Received: 19 February 2017 | Revised: 21 September 2017 | Accepted: 6 November 2017 | Published on: 10 January 2018

DOI: 10.1002/sam.11370

ORIGINAL ARTICLE

WILEY

Sparse estimation of multivariate Poisson log-normal models from count data

Hao Wu[1,2] | **Xinwei Deng**[3] | **Naren Ramakrishnan**[2,4]

[1]Department of Electrical and Computer Engineering, Virginia Tech, Arlington, Virginia,
[2]Discovery Analytics Center, Virginia Tech, Arlington, Virginia 22203, USA
[3]Department of Statistics, Virginia Tech, Blacksburg, Virginia,
[4]Department of Computer Science, Virginia Tech, Arlington, Virginia,

Correspondence
Xinwei Deng, Department of Statistics, Virginia Tech, Blacksburg, VA 24061.
Email: xdeng@vt.edu

Funding Information
Intelligence Advanced Research Projects Activity (IARPA), D12PC000337.

Modeling data with multivariate count responses is a challenging problem because of the discrete nature of the responses. Existing methods for univariate count response cannot be easily extended to the multivariate case since the dependence among multiple responses needs to be properly accommodated. In this paper, we propose a multivariate Poisson log-normal regression model for multivariate count responses by using latent variables. By simultaneously estimating the regression coefficients and inverse covariance matrix over the latent variables with an efficient Monte Carlo EM algorithm, the proposed model takes advantage of the association among multiple count responses to improve the model prediction accuracy. Simulation studies and applications to real-world data are conducted to systematically evaluate the performance of the proposed method in comparison with conventional methods.

KEYWORDS
count data, covariance estimation, Monte Carlo method, Poisson log-normal model, sparse multivariate model

1 | INTRODUCTION

In this decade of data science, multivariate response observations are routinely encountered in various contexts. To model such datasets, multivariate regression and multitask learning models are common techniques to study and investigate relationships between $q \geq 2$ responses and p predictors. The former class of methods (eg, refs. [23,24,27,31]) estimates the $p \times q$ regression coefficients and recovers the correlation structure among response variables using regularization. The latter class of methods focuses on learning the shared features [13–15,18] or common underlying structure(s) among multiple tasks [1,2,5,21,36] using regression approaches and enforcing regularization controls over the coefficient matrix. However, all such multivariate regression or multitask learning models discussed above often deal with continuous responses and are not applicable to count responses.

When responses are count variables, the Poisson model is a natural approach to model them, for example, in domains such as influenza case count modeling [32], traffic accident analysis [7,25], and consumer services [30]. However, Poisson regression models developed in these works are either univariate or inferred via Bayesian approaches, and neither sparsity nor feature selection is typically enforced over the inferred coefficients. When count responses are multivariate, it is challenging to quantify associations among them because of the discrete nature of the data. One approach is to model each dimension of count variables as the sum of independent Poisson variables with some common Poisson variables capturing dependencies [19]. A drawback of this method is that it can only model positive correlations. Recent literature [16,35] models multivariate count data with novel Poisson graphical models that can handle both positive and negative dependencies. However, these works do not consider multivariate count data in the context of regression.

To develop a joint model for data with multivariate count responses, it is important to properly exploit potential hidden associations among the count responses. One way to consider such a joint model of multivariate count responses is via penalty-based model selection from the perspective of parameter regularization. The key idea is to allow the count responses to be independent of each other, while the regression coefficients are required to obey a certain common sparse structure. Hence joint modeling is enabled here because of the joint estimation of regression coefficients through appropriate penalties. Such a modeling strategy leads to an explicit loss

 © 2018 Wiley Periodicals, Inc.

function with tractable computational characteristics. However, this method overlooks essential dependences among multiple count responses, which could result in poor prediction performance. There are also several recent papers that develop models of multivariate count data from the lens of conditional dependency. But these methods are typically restricted to approximated likelihood functions under the framework of generalized linear models.

In this paper, we propose a novel multivariate Poisson log-normal model for data with multiple count responses. The motivation of adopting a log-normal model is to borrow strength from regression under the multivariate normal assumption, which can simultaneously estimate regression coefficients and the covariance structure. For the proposed model, the logarithm of the Poisson rate parameters is modeled as a multivariate normal with a sparse inverse covariance matrix, which combines the strengths of sparse regression and graphical model to improve prediction performance. Thus, this approach can fully exploit the conditional dependency among multiple count responses. Our key contributions to model data with multivariate counts response are as follows:

- The method proposed here combines the strengths of regression and graphical models to improve prediction for multivariate regression.
- The method proposed here also enables interpretable models in terms of the predictors by reducing the number of regression parameters via the Lasso penalty.
- A simple Monte Carlo EM (MCEM) algorithm is developed to facilitate the estimation of parameters, which allows the iterative estimation of regression coefficients by Lasso and the inverse covariance matrix by graphical Lasso.
- By applying the proposed model to real-world influenza-like-illness (ILI) datasets, we study the dependencies between different types of flu viruses in 2 Latin American countries, and demonstrate the effectiveness of the proposed method when modeling multivariate data with count responses.

It is worth pointing out that the proposed method is not restricted to using the Lasso penalty for regression parameters. It can be easily extended to other penalties such as the adaptive Lasso, group Lasso, or fused Lasso [11]. While covariance matrix estimation and inverse covariance matrix estimation have attracted significant attention in the literature [10,24,27], here we use this idea in the context of a multivariate regression for count data. Thus inverse covariance matrix estimation is conducted here to improve prediction performance, not just as an unsupervised procedure. One may call such a strategy *supervised covariance estimation*, which has not been widely studied in the literature. (One exception is the work on multivariate regression for continuous responses [27,34].) Therefore, to the best of our knowledge, our proposed method is a first work to incorporate covariance matrix estimation into a multivariate regression model of count responses.

The rest of the paper is organized as follows. Section 2 elaborates the proposed method. Section 3 focuses on developing an efficient MCEM algorithm for parameter estimation. The simulation study is conducted in Section 4, and a real case study is covered in Section 5. Some discussion of related work is given in Section 6. We conclude the work in Section 7.

2 | MULTIVARIATE POISSON LOG-NORMAL MODEL

In this section, we formally specify the multivariate Poisson log-normal (MVPLN) model, and describe the proposed MCEM algorithm for parameter estimation in detail. For convenience, we use the following notation to present the proposed MVPLN model in the rest of the paper. Normal lower case letters, for example, x and y, represent scalars, while bold lower case letters, for example, $\boldsymbol{x}$ and $\boldsymbol{y}$, are used to represent column vectors, and bold upper case letters in the calligraphic font, for example, $\boldsymbol{\mathcal{X}}$ and $\boldsymbol{\mathcal{Y}}$, denote random column vectors. Let letters with superscript in parentheses, for example, $x^{(i)}$, denote the ith component of the corresponding vector $\boldsymbol{X}$. Matrices are represented by bold upper case letters in normal font, for example, $\boldsymbol{X}$ and $\boldsymbol{Y}$. Letters in lower case with 2 subscripts, for example, $x_{i,j}$, denote the (i,j)th entry of the corresponding matrix $\boldsymbol{X}$.

2.1 | The proposed model

Consider the multivariate random variable $\boldsymbol{\mathcal{Y}} = [\mathcal{Y}^{(1)}, \mathcal{Y}^{(2)}, \ldots, \mathcal{Y}^{(q)}]^T \in \mathcal{Z}_+^q$, where the superscript T denotes the transpose, and $\mathcal{Z}_+$ represents the set of all positive integers. For count data, it is reasonable to make the assumption that $\boldsymbol{\mathcal{Y}}$ follows the multivariate Poisson distribution. Without loss of generality, let us assume that each dimension of $\boldsymbol{\mathcal{Y}}$, say $\mathcal{Y}^{(i)}$, follows the univariate Poisson distribution with parameter $\theta^{(i)}$, and is conditionally independent of other dimensions given $\theta^{(i)}$. That is

$$\mathcal{Y}^{(i)} \sim \text{Poisson}(\theta^{(i)}), \theta^{(i)} \in \mathcal{R}_+, \forall i = 1, 2, \ldots, q. \quad (1)$$

Let $\boldsymbol{x} = [x^{(1)}, x^{(2)}, \ldots, x^{(p)}]^T \in \mathcal{R}^p$ denote the predictor vector. In order to establish relationship between $\boldsymbol{\mathcal{Y}}$ and $\boldsymbol{x}$, we consider the following regression model:

$$\begin{aligned} \boldsymbol{\theta} &= \exp(\boldsymbol{B}^T x + \varepsilon), \\ \varepsilon &\sim N(0, \boldsymbol{\Sigma}), \end{aligned} \quad (2)$$

where $\boldsymbol{B}$ is a $p \times q$ coefficient matrix, and $\boldsymbol{\Sigma}$ is the $q \times q$ covariance matrix which captures the covariance structure of variable $\boldsymbol{\theta} = [\theta^{(1)}, \theta^{(2)}, \ldots, \theta^{(q)}]^T$ given $\boldsymbol{x}$. Notice that the exponential function in Equation (2) is evaluated element-wise. Through the variable $\boldsymbol{\theta}$, we model the

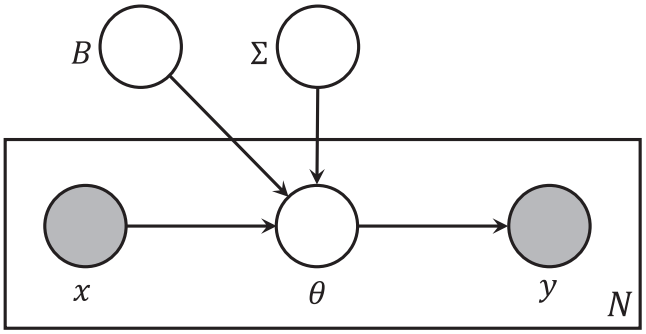

FIGURE 1 Plate notation of the proposed multivariate Poisson log-normal model

covariance structure of the count variable $\mathcal{Y}$ indirectly. Figure 1 shows the plate notation of the proposed MVPLN model.

With the conditional independence assumption, the probability mass function for the multivariate Poisson random variable $\mathcal{Y}$ is

$$p(\mathcal{Y}=\boldsymbol{y}\mid\boldsymbol{\theta})=\prod_{i=1}^{q}p(\mathcal{Y}^{(i)}=y^{(i)}\mid\theta^{(i)})=\prod_{i=1}^{q}\frac{(\theta^{(i)})^{y^{(i)}}\exp(-\theta^{(i)})}{y^{(i)}!}. \tag{3}$$

From the specification of the MVPLN model in Equation (2), since $\boldsymbol{\varepsilon}\sim N(0,\boldsymbol{\Sigma})$, if we let $\boldsymbol{\gamma}=\boldsymbol{B}^T\boldsymbol{x}+\boldsymbol{\varepsilon}$, we know that $\boldsymbol{\gamma}$ follows the multivariate normal distribution $N(\boldsymbol{B}^T\boldsymbol{x},\boldsymbol{\Sigma})$ with density function

$$p(\boldsymbol{\gamma}\mid\boldsymbol{x})=\frac{1}{(2\pi)^{q/2}|\boldsymbol{\Sigma}|^{1/2}}\exp\left(-\frac{1}{2}(\boldsymbol{\gamma}-\boldsymbol{B}^T\boldsymbol{x})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{\gamma}-\boldsymbol{B}^T\boldsymbol{x})\right).$$

Since $\boldsymbol{\theta}=\exp(\boldsymbol{\gamma})=\exp(\boldsymbol{B}^T\boldsymbol{x}+\boldsymbol{\varepsilon})$, $\boldsymbol{\theta}|\boldsymbol{x}$ follows the multivariate log-normal distribution, and we can derive that the density function of $\boldsymbol{\theta}|\boldsymbol{x}$ is

$$p(\boldsymbol{\theta}\mid\boldsymbol{x})=p_{\boldsymbol{\gamma}}(\log(\boldsymbol{\theta})\mid\boldsymbol{x})\left|diag\left(\frac{1}{\theta^{(i)}}\right)\right|=\frac{\exp\left(-\frac{1}{2}(\log\boldsymbol{\theta}-\boldsymbol{B}^T\boldsymbol{x})^T\boldsymbol{\Sigma}^{-1}(\log\boldsymbol{\theta}-\boldsymbol{B}^T\boldsymbol{x})\right)}{(2\pi)^{q/2}|\boldsymbol{\Sigma}|^{1/2}\prod_{i=1}^{q}\theta^{(i)}}. \tag{4}$$

Given n observations of the predictor $\boldsymbol{X}=[\boldsymbol{x}_1,\boldsymbol{x}_2,\ldots,\boldsymbol{x}_n]^T$ and corresponding responses $\boldsymbol{Y}=[\boldsymbol{y}_1,\boldsymbol{y}_2,\ldots,\boldsymbol{y}_n]^T$, the log-likelihood of the MVPLN model is

$$\mathcal{L}(\boldsymbol{B},\boldsymbol{\Sigma})=\sum_{j=1}^{n}\log p(\mathcal{Y}=\boldsymbol{y}_j\mid\boldsymbol{x}_j), \tag{5}$$

where

$$p(\mathcal{Y}=\boldsymbol{y}\mid\boldsymbol{x})=\int_{\boldsymbol{\theta}}p(\mathcal{Y}=\boldsymbol{y},\boldsymbol{\theta}\mid\boldsymbol{x})d\boldsymbol{\theta}=\int_{\boldsymbol{\theta}}p(\mathcal{Y}=\boldsymbol{y}\mid\boldsymbol{\theta})p(\boldsymbol{\theta}\mid\boldsymbol{x})d\boldsymbol{\theta}. \tag{6}$$

Here, $p(\mathcal{Y}=\mathbf{y}\mid\boldsymbol{\theta})$ and $p(\boldsymbol{\theta}|\boldsymbol{X})$ follow multivariate Poisson distribution and multivariate log-normal distribution specified in Equations (3) and (4), respectively. To jointly infer the sparse estimations of the coefficient matrix $\boldsymbol{B}$ and the inverse covariance matrix $\boldsymbol{\Sigma}^{-1}$, we adopt the regularized negative log-likelihood function with l_1 penalties as our loss function. Log-likelihood has been a widely used objective function when estimating regression models, and it has been shown in many research works [29] that the l_1 penalties are able to enforce and recover sparse structures in the model and also help to prevent overfitting. By combining these 2 ingredients together to form the loss function of the proposed models, it achieves our goal of sparse estimations of model parameters $\boldsymbol{B}$ and $\boldsymbol{\Sigma}$. To be specific, the loss function could be written as

$$\mathcal{L}_p(\boldsymbol{B},\boldsymbol{\Sigma})=-\mathcal{L}(\boldsymbol{B},\boldsymbol{\Sigma})+\lambda_1\|\boldsymbol{B}\|_1+\lambda_2\|\boldsymbol{\Sigma}^{-1}\|_1, \tag{7}$$

where $\|\cdot\|_1$ denote the l_1 matrix norm which is defined as $\|\boldsymbol{B}\|_1=\sum_{i,j}\mid b_{ij}\mid$, and $\lambda_1>0$, $\lambda_2>0$ are 2 tuning parameters.

2.2 | Selection of tuning parameters

To determine the optimal values of the tuning parameters λ_1 and λ_2, we adopt the extended Bayesian information criterion (EBIC) approach proposed by Chen and Chen [4] and extended to Gaussian graphical models by Foygel and Drton [8]. Assume $\boldsymbol{B}_{\lambda_1,\lambda_2}$ and $\boldsymbol{\Omega}_{\lambda_1,\lambda_2}$ denote the maximum likelihood estimation (MLE) of the model parameters $\boldsymbol{B}$ and $\boldsymbol{\Omega}$ with regularization parameters λ_1 and λ_2, where $\boldsymbol{\Omega}=\boldsymbol{\Sigma}^{-1}$ represents the inverse of the covariance matrix. The EBIC value for this model is given by the following equation:

$$\begin{aligned}\text{EBIC}_\gamma(\lambda_1,\lambda_2)=&-2\tilde{Q}(\boldsymbol{B}_{\lambda_1,\lambda_2},\boldsymbol{\Omega}_{\lambda_1,\lambda_2})+[v(\boldsymbol{B}_{\lambda_1,\lambda_2})\\&+v(\boldsymbol{\Omega}_{\lambda_1,\lambda_2})]\log n+2\gamma v(\boldsymbol{B}_{\lambda_1,\lambda_2})\log(pq)\\&+4\gamma v(\boldsymbol{\Omega}_{\lambda_1,\lambda_2})\log q,\end{aligned} \tag{8}$$

where $\tilde{\boldsymbol{Q}}(\boldsymbol{B}_{\lambda_1,\lambda_2},\boldsymbol{\Omega}_{\lambda_1,\lambda_2})$ is the approximate expected log-likelihood in Equation (13); $v(\boldsymbol{B}_{\lambda_1,\lambda_2})$ and $v(\boldsymbol{\Omega}_{\lambda_1,\lambda_2})$ denote the number of nonzero entries in $\boldsymbol{B}_{\lambda_1,\lambda_2}$ and $\boldsymbol{\Omega}_{\lambda_1,\lambda_2}$, respectively; and n is the number of training observations. With EBIC, the optimal values for λ_1 and λ_2 are selected by

$$(\widehat{\lambda}_1,\widehat{\lambda}_2)=\underset{\lambda_1,\lambda_2}{\text{argmin}}\ \text{EBIC}_\gamma(\lambda_1,\lambda_2).$$

3 | MONTE CARLO EM ALGORITHM FOR PARAMETER ESTIMATION

In order to obtain the estimates of MVPLN model parameters $\boldsymbol{B}$ and $\boldsymbol{\Sigma}$, we could simply solve the following optimization problem:

$$\widehat{\boldsymbol{B}},\widehat{\boldsymbol{\Sigma}}=\underset{\boldsymbol{B},\boldsymbol{\Sigma}}{\text{argmin}}\ \mathcal{L}_p(\boldsymbol{B},\boldsymbol{\Sigma}). \tag{9}$$

However, it is difficult to directly minimize the objective function defined above because of the complicated integral in Equation (6). Thus, we turn to an iterative approach for the solution. We treat $\boldsymbol{\theta}$ as latent random variables, and apply the EM algorithm to obtain the maximum likelihood parameter estimates (MLEs).

3.1 | E-step

Suppose at iteration $t+1$, the conditional distribution of the latent variable $\boldsymbol{\theta}$ given $\boldsymbol{X}, \boldsymbol{y}, \boldsymbol{B}^{(t)}$, and $\boldsymbol{\Sigma}^{(t)}$ is

$$p(\boldsymbol{\theta} \mid \boldsymbol{\mathcal{Y}} = \boldsymbol{y}, \boldsymbol{x}; \boldsymbol{B}^{(t)}, \boldsymbol{\Sigma}^{(t)}) = \frac{p(\boldsymbol{\mathcal{Y}} = \boldsymbol{y}, \boldsymbol{\theta} \mid \boldsymbol{x}; \boldsymbol{B}^{(t)}, \boldsymbol{\Sigma}^{(t)})}{p(\boldsymbol{\mathcal{Y}} = \boldsymbol{y} \mid \boldsymbol{x}; \boldsymbol{B}^{(t)}, \boldsymbol{\Sigma}^{(t)})}. \quad (10)$$

Here, $\boldsymbol{B}^{(t)}$ and $\Sigma^{(t)}$ represent the current estimates of the matrices $\boldsymbol{B}$ and Σ at the iteration t of the EM algorithm, respectively. Then, the expected log-likelihood would be

$$\begin{aligned} Q(\boldsymbol{B}, \boldsymbol{\Sigma} \mid \boldsymbol{B}^{(t)}, \boldsymbol{\Sigma}^{(t)}) &= E_{p(\boldsymbol{\theta} \mid \boldsymbol{\mathcal{Y}} = \boldsymbol{y}, \boldsymbol{x})}[\mathcal{L}(\boldsymbol{B}, \boldsymbol{\Sigma})] \\ &= \sum_{j=1}^{n} E_{p(\boldsymbol{\theta}_j \mid \boldsymbol{\mathcal{Y}} = \boldsymbol{y}_j, \boldsymbol{x}_j)}[\log p(\boldsymbol{\mathcal{Y}} = \boldsymbol{y}_j, \boldsymbol{\theta}_j \mid \boldsymbol{x}_j; \boldsymbol{B}, \boldsymbol{\Sigma})]. \quad (11) \end{aligned}$$

3.2 | M-step

Find the estimations for the parameters $\boldsymbol{B}$ and $\boldsymbol{\Sigma}$ at iteration $t+1$ by solving the following optimization problem:

$$\begin{aligned} \boldsymbol{B}^{(t+1)}, \boldsymbol{\Sigma}^{(t+1)} = \underset{\boldsymbol{B}, \boldsymbol{\Sigma}}{\operatorname{argmin}} \big\{ &-Q\left(\boldsymbol{B}, \boldsymbol{\Sigma} \mid \boldsymbol{B}^{(t)}, \boldsymbol{\Sigma}^{(t)}\right) \\ &+\lambda_1 \|\boldsymbol{B}\|_1 + \lambda_2 \|\boldsymbol{\Sigma}^{-1}\|_1 \big\}. \quad (12) \end{aligned}$$

The E-step and M-step are repeated alternatively until convergence.

However, it is difficult to directly derive the analytical form of the expected log-likelihood of the model because of the integral in Equation (6) and thus to solve the corresponding optimization problem in the M-step. Here, we adopt a Monte Carlo variant of the EM algorithm for an approximate solution where MCMC techniques are applied in the E-step to obtain an approximate expected log-likelihood function. Then, in the M-step, the MLE of the model parameters is obtained by maximizing the penalized approximate expected log-likelihood.

In our implementation, we take 300 MC samples in the MC E-step and discard first 10% of samples since we observe that with the tailored proposal distribution, the MCMC procedure usually begins to converge within tens of iterations. For the EM algorithm, we consider that it converges if the average element-wise changes of matrix $\boldsymbol{B}$ and $\boldsymbol{\Sigma}^{-1}$ are both within a given epsilon, or it reaches the maximum allowed number of iterations, for example, max_iters. We set max_iters to be 100 in the implementation.

3.3 | Monte Carlo E-step

In the iteration $t+1$ of the MC E-step, instead of trying to derive the closed form of the conditional probability distribution of $\boldsymbol{\theta}_j$, we draw m random samples of $\boldsymbol{\theta}_j$, say $\boldsymbol{\Theta}_j = [\boldsymbol{\theta}_j^{(1)}, \boldsymbol{\theta}_j^{(2)}, \ldots, \boldsymbol{\theta}_j^{(m)}]^T$, from $p(\boldsymbol{\theta}_j \mid \boldsymbol{\mathcal{Y}} = \boldsymbol{y}_j, \boldsymbol{x}_j; \boldsymbol{B}^{(t)}, \boldsymbol{\Sigma}^{(t)})$, and approximate the expected log-likelihood function with

$$\begin{aligned} &\widetilde{Q}(\boldsymbol{B}, \boldsymbol{\Sigma} \mid \boldsymbol{B}^{(t)}, \boldsymbol{\Sigma}^{(t)}) \\ &\quad = \sum_{j=1}^{n} \frac{1}{m} \sum_{\tau=1}^{m} \log p(\boldsymbol{\mathcal{Y}} = \boldsymbol{y}_j, \boldsymbol{\theta}_j^{(\tau)} \mid \boldsymbol{x}_j; \boldsymbol{B}^{(t)}, \boldsymbol{\Sigma}^{(t)}). \quad (13) \end{aligned}$$

Drawing random samples of $\boldsymbol{\theta}_j$ can be achieved with the Metropolis Hasting algorithm. In iteration $t+1$ of the MC E-step, $\boldsymbol{y}_j, \boldsymbol{x}_j, \boldsymbol{B}^{(t)}$, and $\boldsymbol{\Sigma}^{(t)}$ are all known values, which makes $p(\boldsymbol{\mathcal{Y}} = \boldsymbol{y}_j \mid \boldsymbol{x}_j; \boldsymbol{B}^{(t)}, \boldsymbol{\Sigma}^{(t)})$ a constant. In this case, Equation (10) yields

$$p(\boldsymbol{\theta}_j \mid \boldsymbol{\mathcal{Y}} = \boldsymbol{y}_j, \boldsymbol{x}_j; \boldsymbol{B}^{(t)}, \boldsymbol{\Sigma}^{(t)}) \propto p(\boldsymbol{\mathcal{Y}} = \boldsymbol{y}_j, \boldsymbol{\theta}_j \mid \boldsymbol{x}_j; \boldsymbol{B}^{(t)}, \boldsymbol{\Sigma}^{(t)}).$$

Let $f(\boldsymbol{\theta}_j) = p(\boldsymbol{\mathcal{Y}} = \boldsymbol{y}_j, \boldsymbol{\theta}_j \mid \boldsymbol{x}_j; \boldsymbol{B}^{(t)}, \boldsymbol{\Sigma}^{(t)})$, and $g(\boldsymbol{\theta}^* | \boldsymbol{\theta})$ be the density function of the proposal distribution. Algorithm 1 illustrates the Metropolis Hasting algorithm for sampling $\boldsymbol{\theta}_j$ from $p(\boldsymbol{\theta}_j \mid \boldsymbol{\mathcal{Y}} = \boldsymbol{y}_j, \boldsymbol{x}_j; \boldsymbol{B}^{(t)}, \boldsymbol{\Sigma}^{(t)})$.

Algorithm 1: Metropolis algorithm for sampling $\boldsymbol{\theta}_j$

input : $\boldsymbol{y}_j, \boldsymbol{x}_j, \boldsymbol{B}^{(t)}$ and $\boldsymbol{\Sigma}^{(t)}$.

output: m samples $\boldsymbol{\Theta}_j = \left\{\boldsymbol{\theta}_j^{(1)}, \boldsymbol{\theta}_j^{(2)}, \ldots, \boldsymbol{\theta}_j^{(m)}\right\}^T$.

Choose $\boldsymbol{\theta}_j^{(0)}$ as initial value, and let $\tau \leftarrow 1$;
while $|\boldsymbol{\Theta}_j| < m$ **do**
  Draw a candidate $\boldsymbol{\theta}_j^*$ from $g(\boldsymbol{\theta}_j^* \mid \boldsymbol{\theta}_j^{(\tau-1)})$;
  $\alpha \leftarrow \min\left(\frac{f(\boldsymbol{\theta}_j^*)/g(\boldsymbol{\theta}_j^* | \boldsymbol{\theta}_j^{(\tau-1)})}{f(\boldsymbol{\theta}_j^{(\tau-1)})/g(\boldsymbol{\theta}_j^{(\tau-1)} | \boldsymbol{\theta}_j^*)}, 1\right)$;
  Accept $\boldsymbol{\theta}_j^{(*)}$ as $\boldsymbol{\theta}_j^{(\tau)}$ with probability α;
  if $\boldsymbol{\theta}_j^{(*)}$ *is accepted* **then**
    $\boldsymbol{\Theta}_j \leftarrow \boldsymbol{\Theta}_j \cup \{\boldsymbol{\theta}_j^{(\tau)}\}$;
    $\tau \leftarrow \tau + 1$;
  end
end
return $\boldsymbol{\Theta}_j$;

In order to reduce the burn-in period of the Metropolis Hasting algorithm, we adopt the tailored normal distribution [6] as our proposal distribution. The initial value of the location parameter for the tailored normal distribution should be the mode of $p(\boldsymbol{\theta}_j \mid \boldsymbol{\mathcal{Y}} = \boldsymbol{y}_j, \boldsymbol{x}_j; \boldsymbol{B}^{(t)}, \boldsymbol{\Sigma}^{(t)})$, which is

$$\boldsymbol{\theta}_j^{(0)} = \underset{\boldsymbol{\theta}_j}{\operatorname{argmax}} \{\log f(\boldsymbol{\theta}_j)\}, \quad (14)$$

and the covariance matrix is $\tau(-\boldsymbol{H}(\boldsymbol{\theta}_j^{(0)}))^{-1}$, where $\boldsymbol{H}(\boldsymbol{\theta}_j^{(0)})$ denotes the Hessian matrix of $\log f(\boldsymbol{\theta}_j)$ evaluated at $\boldsymbol{\theta}_j^{(0)}$, and τ is a tuning parameter. To solve the optimization problem in Equation (14), let

$$\begin{aligned} F(\boldsymbol{\theta}_j) = \log f(\boldsymbol{\theta}_j) = \log \big(&p(\boldsymbol{\mathcal{Y}} = \boldsymbol{y}_j \mid \boldsymbol{\theta}_j, \boldsymbol{x}_j; \boldsymbol{B}^{(t)}, \boldsymbol{\Sigma}^{(t)}) \\ &\times p(\boldsymbol{\theta}_j \mid \boldsymbol{x}_j; \boldsymbol{B}^{(t)}, \boldsymbol{\Sigma}^{(t)})\big). \end{aligned}$$

By combining Equations (3) and (4), we can derive that

$$\begin{aligned} F(\boldsymbol{\theta}_j) = (\boldsymbol{y}_j - \mathbf{1})^T \log \boldsymbol{\theta}_j - \frac{1}{2}(\log \boldsymbol{\theta}_j - \boldsymbol{B}^{(t)T} \boldsymbol{x}_j)^T \boldsymbol{\Sigma}^{(t)-1} \\ \times \log \boldsymbol{\theta}_j - \boldsymbol{B}^{(t)T} \boldsymbol{x}_j) - \mathbf{1}^T \boldsymbol{\theta}_j + C, \quad (15) \end{aligned}$$

where $\mathbf{1}$ denotes a column vector of 1's, and C represents the sum of all the constants in $\log f(\boldsymbol{\theta}_j)$. Then, the first-order and

second-order derivatives of $F(\boldsymbol{\theta}_j)$ w.r.t. $\boldsymbol{\theta}_j$ are

$$\nabla F(\boldsymbol{\theta}_j) = \frac{dF(\boldsymbol{\theta}_j)}{d\boldsymbol{\theta}_j} = diag\left(\frac{1}{\theta_j^{(i)}}\right) \times \left[(\boldsymbol{y}_j - \mathbf{1}) - \boldsymbol{\Sigma}^{(t)-1}(\log\boldsymbol{\theta}_j - \boldsymbol{B}^{(t)T}\boldsymbol{x}_j)\right] - \mathbf{1}, \tag{16}$$

$$\boldsymbol{H}(\boldsymbol{\theta}_j) = diag\left(-\frac{y_j^{(i)} - 1}{\theta_j^{(i)^2}}\right) + diag\left(-\frac{1}{\theta_j^{(i)^2}}\right) \times diag(\boldsymbol{\Sigma}^{(t)-1}(\log\boldsymbol{\theta}_j - \boldsymbol{B}^{(t)T}\boldsymbol{x}_j)) + diag\left(\frac{1}{\theta_j^{(i)}}\right)\boldsymbol{\Sigma}^{(t)-1}diag\left(\frac{1}{\theta_j^{(i)}}\right). \tag{17}$$

Letting $\nabla F(\boldsymbol{\theta}_j) = 0$, we could get that the initial value $\boldsymbol{\theta}_j^{(0)}$ of the location parameter for the tailored normal distribution is the solution to the following equation:

$$\boldsymbol{\theta}_j + \boldsymbol{\Sigma}^{(t)-1}\log\boldsymbol{\theta}_j = \boldsymbol{y}_j - \mathbf{1} + \boldsymbol{\Sigma}^{(t)-1}\boldsymbol{B}^{(t)T}\boldsymbol{x}_j, \tag{18}$$

which can be solved by any numerical root discovery algorithms. However, taking the performance issue into account, we let $\boldsymbol{\kappa}_j = \log\boldsymbol{\theta}_j$, and adopt a linear approximation to $e^{\boldsymbol{\kappa}_j}$ with its first-order Taylor expansion at $\boldsymbol{\kappa}_j^{(0)} = \log\boldsymbol{y}_j$. In this case, Equation (18) becomes

$$e^{\boldsymbol{\kappa}_j^{(0)}} + diag\left(e^{\boldsymbol{\kappa}_j^{(0)}}\right)\left(\boldsymbol{\kappa}_j - \boldsymbol{\kappa}_j^{(0)}\right) + \boldsymbol{\Sigma}^{(t)-1}\boldsymbol{\kappa}_j = \boldsymbol{y}_j - \mathbf{1} + \boldsymbol{\Sigma}^{(t)-1}\boldsymbol{B}^{(t)T}\boldsymbol{x}_j. \tag{19}$$

By solving Equation (19) for $\boldsymbol{\kappa}_j$, the location parameter (mean) $\boldsymbol{\theta}_j$ of the tailored normal distribution is given by $\boldsymbol{\theta}_j^{(0)} = e^{\boldsymbol{\kappa}_j}$, where

$$\boldsymbol{\kappa}_j = \left(diag\left(e^{\boldsymbol{\kappa}_j^{(0)}}\right) + \boldsymbol{\Sigma}^{(t)-1}\right)^{-1} \times \left(\boldsymbol{y}_j - \mathbf{1} + \boldsymbol{\Sigma}^{(t)-1}\boldsymbol{B}^{(t)T}\boldsymbol{x}_j + diag(e^{\boldsymbol{\kappa}_j^{(0)}})\boldsymbol{\kappa}_j^{(0)} - e^{\boldsymbol{\kappa}_j^{(0)}}\right),$$

and the covariance matrix is given by $\tau(-\boldsymbol{H}(\boldsymbol{\theta}_j^{(0)}))^{-1}$. In case the covariance matrix $\tau(-\boldsymbol{H}(\boldsymbol{\theta}_j^{(0)}))^{-1}$ is not positive semidefinite, the nearest positive semidefinite matrix to $\tau(-\boldsymbol{H}(\boldsymbol{\theta}_j^{(0)}))^{-1}$ is used instead [17].

3.4 | M-step: Maximize the approximate penalized expected log-likelihood

With the MC approximation, we can reformulate the approximate expected log-likelihood as

$$\tilde{Q}(\boldsymbol{B}, \boldsymbol{\Sigma} \mid \boldsymbol{B}^{(t)}, \boldsymbol{\Sigma}^{(t)}) = -\frac{1}{n}\sum_{j=1}^{n}\frac{1}{m}\sum_{\tau=1}^{m} \times \left[\left(\log\boldsymbol{\theta}_j^{(\tau)} - \boldsymbol{B}^T\boldsymbol{x}_j\right)^T\boldsymbol{\Omega}\left(\log\boldsymbol{\theta}_j^{(\tau)} - \boldsymbol{B}^T\boldsymbol{x}_j\right) - \log|\boldsymbol{\Omega}|\right], \tag{20}$$

where $\boldsymbol{\Omega} = \boldsymbol{\Sigma}^{-1}$. If we further let $\boldsymbol{\varphi}_{\tau,j} = (\log\boldsymbol{\theta}_j^{(\tau)} - \boldsymbol{B}^T\boldsymbol{x}_j)$, the optimization problem we need to solve in the M-step of the MCEM algorithm can be written as

$$\boldsymbol{B}^{(t+1)}, \boldsymbol{\Sigma}^{(t+1)} = \underset{\boldsymbol{B},\boldsymbol{\Omega}}{\operatorname{argmin}}\left\{\frac{1}{mn}\operatorname{tr}(\boldsymbol{\Phi}^T\boldsymbol{\Phi}\boldsymbol{\Omega}) - \log|\boldsymbol{\Omega}| + \lambda_1\|\boldsymbol{B}\|_1 + \lambda_2\|\boldsymbol{\Omega}\|_1\right\}, \tag{21}$$

where $\boldsymbol{\Phi} = [\boldsymbol{\varphi}_{1,1}, \boldsymbol{\varphi}_{2,1}, \ldots, \boldsymbol{\varphi}_{m,1}, \boldsymbol{\varphi}_{1,2}, \boldsymbol{\varphi}_{2,2}, \ldots, \boldsymbol{\varphi}_{m,2}, \ldots, \boldsymbol{\varphi}_{m,n}]^T$. The optimization problem defined in Equation (21) is not convex. However, it is convex w.r.t. either $\boldsymbol{B}$ or $\boldsymbol{\Omega}$ with the other fixed [27]. Thus, we present an iterative algorithm that optimizes the objective function in Equation (21) alternatively w.r.t. $\boldsymbol{B}$ and $\boldsymbol{\Omega}$.

With $\boldsymbol{B}$ fixed at $\boldsymbol{B}_0$, the optimization problem in Equation (21) yields

$$\boldsymbol{\Omega}(\boldsymbol{B}_0) = \underset{\boldsymbol{\Omega}}{\operatorname{argmin}}\left\{\frac{1}{mn}\operatorname{tr}(\boldsymbol{\Phi}^T\boldsymbol{\Phi}\boldsymbol{\Omega}) - \log|\boldsymbol{\Omega}| + \lambda_2\|\boldsymbol{\Omega}\|_1\right\}, \tag{22}$$

which is similar to the problem studied by Friedman et al. [10]. We solve this problem with the graphical Lasso approach.

When $\boldsymbol{\Omega}$ is fixed at $\boldsymbol{\Omega}_0$, we have the following optimization problem:

$$\boldsymbol{B}(\boldsymbol{\Omega}_0) = \underset{\boldsymbol{B}}{\operatorname{argmin}}\left\{\frac{1}{mn}\operatorname{tr}(\boldsymbol{\Phi}^T\boldsymbol{\Phi}\boldsymbol{\Omega}_0) + \lambda_1\|\boldsymbol{B}\|_1\right\}, \tag{23}$$

which is similar to the problem solved by Lasso, and we could adopt the cyclical coordinate descent algorithm [9] to obtain the estimate of $\boldsymbol{B}$. However, considering the computational burden already brought in by the MCMC approximation in the MC E-step, we propose an approach to solve the optimization problem in Equation (23) with a quadratic approximation to the l_1 matrix norm $\|\boldsymbol{B}\|_1$.

Let $\widehat{\boldsymbol{B}}$ denote the current estimation of $\boldsymbol{B}$, and $1/\sqrt{|\widehat{\boldsymbol{B}}|}$ represent the matrix in which each entry is the inverse of the square root of the absolute value of the corresponding entry in $\widehat{\boldsymbol{B}}$. Then, the l_1 matrix norm penalty in Equation (23) could be approximated with the following approach:

$$\lambda_1\|\boldsymbol{B}\|_1 \approx \lambda_1\operatorname{tr}(\boldsymbol{B}'^T\boldsymbol{B}'), \quad \text{where} \quad \boldsymbol{B}' = \boldsymbol{B}\circ\frac{1}{\sqrt{|\widehat{\boldsymbol{B}}|}}.$$

Here, $\circ$ denotes the Hadamard (element-wise) product. If we write $\boldsymbol{\Phi}$ into the block matrix

$$\boldsymbol{\Phi} = \begin{bmatrix}\log\boldsymbol{\Theta}_1 - \boldsymbol{x}_1\boldsymbol{B}\\ \log\boldsymbol{\Theta}_2 - \boldsymbol{x}_2\boldsymbol{B}\\ \vdots\\ \log\boldsymbol{\Theta}_n - \boldsymbol{x}_n\boldsymbol{B}\end{bmatrix},$$

where $\boldsymbol{X}_j$ is $m \times p$ matrix with each row being $\boldsymbol{x}_j$ for all $j = 1, 2, \ldots, n$, the objective function of the optimization problem in Equation (23) can be written as

$$\eta(\boldsymbol{B}) = \lambda_1\operatorname{tr}(\boldsymbol{B}'^T\boldsymbol{B}') + \frac{1}{mn}\sum_{j=1}^{n} \times \operatorname{tr}((\log\boldsymbol{\Theta}_j - \boldsymbol{x}_j\boldsymbol{B})^T(\log\boldsymbol{\Theta}_j - \boldsymbol{x}_j\boldsymbol{B})\boldsymbol{\Omega}_0). \tag{24}$$

Taking the first-order derivative of $\eta(\boldsymbol{B})$ w.r.t. $\boldsymbol{B}$ and setting it to zero, we have

$$\left(\sum_{j=1}^{n} \boldsymbol{X}_j^T \boldsymbol{X}_j\right) \boldsymbol{B}\boldsymbol{\Omega}_0 + \boldsymbol{B} \circ \frac{\lambda_1 mn}{|\widehat{\boldsymbol{B}}|} = \left(\sum_{j=1}^{n} \boldsymbol{X}_j^T (\log \boldsymbol{\Theta}_j)\right) \boldsymbol{\Omega}_0. \tag{25}$$

If we let $\left(\sum_{j=1}^{n} \boldsymbol{X}_j^T (\log \boldsymbol{\Theta}_j)\right) \boldsymbol{\Omega}_0 = \boldsymbol{H}$ and $\sum_{j=1}^{n} \boldsymbol{X}_j^T \boldsymbol{X}_j = \boldsymbol{S}$, and apply the matrix vectorization operator vec$(\cdot)$ to both sides of Equation (25), we have

$$(\boldsymbol{\Omega}_0^T \otimes \boldsymbol{S})\text{vec}(\boldsymbol{B}) + \text{vec}\left(\frac{\lambda_1 mn}{|\widehat{\boldsymbol{B}}|}\right) \circ \text{vec}(\boldsymbol{B}) = \text{vec}(\boldsymbol{H}).$$

Here, $\otimes$ represents the Kronecker product. By pulling vec$(\boldsymbol{B})$ out from the left-hand side of the above equation, we can get the solution to the optimization problem in Equation (23) as

$$\text{vec}(\boldsymbol{B}) = \left[\boldsymbol{\Omega}_0 \otimes \boldsymbol{S} + diag\left(\text{vec}\left(\frac{\lambda_1 mn}{|\boldsymbol{B}|}\right)\right)\right]^{-1} \text{vec}(\boldsymbol{H}), \tag{26}$$

and the estimated coefficient matrix $\boldsymbol{B}$ can be obtained by reorganizing the vec$(\boldsymbol{B})$ in the above equation.

By solving $\boldsymbol{B}$ and $\boldsymbol{\Omega}$ alternatively with the other fixed at the value of the last estimate until convergence, we can obtain the MLE of the coefficient matrix $\boldsymbol{B}$ and inverse covariance matrix $\boldsymbol{\Omega}$ for the current iteration of MCEM algorithm. Algorithm 2 summarizes the M-step of the MCEM algorithm.

Algorithm 2: M-step of the MCEM algorithm

input : $\boldsymbol{X}, \{\boldsymbol{\Theta}_j\}, \boldsymbol{\Omega}_0, \boldsymbol{B}_0, \lambda_1$ and λ_2.
output: MLE of $\boldsymbol{B}$ and $\boldsymbol{\Omega}$.

$t \leftarrow -1$;
repeat
  $t \leftarrow t + 1$;
  $\boldsymbol{\Phi} \leftarrow \begin{bmatrix} \log \boldsymbol{\Theta}_1 - \boldsymbol{X}_1 \boldsymbol{B}^{(t)} \\ \log \boldsymbol{\Theta}_2 - \boldsymbol{X}_2 \boldsymbol{B}^{(t)} \\ \vdots \\ \log \boldsymbol{\Theta}_n - \boldsymbol{X}_n \boldsymbol{B}^{(t)} \end{bmatrix}$;
  $\boldsymbol{\Omega}^{(t+1)} \leftarrow$ `Graphical_Lasso`$(\boldsymbol{\Phi}, \lambda_2)$;
  $\boldsymbol{S} \leftarrow \sum_{j=1}^{n} \boldsymbol{X}_j^T \boldsymbol{X}_j$;
  $\boldsymbol{H} \leftarrow \sum_{j=1}^{n} \boldsymbol{X}_j^T (\log \boldsymbol{\Theta}_j) \boldsymbol{\Omega}^{(t+1)}$;
  $\boldsymbol{B}^{(t+1)} \leftarrow \left[\boldsymbol{\Omega}^{(t+1)} \otimes \boldsymbol{S} + \text{diag}\left(\text{vec}\left(\frac{\lambda_1 mn}{|\boldsymbol{B}^{(t)}|}\right)\right)\right]^{-1} \text{vec}(\boldsymbol{H})$;
until *convergence*;
return $(\boldsymbol{B}^{(t+1)}, \boldsymbol{\Omega}^{(t+1)})$;

4 | SIMULATION STUDY

In this simulation study, we compare the proposed MVPLN model with a univariate Lasso regularized Poisson regression model (GLMNET model) (eg, as implemented in the R glmnet package [12]). (The regularized univariate Poisson regression is applied to each response dimension.) Note that the EBIC was proposed to address the inconsistency issue of BIC when the parameter space is large. When the parameter space is relatively small, both BIC and EBIC are consistent and would produce similar results [4]. For convenience, BIC is used to select the regularization parameters for the GLMNET model since the parameter space for the univariate Poisson regression is not very large. The simulation data are generated with the following approach. Each data observation in the $n \times p$ predictor matrix $\boldsymbol{X}$ is independently sampled from a multivariate normal distribution $N(\boldsymbol{\mu}_X, \sigma_X \boldsymbol{I})$, where the location parameter $\boldsymbol{\mu}_X$ is sampled from a uniform distribution Unif$(\boldsymbol{\mu}_{\min}, \boldsymbol{\mu}_{\max})$. The corresponding observations in the $n \times q$ response matrix $\boldsymbol{Y}$ are generated following the definition of the MVPLN model in Equations (1) and (2). In order to enforce sparsity, a fixed number of zeros are randomly placed into each column of the coefficient matrix $\boldsymbol{B}$. The other nonzero entries of $\boldsymbol{B}$ are independently sampled from a univariate normal distribution $N(\mu_B, \sigma_B)$.

In reference to the inverse covariance matrix $\boldsymbol{\Omega} = \boldsymbol{\Sigma}^{-1}$ for $\boldsymbol{\varepsilon}$, we consider 4 scenarios: (1) Random $\boldsymbol{\Omega}$, where the inverse covariance matrix is generated by $\boldsymbol{\Omega} = \boldsymbol{\Psi}^T \boldsymbol{\Psi}$ to ensure the positive semidefinite property. Each entry in $\boldsymbol{\Psi}$ is independently sampled from a uniform distribution $Unif(-1, 1)$; (2) Banded $\boldsymbol{\Omega}$, where the sparsity is enforced by the modified Cholesky decomposition [22]: $\boldsymbol{\Omega} = \boldsymbol{T}^T \boldsymbol{D}^{-1} \boldsymbol{T}$. Here, $\boldsymbol{T}$ is a lower triangular matrix with $1'$s on the diagonal, and $\boldsymbol{D}$ is a diagonal matrix. The nonzero off-diagonal elements in $\boldsymbol{T}$ and diagonal elements in $\boldsymbol{D}$ are independently sampled from the uniform distributions $Unif(-1, 1)$ and $Unif(0, 1)$, respectively; (3) sparse $\boldsymbol{\Omega}$, where the $\boldsymbol{\Omega}$ matrix is generated by performing some random row and column permutations over the banded $\boldsymbol{\Omega}$ matrix; (4) Diagonal $\boldsymbol{\Omega}$, where the diagonal elements are sampled independently from the standard uniform distribution. In order to make sure that the elements in the response matrix $\boldsymbol{Y}$ are within a reasonable range, we scale the matrix $\boldsymbol{\Sigma}$ to make the largest element equal to ψ. By tuning the synthetic data generation parameters $\boldsymbol{\mu}_{\min}$, $\boldsymbol{\mu}_{\max}$, σ_X, μ_B, σ_B, and ψ, we could adjust the range and variations in the generated response $\boldsymbol{Y}$.

In our experiments, we fix the number of observations in the training data at $n = 50$ and the number of observations in the test data at 20. We consider 2 scenarios: (1) the dimension of predictors is less than the number of observations in training data ($p < n$), and (2) the dimension of predictors is greater than or equal to the number of observations in training data ($p \geq n$). We let $p = 30$, $q = 5$ for the case $p < n$, and $p = 70$, $q = 5$ for the case $p \geq n$. For each parameter setting, the simulation is repeated 60 times, and the reported results are averaged across the 60 replications to alleviate the randomness.

4.1 | Estimation accuracy

To measure the model estimation accuracy w.r.t. $\boldsymbol{B}$ and $\boldsymbol{\Omega}$, we report the estimation errors by computing the distance

TABLE 1 Estimation errors w.r.t. $\boldsymbol{B}$ and $\boldsymbol{\Omega}$

		$l(B,\widehat{B})$				$l(\Omega,\widehat{\Omega})$			
		$p<n$		$p>n$		$p<n$		$p>n$	
Ω	ψ	GLMNET	MVPLN	GLMNET	MVPLN	GLMNET	MVPLN	GLMNET	MVPLN
Random	0.4	2.25607	**1.19936**	1.61016	**1.44383**	NA	0.99550	NA	0.99595
		(0.04547)	**(0.01277)**	(0.01830)	**(0.01076)**		(0.00131)		(0.00100)
	1.0	4.35649	**1.70326**	2.41644	**1.74861**	NA	0.99033	NA	0.99151
		(0.09258)	**(0.03620)**	(0.03039)	**(0.02796)**		(0.00153)		(0.00200)
	1.6	5.37513	**1.80392**	2.87839	**1.94325**	NA	0.98928	NA	0.98561
		(0.12519)	**(0.03618)**	(0.04629)	**(0.02844)**		(0.00211)		(0.00452)
	2.2	6.32172	**1.99852**	3.21878	**2.12487**	NA	0.99214	NA	0.98343
		(0.17932)	**(0.04246)**	(0.05822)	**(0.04339)**		(0.00126)		(0.00328)
Banded	0.4	2.12650	**1.16671**	1.49028	**1.38619**	NA	0.98029	NA	0.98500
		(0.05377)	**(0.01882)**	(0.01747)	**(0.01133)**		(0.00204)		(0.00148)
	1.0	3.57945	**1.59062**	2.13400	**1.59255**	NA	0.95796	NA	0.94881
		(0.10031)	**(0.04760)**	(0.03313)	**(0.02344)**		(0.00508)		(0.00563)
	1.6	4.41182	**1.80692**	2.59768	**1.78361**	NA	0.93159	NA	0.92380
		(0.13408)	**(0.06746)**	(0.05930)	**(0.02981)**		(0.00811)		(0.00874)
	2.2	5.21359	**2.04397**	2.84779	**2.01992**	NA	0.93695	NA	0.90552
		(0.18171)	**(0.07308)**	(0.07824)	**(0.05492)**		(0.00681)		(0.00838)
Sparse	0.4	1.98327	**1.11950**	1.52410	**1.40847**	NA	0.98277	NA	0.98205
		(0.06026)	**(0.01556)**	(0.02270)	**(0.01107)**		(0.00259)		(0.00201)
	1.0	3.43339	**1.50384**	2.13721	**1.60572**	NA	0.95978	NA	0.96085
		(0.11127)	**(0.04915)**	(0.03966)	**(0.02315)**		(0.00597)		(0.00425)
	1.6	4.69189	**1.88319**	2.54446	**1.76144**	NA	0.92684	NA	0.92349
		(0.15989)	**(0.07134)**	(0.05723)	**(0.02705)**		(0.00880)		(0.00852)
	2.2	5.09710	**2.12963**	2.74681	**1.91288**	NA	0.96626	NA	0.90581
		(0.21733)	**(0.07617)**	(0.08444)	**(0.04085)**		(0.01344)		(0.00957)
Diagonal	0.4	1.86103	**1.10292**	1.43937	**1.34607**	NA	0.96841	NA	0.97068
		(0.05898)	**(0.01452)**	(0.01870)	**(0.01274)**		(0.00324)		(0.00413)
	1.0	3.29868	**1.53224**	2.01567	**1.56539**	NA	0.88673	NA	0.89628
		(0.09724)	**(0.04655)**	(0.04295)	**(0.02745)**		(0.01313)		(0.01510)
	1.6	4.33160	**1.84269**	2.39551	**1.70712**	NA	0.81895	NA	0.84071
		(0.13345)	**(0.06302)**	(0.05794)	**(0.04889)**		(0.01851)		(0.02020)
	2.2	5.00582	**1.95903**	2.56122	**1.76716**	NA	0.88405	NA	0.81663
		(0.23160)	**(0.08481)**	(0.08119)	**(0.03930)**		(0.02031)		(0.02034)

Abbreviations: GLMNET, a univariate Lasso regularized Poisson regression model; MVPLN, multivariate Poisson log-normal.

The standard errors are shown in the parentheses.

between $\boldsymbol{B}$ and $\widehat{\boldsymbol{B}}$ (or $\boldsymbol{\Omega}=\boldsymbol{\Sigma}^{-1}$ and $\widehat{\boldsymbol{\Omega}} = \widehat{\boldsymbol{\Sigma}}^{-1}$) using the normalized matrix Frobenius norm

$$l(\boldsymbol{B},\widehat{\boldsymbol{B}}) = \frac{\|\boldsymbol{B}-\widehat{\boldsymbol{B}}\|_F}{\|\boldsymbol{B}\|_F}.$$

Here, $\boldsymbol{B}$ denotes the true value of the coefficient matrix, and $\widehat{\boldsymbol{B}}$ represents the estimation provided by the MVPLN or GLMNET models. Table 1 shows the estimation errors of coefficient matrix $\boldsymbol{B}$ and inverse covariance matrix $\boldsymbol{\Omega}$ in various parameter settings. Since the GLMNET model cannot infer the inverse covariance matrix, we omit the corresponding results here. We can see that the proposed MVPLN model consistently outperforms the GLMNET model in all parameter settings, especially when the variation in the simulated data is large (ψ is large). Such promising results demonstrate that the proposed MVPLN model leverages the dependence structures between the multidimensional count responses to improve the estimation accuracy.

4.2 | Prediction accuracy

To evaluate the prediction performance of the proposed model, we report the average root-mean-square error (rMSE) across all the response dimensions over the test data. Figures 2 and 3 show the average rMSE for the cases when $p<n$ and $p \geq n$, respectively. These figures show that when the variations in the simulated data are small (ψ is small), the prediction performances of the proposed MVPLN model and GLMNET model are comparable. As the variations in the data increase, the prediction performance of the proposed MVPLN

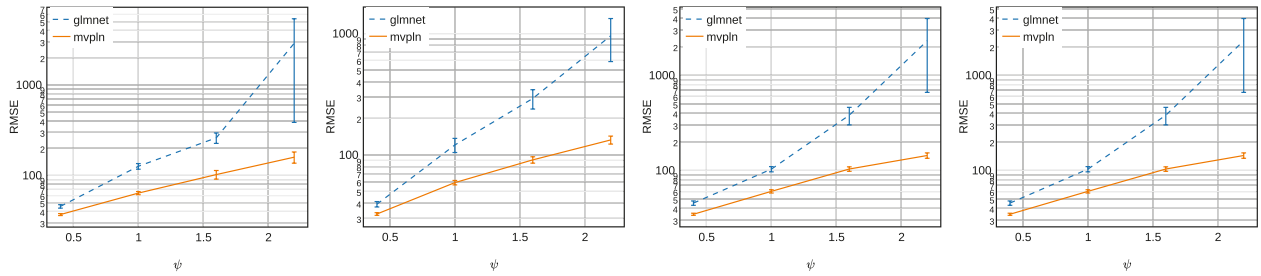

FIGURE 2 Average root-mean-square error (rMSE) across response dimensions over test data when $\mathbf{\Omega}$ is random, sparse, banded, and diagonal (from left to right), and $p < n$. The vertical error bars indicate the standard deviation, and the Y-axis is in log scale

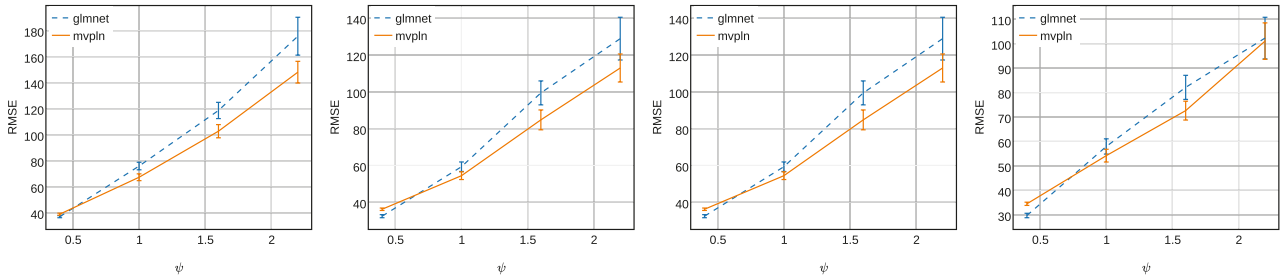

FIGURE 3 Average root-mean-square error (rMSE) across response dimensions over test data when $\mathbf{\Omega}$ is random, sparse, banded, and diagonal (from left to right), and $p \geq n$. The vertical error bars indicate the standard deviation

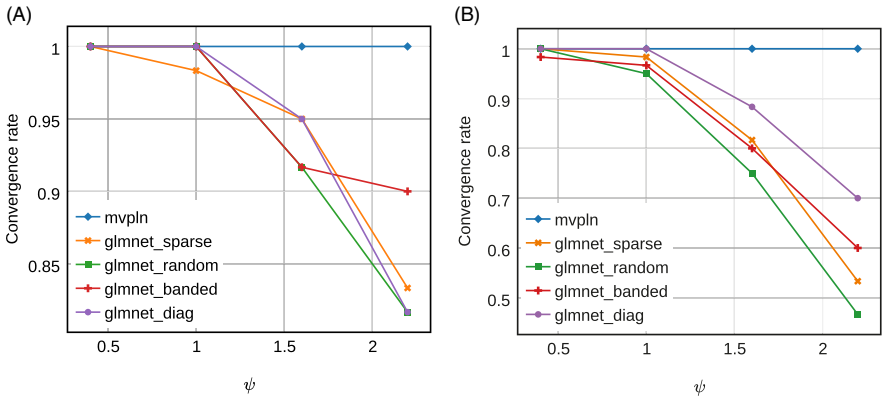

FIGURE 4 Convergence ratio of univariate Lasso regularized Poisson regression model (GLMNET) and multivariate Poisson log-normal (MVPLN) model when $p < n$ (A) and $p > n$ (B). Since MVPLN model always converges, we use a single line to represent these 4 scenarios

model becomes better than that of the GLMNET model. This demonstrates that by incorporating the dependence structures between the count responses, the proposed MVPLN model improves its prediction performance. However, when the variations in data are small, it is difficult for the MVPLN model to take advantage of the inverse covariance matrix estimation substep. On the contrary, approximating the log-likelihood with MCMC techniques would impose negative effects on the model estimation and prediction accuracy. This is why we observe that when ψ is small, the proposed MVPLN model sometimes does not perform as well as the GLMNET model in terms of rMSE.

4.3 | Algorithm convergence

Another aspect we would like to emphasize here is the convergence behavior of the algorithms. During the experiments over the simulated data, we notice that the GLMNET model will not always converge in some parameter settings, especially when ψ is large. As a result, no parameter estimations are given by the GLMNET model. Figure 4 shows the convergence ratio (the fraction of experiment replications that converge and produce valid model estimations) over the simulated data for various parameter settings. We can see from the figure that the larger the variations (larger ψ) in the data, the more frequently the GLMNET model fails to yield a valid model estimation. On the other hand, the proposed MVPLN model consistently produces valid model estimates in all scenarios. These results demonstrate that the proposed MVPLN model is more robust to variations in the underlying multivariate datasets.

5 | REAL CASE STUDY

We apply the proposed MVPLN model to a real ILI dataset from 2 Latin American countries (Brazil and Chile) each with 4 types of ILI diseases (FLUAH3, FLUB, FLUH1N1,

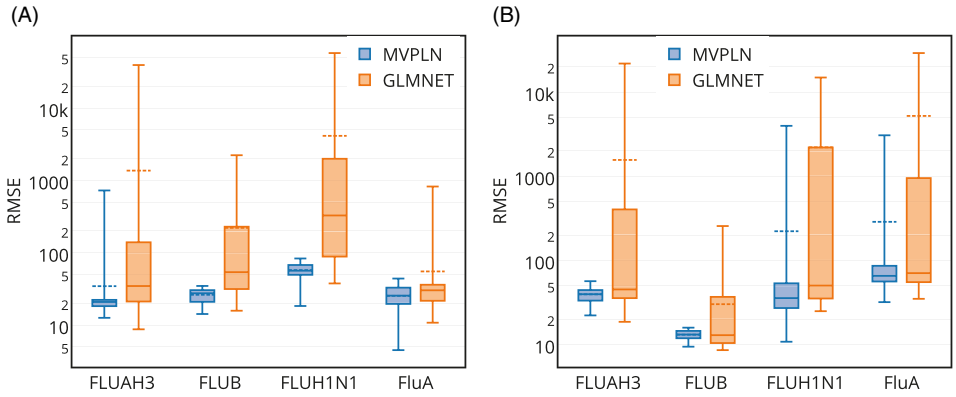

FIGURE 5 Root-mean-square error (rMSE) box plot of multivariate Poisson log-normal (MVPLN) and univariate Lasso regularized Poisson regression model (GLMNET) models on the real influenza-like-illness (ILI) dataset for the countries of Brazil (A) and Chile (B). The dash lines indicate the mean of the rMSE

and FLUA). The weekly count data of these 4 types of flus were collected from WHO FluNet [33] from May 1, 2012, to December 27, 2014, which serves as the multivariate responses of the dataset. The predictors of this ILI dataset are the weekly counts of ILI-related keywords among tweets (approximate compressed size of 570 GB) collected during the same period by the EMBERS project [26]. Before applying the proposed MVPLN model, we scale the count for each ILI keyword to zero mean and unit standard deviation.

TABLE 2 Median of the rMSE for the MVPLN and the GLMNET models in the one-step-ahead predictions on the real ILI datasets of Brazil and Chile

	Brazil		Chile	
	GLMNET	**MVPLN**	**GLMNET**	**MVPLN**
FLUA	0.5447	**0.3258**	4.5019	**2.4111**
FLUAH3	**9.3611**	11.5275	7.7388	**3.9937**
FLUB	6.6138	**2.7946**	6.7833	**5.6197**
FLUH1N1	6.2099	**4.2896**	8.0286	**0.9922**

Abbreviations: GLMNET, a univariate Lasso regularized Poisson regression model; ILI, influenza-like-illness; MVPLN, multivariate Poisson log-normal; rMSE, root-mean-square error.

5.1 | In-sample prediction

It should also be noticed that although this ILI dataset is time-indexed, we chose to model it as merely a multivariate dataset in our first study here since the proposed MVPLN model is not specifically designed to model time series datasets. We use 70% of the preprocessed ILI dataset as the training set and the rest (30%) as the test set. We apply the proposed MVPLN model over the training set, and compute the rMSE of the test set as the criterion for the prediction performance of the model. As a comparison, we also apply the GLMNET model to the same ILI dataset, and compare the rMSE with the proposed MVPLN model. We repeat this experiment for 60 independent runs, and for each run we shuffle the ILI dataset and re-split the training and test sets. Figure 5 shows the rMSE box plots of the proposed MVPLN model and the GLMNET model for Brazil and Chile after removing some extreme outliers. As we can see from the box plots, although the proposed MVPLN model generates slightly large rMSE over the test set for some response dimensions occasionally, in general the rMSEs of the MVPLN model are much smaller and have less variation when compared to the GLMNET model for both Brazil and Chile. This indicates that the proposed MVPLN model is better and more stable in terms of prediction performance over real datasets with count responses. Thus, by leveraging the covariance structure between multiple count responses, the proposed MVPLN model is able to improve the prediction performance. However, we also notice that for some flu types, the proposed MVPLN model sometimes generate a large rMSE value, for example, FLUAH3 in the Brazil dataset, and FLUH1N1 and FLUA in the Chile dataset. A potential reason for this behavior is likely that the data shuffling procedure happens to place most of the large-response data instances into the model training set, which could mislead model estimation and result in overestimation over the test set.

5.2 | One-week-ahead predictions

In this experiment, we take the time domain in the ILI dataset into account and perform a 1-week-ahead prediction of the ILI counts using a model inferred from data of the past N weeks. To be more specific, we train the proposed MVPLN model over the preprocessed ILI dataset filtered by a time window of size N (weeks), forecast the ILI counts for the consecutive week, and then move the time window 1 week ahead to make prediction for the next week. In our experiment, we set the size of this time window to be $N = 80$, and perform 50 steps of each such 1-week-ahead forecast. Table 2 shows the median of the rMSE for the 1-week-ahead prediction over the ILI datasets of Brazil and Chile. As we can see, the proposed MVPLN model performs better than the GLMNET model in terms of the prediction errors for such 1-week-ahead prediction in most of the cases.

As mentioned earlier, the proposed MVPLN model is not particularly designed for modeling time series data. If we extend the current proposed MVPLN model to be able to

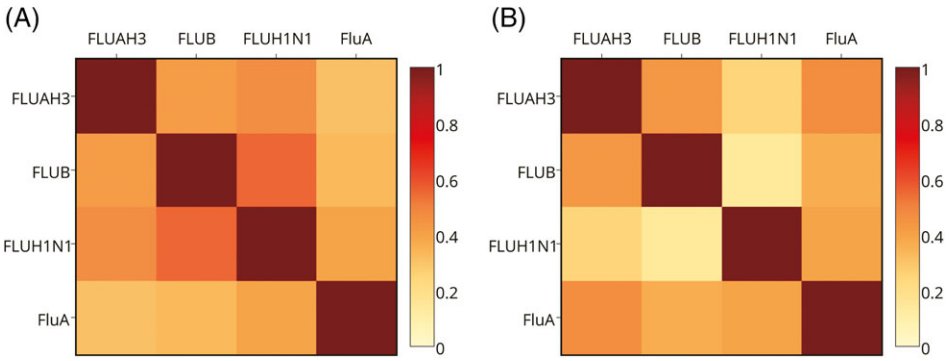

FIGURE 6 Heatmap of the averaged partial correlation matrix estimated by the multivariate Poisson log-normal (MVPLN) model for the one-step-ahead predictions on the real influenza-like-illness (ILI) datasets of Brazil (A) and Chile (B)

deal with time series data with multiple count responses, we believe it would provide more promising results. This could be an interesting direction for future research and investigation on this topic.

We also study the dependencies between the multiple responses (here, different strains of the flu) in the ILI dataset. Based on the estimated inverse covariance matrix in each iteration of the 1-week-ahead predictions, we calculate the corresponding partial correlation matrix. The heatmap in Figure 6 describes the mean of the partial correlation matrix between the 4 types of flus under consideration across 50 iterations of the 1-week-ahead predictions for Brazil and Chile. As the figure shows, FLUA is weakly correlated with FLUAH3 and FLUB in Brazil, while in Chile, FLUH1N1 is weakly correlated with FLUAH3 and FLUB compared to the rest of the combinations of the 4 types of flus in Brazil and Chile. Identifying such dependency structures between different strains of flus will aid public health officials in understanding the prevalence and spread of infectious diseases and support the organization of suitable countermeasures (eg, vaccines).

6 | DISCUSSION OF RELATED WORK

6.1 | Multiresponse regression

From the regression perspective, multiresponse regression aims to estimate the regression coefficients and recover the covariance structure among response variables. The MRCE approach [27] simultaneously conducts a sparse estimation of the coefficient matrix and the covariance structure by optimizing the penalized likelihood. Similarly, Lozano et al. [24] proposed a framework to jointly estimate a sparse functional mapping from multiple predictors to multiple responses along with estimating the conditional dependency structure among the responses via a penalized l_2 distance criterion. Calibrated multivariate regression [23] calibrates the regularization for each regression task w.r.t. its noise level to achieve a better finite sample performance. In block-regularized Lasso [31], the authors studied the multivariate multiresponse linear regression problem via l_1/l_2 regularized Lasso, which couples the multiple responses together.

6.2 | Multitask Learning

Much work in this area has focused on learning the shared features or underlying common structures among multiple tasks. The $l_{2,1}$ matrix norm was adopted as a regularizer to learn a low-dimension shared structure across multiple related tasks [1]. Here, given a matrix $\boldsymbol{X}$ of size $n \times p$, the $l_{2,1}$ norm is defined as $\|\boldsymbol{X}\|_{2,1} = \sum_{i=1}^{n} (\sum_{j=1}^{p} |x_{i,j}|^2)^{\frac{1}{2}}$. In the formulation known as multistage, multitask learning [13], a capped-l_1 penalty was imposed to estimate the sparse shared features. By including a squared norm regularizer, the calibrated multivariate regression model [15] was applied to learn the shared features among tasks. Kumar and Daumé III [21] proposed a framework to learn task grouping and overlaps that enables the selective sharing of information across the tasks. A multitask learning framework was proposed to forecast spatiotemporal events in ref. [37]. A multitask feature learning framework proposed by Gong et al. [14] simultaneously captures shared features among related tasks and identifies outlier tasks by imposing group Lasso penalties over row groups and column groups. Chen et al. [5] proposed a multitask learning framework by penalizing the loss function with trace and $l_{1,2}$ norms to infer the common low-rank structure among relevant tasks and uncover the irrelevant tasks using a group-sparse structure. Here, given the matrix $\boldsymbol{X}$ of size $n \times p$, the $l_{1,2}$ norm is defined as $\|\boldsymbol{X}\|_{1,2} = \sum_{j=1}^{p} (\sum_{i=1}^{n} |x_{i,j}|^2)^{\frac{1}{2}}$. Yu et al. [36] studied a Bayesian multitask learning formulation with t-process, and demonstrated that the t-process could efficiently distinguish the good tasks from noisy or outlier tasks through their empirical studies. Compared to the l-norm as a regularizer, Argyriou et al. [2] proposed a framework to learn common shared structures based on regularization with spectral functions of matrices; in the dirty model for multitask learning [18], the estimated parameter matrix can be decomposed into a row-sparse matrix corresponding to overlapping features and an element-wise sparse matrix corresponding to nonshared features.

6.3 | Poisson Models

In addition to multitask learning research, which mainly focuses on the continuous responses, there are also several

ideas that adopt Poisson models in the realm of applications such as ILI, traffic accident analysis, and consumer services. Wang et al. [32] proposed a dynamic Poisson autoregression model for short-term ILI case count prediction. Ma et al. [25] and El-Basyouny and Sayed [7] provided a multivariate model specification to simultaneously model the crash or collision counts by injury severity in a traffic accident. Wang et al. [30] proposed a multivariate Poisson regression model to predict the purchase patterns of cross-category store brands. However, compared to the proposed model in this paper, the Poisson regression models in the literature are either univariate or inferred with a Bayesian approach and no sparsity is enforced over the coefficients, which loses the variable selection capability. Moreover, the covariance matrix involved in such multivariate models is sampled from a specific prior distribution, which requires prior knowledge about the data. Our approach, in contrast, directly infers the covariance matrix from the data.

Other related works, for example, ref. [20], formulate the multivariate Poisson variable as a linear combination of several independent univariate Poisson random variables, and the covariance structure is directly captured by sharing the common univariate Poisson variables among different dimensions. However, with such formulations, more regression coefficients need to be estimated if we want to capture cross-way covariance patterns. Zoh et al. [38] propose to use Poisson distributions parameterized by linear models over latent variables to model multivariate count variables. This approach induces a factorized covariance matrix to quantify the dependence among mutiple variables. Different from Zoh's work, the proposed MVPLN model aims to establish the connections between the features and count responses and estimate sparse covariance structures among multiple responses directly from the data. In contrast, in Zoh's work, the covariance structure is parameterized by the coefficients of latent variables and is estimated under a Bayesian framework.

7 | CONCLUSION

In this paper, we have proposed and formulated a multivariate Poisson log-normal model for data with count responses. By developing an MCEM algorithm, we accomplished simultaneous sparse estimations of the regression coefficients and of the inverse covariance matrix of the model. Results of simulation studies on synthetic data and an application to a real ILI dataset demonstrated that the proposed MVPLN model could achieve better estimation and prediction performance vs a classical Lasso-regularized Poisson regression model.

Note that the proposed method is not restricted to using the l_1 regularization in (7) to encourage model sparsity. In this work, we adopted the l_1 regularization to achieve general sparse structures in the model parameters ($\boldsymbol{B}$ and $\boldsymbol{\Sigma}$). Alternatively, one can also adopt other penalty terms such as elastic net [39] or group Lasso [28] to encourage certain special sparse structure in the model. In addition, there are a few interesting directions for future work of the proposed model: (1) it would be interesting to investigate some asymptotic properties of the proposed model. (2) To mitigate the convenience issue of the MCEM techniques, we plan to develop a better approximation algorithm, for example, using variational inference [3]. (3) We will also extend the proposed model to better deal with count data with overdispersion and zero-inflation situations.

ACKNOWLEDGMENTS

This work was supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior National Business Center (DoI/NBC) contract number D12PC000337. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoI/NBC, or the U.S. Government.

ORCID

Xinwei Deng http://orcid.org/0000-0002-1560-2405

REFERENCES

1. A. Argyriou, T. Evgeniou, and M. Pontil, *Multi-task feature learning*, NIPS, 2007, pp. 41–48.
2. A. Argyriou, C. A. Micchelli, M. Pontil, and Y. Ying, *A spectral regularization framework for multi-task structure learning*, NIPS, 2007.
3. D. M. Blei, A. Kucubelbir, and J. D. McAuliffe, Variational inference: A review for statisticians, 2016, available at https://arxiv.org/abs/1601.00670.
4. J. Chen and Z. Chen, *Extended Bayesian information criteria for model selection with large model spaces*, Biometrika 95(3) (2008), 759–771. MR2443189
5. J. Chen, J. Zhou, and J. Ye, *Integrating low-rank and group-sparse structures for robust multi-task learning*, KDD '11, 2011, pp. 42–50.
6. S. Chib, E. Greenberg, and R. Winkelmann, *Posterior simulation and bayes factors in panel count data models*, J. Econometrics 86(1) (1998), 33–54.
7. K. El-Basyouny and T. Sayed, *Collision prediction models using multivariate Poisson-lognormal regression*, Accid. Anal. Prev. 41(4) (2009), 820–828.
8. R. Foygel and M. Drton, *Extended Bayesian information criteria for Gaussian graphical models*, NIPS, 2010, pp. 604–612.
9. J. Friedman et al., *Pathwise coordinate optimization*, Ann. Appl. Stat. 1(2) (2007), 302–332. MR2415737
10. J. Friedman, T. Hastie, and R. Tibshirani, *Sparse inverse covariance estimation with the graphical lasso*, Biostatistics 9(3) (2008), 432–441.
11. J. Friedman, T. Hastie, and R. Tibshirani, *The elements of statistical learning*, Springer series in statistics, Vol 2, Springer-Verlag, New York, 2009. MR2722294
12. J. Friedman, T. Hastie, N. Simon, and R. Tibshirani, Lasso and elastic-net regularized generalized linear models, glmnet R package, 2014.
13. P. Gong, J. Ye, and C. Shui Zhang, *Multi-stage multi-task feature learning*, NIPS, 2012.
14. P. Gong, J. Ye, and C. Zhang, *Robust multi-task feature learning*, KDD '12, 2012, pp. 895–903.

15. P. Gong, J. Zhou, W. Fan, and J. Ye, *Efficient multi-task feature learning with calibration*, KDD '14, 2014, pp. 761–770.
16. F. Hadiji et al., *Poisson dependency networks: Gradient boosted models for multivariate count data*, Mach. Learn. 100(2) (2015), 477–507. MR3383979
17. N. J. Higham, *Computing the nearest correlation matrix – A problem from finance*, J. Numer. Anal. 22(3) (2002), 329–343.
18. A. Jalali, S. Sanghavi, C. Ruan, and P. K. Ravikumar, *A dirty model for multi-task learning*, NIPS, 2010, pp. 964–972.
19. D. Karlis, *An EM algorithm for multivariate Poisson distribution and related models*, J. Appl. Stat. 30(1) (2003), 63–77. MR1957361
20. D. Karlis and L. Meligkotsidou, *Multivariate Poisson regression with covariance structure*, Stat. Comput. 15(4) (2005), 255–265. MR2205389
21. A. Kumar and H. Daumé III, *Learning task grouping and overlap in multi-task learning*, ICML '12, 2012.
22. E. Levina et al., *Sparse estimation of large covariance matrices via a nested lasso penalty*, Ann. Appl. Stat. 2(1) (2008), 245–263. MR2415602
23. H. Liu, L. Wang, and T. Zhao, *Multivariate regression with calibration*, NIPS, 2014, pp. 127–135.
24. A. C. Lozano, H. Jiang, and X. Deng, *Robust sparse estimation of multiresponse regression and inverse covariance matrix via the l2 distance*, KDD '13, 2013, pp. 293–301.
25. J. Ma, K. M. Kockelman, and P. Damien, *A multivariate Poisson-lognormal regression model for prediction of crash counts by severity, using Baysian methods*, Accid. Anal. Prev. 40 (2008), 964–975.
26. N. Ramakrishnan, P. Butler, S. Muthiah, N. Self, R. Khandpur, P. Saraf, W. Wang, J. Cadena, A. Vullikanti, G. Korkmaz, C. Kuhlman, A. Marathe, L. Zhao, T. Hua, F. Chen, C. T. Lu, B. Huang, A. Srinivasan, K. Trinh, L. Getoor, G. Katz, A. Doyle, C. Ackermann, I. Zavorin, J. Ford, K. Summers, Y. Fayed, J. Arredondo, D. Gupta, and D. Mares, *'Beating the news' with embers: Forecasting civil unrest using open source indicators*, KDD '14, 2014, pp. 1799–1808.
27. A. J. Rothman, E. Levina, and J. Zhu, *Sparse multivariate regression with covariance estimation*, J. Comput. Graph. Statist. 19(4) (2010), 947–962. MR2791263
28. N. Simon et al., *A sparse-group lasso*, J. Comput. Graph. Statist. 22(2) (2013), 231–245. MR3173712
29. R. Tibshirani, *Regression shrinkage and selection via the lasso*, J. R. Stat. Soc. Ser. B 58 (1994), 267–288. MR2815776
30. H. Wang, M. U. Kalwani, and T. Akçura, *A Bayesian multivariate Poisson regression model of cross-category store brand purchasing behavior*, J. Retailing Consumer Serv. 14(6) (2007), 369–382.
31. W. Wang, Y. Liang, and E. P. Xing, *Block regularized lasso for multivariate multi-response linear regression*, AISTATS, 2013, pp. 608–617.
32. Z. Wang, P. Chakraborty, S. R. Mekaru, J. S. Brownstein, J. Ye, and N. Ramakrishnan, *Dynamic Poisson autoregression for influenza-like-illness case count prediction*, KDD '15, 2015, pp. 1285–1294.
33. WHO FluNet, 2015, available at http://www.who.int/influenza/gisrs_laboratory/flunet/en/.
34. M. Wytock and J. Z. Kolter, *Sparse Gaussian conditional random fields: Algorithms, theory, and application to energy forecasting*, ICML '13, 2013, pp. 1265–1273.
35. E. Yang, P. K. Ravikumar, G. I. Allen, and Z. Liu, *On Poisson graphical models*, NIPS '13, 2013.
36. S. Yu, V. Tresp, and K. Yu, *Robust multi-task learning with t-processes*, ICML '07, 2007.
37. L. Zhao, Q. Sun, J. Ye, F. Chen, C.-T. Lu, and N. Ramakrishnan, *Multi-task learning for spatio-temporal event forecasting*, KDD '15, 2015, pp. 1503–1512.
38. R. S. Zoh et al., *PCAN: Probabilistic correlation analysis of two non-normal data sets, Biometrics* 72(4) (2016), 1358–1368. MR3591620
39. H. Zou and T. Hastie, *Regularization and variable selection via the elastic net*, J. R. Stat. Soc. Ser. B (Stat. Methodol.) 67(2) (2005), 301–320. MR2137327

How to cite this article: Wu H, Deng X, Ramakrishnan N. Sparse estimation of multivariate Poisson log-normal models from count data. *Stat Anal Data Min: The ASA Data Sci Journal*, 2018;11:66–77. https://doi.org/10.1002/sam.11370.