} = \Gamma_{22} = \gamma_0$, and hence $\alpha_1 = \alpha_2 = \alpha$ according to (12). We fix $\alpha$ but vary $\mu$, $\gamma_0$, $\Gamma_{12}$ and $\Gamma_{21}$ to get different realizations of the cross-correlation structure with the same mean vector. The enclosed areas are the attainable regions for SSMP with $\alpha = 1$, 3 and 10, respectively. We could see that SSMP has the flexibility to accommodate different overdispersion magnitudes, which can be controlled by choosing appropriate $\gamma_0$ while maintaining the same $\alpha$. On the other hand, for a small $\alpha$, the model can hardly describe circumstances with moderate or strong negative cross-correlations. As $\alpha$ increases, the model is able to describe larger negative cross-correlations.

**Proposition 3 (Autocorrelation of SSMP)** Provided that $X_t$ is stationary, the unconditional autocorrelations of $Y_t$ can be expressed as

$$\text{Corr}(Y_{ti}, Y_{(t-\tau)j}) = \frac{E\left[\exp(X_{ti} + X_{(t-\tau)j})\right] - \alpha_i \alpha_j}{\alpha_i \alpha_j \sqrt{\left[\alpha_i^{-1} + \exp(\Gamma_{ii}) - 1\right] \left[\alpha_j^{-1} + \exp(\Gamma_{jj}) - 1\right]}}$$

(17)

for $\tau = 0, 1, \ldots, i, j = 1, \ldots, d$.

The unconditional autocorrelations of $Y_t$ depends on $E\left[\exp(X_{ti} + X_{(t-\tau)j})\right]$, which can be calculated from the characteristic function of $X_t$ (see Appendix for details). The autocorrelations can be either positive or negative, but still we have

$$|\text{Corr}(Y_{ti}, Y_{(t-\tau)j})| < |\text{Corr}(X_{ti}, X_{(t-\tau)j})|.$$

(18)

Similarly in Figure 2b we plot the descriptive power of SSMP with respect to attainable cross-correlations and first-order autocorrelations with the same settings as Figure 2a. We see that SSMP can describe processes of counts with certain negative autocorrelations, but not as flexible as positive ones. With the
increase in $\alpha$, the attainable regions increase for both kinds of correlations. In summary, Proposition 2 and 3 indicate that SSMP is more suitable to describe count data with moderate or big mean values.

![Figure 2: The regions of cross-correlation, overdispersion and autocorrelation attainable for SSMP with fixed $\alpha$ and tunable $\{\mu, \gamma_0, \Gamma_{12}, \Gamma_{21}\}$ for bivariate counts.]

### 3 Parameter Estimation and Inference

#### 3.1 Prediction & Inference

Now we study how to make inference and predictions based on SSMP. Here we temporarily assume that the model parameters $\Theta$ are known, and will discuss how to estimate them in Section 3.2.

Because in SSMP, $\{Y_{1:T}\}$ are observable while $\{X_{1:T}\}$ are latent, most of the inference problems focus on estimating the latent process $p_\Theta(X_{1:T}|Y_{1:T})$ given the total $T$ observations $\{Y_{1:T}\}$. More specifically, we first focus on predicting $X_{t+1}$ based on the previous and current observations $\{Y_{1:t}\}$, i.e., $p_\Theta(X_{t+1}|Y_{1:t})$. Secondly, we study the inference of $X_t$ based on the total observations $\{Y_{1:T}\}$, i.e., $p_\Theta(X_t|Y_{1:T})$ for $t = 1, \ldots, T$. The challenge involved is because the posterior distribution $p_\Theta(X_t|Y_{1:t})$ has no close form for arbitrary $\Theta$, we need to resort to numerical methods, such as numerical integration or Markov chain Monte Carlo (MCMC). In addition, we often need to update the predictions or estimations upon arrivals of new observations, e.g., to obtain $p_\Theta(X_{t+2}|Y_{1:(t+1)})$ from $p_\Theta(X_{t+1}|Y_{1:t})$. As a result, it is desirable to have computationally efficient algorithms to make inference and predictions sequentially.

Considering the performance and computational efficiency, in state space model analysis, particle filtering (PF, also called Sequential Monte Carlo) and smoothing (PS) methods are particularly useful. PF is designed to approximate $p_\Theta(X_t|Y_{1:t})$ in a sequential way with acceptably small computational complexity (Liu and Chen, 1998; Hürzeler and Künsch, 1998; Doucet et al., 2000). Its basic idea is to
compute \( p_{\Theta}(X_t|Y_{1:t}) \) by importance sampling, i.e., approximating \( p_{\Theta}(X_t|Y_{1:t}) \) by a set of samples, called particles with their associated weights. The weight assigned to each particle is proportional to its probability of being sampled from the posterior distribution. When new data are observed, new particles and their associated weights can be efficiently obtained with affordable computational burden.

In particular, for the prediction task, we assume at time \( t \), the posterior density is approximated by weighted Dirac delta functions as

\[
p_{\Theta}(X_t|Y_{1:t}) \approx \sum_{i=1}^{N_p} W_t^i \cdot \delta(X_t - x_t^i).
\]  

(19)

Here in (19), \( \delta \) is the Dirac delta function; \( N_p \) is the number of particles; The normalized weights \( W_t^i \) satisfy \( \sum_{i=1}^{N_p} W_t^i = 1 \). To obtain the prediction distribution \( p_{\Theta}(X_{t+1}|Y_{1:t}) \), we can first generate samples of \( X_{t+1} \) from \( p_{\Theta}(X_{t+1}|Y_{1:t}) \). For this purpose, each particle \( x_{t+1}^i \) is propagated following the state equation in (10) with a random noise \( \epsilon_{t+1}^i \) drawn from the state noise distribution \( \mathcal{N}(0, \Sigma) \), i.e.,

\[
x_{t+1}^i = \mu + \Phi(x_t^i - \mu) + \epsilon_{t+1}^i.
\]

Then we have

\[
p_{\Theta}(X_{t+1}|Y_{1:t}) \approx \sum_{i=1}^{N_p} W_t^i \cdot \delta(X_{t+1} - x_{t+1}^i).
\]  

(20)

When the new observation \( Y_{t+1} \) comes, we can update the conditional distribution of \( X_{t+1} \) and approximate \( p_{\Theta}(X_{t+1}|Y_{1:t+1}) \). In fact, the updated distribution takes the same form as (20) and uses the same set of particles. It only needs to update every particle’s weight based on the likelihood \( p_{\Theta}(Y_{t+1}|x_{t+1}^i) \) according to the Bayes rule, i.e.,

\[
p_{\Theta}(X_{t+1}|Y_{1:(t+1)}) \approx \sum_{i=1}^{N_p} W_{t+1}^i \cdot \delta(X_{t+1} - x_{t+1}^i), \quad \text{where } W_{t+1}^i \propto W_t^i \cdot p(Y_{t+1}|x_{t+1}^i).
\]

The convergence of the approximated distribution by PF to the true \( p_{\Theta}(X_t|Y_{1:t}) \) is guaranteed by the central limit theorem (Liu, 2008) which ensures its estimation accuracy. Due to its computational efficiency and sequential nature, PF has been widely applied for nonlinear state space model inference and latent process tracking. See Doucet et al. (2001) and the references therein for a detailed introduction.

One problem of PF is that the distribution of the particles’ weights becomes more and more skewed as \( t \) increases. Hence, after some iterations, only very few particles have non-zero weights. This phenomenon is called degeneracy. We can evaluate it by the so-called effective sample size (ESS) (Liu, 2008), which is given by \( \text{ESS} = \left( \sum_{i=1}^{N_p} (W_t^i)^2 \right)^{-1} \). An intuitive solution for degeneracy is to multiply
the particles with higher normalized weights, and discard the particles with lower weights. This can be done by adding a resampling step. Specifically, if ESS is smaller than a pre-specified threshold α, we resample from the set \{((W^i_t, x^i_t), i = 1, \ldots, N_p)\} with the probabilities \(p(x^i_t = x^i_t) = W^i_t, i = 1, \ldots, N_p\) with replacement \(N_p\) times, to get a new set \{((\frac{1}{N_p}, x^j_t), j = 1, \ldots, N_p)\}. In this way the skewness of the weights’ distribution can be reduced. The detailed particle filtering procedure involving the resampling step is summarized in Algorithm 1.

**Algorithm 1 Particle filtering (PF)**

At time \(t = 1\)
1: Initialization: sample \(x^i_1 \sim p_0(X|1)\) for \(i = 1, \ldots, N_p\).
2: Compute the weights \(w^i_1 = p(Y_1|x^i_1)\) for \(i = 1, \ldots, N_p\) and normalize the weights \(W^i_1 = \frac{w^i_1}{\sum_{i=1}^{N_p} w^i_1}, i = 1, \ldots, N_p\).
3: Calculate the filtered distribution \(p(X_1|Y_1) \approx \sum_{i=1}^{N_p} W^i_1 \delta(x_1 - x^i_1)\).

At time \(t \geq 2\)
4: Sample \(x^i_t \sim p_\Theta(X_t|x^i_{t-1})\) for \(i = 1, \ldots, N_p\).
5: Compute the weights \(w^i_t = W^i_{t-1} \cdot p(Y_t|x^i_t)\) for \(i = 1, \ldots, N_p\), and normalize the weights \(W^i_t = \frac{w^i_t}{\sum_{i=1}^{N_p} w^i_t}, i = 1, \ldots, N_p\).
6: Calculate the filtered distribution \(p_\Theta(X_t|Y_{1:t}) \approx \sum_{i=1}^{N_p} W^i_t \delta(x_t - x^i_t)\).
7: If the resample criterion is satisfied, i.e., \(\text{ESS} = (\sum_{i=1}^{N_p} (W^i_t)^2)^{-1} < \alpha\), then resample with replacement \(N_p\) times from \(\{x^i_t, i = 1 : N_p\}\) with the probabilities \(p(x^i_t = x^i_t) = W^i_t, i = 1, \ldots, N_p,\) and replace the previous set \{\((W^i_t, x^i_t), i = 1, \ldots, N_p\)\} by \{\((\frac{1}{N_p}, x^j_t), j = 1, \ldots, N_p\)\}.
8: Terminate when \(t = T\); otherwise \(t = t + 1\), and go back to 4.

Now we consider estimating the latent process given the total \(T\) observations, i.e., \(p_\Theta(X_t|Y_{1:T}), t = 1, \ldots, T\), by PS. Its idea is to approximate \(p_\Theta(X_t|Y_{1:T})\) with the same particles as filtering but readjust their weights by considering the information of the future observations \{\(Y_{t+1:T}\)\}, i.e.,

\[
p_\Theta(X_t|Y_{1:T}) \approx \sum_{i=1}^{N_p} W^i_{t|T} \delta(x_t - x^i_t),
\]

for \(t = 1, \ldots, T\), where

\[
W^i_{t|T} = W^i_t \frac{\sum_{j=1}^{N_p} W^j_{t+1|T} p_\Theta(x^j_{t+1}|x^i_t)}{\sum_{l=1}^{N_p} W^l_{t+1|T} p_\Theta(x^l_{t+1}|x^i_t)}, t = 1, \ldots, T - 1, i = 1, \ldots, N_p,
\]

and \(W^i_{T|T} = W^i_T, i = 1, \ldots, N_p\). The detailed particle smoothing procedure are summarized in Algorithm 2.

The computational complexities of filtering algorithms are generally much lower compared with other inference procedures, because they allow sequential updating when samples are observed incrementally. In particular, the PF updates the conditional distribution \(P(X_{t+1}|Y_{1:(t+1)})\) from \(P(X_t|Y_{1:t})\)
Algorithm 2 Particle smoothing (PS)

1: Start by setting $W_{i|T} = W_{i}^p$ for $i = 1, \ldots, N_p$.
2: For each $t = T - 1, \ldots, 1$, compute the smoothed weights by

$$W_{i|T} = W_{i}^t \sum_{j=1}^{N_p} \frac{W_{i}^j p_\Theta(x_{i+1}^j | x_t^i)}{\sum_{i=1}^{N_p} W_{i}^t p_\Theta(x_{i+1}^j | x_t^i)}, i = 1, \ldots, N_p.$$ 

3: Calculate the smoothed distribution $p_\Theta(X_t | Y_{1:T}) \approx \sum_{i=1}^{N_p} W_{i|T}^t \delta(X_t - x_t^i)$ for $t = 1, \cdots, T$.

when observing the new sample $Y_{t+1}$. Given the particle size $N_p$, the complexity of particle filtering at each step is $O(N_p)$. On the other hand, the PS updates the distribution $P(X_i | Y_{1:(t+1)})$ from $P(X_i | Y_{1:t})$ for $i = 1, \cdots, t$. It has complexity $O(tN_p^2)$ to update all state estimations at time $t$.

In our experiments using a laptop with Intel i5 CPU, it takes 0.015s for one PF iteration to obtain $P(X_i | Y_{1:500})$, $i = 1, \cdots, 500$, and 15s for PS to obtain $P(X_i | Y_{1:500})$, $i = 1, \cdots, 500$ for a two dimensional SSMP process with $N_p = 500$. In general, we believe the computational load with increasing $N_p$ should be of less concern due to the fast development of high performance parallel computing. In fact, PF is inherently parallel since it essentially consists of exploration of the state space by random but independent particles. The particles only interact when their weights need normalization. As a result, parallel particle filtering can be developed to take advantage of the high performance computing. We refer the readers to (Brun et al., 2002; Durham and Geweke, 2011) and subsequent references for more details in such development. In addition, more PF algorithms have been developed to be scalable in the ultra-high dimensional cases with small computation complexity (e.g. Beskos et al., 2014b). These algorithms may shed light upon the application of SSMP in high-dimensional cases, which we will explore further in our future studies.

It should also be noted that in general particle filtering requires more particle samples for higher dimensional state space model estimation. This is because the space to be sampled increases drastically with the dimension $d$, which makes it much harder for particles to efficiently propagate to the subspace with identifiably nonzero PDF. Therefore, weight degeneracy is the fundamental obstacle for particle filtering in high-dimensional models. To avoid collapse, the particle sample size $N_p$ should be large enough. For Gaussian state space models, some particle filtering methods, such as the bootstrap particle filter, can keep stable and consequently converge as long as $N_p$ grows exponentially fast with $d$ (Bengtsson et al., 2008). The corresponding estimation error is bounded in the order of $c_t / \sqrt{N_p}$, where $c_t$ is a constant independent with $d$ (Theorem 4.3.1, Smith et al., 2013). In addition, it should be also noted that $c_t$ increases exponentially with $t$. Therefore $N_p$ should also increase exponentially in $t$ in order to
achieve a given accuracy at time \( t \). Considering these features, we suggest \( N_p \) be exponential in \( d \) and \( T \) (the total time series length) to ensure the performance of the algorithm. According to some additional simulation results, we find that for SSMP with \( d = 10 \) and \( T = 500 \), \( N_p = 1000 \) is sufficient to guarantee the convergence of PF, and consequently the accuracy of the inference (in terms of parameter estimation and prediction). For extremely long time series, recalibration using full Bayesian MCMC can be performed periodically with interval \( T^* \). \( T^* \) is the maximum time length for which the approximation of PF can achieve a pre-specified accuracy. The full Bayesian MCMC can guarantee obtaining accurate samples from the posterior distributions \( P(X_{kT^*}|Y_{1:kT^*}) \), \( k = 1, 2, \cdots \). Consequently, these samples can be used as particles for future filtering before next calibration, which can effectively avoid accumulation of errors without substantially increasing computational load.

### 3.2 Parameter Estimation

This section considers estimation of the parameters \( \Theta \) of SSMP. In the Maximum Likelihood Estimation (MLE) framework, a natural and efficient estimation method to deal with latent variables is the Expectation Maximization (EM) algorithm. The EM algorithm is an iterative procedure to seek for \( \Theta^{(q)} \) in the \( q^{th} \) iteration such that the likelihood is increased from that in the \( (q-1)^{th} \) iteration. Its key idea is to postulate the “missing” data \( \{X_{1:T}\} \) and to consider maximizing the likelihood function given the complete data \( \{X_{1:T}, Y_{1:T}\} \). Underlying this strategy is the idea that maximizing the “complete” log-likelihood \( \log p_{\Theta}(X_{1:T}, Y_{1:T}) \) is easier than maximizing the incomplete one \( \log p_{\Theta}(Y_{1:T}) \). Here due to the Markovian structure of SSMP, the complete data log-likelihood has the form

\[
\log p_{\Theta}(X_{1:T}, Y_{1:T}) = \log p_0(X_1) + \sum_{t=1}^{T-1} \log p_{\Theta}(X_{t+1}|X_t) + \sum_{t=1}^{T} \log p(Y_t|X_t). \tag{23}
\]

However, because \( \{X_{1:T}\} \) are unavailable, the EM algorithm replaces (23) by \( Q(\Theta, \Theta^{(q)}) \), which is the conditional expectation of (23) with respect to \( \{X_{1:T}\} \) given the observations \( \{Y_{1:T}\} \) using the parameters \( \Theta^{(q)} \) in the current iteration, i.e.,

\[
E \text{ step: } Q(\Theta, \Theta^{(q)}) = \int p_{\Theta^{(q)}}(X_{1:T}|Y_{1:T}) \cdot \log p_{\Theta}(X_{1:T}, Y_{1:T}) dX_{1:T}. \tag{24}
\]

Then we want to find the revised parameter estimates \( \Theta^{(q+1)} \) that maximize the function

\[
M \text{ step: } \Theta^{(q+1)} = \arg \max_\Theta Q(\Theta, \Theta^{(q)}). \tag{25}
\]
For SSMP, we can get $Q(\Theta, \Theta^{(q)})$ in (24) as

$$Q(\Theta, \Theta^{(q)}) = \sum_{t=1}^{T-1} \int p_{\Theta^{(q)}}(X_t, X_{t+1} | Y_{1:T}) \cdot \log p_{\Theta}(X_{t+1} | X_t) dX_t dX_{t+1}. \tag{26}$$

Unfortunately here $p_{\Theta^{(q)}}(X_t, X_{t+1} | Y_{1:T})$ is not analytical and consequently $Q(\Theta, \Theta^{(q)})$ is intractable.
However, on the other hand, the particles used in PF & PS can be viewed as samples from the conditional distribution. As a result, we can use these particles to approximate $p_{\Theta^{(q)}}(X_t, X_{t+1} | Y_{1:T})$, and consequently to implement the Monte Carlo EM (MCEM) algorithm for parameter estimation. In more details, from (19) and (21), we can get

$$\hat{Q}(\Theta, \Theta^{(q)}) \approx \sum_{t=1}^{T-1} \sum_{i=1}^{N_p} \sum_{j=1}^{N_p} W^{ij}_{t,t+1/T} \log p_{\Theta}(x^j_{t+1} | x^i_t), \quad \text{where} \quad W^{ij}_{t,t+1/T} = \frac{W^j_{t+1/T} p_{\Theta^{(q)}}(x^j_{t+1} | x^i_t)}{\sum_{i=1}^{N_p} W^i_{t+1/T} p_{\Theta^{(q)}}(x^j_{t+1} | x^i_t)}. \tag{27}$$

For more detailed derivations together with the convergence properties about PF & PS based MCEM, please refer to Schönh et al. (2011).

In the M step, with the gradient available for (27), we get $\Theta^{(q+1)} = \{\mu^{(q+1)}, \Phi^{(q+1)}, \Sigma^{(q+1)}\}$ as

$$\Pi^{(q+1)} = [(I - \Phi^{(q+1)}) \mu^{(q+1)}, \Phi^{(q+1)}]' = \left( \sum_{t=1}^{T-1} \sum_{i=1}^{N_p} \sum_{j=1}^{N_p} W^{ij}_{t,t+1/T} x^j_{t+1} z^i_\ell \right) \left( \sum_{t=1}^{T-1} \sum_{i=1}^{N_p} \sum_{j=1}^{N_p} W^{ij}_{t,t+1/T} z^i_\ell z^j_\ell \right)^{-1}, \tag{28}$$

$$\Sigma^{(q+1)} = \frac{1}{T-1} \sum_{t=1}^{T-1} \sum_{i=1}^{N_p} \sum_{j=1}^{N_p} W^{ij}_{t,t+1/T} (x^j_{t+1} - \Pi^{(q+1)} z^i_\ell)(x^j_{t+1} - \Pi^{(q+1)} z^i_\ell)'. \tag{29}$$

where $z^i_\ell = [1, x^i_\ell]$. For detailed derivation of the EM algorithm, please refer to Appendix. The procedure of the particle EM-based estimation is summarized in Algorithm 3.

**Algorithm 3 Monte Carlo EM algorithm (MCEM)**

1: Set $q = 0$ and initialize $\Theta^{(q)}$ such that $\log p_{\Theta^{(q)}}(Y_{1:T})$ is finite.
2: Expectation (E) Step:
   - Run Algorithm 1 and 2 to obtain the filtered and smoothed distributions for $t = 1, \ldots, T$.
   - Calculate $\hat{Q}(\Theta, \Theta^{(q)})$ according to (27).
3: Maximization (M) Step:
   - Compute $\Theta^{(q+1)}$ according to (28) and (29).
4: Check the non-termination condition $\hat{Q}(\Theta^{(q+1)}, \Theta^{(q)}) - \hat{Q}(\Theta^{(q)}, \Theta^{(q)}) \geq \epsilon$ for some $\epsilon \geq 0$. If satisfied, update $q \rightarrow q + 1$, and return to 2, otherwise terminate.
The following simulation has been conducted to illustrate the performance. We set $N_p = 500$ to estimate a 2-dimensional SSMP process with series length $T = 500$. The parameters are set to be $\Phi = [0.6, 0.1; 0.2, 0.7]$, $\mu = [4, 4]$, and $\Sigma = 0.25I$. We replicate the simulation 200 times. Each replication includes data generation, estimation, and prediction, to evaluate the performance of the proposed method. For every replication, we randomly pick an initial value of $\Theta^{(0)}$ and estimate the parameters iteratively based on the MCEM algorithm. Table 2 lists the estimation results. We observe that both the bias and the rooted mean square error (RMSE) of the estimators are acceptably small, illustrating the satisfactory estimation accuracy and stability of the MCEM algorithm. Then we use the estimated parameters $\Theta$ to track the latent states $\{X_{T+1:T+100}\}$ for the subsequent 100 observations $\{Y_{T+1:T+100}\}$ according to Algorithm 1. The filtering results in one replication are shown in Figure 3. We can see that the tracked states (red crosses) based on PF almost overlap with the true states (the blue circles) with slight differences.

Table 2: Estimation bias and RMSE of Algorithm 3 based on 200 replicates.

<table>
<thead>
<tr>
<th></th>
<th>True Value</th>
<th>Estimate</th>
<th>Bias</th>
<th>RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\phi_1$</td>
<td>0.6</td>
<td>-0.0098</td>
<td>0.0409</td>
<td></td>
</tr>
<tr>
<td>$\phi_2$</td>
<td>0.2</td>
<td>-0.0035</td>
<td>0.0414</td>
<td></td>
</tr>
<tr>
<td>$\phi_3$</td>
<td>0.1</td>
<td>-0.0020</td>
<td>0.0290</td>
<td></td>
</tr>
<tr>
<td>$\phi_4$</td>
<td>0.7</td>
<td>-0.0077</td>
<td>0.0334</td>
<td></td>
</tr>
<tr>
<td>$\mu_1$</td>
<td>4.0</td>
<td>0.0104</td>
<td>0.0786</td>
<td></td>
</tr>
<tr>
<td>$\mu_2$</td>
<td>4.0</td>
<td>0.0104</td>
<td>0.1025</td>
<td></td>
</tr>
<tr>
<td>$\sigma_{11}$</td>
<td>0.25</td>
<td>-0.0036</td>
<td>0.0174</td>
<td></td>
</tr>
<tr>
<td>$\sigma_{12}$</td>
<td>0</td>
<td>0.002</td>
<td>0.0140</td>
<td></td>
</tr>
<tr>
<td>$\sigma_{22}$</td>
<td>0.25</td>
<td>-0.0031</td>
<td>0.0197</td>
<td></td>
</tr>
</tbody>
</table>

Figure 3: The tracked states $\{X_{T+1:T+100}\}$ based on Algorithm 1 with parameters estimated by Algorithm 3.
4 Case Studies

4.1 Simulation Studies

As emphasized earlier, the most advantageous property of SSMP is that it allows for more flexible cross-correlation and auto-correlation structures of count data compared with other state-of-the-art models. Here we demonstrate this point using some numerical studies. We compare SSMP with the log-normal mixture Poisson model (LP) of Aitchison and Ho (1989) which is a sub-model of SSMP by setting the autocorrelation structure $\Phi = 0$, and the other two time series models, MACP of Heinen and Rengifo (2007) with orders $p = 1$ and $q = 1$ (shorted as MACP (1,1)), and MINAR(1) of Pedeli and Karlis (2013b). In our experiments, 500 observations are generated from each model, and then fitted by all these models using MLE methods separately. Here we adopt the Bayesian information criterion (BIC) to evaluate their fitting performances. In particular,

$$BIC = -2 \ln \hat{L} + k \ln(n)$$

where $\hat{L}$ is the fitted likelihood, $k$ is the number of parameters in the model, $n$ is the number of observations. The parameters to be estimated for every model is listed in Table 3 with the same notations as those in the original papers, together with the corresponding $k$ for a 2-dimensional Poisson process.

Table 3: Parameters to be estimated for different models together with $k$ for a 2-dimensional Poisson process

<table>
<thead>
<tr>
<th>Model</th>
<th>MINAR(1)</th>
<th>MACP(1,1)</th>
<th>SSMP</th>
<th>LP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameters</td>
<td>$A, \lambda_1, \lambda_2, \phi$</td>
<td>$A, B, \omega, \phi, (\Sigma)$</td>
<td>$\Phi, \mu, \Sigma$</td>
<td>$\mu, \Sigma$</td>
</tr>
<tr>
<td>$k$</td>
<td>7</td>
<td>11(14)</td>
<td>9</td>
<td>5</td>
</tr>
</tbody>
</table>

For illustration purposes, here we simply consider processes of bivariate counts with either positive or negative cross-correlations or autocorrelations. Table S.2 summarizes the true parameters and the fitted ones of all these models, and Table 4 presents their BIC values with the corresponding logarithm of the fitted likelihood in the parenthesis.

Furthermore, we consider the prediction power of the fitted models. Following Czado et al. (2009), we adopt the Dawid-Sebastiani (DS) score to evaluate the prediction power. In particular, for every fitted model, we can derive the one-step ahead prediction probability $P(Y_t|Y_{1:t-1}, \Theta)$ for $t = 2, ..., T$ and calculate the DS score for every dimension. Denote $\mu_t = E(Y_t|Y_{1:t-1}) \equiv [\mu_{t,1}, \mu_{t,2}, \cdots, \mu_{t,p}]$ and $\sigma_t = Cov(Y_t|Y_{1:t-1})$ with diagonal element $[\sigma_{t,1}, \sigma_{t,2}, \cdots, \sigma_{t,p}]$. Then the DS score for dimension $i$
is defined as

$$DSS_{t,i}(Y_{t,i}) = \frac{Y_{t,i} - \mu_{t,i}}{\sigma_{t,i}} + 2 \log(\sigma_{t,i}).$$

Table 5 reports the mean of the DS scores over the $T - 2$ samples for different fitted models, with the mean squared prediction error (MSPE) shown in the parenthesis as well. The results are consistent with those in Table 4. In both tables, the first sign in SSMP indicates sign of the cross-correlation, while the second sign indicates sign of the autocorrelation.

We can see that generally, the results are consistent with the results based on BIC values. The fitted model which has the same form as the true model has the lowest DS score, meaning the best prediction performance. However, SSMP is always the second best with a slight higher DS score or BIC value. This demonstrates that SSMP is able to describe count data generating from different models (mechanisms). Furthermore, for data coming from MINAR(1,1) or MACP(1,1) or positive auto-correlated and cross-correlated SSMP (the fifth row), the DS scores of different fitted models are similar. This indicates that all three models can describe these processes equally well. On the other hand, MINAR(1,1) has largest BIC values and DS scores for data coming from LP or negative cross-correlated or auto-correlated SSMP data (the forth or sixth row). The reason for the former is that the LP data has large overdispersion, and the reason for the later is that MINAR cannot describe either negative cross-correlations or negative autocorrelations. On the contrary, though MACP(1,1) can fit the LP data or negative cross-correlated SSMP data, it still fails to predict data with negative auto-correlations. As to LP, it has comparatively small BIC values, but large DS scores for almost all the cases, indicating that LP may overfit the data.

Table 4: The BIC values of the fitted models with the corresponding logarithm of the fitted likelihood in the parenthesis.

<table>
<thead>
<tr>
<th>Data Model</th>
<th>MINAR(1)</th>
<th>MACP(1,1)</th>
<th>SSMP</th>
<th>LP</th>
</tr>
</thead>
<tbody>
<tr>
<td>MINAR(1)</td>
<td>5324(-2640)</td>
<td>5378(-2655)</td>
<td>5442(-2693)</td>
<td>5499(-2730)</td>
</tr>
<tr>
<td>MACP(1,1)</td>
<td>5742(-2849)</td>
<td>5624(-2781)</td>
<td>5636(-2790)</td>
<td>6634(-3296)</td>
</tr>
<tr>
<td>LP</td>
<td>6241(-3099)</td>
<td>6036(-2984)</td>
<td>5924(-2934)</td>
<td>5867(-2915)</td>
</tr>
<tr>
<td>SSMP(++)</td>
<td>5800(-2878)</td>
<td>5456(-2694)</td>
<td>5426(-2685)</td>
<td>5469(-2720)</td>
</tr>
<tr>
<td>SSMP(+-)</td>
<td>5634(-2790)</td>
<td>5432(-2682)</td>
<td>5426(-2685)</td>
<td>5469(-2720)</td>
</tr>
</tbody>
</table>

4.2 A Real Application in the Power Utility Industry

Now we use the proposed SSMP as well as the other three models to analyze the dynamic interactions of different types of damages which occurred in a power utility system. The dataset records the counts
Table 5: The DS score for different models (with the mean squared prediction error shown in the parenthesis).

<table>
<thead>
<tr>
<th>Fitted models</th>
<th>Data Model</th>
<th>MINAR(1)</th>
<th>MACP(1,1)</th>
<th>SSMP</th>
<th>LP</th>
</tr>
</thead>
<tbody>
<tr>
<td>MINAR(1)</td>
<td>3.45(10.74)</td>
<td>3.48(12.03)</td>
<td>3.43(10.98)</td>
<td>3.66(14.15)</td>
<td></td>
</tr>
<tr>
<td>MACP(1,1)</td>
<td>3.71(15.03)</td>
<td>3.73(15.39)</td>
<td>3.79(15.97)</td>
<td>4.70(40.50)</td>
<td></td>
</tr>
<tr>
<td>LP</td>
<td>5.05(30.44)</td>
<td>4.39(27.86)</td>
<td>4.46(27.92)</td>
<td>4.30(27.17)</td>
<td></td>
</tr>
<tr>
<td>SSMP(-)</td>
<td>3.85(16.01)</td>
<td>3.59(13.96)</td>
<td>3.42(10.49)</td>
<td>3.74(15.54)</td>
<td></td>
</tr>
<tr>
<td>SSMP(++)</td>
<td>3.68(14.65)</td>
<td>3.63(14.01)</td>
<td>3.61(12.69)</td>
<td>3.72(15.17)</td>
<td></td>
</tr>
<tr>
<td>SSMP(+-)</td>
<td>4.58(24.89)</td>
<td>22.36(75.58)</td>
<td>3.68(12.35)</td>
<td>3.95(14.68)</td>
<td></td>
</tr>
</tbody>
</table>

of three types of damages in a region every day for a period of $T = 100$ days. Due to confidentiality reasons, we reserve the detailed information of these three types of damages, but simply denote them as type A, type B, and type C. Different types of damages are caused by some common weather-related conditions or accidents. Therefore they may have certain contemporaneous correlations with each other and serial correlations with their previous observations. Figure 4 draws the logarithm of the count data over time. Their similar change patterns reveal their cross-correlation structure to some degree.

Figure S.1 (hereinafter, S. denotes the figures and tables in the on-line supplementary file) draws the autocorrelation function of these counts, from which we can see strong serial correlations. Descriptive statistics for the count data are also summarized in the left part of Table 6. We can see that the empirical marginal distributions of the count data are clearly over-dispersed, and their sample correlations are all positive.

![Figure 4: Logarithm of the count data in the power utility system.](image)

We use these samples to fit SSMP. The parameter estimates based on Algorithm 3 with $N = 21$ replications are illustrated in Table 7. In every replication we initialize $\Theta^{(0)}$ by adding some random noise to the calculated $\{\mu, \Phi, \Sigma\}$ in the left part of Table 6. We set $N_p = 1000$ to ensure the accuracy of the E-step. Each iteration takes around 200 seconds to complete on a single-core personal computer. Figure S.2 shows the convergence process of every replication. We see generally the estimates converge fast in the first $n = 5$ iterations. Based on the fitted model, we draw the one-step ahead prediction...
Table 6: Descriptive statistics for the count data in the power utility system.

<table>
<thead>
<tr>
<th>Data</th>
<th>Fitted SSMP</th>
<th></th>
<th></th>
<th>Fitted LP</th>
<th></th>
<th></th>
<th>Fitted MACP(1,1)</th>
<th></th>
<th></th>
<th>Fitted MINAR(1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td>40.1 26.2 21.6</td>
<td>39.9 27.4 21.8</td>
<td>40.2 25.7 20.8</td>
<td>43.1 30.7 26.3</td>
<td>64.7 42.0 38.8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Standard dev.</td>
<td>40.6 19.5 20.6</td>
<td>38.7 16.5 18.5</td>
<td>40.0 15.6 17.2</td>
<td>28.2 15.5 17.1</td>
<td>44.7 42.0 38.8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Overdispersion</td>
<td>41.1 14.5 19.6</td>
<td>37.5 9.9 15.4</td>
<td>39.7 9.5 14.2</td>
<td>18.4 7.8 11.1</td>
<td>1.1 1 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Cross-correlation:
- Mean: 1.00
- Standard dev.: 0.48 1.00
- Overdispersion: 0.78 0.74 1.00

Lag 1:
- Mean: 0.45 0.26 0.40
- Standard dev.: 0.33 0.17 0.28
- Overdispersion: 0.36 0.14 0.40

Autocorrelation:
- Mean: 0.34 0.65 0.53
- Standard dev.: 0.27 0.57 0.43
- Overdispersion: 0.24 0.36 0.54

Loglikelihood:
- Mean: -769 679(559)
- Standard dev.: -815 7.51(809)
- Overdispersion: -1258 7.11(587)

Table 7: The parameter estimates for SSMP based on Algorithm (3) with \( N = 21 \) replicates (The standard deviations are shown in parentheses).

<table>
<thead>
<tr>
<th></th>
<th>type A</th>
<th>type B</th>
<th>type C</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \mu_i )</td>
<td>3.361 (0.0006)</td>
<td>3.170 (0.0007)</td>
<td>2.830 (0.0007)</td>
</tr>
<tr>
<td>( \phi_{ij} )</td>
<td>0.315 (0.0015)</td>
<td>-0.043 (0.0037)</td>
<td>0.182 (0.0035)</td>
</tr>
<tr>
<td></td>
<td>-0.019 (0.0015)</td>
<td>0.579 (0.0028)</td>
<td>0.108 (0.0024)</td>
</tr>
<tr>
<td></td>
<td>0.154 (0.0011)</td>
<td>0.246 (0.0029)</td>
<td>0.258 (0.0022)</td>
</tr>
<tr>
<td>( \sigma_{ij} )</td>
<td>0.532 (0.0009)</td>
<td>0.124 (0.0007)</td>
<td>0.154 (0.0009)</td>
</tr>
<tr>
<td></td>
<td>0.272 (0.0007)</td>
<td>0.138 (0.0009)</td>
<td>0.361 (0.0015)</td>
</tr>
</tbody>
</table>

From the parameter estimates in Table 7 and Table S.1, we can compute the implied estimates for the moments of the unconditional distribution of the count data for every fitted model, and compare them with their sample counterparts as shown in the left part of Table 6. We can see that for SSMP, its calculated moments and overdispersion closely match the sample counterparts, indicating that SSMP provides good representations of the cross-correlations and autocorrelations of the count data. Though MACP has poor fitted log-likelihood, it yet has almost as good moment results as SSMP. This good result come from the Gaussian copula, which successfully captures the left correlation structures besides the

of \( \log(Y_t) \) in Figure S.3. We also fit the LP, MACP(1,1) and MINAR(1) models using the data. Their corresponding parameter estimates are shown in Table S.1. It should be noted that because the dimension of the count data is 3, we adopt Pedeli and Karlis (2013a), the only MINAR(1) model dealing with counts with more than 2 dimensions for comparison. However this work assumes the binomial thinning matrix only has diagonal components. Their fitted log-likelihood is shown in Table 6.
conditional model. As to LP, it can still describe the cross-correlations satisfactorily, however it cannot describe the autocorrelations at all. While for MINAR(1), it fails to provide convincing fitting results for any statistic, which is mainly due to its prohibition of overdispersion and off-diagonal autocorrelations. We also compared the prediction power of these fitted models in terms of Dawid-Sebastiani (DS) scores as well as the mean squared prediction error (MSPE) as shown in Table 6, where SSMP gives the smallest DS score and MSPE.

We further do model diagnoses. If a model is well specified, its normalized Pearson residuals for every dimension should have zero mean and unit variance, and be serially uncorrelated. From the bottom part of Table 6, we can conclude that SSMP and MACP(1,1) are more competitive to describe the count data compared with LP and MINAR(1). For SSMP, in every dimension its residual mean is close to zero and its variance is close to one. Its lag 1 autocorrelation matrix has almost zero values. The Ljung-Box statistics for the residuals and their substantial $p$-values also demonstrate their serial independence to some degree. For MACP(1,1), its residuals also have almost zero mean and unit variance, however they tend to be slightly autocorrelated with the Ljung-Box statistic $Q_3$ for the second dimension bigger than the $p$ value of the 95% confidence interval. For the other two models, the results are even worse with their residuals significantly autocorrelated. To further check the distributional assumptions of SSMP and MACP(1,1), we use the "randomized" version of the probability integral transform (PIT) proposed by Liesenfeld et al. (2008) for diagnosis (Jung et al., 2011). If a model is correctly specified, its randomized PIT values should follow the standard uniform distribution. Figure 5 shows the corresponding quantile-quantile plots of the randomized PIT values for SSMP and MACP(1,1). The PIT values of SSMP in every dimension nearly coincide with the 45-degree line, indicating their similarity to the uniform distribution. The formal Kolmogorov-Smirnov (KS) test for every dimension also does not reject the uniformity assumption. While the plots of MACP(1,1) derive more from the 45-degree line, with the KS tests rejecting the uniformity assumption. All in all, these results above show that SSMP provides a much better description of the dynamic interactions of the count data than the other three models.

5 Concluding Remarks

Though multivariate time series of counts are very common in practice, models to describe them allowing for flexible cross-correlation and autocorrelation structures are yet to be addressed. To fill this gap, this paper proposes an easy-to-interpret state space model to describe autocorrelated multivariate counts. This model can represent the contemporaneous and serial correlations of counts in a flexible way.
and capture the overdispersion as well. Hence this model provides a useful framework for multivariate count series analysis. A stable and efficient estimation procedure for this model is provided based on the MCEM algorithm together with PF & PS methods. PF can also track the latent states of the model accurately in a sequential way with small computational complexity. Comparisons with other state-of-the-art models of multivariate counts demonstrate the superiority and more generality of our proposed model. This point is also illustrated by applying the proposed model in a real dataset from the power utility industry.

Along this research direction, we can explore the research in the following aspects. Firstly, though efficient statistical process control (SPC) for autocorrelated multivariate counts is in high demand, so far to our best knowledge, there is no work targeting on it. Our proposed model may shed light on this field by constructing a SPC scheme to monitor the model parameters. Secondly, in some applications count data involve spacial information and are further spatially correlated. It is interesting to extend the current model by taking the spacial interdependence of count data into consideration, i.e., to construct a multi-layer time series model, which aims to analyze not only the lead-lag relations within and between different time series, but also those within and between different spacial layers. Last but not least, from the implementable point of view, it is reasonable to consider how to deal with missing data, or how to add regression covariates into the model for a better explanation of count data.
Acknowledgment

The authors are grateful to the numerous valuable comments provided by the editors and referees. Nan Chen was partially supported by Singapore AcRF Funding R-266-000-085-112 and National Research Foundation Singapore under its Campus for Research Excellence and Technological Enterprise (CREATE).

Appendix

A.1 The Moment Properties of SSMP

A.1.1 Mean

The mean of $Y_t$ is

$$E(Y_t) = E[E(Y_t|X_t)] = E[exp(X_t)] = exp(\mu + \frac{1}{2}\Gamma), \quad (30)$$

with

$$E(Y_{ti}) = exp(\mu_i + \frac{1}{2}\Gamma_{ii}). \quad (31)$$

A.1.2 Covariance Matrix

The variance of $Y_t$ is

$$Var(Y_t) = E[Var(Y_t|X_t)] + Var[E(Y_t|X_t)]$$
$$= E[diag(exp(X_t)))] + Var[exp(X_t)], \quad (32)$$

therefore

$$Var(Y_{ti}) = E(Y_{ti}) + e^{2\mu_i+\Gamma_{ii}(e^{\Gamma_{ii}} - 1)},$$
$$Cov(Y_{ti}, Y_{tj}) = e^{\mu_i+\mu_j + \frac{1}{2}(\Gamma_{ii}+\Gamma_{jj})(e^{\Gamma_{ij}} - 1)}. $$
A.1.3 Autocorrelation Structure

the autocorrelation of $Y_t$ is

\[ \text{Cov}(Y_t, Y_{t-\tau}) = \text{E} \left[ \text{Cov}(Y_t, Y_{t-\tau}) | X_t, X_{t-\tau} \right] + \text{Cov} \left[ \text{E}(Y_t | X_t), \text{E}(Y_{t-\tau} | X_{t-\tau}) \right] \]

\[ = \text{Cov}(\exp(X_t), \exp(X_{t-\tau})), \tag{33} \]

because of

\[ \text{E} \left[ \text{Cov}(Y_t, Y_{t-\tau}) | X_t, X_{t-\tau} \right] = 0, \]

then we have

\[ \text{Cov}(Y_{ti}, Y_{(t-\tau)ij}) = \text{Cov}(\exp(X_{ti}), \exp(X_{(t-\tau)ij})) \]

\[ = \text{E} \left[ \exp(X_{ti}) \exp(X_{(t-\tau)ij}) \right] - \text{E} \left[ \exp(X_{ti}) \right] \text{E} \left[ \exp(X_{(t-\tau)ij}) \right] \]

\[ = \text{E} \left[ \exp(X_{ti} + X_{(t-\tau)ij}) \right] - \text{E} \left[ \exp(X_{ti}) \right] \text{E} \left[ \exp(X_{(t-\tau)ij}) \right] \]

\[ = \exp(\mu_i + \mu_j) \left\{ \text{E} \left[ \exp(X_{ti} - \mu_i + X_{(t-\tau)ij} - \mu_j) \right] - \exp \left( \frac{\Gamma_{ii}}{2} + \frac{\Gamma_{jj}}{2} \right) \right\}. \tag{34} \]

Since

\[ X_t - \mu = \Phi(\Phi(X_{t-2} - \mu) + \epsilon_{t-1}) + \epsilon_t \]

\[ = \Phi^\tau(X_{t-\tau} - \mu) + \Phi^{\tau-1}\epsilon_{t-\tau+1} + \cdots + \Phi^1\epsilon_{t-1} + \epsilon_t, \]

we define $\phi^\tau_{i-k} = \left[ \phi^\tau_{i,1}, \cdots, \phi^\tau_{i,d} \right]^\prime$ as a row vector which represents the $i^{th}$ row of $\Phi^{\tau-k}$. Then

\[ X_{ti} + X_{t-\tau j} - \mu_i - \mu_j = \phi^\tau_i(X_{t-\tau} - \mu) + \phi^{\tau-1}_i\epsilon_{t-\tau+1} + \cdots + \phi^1_i\epsilon_{t-1} + \epsilon_t + X_{t-\tau j}. \]

We define $\phi^\tau_i = \left[ \phi^\tau_{i,1}, \cdots, \phi^\tau_{i,j} + 1, \cdots, \phi^\tau_{i,d} \right]^\prime$, then

\[ X_{ti} + X_{t-\tau j} - \mu_i - \mu_j = \phi^{\tau*}_i(X_{t-\tau} - \mu) + \phi^{\tau-1}_i\epsilon_{t-\tau+1} + \cdots + \phi^1_i\epsilon_{t-1} + \epsilon_t, \]

and consequently

\[ \text{E} \left[ \exp(X_{ti} + X_{(t-\tau)j} - \mu_i - \mu_j) \right] = \text{E} \left[ \exp(\phi^{\tau*}_i(X_{t-\tau} - \mu)) \right] \cdot \text{E} \left[ \exp(\phi^{\tau-1}_i\epsilon_{t-\tau+1}) \right] \]

\[ \cdots \cdot \text{E} \left[ \exp(\phi^1_i\epsilon_{t-1}) \right] \cdot \text{E} \left[ \exp(\epsilon_{t,t}) \right]. \]
According to the moment-generating function of the multivariate normal distribution, if \( \mathbf{X} \sim \mathbf{N}(\mu, \Sigma) \), then the moment-generating function is given by \( m_x(t) = \mathbb{E}\{\exp(t' \mathbf{X})\} = \exp(t' \mu + \frac{1}{2}t' \Sigma t) \). Then we have

\[
\mathbb{E}\left[\exp(X_{ti} + X_{(t-\tau)j} - \mu_i - \mu_j)\right] = \exp\left(\frac{1}{2} \phi_i^\top \Gamma(0) \phi_i + \frac{1}{2} \phi_i^\top \Sigma \phi_i^{-1} + \cdots + \frac{1}{2} \phi_i^\top \Sigma \phi_i + \frac{1}{2} \Sigma_{ii}\right).
\]

(35)

Plug (34) into (35). Finally, we have the covariance of \( Y_{ti} \) and \( Y_{(t-\tau)j} \). Then the autocorrelation could be got by

\[
\rho(Y_{ti}, Y_{(t-\tau)j}) = \frac{\text{Cov}(Y_{ti}, Y_{(t-\tau)j})}{\sqrt{\text{Var}(Y_{ti}) \text{Var}(Y_{(t-\tau)j})}}.
\]

(36)

### A.2 The MCEM algorithm based on PF & PS Methods

For the M step, to get

\[
\Theta^* = \arg \max_{\Theta} Q(\Theta^*, \Theta),
\]

(37)

We need take the gradient of \( \hat{Q}(\Theta^*, \Theta) \) with respect to \( \Theta^* \), i.e.,

\[
\frac{\partial \hat{Q}(\Theta^*, \Theta)}{\partial \Theta^*} = \sum_{t=1}^{T-1} \sum_{i=1}^{N_p} \sum_{j=1}^{N_p} W_{ij} \frac{\partial \log p_{\Theta^*}(x_{t+1}^j|x_t^i)}{\partial \Theta^*}.
\]

(38)

By rewriting (10) as

\[
\mathbf{X}_{t+1} = \Phi \mathbf{X}_t + (I - \Phi) \mu + \epsilon_t = \Phi \mathbf{X}_t + \mathbf{c} + \epsilon_t,
\]

(39)

where \( \mathbf{c} = (I - \Phi) \mu \) and dropping the constant, we have

\[
\log p_{\Theta^*}(\mathbf{X}_{t+1}|\mathbf{X}_t) = - \log |\Sigma^*| - \frac{1}{2} (\mathbf{X}_{t+1} - \Phi^* \mathbf{X}_t - \mathbf{c}^*)' \Sigma^*^{-1} (\mathbf{X}_{t+1} - \Phi^* \mathbf{X}_t - \mathbf{c}^*)
\]

\[
= - \log |\Sigma^*| - \frac{1}{2} (\mathbf{X}_{t+1} - \Pi^* \mathbf{Z}_t)' \Sigma^*^{-1} (\mathbf{X}_{t+1} - \Pi^* \mathbf{Z}_t)
\]

\[
= - \log |\Sigma^*| - \frac{1}{2} \text{tr}((\mathbf{X}_{t+1} - \Pi^* \mathbf{Z}_t)' \Sigma^*^{-1} (\mathbf{X}_{t+1} - \Pi^* \mathbf{Z}_t)),
\]

(40)

where \( \Pi^* = [\mathbf{c}^*, \Phi^*]' \), and \( \mathbf{Z}_t = [1, \mathbf{X}_t] \).
Taking the derivative of $\log p_{\Theta^*}(X_{t+1}|X_t)$, we have

$$
\frac{\partial \log p_{\Theta^*}(X_{t+1}|X_t)}{\partial \Pi^*} = \Sigma^{-1}(X_{t+1}Z_t^* - \Pi^*Z_tZ_t^*),
$$

$$
\frac{\partial \log p_{\Theta^*}(X_{t+1}|X_t)}{\partial \Sigma^{-1}} = \frac{1}{2} (X_{t+1} - \Pi^*Z_t)(X_{t+1} - \Pi^*Z_t)^*.
$$

(41)

Therefore we have

$$
\frac{\partial \hat{Q}(\Theta^*, \Theta)}{\partial \Pi^*} = \Sigma^{-1} \sum_{t=1}^{T-1} \sum_{i=1}^{N_p} \sum_{j=1}^{N_p} W_{t,t+1}^{ij}(x_{t+1}^j - \Pi^*Z_t^*)Z_t^* = 0,
$$

$$
\frac{\partial \hat{Q}(\Theta^*, \Theta)}{\partial \Sigma^{-1}} = \frac{T-1}{2} \Sigma^* - \frac{1}{2} \sum_{t=1}^{T-1} \sum_{i=1}^{N_p} \sum_{j=1}^{N_p} W_{t,t+1}^{ij}(x_{t+1}^j - \Pi^*Z_t^*)Z_t^* = 0,
$$

(42)

with its solution

$$
\Pi^* = [(I - \Phi^*)\mu^*, \Phi^*]^* = \left( \sum_{t=1}^{T-1} \sum_{i=1}^{N_p} \sum_{j=1}^{N_p} W_{t,t+1}^{ij}x_{t+1}^jZ_t^* \right)^{-1},
$$

(43)

and

$$
\Sigma^* = \frac{1}{T-1} \left( \sum_{t=1}^{T-1} \sum_{i=1}^{N_p} \sum_{j=1}^{N_p} W_{t,t+1}^{ij}(x_{t+1}^j - \Pi^*Z_t^*)(x_{t+1}^j - \Pi^*Z_t^*)^* \right).
$$

(44)

References


**Biography**

Chen Zhang is a Ph.D candidate in the Department of Industrial and Systems Engineering at National University of Singapore. She received her B.Eng. degree in Electronic Science and Technology (Optics) from Tianjin University. Her research interests include developing new approaches for modeling and monitoring of engineering systems with complex data. She is a member of IIE and INFORMS.

Nan Chen is an Assistant Professor in the Department of Industrial and Systems Engineering at National University of Singapore. He obtained his B.S. degree in Automation from Tsinghua University, and M.S.
degree in Computer Science, M.S. degree in Statistics, and Ph.D. degree in Industrial Engineering from University of Wisconsin-Madison. His research interests include statistical modeling and surveillance of complex systems, simulation modeling & design, condition monitoring and degradation modeling. He is a member of INFORMS, IIE, and IEEE.

Zhiguo Li is a Research Staff Member in statistics and data science with IBM T. J. Watson Research Center. He holds a Ph.D. in Industrial Engineering and an M.S. in Statistics from the University of Wisconsin-Madison, and an M.S. and a B.S. in Automotive Engineering from Tsinghua University. He has many years of experience in industrial statistics, data mining, reliability and quality engineering, and system dynamics with the applications to different engineering and business processes/systems including product management/supply chain, energy and utilities, printing systems, medical imaging systems, automobiles, manufacturing processes, and computer network. He is a member of INFORMS, IIE and IEEE.