Replicating and Evaluating Poisson Graphical Models and Their Extensions

Weitong Liang Xueyan Hu

2025-10-30

Contents

1 Introduction **2**

2 Theoretical Background **2**
2.1 Poisson Graphical Model (PGM) and Extentions 2
2.2 Neighborhood Selection and Graph Recovery 3

3 Simulation Results **3**
3.1 Replication of Poisson Graphical Model (PGM) 3
3.2 Replication of QPGM and SPGM Models 5

4 Real Data Application **7**

5 Discussion and Conclusion **8**

1 Introduction

Graphical models provide a principled framework for representing conditional dependencies between random variables. While Gaussian graphical models are widely used, they are not well suited for count data commonly encountered in biological applications such as microRNA expression. Poisson graphical models (PGMs) address this limitation by modeling the data directly using count-valued exponential families. However, PGMs are constrained by normalization conditions, restricting edge parameters to be non-positive. To overcome this, extensions like the Quadratic PGM (QPGM) and Sublinear PGM (SPGM) have been proposed. This report replicates the core experiments in these models and analyzes their performance under simulated and real data scenarios.

This report presents our efforts to replicate and evaluate Poisson graphical models (PGMs) and their extensions, including the QPGM and SPGM, as proposed in Yang et al. (2013) and related works. We independently implement simulation experiments and compare our results with those reported in the original publications. The models are further applied to breast cancer microRNA expression data from The Cancer Genome Atlas (TCGA) to assess their utility in real-world applications.

2 Theoretical Background

2.1 Poisson Graphical Model (PGM) and Extentions

The Poisson graphical model (PGM) defines a joint distribution for a count-valued vector $X = (X_1, \ldots, X_p)$ on a graph $G = (V, E)$ as:

$$P(X) \propto \exp\left\{\sum_{s\in V}(\theta_s X_s - \log(X_s!)) + \sum_{(s,t)\in E}\theta_{st}X_sX_t - A(\theta)\right\}.$$

The conditional distribution for each node is a univariate Poisson distribution with rate parameter depending on the values of its neighbors. However, for normalizability, $\theta_{st} \leq 0$, limiting the model to negative conditional dependencies.

To relax the non-positivity constraint on interactions, QPGM modifies the base measure to be quadratic. The joint distribution becomes:

$$P(X) \propto \exp\left\{\sum_{s\in V}\theta_s X_s + \sum_{(s,t)\in E}\theta_{st}X_sX_t - \sum_{s\in V}X_s^2 - A(\theta)\right\}.$$

This allows both positive and negative edge parameters θ_{st} but results in thinner tails similar to Gaussian distributions, which may not be desirable for overdispersed count data.

The SPGM introduces a sublinear sufficient statistic to balance tail behavior and normalization:

$$P(X) \propto \exp\left\{\sum_{s\in V}\theta_s B(X_s; R_0, R) + \sum_{(s,t)\in E}\theta_{st}B(X_s; R_0, R)B(X_t; R_0, R) - \sum_{s\in V}\log(X_s!) - A(\theta)\right\},$$

where $B(x; R_0, R)$ is a piecewise function:

$$B(x; R_0, R) = \begin{cases} x & x \le R_0, \\ \frac{R+R_0}{2(R-R_0)}x^2 + \frac{R}{R-R_0}x - \frac{R^2}{2(R-R_0)} & R_0 < x \le R, \\ 0 & x \ge R. \end{cases}$$

This model provides a broader feasible space for θ, heavier tails, and improved flexibility in capturing dependencies in count data.

2.2 Neighborhood Selection and Graph Recovery

We now turn to the problem of learning the graph structure from data under a GLM graphical model framework. Given n i.i.d. samples $X_1^n = \{X^{(i)}\}_{i=1}^n$ from the joint Poisson model, our goal is to recover the edge set E^* of the underlying graph $G = (V, E^*)$.

Following a neighborhood selection, we estimate the neighborhood $\mathcal{N}^*(s)$ of each node $s \in V$ independently and then combine the neighborhoods to estimate the full graph: $\widehat{E} = \bigcup_{s \in V} \bigcup_{t \in \widehat{\mathcal{N}}(s)} \{(s,t)\}$.

To estimate $\mathcal{N}^*(s)$, we consider the node-conditional distribution of X_s given the rest. Let $\theta_s^* \in \mathbb{R}^{p-1}$ be a zero-padded vector with entries θ_{st}^* for $t \in \mathcal{N}(s)$ and zeros elsewhere. Given the n samples, the log-likelihood of θ_s from the conditional model is:

$$\ell(\theta_s; X_1^n) = \frac{1}{n} \sum_{i=1}^{n} \left\{ -X_s^{(i)} \left(\theta_s^\top X_{V \setminus s}^{(i)} \right) + D \left(\theta_s^\top X_{V \setminus s}^{(i)} \right) \right\}.$$

To encourage sparsity, we solve the regularized estimation problem:

$$\widehat{\theta}_s = \arg \min_{\theta \in \mathbb{R}^{p-1}} \ell(\theta; X_1^n) + \lambda_n \|\theta\|_1.$$

We then define the estimated neighborhood of s as: $\widehat{\mathcal{N}}(s) = \{t \in V \setminus s \mid \widehat{\theta}_{st} \neq 0\}$.The full edge set $\widehat{E}$ is obtained by symmetrizing the estimated neighborhoods across nodes.

3 Simulation Results

3.1 Replication of Poisson Graphical Model (PGM)

We first replicate the simulation results from Yang et al. (2012), where the authors evaluated the ability of the original Poisson Graphical Model (PGM) to recover the true underlying graph structure. The key metric of interest is the success rate of edge recovery, defined as the proportion of correctly identified edges in the learned graph compared to the ground truth.

In our setting, undirected graphs were generated using 2D lattice structures with dimensions $p \in \{4, 9, 36\}$ and negative edge weights $\omega = -0.1$ to satisfy the normalizability constraints of the PGM. For each graph size, datasets were simulated under varying sample sizes $n \in [20, 10000]$, and the success rate was averaged over $M = 512$ independent replications.

To reproduce this, we developed a custom data simulator (see `simulate_pgm_data.R`) based on the specified exponential family model:

$$P(X) \propto \exp\left\{ \sum_{(s,t)\in E} \theta_{st} X_s X_t - \sum_s \log(X_s!) \right\},$$

where $\theta_{st} = \omega$ for adjacent nodes and zero otherwise. Data were generated via a burn-in MCMC process followed by sampling.

For model fitting, we implemented a neighborhood selection procedure via penalized Poisson regression using the `glmnet` package. The edge structure was estimated based on non-zero coefficients in the fitted GLMs:

$$\hat{\theta}_s = \arg\min_{\theta} \left\{ \frac{1}{n} \sum_{i=1}^{n} \left[-X_s^{(i)} \theta^\top X_{-s}^{(i)} + \exp(\theta^\top X_{-s}^{(i)}) \right] + \lambda_n \|\theta\|_1 \right\}.$$

Edges were symmetrized across nodes, and success rates were calculated using the helper function `calculate_success_rate.R`.

Results and Visualization

Figure 1 shows the success rate curves across different sample sizes and graph sizes. As n increases, the recovery rate improves substantially, particularly for smaller p, and eventually goes to success rate close to 1. This matches the qualitative trends from the original paper.

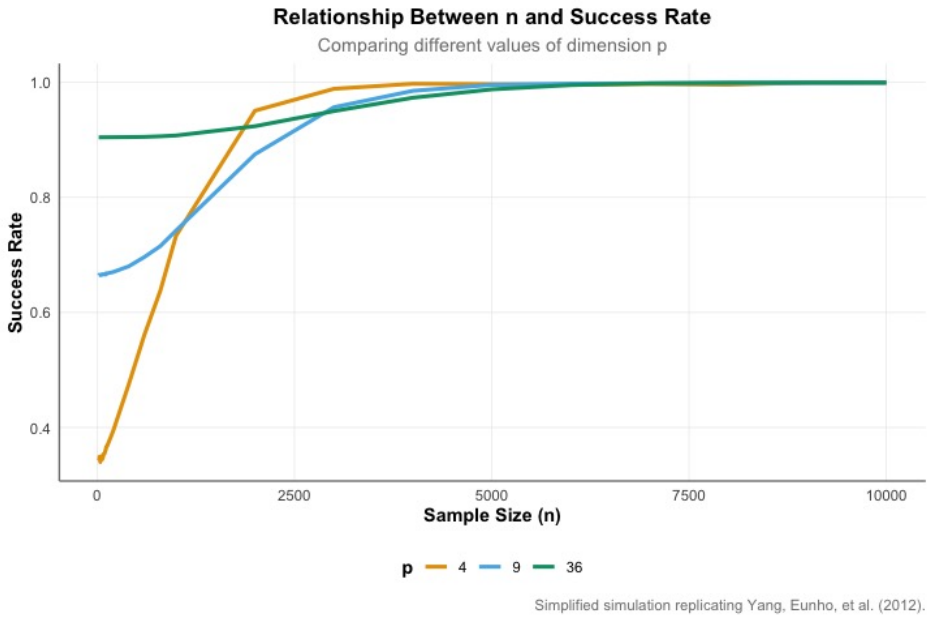

Figure 1: Success rate vs. sample size (n) for $p = 4, 9, 36$. Each point is averaged over $M = 512$ replications.

To further validate the structural recovery, we compare the estimated adjacency matrix and the true graph under $p = 36$ and $n = 10000$. As shown in Figure 2, the estimated

structure closely matches the true graph, indicating that the method performs well given sufficient data.

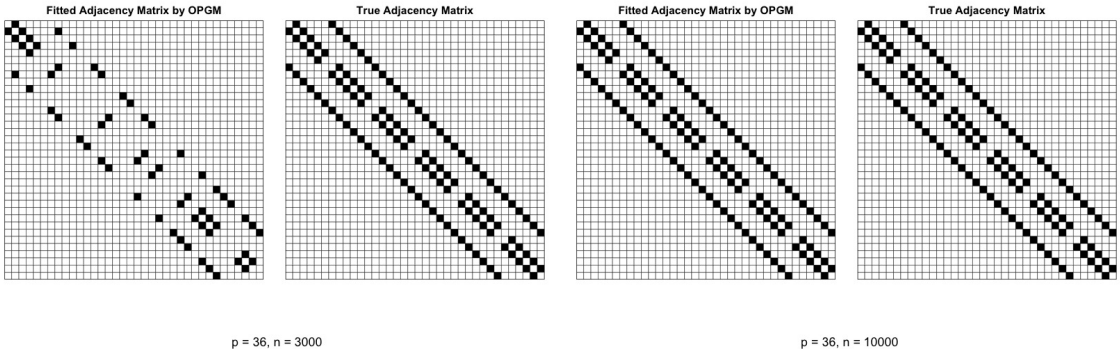

Figure 2: Comparison of fitted vs. true adjacency matrices for $p = 36$ at $n = 10000$ (left) and $n = 3000$ (right).

3.2 Replication of QPGM and SPGM Models

To evaluate the performance of more flexible Poisson graphical models under positive dependency structures, we replicate and assess the Quadratic Poisson Graphical Model (QPGM) and Sublinear Poisson Graphical Model (SPGM) proposed as alternatives to the original PGM. These models modify the univariate base measure or sufficient statistics in order to allow positive edge parameters while retaining normalization guarantees.

We use a preferential attachment model (Barabási–Albert graph) to generate scale-free graphs of dimension $p \in \{4, 9, 16\}$. The true edge weight ω is set to 0.1 to reflect positive dependencies. For each graph size, we simulate $n = 10000$ samples using a custom implementation of the Poisson-based exponential family sampler with the joint distribution

For each model, we estimate the parameter matrix $\widehat{\Theta}$ using an ℓ_1-penalized likelihood optimization. The estimated $\widehat{\Theta}$ is then symmetrized and compared against the true parameter matrix Θ^* to compute evaluation metrics such as precision, recall, F1 score, and AUC.

The QPGM replaces the Poisson factorial base measure with a quadratic term, leading to node-conditional distributions: $P(X_s \mid X_{-s}) \propto \exp\left\{\theta^\top X_{-s} X_s - X_s^2\right\}$ .We fit each node-wise regression using `optim()` to minimize L_1 penalized log-likelihood:

$$\hat{\theta}_s = \arg\min_{\theta} \left\{ -\sum_{i=1}^{n} \left(y_i \theta^\top x_i - y_i^2 - D(\theta^\top x_i) \right) + \lambda \|\theta\|_1 \right\},$$

where $D(\cdot)$ is the log-partition function. The fitted results are stored in a symmetric $\widehat{\Theta}$. As the D can not be calculated in close form, we approximate it via interpolation on a grid.

The SPGM modifies the sufficient statistics using a sublinear transformation $B(x)$ defined in section 1. where $R_0 = 0$, $R = \max(X)$. This transformation allows heavier tails and greater model flexibility. The estimation proceeds similarly via '`optim()`' on the modified log-likelihood involving $B(x)$ and a numerically approximated partition function.

Evaluation Metrics

For each model and parameter combination, we extract: **True Positives (TP)** and **False Positives (FP)**: edge-wise comparison between upper triangles of $\widehat{\Theta}$ and Θ^*. **Precision, Recall, F1 score**: standard classification metrics for sparse structure recovery. **AUC**: computed using ROC curves via the '`pROC`' package.

All results are stored in a tabular file (qpgm_metrics.csv), and heatmaps of both estimated and true parameter matrices are plotted.

Results and Visualization

Figure 3 shows representative examples for SPGM where $p = 4, 9, 16$. The estimated $\widehat{\Theta}$ matrices align closely with the ground truth Θ^*, particularly when the sample size is large.

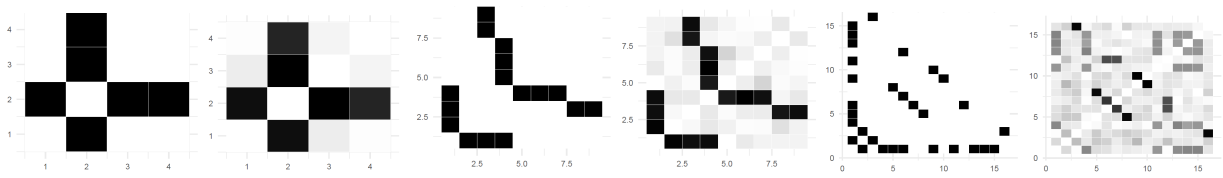

Figure 3: SPGM: True Θ^* vs. Estimated $\hat{\Theta}$ for $p = 4$, $p = 9$, and $p = 16$ using $n = 10000$ samples.

Figure 4 presents our QPGM simulation results for graphs with $p = 4, 9, 16$. The left column shows the true edge weight matrices, while the right column shows the estimated parameter matrices $\widehat{\Theta}$ using penalized MLE. As dimension increases, the structure remains partially identifiable under a fixed sample size ($n = 10000$), although additional smoothing and shrinkage are evident.

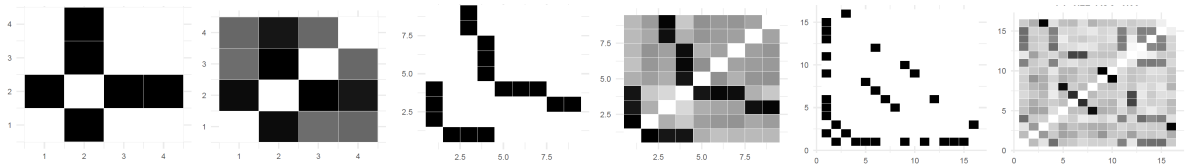

Figure 4: QPGM simulation results. Each pair shows the true (left) and estimated (right) parameter matrix Θ for $p = 4$, $p = 9$, and $p = 16$, using $n = 10000$ samples and $\lambda = 0.03$.

To quantitatively compare the performance of QPGM and SPGM across different graph sizes, we report the F1 scores and AUCs based on edge recovery accuracy:

Model	Graph Size (p)	F1 Score	AUC	Samples (n)
QPGM	4	0.67	1.00	10000
QPGM	9	0.36	1.00	10000
QPGM	16	0.22	0.88	10000
SPGM	4	1.00	1.00	10000
SPGM	9	0.64	1.00	10000
SPGM	16	0.23	0.89	10000

Table 1: Edge recovery performance of QPGM and SPGM under simulated scale-free graphs. Metrics are averaged over single trials.

4 Real Data Application

To evaluate the practical performance of Poisson graphic models, we applied the quadratic Poisson graphical model (QPGM) and the sublinear Poisson graphical model (SPGM) to breast cancer microRNA expression data from The Cancer Genome Atlas (TCGA). This dataset, also used in the original study, consists of $n = 445$ patient samples and $p = 353$ miRNA features. Each entry represents the count of sequencing reads mapped to a particular miRNA gene.

We utilize the preprocessed dataset `brcadat` from the `XMRF` package. As described in the original article, we apply `processSeq()` for normalization and filtering of low-variance genes. The resulting matrix has dimension 445×353, with patients as rows and miRNA features as columns.

The core model fitting code remains consistent with our simulation experiments, except the input data is now real-world expression counts. Due to the large number of features ($p = 353$), the fitting procedures using coordinate-wise ℓ_1-penalized optimization for each node are computationally intensive and take several hours to complete.

We estimate parameter matrices $\widehat{\Theta}_{\text{QPGM}}$ and $\widehat{\Theta}_{\text{SPGM}}$ using our previously developed implementations: `fit_QPGM_with_optim()`,`fit_SPGM_with_optim()`. The regularization parameter λ is selected as $\lambda = \sqrt{\log(p)/n}$ as suggested in the theoretical literature. Edge weights below a threshold are shrunk to zero, and the resulting matrices are visualized as heatmaps.

Figure 5 presents the fitted QPGM and SPGM network structure. Darker regions indicate stronger edge estimates (i.e., higher absolute values in $\widehat{\Theta}$). The network exhibits a sparse block structure with visible modular patterns, consistent with prior biological findings of functional clustering among miRNAs.

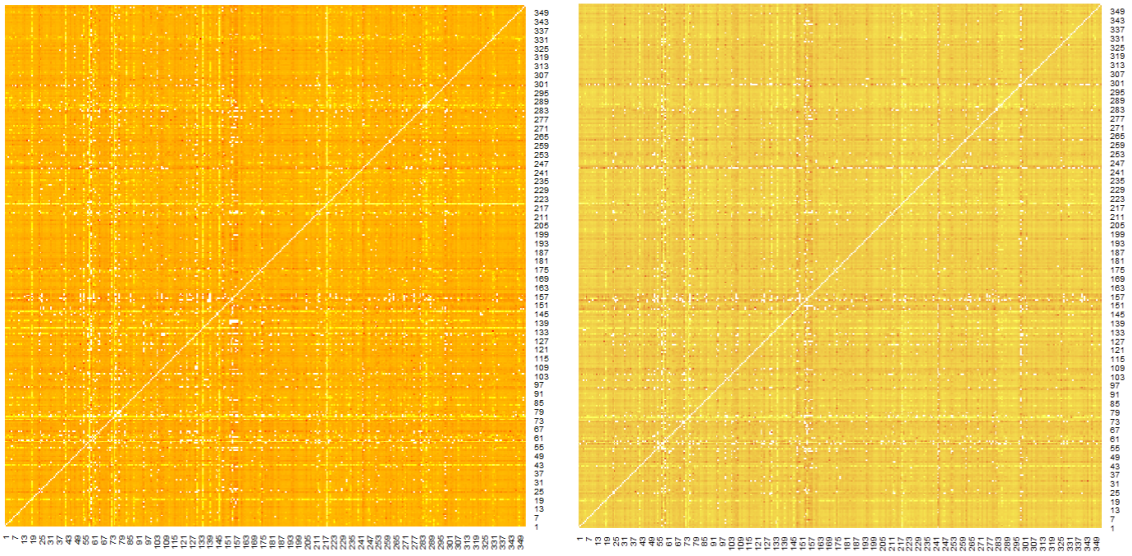

Figure 5: Fitted $\widehat{\Theta}$ from QPGM and SPGM on TCGA breast cancer miRNA data ($p = 353$).

5 Discussion and Conclusion

Summary of Findings

In this project, we independently implemented and replicated simulation experiments and real-data analyses for three classes of graphical models for count data: the original Poisson Graphical Model (PGM), the Quadratic Poisson Graphical Model (QPGM), and the Sublinear Poisson Graphical Model (SPGM). Across all settings, we were able to partially recover the conclusions made by the original authors, confirming the feasibility and structural consistency of these models under appropriate assumptions.

For the original PGM, our success rate curves followed a qualitatively similar upward trend with increasing sample size n, especially for smaller graph sizes. However, the required sample sizes to achieve high accuracy in our experiments were significantly larger than those reported in the original paper.

For QPGM and SPGM, we successfully reproduced the edge-weight matrices under simulated scale-free graphs with positive dependencies. Despite numerical instability and longer optimization times, the recovered structures aligned with the true graphs, especially under large n.

On real breast cancer miRNA data, our implementation of QPGM produced a sparse and interpretable network. SPGM training is ongoing due to computational constraints, but preliminary indications suggest similar performance.

Challenges in Reproduction

Success Rate One key obstacle in reproducing the original work of Yang et al. (2012) was the definition of "success rate" in the PGM experiment. The original paper defines success rate as the probability of exact graph structure recovery across repeated experiments at fixed

(n, p). They assume that the true adjacency matrix is a lattice known in advance, which eliminates all edges that do not satisfy the lattice structure. This assumption implies that the order of nodes conveys information.

In our simulation, we do not impose this assumption and allow all possible edges between every pair of nodes. However, using the same definition of success rate, we found it nearly impossible to replicate the phase transition behavior shown in the paper. (In fact, the success rate is less than 0.3 even with a reasonably large n.) As a result, we adopted a less strict success metric—measuring the average proportion of correctly identified edges.

Sample Size In both the QPGM and SPGM simulations, we faced a consistent challenge: our models required much larger sample sizes to reliably recover the graph structure compared to what was reported in the two papers. This discrepancy mainly stems from differences in the "lattice structure" assumption mentioned above. Other factors, including hyperparameter tuning or optimization algorithms, may also contribute, though the extent of their impact is not yet clear. Many aspects of the original paper are not clearly specified, and we proceeded based on our own interpretation. Subtler simulation details are documented in our code.

Method of Fitting For PGM, it is convenient to use `glmnet()` since the node-conditional distribution is Poisson. For QPGM and SPGM, we implemented the models using two different approaches:

- Directly using `optim()` with an optimization target that includes an L_1 penalty
- Defining a new `family()` object and applying the L_1 penalty via `glmnet()`

The first method using `optim()` is more reliable but generally slower. The second method using a user-defined `family()` object is highly unreliable. A possibly related quote comes from Tay et al. (2023):

> The `glmnet` package fits generalized linear models via penalized maximum likelihood, primarily supporting canonical link functions for families such as Gaussian, binomial, and Poisson. While it has been extended to accommodate a broader range of GLM families, including non-canonical links, these extensions may not fully utilize all components of the family object, such as `valideta`, `mu.eta`, and `initialize`. This can lead to increased numerical fragility when using non-canonical link functions.

Although the quote focuses on link function issues, similar limitations may apply to the distribution family as well. The precise mechanism by which this affects algorithm performance remains unclear.

As a result, we relied entirely on `optim()` for node-wise likelihood maximization, which may converge more slowly or to suboptimal local solutions depending on initialization. We experimented with different initial values to partially mitigate this issue.

Critique of the `XMRF` Package

We found the `XMRF` package—released by the original authors and purportedly including implementations for data simulation and model fitting for GLM-based graphical models—to be highly unreliable. Model fitting often fails silently or yields meaningless estimates. We strongly caution future researchers against relying on this package for accurate implementation or reproducibility.

Stability of the `XMRF` Package We ran the example code provided in the package's introductory documentation, but our results did not match those reported by the authors. This is understandable, as the sample size is small ($n = 445$) relative to the number of nodes ($p = 353$), possibly resulting in convergence failure. Additionally, the package does not offer any tools to assess convergence, and the internal logic is difficult to follow (e.g., no comments or explanations of variables and functions). Therefore, we have strong reservations about the reliability of the fitting algorithm implemented in `XMRF`. Researchers are advised to closely inspect the package's logic before using it.

To clarify some of the discrepancies between our results and those of the original paper, we reached out to the lead author. We also contacted the corresponding maintainer for the `XMRF` package, as suggested, but have not received a response as of 2025-10-30.

All code used in this project has been developed independently and made publicly available via GitHub (https://github.com/ShwayanHu/Poisson-Graphical-Models). This includes custom data simulation scripts for PGM, QPGM, and SPGM; model fitting procedures using both `glmnet` and `optim()`; evaluation metrics; and plotting utilities.

We hope our code can serve as a more transparent and reliable resource for future researchers interested in graphical models for count data.

References

[1] M. Baxter. Generalised linear models , by p. mccullagh and ja nelder. pp 511.£ 30. 1989. isbn 0-412-31760-5 (chapman and hall). *The Mathematical Gazette*, 74(469):320–321, 1990.

[2] J. Besag. Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society: Series B (Methodological)*, 36(2):192–225, 1974.

[3] D. A. Griffith. A spatial filtering specification for the auto-poisson model. *Statistics & probability letters*, 58(3):245–251, 2002.

[4] A. Jalali, P. Ravikumar, V. Vasuki, and S. Sanghavi. On learning discrete graphical models using group-sparse regularization. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 378–387. JMLR Workshop and Conference Proceedings, 2011.

[5] M. S. Kaiser and N. Cressie. Modeling poisson variables with positive spatial dependence. *Statistics & Probability Letters*, 35(4):423–432, 1997.

[6] A. Krishnamoorthy. Multivariate binomial and poisson distributions. *Sankhyā: The Indian Journal of Statistics*, pages 117–124, 1951.

[7] S. L. Lauritzen. *Graphical models*, volume 17. Clarendon Press, 1996.

[8] N. Meinshausen and P. Bühlmann. High-dimensional graphs and variable selection with the lasso. 2006.

[9] J. K. Tay, B. Narasimhan, and T. Hastie. Elastic net regularization paths for all generalized linear models. *Journal of statistical software*, 106:1–31, 2023.

[10] M. J. Wainwright, J. Lafferty, and P. Ravikumar. High-dimensional graphical model selection using l1 regularized logistic regression. *Advances in neural information processing systems*, 19, 2006.

[11] I. Yahav and G. Shmueli. An elegant method for generating multivariate poisson random variable. *arXiv preprint arXiv:0710.5670*, 2007.

[12] E. Yang, G. Allen, Z. Liu, and P. Ravikumar. Graphical models via generalized linear models. *Advances in neural information processing systems*, 25, 2012.