Copula-based spatio-temporal modelling for count data

PuXue Qiao

Submitted in total fulfillment of the requirements of the degree of Doctor of Philosophy

School of Mathematics & Statistics
The University of Melbourne

August, 2019

Abstract

Modelling of spatio-temporal count data has received considerable attention in recent statistical research. However, the presence of massive correlation between locations, time points and variables imposes a great computational challenge. In existing literature, latent models under the Bayesian framework are predominately used. Despite numerous theoretical and practical advantages, likelihood analysis of spatio-temporal modelling on count data is less wide spread, due to the difficulty in identifying the general class of multivariate distributions for discrete responses.

In this thesis, we propose a Gaussian copula regression model (copSTM) for the analysis of multivariate spatio-temporal data on lattice. Temporal effects are modelled through the conditional marginal expectations of the response variables using an observation-driven time series model, while spatial and cross-variable correlations are captured in a block dependence structure, allowing for both positive and negative correlations. The proposed copSTM model is flexible and sufficiently generalizable to many situations. We provide pairwise composite likelihood inference tools. Numerical examples suggest that the proposed composite likelihood estimator produces satisfactory estimation performance.

While variable selection of generalized linear models is a well developed topic, model subsetting in applications of Gaussian copula models remains a relatively open research area. The main reason is the computational burden that is already quite heavy for simply fitting the model. It is therefore not computationally affordable to evaluate many candidate sub-models. This makes penalized likelihood approaches extremely inefficient because they need to search through different levels of penalty strength, apart from the fact suggested by our numerical experience that optimization of penalized composite likelihoods with many popular penalty terms (e.g LASSO and SCAD) usually does not converge in copula models. Thus, we propose to use a criterion-based selection approach that borrows strength from the Gibbs sampling technique. The methodology guarantees to converge to the model with the lowest criterion value, yet without searching through all possible models exhaustively.

Finally, we present an R package implementing the estimation and selection of the copSTM model in C++. We show examples comparing our package to many available R packages (on some special cases of the copSTM), confirming the correctness and ef-

ficiency of the package functions. The package `copSTM` provides a competitive toolkit option for the analysis spatio-temporal count data on lattice in terms of both model flexibility and computational efficiency.

Declaration

This is to certify that

• The thesis comprises only my original work towards the PhD,

• due acknowledgement has been made in the text to all other material used,

• the thesis is less than 100,000 words in length, exclusive of tables, maps, bibliographies and appendices.

– PuXue Qiao
August, 2019

Acknowledgements

First and foremost, I would like to express my greatest appreciation to my principal PhD supervisor, A/Prof Guoqi Qian. I would like to thank him for his guidance, continuous support and trust, without which this thesis would have not been possible. His advice on both research as well as on my career has been invaluable.

I would also like to thank A/Prof Frédéric Hollande, who has taught me, both consciously and unconsciously, how to be a good researcher and team collaborator. It has been a very enjoyable and fruitful experience working with his team. Special thanks to Dr. Christina Mølck for her patience and tremendous effort in our collaborative work.

I am also grateful to my co-supervisor, Dr. Davide Ferrari, who used to be my principle supervisor before his resignment, for leading me into this exciting area and for his effort and advice to get our first paper published.

My sincere thanks also go to Jackson Kwock and Yuqing Pan for their generous help and inspiration.

Finally, I would like to thank my family, my Mum and Dad, uncle and aunt, and my beloved partner, Daniel Garcia, for their constant love, for supporting me in everyway and for encouraging me throughout this experience.

To my family

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Longitudinal image data based on fluorescent proteins play a crucial role for both in vivo and in vitro analysis of various biological processes such as gene expression and cell lineage fate. Assessing the growth patterns of different cell types within a heterogeneous population and monitoring their interactions enables biomedical researchers to determine the role of different cell types in important biological processes such as organ development and regeneration, malignant growth or immune responses under various experimental conditions. For example, tumor progression has been shown to be affected by bidirectional interactions among cancer cells or between cancer cells and cells from the microenvironment, including tumor-infiltrating immune cells Medema and Vermeulen (2011). Being able to study these interactions in a laboratory setting is therefore highly relevant, but is complicated by the difficulty of dissecting the effect of the different cell types as soon as the number of cell types exceeds two. In the present study we used longitudinal image data collected from multicolor live-cell imaging growth experiments of co-cultures of cancer cells and fibroblasts (a key cell type in the tumor microenvironment) as well as behaviourally distinct (cloned) cancer cells. Using a high-content imaging system, we were able to acquire characteristics for each individual cell at subsequent times, including fluorescent properties, spatial coordinates, and morphological features. The motivation of this work was to design a model allowing the determination of spatio-temporal growth interactions between these multiple cell populations.

In longitudinal growth experiments, the two important goals are to determine growth rates for different cell populations and to assess how interactions between cell types may affect their growth. Whilst a wide range of descriptive data analysis approaches has been used in applications, inference based on a comprehensive model of multicolor cell data is an open research area. The main challenges are related to the difficulties related to tracking individual cells across time from image data and the presence of complicated

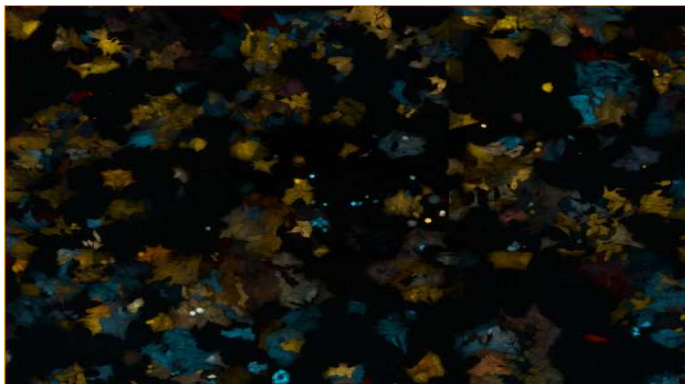


Figure 1.1: Microscope images for the cancer cell growth data obtained from a high-content imager (Operetta, Perkin Elmer).

spatio-temporal interactions amongst cells.

1.2 Cancer cell growth data with RGB marking

The technique of RGB marking has been recently introduced to facilitate the identification of individual cell clones (Weber et al, 2012). Since the colored cells are easily identifiable within whole organ structures, scientists can track the cells and determine their role during processes such as organ regeneration, malignant outgrowth or immune responses. A raw image data representing colorectal cancer cells are shown in Figure1.1.

Typical longitudinal experiments consist of a relatively small number of measurements (e.g. 5 to 20 images taken every few hours), which is adequate for monitoring cell growth. However, tracking individual cells would typically require more frequent measurements, complicating the practicality of the experiments in terms of the storage cost of very large image files and the cytotoxicity induced by the imaging process. Tracking individual cell trajectories is difficult also because of cell migration, overlapping cells, changes in cell morphology, image artifacts, cell death and division. But obtaining a clustering of cell types with respect to colour is feasible and can be automated.

RGB marking introduces three lentiviral vectors in individual cells encoding the basic colors red, green and blue. The cells were imaged on a high-content imager (Operetta, Perkin Elmer). In particular, the data set consists of measurements on colorectal cancer cell lines expressing various quantities of three different fluorescent proteins: Cerulean (blue), Venus (yellow/green), and mCherry (red).

The genes coding for the fluorescent proteins were transferred into the cells via lentivirus-mediated transduction at a less than 100% efficiency so that most cells expressed different quantitative combinations of all three fluorescent proteins as described by Weber et al (2012). Due to variability of the vector insertion, single RGB-marked cells express fluorescent proteins at different and very characteristic levels. The underlying principle

of additive color mixing, similar to that in computer or TV screens, generates different color combinations that can be used to discriminate individual cell clones.

The final data consisted of fluorescent intensities of red, blue and green color channels (electromagnetic wavelength in nanometers, nm), spatial coordinates and morphology parameters including cell areas, roundness etc. Here we propose to preprocess the data by clustering cell populations according to their colour combinations, so that in the downstream analysis, we can focus on modelling spatio-temporal impacts on growth between different cell clusters.

1.2.1 Preprocessing the data

To cluster cell types according to the colours they express, we first standardise each of the three colour Original intensities by subtracting its mean and dividing by its standard deviation, which keeps the shape of data while scales the intensities to achieve the same mean and standard deviation. Figure 1.2 show histograms of original and standardised colour intensities of Ceuriliean, Venus and mCherry respectively.

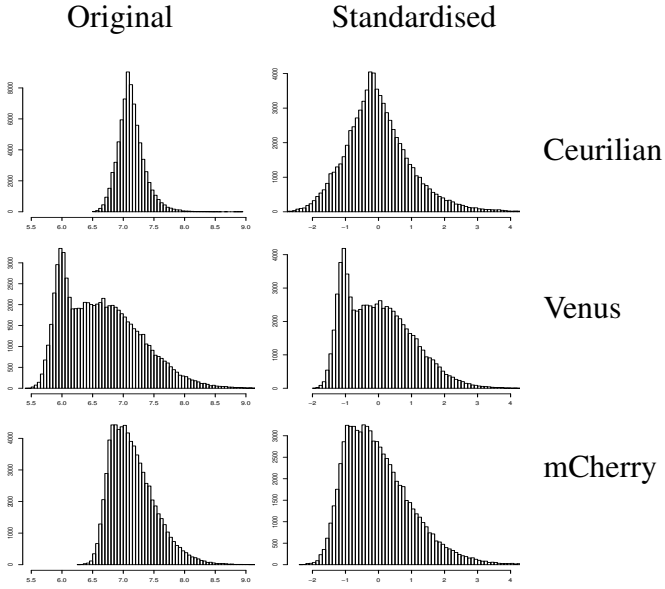


Figure 1.2: Histograms of original and standardised RGB colour intensities.

One of the most popular clustering methods is the K-means, of which the attractiveness lies in its simplicity and in its local-minimum convergence probabilities. However, one shortcoming of K-means is that it is very sensitive to outliers. Since we observe a concentrated collection of outliers on the lower end of the histogram of Venus in Figure 1.2, a traditional K-means does not suit. Thus, we carry out a robust version of K-means with an R package `RSKC` by Kondo et al. (2016). Instead of updating the cluster centers to the sample mean of all the observations in each cluster, the robust K-means trims $\alpha 100\%$ of the observations with the largest distance to their cluster centres, and update the cluster centers with the remaining observations. Our numerical result show that $\alpha = 0.1$ gives reasonably stable clustering results.

Another difficulty is that the intrinsic variability of the underlying biological mechanisms make the actual number of distinguishable colors generated by RGB marking in a tissue difficult to predict. In addition, cell intensities for different colors are known to vary depending on the cell area, which is an indicator of cell morphology.

In the following section, we introduce a regression clustering method that not only

cluster individual cells according to colour combinations, but also takes into account the dependency between response (cell colours) and explanatory variables, such as the cell area.

1.3 A Semi-supervised Regression Clustering

This section is a condensed version of a collaborative work with Qian et al. (2016).

1.3.1 Methodology

Let $\boldsymbol{z}_1 = (y_1, \boldsymbol{x}_1')', \cdots, \boldsymbol{z}_n = (y_n, \boldsymbol{x}_n')'$ be the observed data set of n data points, where $(y_j, \boldsymbol{x}_j'), j = 1, \ldots, n$, where $\boldsymbol{x}_j$ is an explanatory column vector and $y_j \in \mathbb{R}$ a random dependent variable for the jth data point. Suppose the data coming from k different populations. The goal of regression clustering is to recover the latent partitioning $\Pi = (\mathscr{C}_1, \cdots, \mathscr{C}_k)$ of $\{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n\}$ to conform to their respective populations as much as possible.

A grouped linear model is used to describe the data,

$$y_j = \boldsymbol{x}_j' \boldsymbol{\beta}_i + e_j, \quad e_j \sim N(0, \sigma_i^2) \quad \text{for all } j \in \mathscr{C}_i;\ i = 1, \ldots, k. \tag{1.1}$$

Optimal parameter estimation and partition can be achieved using the maximum likelihood principle. Under the fixed partition model (1.1), the log-likelihood function is given by

$$\log L_n(k, \Pi, (\boldsymbol{\beta}_1, \sigma_1^2), \cdots, (\boldsymbol{\beta}_k, \sigma_k^2)) = -\frac{1}{2} \sum_{i=1}^{k} \sum_{j \in \mathscr{C}_i} \left(\log 2\pi + \log \sigma_i^2 + \frac{(y_j - \boldsymbol{\beta}_i' \boldsymbol{x}_j)^2}{\sigma_i^2} \right). \tag{1.2}$$

It is clear that the best estimates of the parameters and the partition should be those maximizing the log-likelihood (1.2) for given k. However, due to the large number of possible partitions, it is almost impossible to find the global optimal partition by enumeration. Thus we use an iterative estimation method to find a local optimal estimates of $(\boldsymbol{\beta}_i, \sigma_i^2)_{i=1,\ldots,k}$ and Π for a given k, that extends the exchange method of Späth (1979, 1982).

When fixing $(\boldsymbol{\beta}_i, \sigma_i^2)_{i=1,\ldots,k}$ at given estimates $(\hat{\boldsymbol{\beta}}_i, \hat{\sigma}_i{}^2)_{i=1,\ldots,k}$, (1.2) achieves the maximum if each data point j belongs to cluster

$$\hat{\mathscr{C}}_i = \arg \min_{1 \le i \le k} \left(\log \hat{\sigma}_i^2 + \frac{(y_j - \hat{\boldsymbol{\beta}}_i' \boldsymbol{x}_j)^2}{\hat{\sigma}_i{}^2} \right). \tag{1.3}$$

At given $\hat{\mathscr{C}}_i$, $i = 1, \cdots, k$, (1.2) is the sum of the usual log-likelihood functions for homogeneous linear regressions within clusters. Hence, it is maximized at the least squares(LS)

estimates $\hat{\boldsymbol{\beta}}_i$ obtained based on the data points within $\hat{\mathscr{C}}_i$, and

$$\hat{\sigma}_i^2 = \frac{\sum_{j\in\hat{\mathscr{C}}_i}(y_j - \hat{\boldsymbol{\beta}}_i' \boldsymbol{x}_j)^2}{\hat{n}_i}, \quad \text{where } \hat{n}_i = |\hat{\mathscr{C}}_i| \text{ is the size of } \hat{\mathscr{C}}_i,\ i = 1,\ldots,k. \tag{1.4}$$

Then $\log \hat{L}_n$ is monotonically increased if the steps (1.3) and (1.4) are carried out alternately. This procedure leads to a local maximum in finitely many steps. It is expected to be a good approximation of the global maximum if an initial partition is properly chosen.

It is well-known that the least squares method is very sensitive to outliers and violation of the normality assumption in the data. Robust methods can be developed to overcome this vulnerability. Among them, procedures based on M-estimation(RM) are considered here. M-estimation can be regarded as a generalization of the maximum likelihood estimation. A particular one is the maximum likelihood estimation based on Huber's least favourable distribution, whose density function is the normal at around the origin and the exponential in the tails. Using Huber's M-estimation method, we can drop the assumption $e_j \sim N(0,\sigma_i^2)$ in (1.1) and estimate $\boldsymbol{\beta}_i$ by minimizing $\sum_{j\in\hat{\mathscr{C}}_i}\rho_c(y_j - \boldsymbol{\beta}_i'\boldsymbol{x}_j)$ for given partition $\hat{\mathscr{C}}_i$, $i = 1,\ldots,k$. Here $\rho_c(\cdot)$ is Huber's discrepancy function defined as

$$\rho_c(t) = \begin{cases} \dfrac{1}{2}t^2, & |t| < c, \\ c|t| - \dfrac{1}{2}c^2, & |t| \geq c, \end{cases} \tag{1.5}$$

where c is determined by the scale parameter in Huber's least favourable distribution. We find that assuming a constant scale parameter across all clusters tends to give a better robust results, so adopt this assumption in this section. Now for given estimates $\hat{\boldsymbol{\beta}}_i$, $i = 1,\ldots,k$, each data point j is assigned or reassigned to cluster $\hat{\mathscr{C}}_i = \arg\min_{1\leq i\leq k}\rho_c(y_j - \hat{\boldsymbol{\beta}}_i'\boldsymbol{x}_j)$. At this point, it can be seen that, instead of $\log \hat{L}_n$, the function $\sum_{i=1}^k\sum_{j\in\mathscr{C}_i}\rho_c(y_j - \boldsymbol{\beta}_i'\boldsymbol{x}_j)$ will be monotonically increased if the above two M-estimation steps are carried out alternately. This gives a robust counterpart of the likelihood-based local optimal estimation and selection introduced earlier in this section.

1.3.2 Determining the number of clusters

Depending on the experiment setups, the number of cell populations is known in some of the data sets analysed in this thesis, if not, one can decide the optimal number of clusters by minimizing an information criterion function.

For LS regression clustering, Shao and Wu (2005) develop a criterion as

$$D(\Pi_k) = \sum_{i=1}^{k} ||\boldsymbol{y}_{\mathscr{C}_i} - X_{\mathscr{C}_i}\hat{\boldsymbol{\beta}}_i||^2 + q(k)A_n, \tag{1.6}$$

where $q(k)$ is a strictly increasing positive function of k, A_n is a sequence of positive constants, $\hat{\boldsymbol{\beta}}_i$ are least squares estimators, and $||\cdot||$ is the Euclidean norm. Typically $q(k) = kp$ and $A_n \propto \log(n)$ or $A_n \propto \log\log(n)$ are chosen.

For the RM method, we adopt the robust information criterion by Rao et al. (2007):

$$R(\Pi_k) = \sum_{s=1}^{k} \sum_{j \in \mathscr{C}_s} \rho_c(y_{j,\mathscr{C}_s} - \boldsymbol{x}'_{j,\mathscr{C}_s} \widehat{\boldsymbol{\beta}}_s) + q(k) A_n, \tag{1.7}$$

where ρ_c is the Huber's discrepancy function, and $\widehat{\boldsymbol{\beta}}_s$ are the M-estimators.

1.3.3 Point-wise iterative algorithm

The first terms in information criteria (1.6) and (1.7) measure the goodness-of-fit of the model, similar to which we define the within-cluster sum of residual squares as

$$SRSS(\Pi_k, \boldsymbol{\beta}_1, \cdots, \boldsymbol{\beta}_k) = \sum_{i=1}^{k} ||\boldsymbol{y}_{\mathscr{C}_i} - X_{\mathscr{C}_i} \boldsymbol{\beta}_i||^2 \tag{1.8}$$

for LS regression clustering and

$$RRSS(\Pi_k, \boldsymbol{\beta}_1, \cdots, \boldsymbol{\beta}_k) = \sum_{i=1}^{k} \sum_{j=1}^{n_i} \rho_c(y_{j,\mathscr{C}_i} - \boldsymbol{x}'_{j,\mathscr{C}_i} \boldsymbol{\beta}_i) \tag{1.9}$$

for an M-estimation based robust regression clustering.

For a fixed number of clusters, k, the iterative optimization procedure can be accomplished according to the following algorithm:

(i) Label all the observations from 1 to n (order does not matter). Given an initial partition $\Pi_k = \{\mathscr{C}_1, \cdots, \mathscr{C}_k\}$ of $\{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n\}$, fit a regression model (or a robust regression model with a $\rho_c(\cdot)$ function) in each of the k clusters and obtain the sum of the residual squares sums RRSS_0 for this partition. Let $i = 0$.

(ii) Set $i = i + 1$, and reset $i = 1$ if $i > n$. Identify $\mathscr{C}_j$ such that $i \in \mathscr{C}_j$. Then move i into $\mathscr{C}_h$ with $h = 1, \cdots, k$, $h \neq j$ respectively. For each of these $k - 1$ relocations, re-fit the model by regression clustering (or robust regression clustering) and calculate the sum of the residual squares sums (or RRSS) accordingly. Denote the smallest one by SRSS_h or RRSS_h. If $\text{SRSS}_h < \text{SRSS}_0$ (or $\text{RRSS}_h < \text{SRSS}_0$ in robust procedure), redefine $\mathscr{C}_j = \mathscr{C}_j - \{i\}$, $\mathscr{C}_h = \mathscr{C}_h + \{i\}$, and set $\text{SRSS}_0 = \text{SRSS}_h$ (or $\text{RRSS}_0 = \text{RRSS}_h$). Otherwise return to the beginning of (ii).

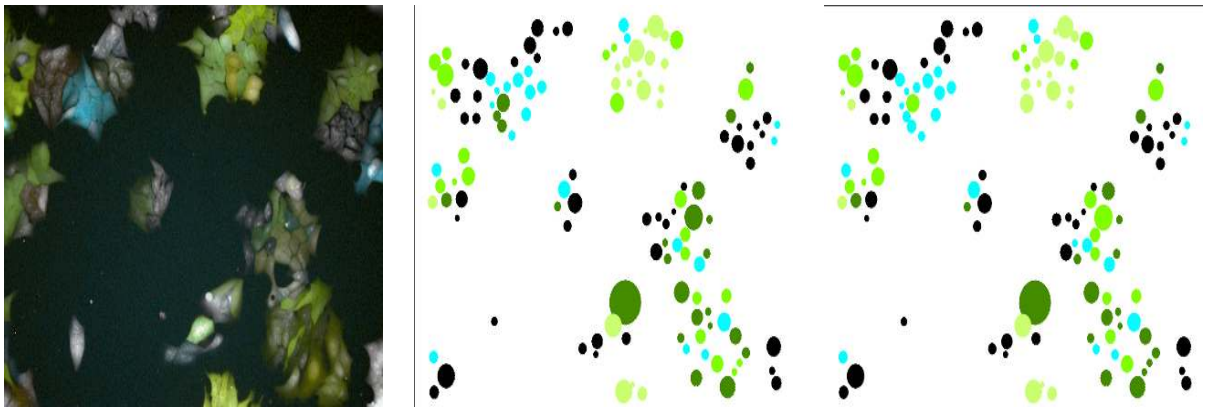
(iii) Repeat (ii) until the objective function (1.8) or (1.9) does not decrease any further, which means no observation relocation is necessary and the optimal clustering is achieved.

1.3.4 Clustering RGB marked cancer cell data

In this section, we use a small data set that contains 128 cells, as an illustrative example showing the performance of the proposed clustering method. Figure 1.3 (a) shows the raw image data.

Clustering result

Recall that the iterative algorithm described in Section 1.3.3 guarantees only local minimization, thus it is important to start from a good initial partition. We take the result from the robust K-means discussed in Section 1.2.1 as an initial clustering. In Figure 1.3 (b), we show the clustering result of the robust K-means with $\alpha = 0.1$, in which the colorectal cancer cells are represented by the colored dots. The five different colours indicate five clusters. Each colour is specified as the combination of the three channels (red, green and blue) according to the cluster center, and spatial locations of the dots are the same with the cells, so as to resemble the real image in (a). The size of the dots indicate the area of the corresponding cell. In (c), we show the final clustering result obtained by the LS regression clustering, where the response is the three dimensional vector of standardised RGB colour intensities and explanatory variable is the cell area. The difference between (b) and (c) suggests that the cell morphology information used in the regression clustering plays a role in separating different cell types.



(a) (b) (c)

Figure 1.3: (a) Raw spatial data on 128 colorectal cancer cells. (b) Initial clustering by robust K-means, with cells represented by the coloured dots. Colours indicate clusters, size of the dots suggests the area of the cells. (c) Clustering result using LS multivariate regression clustering with the cell area being the explanatory variable.

Figure 1.4 shows a scattered plot of observed 3 dimensional responses, in which the position of each dot represents the 3-dimensional colour vector of a cell, and the colour of a dot indicates which cluster this cell belongs to. For example, the cluster at the bottom

left of the image contains cells that express low fluorescent protein intensities in all three channels, therefore the cluster is coloured as (almost) black.

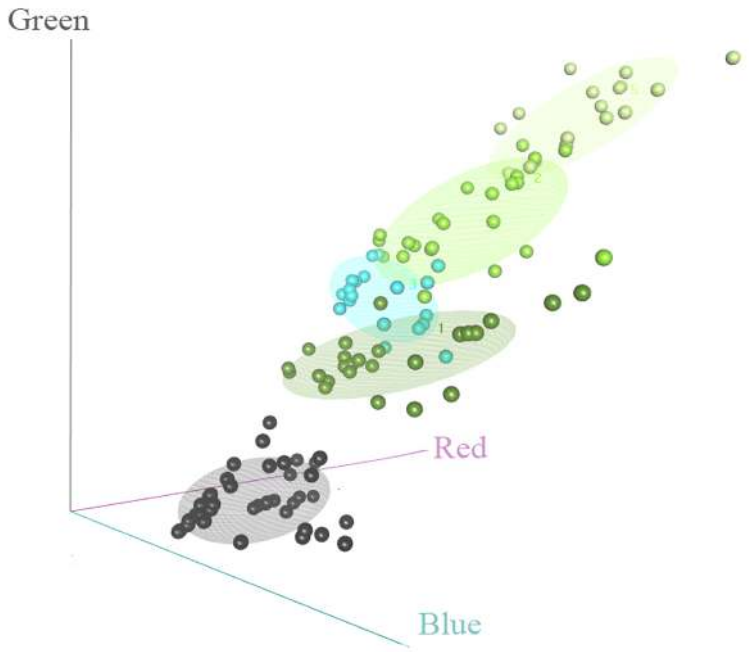


Figure 1.4: 3D-scattered plot of the clustered RGB intensities of the 128 cells. Colors of the points show the same 5 clusters shown in

In Table 1.1, we summarize the outcome for the LS regression clustering.

Cluster	Mean (Red, Green, Blue)	$\widehat{SD}$ (Red, Green, Blue)	Cluster size
1	(4.99, 5.02, 5.78)	(0.10, 0.17, 0.25)	25
2	(5.66, 5.83, 5.78)	(0.23, 0.26, 0.18)	23
3	(5.40, 5.36, 5.57)	(0.12, 0.18, 0.14)	20
4	(4.55, 4.19, 5.52)	(0.28, 0.16, 0.11)	41
5	(6.32, 6.50, 5.92)	(0.22, 0.24, 0.18)	19

Table 1.1: Summary statistics of clusters obtained from the multivariate LS regression clustering by including the cell area covariate. Table columns show sample means and standard deviations of cluster centers, as well as the number of cells in each cluster.

We only show clustering result of the LS regression clustering, because it outperforms the RM method in our case, which we will show later in this section.

Selecting the number of clusters

To select the optimal number of clusters, we used the information criterion function (1.6) for LS and (1.7) for RM, with $q(k) = k$, where k is the unknown number of clusters that

we are seeking for. Figure 1.5 shows the optimal numbers of clusters using $A_n = \log\log n$ (C1) and $A_n = \log n$ (C2) for both clustering approaches. Robust clustering is carried out using Huber's discrepancy function (1.5) with the tuning constant $c = 1.345$ being chosen. The resulting optimal number of clusters based on C1 is 5 by both LS and RM regression clustering criterion, which is compatible with biological considerations.

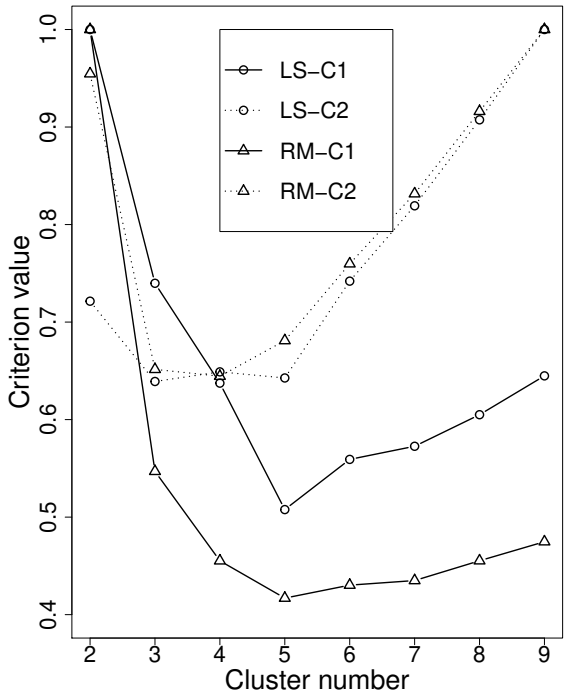


Figure 1.5: Information criteria for Selecting of the number of clusters. Criterion function is specified in (1.6) for LS and (1.7) for RM, with A_n equal to $\log\log n$ (C1) and $\log n$ (C2). All criterion values are scaled to between 0 and 1.

Clustering assessment

Finally, we assess the performances of the LS and RM regression clustering, and compare them with that of the robust K-means method. The prediction strength (PS) statistic introduced by Tibshirani and Walther (2005) is used for the assessment.

For a candidate number of clusters k ($k = 5$ in our case), let $\hat{\mathscr{C}}_{te} = \{\hat{\mathscr{C}}_{te,1}, \ldots, \hat{\mathscr{C}}_{te,k}\}$ denote the partition of the test set resulting from regression clustering on all the data. Let $n_1, \ldots, n_k$ be the number of observations in these clusters. Let $\hat{\mathscr{C}}_{tr}$ be the partition of the test set resulting from regression clustering on the training set. In particular, in the latter case each data point in the test set is clustered using (1.3) with $\hat{\beta}_i$, $i = 1, \ldots, k$ produced by the training set.

Following notations of Tibshirani and Walther (2005), denote $D[\hat{\mathscr{C}}_{tr}, \hat{\mathscr{C}}_{te}]$ as the $n \times n$ co-membership matrix, with ii'th element $D[\hat{\mathscr{C}}_{tr}, \hat{\mathscr{C}}_{te}]_{ii'} = 1$, if a pair of observations i and

i' that belong to the same cluster in $\hat{\mathscr{C}}_{te}$ (i.e. $i \neq i' \in \hat{\mathscr{C}}_{te,j}, j = 1,\ldots,k$) also fall into the same cluster in $\hat{\mathscr{C}}_{tr}$, and 0 otherwise. The prediction strength statistic can be written as

$$PS = min_{1\leq j\leq k} \frac{1}{n_j(n_j-1)} \sum_{i\neq i'\in \hat{\mathscr{C}}_{te,j}} D[\hat{\mathscr{C}}_{tr}, \hat{\mathscr{C}}_{te}]_{ii'}.$$

Therefore, the prediction strength is the proportion of observation pairs in the worst performing test cluster whose clustering results remain unchanged when clustering them by the training set clustering rule. Clearly, a regression clustering result has higher predictive power if the associated PS is higher.

For our data, we assess the clustering performance by cross-validation using 4 random partitions of our sample. Cross-validated prediction strength values for K-means, LS and RM regression clustering methods are 0.44, 0.80 and 0.66, respectively. This suggests that the LS regression clustering is superior to the to robust K-means. Moreover, due to the absence of strong deviations from the multivariate normal model for these data, the out-of-sample prediction strength of the LS regression clustering is larger than that of the robust RM regression clustering approach. Thus, we adapt the LS regression clustering to our datasets.

In the remaining chapters of this thesis, we assume the data has already been clustered and focus only on modelling the spatio-temporal interactions on growth between cell types.

1.4 A selective overview on Spatio-Temporal Models

1.4.1 Conditionally Auto-Regressive (CAR) Models

To model spatio-temporal data, one could choose to approximate the spatio-temporal process by a spatial process of time series, that is, to view the process as a multivariate spatial process where the multivariate dependencies are inherited from temporal dependencies. In other words, it can be seen as a temporal extension of spatial processes.

The most popular way of developing a spatial process is through the conditionally auto-regressive (CAR) model proposed by Besag (1974). Under Gaussian assumption, the general form of a CAR can be expressed as

$$Z_i | Z_{N(i)} \sim N\left(f\left(Z_{N(i)}\right), \tau_i^2\right), \tag{1.10}$$

where i indices for location, $N(i)$ denotes the neighbourhood of i and function $f(\cdot)$ is most often taken as a weighted sum of observations in the neighbourhood $\sum_{j\neq i} w_{ij} Z_j$.

Waller et al. (1997) extend the CAR model into a spatio-temporal setting by allowing spatial effects to vary across time. Specifically, let $Y_{i,t}$ be observed data at time t, $Y_{i,t} =$

$X_{i,t}\gamma + \theta_{i,t} + Z_{i,t}$, where $X_{i,t}$ are covariates, $\theta_{i,t}$ denotes a random effect variable with zero expectation and $Z_{i,t}$ denotes spatial effects that follow a CAR model in (1.10) for each t independently. However, the model lacks a specification of temporal dependency, as also noted by Knorr-Held (2000), who discussed a variety of possible spatio-temporal interactions, although the focus is still on single outcomes.

The extensions to multivariate spatio-temporal settings has been more recent, Quick et al. (2015) proposed a non-separable multivariate space-time CAR model (MSTCAR) on Gaussian data, which itself is a special case of the multivariate CAR models of Gelfand and Vounatsou (2003). Quick et al. (2017) generalize the MSTCAR model to accomodate Poisson data of multiple groups, where both temporal and between group dependencies are modelled as multivariate dependencies.

$$Y_{i,t}^{(c)} = X_{i,t}^{(c)}\gamma + Z_{i,t}^{(c)} + \varepsilon_{i,t}^{(c)},$$

where $\boldsymbol{Z} \sim \text{MCAR}(1,\Sigma)$, that is

$$\boldsymbol{Z}_{i,\cdot}^{(\cdot)} | \boldsymbol{Z}_{N(i),\cdot}^{(\cdot)}, \Sigma \sim N\left(\sum_{j \in N(i)} Z_{j,\cdot}^{(\cdot)} / |N(i)|, \frac{1}{|N(i)|}\Sigma \right),$$

where Σ is a non-separable covariance matrix capturing temporal and between group correlations. Other works related to spatial process of time series include Sans et al. (2008) and Quick et al. (2016), see Carlin et al. (2014) for a more complete coverage.

1.4.2 Latent Process Time Series Models

Alternatively, one also think of the process as a time series of spatial process, or a spatial extension of time series. This is the approach we take in our spatio-temporal modelling. The underlying notion is that "the temporal dependence is more natural to model than the spatial dependence" (Cressie and Wikle, 2015).

Gaussian data response

Data for which Gaussian distribution is assumed is typically modelled by a multivariate Gaussian process with an additive zero-mean error, called the data model:

$$Y_{i,t}^{(c)} = Z_{i,t}^{(c)} + \varepsilon_{i,t}^{(c)},$$

where $Y_{i,t}^{(c)}$ denotes observed data with indices i for locations, t for time and c for groups or variables, and $Z_{i,t}^{(c)}$ is a Gaussian process. The major difference between latent process models is the specification of $Z_{i,t}$, which is usually called the process model. For simplicity, we abuse the use of ε and denote all independent zero-mean errors with the same

notation, although those in different equations should be different variables.

Wikle et al. (1998) propose the hierarchical space-time model for climate data: $Z_{i,t} = \alpha_i + M(t;\gamma_i) + U_{i,t}$, where α_i represents site specific intercept, $M(t;\gamma_i)$ models seasonal effects for each site and $U_{i,t}$ is a spatio-temporal dynamic process, which follows the VAR(1) model introduced by Cressie and Wikle (2015) in Section 6.4.2 as a special case of the spatial-temporal autoregressive moving-average (STARMA) model. Specifically, the VAR(1) model has the form

$$\boldsymbol{U}_t = \boldsymbol{H}_t \boldsymbol{U}_{t-1} + \boldsymbol{\varepsilon}_t, \tag{1.11}$$

where $\boldsymbol{H}_t$ is a matrix of regression coefficients.

Shaddick and Wakefield (2002) implement a similar model on pollutant data, different in that: i) external covariates, $X_{i,t}^{(c)}$, are included; ii) $\boldsymbol{Z}_t$ follows the VAR(1) model directly with $\boldsymbol{H}_t$ taken as an identity matrix; iii) the presence of multiple variables (or pollutants). The process model is then written as

$$Z_{i,t}^{(c)} = X_{i,t}^{(c)}\gamma + Z_{i,t-1}^{(c)} + \varepsilon_{i,t}^{(c)}, \tag{1.12}$$

where c denotes the types of pollutants and $X_{i,t}^{(c)}$ represents explanatory variables, such as the temperature and location of each site.

Bradley et al. (2015) propose a general model for multivariate spatio-temporal Gaussian data, the MSTM, and show the prediction performance in terms of space (i.e. prediction at unobserved locations). The process model is expressed as

$$Z_{i,t}^{(c)} = X_{i,t}^{(c)}\gamma + \boldsymbol{S}_{i,t}^{(c)}\boldsymbol{U}_t + \varepsilon_{i,t}^{(c)}, \tag{1.13}$$

where the column vectors of $\boldsymbol{S}_{i,t}^{(c)}$ are Moran's I basis functions, and $\boldsymbol{U}_t$ follows the same VAR(1) model structure in (1.11). However $\boldsymbol{H}_t$ does not contain any parameter to be estimated, instead it is a carefully chosen propagator matrix. Such structure allows effective rank reduction for high dimensional data. Bradley et al. (2016) implement this model on survey data about unemployment statistics.

Poisson data response

However, assessing and modelling multivariate dependence when the outcomes are discrete can be challenging, since traditional methods for detecting dependence in additive Gaussian error are inappropriate. To address this issue, the most common approach is to assume that the conditional expectation of the observed process (on log-scale), as a latent process, has a nice distribution, for example, multivariate Gaussian. The spatial dependency is then modelled by the latent process, which cannot be observed directly

and which evolves independently of the past and present values of the observed process. There are extensive work relating to latent spatial-temporal models under the Bayesian framework. Typical examples are shown as follows.

Data models for count data usually take the form

$$\begin{aligned} &Y_{i,t}^{(c)} | \boldsymbol{Y}_{t-1} \sim \text{Poisson}(\lambda_{i,t}^{(c)}), \\ &\text{where } \; g\left(\lambda_{i,t}^{(c)}\right) = Z_{i,t}^{(c)} + \varepsilon_{i,t}^{(c)} \end{aligned}$$

and $g(\cdot)$ is the link function, which is log in this case.

This conveniently maps the problem of modelling Poisson data into that of a latent Gaussian process in the conditional expectation $\lambda_{i,t}^{(c)}$. Thus, the process models for Gaussian data from previous paragraphs can also be adopted in this context. For example, Mugglin et al. (2002) use a process model similar to (1.12): $Z_{i,t} = \log(n_i) + X_{i,t}\gamma + U_{i,t}$, where $U_{i,t}$ is defined the same as (1.11). Holan and Wikle (2015) propose a process model that resembles that of (1.13) except for the different choice of basis functions and matrix $\boldsymbol{H}_t$. Bradley et al. (2017) extend the MSTM proposed by Bradley et al. (2015) to accomodate count-valued data, by introducing a conjugate distribution of Poisson distribution, the log-gamma distribution for the random effects in (1.13), $\boldsymbol{U}_t$.

Remarks

Following Cox et al. (1981), in the analysis of time series data, this type of modelling approach is termed as parameter driven models. Unfortunately, as stated by several authors, parameter driven models requires considerable computational effort and are not yet ready for complex model settings (Davis et al., 2003; Benjamin et al., 2003; Schrödle et al., 2012; Dunsmuir et al., 2015).

1.4.3 Observation-driven Time Series Models and Spatial Extensions

In contrast, in the other type of models termed by Cox et al. (1981), called observation-driven models, time dependence arises because the conditional expectation of the outcome given the past depends explicitly on the past values. Zeger and Qaqish (1988) review various observation-driven time series models with a quasi-likelihood estimation. Fokianos and Tjøstheim (2011) develop and study the probabilistic properties of a log-linear autoregressive time series model for Poisson data,

$$\lambda_t = \alpha_0 + \alpha\lambda_{t-1} + \beta\log(y_{t-1}+1),$$

as an extension of the model considered by Fokianos et al. (2009). Although λ_{t-1} is present in the autoregressive terms, it can be fully expressed by $\log(y_{t-l}+1), l > 1$,

through repeated substitution. The model therefore falls to the category of observational driven models. See Dunsmuir et al. (2015) and Kedem and Fokianos (2005) for a complete review.

Benjamin et al. (2003) propose a quite general class of models, called generalised autoregressive moving average (GARMA) models. The most general form of the generalised autoregressive moving average model, GARMA(p,q), is defined as

$$g(\mu_t) = X_t\alpha + \sum_{t'=1}^{p} \beta_{t'} H_{t'}(y_{t-t'}) + \sum_{t'=1}^{q} \gamma_{t'} D_{t'}(\mu_{t-t'}),$$

where $g(\cdot)$ is the canonical link function, $H_{t'}(\cdot)$ and $D_{t'}(\cdot)$ are known functions for all t'. The autoregressive part of the model proposed in Chapter 3 borrows the structure of the GARMA model .

However, spatial extension for discrete observational-driven time series is challenging. Unlike parameter-driven models, which impose continuous distributional assumptions on expectation of the observe process, observation-driven models face the difficulty of identifying the multivariate distribution of discrete variables. Literature about spatio-temporal models of this kind is relatively sparse. Held et al. (2005) propose a multivariate time series model

$$\lambda_{i,t} = \beta y_{i,t-1} + v_{i,t}, \tag{1.14}$$

where $y_{i,t}$ denote observed count data at time t and geographical region i, β is the autoregressive parameter and

$$\log v_{i,t} = \alpha_i + \sum_s (\gamma_s \sin(\omega_s t) + \delta_s \cos(\omega_s t)) \tag{1.15}$$

captures seasonal dependency, where ω_s are Fourier frequencies and α_i's are model parameters are allowed to vary across space. Yet, the model lacks specification of spatial dependency.

Paul et al. (2008) extended the second part of the model to allow seasonality terms to vary across regions by changing γ_s and δ_s in (1.15) to $\gamma_{i,s}$ and $\delta_{i,s}$ respectively. They also generalised the first part of the model (1.14) as

$$\lambda_{i,t} = \beta_i y_{i,t-1} + \phi_i \sum_{j \neq i} w_{ji} y_{j,t-l} + v_{i,t}, \tag{1.16}$$

so that different locations are allowed to have different autoregressive parameters, and the influence of $y_{j,t-l}$ for $j \neq i$ or lag $l > 1$ is quantified by model parameter ϕ_i. The weight w_{ji} could be set as an indicator of whether j within a neighbouring area of i, in this case, ϕ_i captures spatial dependences of previous time points with lag l, but still lacking the spatial correlations at the same time point.

Paul and Held (2011) further extend the model by introducing random effects. Specifically, they decompose unknown quantities in the model (including some model parameters), specifically $\beta_i, \phi_i, v_{i,t}$, additively on the log scale, for example $\log(\beta_i) = a^{(\beta)} + b_i^{(\beta)}$, where $a^{(\beta)}$ denotes intercept of β and $b_i^{(\beta)}$ is a random effect.

Note that these models are modelling directly on the conditional expectation of the count data, meaning they are using an identity link function, instead of the canonical log-link. Thus, it is required that the parameters are positive to ensure that the resulting conditional expectation is positive.

Remarks

While parameter driven models require complicated estimation techniques, observation models enable inferences in a (penalised) likelihood framework and therefore can be easily fitted even for quite complex regression models (Davis et al., 2003). Schrödle et al. (2012) proposed a parameter-driven version of spatio-temporal model, which is very similar to the observation-driven model proposed by Paul et al. (2008) in (1.16):

$$\lambda_{i,t} = \beta\lambda_{i,t-1} + \phi\sum_{j\neq i} w_{ji}\lambda_{j,t-1} + \varepsilon_{i,t},$$

where $\varepsilon_{i,t}$ is a white noise. They then compare the performance of both versions and conclude that the parameter-driven models perform slightly better in terms of prediction in some cases, however, while the computation time for the observation-driven model is mostly less than a second, fitting a parameter-driven model takes several hours if it ever converges, because of the complexity with the latent autoregressive process. Besides, their model contains only five parameters, while in our application, the number of parameters of interest grows quadratically with the number of cell populations, which makes the parameter-driven models intractable even with a moderate number of cell populations. For this reason, we choose to work with a spatial extension of observation-driven time series.

1.5 Thesis Outline

In this thesis, we develop modelling tools for the analysis of multivariate spatio-temporal count data on lattice. Chapter 2 is directly motivated by the RGB marked cell data imaged on a high-content imager (Operetta, PerkinElmer) in longitudinal experiments. We are primarily concerned about how interactions between cell types may affect their growth. To do this, we divide the image into a regular lattice and transfer each image data into a spatial lattice data of cell counts in each tile. We propose a conditional autoregressive model with Poisson response, where model parameters can directly be interpreted

as impacts between different cell populations in neighbouring tiles. Numerical results from simulated and real data confirm the validity of the proposed approach in terms of prediction, goodness-of-fit and estimation accuracy.

In Chapter 3, we extend the model in Chapter 2 to incorporate correlation parameters through a Gaussian copula regression model, where temporal dependences are modelled through the marginal expectations by an log-linear autoregressive model while both spatial and cross groups correlations are incorporated by the Gaussian copula. Poisson and Negative binomial marginal distributions are implemented. The model allows for both positive and negative correlations and could potentially be generalized to handle any correlation structures. We provide pairwise composite likelihood inference tools in closed forms, the proposed methodology strikes a good balance between computational feasibility and statistical accuracy, which we demonstrate with examples.

In Chapter 4, we implement an information criterion-based model selection method in the applications of the copula-based model proposed in Chapter 3. The method guarantees to converge to the model with the lowest information criterion and is efficient enough to handle large candidate model set. Although in this thesis we only focus on selection of the copula-based model, the proposed methodology is extremely flexible and can be applied to a wide variety of regression based models as an efficient variable subsetting toolkit, as long as the information criterion is properly chosen.

To make our numerical results reproducible and our methodologies available to other practitioners, we present in Chapter 5 an R package implementing estimation and selection tools for the proposed copula-based model. The real data analysed in this thesis is also made available in the package. We demonstrate the usage of our package with examples and compare the performance with other packages on both simulated and real data. Although to our knowledge there is no existing package that implements exactly the same type of model as ours, we obtain similar results with other package functions on some special cases, confirming the correctness of our computations. Besides, our package is usually faster than other packages when performing the same task. Finally, we provide for non-R users a web application using the package Shiny (Chang et al., 2018), in which we offer tools for visualising data spatially and temporally, as well as estimation and selection tools for the simpler model proposed in Chapter 2. The application can handle relatively large data set, in the built-in example data, there are 13 groups with over 200 parameters, yet estimation typically takes only a few seconds.

In the final chapter of this thesis, we present an overall summary of the advantages and limitations of the research. We also discuss future research directions that lead on from this work.

Chapter 2

A Spatio-Temporal Model for Longitudinal Count Data on Multicolour Cell Growth

2.1 Introduction

In this chapter, we develop a conditional spatial-temporal model for grouped count data on tiled images, and provide its application in the context of longitudinal cancer cell monitoring experiments. Our model enables us to measure the effect on the growth rate of each cell population and changes due to local cross-population interactions.

Specifically, in order to describe the spatial distribution for different cell types, we first divide an image into a number of contiguous regions (tiles) to form a regular lattice structure as shown in Figure 2.1. We then record the frequency of cells of different colors in each tile at subsequent time points, and based on which we model the spatial and temporal dependencies of the cell growth. Finally, we propose a Poisson model with intensity modelled as a log-linear form similar to those in Knorr-Held and Richardson (2003) and Fokianos and Tjøstheim (2011), and we quantify spatio-temporal impacts of different cell populations in neighboring tiles through model parameters, as illustrated in Figure 2.1 (b). Impacts are allowed to be positive or negative, and unlike latent models that describe between group dependence through a covariance matrix, influences do not have to be symmetrical in our model.

Since the model complexity can be potentially very large in the presence of many cell types, it is also important to address the question of how to select an appropriate model by retaining only the meaningful spatio-temporal interactions between cell populations We carry out a model selection using the common model selection criteria for parametric models, the Akaike and the Bayesian information criteria (AIC and BIC). For this chapter only, we search through all candidate models to find the one with the lowest criterion

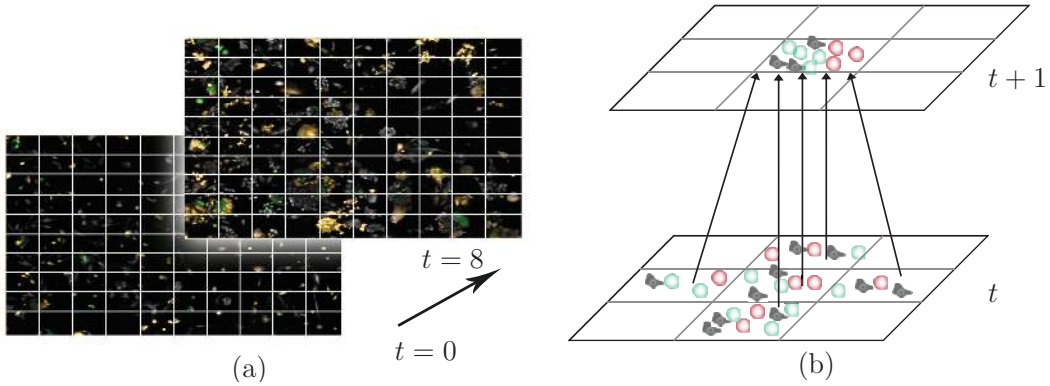


Figure 2.1: (a) Microscope images for the cancer cell growth data obtained from a high-content imager (Operetta, Perkin Elmer) at the initial and final time points of the experiment. In each image, colors for non-fluorescent fibroblasts, as well as red and green fluorescent cancer cells are merged. (b) Illustration of the local structure for the model in (2.1). The two planes correspond to 3×3 tiles at times t and $t+1$. The average number of cells of color c in a given tile at time $t+1$ is assumed to depend on the number of cells of other colors in contiguous neighboring tiles at time t.

value, since we have a reasonably small number of parameters in our experiments. But we propose in Chapter 4 a more elegant way of searching for the best model when a large candidate model set is expected.

The remainder of this chapter is organized as follows. In Section 2.2, we introduce the conditional spatio-temporal lattice model for grouped count data and develop maximum likelihood inference tools. In the same section, we discuss the asymptotic properties of our estimator and standard errors. In Section 2.3, we study the performance of the estimator using simulated data. In Section 2.4, we apply our method to analyze datasets from two in-vitro experiments: One where cancer cells are co-cultured with fibroblasts, and one where individually recognisable cloned cancer cell populations are cultured together in different combinations. In Section 2.5, we conclude and give final remarks.

2.2 Methods

2.2.1 Multicolour spatial autoregressive model on the lattice

Let $\mathscr{L} \in \mathbb{N}^2$ be a discrete lattice. In the context of our application, the lattice is obtained by tilling a microscope image into $n_{\mathscr{L}}$ tiles, denoted by $\mathscr{L}_n (\subset \mathscr{L})$. The total number of tiles $n_{\mathscr{L}}$ is a monotonically increasing function of n. One can choose various forms of lattice, for example, the regular or hexagonal lattices. For simplicity, we tile the image into $n \times n$ regular rectangular tiles, which makes $n_{\mathscr{L}} = n^2$. An example of a tiled image with $n = 10$ is shown in Figure 2.1 (a). Denote a pair of neighbouring tiles $\{i, j\}$ with $i \sim j$, if tiles i and j share the same border or coincide ($i = j$). Each tile may contain

cells of different colours; thus, we let $\mathscr{C} = \{1,\ldots,n_{\mathscr{C}}\}$ be a finite set of colours and denote by $n_{\mathscr{C}}$ the total number of colours. Let $\boldsymbol{Y} = \{\boldsymbol{Y}_t, t = 1,\ldots,T\}$ be the sample of observations where $\boldsymbol{Y}_t = \{\boldsymbol{Y}_t^{(c)}, c \in \mathscr{C}\}$ is the collection of observations at time point t, and $\boldsymbol{Y}_t^{(c)} = (Y_{1,t}^{(c)},\ldots,Y_{n_{\mathscr{L}},t}^{(c)})^\top$ is the vector of observed frequencies for color c on the lattice $\mathscr{L}_n$ at time t. The joint distribution for the spatio-temporal process on the lattice is difficult to specify, due to local spatial interactions for neighboring tiles and global interactions occurring at the level of the entire image. Therefore, in this chapter, we focus mainly on modelling the conditional marginal expectations of $\boldsymbol{Y}$.

Due to the fact that cells tend to be clustered together due to the cell division process and other biological mechanisms; it is not uncommon to observe low counts in a considerable portion of tiles, thus, a Gaussian assumption would not be appropriate.

We suppose that the count for the ith tile $Y_{i,t}^{(c)}$ follows a marginal Poisson distribution $Y_{i,t}^{(c)}|\boldsymbol{Y}_{t-1} \sim \text{Pois}(\lambda_{i,t}^{(c)})$. The intensity is modelled through a canonical log-link $v_{i,t}^{(c)} = \log\lambda_{i,t}^{(c)}$, where $v_{i,t}^{(c)}$ takes the following spatial autoregressive form:

$$v_{i,t}^{(c)} = \alpha^{(c)} + \sum_{c' \in \mathscr{C}} \beta^{(c|c')} S_{i,t-1}^{(c')}, \tag{2.1}$$

$$S_{i,t-1}^{(c')} = \frac{1}{n_i} \sum_{i \sim j:\ j \in \mathscr{L}_n} \log\left(1 + Y_{j,t-1}^{(c')}\right), \tag{2.2}$$

for all $c \in \mathscr{C}, t = 1,\ldots,T$, with $n_i = \{\#j : i \sim j, j \in \mathscr{L}_n\}$ being the number of tiles in a neighbourhood of tile i. Although we are adopting the regular grids for simplicity, the model is readily applicable to other tiling strategies. Changing the tiling strategy would only change the realisations of $S_{i,t-1}^{(c')}$ in (2).

Here, we assume that the conditional count for different tiles at time t is independent conditioning on information from $t-1$, i.e.

$$P\big(Y_{i,t}^{(c)} Y_{j,t}^{(c')} | \boldsymbol{Y}_{t-1}\big) = P\big(Y_{i,t}^{(c)} | \boldsymbol{Y}_{t-1}\big) P\big(Y_{j,t}^{(c')} | \boldsymbol{Y}_{t-1}\big),$$

for all $c, c' \in \mathscr{C}, t = 1,\ldots,T$, and $i, j \in \mathscr{L}_n, i \neq j$. This does not suggest that they ($Y_{i,t}^{(c)}$ and $Y_{j,t}^{(c')}$) are independent, but rather that their spatio-temporal dependence is due to the structure of intensity $\lambda_{i,t}^{(c)}$ in (2.1). Conditional independence is a commonly used assumption for spatio-temporal models in a non-gaussian setting Waller et al. (1997); Wikle and Anderson (2003), since it's exceedingly difficult to work with multivariate non-Gaussian distribution Cressie and Wikle (2015).

The elements of the parameter vector $\boldsymbol{\alpha} = (\alpha^{(1)},\ldots,\alpha^{(n_{\mathscr{C}})})^\top$ correspond to a baseline average count for cells of different colours. The spatio-temporal interactions are regressed on the statistic $S_{i,t-1}^{(c')}$ in (2.2), which essentially counts the average number of cells of colour c' in the neighbourhood of tile i at time $t-1$. Hence, the autoregressive parameter

$\beta^{(c|c')}$ is interpreted as positive or negative change in the average number of cells with colour c, due to interactions with cells of colour c' in neighbouring tiles. A positive (or a negative) sign of $\beta^{(c|c')}$ means that the presence of cells of colour c' in neighboring tiles promotes (or inhibits) the growth of cells of colour c. The spatio-temporal effects $\beta^{(c|c')}, c, c' \in \mathscr{C}$, are collected in the $n_{\mathscr{C}} \times n_{\mathscr{C}}$ weighted incidence matrix $\mathscr{B}$. This may be used to generate weighted directed graphs, as shown in the example of Figure 2.2, where the nodes of the directed graph correspond to cell types, and the directed edges are negative or positive spatio-temporal interactions between cell types.

Equation (2.1) could be extended to some more specific form, for example, $v_{i,t}^{(c)} = \alpha^{(c)} + \sum_{c' \in \mathscr{C}} \beta_1^{(c|c')} S_{i,t-1}^{(c')} + \beta_0^{(c|c')} \log\left(1 + Y_{i,t-1}^{(c')}\right)$, where $\beta_1^{(c|c')}$ are interpreted as the effect of cells of color c' from neighbouring (but not the same) tiles have on the growth of cells with color c, while $\beta_0^{(c|c')}$ as the effect of cells of color c' from the same tile. However, we stick to the model in (2.1) because we have no evidence showing that the more complex model is advantageous from model selection view point.

We choose to work with a log-linear form for the autoregressive equation of $v_{i,t}^{(c)}$ in Equation (2.1), where we apply a logarithmic transform and add 1 to the counts at time $t-1$, $Y_{i,t-1}^{(c)}$. It offers several advantages compared to the more commonly used linear form. First, $\lambda_{i,t}^{(c)}$ and $Y_{i,t-1}^{(c)}$ are transformed on the same scale. Moreover, this model can accommodate both positive and negative correlations, while it is not possible to account for positive association in a stationary model if past counts are directly included as explanatory variables. For example, with the model $v_{i,t} = \alpha + \beta Y_{i,t-1}$ for a single colour, the intensity would be $\lambda_{i,t} = \exp(\alpha)\exp(\beta Y_{i,t-1})$, which may lead to instability of the Poisson means if $\beta > 0$ since $\lambda_{i,t}$ is allowed to increase exponentially fast. Finally, adding 1 to $Y_{i,t-1}^{(c)}$ is for coping with zero data values, since $\log(Y_{i,t-1}^{(c)})$ is not defined when $Y_{i,t-1}^{(c)} = 0$, which arises often, and it maps zeros of $Y_{i,t-1}^{(c)}$ into zeros of $\log(1 + Y_{i,t-1}^{(c)})$.

2.2.2 Likelihood inference

Let $\boldsymbol{\theta}$ be the overall parameter vector $\boldsymbol{\theta} = (\boldsymbol{\alpha}^\top, \mathrm{vec}(\mathscr{B})^\top)^\top \in \mathbb{R}^p$, where $\boldsymbol{\alpha}$ is a $n_{\mathscr{C}}$-dimensional vector defined in Section 2.2.1 and $\mathscr{B}$ is a $n_{\mathscr{C}} \times n_{\mathscr{C}}$ matrix of colour interaction effects, $p = n_{\mathscr{C}}(1 + n_{\mathscr{C}})$ is the total number of parameters. Then the maximum likelihood estimator(MLE) is written as

$$L_n(\boldsymbol{\theta}) = \prod_{t=1}^{T} \prod_{c \in \mathscr{C}} \prod_{i \in \mathscr{L}_n} P(Y_{i,t}^{(c)} | \boldsymbol{Y}_{t-1}; \boldsymbol{\theta})^{w_{i,t}^{(c)}} = \prod_{t=1}^{T} \prod_{c \in \mathscr{C}} \prod_{i \in \mathscr{L}_n} \left(e^{-\lambda_{i,t}^{(c)}(\boldsymbol{\theta})} \frac{\lambda_{i,t}^{(c)}(\boldsymbol{\theta})^{y_{i,t}^{(c)}}}{y_{i,t}^{(c)}!} \right)^{w_{i,t}^{(c)}}, \quad (2.3)$$

where $\lambda_{i,t}^{(c)}(\boldsymbol{\theta})$ is the expected number of cells with colour c in tile i at time t defined in (2.1). The MLE $\hat{\boldsymbol{\theta}}$ is obtained by maximizing the log-likelihood function

$$\ell_n(\boldsymbol{\theta}) = \sum_{i\in\mathscr{L}_n}\sum_{t=1}^{T}\sum_{c\in\mathscr{C}}\left[Y_{i,t}^{(c)}v_{i,t}^{(c)}(\boldsymbol{\theta}) - \exp\left\{v_{i,t}^{(c)}(\boldsymbol{\theta})\right\}\right], \tag{2.4}$$

where $v_{i,t}^{(c)}(\boldsymbol{\theta}) \equiv \log\lambda_{i,t}^{(c)}(\boldsymbol{\theta})$. Equivalently, $\hat{\boldsymbol{\theta}}$ is formed by solving the estimating equations

$$0 = \boldsymbol{u}_n(\boldsymbol{\theta}) \equiv \frac{1}{n_{\mathscr{L}}}\nabla\ell_n(\boldsymbol{\theta}) = \frac{1}{n_{\mathscr{L}}}\sum_{i\in\mathscr{L}_n}\sum_{t=1}^{T}\boldsymbol{\gamma}_{i,t}(\boldsymbol{\theta})\otimes\nabla\boldsymbol{v}_{i,t}, \tag{2.5}$$

where $\boldsymbol{\gamma}_{i,t}(\boldsymbol{\theta}) = \left(y_{i,t}^{(1)} - \exp\left\{v_{i,t}^{(1)}(\boldsymbol{\theta})\right\}, \ldots, y_{i,t}^{(n_{\mathscr{C}})} - \exp\left\{v_{i,t}^{(n_{\mathscr{C}})}(\boldsymbol{\theta})\right\}\right)$, $\otimes$ denotes the Kronecker product, ∇ is the gradient operator with respect to $\boldsymbol{\theta}$ and $\nabla\boldsymbol{v}_{i,t} \equiv \nabla\boldsymbol{v}_{i,t}^{(c)}(\boldsymbol{\theta}) = (1, S_{i,t-1}^{(1)}, \ldots, S_{i,t-1}^{(n_{\mathscr{C}})})^{\top}$. The solution to Equation (2.5) is obtained by a standard Fisher scoring algorithm, which is found to be stable and converges fast in all our numerical examples.

Finally, in practical applications it is also important to address the question of how to select an appropriate model by retaining only the meaningful spatio-temporal interactions between cell populations, and avoid over-parametrized models. Model selection plays an important role by balancing goodness-of-fit and model complexity. Here, we select non-zero model parameters based traditional model selection approaches: the Akaike Information criterion, $AIC = -2\ell(\hat{\boldsymbol{\theta}}) + 2p$, and the Bayesian information criterion, $BIC = -2\ell(\hat{\boldsymbol{\theta}}) + p\log(|n_{\mathscr{L}}T|)$.

2.2.3 Asymptotic properties and standard errors

In this section, we overview the asymptotic behavior of the estimator introduced in Section 2.2.2. In typical longitudinal experiments, the number of time points seldom go beyond 50 due to experimental, storage and processing cost, while $n_{\mathscr{L}}$ can be relatively large. So we work under the framework where T is assumed to be finite, while $n_{\mathscr{L}}$ is allowed to grow to infinity. This reflects the notion that the statistician is allowed to choose an increasingly fine tiling grid as the number of cells increases. If the regularity conditions stated in the Appendix hold, then $\sqrt{n_{\mathscr{L}}}\boldsymbol{H}_n(\boldsymbol{\theta}_0)^{1/2}(\hat{\boldsymbol{\theta}}_n - \boldsymbol{\theta}_0)$ converges in distribution to a p-variate normal distribution with zero mean vector and identity variance, as $n_{\mathscr{L}} \to \infty$, with $\boldsymbol{H}_n(\boldsymbol{\theta})$ given in (2.6). Asymptotic normality of $\hat{\boldsymbol{\theta}}_n$ follows by applying the limit theorems for M-estimators for nonlinear spatial models developed by Jenish and Prucha (2009). One condition required to ensure this behaviour is that $\boldsymbol{Y}_t$ has constant entries at the initial time point $t = 0$, which is quite realistic since typically cells are seeded randomly at the beginning of the experiment. Our proofs mostly check α-mixing conditions and $\mathscr{L}_2$-Uniform Integrability of the score functions $\boldsymbol{u}_{i,t}(\boldsymbol{\theta})$ ensures a pointwise law of

large numbers, with additional stochastic equicontinuity, a uniform version of the law of large numbers required by Jenish and Prucha (2009).

The asymptotic asymptotic variance of $\hat{\boldsymbol{\theta}}$ is $\boldsymbol{V}_n(\hat{\boldsymbol{\theta}}) = \boldsymbol{H}_n^{-1}(\boldsymbol{\theta}_0)$, where $\boldsymbol{H}_n(\boldsymbol{\theta})$ is the $p \times p$ Hessian matrix

$$\boldsymbol{H}_n(\boldsymbol{\theta}) = -E\left[\nabla^2 \ell(\boldsymbol{\theta})\right] = -E\left(\sum_{i \in \mathscr{L}_n} \nabla \boldsymbol{u}_i(\boldsymbol{\theta})\right), \tag{2.6}$$

with $\boldsymbol{u}_i(\boldsymbol{\theta}) = \boldsymbol{u}_{i,1}(\boldsymbol{\theta}) + \cdots + \boldsymbol{u}_{i,T}(\boldsymbol{\theta})$ being the partial score function for the ith tile. Direct evaluation of $\boldsymbol{H}(\boldsymbol{\theta})$ may be challenging since the expectations in (2.6) is intractable. Thus, we estimate $\boldsymbol{H}_n(\boldsymbol{\theta})$ by the empirical counterpart

$$\hat{\boldsymbol{H}}_n(\boldsymbol{\theta}) = \begin{pmatrix} \hat{\boldsymbol{H}}^{(1)}(\boldsymbol{\theta}) & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \hat{\boldsymbol{H}}^{(2)}(\boldsymbol{\theta}) & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \hat{\boldsymbol{H}}^{(n_{\mathscr{C}})}(\boldsymbol{\theta}) \end{pmatrix},$$

where

$$\hat{\boldsymbol{H}}^{(c)}(\boldsymbol{\theta}) = \sum_{i \in \mathscr{L}_n} \sum_{t=1}^{T} \exp\left[v_{i,t}^{(c)}(\boldsymbol{\theta})\right] [\nabla \boldsymbol{v}_{i,t}] [\nabla \boldsymbol{v}_{i,t}]^\top. \tag{2.7}$$

Note that the above estimators approximate the quantities in formula (2.6) by conditional expectations. Our numerical results suggest that the above variance approximation yields confidence intervals with coverage very close to the nominal level $(1-\alpha)$. Besides the above formulas, we also consider confidence intervals obtained by a parametric bootstrap approach. Specifically, we generate B bootstrap samples $\boldsymbol{Y}_{(1)}^*, \ldots, \boldsymbol{Y}_{(B)}^*$ by sampling at subsequent times from the conditional model specified in Equations (2.1) and (2.2) with $\boldsymbol{\theta} = \hat{\boldsymbol{\theta}}$. From such bootstrap samples, we obtain bootstrapped estimators, $\hat{\boldsymbol{\theta}}_{(1)}^*, \ldots, \hat{\boldsymbol{\theta}}_{(B)}^*$, which are used to estimate $\mathrm{Var}(\hat{\boldsymbol{\theta}}_0)$ by the usual covariance estimator $\hat{\boldsymbol{V}}_{boot}(\hat{\boldsymbol{\theta}}) = \sum_{b=1}^{B} (\hat{\boldsymbol{\theta}}_{(b)}^* - \overline{\boldsymbol{\theta}}^*)^2/(B-1)$, where $\overline{\boldsymbol{\theta}}^* = \sum_{b=1}^{B} \hat{\boldsymbol{\theta}}_{(b)}^*/B$. Finally, a $(1-\alpha)100\%$ confidence interval for $\boldsymbol{\theta}_j$ is obtained as $\hat{\boldsymbol{\theta}}_j \pm z_{1-\alpha/2}\{\hat{\boldsymbol{V}}\}_{jj}^{1/2}$, where z_q is the q-quantile of a standard normal distribution, and $\hat{\boldsymbol{V}}$ is an estimate of $\mathrm{Var}(\hat{\boldsymbol{\theta}})$ obtained by either Equation (2.7) or bootstrap resampling.

2.3 Monte Carlo simulations

In our Monte Carlo experiments, we generate data from a Poisson model as follows. At time $t = 0$, we populate $n_{\mathscr{L}}$ tiles using equal counts for cells of different colors. For $t = 1, \ldots, T$, observations are drawn from the multivariate Poisson model $Y_{i,t}^{(c)} | \boldsymbol{Y}_{t-1} \sim$

Poisson$(\lambda_{i,t}^{(c)}), c \in \mathscr{C}$. Recall that the rate $\lambda_{i,t}^{(c)}$ defined in Section 2.2.1 contains autoregressive coefficients $\beta^{(c|c')}$, which are collected in the $n_{\mathscr{C}} \times n_{\mathscr{C}}$ matrix $\boldsymbol{\mathscr{B}}$.

We assess the performance of MLE under different settings concerning the size and sparsity of $\boldsymbol{\mathscr{B}}$. Consider the three models with the following choices of $\boldsymbol{\mathscr{B}}$:

$$\boldsymbol{\mathscr{B}_1} = \begin{pmatrix} 0.7 & -0.7 & 0.7 \\ 0.7 & 0.7 & -0.7 \\ -0.7 & 0.7 & 0.7 \end{pmatrix}, \boldsymbol{\mathscr{B}_2} = \begin{pmatrix} 0.05 & -0.15 & 0.25 \\ 0.35 & 0.45 & -0.55 \\ -0.65 & 0.75 & 0.85 \end{pmatrix}, \boldsymbol{\mathscr{B}_3} = \begin{pmatrix} 0.7 & -0.7 & 0.7 \\ 0 & 0.7 & 0 \\ 0 & 0 & 0.7 \end{pmatrix}.$$

Denote Model i as the model corresponding to $\boldsymbol{\mathscr{B}}_i, i = 1,2,3$. In Model 1, all the effects in $\boldsymbol{\mathscr{B}}$ have the same size; in Model 2, the effects have decreasing sizes; Model 3 is the same as Model 1, but with some interactions exactly equal to zero.

We set $\alpha^{(1)} = \cdots = \alpha^{(n_{\mathscr{C}})} = -0.1$ for all three models, in which the parameter choices reflect the situation where the generated process $\boldsymbol{Y}$ has a moderate growth.

In Tables 2.1 and 2.2, we show results based on 1000 Monte Carlo runs generated from Models 1-3, for $n = 25, n_{\mathscr{C}} = 3$ and $T = 10$ and 25. In Table 2.1, we show Monte Carlo estimates of squared bias and variance of $\hat{\boldsymbol{\theta}}$. Both squared bias and variance of our estimator are quite small in all three models, and decrease as T gets larger. The variances of Model 2 are slightly larger than those in the other two models due to the increasing difficulty in estimating parameters close to zero.

	$T = 10$		$T = 25$	
	$\widehat{\text{Bias}}^2$	$\widehat{\text{Var}}$	$\widehat{\text{Bias}}^2$	$\widehat{\text{Var}}$
Model 1	0.45(0.57)	5.75(0.26)	0.29(0.32)	2.36(0.11)
Model 2	0.64(0.91)	9.66(0.42)	0.67(0.71)	4.45(0.20)
Model 3	0.77(0.97)	8.09(0.36)	0.52(0.51)	3.47(0.16)

Table 2.1: Monte Carlo estimates for squared bias ($\times 10^{-6}$) and variance ($\times 10^{-4}$) of the MLE from 1000 simulated runs with the number of time points $T = 10$ and 25. The three models differ in parameter settings described in Section 2.3. Simulation standard errors are shown in parenthesis.

In Table 2.2, we report the coverage probability for symmetric confidence intervals of the form $\hat{\boldsymbol{\theta}} \pm z_{1-\alpha/2}\widehat{sd}(\hat{\boldsymbol{\theta}})$, where z_q is the $q-$quantile for a standard normal distribution, with $\alpha = 0.01, 0.05, 0.10$. The standard error, $\widehat{sd}(\hat{\boldsymbol{\theta}})$, is obtained by the squared root of diagonal elements of $\boldsymbol{V}_n(\hat{\boldsymbol{\theta}})$ and the parametric bootstrap estimate, $\hat{\boldsymbol{V}}_{est}$ and $\hat{\boldsymbol{V}}_{boot}$, described in Section 2.2.3. The coverage probability of the confidence intervals are very close to the nominal level for both methods.

In Table 2.3, we show results for the model selection based on 1000 Monte Carlo

		$T=10$		$T=25$	
		$\hat{\boldsymbol{V}}_{boot}$	$\hat{\boldsymbol{V}}_{est}$	$\hat{\boldsymbol{V}}_{boot}$	$\hat{\boldsymbol{V}}_{est}$
$\alpha=0.01$	Model 1	98.6	99.0	98.9	99.0
	Model 2	99.0	99.0	98.8	98.9
	Model 3	98.9	99.0	98.9	98.9
$\alpha=0.05$	Model 1	94.2	95.2	94.9	95.0
	Model 2	95.2	95.1	95.0	95.3
	Model 3	95.4	95.5	94.9	95.1
$\alpha=0.10$	Model 1	89.2	90.3	90.1	90.3
	Model 2	90.6	90.0	89.7	90.0
	Model 3	90.6	90.6	90.2	90.2

Table 2.2: Monte Carlo estimates for the coverage probability of $(1-\alpha)\%$ confidence intervals $\hat{\boldsymbol{\theta}} \pm z_{1-\alpha/2}\widehat{sd}(\hat{\boldsymbol{\theta}})$, with $\widehat{sd}(\hat{\boldsymbol{\theta}})$ obtained from the parametric bootstrap ($\hat{\boldsymbol{V}}_{boot}$) and the estimated inverse Hessian matrix ($\hat{\boldsymbol{V}}_{est}$) specified in Section 2.2 and 2.3 respectively.

samples from Model 3 using the AIC and the BIC given in Section 2 for $n=25$ and $T=10,25$. We report Type A error (a term is not selected when it actually belongs to the true model) and Type B error (a term is selected when it is not in the true model). For both AIC and BIC model selection is more accurate for large T. As expected AIC tends to over select, and BIC outperforms AIC, with zero Type A error, and very low Type B error.

	$T=10$		$T=25$	
	Type A	Type B	Type A	Type B
AIC	0.00	10.00	0.00	10.38
BIC	0.00	0.22	0.00	0.20

Table 2.3: Monte Carlo estimates for % Type A error (a term is not selected when it actually belongs to the true model) and % Type B error (a term is selected when it is not in the true model) using AIC and BIC criteria. Results are based on 1000 Monte Carlo samples generated from Model 3 with $n=25$ and $T=10,25$.

Finally, we compare the performance of our model with the following Multivariate conditional autoregressive (MCAR) model proposed by Leroux et al. (2000):

$$Y_{i,t}^{(c)} \sim \text{Pois}\left(\exp\left(\boldsymbol{x}_{i,t}^{T}\boldsymbol{\beta}+\boldsymbol{Z}_i\right)\right),$$

where $\boldsymbol{Z}_i, i \in \mathscr{L}_n$ are random effects with conditional distribution

$$\boldsymbol{Z}_i|\boldsymbol{Z}_{-i} \sim N\left(\frac{\rho\sum_{j\sim i: j\in\mathscr{L}_n}\boldsymbol{Z}_j}{\rho n_i+1-\rho}, \frac{\boldsymbol{\Sigma}_Z}{\rho n_i+1-\rho}\right),$$

where ρ is a spatial autocorrelation parameter, with $\rho = 0$ corresponding to independence, while $\rho = 1$ corresponds to the intrinsic model, and $\boldsymbol{\Sigma}_Z$ is a $n_{\mathscr{C}}T \times n_{\mathscr{C}}T$ between variable covariance matrix, which is assumed to have no fixed structure, and n_i is the number of tiles in a neighbourhood of tile i as defined in Section 2.1. Let $\boldsymbol{\beta} = (\alpha^T, \text{vec}(\mathscr{B})^T)^T$ be a vector of regression parameters, where $\mathscr{B}$ is defined in Section 2.1 and α is the intercept. Let the covariate $\boldsymbol{x}_{i,t}$ be a $n_{\mathscr{C}}^2$-dimensional vector consists of $n_{\mathscr{C}}$ vectors: $(S_{i,t-1}^{(1)}, \ldots, S_{i,t-1}^{(n_{\mathscr{C}})})$, where $S_{i,t-1}^{(c)}$ carries the information from the neighbouring tiles on the previous time point, defined in (2).

An independent Gaussian prior, $N(0, 100000)$, is specified for each regression parameter in $\boldsymbol{\beta}$. A uniform prior on the unit interval, $U(0,1)$, is specified for ρ. For covariance matrix $\boldsymbol{\Sigma}_Z$, assume an inverse Wishart distribution with identity scale matrix and $n_{\mathscr{C}}T$ degree of freedom.

To evaluate the performance of MLE under our model and estimators obtained by the MCAR model, we generate 1000 set of data from Model 1 described in Section 3. Estimation of the MCAR model is done by MCMC sampling, using R package `CARBayes` by Lee (2013). Table 4 show Monte Carlo estimates of squared bias, variance, the coverage probability of 95% confidence intervals and computation time for $n, T \in \{10, 25\}$ and $n_{\mathscr{C}} = 1, 2, 3$. Two of the settings are the same as those shown for Model 1 in Table 1 in Section 3: $n = 25, n_{\mathscr{C}} = 3, T = 10$ and $n = 25, n_{\mathscr{C}} = 3, T = 25$. In estimation of MCAR, we also show results of two MCMC settings: 1. MCAR1: 1000 MCMC samples generated and 200 discarded as the burn-in period; 2. MCAR2: 5000 samples with 100 discarded. Coverage probabilities of our model is computed as $\hat{\boldsymbol{\theta}} \pm z_{0.975}\widehat{sd}(\hat{\boldsymbol{\theta}})$, where z_q is the $q-$quantile for a standard normal distribution. The standard error, $\widehat{sd}(\hat{\boldsymbol{\theta}})$, is obtained by taking the squared root of diagonal elements of $\boldsymbol{V}_n(\hat{\boldsymbol{\theta}})$ described in Section 2.3.

In overall, our method performs better than MCAR at analysing the kind of data that we generate, especially when n and/or T is small, with much smaller bias and variance, as well as computation time. The performance of MCAR improves significantly as the model gets more complicated (i.e. larger $n_{\mathscr{C}}$), and when n and T increases. In the case where $n = 25, T = 25$ and $n_{\mathscr{C}} = 3$, it almost performs equally well as our model, however, it takes almost an hour to obtain the estimates, while our method requires less than a minute. Besides, for the coverage probabilities to reach the nominal level, it seems that MCAR requires larger MCMC sample size as the model gets more complicated, while those of our model has been stable and close to the nominal level in all cases.

n = 10		$T = 10$				$T = 25$			
		$\widehat{\text{Bias}}^2$	$\widehat{\text{Var}}$	95%	Time	$\widehat{\text{Bias}}^2$	$\widehat{\text{Var}}$	95%	Time
$n_{\mathscr{C}} = 1$	Model 1	2.89(3.40)	21.22(0.98)	94.5	1s	1.29(1.62)	10.18(0.46)	94.6	3s
	MCAR1	420.20(44.63)	33.98(1.60)	90.1	4s	395.58(29.60)	13.75(0.60)	93.4	10s
	MCAR2	381.20(44.99)	30.07(1.33)	96.7	17s	143.09(15.44)	13.54(0.58)	94.8	45s
$n_{\mathscr{C}} = 2$	Model 1	3.86(3.87)	43.81(1.95)	95.3	2s	3.51(1.96)	33.64(1.52)	94.6	3s
	MCAR1	348.09(34.23)	90.85(4.11)	89.7	8s	177.33(15.29)	31.63(1.41)	92.7	25s
	MCAR2	202.02(38.77)	85.59(3.97)	93.7	34s	26.82(9.87)	32.64(1.56)	93.1	105s
$n_{\mathscr{C}} = 3$	Model 1	4.83(5.10)	34.66(1.59)	94.9	3s	4.4(2.48)	14.21(0.65)	94.7	7s
	MCAR1	217.09(17.38)	44.89(2.08)	82.6	12s	72.72(6.68)	13.73(0.62)	88.2	46s
	MCAR2	82.64(14.25)	38.71(1.77)	93.0	52s	13.64(3.75)	13.27(0.61)	93.0	190s
n = 25		$T = 10$				$T = 25$			
		$\widehat{\text{Bias}}^2$	$\widehat{\text{Var}}$	95%	Time	$\widehat{\text{Bias}}^2$	$\widehat{\text{Var}}$	95%	Time
$n_{\mathscr{C}} = 1$	Model 1	0.51(0.56)	3.18(0.14)	94.9	10s	0.40(0.37)	1.73(0.08)	94.5	23s
	MCAR1	20.98(3.82)	3.99(0.17)	93.4	31s	41.21(1.63)	2.41(0.08)	92.0	70s
	MCAR2	4.35(1.57)	4.64(0.21)	95.6	145s	10.37(3.75)	1.98(0.11)	93.5	345s
$n_{\mathscr{C}} = 2$	Model 1	0.76(0.34)	13.22(0.56)	94.4	8s	0.59(0.67)	5.69(0.25)	94.4	19s
	MCAR1	26.17(5.61)	14.33(0.62)	91.9	54s	16.14(1.65)	5.54(0.24)	92.6	157s
	MCAR2	10.84(4.67)	13.87(0.60)	93.7	260s	3.15(0.83)	5.44(0.24)	92.9	2290s
$n_{\mathscr{C}} = 3$	Model	0.67(0.66)	5.91(0.27)	94.6	24s	0.31(0.44)	2.35(0.14)	94.9	55s
	MCAR1	15.42(2.17)	12.66(0.59)	60.6	82s	14.10(1.48)	4.43(0.28)	30.5	300s
	MCAR2	2.13(2.14)	6.53(0.30)	92.3	390s	0.64(0.73)	2.16(0.13)	92.7	3387s

Table 2.4: Monte Carlo estimates for squared bias ($\times 10^{-6}$), variance ($\times 10^{-4}$), the coverage probability of 95% confidence intervals as well as computation time for $n, T \in \{10, 25\}$ and $n_{\mathscr{C}} = 1, 2, 3$ of MLE of our model, and MCAR, where in MCAR1, 1000 MCMC samples generated and 200 discarded as the burn-in period; and in MCAR2, 5000 samples with 100 discarded. True values of regression parameters are shown as $\mathscr{B}_1$ in Section 3. Estimates are obtained from 1000 Monte Carlo runs.

2.4 Analysis of the cancer cell growth data

Cancer cell behaviour is believed to be determined by several factors including genetic profile and differentiation state. However, the presence of other cancer cells and non-cancer cells has also been shown to have a great impact on overall tumor behaviour (Tabassum and Polyak, 2015; Kalluri and Zeisberg, 2006). It is therefore important to be able to dissect and quantify these interactions in complex culture systems. The data sets in this section represent two scenarios: cancer cell-fibroblast co-culture and cloned cancer cell co-culture experiments. The data sets analyzed consist of counts of cell types (different cancer cell populations expressing different fluorescent proteins, and non-fluorescent fibroblasts) from 9 subsequent images taken at an 8-hour frequency over a period of 3 days using the Operetta high-content imager (Perkin Elmer). Information regarding cell type (fluorescent profile) and spatial coordinates for each individual cell were extracted using the associated software (Harmony, Perkin Elmer).

Each image was subsequently tiled using a 25×25 regular grid. We choose the num-

ber of tiles for a balance between the fit of the model and capturing the local impact between cell populations. More specifically, decreasing tile sizes enables one to detect local impacts between cell populations, which is one of the objectives of our analysis. However, if the tiles are too small, we will end up with mostly no cells in most tiles. In this situation the conditional Poisson model would not fit well the data. On the other hand, when the tiles are too large the model would fit the data well (the conditional Poisson would be approximately a conditional normal model), but we lose information on local impacts. We recommend 0 to 20 average cells per tile, since for such choice our diagnostic and goodness-of-fit analyses suggest that the conditional Poisson model fits well the data whilst enabling us to measure local correlation effects between populations.

2.4.1 Cancer cell-fibroblast co-culture experiment

In this experiment, cancer cells are co-cultured with fibroblasts, a predominant cell type in the tumor microenvironment, believed to affect tumor progression, partly due to interactions with and activation by cancer cells (Kalluri and Zeisberg, 2006). In this experiment, fibroblasts (F) are non-fluorescent whereas cancer cells fluoresce either in the red (R) or green (G) channels due to the experimental expression of mCherry or GFP proteins, respectively. Cells were initially seeded at a ratio of 1:1:2 (R:G:F).

Model selection and inference. We applied our methodology to quantify the magnitude and direction of the impacts have on growth for the considered cell types. To select the relevant terms in the intensity expression (2.1), we carry out model selection using the BIC model selection criterion. In Table 5, we show estimated parameters for the full and the BIC models, with bootstrap 95% confidence intervals in parenthesis. Figure 2.2 illustrates estimated spatio-temporal impacts between cell types using a directed graph. The solid and dashed arrows represent respectively significant and not significant impacts between cell types at the 95% confidence level. Significant impacts coincide with parameters selected by BIC.

The interactions within each cell type ($\hat{\beta}^{(c|c)}, c = R, G, F$) are significant, which is consistent with healthy growing cells. As anticipated, the effects $\hat{\beta}^{(c|c)}$ for the cancer cells are larger than those for the slower growing fibroblasts. The validity of the estimated parameters is also supported by the similar sizes of the parameters for the green and red cancer cells. This is expected, since the red and green cancer cells are biologically identical except for the fluorescent protein they express. Interestingly, the size of the estimated effects within both types of cancer cells ($\hat{\beta}^{(c|c)}, c = R, G$) are larger than the impact they have on one another ($\hat{\beta}^{(G|R)}$ and $\hat{\beta}^{(R|G)}$). This is not surprising, since $\hat{\beta}^{(c|c)} (c = R, G)$ reflects not only impacts between cells from the same cell population, but also cell proliferation. The fact that we are able to detect the impacts between the red and green cancer cells

confirms that our methodology is sensitive enough to detect biologically relevant impacts even though no interactions were found between the cancer cells and the fibroblasts. This might be due to the fact that we used normal fibroblasts that had not previously been in contact with cancer cells and thus had not been activated to support tumor progression as is the case with cancer-activated fibroblasts.

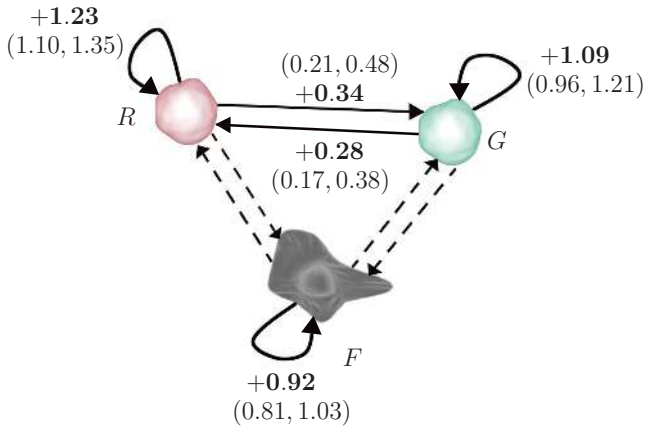


Figure 2.2: Directed graph showing fitted spatio-temporal interactions between GFP cancer cells (G), mCherry cancer cells (R) and fibroblasts (F). The solid and dashed arrows represent respectively the significant and not significant interactions between cell types at the 95% confidence level.

Goodness-of-fit and one-step ahead prediction To illustrate the goodness-of-fit of the estimated model, we generate cell counts for each type in each tile, $\hat{y}_{i,t}^{(c)}$, from the Pois($\hat{\lambda}_{i,t}^{(c)}$) distribution for $t \geq 1$, where $\hat{\lambda}_{i,t}^{(c)}$ is computed using observations at time $t-1$, with parameters estimated from the entire dataset. In Figure 4, we compare the actually observed and generated cell counts for GFP cancer cells (G) and mCherry cancer cells (R) and fibroblasts (F) across the entire image. The solid and dashed curves for all cell types are close, suggesting that the model fits the data reasonably well. As anticipated, the overall growth rate for the red and green cancer cells are similar, and sensibly larger than the growth rate for fibroblasts.

To assess the prediction performance of our method, we consider one-step-ahead forecasting using parameters estimated from a moving window of five time points. In Figure 2.3, we show quantiles of observed cell counts against predicted counts for each tile. The upper and lower 95% confidence bounds are computed non-parametrically by taking $\hat{F}_1^{-1}\big(\hat{F}_0(y_t^{(c)}) - 0.95\big)$ and $\hat{F}_1^{-1}\big(\hat{F}_0(y_t^{(c)}) + 0.95\big)$, where $\hat{F}_0$ and $\hat{F}_1$ are the empirical distributions of the observations and predictions at time t respectively Koenker (2004). The identity line falls within the confidence bands in each plot, indicating a satisfactory prediction performance.

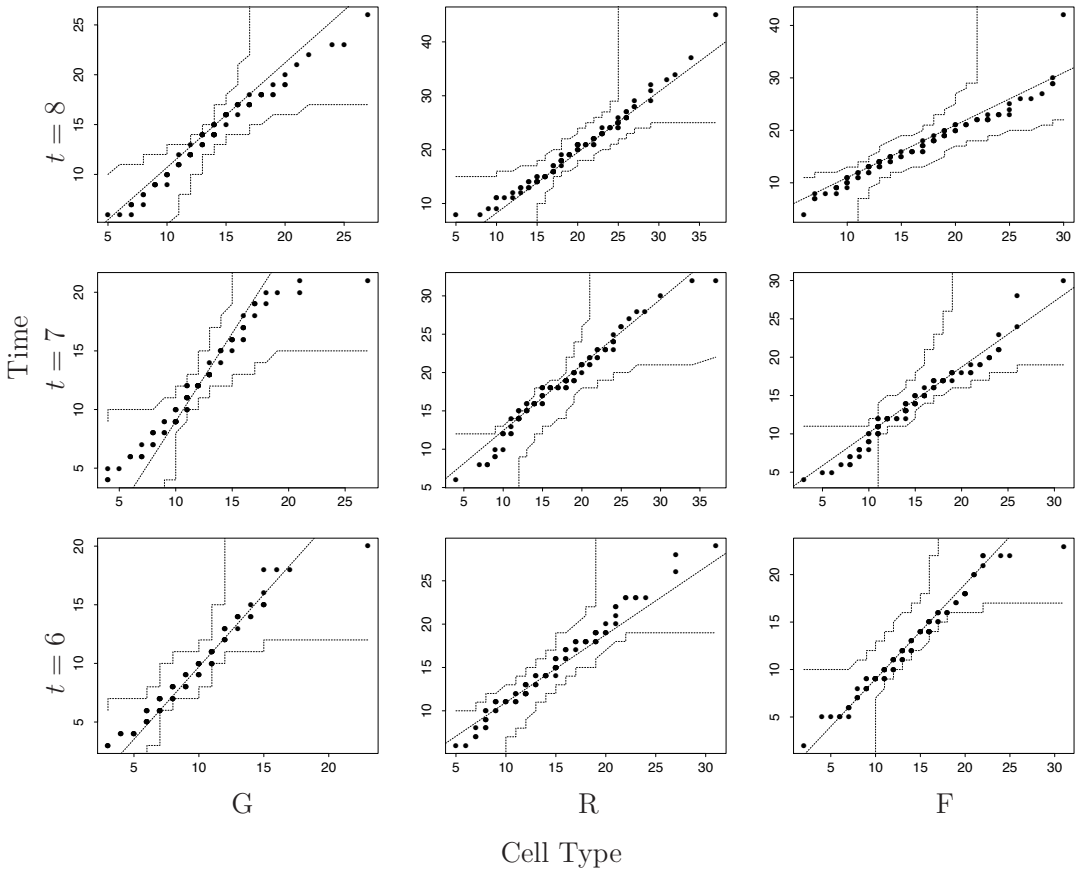


Figure 2.3: QQ-plots for cell growth, comparing observed (horizontal axis) and one-time ahead predicted (vertical axis) cell counts per tile on the entire image at times $t = 6, 7, 8$ for GFP cancer cells (G), mCherry cancer cells (R) and fibroblasts (F). One-time ahead predictions are based on the model fitted using a moving window of five time points.

Comparison with MCAR model Next, we compare the estimates as well as the goodness-of-fit on the real data with the MCAR model. Parameter estimates are shown in Table 5, with 95% confidence intervals given in parenthesis. Results from both models are mostly consistent with each other, specifically, both models show that impacts within each cell type ($\hat{\beta}^{(c|c)}, c = R, G, F$) are significant, the effects $\hat{\beta}^{(c|c)}$ for cancer cells are larger than those for the slower growing fibroblasts, the green and red cancer cells have positive impact on each other, and cancer cells have no impact on fibroblasts. The only difference is, the MCAR model shows a negative impact of fibroblasts on the green cancer cells only, while our model detect no significant impact on either cancer cells. Since the red and green cancer cells are biologically identical except for the fluorescent protein they express, we expect a symmetrical result with both cancer cells.

In Figure 2.4, apart from the observed (solid curve) and generated (dashed curve) cell counts from our model, we also show the generated cell counts from the MCAR model (dotted curve) for the green cancer cells (G), red cancer cells (R) and fibroblasts (F) across the entire image. Compared to the dotted curves, the dashed curves are slightly closer to

	Full model		
$c =$	G	R	F
$\widehat{\alpha}^{(c)}$	-0.99 (-1.19, -0.79)	-0.50 (-0.70, -0.30)	-0.26 (-0.45, -0.06)
$\widehat{\beta}^{(G\|c)}$	1.23 (1.10, 1.35)	0.34 (0.21, 0.48)	0.12 (-0.03, 0.27)
$\widehat{\beta}^{(R\|c)}$	0.28 (0.17, 0.38)	1.09 (0.96, 1.21)	0.02 (-0.09, 0.13)
$\widehat{\beta}^{(F\|c)}$	0.10 (-0.01, 0.21)	0.02 (-0.07, 0.12)	0.92 (0.81, 1.03)
	BIC model		
$c =$	G	R	F
$\widehat{\alpha}^{(c)}$	-0.88 (-1.04, -0.72)	-0.49 (-0.66, -0.31)	-0.19 (-0.36, -0.02)
$\widehat{\beta}^{(G\|c)}$	1.24 (1.11, 1.37)	0.35 (0.21, 0.48)	/
$\widehat{\beta}^{(R\|c)}$	0.28 (0.17, 0.39)	1.09 (0.96, 1.21)	/
$\widehat{\beta}^{(F\|c)}$	/	/	0.93 (0.82, 1.04)
	MCAR		
$c =$	G	R	F
$\widehat{\alpha}^{(c)}$	-0.45 (-0.54, -0.38)	-0.45 (-0.54, -0.38)	-0.45 (-0.54, -0.38)
$\widehat{\beta}^{(G\|c)}$	1.06 (0.93, 1.16)	0.16 (0.09, 1.16)	-0.15 (-0.22, -0.09)
$\widehat{\beta}^{(R\|c)}$	0.25 (0.15,0.31)	1.01 (0.92,1.08)	0.05 (-0.02,0.10)
$\widehat{\beta}^{(F\|c)}$	0.03 (-0.06,0.20)	0.03 (-0.07,0.19)	0.96 (0.83,1.08)

Table 2.5: Estimated parameters for the full, the BIC models and the MCAR model based on the cancer cell growth data described in Section 2.4. Bootstrap 95% confidence intervals based on 50 bootstrap samples are given in parenthesis.

the solid ones, which means our model seems more appropriate for analysing this type of data than the MCAR model.

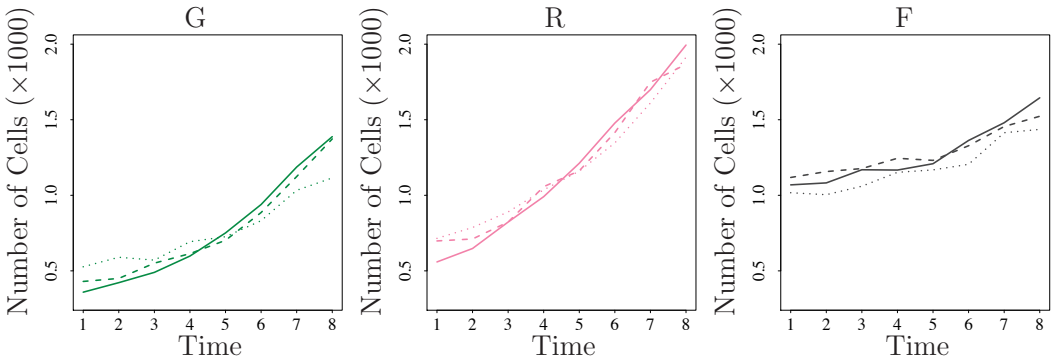


Figure 2.4: Goodness-of-fit of the estimated models. Observed (solid) and predicted (dashed for our model and dotted for the MCAR model) number of GFP cancer cells (G), mCherry cancer cells (R) cancer cells and fibroblasts (F) for the entire image. Predicted cell counts for each cell type in each tile $\hat{y}_{i,t}^{(c)}$ is generated from the conditional Poisson model with intensity $\hat{\lambda}_{i,t}^{(c)}$ defined in Equation (2.1) and (2.2), where the coefficients $\hat{\beta}^{(c|c')}$ are estimated from the entire dataset.

2.4.2 Cloned cancer cell co-culture experiment

In the second example, cloned cancer cells showing different behaviors are cultured together in different combinations. Three cloned cancer cell populations (populations were generated from one single cell), called F7, F8, and G10, were co-cultured in pairs (seeded at 1:1 ratio) or all together (at a 1:1:1 ratio). A total number of 3,000 cells was initially seeded for all tested co-cultures. The three cloned cancer cell populations can be readily distinguished based on image data, due to their experimentally-induced differential expression of Red, Green and Blue fluorescent proteins. Unlike for the cancer cell-fibroblast

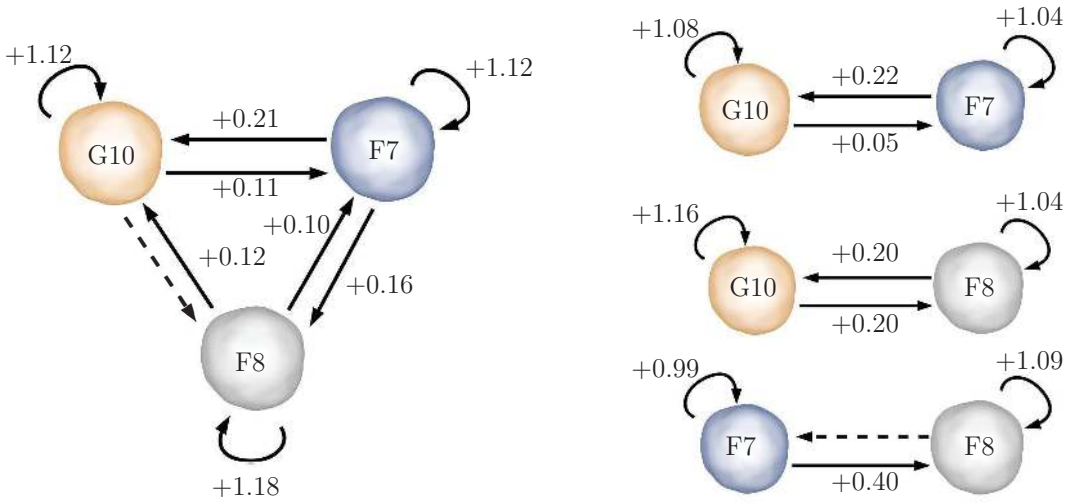


Figure 2.5: Directed graph showing fitted spatio-temporal interactions between three cloned cancer cell populations: G10, F7 and F8. The solid and dashed arrows represent respectively the significant and not significant interactions between cell types at the 99% confidence level.

co-culture experiment, cancer cell populations in this second example display different growth behaviours. As shown in Figure 2.5 these different behaviours translate into different interaction patterns in terms of size and symmetry of interactions. Interactions between two individual clones is frequently modified upon addition of a third different clone, which can affect the amplitude of these interactions (F7 on F8, F8 on G10, G10 on F7), trigger an otherwise undetectable interaction (F8 on F7), or repress an interaction detected in the pairwise setting (G10 on F8). In contrast, other interactions remain similar in pairwise and triple co-cultures. These comparisons of the pairwise and triple co-cultures confirms the importance of studying cellular behaviour in a relevant context as the majority of interactions between the different cloned cell populations are changed when another clone is added. This is consistent with a recently published study showing that the growth properties of cloned cell populations varies depending on whether they are cultured alone or together with other cloned populations (Mohme et al., 2017).

2.5 Conclusion and final remarks

In this chapter, we introduced a conditional autoregressive model and accompanying inference tools for spatio-temporal cell count data. The proposed methodology enables one to measure the overall cell growth rate in longitudinal experiments and spatio-temporal interactions with either homogeneous or heterogeneous cell populations. The proposed inference approach is computationally tractable and strikes a good balance between computational feasibility and statistical accuracy. Numerical findings from simulated and real data in Sections 3 and 4 confirm the validity of the proposed approach in terms of prediction, goodness-of-fit and estimation accuracy.

The data sets described in this chapter serve as a proof-of-concept that the proposed methodology works. However, the potential applications and the relevant questions that the methodology can help to answer in cancer cell biology are plentiful. To build on from the examples given in this chapter, the methodology can be used to study interactions between cancer cells and a wide range of cancer-relevant cell types such as cancer-activated fibroblasts, macrophages, and other immune cells when co-cultured. Since a substantial proportion of cancer cells in tumors are in close proximity to other cell types that have been shown to affect tumor progression, using these co-cultures is more representative of the situation in a patient compared to studying cancer cells on their own. In addition to just giving the final cell number, the presented approach can dissect which cell types affect the growth of others and to what extent in complex heterogeneous populations. This could be relevant in a drug discovery setting to determine if a drug affects cancer cell growth due to internal effects (on other cancer cells) or by interfering with the interaction between the cancer cells and other cell types. Finding drugs with different targets and mechanisms of action are particularly sought after as they provide a wider target profile, increasing the chance of patients responding as well as reducing the risk of tumors becoming resistant. The impact of different genes and associated pathways in different cell types in relation to inter-cellular interactions can also be studied by genetically modifying the cell type(s) in question before mixing the cells together. This could be beneficial to identify new potential drug targets. Our approach is also applicable in other kinds of studies where local spatial cell-cell interactions are believed to affect cell growth such as studies of neurodegenerative diseases (Garden and La Spada, 2012) and wound healing/tissue re-generation (Leoni et al., 2015). In addition to evaluating cell growth, our approach can also be used to study transitions between cellular phenotypes upon interaction with other cell types, provided that the different phenotypes studied can be distinguished from one another based on the image data. Finally, it is worth noting that issues may arise when cells become too confluent/dense, this may lead to segmentation problems of the imaging system. If they become completely confluent, they are likely to progressively stop growing. If one wants to measure for longer period of time, experiments can be performed in larger wells/plates

or with smaller starting cell numbers.

Our methods offer several practical advantages to researchers interested in analysing count data on heterogeneous cell populations. First, the conditional Poisson model does not require tracking individual cells across time, a process that is often difficult to automate due to cell movement, morphology changes at subsequent time points, and additional complications related to storage of large data files. Second, we are able to quantify local spatio-temporal interactions between different cell populations from a very simple experimental set-up where the different cell populations are grown together in a single experimental condition (co-culture). An alternative, solely experimentally-based strategy would require monitoring the different cell types alone and together at different cell densities (number of cells per condition) in order to make inferences in terms of potential interactions. However, such an approach would give no possibility of evaluating the spatial relations in the co-culture conditions and would still restrict the number of simultaneously tested cell types to two.

In the future, we foresee several useful extensions of the current methodology, possibly enabling the treatment of more complex experimental settings. First, complex experiments involving a large number of cell populations, $n_{\mathscr{C}}$, would imply an over-parametrized model. Clearly, this large number of parameters would be detrimental to both statistical accuracy and reliable optimization of the likelihood objective function $\ell_n(\theta)$ (3.4). To address these issues, one possible direction is a penalized likelihood of form $\ell_n(\theta) - \text{pen}_\lambda(\theta)$, where $\text{pen}(\theta)$ is a nonnegative sparsity-inducing penalty function. For example, in a different likelihood setting, Bradic et al. (2011) consider the L_1-type penalty $\text{pen}(\theta) = \lambda \sum |\theta|$, $\lambda > 0$.

Second, for certain experiments, it would be desirable to modify the statistics in (2.2) to include additional information on cell growth such as the distance between heterogeneous cells, and covariates describing cell morphology. Beside, it would be useful to develop a more principled way to select the tile sizes/number, and consider tiling the microscope image into a hexagonal lattice, which is a more natural choice in real application, since the distance between neighbouring tiles would be more even than that of a regular lattice.

Thirdly, although numerical results (results not reported here) show that our method are quite robust in the presence of mild outliers (with around 5% of contaminated data), for more severe situations, we expect that severe or numerous outliers will have some influence on the estimates since the Poisson score function is unbounded. To address this problem, the log-likelihood scores in Equation (5) should be replaced by some other robust alternative. Following Ferrari and Vecchia (2011) and La Vecchia et al. (2015), robustness can be obtained by the so-called q-entropy estimation method simply obtained by replacing the usual logarithm in the log-likelihood estimating equation by the q-logarithm logarithm function $\log_q(x) = (x^{1-q} - 1)/(1-q)$if $q \neq 1$, and $\log_q(u) = \log(x)$ if $q = 1$,

for all $x > 0$. This ensures a bounded influence function for the implied estimator and therefore guarantees control of the bias under contamination.

Last but not the least, it would be desirable to take into consideration the correlations between response variables, for example, the correlation between $Y_{i,t}^{(}c_1)$ and $Y_{j,t}^{(}c_2)$ for $i \neq j, c_1 \neq c_2$. The correlation structure can be incorporated via a Gaussian copula or a multivariate latent process.

2.6 Appendix

In the first part of this section, we provide technical lemmas required to prove asymptotic properties of the estimator $\hat{\boldsymbol{\theta}}_n$.

Denote $E_t[\cdot]$ as the expectation with respect to $\boldsymbol{Y}_t = \{\boldsymbol{Y}_{i,t}, i \in \mathscr{L}_n\}$, and $E[\cdot]$ as the expectation of $\boldsymbol{Y} = \{\boldsymbol{Y}_t, t = 1, \ldots, T\}$. Let $N_{i,r}$ be the set of tiles in the neighbourhood of tile i, with radius r. Specifically, for two locations i and j, we say $j \in N_{i,r}$ if $\|i - j\| \leq r$. Thus, the neighbourhood defined in Section 2 is of radius 1, i.e. $\{j : j \sim i\} = \{j : j \in N_{i,1}\}$. Denote $n_r = \max_{i \in \mathscr{L}_n} |N_{i,r}| = r^2 + r + 1$. Actually, for any tile i that is not on the boundary of the image, $|N_{i,r}| = n_r$.

In the remainder of this paper we use the following assumptions:

- A.1: The parameter space $\boldsymbol{\Theta}$ is a compact subset of $\mathbb{R}^p$, and that $\boldsymbol{\theta}_0$ is the unique maximiser of $\ell(\boldsymbol{\theta}) = \lim_{n_{\mathscr{L}} \to \infty} \ell_n(\boldsymbol{\theta})$.

- A.2: The $(n_{\mathscr{C}} + 1) \times n_{\mathscr{L}} T$ matrix $(\nabla \boldsymbol{v}_{1,1}, \nabla \boldsymbol{v}_{1,2}, \ldots, \nabla \boldsymbol{v}_{1,T}, \nabla \boldsymbol{v}_{2,1}, \ldots, \nabla \boldsymbol{v}_{n,T})$ is full rank.

Lemma 1. *Let $Y_1, \ldots, Y_n$ be independent Poisson random variables with mean $\lambda_1, \ldots, \lambda_n$ respectively, where N is a finite positive integer. Then for any positive integer h,*

$$E\left[\max_{i=1,\ldots,n} Y_i^h\right] \leq n^h \max_{i=1,\ldots,n} E\left[Y_i^h\right].$$

Proof.

$$\begin{aligned} E\left[\max_{i=1,\ldots,n} Y_i^h\right] &\leq E\left[\left(\sum_{i=1}^n Y_i\right)^h\right] \\ &\leq n^{h-1} E\left[\sum_{i=1}^n Y_i^h\right] \quad \text{(convexity)} \\ &\leq n^h \max_{i=1,\ldots,n} E\left[Y_i^h\right]. \end{aligned}$$

□

Lemma 2. *Denote* $\tilde{Y}_{N_{i,r},t} = \max_{j\in N_{i,r}, c\in\mathscr{C}} Y_{j,t}^{(c)}$*, with corresponding observation* $\tilde{y}_{N_{i,r},t}$ *and conditional mean* $\tilde{\lambda}_{N_{i,r},t}$*, then*

$$E\left[\left(\tilde{Y}_{N_{i,r},t}+1\right)^B\right] \leq w_{r,t}\sum_{k=0}^{B^t} f_t(k)e^{k\tilde{\alpha}}\left(1+\tilde{y}_{N_{i,r+t},0}\right)^{Bk}, \quad t=1,2,\ldots,T \tag{2.8}$$

where

$$f_t(k) = \sum_{h=\lceil k/B\rceil}^{B^{t-1}} e^{\tilde{\alpha}h} g(k,Bh) f_{t-1}(h), \quad g(a,b) = \sum_{k=a}^{b}\binom{b}{h}\left\{{h \atop a}\right\},$$
$$f_1(k) = g(k,B) = \sum_{h=k}^{B}\binom{B}{h}\left\{{h \atop k}\right\}, \quad w_{r,t} = \prod_{k=0}^{t-1} n_{r+k}2^{n_{r+k}},$$

the $\{\cdot\}$ *denotes Stirling number of the second kind,* $\tilde{\alpha} = \max_{c\in\mathscr{C}}\alpha^{(c)}$, $B = \max_c\left(\sum_{c'\in\mathscr{C}}\beta^{(c|c')}\right)n_1$.

Proof.

$$\lambda_{i,t}^{(c)} = \exp\left[\alpha^{(c)} + \sum_{c'\in\mathscr{C}}\beta^{(c|c')}\sum_{j\in N_{i,1}}\log\left(y_{j,t-1}^{(c')}+1\right)\right] \leq e^{\tilde{\alpha}}\left(\tilde{y}_{N_{i,1},t-1}+1\right)^B. \tag{2.9}$$

Similarly, for any $c\in\mathscr{C}$, we have $\lambda_{N_{i,r},t}^{(c)} \leq e^{\tilde{\alpha}}\left(\tilde{y}_{N_{i,r+1},t-1}+1\right)^B$, since $\{j'\in N_{j,1}; j\in N_{i,r}, i\in\mathscr{L}_n, r>0\} = \{j\in N_{i,r+1}; i\in\mathscr{L}_n, r>0\}$.

Next, we proceed by induction. For $T=1$, by the conditional independence assumption and Lemma 1, we have

$$E_{T-1}\left[E_T\left(\left(\tilde{Y}_{N_{i,r},T}+1\right)^B|\boldsymbol{Y}_{T-1}\right)\right] = E_{T-1}\left[\sum_{h=0}^{B}\binom{B}{h}E_T\left(\max_{j\in N_{i,r},c\in\mathscr{C}} Y_{i,T}^{(c)h}|\boldsymbol{Y}_{T-1}\right)\right]$$
$$< n_r 2^{n_r} E_{T-1}\left[\sum_{k=0}^{B}\sum_{h=k}^{B}\binom{B}{h}\left\{{h \atop k}\right\}\tilde{\lambda}_{N_{i,r},T}^k\right] \leq w_{r,1}\sum_{k=0}^{B} f_1(k)e^{k\tilde{\alpha}}E_{T-1}\left[\left(1+\tilde{Y}_{N_{i,r+1},T-1}\right)^{Bk}\right].$$

Since $T-1=0$ and Y_t has constant entries at time point 0, $E_{T-1}\left[\left(1+\tilde{Y}_{N_{i,r+1},T-1}\right)^{Bk}\right] = \left(1+\tilde{y}_{N_{i,r+1},0}\right)^{Bk}$.

Suppose (2.8) is true for $T=t$, then for $T=t+1$, we have

$$
\begin{aligned}
&E_{T-t-1}E_{T-t}E_{T-t+1}\dots E_T\left[\left(\tilde{Y}_{N_{i,r},T}+1\right)^B\middle|\boldsymbol{Y}_{T-1},\dots,\boldsymbol{Y}_{T-t-1}\right]\\
&\le E_{T-t-1}\left\{w_{r,t}\sum_{k=0}^{B^t}f_t(k)e^{k\tilde{\alpha}}E_{T-t}\left[\left(1+\tilde{Y}_{N_{i,r+t},T-t}\right)^{Bk}\middle|\boldsymbol{Y}_{T-t-1}\right]\right\}\\
&=w_{r,t}\sum_{k=0}^{B^t}f_t(k)e^{k\tilde{\alpha}}\left\{\sum_{k'=0}^{Bk}\binom{Bk}{k'}E_{T-t-1}\left[E_{T-t}\left(\tilde{Y}^{k'}_{N_{i,r+t},T-t}\middle|\boldsymbol{Y}_{T-t-1}\right)\right]\right\}\\
&\le w_{r,t}\sum_{k=0}^{B^t}f_t(k)e^{k\tilde{\alpha}}\left\{\sum_{k'=0}^{Bk}\binom{Bk}{k'}E_{T-t-1}\left[n_{r+t}2^{n_{r+t}}\max_{j\in N_{i,r+t},c\in\mathscr{C}}E_{T-t}\left(Y^{(c)}_{j,T-t}{}^{k'}\middle|\boldsymbol{Y}_{T-t-1}\right)\right]\right\}\\
&=w_{r,t+1}\sum_{k=0}^{B^t}f_t(k)e^{k\tilde{\alpha}}\left[\sum_{k''=0}^{Bk}\sum_{k'=k''}^{Bk}\binom{Bk}{k'}\left\{{k'\atop k''}\right\}E_{T-t-1}\left(\tilde{\lambda}^{k''}_{N_{i,r+t},T-t}\right)\right]\\
&\le w_{r,t+1}\sum_{k''=0}^{B^{t+1}}\sum_{k=\lceil k''/B\rceil}^{B^t}f_t(k)e^{k\tilde{\alpha}}g(k'',Bk)e^{k''\tilde{\alpha}}E_{T-t-1}\left[\left(1+\tilde{Y}_{N_{i,r+t+1},T-t-1}\right)^{Bk''}\right]\\
&=w_{r,t+1}\sum_{k''=0}^{B^{t+1}}f_{t+1}(k'')e^{k''\bar{\alpha}}\left(1+\tilde{y}_{N_{i,r+t+1},0}\right)^{Bk''}.
\end{aligned}
$$

□

Lemma 3. *Given Assumption A.1, for any finite constant $a,b\ge 0$ and $\theta\in\Theta$, $E\left(\lambda^{(c)a}_{i,t}S^{(c')}_{i,t-1}{}^{b}\right)<\infty,\quad \forall c,c'\in\mathscr{C},i\in\mathscr{L}_n,t=1,\dots,T.$*

Proof. By the definition of $f_t(k)$ given in Lemma 2, we know that $f_t(k)$ is bounded for all bounded t under assumption A.1. Thus, Lemma 2 implies

$$
\begin{aligned}
E\left(\lambda^{(c)a}_{i,t}S^{(c')}_{i,t-1}{}^{b}\right)&=E\left[\left(\sum_{j\in N_{i,1}}\log(1+Y^{(c')}_{j,t-1})\right)^b\lambda^{(c)a}_{i,t}\right]\\
&\le E\left[\left(1+\tilde{Y}_{N_{i,1},t-1}\right)^{bB}\lambda^{(c)a}_{i,t}\right]\le E\left[e^{a\bar{\alpha}}\left(1+\tilde{Y}_{N_{i,1},t-1}\right)^{(a+b)B}\right]\\
&\le e^{a\bar{\alpha}}w_{1,t}\sum_{k=0}^{B^t}f_t(k)e^{k\tilde{\alpha}}\left(1+\tilde{y}_{N_{i,1+t},0}\right)^{Bk}<\infty.
\end{aligned}
$$

□

For simplicity, define the distance between tile i and j as $d(i,j)=r$ if $r-1<\|i-j\|\le r$.

Lemma 4. *For any $i\in\mathscr{L}_n,t_1=1,\dots,T,$*

$$Cov(Y_{i,t_1},Y_{j,t_2})=0,\quad \text{for } \forall j\in\mathscr{L}_n,t_2=1,\dots,T,\ \text{if } d(i,j)>t_1+t_2.$$

and

$$|(j,t_2) : Cov(Y_{j,t_2}, Y_{i,t_1}) \neq 0; j \in \mathscr{L}_n, t_2 = 1,\ldots,T,| \leq T(8T^2+4T+1)$$

Proof. Let $N^*_{i,t} = \{j : \mathrm{Cov}(Y_{j,0}, Y_{i,t}) \neq 0; j \in \mathscr{L}\}$ be the collection of counts in tiles at time 0 that are correlated with the count in tile i at time t ($Y_{i,t}$). Due to the neighbourhood structure in the autoregressive term described in Section 2, one can easily tell that $N^*_{i,t}$ is a neighbourhood around tile i, with the radius equal to t. Due to the condition that Y_t has constant entries at time 0, we have $\mathrm{Cov}(Y_{i,t_1}, Y_{j,t_2}) = 0$ if $N^*_{i,t_1} \cap N^*_{j,t_2} = \emptyset$, which is true when $d(i,j) > t_1 + t_2$.

For any $(i,t_1) \in D_n$, $\{(j,t_2) : N^*_{i,t_1} \cap N^*_{j,t_2} \neq \emptyset\}$ is a neighborhood around tile i, with a radius $t_1 + t_2$. Since $n_r = 2r^2 + 2r + 1$, we have

$$|(j,t_2) : N^*_{i,t_1} \cap N^*_{j,t_2} \neq \emptyset| \leq T\,|j : N^*_{i,T} \cap N^*_{j,T} \neq \emptyset| = TN_{2T} = T(8T^2+4T+1).$$

□

In the second part of this section, we study the asymptotic properties of the estimator $\hat{\theta}_n$.

Proposition 1 (Existence and uniqueness)**.** *If assumption A.3 holds, then there exist unique maximizer of $\ell_n(\boldsymbol{\theta})$, denoted by $\hat{\boldsymbol{\theta}}_n$.*

Proof. First, since $\boldsymbol{\Theta}$ is compact and $\ell_n(\boldsymbol{\theta})$ is continuous, at least one maximiser of $\ell_n(\boldsymbol{\theta})$ exist. Next, we wish to prove that the maximiser is unique. The $p \times p$ Hessian matrix of $-\ell_n(\boldsymbol{\theta})$ can be written as a block matrix

$$\boldsymbol{H}_n(\boldsymbol{\theta}) = -\nabla^2 \ell_n(\boldsymbol{\theta}) = \begin{pmatrix} \boldsymbol{H}_n^{(1)}(\boldsymbol{\theta}) & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \boldsymbol{H}_n^{(2)}(\boldsymbol{\theta}) & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \boldsymbol{H}_n^{(n_{\mathscr{C}})}(\boldsymbol{\theta}) \end{pmatrix},$$

where $\boldsymbol{H}_n^{(c)}(\boldsymbol{\theta}) = \sum_{i \in \mathscr{L}_n} \sum_{t=1}^{T} \exp\left[\boldsymbol{v}_{i,t}^{(c)}(\boldsymbol{\theta})\right] [\nabla \boldsymbol{v}_{i,t}] [\nabla \boldsymbol{v}_{i,t}]^\top$ is a $(n_{\mathscr{C}}+1) \times (n_{\mathscr{C}}+1)$ matrix. Matrix $[\nabla \boldsymbol{v}_{i,t}] [\nabla \boldsymbol{v}_{i,t}]^\top$ is positive semidefinite with rank 1. By Assumption A.2, $\sum_{i \in \mathscr{L}_n} \sum_{t=1}^{T} [\nabla \boldsymbol{v}_{i,t}] [\nabla \boldsymbol{v}_{i,t}]^\top$ is full rank, which means $\boldsymbol{H}_n^{(c)}(\boldsymbol{\theta})$ is positive definite for all $c \in \mathscr{C}$ and $\boldsymbol{\theta} \in \boldsymbol{\Theta}$, since $\exp\left[\boldsymbol{v}_{i,t}^{(c)}(\boldsymbol{\theta})\right] > 0$. This shows that $-\ell_n(\boldsymbol{\theta})$ is strictly convex, which implies $\hat{\boldsymbol{\theta}}_n$ is unique. □

Proposition 2 (Consistency)**.** *If the regularity assumption A.1 holds, then $\hat{\boldsymbol{\theta}}_n \xrightarrow{p} \boldsymbol{\theta}_0$ with probability tending 1, as $n_{\mathscr{L}} \to \infty$.*

Proof. We proceed by verifying the conditions of Theorem 2 in Jenish and Prucha (2009).

First we show that the score functions are $\mathscr{L}_p$-Uniform Integrable for $p < 3$, i.e.

$$\lim_{n\to\infty} \sup_{\substack{i\in\mathscr{L}_n \\ t=1,\dots,T}} \sup_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} E\left[\boldsymbol{u}_{i,t}^p(\boldsymbol{\theta}) I\big(\boldsymbol{u}_{i,t}(\boldsymbol{\theta}) > k\big)\right] \to \mathbf{0}, \quad \text{as } k \to \infty. \tag{2.10}$$

The general form of each entry of $\boldsymbol{u}_{i,t}(\boldsymbol{\theta})$ is $(\lambda_{i,t}^{(c)} - y_{i,t}^{(c)}) S_{i,t-1}^{c'}$, take $p = 3$, we have

$$\begin{aligned}
&E\left[\left((\lambda_{i,t}^{(c)} - y_{i,t}^{(c)}) S_{i,t-1}^{c'}\right)^3\right] \\
&= E_1 \dots E_{t-2} E_{t-1} \left[E_t \left[\left((\lambda_{i,t}^{(c)} - Y_{i,t}^{(c)}) S_{i,t-1}^{c'}\right)^3 | \boldsymbol{Y}_{t-1}\right] | \boldsymbol{Y}_{t-2}\right] \dots \\
&= E_1 \dots E_{t-2} E_{t-1} \left[{S_{i,t-1}^{c'}}^3 \left[{\lambda_{i,t}^{(c)}}^3 - 3{\lambda_{i,t}^{(c)}}^2 E_t\left[Y_{i,t}^{(c)} | \boldsymbol{Y}_{t-1}\right] + 3\lambda_{i,t}^{(c)} E_t\left[{Y_{i,t}^{(c)}}^2 | \boldsymbol{Y}_{t-1}\right] + E_t\left[{Y_{i,t}^{(c)}}^3 | \boldsymbol{Y}_{t-1}\right]\right] | \boldsymbol{Y}_{t-2}\right] \dots \\
&= E_1 \dots E_{t-2} E_{t-1} \left[{S_{i,t-1}^{c'}}^3 \left(2{\lambda_{i,t}^{(c)}}^3 + 6{\lambda_{i,t}^{(c)}}^2 + \lambda_{i,t}^{(c)}\right) | \boldsymbol{Y}_{t-2}\right] \dots,
\end{aligned}$$

which is finite by lemma 3. This gives us the $\mathscr{L}_3-$boundedness of $\boldsymbol{u}_{i,t}(\boldsymbol{\theta})$, i.e.

$$\lim_{n\to\infty} \sup_{\substack{i\in\mathscr{L}_n \\ t=1,\dots,T}} \sup_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} E\left[{\boldsymbol{u}_{i,t}^{(c)}(\boldsymbol{\theta})}^3\right] < \infty,$$

which implies $\mathscr{L}_p$-Uniform Integrability, for $p < 3$.

Second, we show the stochastic equicontinuity of $\boldsymbol{u}_{i,t}(y;\boldsymbol{\theta})$, i.e.

$$\lim_{n\to\infty} \sup_{\substack{i\in\mathscr{L}_n \\ t=1,\dots,T}} P\left(\sup_{\substack{\boldsymbol{\theta},\boldsymbol{\theta}'\in\boldsymbol{\Theta} \\ \|\boldsymbol{\theta}-\boldsymbol{\theta}'\|<\delta}} |\boldsymbol{u}_{i,t}(\boldsymbol{\theta}) - \boldsymbol{u}_{i,t}(\boldsymbol{\theta}')| > \varepsilon\right) = \mathbf{0}.$$

The $\nabla \boldsymbol{u}_{i,t}(\boldsymbol{\theta})$ is a $p \times p$ matrix, with each column being either $\dfrac{\partial \boldsymbol{\gamma}_{i,t}(\boldsymbol{\theta})}{\partial \beta^{(c|c')}} \otimes \nabla \boldsymbol{v}_{i,t}$ or $\dfrac{\partial \boldsymbol{\gamma}_{i,t}(\boldsymbol{\theta})}{\partial \alpha^{(c)}} \otimes \nabla \boldsymbol{v}_{i,t}$, and

$$\frac{\partial \boldsymbol{\gamma}_{i,t}(\boldsymbol{\theta})}{\partial \beta^{(c|c')}} = (0,\dots,0,\lambda_{i,t}^{(c)} S_{i,t}^{(c)},0,\dots), \quad \text{and} \quad \frac{\partial \boldsymbol{\gamma}_{i,t}(\boldsymbol{\theta})}{\partial \alpha^{(c)}} = (0,\dots,0,\lambda_{i,t}^{(c)},0,\dots).$$

Thus, the non-zero entries of $E \sup_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} [\nabla \boldsymbol{u}_{i,t}(\boldsymbol{\theta})]$ have the general form: $E \sup_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} \left[\lambda_{i,t}^{(c)} S_{i,t}^{(c)} S_{i,t}^{(c')}\right]$, which are bounded by an equivalent analogous to Lemma 3.

Thirdly, we check $\alpha-$mixing conditions. Let U and V be two subsets of D_n, and let $\sigma(U) = \sigma\{Y_{i,t}; (i,t) \in U\}$ be the $\sigma-$algebra generated by random variables $Y_{i,t}, (i,t) \in U$. Define

$$\alpha(U,V) = \sup\big\{|P(A\cap B) - P(A)P(B)|; A \in \sigma(U), B \in \sigma(V)\big\}.$$

Then the $\alpha-$mixing coefficient for the random field $\{Y_{i,t}, i \in \mathscr{L}_n, t = 1,\dots,T\}$ is defined

as

$$\alpha(k,l,m) = \sup\big\{\alpha(U,V), |U| \le k, |V| \le l, d(U,V) \ge m\big\}.$$

Following Bai et al.Bai et al. (2012), in an $a-$dimensional space, we need (a) $\exists \delta > 0 s.t. \sum_{m=1}^{\infty} m^{a-1}\alpha(1,1,m)^{\delta/(2+\delta)} < \infty$, (b) For $k+l \le 4, \sum_{m=1}^{\infty} m^{a-1}\alpha(k,l,m) < \infty$, (c) $\exists \varepsilon > 0 \quad s.t.\ \alpha(1,\infty,m) = \mathscr{O}(m^{-a-\varepsilon})$, where $k,l,m \in \mathbb{N}$ and $d(U,V) = \min\{\|i-j\| : i \in U, j \in V\}$ is the distance between sets U and V.

For any fixed $i_1,\ldots,i_k \in \mathscr{L}_n, k < \infty$ and $t_1 = 0,\ldots,T$, consider $U = \{Y_{i,t_1} = y_{i,t_1},\ldots,Y_{i_k,t_1} = y_{i_k,t_1}\}$ and $V = \{Y_{j,t_2} = y_{j,t_2}; j \in \mathscr{L}_n, t_2 = 0,\ldots,T\}$, then $|U| = k$ and $|V| \to \infty$ as $n \to \infty$. By Lemma 4, we have $P(Y_{i,t_1} = y_{i,t_1}, Y_{j,t_2} = y_{i,t_1}) - P(Y_{i,t_1} = y_{i,t_1})P(Y_{j,t_2} = y_{j,t_2}) = 0$, if $d(i,j) > t_1 + t_2$. Thus, $\alpha(U,V) = 0$ for any $|U| = k$, provided that $d(U,V) > 2T$, that is, $\alpha(k,\infty,m) = 0$ if $m > 2T$. This implies all three mixing conditions.

Finally, by Theorem 3 in Jenish and Prucha Jenish and Prucha (2009), Uniform Integrability in (2.10) and mixing condition (a) ensure that the score functions $\boldsymbol{u}_{i,t}(\boldsymbol{y};\boldsymbol{\theta})$ satisfy a point wise law of large numbers in the sense that

$$\frac{1}{n_{\mathscr{L}}} \sum_{i \in \mathscr{L}_n} \sum_{t=1}^{T} \sup_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} \Big(\boldsymbol{u}_{i,t}(\boldsymbol{y},\boldsymbol{\theta}) - E\boldsymbol{u}_{i,t}(\boldsymbol{y};\boldsymbol{\theta}) \Big) \xrightarrow{p} \mathbf{0}, \text{as } n_{\mathscr{L}} \to \infty,$$

for all $\boldsymbol{\theta} \in \boldsymbol{\Theta}$. □

Proposition 3. *If the regularity assumptions A.1 and A.2 hold, we have* $\sqrt{n_{\mathscr{L}}}\boldsymbol{V}_n(\boldsymbol{\theta})^{-1/2}(\hat{\boldsymbol{\theta}}_n - \boldsymbol{\theta}_0)$ *converges in distribution to a* $p-$*variate Normal with zero mean vector and identity variance, as* $n_{\mathscr{L}} \to \infty$.

Proof. First, we show the uniform law of large numbers for $\nabla \boldsymbol{u}_n(\boldsymbol{\theta})$:

$$\sup_{\boldsymbol{\theta}} \|\nabla \boldsymbol{u}_n(\boldsymbol{\theta}) - E\left[\nabla \boldsymbol{u}_n(\boldsymbol{\theta})\right]\| \xrightarrow{p} \mathbf{0}, \quad \text{as } n_{\mathscr{L}} \to \infty, \tag{2.11}$$

where $\boldsymbol{u}_n(\boldsymbol{\theta}) = \nabla \ell_n(\boldsymbol{\theta})/n_{\mathscr{L}}$ as defined in Section 2. Note that

$$\begin{aligned}
\mathrm{Var}\left(\nabla \boldsymbol{u}_n(\boldsymbol{\theta})\right) &= \frac{1}{n_{\mathscr{L}}^2} \mathrm{Var}\left(\sum_{i=1}^{n} \sum_{t=1}^{T} \nabla \boldsymbol{u}_{i,t}(\boldsymbol{\theta}) \right) \\
&= \frac{1}{n_{\mathscr{L}}^2} \sum_{i \in \mathscr{L}_n} \sum_{t=1}^{T} \mathrm{Var}\left(\nabla \boldsymbol{u}_{i,t}(\boldsymbol{\theta})\right) \\
&\quad + \frac{1}{n_{\mathscr{L}}^2} \sum_{i \in \mathscr{L}_n} \sum_{t_1=1}^{T} \sum_{\substack{j \in \mathscr{L}_n \\ j \neq i}} \sum_{\substack{t_2=1 \\ t_2 \neq t_1}}^{T} \mathrm{Cov}\left(\nabla \boldsymbol{u}_{i,t_1}(\boldsymbol{\theta}), \nabla \boldsymbol{u}_{j,t_2}(\boldsymbol{\theta})\right)
\end{aligned} \tag{2.12}$$

The first term in (2.12) is $\mathscr{O}(n_{\mathscr{L}}^{-1})$, since $\mathrm{Var}\left(\nabla \boldsymbol{u}_{i,t}(\boldsymbol{\theta})\right) \le \left[E\left(\nabla \boldsymbol{u}_{i,t}(\boldsymbol{\theta})\right)\right]^2$, which is shown to be finite in the proof of Proposition 2.

For the second term in (2.12), by Lemma 2 we have

$$\frac{1}{n_{\mathscr{L}}^2}\sum_{i\in\mathscr{L}_n}\sum_{t_1=1}^{T}\sum_{\substack{j\in\mathscr{L}_n\\ j\neq i}}\sum_{\substack{t_2=1\\ t_2\neq t_1}}^{T}\mathrm{Cov}\left(\nabla \boldsymbol{u}_{i,t_1}(\boldsymbol{\theta}),\nabla \boldsymbol{u}_{j,t_2}(\boldsymbol{\theta})\right)$$
$$\leq \frac{1}{n_{\mathscr{L}}^2}\sum_{i\in\mathscr{L}_n}\sum_{t_1=1}^{T} T(8T^2+4T+1)\max_{\substack{j:d(i,j)\leq 2T\\ t_2\neq t_1}}\mathrm{Cov}\left(\nabla \boldsymbol{u}_{i,t_1}(\boldsymbol{\theta}),\nabla \boldsymbol{u}_{j,t_2}(\boldsymbol{\theta})\right),$$

where $\mathrm{Cov}\left(\nabla \boldsymbol{u}_{i,t_1}(\boldsymbol{\theta}),\nabla \boldsymbol{u}_{j,t_2}(\boldsymbol{\theta})\right)\leq E\left(\nabla \boldsymbol{u}_{i,t_1}(\boldsymbol{\theta}),\nabla \boldsymbol{u}_{j,t_2}(\boldsymbol{\theta})\right)\leq E\left(\nabla \boldsymbol{u}_{i,t_1}(\boldsymbol{\theta})\right)^2+E\left(\nabla \boldsymbol{u}_{i,t_2}(\boldsymbol{\theta})\right)^2$ is finite by Lemma 2. Thus, the second term in (2.12) is also of order $\mathscr{O}(n_{\mathscr{L}}^{-1})$ element wise, which means $\mathrm{Var}\left(\nabla \boldsymbol{u}_n(\boldsymbol{\theta})\right)\to \mathbf{0}$ as $n\to\infty$. Therefore, (2.11) follows by Chebyshev's inequality.

Second, $\boldsymbol{V}_n(\boldsymbol{\theta})=1/n_{\mathscr{L}}\mathrm{Var}\left(\sum_{i\in\mathscr{L}_n}\sum_{t=1}^{T}\boldsymbol{u}_{it}(\boldsymbol{\theta})\right)=-1/n_{\mathscr{L}}E\left(\sum_{i\in\mathscr{L}_n}\sum_{t=1}^{T}\boldsymbol{u}_{it}(\boldsymbol{\theta})\right)=1/n_{\mathscr{L}}\boldsymbol{H}_n(\boldsymbol{\theta})$, which is shown to be positive definite under Assumption A.2 in Proposition 1. Thus, together with uniform Integrability in (2.10) and the mixing conditions, by Theorem 1 in Jenish and Prucha (2009), we have

$$\sqrt{n_{\mathscr{L}}}\boldsymbol{V}_n(\boldsymbol{\theta})^{-1/2}\boldsymbol{u}_n(\boldsymbol{\theta})\to N(\mathbf{0},\boldsymbol{I}_p) \tag{2.13}$$

Finally, by Taylor's expansion,

$$\begin{aligned}\boldsymbol{u}_n(\hat{\boldsymbol{\theta}}_n)=\mathbf{0}=&\boldsymbol{u}_n(\boldsymbol{\theta}_0)+\nabla \boldsymbol{u}_n(\boldsymbol{\theta}_0)(\hat{\boldsymbol{\theta}}_n-\boldsymbol{\theta}_0)+\frac{1}{2}\nabla^2\boldsymbol{u}_n(\boldsymbol{\theta}_0)(\tilde{\boldsymbol{\theta}}_n-\boldsymbol{\theta}_0)^2\\ \Rightarrow \mathbf{0}=&\sqrt{n_{\mathscr{L}}}\boldsymbol{V}_n(\boldsymbol{\theta}_0)^{-1/2}\boldsymbol{u}_n(\boldsymbol{\theta}_0)+\sqrt{n_{\mathscr{L}}}\boldsymbol{V}_n(\boldsymbol{\theta})^{-1/2}\nabla \boldsymbol{u}_n(\boldsymbol{\theta}_0)(\hat{\boldsymbol{\theta}}_n-\boldsymbol{\theta}_0)+\\ &\frac{1}{2}\sqrt{n_{\mathscr{L}}}\boldsymbol{V}_n(\boldsymbol{\theta}_0)^{-1/2}\nabla^2\boldsymbol{u}_n(\tilde{\boldsymbol{\theta}}_n)(\hat{\boldsymbol{\theta}}_n-\boldsymbol{\theta}_0)^2,\end{aligned} \tag{2.14}$$

where $\tilde{\boldsymbol{\theta}}_n$ is a vector with elements between $\hat{\boldsymbol{\theta}}_n$ and $\boldsymbol{\theta}_0$. Since $\hat{\boldsymbol{\theta}}_n=\boldsymbol{\theta}_0+o_p(\mathbf{1})$ by Proposition 2, we have $(\tilde{\boldsymbol{\theta}}_n-\boldsymbol{\theta}_0)^2=(\hat{\boldsymbol{\theta}}_n-\boldsymbol{\theta}_0)o_p(\mathbf{1})$. The second derivative $\nabla^2\boldsymbol{u}_n(\boldsymbol{\theta})$ is a $p\times p\times p$ matrix, with entries being either 0 or $\lambda_{it}^{(c)}S_{i,t-1}^{(c_1)}S_{i,t-1}^{(c_2)}S_{i,t-1}^{(c_3)}$, where $i=1,\ldots,n$, and $t=1,\ldots,T$, and $c,c_1,c_2,c_3\in\mathscr{C}$. Due to the structure of $\lambda_{it}^{(c)}$ and $S_{i,t-1}^{(c)}$ in Section 2, all non-zero elements in $\nabla^2\boldsymbol{u}_n(\boldsymbol{\theta})$ are monotone with respect to $\boldsymbol{\theta}$. Thus, there exists $\boldsymbol{\theta}_s\in\boldsymbol{\Theta}$ such that $\nabla^2\boldsymbol{u}_n(\boldsymbol{\theta}_s)\geq\nabla^2\boldsymbol{u}_n(\boldsymbol{\theta})$ for all $\boldsymbol{\theta}\in\boldsymbol{\Theta}$. Therefore, we have $E\sup_{\boldsymbol{\theta}\in\boldsymbol{\Theta}}\nabla^2\boldsymbol{u}_n(\boldsymbol{\theta})=\sup_{\boldsymbol{\theta}\in\boldsymbol{\Theta}}E\nabla^2\boldsymbol{u}_n(\boldsymbol{\theta})$, which can be shown to be finite by an equivalent analogous to Lemma 3.

Thus, (2.14) can be written as

$$\mathbf{0}=\sqrt{n_{\mathscr{L}}}\boldsymbol{V}_n(\boldsymbol{\theta}_0)^{-1/2}\boldsymbol{u}_n(\boldsymbol{\theta}_0)+\sqrt{n_{\mathscr{L}}}\boldsymbol{V}_n(\boldsymbol{\theta})^{-1/2}\left(\nabla \boldsymbol{u}_n(\boldsymbol{\theta}_0)+o_p(\mathbf{1})\right)(\hat{\boldsymbol{\theta}}_n-\boldsymbol{\theta}_0),$$

By (2.11), $\nabla \boldsymbol{u}_n(\boldsymbol{\theta}_0)\xrightarrow{p}E\left[\nabla \boldsymbol{u}_n(\boldsymbol{\theta}_0)\right]=-\boldsymbol{V}_n(\boldsymbol{\theta}_0)$, since $\ell_n(\boldsymbol{\theta})$ is the full likelihood. There-

fore, by (2.13) and (2.14), we have

$$\sqrt{n\mathscr{L}}\boldsymbol{V}_n(\boldsymbol{\theta}_0)^{1/2}(\hat{\boldsymbol{\theta}}_n - \boldsymbol{\theta}_0) \xrightarrow{d} N(\mathbf{0}, \boldsymbol{I}_p).$$

□

Chapter 3

A Copula-based Multivariate Spatio-temporal Model (copSTM)

3.1 Introduction

Correlated data with discrete marginal distributions are typically modelled by the generalized estimating equations (GEE) approach (Liang and Zeger, 1986). The GEE method is suitable when the regression coefficients in the marginal model is of central interest since it treats dependence parameters as nuisance components, and thus not appropriate if the estimation of correlation parameters is also important. One popular approach for evaluating the dependence structure is through latent models under the Bayesian framework by assuming a multivariate Gaussian process for the conditional marginal expectation of the responses, However, estimation of latent models requires the MCMC algorithms that are known to be very time consuming (Davis et al., 2003).

The maximum likelihood (ML) approach for inference are generally considered one of the most statistically efficient options for estimating the model parameters. However, likelihood analysis of discrete marginal regression models is less widespread (Diggle et al., 2002). The main reason is the difficulty in identifying the general class of multivariate distributions for discrete responses. Gaussian copulas (Xue-Kun Song, 2000) provide a flexible general framework for modelling dependent responses of any distributions by combining the marginal regression modelling with the separate specification of dependence structure. They can accommodate full dependence with correlation coefficients approaching one, and full independence with zero correlation. They also allow for positive and negative correlations. Such level of felxibility is not offered by other copulas such as Archimedean copulas (Kazianka and Pilz, 2010).

Despite all the merits, Gaussian copula regression models had still a limited use since the evaluation of the likelihood function for discrete dependent responses requires approximation of high-dimensional integrals. One possible solution is to employ the simulation

methods, for example, Masarotto et al. (2012) adopted a sequential importance sampling algorithm and Nikoloulopoulos (2013) studied simulated maximum likelihood based on randomized quasi-Monte Carlo integration.

In this chapter, we adopt the composite likelihood methods (CL) to reduce the computational burdens by reducing the integral dimensionality in the likelihood function. A composite likelihood is constructed from low-dimensional likelihood objects defined over small subsets of data. Besag (1974) was an early proponent of composite likelihood estimation for data with spatial dependence; Lindsay (1988) developed composite likelihood inference in its generality and systematically studied its properties. Over the years, composite likelihood methods have been demonstrated to have desirable theoretical properties including estimation consistency and asymptotic normality, and have been successfully used in a range of complex applications, including models for survival data, genetics (Gao and Song, 2010) and spatial statistics (Heagerty and Lele, 1998; Varin and Vidoni, 2005; Bai et al., 2014). See Varin et al. (2011) for a comprehensive survey in this regard. In this chapter, we carry out model estimation through a pairwise composite likelihood approach that reaches satisfactory balance between statistical efficiency and computational complexity.

The following is a brief summary of the chapter. Section 3.2, we develop the copula based spatio-temporal model on lattice for multivariate count data and provide composite likelihood inference tools. In Section 3.3, we study the performance of the composite likelihood estimator using simulated data. In Section 3.4, we apply our method to analyze the RGB marked cancer cell growth data. In Section 3.5, we conclude and give possible future directions.

3.2 The Gaussian copula spatio-tempoal multivariate model on lattice (copSTM)

Let $\mathscr{L} \in \mathbb{N}^2$ be a discrete regular lattice. In the context of our application, the lattice is obtained by tiling a microscope image into $n_{\mathscr{L}}$ rectangular tiles, denoted by $\mathscr{L}_n(\subset \mathscr{L})$. The total number of tiles $n_{\mathscr{L}}$ is a monotonically increasing function of n. For simplicity, we tile the image into $n \times n$ tiles, that is, $n_{\mathscr{L}} = n^2$. Denote a pair of neighbouring tiles $\{i, j\}$ with $i \sim j$, if the two tiles are in the neighbourhood of each other. Here we take the Moore neighbourhood that is composed of a central tile and the eight tiles surrounding it. For example, in Figure 3.1, the blue tiles form a complete neighbourhood.

Let $Y_{i,t}^{(c)}$ represent a discrete random variable and $y_{i,t}^{(c)}$ be the corresponding realisations, where subscript t denotes discrete time $(t = 1, \ldots, T)$, i indexes tiles in space $(i = 1, \ldots, n_{\mathscr{L}})$ and superscript c indexes different groups of interest $(c = 1, \ldots, n_{\mathscr{C}})$. Consider $\{\boldsymbol{Y}_t\}$ as a multivariate time series, where $\boldsymbol{Y}_t = (\boldsymbol{Y}_{1,t}, \ldots, \boldsymbol{Y}_{n_{\mathscr{L}},t})$ and $\boldsymbol{Y}_{i,t} = (Y_{i,t}^{(1)}, \ldots, Y_{i,t}^{(n_{\mathscr{C}})})$.

Thus, $\boldsymbol{Y}_t$ is a d-dimensional vector of random variables, where $d = n_{\mathscr{C}} n_{\mathscr{L}}$. Let $\boldsymbol{X}_t = (\boldsymbol{X}_{1,t}^{(1)}, \ldots, \boldsymbol{X}_{n_{\mathscr{L}},t}^{(n_{\mathscr{C}})})$ be the d-dimensional explanatory variables or covariates, and $\mathscr{F}_t = \sigma\{\boldsymbol{Y}_{t-1}, \ldots, \boldsymbol{Y}_1, \boldsymbol{X}_t, \ldots\}$ represent all the information that is known to the observer before time t, which include past responses, as well as past and present covariates.

The model structure of the copSTM is demonstrated in Figure 3.1, where each plane/lattice represents a multivariate response variable at one time point $\boldsymbol{Y}_t$. Suppose the variable represented by the red tile is a $n_{\mathscr{C}}$-dimensional vector $\boldsymbol{Y}_{i,t}$, we model the temporal dependence through a marginal regression model. Specifically, we assume that the expectation of the conditional marginal distribution $E[Y_{i,t}^{(c)}|\mathscr{F}_t], c = 1, \ldots, n_{\mathscr{C}}$ depends on observations in the neighbourhood of the previous time point, $\boldsymbol{Y}_{j,t-1}, j \sim i$, that is the blue tiles. On the other hand, the spatial dependence is captured by a $d \times d$ correlation matrix of Y_t through the Gaussian copula model. For the computational concern, we only consider spatial as well as cross group correlations in neighbouring locations, that is for $\boldsymbol{Y}_{i,t}$, we only consider correlation with the green tiles, $\boldsymbol{Y}_{j,t}, j \sim i$.

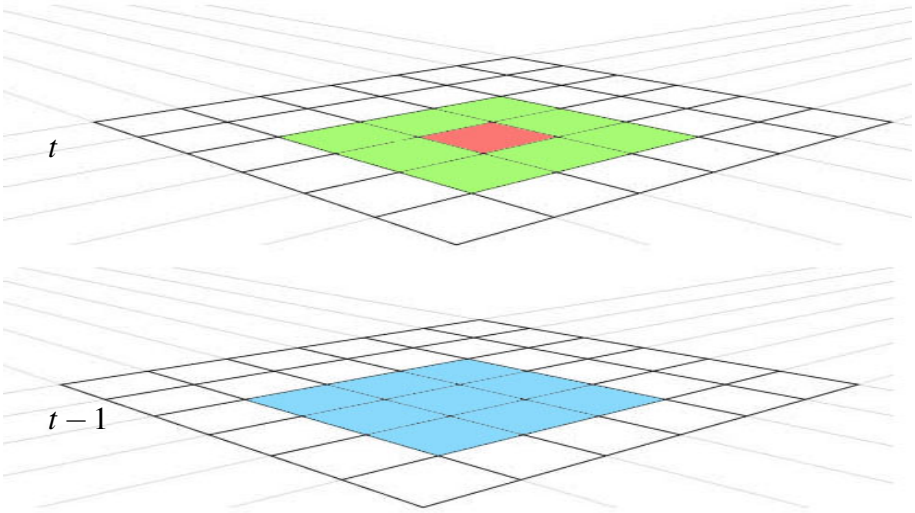


Figure 3.1: Illustration of the spatial and temporal structure of the copSTM model.

3.2.1 Observation-driven autoregressive model

The temporal part of the copSTM is captured through the conditional marginal expectations of $Y_{i,t}^{(c)}$ by an autoregressive model that falls into a general class of time series models, the generalised autoregressive moving average model, GARMA(p,q), proposed by Benjamin et al. (2003). Following notation of Benjamin et al. (2003), the marginal distribution of response $Y_{i,t}^{(c)}$ conditional of $\mathscr{F}_t$ is assumed to be a member of the exponential family with expectation $\mu_{i,t}^{(c)} = E[Y_{i,t}^{(c)}|\mathscr{F}_t]$. The most general form of the GARMA(p,q)

model is defined as

$$g(\mu_t) = X_t\alpha + \sum_{t'=1}^{p} \beta_{t'} H_{t'}(y_{t-t'}) + \sum_{t'=1}^{q} \gamma_{t'} D_{t'}(\mu_{t-t'}),$$

where $g(\cdot)$ is the canonical link function, $H_{t'}(\cdot)$ and $D_{t'}(\cdot)$ are known functions for all t'. Benjamin et al. (2003) also state that the GARMA model of the above form " is too general for practical application". Thus, in this chapter we focus on a special case of the GARMA model with $p = 1$, $q = 0$ and the logarithm link function to form a log-linear model

$$\log(\mu_{i,t}^{(c)}) = \beta_0^{(c)} + \sum_{c'=1}^{n_{\mathscr{C}}} \beta^{(c|c')} H(y_{i,t-1}^{(c')}), \tag{3.1}$$

where $H(y_{i,t-1}^{(c')}) = \left[\sum_{i\sim j} \log\left(1 + Y_{j,t-1}^{(c')}\right)\right] / n_i$. Transforming past observations by log so that they are on the same scale as the linear predictor $\log(\mu_{i,t}^{(c)})$. Several authors show that the addition of a constant to each observation for avoiding zero values does not affect inference, and that 1 is a reasonable choice for the constant since it conveniently maps zero count with zero values of $H(\cdot)$ (Knorr-Held and Richardson, 2003; Fokianos and Tjøstheim, 2011). Apart from slightly different notations, this expression is similar to the model proposed in Chapter 2. Regression coefficients $\beta^{(c|c')}$ are interpreted as impacts on the growth of group c due to the presence of group c' in the neighbourhood at the previous time period and can be presented in the same kind of incidence matrix as in Chapter 2 with the cth column being $\boldsymbol{\beta}_c = (\beta_0^{(c)}, \beta^{(c|1)}, \ldots, \beta^{(c|n_{\mathscr{C}})})$, where $\beta_0^{(c)}$ is the intercept of group c.

Assume that $Y_{i,t}^{(c)}$ given the past is marginally Poisson distributed, i.e. $Y_{i,t}^{(c)}|\mathscr{F}_{t-1} \sim$ Poisson$(\mu_{i,t}^{(c)})$ then it implies that

$$P(Y_{i,t}^{(c)} = y|\mathscr{F}_{t-1}) = \frac{\mu_{i,t}^{(c)^y} \exp(-\mu_{i,t}^{(c)})}{y!}, \quad y = 0, 1, ...$$

and Var$(Y_{i,t}^{(c)}|\mathscr{F}_{t-1})$ = E$(Y_{i,t}^{(c)}|\mathscr{F}_{t-1})$ = $\mu_{i,t}^{(c)}$. Hence, in the case of a conditional Poisson response marginal model, the conditional mean is identical to the conditional variance of the observed variable. The Negative binomial distribution allows for a conditional variance to be larger than the mean $\mu_{i,t}^{(c)}$, which is often referred to as overdispersion. Following Christou and Fokianos (2014), it is assumed that $Y_{i,t}^{(c)}|\mathscr{F}_{t-1} \sim \text{NegBin}(\mu_{i,t}^{(c)}, \phi)$, where the negative binomial distribution is parameterized in terms of its mean with an additional dispersion parameter $\phi > 0$,

$$P(Y_{i,t}^{(c)} = y|\mathscr{F}_{t-1}) = \frac{\Gamma(\phi + y)}{\Gamma(y+1)\Gamma(\phi)} \left(\frac{\phi}{\phi + \mu_{i,t}^{(c)}}\right)^{\phi} \left(\frac{\mu_{i,t}^{(c)}}{\phi + \mu_{i,t}^{(c)}}\right)^{y}, \quad y = 0, 1, ...$$

In this case, $\text{Var}(Y_{i,t}^{(c)}|\mathscr{F}_{t-1}) = \mu_{i,t}^{(c)} + {\mu_{i,t}^{(c)}}^2/\phi$, and thus the conditional variance increases quadratically with $\mu_{i,t}^{(c)}$. The Poisson distribution is a limiting case of the negative binomial when $\phi \to \infty$. In this chapter and the rest of this thesis, we focus on Poisson and Negative binomial marginal distributed responses for our models, but parameters of interest are regression coefficients $\boldsymbol{\beta}$ and correlation parameters discussed in the following section, while the dispersion parameter ϕ is considered as nuisance parameter.

3.2.2 Gaussian-copula model

To account for spatial and cross variable dependence at the same time point, we use a copula-based model, which allows us to model the marginal distributions and correlation structure separately before joining them by way of the probability integral transform (Sklar, 1959). Consider a random vector $\boldsymbol{Y} = (Y_1, \ldots, Y_d)$ with joint distribution F and write $F_i = F_i(y_i|\boldsymbol{\beta}, x_i)$ for the marginal distribution of Y_i. A copula associated with F (equivalently, Y) is a function C that satisfies $F(y) = C(F_1(y_1), \ldots, F_d(y_d)), y = (y_1, \ldots, y_d) \in \mathbb{R}^d$.

In this chapter, we focus our attention to Gaussian copulas introduced by Xue-Kun Song (2000), similar to the model developed by Masarotto et al. (2012). Specifically, let $F_{\mu_{i,t}^{(c)}}$ be the marginal cumulative distribution function of $Y_{i,t}^{(c)}|Y_{t-1}$ and $\boldsymbol{\Sigma}$ be a working $d \times d$ correlation matrix independent of t, then we have the copula representation

$$\begin{aligned}
\boldsymbol{Z} &= (Z_1, \ldots, Z_d) \sim MVN(0, \boldsymbol{\Sigma}), \\
\boldsymbol{Y}_t|\boldsymbol{Y}_{t-1} &= (Y_{1,t}^{(1)}, \ldots, Y_{1,t}^{(n_{\mathscr{C}})}, Y_{2,t}^{(1)}, \ldots, Y_{n_{\mathscr{L}},t}^{(n_{\mathscr{C}})}|\boldsymbol{Y}_{t-1}) \\
&= \left(F^{-1}_{\mu_{1,t}^{(1)}}\left[\Phi(Z_1)\right], \ldots, F^{-1}_{\mu_{1,t}^{(n_{\mathscr{C}})}}\left[\Phi(Z_{n_{\mathscr{C}}})\right], F^{-1}_{\mu_{2,t}^{(1)}}\left[\Phi(Z_{n_{\mathscr{C}}+1})\right], \ldots, F^{-1}_{\mu_{n_{\mathscr{L}},t}^{(n_{\mathscr{C}})}}\left[\Phi(Z_d)\right] \right),
\end{aligned} \tag{3.2}$$

where Φ is the standard normal cdf and $F_i^{-1}(u) = \inf\{y : F_i(y) \leq u\}$ for $0 \leq u \leq 1$. It is worth noting that the mapping between Y_i and Z_i is one-to-one only in the continuous case, in other cases, the mapping is one-to many. The copula model specification allows any kind of continuous, discrete and categorical marginal distributions. It conveniently models the marginal component in (3.1) and the dependence component $\boldsymbol{\Sigma}$ separately, allowing for extremely flexible correlation structures.

Various forms of dependence in the data can be modelled by suitably parameterising the correlation matrix $\boldsymbol{\Sigma}$ as a function of a vector of parameter $\boldsymbol{\rho}$. In this thesis, we consider a straightforward construction of the correlation structure but the model can easily be generalized to handle any other isotropic correlation functions, for instance, the popular Matérn family or the power exponential family. The dependence structure is built

based on the assumption that observations in the same neighbourhood are more correlated than those that are further apart. Thus, we specify correlation parameters only for those variables in the same or neighbouring tiles:

$$\text{Cor}\left(Y_{i,t}^{(c)}, Y_{j,t}^{(c')} | \boldsymbol{Y}_{t-1}\right) = \begin{cases} 1 & \text{if } i = j, c = c' \\ \rho_0^{(c,c')} & \text{if } i = j, c \neq c' \\ \rho_1^{(c,c')} & \text{if } i \sim j, c \neq c' \\ \rho_0^{(c)} & \text{if } i \sim j, c = c' \end{cases},$$

where $\rho_0^{(c)}$ denote spatial correlation of the same group between neighbouring tiles, $\rho_0^{(c,c')}$ and $\rho_1^{(c,c')}$ denote correlation between group c and c' in the same tile and neighbouring tiles respectively. Correlations between variables in non-neighbouring tiles are not parameterized, thus are considered as independent in the composite likelihood specified in the following section. But we show in our numerical examples that the model estimate is relatively stable with mild correlations between non-neighbouring tiles.

With this parameterization, $\boldsymbol{\Sigma}$ takes the form of a block adjacency matrix

$$\begin{pmatrix} \boldsymbol{R}_0 & \boldsymbol{R}_1 & \boldsymbol{0} & \boldsymbol{0} & \dots & \boldsymbol{R}_1 & \boldsymbol{0} & \dots & \dots \\ \boldsymbol{R}_1 & \boldsymbol{R}_0 & \boldsymbol{R}_1 & \boldsymbol{0} & \dots & \dots & \boldsymbol{R}_1 & \dots & \dots \\ \boldsymbol{0} & \boldsymbol{R}_1 & \boldsymbol{R}_0 & \boldsymbol{R}_1 & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \boldsymbol{R}_0 \end{pmatrix}_{n_{\mathscr{L}} \times n_{\mathscr{L}}},$$

where each entry is a $n_{\mathscr{C}} \times n_{\mathscr{C}}$ matrix representing cross group/cluster correlation in the same tile $\boldsymbol{R}_0$ and neighbouring tiles $\boldsymbol{R}_1$ parameterized as

$$\boldsymbol{R}_0 = \begin{pmatrix} 1 & \rho_0^{(1,2)} & \rho_0^{(1,3)} & \dots & \rho_0^{(1,n_{\mathscr{C}})} \\ \rho_0^{(1,2)} & 1 & \rho_0^{(2,3)} & \dots & \rho_0^{(2,n_{\mathscr{C}})} \\ \dots & \dots & \dots & \dots & \dots \\ \rho_0^{(1,n_{\mathscr{C}})} & \dots & \dots & \dots & 1 \end{pmatrix}, \quad \boldsymbol{R}_1 = \begin{pmatrix} \rho_1^{(1)} & \rho_1^{(1,2)} & \rho_1^{(1,3)} & \dots & \rho_1^{(1,n_{\mathscr{C}})} \\ \rho_1^{(1,2)} & \rho_1^{(2)} & \rho_1^{(2,3)} & \dots & \rho_1^{(2,n_{\mathscr{C}})} \\ \dots & \dots & \dots & \dots & \dots \\ \rho_1^{(1,n_{\mathscr{C}})} & \dots & \dots & \dots & \rho_1^{(n_{\mathscr{C}})} \end{pmatrix}.$$

3.2.3 Composite likelihood inference with discrete marginals

The joint likelihood of the copSTM model is defined as

$$\begin{aligned} L(\boldsymbol{\theta}) &= \prod_{t=1}^{T} P\left(Y_{i,t}^{(c)} | \boldsymbol{Y}_{t-1}; \boldsymbol{\theta}\right) \\ &= \prod_{t=1}^{T} P\left(Y_{1,t}^{(1)} = y_{1,t}^{(1)}, \ldots, Y_{n_{\mathscr{L}},t}^{(n_{\mathscr{C}})} = y_{n_{\mathscr{L}},t}^{(n_{\mathscr{C}})} | \boldsymbol{Y}_{t-1}\right) \\ &= \prod_{t=1}^{T} P\left(y_{1,t}^{(1)} - 1 < Y_{1,t}^{(1)} \leq y_{1,t}^{(1)}, \ldots, y_{n_{\mathscr{L}},t}^{(n_{\mathscr{C}})} < Y_{n_{\mathscr{L}},t}^{(n_{\mathscr{C}})} \leq y_{n_{\mathscr{L}},t}^{(n_{\mathscr{C}})} | \boldsymbol{Y}_{t-1}\right) \\ &= \prod_{t=1}^{T} \int_{a_{1,t}^{(1)}}^{b_{1,t}^{(1)}} \cdots \int_{a_{n_{\mathscr{L}},t}^{(n_{\mathscr{C}})}}^{b_{n_{\mathscr{L}},t}^{(n_{\mathscr{C}})}} f_{\boldsymbol{\Sigma}}(z_1, \ldots, z_d) dz_1 \ldots dz_d, \end{aligned} \tag{3.3}$$

where the upper and lower bounds of the integrals are $b_{i,t}^{(c)} = b_{i,t}^{(c)}(\boldsymbol{\beta}^{(c)}, y_{i,t}^{(c)}) = \Phi^{-1}[F_{\mu_{i,t}^{(c)}}(y_{i,t}^{(c)})]$ and $a_{i,t}^{(c)} = a_{i,t}^{(c)}(\boldsymbol{\beta}^{(c)}, y_{i,t}^{(c)}) = \Phi^{-1}[F_{\mu_{i,t}^{(c)}}(y_{i,t}^{(c)} - 1)]$ respectively, and $f_{\boldsymbol{\Sigma}}$ denotes the density function of a d-dimensional multivariate normal distribution, with mean zero, and correlation matrix $\boldsymbol{\Sigma}$ specified in Section 3.2.2. The full parameter vector $\boldsymbol{\theta} = (\boldsymbol{\beta}, \boldsymbol{\rho})'$, where $\boldsymbol{\beta} = (\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_{n_{\mathscr{C}}})'$ and $\boldsymbol{\rho} = (\rho^{(1)}, \ldots, \rho^{(n_{\mathscr{C}})}, \rho_0^{(1,2)}, \ldots, \rho_0^{(n_{\mathscr{C}}-1, n_{\mathscr{C}})}, \rho_1^{(1,2)}, \ldots, \rho_1^{(n_{\mathscr{C}}-1, n_{\mathscr{C}})})$ with dimension $p = n_{\mathscr{C}}(n_{\mathscr{C}} + 1)$ (for $\boldsymbol{\beta}$) $+ n_{\mathscr{C}}^2$ (for $\boldsymbol{\rho}$).

The difficulty with the usual maximum likelihood method is apparent, because the rectangle probability is difficult to compute accurately in high dimensions, making the optimization of (3.3) computationally intractable for large d. In contrast to the full joint likelihood (3.3), the composite likelihood (CL) function (Lindsay, 1988) is constructed from likelihoods of subsets of the data, proceeding as though these subsets were independent, thus reducing the computational burden. Here, we adopt the pairwise CL as a compromise between statistical and computational efficiency.

In consideration of computational efficiency, as well as the correlation structure assumed in Section 3.2.2, we allow only pairs of observations in neighbouring tiles. Thus, the pairwise log-CL function is

$$cl(\boldsymbol{\theta}; \boldsymbol{y}) = \sum_{t=1}^{T} \sum_{\substack{i_1 \sim i_2 \\ i_1, i_2 \in \{1, \ldots, n_{\mathscr{L}}\}}} \sum_{c_1=1}^{n_{\mathscr{C}}} \sum_{c_2=1}^{n_{\mathscr{C}}} \log \left[\int_{a_{i_1,t}^{(c_1)}}^{b_{i_1,t}^{(c_1)}} \int_{a_{i_2,t}^{(c_2)}}^{b_{i_2,t}^{(c_2)}} f_{\rho}(z_1, z_2) dz_1 dz_2 \right], \tag{3.4}$$

where $f_{\rho}(z_1, z_2)$ denotes a 2-dimensional multivariate normal density function, where the mean is vector zero and the correlation is ρ, which equals the correlation between $Y_{i_1,t}^{(c_1)}$ and $Y_{i_2,t}^{(c_2)}$ defined in Section 3.2.2. Note that there is exactly one correlation parameter in each component likelihood.

The maximum CL estimates of $\boldsymbol{\theta}$ are obtained by maximizing $cl(\boldsymbol{\theta}; \boldsymbol{y})$, which proceeds iteratively using Fisher-scoring updates: $\hat{\boldsymbol{\theta}}^{(k+1)} = \hat{\boldsymbol{\theta}}^{(k)} + \boldsymbol{H}(\hat{\boldsymbol{\theta}}^{(k)}; \boldsymbol{y})^{-1} \boldsymbol{u}(\hat{\boldsymbol{\theta}}^{(k)}; \boldsymbol{y})$,

where

$$\boldsymbol{H}(\boldsymbol{\theta};\boldsymbol{y}) = -E\left[\frac{\partial^2 cl(\boldsymbol{\theta};\boldsymbol{y})}{\partial \boldsymbol{\theta}^2}\right], \quad \boldsymbol{u}(\boldsymbol{\theta};\boldsymbol{y}) = \frac{\partial cl(\boldsymbol{\theta};\boldsymbol{y})}{\partial \boldsymbol{\theta}}, \tag{3.5}$$

are the Hessian matrix and score vector respectively. Except for the integral bounds $a_{i,t}^{(c)}$ and $b_{i,t}^{(c)}$, the first and second derivatives of $cl(\boldsymbol{\theta};\boldsymbol{y})$ can be derived analytically (see Appendix for details).

3.2.4 Standard error estimation

As the data dimension $d \times T$ increases, theory of unbiased estimating equations (Godambe, 1960) suggested that under appropriate regularity conditions, the estimator $\hat{\boldsymbol{\theta}}$ follows approximately a p-variate normal distribution with mean equal to the true parameter $\boldsymbol{\theta}_0$ and asymptotic variance $\boldsymbol{V}(\hat{\boldsymbol{\theta}}) = \boldsymbol{G}^{-1}(\boldsymbol{\theta}_0)$, where $\boldsymbol{G}(\hat{\boldsymbol{\theta}}) = \boldsymbol{H}(\hat{\boldsymbol{\theta}})\boldsymbol{K}^{-1}(\hat{\boldsymbol{\theta}})\boldsymbol{H}(\hat{\boldsymbol{\theta}})$ is the $p \times p$ Godambe information matrix associated with the log-CL function (3.4), $\boldsymbol{H}(\boldsymbol{\theta})$ is defined in (3.5) and $\boldsymbol{K}(\boldsymbol{\theta}) = \text{Var}[\boldsymbol{u}(\boldsymbol{\theta};\boldsymbol{y})]$. If $cl(\boldsymbol{\theta};\boldsymbol{y})$ is a true log-likelihood function, then $\boldsymbol{H} = \boldsymbol{K}$.

The estimation of $\boldsymbol{H}(\boldsymbol{\theta})$ is relatively straightforward, with the details shown in the Appendix, difficulties arise for the variability matrix $\boldsymbol{K}(\boldsymbol{\theta})$. Similar to Varin et al. (2011) Section 5.1 and Cattelan and Sartori (2016), we estimate $\boldsymbol{K}(\boldsymbol{\theta})$ via a parametric bootstrap approach. Specifically, the estimator of $\boldsymbol{K}(\boldsymbol{\theta})$ is

$$\hat{\boldsymbol{K}}(\boldsymbol{\theta}) = \frac{1}{B}\sum_{b=1}^{B} \boldsymbol{u}(\hat{\boldsymbol{\theta}};\boldsymbol{y}^*_{(b)})\boldsymbol{u}(\hat{\boldsymbol{\theta}};\boldsymbol{y}^*_{(b)})^T, \tag{3.6}$$

where $\boldsymbol{y}^*_{(b)}$ denote the bth bootstrap sample simulated from the copSTM model at $\boldsymbol{\theta} = \hat{\boldsymbol{\theta}}$.

3.2.5 Goodness of fit test

To test the goodness of fit of the estimated model, we carry out a composite likelihood ratio test, with test statistic

$$W = 2\left[cl(\hat{\boldsymbol{\theta}};\boldsymbol{y}) - cl(\boldsymbol{\theta}_0;\boldsymbol{y})\right].$$

Its asymptotic distribution is a weighted sum of the square of p independent standard normal variables, precisely $\sum_{i=1}^{p} w_i Z_i^2$, where $w_1,\ldots,w_p$ are the eigenvalues of $[\boldsymbol{H}(\boldsymbol{\theta})]^{-1}\boldsymbol{G}(\hat{\boldsymbol{\theta}})$ (Varin et al., 2011).

The non-standard distribution prevents the use of composite likelihood ratio test statistics when $p > 1$. Thus, we adopt two adjusted likelihood ratio tests:

$$W_1 = \frac{W}{\bar{w}} \sim \chi^2_p, \text{ where } \bar{w} = \frac{\sum_{i=1}^{p} w_i}{p}, \tag{3.7}$$

$$W_2 = \frac{vW}{(p\bar{w})} \sim \chi^2_v, \text{ where } v = \frac{(\sum_{i=1}^{p} w_i)^2}{\sum_{i=1}^{p} w_i^2}, \tag{3.8}$$

proposed by Geys et al. (1999) and Satterthwaite (1946) respectively.

3.3 Monte Carlo simulations

First, to assess the performance of the pairwise CL estimator of the copSTM, we carry out a simulation experiment with settings for different number of tiles (i.e. $n_{\mathscr{L}} = 10 \times 10, 25 \times 25$) and time points (i.e. $T = 10, 25$). We fix $n_{\mathscr{C}}$ to be 3, which creates 21 parameters (12 regression coefficients and 9 correlation parameters). We also consider different setups of the correlation parameters $(\rho_1^{(1)}, \rho_1^{(2)}, \rho_1^{(3)}, \rho_0^{(1,2)}, \rho_0^{(1,3)}, \rho_0^{(2,3)}, \rho_1^{(1,2)}, \rho_1^{(1,3)}, \rho_1^{(2,3)})$, specifically, let $\boldsymbol{\rho}_a = (0, 0, 0, -0.6, 0.3, -0.2, 0, 0, 0)'$, while $\boldsymbol{\rho}_b = (0.3, 0.2, 0.1, -0.6, 0.2, -0.3, -0.1, 0.1, -0.2)$. The main difference between $\boldsymbol{\rho}_a$ and $\boldsymbol{\rho}_b$, is that $\boldsymbol{\rho}_a$ naturally produces a positive definite block correlation matrix $\boldsymbol{\Sigma}$, while $\boldsymbol{\rho}_b$ does not. The working correlation matrix used in the case of $\boldsymbol{\rho}_b$ is the nearest positive definite matrix of the original block matrix, calculated by the `nearPD` function in R package `Matrix`. This introduces some noise in the correlation structure, and thus the performance under $\boldsymbol{\rho}_b$ is expected to be at least slightly worse than that under $\boldsymbol{\rho}_a$. In each setting, 500 data sets are generated from the copSTM specified in (3.1) and (3.2).

Table 3.1 and Table 3.2 show Monte Carlo estimates of absolute values of bias and variance of $\hat{\boldsymbol{\theta}}$, as well as the true value of the parameters for the Poisson and Negative binomial marginals respectively. As expected, increasing the sample size by increasing T or n from 10 to 25 dramatically improves estimation performance for both marginals, reaching around a half of the bias and one third of the variance compared to the first case where both $T = n = 10$. In general, the case where $n = 25$ provides smaller bias and variance than the one where $T = 25$, which is most likely due to a larger increase in sample size if we consider sample size as n^2T. Also as anticipated, both estimated bias and variance for $\boldsymbol{\rho}_b$ are larger than those for $\boldsymbol{\rho}_a$, however the difference is quite mild when $n = 25$. More often than not, results are very similar with both marginal distributions, with the only exception for longer time series data (i.e. $T = 25$), where the negative binomial marginal case looks a bit unstable in estimating some correlation parameters.

Next, we assess the estimation of the standard errors of parameter $\boldsymbol{\theta}$, which we take as the square root of diagonal elements of $\boldsymbol{G}(\hat{\boldsymbol{\theta}})$ specified in Section 3.2.4. We show estimates for the coverage probability of 90%, 95% and 99% confidence intervals with increasing bootstrap sample sizes in Table 3.3 and Table 3.4 for Poisson and Negative binomial marginals respectively. All results shown in the table are averages taken over mean parameters ($\boldsymbol{\beta}$) and correlation parameters ($\boldsymbol{\rho}$). For Poisson marginals, coverage probabilities seem quite unstable when $B = 50$, especially for the second setup of corre-

	$n_{\mathscr{C}}$	$n_{\mathscr{L}}$	T	parameters	$\beta^{(1\|1)}$	$\beta^{(1\|2)}$	$\beta^{(1\|3)}$	$\beta^{(2\|1)}$	$\beta^{(2\|2)}$	$\beta^{(2\|3)}$	$\beta^{(3\|1)}$	$\beta^{(3\|2)}$	$\beta^{(3\|3)}$
				true value	1	0.5	-0.6	0.4	1	-0.2	0.2	0.1	1
$\widehat{\|\text{Bias}\|}$	ρ_a	10×10	10		11.45	12.52	20.07	5.36	5.51	9.28	2.07	1.81	2.61
		25×25	10		4.59	5.03	8.17	1.95	2.00	3.51	0.75	0.66	1.09
		10×10	25		6.57	8.25	5.85	3.28	4.25	3.19	1.57	1.43	1.12
	ρ_b	10×10	10		13.55	15.27	23.93	5.50	5.63	10.46	2.07	1.61	2.62
		25×25	10		5.66	6.38	9.78	2.09	2.12	3.96	0.79	0.70	1.28
		10×10	25		9.77	11.67	7.21	3.98	4.16	3.13	1.76	1.53	1.15
$\widehat{\text{Var}}$	ρ_a	10×10	10		20.12	24.64	63.85	4.32	4.75	13.17	0.66	0.50	1.04
		25×25	10		3.33	4.06	10.71	0.59	0.62	1.90	0.08	0.07	0.19
		10×10	25		6.68	10.39	5.34	1.65	2.90	1.61	0.39	0.32	0.19
	ρ_b	10×10	10		30.03	40.16	87.58	4.30	4.79	16.7	0.65	0.44	1.02
		25×25	10		4.67	6.22	13.58	0.70	0.69	2.43	0.10	0.08	0.23
		10×10	25		14.06	21.53	8.52	2.32	3.03	1.67	0.47	0.34	0.22
				parameters	$\rho^{(1)}$	$\rho^{(2)}$	$\rho^{(3)}$	$\rho_0^{(1,2)}$	$\rho_0^{(1,3)}$	$\rho_0^{(2,3)}$	$\rho_1^{(1,2)}$	$\rho_1^{(1,3)}$	$\rho_1^{(2,3)}$
	$n_{\mathscr{C}}$	$n_{\mathscr{L}}$	T	true value$_a$	0	0	0	-0.6	0.2	-0.3	0	0	0
				true value$_b$	0.3	0.2	0.1	-0.6	0.2	-0.3	-0.1	0.1	-0.2
$\widehat{\|\text{Bias}\|}$	ρ_a	10×10	10		2.56	1.67	1.38	2.27	3.51	2.54	1.53	1.32	1.03
		25×25	10		0.87	0.61	0.57	1.01	1.42	0.99	0.58	0.54	0.44
		10×10	25		1.43	1.02	0.86	1.61	2.16	1.66	0.91	0.84	0.65
	ρ_b	10×10	10		2.99	2.02	1.49	2.92	3.45	2.89	2.03	1.79	1.22
		25×25	10		1.37	0.95	0.64	1.51	1.29	0.99	1.00	0.80	0.55
		10×10	25		1.92	1.25	1.24	1.68	2.04	1.52	1.30	1.11	0.93
$\widehat{\text{Var}}$	ρ_a	10×10	10		0.94	0.41	0.30	0.89	1.88	1.00	0.38	0.27	0.17
		25×25	10		0.12	0.06	0.05	0.16	0.30	0.16	0.05	0.05	0.03
		10×10	25		0.34	0.16	0.11	0.41	0.71	0.43	0.14	0.11	0.07
	ρ_b	10×10	10		0.99	0.58	0.38	1.35	1.83	1.40	0.61	0.53	0.24
		25×25	10		0.21	0.11	0.06	0.29	0.27	0.15	0.14	0.10	0.04
		10×10	25		0.47	0.22	0.21	0.45	0.66	0.38	0.25	0.18	0.11

Table 3.1: Monte Carlo estimates for the absolute values of the bias ($\times10^{-2}$) and variance ($\times10^{-3}$) of the CL estimator with Poisson marginals under different setups of $n_{\mathscr{C}}, n_{\mathscr{L}}$ and T.

lation parameters ρ_b, with an overestimation for standard errors of regression coefficients and underestimation for those of correlation parameters. When $B = 100$, all coverage probabilities become stable and very close to the nominal level. However, we do not observe much improvement as B increase from 100 to 500. Most results look very similar with both marginals. Generally, all results with ρ_a performs better than those with ρ_b, and $T = 25$ with ρ_b seems to be the most difficult case, especially for the Negative binomial marginals. Coverage probabilities show an underestimation of standard errors with the misspecification of the correlation matrix, although estimation is improving with the increase of B, the coverage is still below the nominal level even with the largest $B = 500$.

Apart from the coverage probabilities, we include at the end of both tables the computational time required, which grows in proportion of the size of B as anticipated. We observe that for $n = 25$, the running time needed is about 15 times than $n = 10$, this

	$n_{\mathscr{C}}$	$n_{\mathscr{L}}$	T	parameters	$\beta^{(1\|1)}$	$\beta^{(1\|2)}$	$\beta^{(1\|3)}$	$\beta^{(2\|1)}$	$\beta^{(2\|2)}$	$\beta^{(2\|3)}$	$\beta^{(3\|1)}$	$\beta^{(3\|2)}$	$\beta^{(3\|3)}$
				true value	1	0.5	-0.6	0.4	1	-0.2	0.2	0.1	1
$\widehat{\|\text{Bias}\|}$	ρ_a	10×10	10		11.88	13.25	19.98	5.06	5.41	9.82	1.81	1.67	2.78
		25×25	10		4.92	5.63	8.29	2.16	2.25	3.58	0.86	0.76	1.17
		10×10	25		5.93	7.98	5.58	3.7	4.56	3.21	1.70	1.59	1.19
	ρ_b	10×10	10		15.12	17.28	25.09	5.99	5.67	10.75	2.14	1.74	3.11
		25×25	10		5.86	6.45	9.90	2.40	2.57	4.39	0.77	0.71	1.19
		10×10	25		9.50	11.83	9.09	4.43	5.07	3.82	2.11	2.32	1.75
$\widehat{\text{Var}}$	ρ_a	10×10	10		24.43	30.42	63.90	3.77	4.31	14.37	0.50	0.44	1.21
		25×25	10		3.87	4.86	10.33	0.70	0.77	1.85	0.10	0.08	0.21
		10×10	25		5.58	9.62	4.99	2.19	3.05	1.57	0.48	0.53	0.25
	ρ_b	10×10	10		35.26	44.71	96.63	5.36	4.89	18.25	0.67	0.45	1.44
		25×25	10		5.34	6.43	14.81	0.91	1.02	3.18	0.09	0.08	0.23
		10×10	25		13.76	20.96	13.79	3.55	3.78	2.36	0.67	1.35	0.66
				parameters	$\rho^{(1)}$	$\rho^{(2)}$	$\rho^{(3)}$	$\rho_0^{(1,2)}$	$\rho_0^{(1,3)}$	$\rho_0^{(2,3)}$	$\rho_1^{(1,2)}$	$\rho_1^{(1,3)}$	$\rho_1^{(2,3)}$
	$n_{\mathscr{C}}$	$n_{\mathscr{L}}$	T	true value$_a$	0	0	0	-0.6	0.2	-0.3	0	0	0
				true value$_b$	0.3	0.2	0.1	-0.6	0.2	-0.3	-0.1	0.1	-0.2
$\widehat{\|\text{Bias}\|}$	ρ_a	10×10	10		2.57	1.58	1.47	2.62	3.31	2.33	1.65	1.33	1.03
		25×25	10		0.88	0.67	0.55	0.99	1.16	0.96	0.61	0.51	0.40
		10×10	25		1.56	1.31	1.74	1.90	2.06	2.03	1.02	0.86	1.05
	ρ_b	10×10	10		3.64	2.44	1.73	3.18	3.77	2.63	2.52	2.13	1.48
		25×25	10		1.21	0.95	0.67	1.46	1.41	1.11	0.89	0.72	0.56
		10×10	25		2.10	2.27	7.57	3.23	3.15	3.31	2.08	2.14	2.43
$\widehat{\text{Var}}$	ρ_a	10×10	10		0.89	0.35	0.34	1.00	1.72	0.84	0.38	0.27	0.17
		25×25	10		0.11	0.07	0.05	0.16	0.22	0.15	0.06	0.04	0.03
		10×10	25		0.36	0.41	5.97	0.82	0.76	1.00	0.19	0.25	0.83
	ρ_b	10×10	10		1.64	0.75	0.43	1.70	2.21	1.15	0.93	0.65	0.35
		25×25	10		0.15	0.11	0.07	0.24	0.29	0.20	0.12	0.07	0.05
		10×10	25		0.53	1.36	39.83	3.47	1.79	3.42	0.78	1.07	2.62

Table 3.2: Monte Carlo estimates for the absolute values of the bias ($\times10^{-2}$) and variance ($\times10^{-3}$) of the CL estimator with Negative binomial marginals under different setups of $n_{\mathscr{C}}, n_{\mathscr{L}}$ and T.

because the number of component pairwise likelihoods grows roughly at the order of $n_{\mathscr{L}}^2$.

Finally, we carry out the adjusted composite likelihood ratio tests described in Section 3.5. Specifically, for each simulated dataset, we carry out both tests as described in (3.7) and (3.8) with significance level 5% and $\hat{\boldsymbol{\theta}}$ taken as the parameter estimates from the dataset. In Table 3.5, we report the percentage of rejections among the 100 simulated datasets for both marginals. While the Geys et al. (1999) likelihood ratio test (W_1) tends to produce more type 1 errors than anticipated, the rejection rate of the test proposed by Satterthwaite (1946) (W_2) is very close to the significance level in all settings.

				β			ρ			
B	$n_{\mathscr{C}}$	$n_{\mathscr{L}}$	T	90%	95%	99%	90%	95%	99%	Time (h:m.s)
50	ρ_a	10×10	10	89.2	93.7	98.6	89.4	93.1	97.9	00:01.16
		25×25	10	92.2	96.2	99.2	90.0	94.7	98.8	00:15.12
		10×10	25	92.5	96.0	98.8	91.3	94.9	99.0	00:03.00
	ρ_b	10×10	10	93.2	97.0	99.7	92.3	96.1	99.6	00:01.14
		25×25	10	94.2	97.4	99.0	88.6	93.8	98.5	00:15.23
		10×10	25	95.0	97.3	99.1	87.3	92.6	98.0	00:03.03
100	ρ_a	10×10	10	91.6	95.5	98.1	90.8	95.2	98.8	00:02.24
		25×25	10	91.9	96.0	98.8	91.0	96.7	99.2	00:29.27
		10×10	25	92.9	97.4	99.2	89.9	94.8	98.9	00:05.38
	ρ_b	10×10	10	90.2	94.3	98.2	89.9	94.8	99.4	00:02.28
		25×25	10	89.9	94.4	98.7	89.8	95.2	99.1	00:29.37
		10×10	25	92.8	96.6	98.9	91.2	96.2	99.4	00:05.54
200	ρ_a	10×10	10	90.7	96.2	98.9	90.1	94.9	98.9	00:04.40
		25×25	10	92.0	95.2	98.8	91.2	95.6	99.4	01:10.00
		10×10	25	92.7	96.8	99.2	92.1	96.6	98.7	00:11.33
	ρ_b	10×10	10	91.4	94.4	99.0	91.7	95.7	98.9	00:04.47
		25×25	10	91.2	95.3	99.6	88.9	94.7	99.2	01:21.00
		10×10	25	91.8	96.7	99.6	90.8	96.3	99.2	00:11.51
500	ρ_a	10×10	10	90.2	95.6	99.3	89.8	95.0	98.6	00:11.30
		25×25	10	91.4	95.2	98.9	90.3	95.1	99.2	02:38.35
		10×10	25	92.9	96.4	99.2	90.2	95.6	99.6	00:28.01
	ρ_b	10×10	10	90.3	94.9	99.0	90.3	94.4	99.0	00:11.53
		25×25	10	89.8	94.4	97.7	90.3	95.6	99.4	02:43.39
		10×10	25	91.4	95.6	98.8	89.2	96.3	99.0	00:28.36

Table 3.3: Monte Carlo estimates for the coverage probability of 90%, 95% and 99% confidence intervals for Poisson marginals, where the estimate of the standard errors of $\boldsymbol{\theta}$ is given in Section 3.2.4.All coverages shown in the table are averages taken over mean parameters (β), and correlation parameters (ρ).

3.4 Real data analysis

In this section, we reanalyze the cancer cell growth data from the co-culture experiment with fibroblasts analyzed in Chapter 2. Recall that the datasets analyzed consist of counts of cell types (different cancer cell populations expressing different fluorescent proteins, and non-fluorescent fibroblasts) from 9 subsequent images taken at 8-hour frequency over a period of 3 days. In the experiment, fibroblasts (F) are non-fluorescent whereas cancer cells fluoresce either in the red (R) or green (G) channels due to the experimental expression of mCherry or GFP proteins respectively. Each image was subsequently tiled using

				β			ρ			
B	$n_{\mathscr{C}}$	$n_{\mathscr{L}}$	T	90%	95%	99%	90%	95%	99%	Time (h:m.s)
50	ρ_a	10×10	10	92.5	96.0	98.5	89.7	93.6	98.3	00:02.05
		25×25	10	93.4	97.3	99.2	91.4	96.0	98.9	00:16.22
		10×10	25	94.3	97.5	99.3	88.5	93.2	97.7	00:04.51
	ρ_b	10×10	10	90.9	95.1	99.3	88.0	92.5	97.9	00:02.54
		25×25	10	90.3	94.4	99.0	88.6	94.1	98.4	00:16.48
		10×10	25	90.8	94.0	97.4	80.8	86.6	93.9	00:05.04
100	ρ_a	10×10	10	91.2	94.7	98.6	89.8	94.6	98.4	00:06.57
		25×25	10	92.4	96.0	99.0	91.1	95.4	99.3	00:32.42
		10×10	25	94.1	97.2	99.2	89.3	93.9	98.3	00:15.23
	ρ_b	10×10	10	91.2	95.8	98.9	88.2	93.4	98.6	00:06.08
		25×25	10	89.5	94.3	98.4	92.0	97.0	99.4	00:33.32
		10×10	25	91.3	94.8	98.3	82.6	87.8	95.3	00:19.18
200	ρ_a	10×10	10	91.1	95.1	98.6	89.2	94.4	98.3	00:04.58
		25×25	10	91.4	95.3	99.1	90.9	95.5	99.0	01:03.52
		10×10	25	93.6	96.8	99.1	88.3	93.9	98.6	00:34.45
	ρ_b	10×10	10	91.2	95.1	98.4	88.5	93.5	98.5	00:05.07
		25×25	10	92.6	96.4	99.3	89.8	95.0	98.3	01:07.42
		10×10	25	91.5	95.0	98.5	84.9	89.1	95.5	00:30.56
500	ρ_a	10×10	10	90.7	94.4	98.1	90.9	95.3	99.1	00:24.42
		25×25	10	91.4	95.3	99.1	90.7	96.0	99.3	02:21.08
		10×10	25	93.4	96.3	98.9	90.3	95.3	98.9	01:05.20
	ρ_b	10×10	10	90.7	94.8	99.0	90.0	95.5	99.3	00:25.06
		25×25	10	90.2	95.1	98.5	90.1	95.4	99.5	02:51.50
		10×10	25	91.2	95.3	99.2	85.2	92.8	96.7	01:24.05

Table 3.4: Monte Carlo estimates for the coverage probability of 90%, 95% and 99% confidence intervals for Negative binomial marginals, where the estimate of the standard error of $\boldsymbol{\theta}$ is given in Section 3.2.4. All coverages shown in the table are averages taken over mean parameters (β), and correlation parameters (ρ).

a 25×25 regular grid.

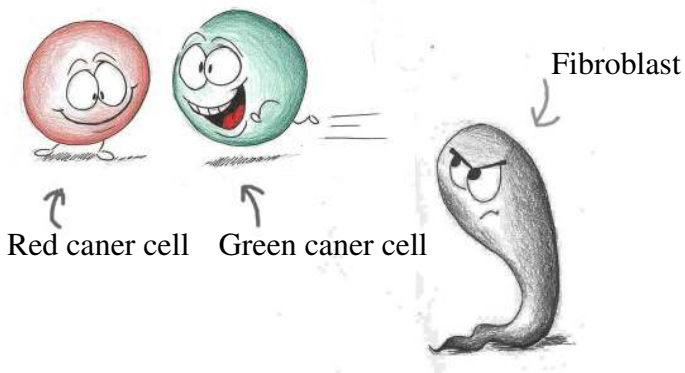
Table 3.6 shows estimated parameters of both models, with 95% bootstrap confidence intervals in parenthesis. Estimates significantly different from zero are shown in bold numbers. As anticipated, the regression parameter estimates in both models are consistent with each other, in that: both models show that impacts within each cell type ($\hat{\beta}^{(c|c)}, c = R,G,F$) are significant, the effects $\hat{\beta}^{(c|c)}$ for cancer cells are larger than those for the slower growing fibroblasts, the green and red cancer cells have positive impact on each other, while cancer cells and fibroblasts have no impact on each other. Also note that impacts related to the red and green cancer cells are symmetric, that is $\hat{\beta}^{(R|c)}$ is similar to

			Poisson		Negative Binomial	
	$n_{\mathscr{L}}$	T	$W_1(\%)$	$W_2(\%)$	$W_1(\%)$	$W_2(\%)$
ρ_a	10×10	10	11	6	13	5
	25×25	10	10	4	12	4
	10×10	25	11	6	14	6
ρ_b	10×10	10	11	4	14	6
	25×25	10	10	5	12	5
	10×10	25	12	6	15	7

Table 3.5: Percentage of rejection of goodness-of-fit test (at significance level of 5%) among 100 simulated data sets with Poisson and Negative binomial marginals. Goodness-of-fit tests with adjusted composite likelihood ratio tests with test statistics described in (3.7) and (3.8).

$\hat{\beta}^{(G|c)}$, and $\hat{\beta}^{(c|R)}$ is similar to $\hat{\beta}^{(c|G)}$ for all $c \in \{R, G, F\}$, which aligns with the fact that the red and green cancer cells are biologically identical except for the fluorescent protein they express.

Estimated correlation parameters from the copSTM model are displayed as two correlation matrices, correspond to $\boldsymbol{R}_0$ and $\boldsymbol{R}_1$ specified in Section 3.2.2. Recall that $\widehat{\rho_0}^{(c,c')}$ is the between group correlation within the same tile and $\widehat{\rho_1}^{(c,c')}$ denotes the between and within group correlation in neighbouring (but not the same) tiles. Diagonal elements in the second matrix, $\widehat{\rho_1}^{(c,c)}$ is the same $\widehat{\rho}_1^{(c)}$ in Section 3.2.2 and 3.3. Both correlation matrices are shown as upper triangular matrices, omitting repetitive symmetrical entries. Significant positive $\widehat{\rho_0}^{(R,G)}$ indicates that the cancer cells of the two colours not only promote the growth of each other, but also likely to tend to stay together, i.e. the more green cancer cells in one tile, the more red cancer cells are likely to be in the same tile, and vice versa. On the other hand, both types of cancer cells have negative local correlations with fibroblasts ($\widehat{\rho_0}^{(G,F)}$ and $\widehat{\rho_0}^{(R,F)}$). It seems to suggest although cancer cells are likely to stay close to each other, they tend to keep some distance from fibroblasts.



The second goal is to assess the goodness-of-fit and prediction performance of the estimated model. To illustrate the goodness-of-fit, we generate cell counts of each time point,

	Model 1		
$c=$	G	R	F
$\widehat{\beta}_0^{(c)}$	**-0.99 (-1.19, -0.79)**	**-0.50 (-0.70, -0.30)**	**-0.26 (-0.45, -0.06)**
$\widehat{\beta}^{(G\|c)}$	**1.23 (1.10, 1.35)**	**0.34 (0.21, 0.48)**	0.12 (-0.03, 0.27)
$\widehat{\beta}^{(R\|c)}$	**0.28 (0.17, 0.38)**	**1.09 (0.96, 1.21)**	0.02 (-0.09, 0.13)
$\widehat{\beta}^{(F\|c)}$	0.10 (-0.01, 0.21)	0.02 (-0.07, 0.12)	**0.92 (0.81, 1.03)**
	copMSTM model		
$c=$	G	R	F
$\widehat{\beta}_0^{(c)}$	**-1.06 (-1.18, -1.94)**	**-0.60 (-0.69, -0.51)**	**-0.42 (-0.52, -0.32)**
$\widehat{\beta}^{(G\|c)}$	**1.45 (1.35, 1.56)**	**0.22 (0.14, 0.30)**	0.06 (-0.03, 0.15)
$\widehat{\beta}^{(R\|c)}$	**0.31 (0.22, 0.41)**	**1.26 (1.18, 1.34)**	0.05 (-0.04, 0.14)
$\widehat{\beta}^{(F\|c)}$	0.08 (-0.02, 0.18)	0.01 (-0.08, 0.09)	**1.08 (0.99, 1.17)**
$\widehat{\rho}_0^{(G,c)}$	1	**0.05(0.02, 0.08)**	**-0.03 (-0.05, -0.01)**
$\widehat{\rho}_0^{(R,c)}$	–	1	**-0.02 (-0.03, -0.01)**
$\widehat{\rho}_0^{(F,c)}$	–	–	1
$\widehat{\rho}_1^{(G,c)}$	0.01 (-0.03, 0.01)	0.00(-0.00, 0.02)	-0.01 (-0.02, 0.00)
$\widehat{\rho}_1^{(R,c)}$	–	-0.00 (-0.02, 0.01)	-0.02(-0.03, -0.00)
$\widehat{\rho}_1^{(F,c)}$	–	–	0.01 (-0.00, 0.02)

Table 3.6: Estimated parameters of the model in Qiao et al. (2018) (denoted as Model 1) and the copSTM model based on the cancer cell growth data. Bootstrap 95% confidence intervals based on 100 bootstrap samples are given in parenthesis.

$y_t = (y_{1,t}, \ldots, y_{n_{\mathscr{L}},t}), t = 1, \ldots, 8$ from a multivariate Poisson distribution, with marginal expectation $\mu_{i,t}^{(c)}$ described in (3.1) with $\boldsymbol{\beta}^{(c)}$ and $H(y_{i,t-1}^{(c)})$ specified earlier in this section, and correlation matrix as the block matrix described in Section 3.2.2, with regression and correlation parameters taken as the estimated values shown in Table 3.6 and y_{t-1} as observations from the previous time point. For each time point, we compare the generated total cell count of each type (green cancer cells (G), red cancer cells (R) and fibroblasts (F)) across the image with the observed count. In Figure 3.2, The observed cell count growth curves are shown as solid lines, while the generated counts correspond to dashed lines. The solid and dashed curves for all cell types are close, suggesting that the model fits the data reasonably well. As anticipated, the overall growth rate for the red and green cancer cells are similar, and sensibly larger than the growth rate for fibroblasts.

Next, we carry out a one-step-ahead forecasting using parameters estimated from a moving window of five time points. In Figure 3.3, we show quantiles of observed cell counts against predicted counts for each tile. The upper and lower 95% confidence bounds are computed non-parametrically by taking $\hat{F}_1^{-1}\big(\hat{F}_0(y_t^{(c)}) - q_\alpha\big)$ and $\hat{F}_1^{-1}\big(\hat{F}_0(y_t^{(c)}) + q_\alpha\big)$, where $\hat{F}_0$ and $\hat{F}_1$ are the empirical distributions of the observations and predictions at time t respectively and q_α is the $(1-\alpha)$ quantile of the Kolmogorov-Smirnov statistic

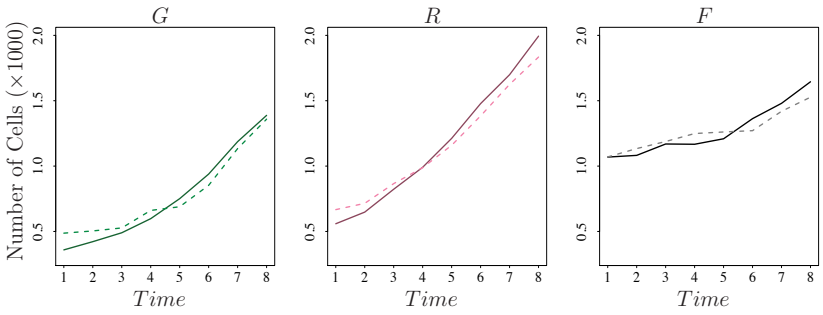


Figure 3.2: Goodness-of-fit of the estimated models. Observed (solid) and predicted (dashed) number of green cancer cells (G), red cancer cells (R) cancer cells and fibroblasts (F) for the entire image at time points $t = 1, \ldots, 8$.

$\sup_y |\hat{F}_0(y) - \hat{F}_1(y)|$ (Nair, 1982). The identity line falls within the confidence bands in each plot, indicating a satisfactory prediction performance.

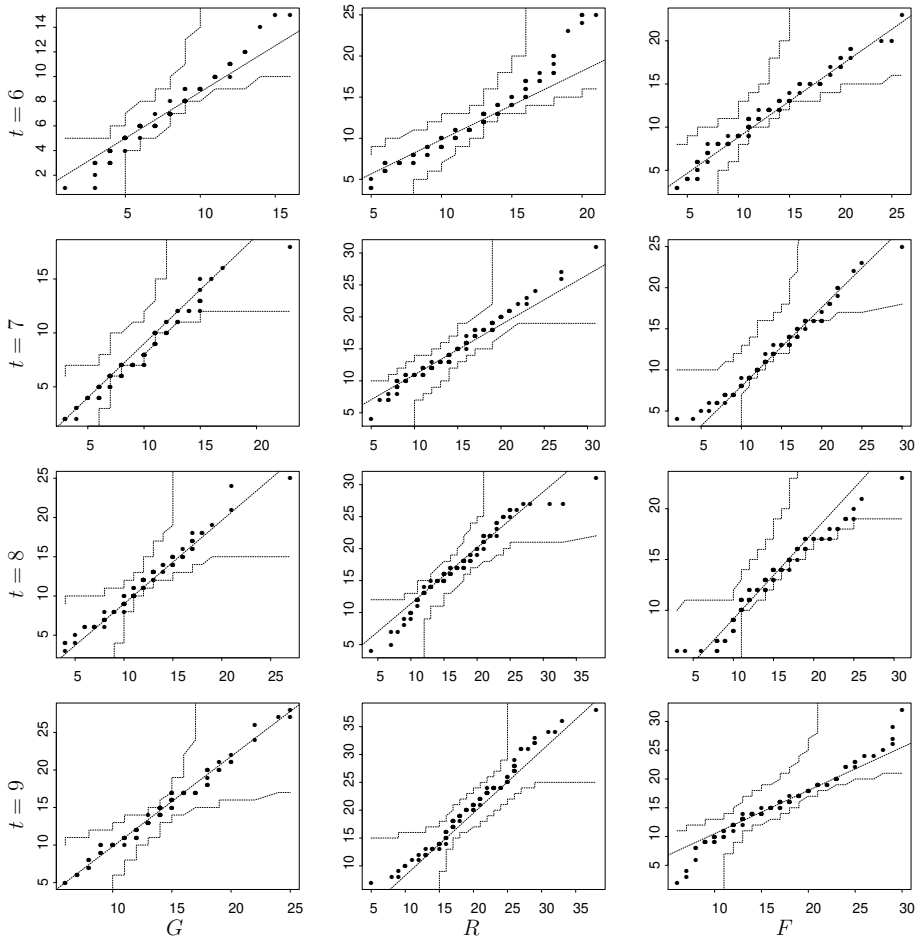


Figure 3.3: QQ-plots for cell growth, comparing observed (horizontal axis) and one-time ahead predicted (vertical axis) cell counts per tile on the entire image at times $t = 6, 7, 8$ for GFP cancer cells (G), mCherry cancer cells (R) and fibroblasts (F). One-time ahead predictions are based on the model fitted using a moving window of five time points.

3.5 Discussions

In the analysis of spatio-temporal data, Bayesian methods are predominantly used owing to the numerical limitations of the traditional likelihood methods. The proposed Gaussian copula regression model (copSTM) and pairwise CL inference offer a competitive alternative for analyzing correlated count data.

Specifically, we consider count data set observed on a $n \times n$ regular lattice for T consecutive time periods, yielding an array of $n \times n \times T$ spatial-temporal observations. Temporal parameters are captured as regression coefficients in a GARMA(1, 0) marginal model, while dependence between observed responses at the same time point are modelled through the Gaussian copula. The model specification allows for straightforward interpretation of marginal regression coefficients and great flexibility in specification of correlation structures. The correlation matrix accommodates for both spatial and between group correlations, and all correlations are allowed to be positive or negative.

On the modelling side, although we particularly focus on count lattice data in this chapter, the proposed copSTM model is so flexible that it can easily be extended to fit other types of spatio-temporal data. First, apart from Poisson and Negative binomial marginals considered in this chapter, the marginal distribution could be any exponential family distributions. Second, the marginal condition mean can be generalized to depend on previous observations with time lag greater than one as well as on its own previous values. This makes a GARMA(p, q) model where $p, q \geq 1$, in which case the score function needs to be computed recursively. Besides, external explanatory variables may also be allowed. Thirdly, the correlation matrix can be extended to other dependence structure designs. For example, since the straightforward correlation parametrization of our model has positive definite problems, it may be helpful to consider the correlation matrix proposed by Tang et al. (2019), which is parametrized in hyperspherical coordinates, with no constraint on parameters and guarantees to be nonnegative definite. Although in such design, the correlation parameters are less interpretable, it is a possible future research direction. Other dependence structures worth considering include the popular isotropic correlation functions like the Matérn, the power exponential and the spherical families adopted by Han and De Oliveira (2018) and Bai et al. (2014). By including these Euclidean distances-based correlation functions, the model can be extended to handle geographical data.

On the inference side, a clear advantage of the proposed pairwise CL method is the computational feasibility. Besides, we derive the closed forms of the score function and Hessian matrix, making the model fitting fairly fast. It is shown in our simulation studies that the CL estimator performs well for both Poisson and Negative binomial data, and that a certain level of misspecification can be tolerated. Our numerical experiences also suggest that the parametric bootstrap works well in estimating standard errors, apart

from being slightly computationally costly. One possible future extension is the further improvement of statistical efficiency (maybe at some cost of computational efficiency), which can be achieved by adding some properly chosen weights to the CL estimating functions. For example, the joint composite estimating function (JCEF) proposed by Bai et al. (2014) for the analysis of spatial data clustered in terms of locations, where pairwise score functions are grouped into between and within cluster pairs: $u_B(\boldsymbol{\theta})$ and $u_W(\boldsymbol{\theta})$ respectively. Then the JCEF is defined as $\boldsymbol{\Gamma}(\boldsymbol{\theta})'\boldsymbol{K}(\boldsymbol{\theta})^{-1}\boldsymbol{\Gamma}(\boldsymbol{\theta})$, where $\Gamma(\boldsymbol{\theta}) = (u_B(\boldsymbol{\theta}), u_W(\boldsymbol{\theta}))$ and $\boldsymbol{K}(\boldsymbol{\theta}) = \mathrm{Var}(\Gamma(\boldsymbol{\theta}))$ can be estimated with the parametric bootstrap method.

Last but not the least, the number of parameters in the proposed model grows quadratically with the number of groups $n_{\mathscr{C}}$ at each location (specifically $2n_{\mathscr{C}}^2 + n_{\mathscr{C}}$), therefore it would be useful to perform model selection in order to control model complexity and avoid an over-parametrized model. Although our likelihood framework allows us to facilitate a penalized likelihood function like LASSO or SCAD, these methods encounter convergence problems with the presence of correlations. Therefore, we plan to explore a model selection method first introduced by Qian and Field (2002), that employs the Gibbs sampler and is consistent and efficient even with a large model space.

3.6 Appendix

1. Score vector

The score vector is

$$u(\boldsymbol{\theta};\boldsymbol{y}) = \frac{\partial cl(\boldsymbol{\theta};\boldsymbol{y})}{\partial \boldsymbol{\theta}} = \sum_{t=1}^{T}\sum_{i_1\sim i_2}\sum_{c_1=1}^{n_{\mathscr{C}}}\sum_{c_2=1}^{n_{\mathscr{C}}} \frac{\partial cl(\boldsymbol{\theta};\, y_{i_1,t}^{(c_1)}, y_{i_2,t}^{(c_2)})}{\partial \boldsymbol{\theta}},$$

where

$$cl(\boldsymbol{\theta};\, y_{i_1,t}^{(c_1)}, y_{i_2,t}^{(c_2)}) = \log\left[CL(\boldsymbol{\theta};\, y_{i_1,t}^{(c_1)}, y_{i_2,t}^{(c_2)})\right] = \log\left[\int_{a_{i_1,t}^{(c_1)}}^{b_{i_1,t}^{(c_1)}}\int_{a_{i_2,t}^{(c_2)}}^{b_{i_2,t}^{(c_2)}} \phi_\rho(z_1,z_2)dz_1dz_2\right]$$

is the component CL of one pair.

1.1 Score vector of mean parameters

The first $2n_{\mathscr{C}}$ entries of the score vector are the derivatives of the CL function with respect to mean parameters. Partial derivatives with respect to $\alpha^{(c)}$ and $\beta^{(c)}$ have the same form, so without loss of generosity, consider only the derivative with respect to $\alpha^{(c_1)}$.

For simplicity, let $a_1 = a_{i_1,t}^{(c_1)}, b_1 = b_{i_1,t}^{(c_1)}, a_2 = a_{i_2t}^{(c_2)}, b_2 = b_{i_2,t}^{(c_2)}$.

- If $c_1 \neq c_2$, then

$$\frac{\partial cl(\boldsymbol{\theta};\, y_{i_1,t}^{(c_1)}, y_{i_2,t}^{(c_2)})}{\partial \alpha^{(c_1)}} = \frac{\frac{\partial b_1}{\partial \alpha^{(c_1)}} f(a_2, b_2, b_1, \rho) - \frac{\partial a_1}{\partial \alpha^{(c_1)}} f(a_2, b_2, a_1, \rho)}{CL(\boldsymbol{\theta};\, y_{i_1,t}^{(c_1)}, y_{i_2,t}^{(c_2)})}, \tag{3.9}$$

  where $\rho = \rho_0^{(c_1,c_2)}$ if $i_1 = i_2$ and $\rho = \rho_1^{(c_1,c_2)}$ otherwise.

- If $c_1 = c_2 = c$, then

$$\frac{\partial cl(\boldsymbol{\theta};\, y_{i_1,t}^{(c)}, y_{i_2,t}^{(c)})}{\partial \alpha^{(c)}} = \frac{1}{CL(\boldsymbol{\theta};\, y_{i_1,t}^{(c)}, y_{i_2,t}^{(c)})} \left[\frac{\partial b_1}{\partial \alpha^{(c)}} f(a_2, b_2, b_1, \rho) - \frac{\partial a_1}{\partial \alpha^{(c)}} f(a_2, b_2, a_1, \rho) + \frac{\partial b_2}{\partial \alpha^{(c)}} f(a_1, b_1, b_2, \rho) - \frac{\partial a_2}{\partial \alpha^{(c)}} f(a_1, b_1, a_2, \rho) \right], \tag{3.10}$$

  where $\rho = \rho_1^{(c)}$ and

$$f(a,b,c,\rho) = \int_a^b \phi_\rho(c,z)dz = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{c^2}{2}\right) \int_a^b \frac{1}{\sqrt{2\pi}\sqrt{1-\rho^2}} \exp\left(-\frac{1}{2}\frac{(z-\rho c)^2}{1-\rho^2}\right) dz, \tag{3.11}$$

  in which the second term is the probability of $Z' \sim N(\rho c, 1-\rho^2)$ between a and b.

1.2 Score vector of correlation parameters

First, note that among all $n_{\mathscr{C}}^2$ correlation parameters, only one appears in each component likelihood, thus

$$\begin{aligned}
&\frac{\partial cl(\boldsymbol{\theta};\, y_{i_1,t}^{(c_1)}, y_{i_2,t}^{(c_2)})}{\partial \rho_1^{(c)}} \neq 0 \text{ only when } c_1 = c_2 = c \text{ and } i_1 \neq i_2, \\
&\frac{\partial cl(\boldsymbol{\theta};\, y_{i_1,t}^{(c_1)}, y_{i_2,t}^{(c_2)})}{\partial \rho_0^{(c_1,c_2)}} \neq 0 \text{ only when } i_1 = i_2, \\
&\frac{\partial cl(\boldsymbol{\theta};\, y_{i_1,t}^{(c_1)}, y_{i_2,t}^{(c_2)})}{\partial \rho_1^{(c_1,c_2)}} \neq 0 \text{ only when } i_1 \neq i_2.
\end{aligned}$$

The general form of the derivative of the component likelihood with respect to correlation parameters is

$$\frac{\partial cl(\boldsymbol{\theta};\, y_{i_1,t}^{(c_1)}, y_{i_2,t}^{(c_2)})}{\partial \rho} = \frac{\int_{a_1}^{b_1} \int_{a_2}^{b_2} \frac{\partial \phi_\rho(z_1,z_2)}{\partial \rho} dz_1 dz_2}{CL(\boldsymbol{\theta};\, y_{i_1,t}^{(c_1)}, y_{i_2,t}^{(c_2)})}, \tag{3.12}$$

where $\rho \in \{\rho_1^{(c)}, \rho_0^{(c_1,c_2)}, \rho_1^{(c_1,c_2)}\}$, $c, c_1, c_2 \in \{1, \ldots, n_{\mathscr{C}}\}$. The numerator of (3.12)

$$\begin{aligned}
&\int_{a_1}^{b_1}\int_{a_2}^{b_2} \frac{\partial \phi_\rho(z_1,z_2)}{\partial \rho} dz_1 dz_2 = \int_{a_1}^{b_1}\int_{a_2}^{b_2} \frac{\partial}{\partial \rho}\left[\frac{1}{2\pi|\boldsymbol{\Sigma}|^{1/2}} \exp\left(-\frac{1}{2}\boldsymbol{z}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{z}\right)\right] dz_1 dz_2, \\
&\text{where} \boldsymbol{z} = (z_1,z_2)^T, \boldsymbol{\Sigma} = \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}, \\
=&\int_{a_1}^{b_1}\int_{a_2}^{b_2} -\frac{1}{4\pi|\boldsymbol{\Sigma}|^{3/2}}|\boldsymbol{\Sigma}| \text{tr}\left(\boldsymbol{\Sigma}^{-1}\frac{d\boldsymbol{\Sigma}}{d\rho}\right) \exp\left(-\frac{1}{2}\boldsymbol{z}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{z}\right) dz_1 dz_2 \\
&+\int_{a_1}^{b_1}\int_{a_2}^{b_2} -\frac{1}{4\pi|\boldsymbol{\Sigma}|^{1/2}}\left(\boldsymbol{z}^T\boldsymbol{\Sigma}^{-1}\frac{d\boldsymbol{\Sigma}}{d\rho}\boldsymbol{\Sigma}^{-1}\boldsymbol{z}\right) dz_1 dz_2 \qquad (3.13)
\end{aligned}$$

$$\begin{aligned}
\text{The first term in (3.13)} &= -\frac{1}{2}\text{tr}\left(\boldsymbol{\Sigma}^{-1}\frac{d\boldsymbol{\Sigma}}{d\rho}\right)\int_{a_1}^{b_1}\int_{a_2}^{b_2}\frac{1}{2\pi|\boldsymbol{\Sigma}|^{1/2}}\exp\left(-\frac{1}{2}\boldsymbol{z}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{z}\right)dz_1dz_2 \\
&= \frac{\rho}{(1-\rho^2)} CL(\boldsymbol{\theta}; y_{i_1,t}^{(c_1)}, y_{i_2,t}^{(c_2)})
\end{aligned}$$

$$\begin{aligned}
\text{The second term in (3.13)} &= \frac{\rho}{(1-\rho^2)^2}\int_{a_1}^{b_1}\int_{a_2}^{b_2} z_1^2\phi_\rho(z_1,z_2)dz_1dz_2 \\
&-\frac{1+\rho^2}{(1-\rho^2)^2}\int_{a_1}^{b_1}\int_{a_2}^{b_2} z_1z_2\phi_\rho(z_1,z_2)dz_1dz_2 \\
&+\frac{\rho}{(1-\rho^2)^2}\int_{a_1}^{b_1}\int_{a_2}^{b_2} z_2^2\phi_\rho(z_1,z_2)dz_1dz_2
\end{aligned}$$

Let $A = \int_{a_1}^{b_1}\int_{a_2}^{b_2} z_1^2\phi_\rho(z_1,z_2)dz_1dz_2$, $B = \int_{a_1}^{b_1}\int_{a_2}^{b_2} z_1z_2\phi_\rho(z_1,z_2)dz_1dz_2$ and $C = \int_{a_1}^{b_1}\int_{a_2}^{b_2} z_2^2\phi_\rho(z_1,z_2)dz_1dz_2$. Then (3.13) can be simplified as

$$\int_{a_1}^{b_1}\int_{a_2}^{b_2}\frac{\partial\phi_\rho(z_1,z_2)}{\partial\rho}dz_1dz_2 = \frac{\rho}{(1-\rho^2)}CL(\boldsymbol{\theta}; y_{i_1,t}^{(c_1)}, y_{i_2,t}^{(c_2)}) + \frac{\rho}{(1-\rho^2)^2}A - \frac{1+\rho^2}{(1-\rho^2)^2}B + \frac{\rho}{(1-\rho^2)^2}C. \qquad (3.14)$$

Next, we derive A, B and C.

First, we start from A,

$$\begin{aligned}
A =& \frac{1}{2\pi\sqrt{1-\rho^2}}\int_{a_1}^{b_1}\int_{a_2}^{b_2} z_1^2\exp\left(-\frac{1}{2}\frac{z_1^2-2\rho z_1z_2+z_2^2}{1-\rho^2}\right)dz_2dz_1 \\
=& \frac{1}{2\pi\sqrt{1-\rho^2}}\int_{a_1}^{b_1}\int_{a_2}^{b_2}(z_1^2-\rho z_1z_2)\exp\left(-\frac{1}{2}\frac{z_1^2-2\rho z_1z_2+z_2^2}{1-\rho^2}\right)dz_2dz_1 \\
&+\frac{1}{2\pi\sqrt{1-\rho^2}}\int_{a_1}^{b_1}\int_{a_2}^{b_2}\rho z_1z_2\exp\left(-\frac{1}{2}\frac{z_1^2-2\rho z_1z_2+z_2^2}{1-\rho^2}\right)dz_2dz_1 \\
=& \frac{1}{2\pi\sqrt{1-\rho^2}}\int_{a_2}^{b_2}\int_{a_1}^{b_1} -(1-\rho^2)z_1d\exp\left(-\frac{1}{2}\frac{z_1^2-2\rho z_1z_2+z_2^2}{1-\rho^2}\right)dz_2 + \rho B \qquad (3.15)
\end{aligned}$$

$$\begin{aligned}\text{The first term in (3.15)} =& -\frac{\sqrt{1-\rho^2}}{2\pi}\int_{a_2}^{b_2}\left[z_1\exp\left(-\frac{1}{2}\frac{z_1^2-2\rho z_1z_2+z_2^2}{1-\rho^2}\right)\Big|_{a_1}^{b_1}\right]dz_2\\ &+(1-\rho^2)\int_{a_2}^{b_2}\int_{a_1}^{b_1}\frac{1}{2\pi\sqrt{1-\rho^2}}\exp\left(-\frac{1}{2}\frac{z_1^2-2\rho z_1z_2+z_2^2}{1-\rho^2}\right)dz_1dz_2\\ =&(1-\rho^2)\left[b_1f(a_2,b_2,b_1,\rho)-a_1f(a_2,b_2,a_1,\rho)+CL(\boldsymbol{\theta};y_{i_1,t}^{(c_1)},y_{i_2,t}^{(c_2)})\right],\end{aligned}$$

where $f(a,b,c,\rho)$ is defined in (3.11). Thus, the first equation between A and B is obtained by substituting the above expression into (3.15):

$$A=(1-\rho^2)\left[b_1f(a_2,b_2,b_1,\rho)-a_1f(a_2,b_2,a_1,\rho)+CL(\boldsymbol{\theta};y_{i_1,t}^{(c_1)},y_{i_2,t}^{(c_2)})\right]+\rho B \quad (3.16)$$

Second, we start from B,

$$\begin{aligned}B=&\frac{1}{2\pi\sqrt{1-\rho^2}}\int_{a_1}^{b_1}\int_{a_2}^{b_2}z_1z_2\exp\left(-\frac{1}{2}\frac{z_1^2-2\rho z_1z_2+z_2^2}{1-\rho^2}\right)dz_2dz_1\\ =&\frac{1}{2\pi\sqrt{1-\rho^2}}\int_{a_1}^{b_1}\int_{a_2}^{b_2}(z_1z_2-\rho z_1^2)\exp\left(-\frac{1}{2}\frac{z_1^2-2\rho z_1z_2+z_2^2}{1-\rho^2}\right)dz_2dz_1\\ &+\frac{1}{2\pi\sqrt{1-\rho^2}}\int_{a_1}^{b_1}\int_{a_2}^{b_2}\rho z_1^2\exp\left(-\frac{1}{2}\frac{z_1^2-2\rho z_1z_2+z_2^2}{1-\rho^2}\right)dz_2dz_1 \qquad (3.17)\end{aligned}$$

The second term in (3.17) is ρA, the first term can be simplified as

$$\begin{aligned}&-(1-\rho^2)\int_{a_1}^{b_1}z_1\frac{1}{2\pi\sqrt{1-\rho^2}}\left[\exp\left(-\frac{1}{2}\frac{z_1^2-2\rho z_1z_2+z_2^2}{1-\rho^2}\right)\Big|_{a_2}^{b_2}\right]dz_1\\ =&(1-\rho^2)\left[\int_{a_1}^{b_1}z_1\phi_\rho(a_2,z_1)dz_1-\int_{a_1}^{b_1}z_1\phi_\rho(b_2,z_1)dz_1\right]\end{aligned}$$

It can be derived that

$$\int_a^b z\phi_\rho(c,z)dz=(1-\rho^2)\left[\phi_\rho(a,c)-\phi_\rho(b,c)\right]+\rho cf(a,b,c,\rho). \quad (3.18)$$

Substituting (3.18) to the first term in (3.17), we can get the second equation between A and B:

$$\begin{aligned}B=&(1-\rho^2)^2\left[\phi_\rho(a_1,a_2)-\phi_\rho(b_1,a_2)-\phi_\rho(a_1,b_2)+\phi_\rho(b_2,b_1)\right]\\ &+(1-\rho^2)\rho\left[a_2f(a_1,b_1,a_2,\rho)-b_2f(a_1,b_1,b_2,\rho)\right]+\rho A. \qquad (3.19)\end{aligned}$$

Solving Equations (3.16) and (3.19) gives the expression of A and B, the expression of C is symmetrical to A in terms of indexes.

Finally, substitute A, B and C to (3.14), which is the numerator of (3.12), one can get, after some algebra, the partial derivative of the composite likelihood with respect to

correlation parameters

$$\frac{\partial cl(\boldsymbol{\theta};\, y_{i_1,t}^{(c_1)}, y_{i_2,t}^{(c_2)})}{\partial \rho} = \frac{\phi_\rho(b_1,a_2)+\phi_\rho(a_1,b_2)-\phi_\rho(a_1,a_2)-\phi_\rho(b_2,b_1)}{CL(\boldsymbol{\theta};\, y_{i_1,t}^{(c_1)}, y_{i_2,t}^{(c_2)})} \tag{3.20}$$

2. Hessian matrix

The Hessian matrix is estimated as

$$\hat{\boldsymbol{H}}(\boldsymbol{\theta}) = \sum_{t=1}^{T} \sum_{i_1 \sim i_2} \sum_{c_1=1}^{n_{\mathscr{C}}} \sum_{c_2=1}^{n_{\mathscr{C}}} \frac{\partial^2 cl(\boldsymbol{\theta};\, y_{i_1,t}^{(c_1)}, y_{i_2,t}^{(c_2)})}{\partial \boldsymbol{\theta}^2},$$

where the second derivative of $cl(\boldsymbol{\theta};\, y_{i_1,t}^{(c_1)}, y_{i_2,t}^{(c_2)})$ can be written as a bock matrix

$$\frac{\partial^2 cl(\boldsymbol{\theta};\, y_{i_1,t}^{(c_1)}, y_{i_2,t}^{(c_2)})}{\partial \boldsymbol{\theta}^2} = \begin{bmatrix} \hat{\boldsymbol{H}}_{11}(\boldsymbol{\theta}, y_{i_1,t}^{(c_1)}, y_{i_2,t}^{(c_2)}) & \hat{\boldsymbol{H}}_{12}(\boldsymbol{\theta}, y_{i_1,t}^{(c_1)}, y_{i_2,t}^{(c_2)}) \\ \hat{\boldsymbol{H}}_{12}(\boldsymbol{\theta}, y_{i_1,t}^{(c_1)}, y_{i_2,t}^{(c_2)})^T & \hat{\boldsymbol{H}}_{22}(\boldsymbol{\theta}, y_{i_1,t}^{(c_1)}, y_{i_2,t}^{(c_2)}) \end{bmatrix},$$

where $\hat{\boldsymbol{H}}_{11}$ and $\hat{\boldsymbol{H}}_{22}$ are the second derivative of $cl(\boldsymbol{\theta};\, y_{i_1,t}^{(c_1)}, y_{i_2,t}^{(c_2)})$ with respect to mean and correlation parameters respectively, and $\hat{\boldsymbol{H}}_{12}$ is the mixed derivative of $cl(\boldsymbol{\theta};\, y_{i_1,t}^{(c_1)}, y_{i_2,t}^{(c_2)})$ with respect to mean and correlation parameters.

2.1 $\hat{\boldsymbol{H}}_{22}(\boldsymbol{\theta};\, y_{i_1,t}^{(c_1)}, y_{i_2,t}^{(c_2)})$

Since only one correlation parameter appear in each component composite likelihood,

$$\frac{\partial^2 cl(\boldsymbol{\theta};\, y_{i_1,t}^{(c_1)}, y_{i_2,t}^{(c_2)})}{\partial \rho_1 \partial \rho_2} = 0 \quad \text{if} \quad \rho_1 \neq \rho_2,$$

which means $\hat{\boldsymbol{H}}_{22}$ is a $n_{\mathscr{C}}^2 \times n_{\mathscr{C}}^2$ diagonal matrix. Taking derivative of (3.20) with respect to ρ, the diagonal entries of $\hat{\boldsymbol{H}}_{22}$ are

$$\frac{\partial^2 cl(\boldsymbol{\theta};\, y_{i_1,t}^{(c_1)}, y_{i_2,t}^{(c_2)})}{\partial \rho^2} = \frac{\dfrac{\partial g(\boldsymbol{\theta};\, y_{i_1,t}^{(c_1)}, y_{i_2,t}^{(c_2)})}{\partial \rho} CL(\boldsymbol{\theta};\, y_{i_1,t}^{(c_1)}, y_{i_2,t}^{(c_2)}) - g(\boldsymbol{\theta};\, y_{i_1,t}^{(c_1)}, y_{i_2,t}^{(c_2)})^2}{CL(\boldsymbol{\theta};\, y_{i_1,t}^{(c_1)}, y_{i_2,t}^{(c_2)})^2},$$

where

$$g(\boldsymbol{\theta};\, y_{i_1,t}^{(c_1)}, y_{i_2,t}^{(c_2)}) = \phi_\rho(b_1,a_2) + \phi_\rho(a_1,b_2) - \phi_\rho(a_1,a_2) - \phi_\rho(b_2,b_1) \tag{3.21}$$

$$\frac{\partial g(\boldsymbol{\theta};\, y_{i_1,t}^{(c_1)}, y_{i_2,t}^{(c_2)})}{\partial \rho} = \frac{d\phi_\rho(b_1,a_2)}{d\rho} + \frac{d\phi_\rho(a_1,b_2)}{d\rho} - \frac{d\phi_\rho(a_1,a_2)}{d\rho} - \frac{d\phi_\rho(b_1,b_2)}{d\rho}$$

$$\frac{d\phi_\rho(x,y)}{d\rho} = \phi_\rho(x,y)\frac{1}{(1-\rho^2)^{3/2}}\left[\rho - \frac{\rho x^2 - (1+\rho^2)xy + \rho y^2}{1-\rho^2}\right].$$

2.2 $\hat{\boldsymbol{H}}_{12}(\boldsymbol{\theta};\, y_{i_1,t}^{(c_1)}, y_{i_2,t}^{(c_2)})$

This is a $(2n_{\mathscr{C}}) \times (n_{\mathscr{C}}^2)$ matrix, the entries are the mixed derivatives of $cl(\boldsymbol{\theta};\, y_{i_1,t}^{(c_1)}, y_{i_2,t}^{(c_2)})$ with respect to the mean and correlation parameters. Without loss of generosity, consider only $\partial^2 cl(\boldsymbol{\theta};\, y_{i_1,t}^{(c_1)}, y_{i_2,t}^{(c_2)})/\partial\rho\partial\alpha^{(c_1)}$, where $c_1, c_2 \in \{1,\ldots,n_{\mathscr{C}}\}$. Taking derivative of (3.20) with respect to $\alpha^{(c_1)}$, we have

$$\frac{\partial^2 cl(\boldsymbol{\theta};\, y_{i_1,t}^{(c_1)}, y_{i_2,t}^{(c_2)})}{\partial\rho\partial\alpha^{(c_1)}} = \frac{\dfrac{\partial g(\boldsymbol{\theta};\, y_{i_1,t}^{(c_1)}, y_{i_2,t}^{(c_2)})}{\partial\alpha^{(c_1)}} - \dfrac{cl(\boldsymbol{\theta};\, y_{i_1,t}^{(c_1)}, y_{i_2,t}^{(c_2)})}{\partial\alpha^{(c_1)}} g(\boldsymbol{\theta};\, y_{i_1,t}^{(c_1)}, y_{i_2,t}^{(c_2)})}{CL(\boldsymbol{\theta};\, y_{i_1,t}^{(c_1)}, y_{i_2,t}^{(c_2)})},$$

where $g(\boldsymbol{\theta};\, y_{i_1,t}^{(c_1)}, y_{i_2,t}^{(c_2)})$ is defined in (3.21).

- If $c_1 \neq c_2$, then $\partial cl(\boldsymbol{\theta};\, y_{i_1,t}^{(c_1)}, y_{i_2,t}^{(c_2)})/\partial\alpha^{(c_1)}$ is defined in (3.9) and

$$\frac{\partial g(\boldsymbol{\theta};\, y_{i_1,t}^{(c_1)}, y_{i_2,t}^{(c_2)})}{\partial\alpha^{(c_1)}} = \frac{\partial a_1}{\partial\alpha^{(c_1)}}\left(\frac{\partial\phi(a_1,b_2)}{\partial a_1} - \frac{\partial\phi(a_1,a_2)}{\partial a_1}\right) + \frac{\partial b_1}{\partial\alpha^{(c_1)}}\left(\frac{\partial\phi(b_1,a_2)}{\partial b_1} - \frac{\partial\phi(b_1,b_2)}{\partial b_1}\right).$$

- If $c_1 = c_2$, then $\partial cl(\boldsymbol{\theta};\, y_{i_1,t}^{(c_1)}, y_{i_2,t}^{(c_2)})/\partial\alpha^{(c_1)}$ is defined in (3.10) and

$$\begin{aligned}\frac{\partial g(\boldsymbol{\theta};\, y_{i_1,t}^{(c_1)}, y_{i_2,t}^{(c_2)})}{\partial\alpha^{(c_1)}} =& \frac{\partial a_1}{\partial\alpha^{(c_1)}}\left(\frac{\partial\phi(a_1,b_2)}{\partial a_1} - \frac{\partial\phi(a_1,a_2)}{\partial a_1}\right) + \frac{\partial b_1}{\partial\alpha^{(c_1)}}\left(\frac{\partial\phi(b_1,a_2)}{\partial b_1} - \frac{\partial\phi(b_1,b_2)}{\partial b_1}\right) + \\ & \frac{\partial a_1}{\partial\alpha^{(c_1)}}\left(\frac{\partial\phi(b_1,a_2)}{\partial a_1} - \frac{\partial\phi(a_1,a_2)}{\partial a_2}\right) + \frac{\partial b_1}{\partial\alpha^{(c_1)}}\left(\frac{\partial\phi(a_1,b_2)}{\partial b_1} - \frac{\partial\phi(b_1,b_2)}{\partial b_2}\right),\end{aligned}$$

  where

$$\frac{\partial\phi(x,y)}{\partial x} = \frac{\rho y - x}{1-\rho^2}\phi(x,y).$$

2.3 $\hat{\boldsymbol{H}}_{11}(\boldsymbol{\theta};\, y_{i_1,t}^{(c_1)}, y_{i_2,t}^{(c_2)})$

This is a $(2n_{\mathscr{C}}) \times (2n_{\mathscr{C}})$ matrix, the entries are the second derivatives of $cl(\boldsymbol{\theta};\, y_{i_1,t}^{(c_1)}, y_{i_2,t}^{(c_2)})$ with respect to the mean parameters. Without loss of generosity, consider only the deriva-

tives with respect to $\alpha^{(c_1)}$ and/or $\alpha^{(c_2)}$.

It can be seen from (3.9) and (3.10), that

$$\frac{\partial cl(\boldsymbol{\theta};\, y_{i_1,t}^{(c_1)}, y_{i_2,t}^{(c_2)})}{\partial \alpha^{(c_1)}} = \frac{h(\boldsymbol{\theta},;\, y_{i_1,t}^{(c_1)}, y_{i_2,t}^{(c_2)})}{CL(\boldsymbol{\theta};\, y_{i_1,t}^{(c_1)}, y_{i_2,t}^{(c_2)})}, \tag{3.22}$$

where

$$\begin{aligned}
h(\boldsymbol{\theta},;\, y_{i_1,t}^{(c_1)}, y_{i_2,t}^{(c_2)}) =& \frac{\partial b_1}{\partial \alpha^{(c_1)}} f(a_2,b_2,b_1,\rho) - \frac{\partial a_1}{\partial \alpha^{(c_1)}} f(a_2,b_2,a_1,\rho) \quad \text{if } c_1 \neq c_2, \\
h(\boldsymbol{\theta},;\, y_{i_1,t}^{(c_1)}, y_{i_2,t}^{(c_2)}) =& \frac{\partial b_1}{\partial \alpha^{(c)}} f(a_2,b_2,b_1,\rho) - \frac{\partial a_1}{\partial \alpha^{(c)}} f(a_2,b_2,a_1,\rho) + \\
& \frac{\partial b_2}{\partial \alpha^{(c)}} f(a_1,b_1,b_2,\rho) - \frac{\partial a_2}{\partial \alpha^{(c)}} f(a_1,b_1,a_2,\rho) \quad \text{if } c_1 = c_2.
\end{aligned}$$

Thus, taking a derivative of (3.22) with respect to $\alpha^{(c)}, c \in \{c_1, c_2\}$ gives

$$\frac{\partial^2 cl(\boldsymbol{\theta};\, y_{i_1,t}^{(c_1)}, y_{i_2,t}^{(c_2)})}{\partial \alpha^{(c_1)} \partial \alpha^{(c)}} = \frac{\dfrac{\partial h(\boldsymbol{\theta},;\, y_{i_1,t}^{(c_1)}, y_{i_2,t}^{(c_2)})}{\partial \alpha^{(c)}} - \dfrac{\partial cl(\boldsymbol{\theta};\, y_{i_1,t}^{(c_1)}, y_{i_2,t}^{(c_2)})}{\partial \alpha^{(c)}} h(\boldsymbol{\theta},;\, y_{i_1,t}^{(c_1)}, y_{i_2,t}^{(c_2)})}{CL(\boldsymbol{\theta};\, y_{i_1,t}^{(c_1)}, y_{i_2,t}^{(c_2)})},$$

where $\partial cl(\boldsymbol{\theta};\, y_{i_1,t}^{(c_1)}, y_{i_2,t}^{(c_2)})/\partial \alpha^{(c_1)}$ is given by (3.9), and $\partial cl(\boldsymbol{\theta};\, y_{i_1,t}^{(c_1)}, y_{i_2,t}^{(c_2)})/\partial \alpha^{(c_2)}$ can be derived similarly. Only $\partial h(\boldsymbol{\theta},;\, y_{i_1,t}^{(c_1)}, y_{i_2,t}^{(c_2)})/\partial \alpha^{(c)}$ differs in three case:

- If $c_1 \neq c_2,\ c = c_2$, i.e. $\dfrac{\partial^2 cl(\boldsymbol{\theta};\, y_{i_1,t}^{(c_1)}, y_{i_2,t}^{(c_2)})}{\partial \alpha^{(c_1)} \partial \alpha^{(c_2)}}$, then

$$\begin{aligned}
\frac{\partial h(\boldsymbol{\theta},;\, y_{i_1,t}^{(c_1)}, y_{i_2,t}^{(c_2)})}{\alpha^{(c_2)}} =& \frac{\partial b_1}{\partial \alpha^{(c_1)}} \left[\frac{\partial b_2}{\partial \alpha^{(2)}} \phi_\rho(b_1,b_2) - \frac{\partial b_2}{\partial \alpha^{(2)}} \phi_\rho(b_1,a_2) \right] \\
& - \frac{\partial a_1}{\partial \alpha^{(c_1)}} \left[\frac{\partial b_2}{\partial \alpha^{(2)}} \phi_\rho(a_1,b_2) - \frac{\partial a_2}{\partial \alpha^{(2)}} \phi_\rho(a_1,a_2) \right],
\end{aligned}$$

  since $\dfrac{\partial f(a,b,c,\rho)}{\partial a} = \phi_\rho(a,c)$ and $\dfrac{\partial f(a,b,c,\rho)}{\partial b} = \phi_\rho(b,c)$.

- If $c_1 \neq c_2,\ c = c_1$, i.e. $\dfrac{\partial^2 cl(\boldsymbol{\theta};\, y_{i_1,t}^{(c_1)}, y_{i_2,t}^{(c_2)})}{\partial^2 \alpha^{(c_1)}}$, then

$$\begin{aligned}
\frac{\partial h(\boldsymbol{\theta},;\, y_{i_1,t}^{(c_1)}, y_{i_2,t}^{(c_2)})}{\alpha^{(c_1)}} =& \frac{\partial^2 b_1}{\partial \alpha^{(c_1)^2}} f(a_2,b_2,b_1,\rho) + \left(\frac{\partial b_1}{\partial \alpha^{(c_1)}} \right)^2 \frac{f(a_2,b_2,b_1,\rho)}{\partial b_1} \\
& - \frac{\partial^2 a_1}{\partial \alpha^{(c_1)^2}} f(a_2,b_2,a_1,\rho) - \left(\frac{\partial a_1}{\partial \alpha^{(c_1)}} \right)^2 \frac{f(a_2,b_2,a_1,\rho)}{\partial a_1},
\end{aligned}$$

where

$$\begin{aligned}
\frac{\partial f(a,b,c,\rho)}{\partial c} &= \int_a^b \frac{\partial \phi_\rho(c,z)}{\partial c} dz \\
&= \int_a^b \left(-\frac{c-\rho z}{1-\rho^2}\right) \frac{1}{2\pi\sqrt{1-\rho^2}} \exp\left(-\frac{1}{2}\frac{c^2-2\rho cz+z^2}{1-\rho^2}\right) dz \\
&= \frac{1}{1-\rho^2}\left[-cf(a,b,c,\rho) + \rho \int_a^b z\phi_\rho(c,z)dz\right] \\
&\text{where} \int_a^b z\phi_\rho(c,z)dz = -(1-\rho^2)\left[\phi_\rho(b,c) - \phi_\rho(a,c)\right] + \rho c f(a,b,c,\rho), \\
&= -cf(a,b,c,\rho) + \rho\left[\phi_\rho(a,c) - \phi_\rho(b,c)\right]. \qquad (3.23)
\end{aligned}$$

- If $c_1 = c_2 = c$, i.e. $\dfrac{\partial^2 cl(\boldsymbol{\theta};\, y_{i_1,t}^{(c_1)}, y_{i_2,t}^{(c_1)})}{\partial^2 \alpha^{(c_1)}}$, then

$$\begin{aligned}
\frac{\partial h(\boldsymbol{\theta},;\, y_{i_1,t}^{(c_1)}, y_{i_2,t}^{(c_1)})}{\alpha^{(c_1)}} =& \frac{\partial^2 b_1}{\partial \alpha^{(c_1)^2}} f(a_2,b_2,b_1,\rho) + \left(\frac{\partial b_1}{\partial \alpha^{(c)}}\right)^2 \frac{f(a_2,b_2,b_1,\rho)}{\partial b_1} + \\
& \frac{\partial^2 a_1}{\partial \alpha^{(c_1)^2}} f(a_2,b_2,a_1,\rho) + \left(\frac{\partial a_1}{\partial \alpha^{(c)}}\right)^2 \frac{f(a_2,b_2,a_1,\rho)}{\partial a_1} + \\
& \frac{\partial^2 b_2}{\partial \alpha^{(c_1)^2}} f(a_2,b_2,b_2,\rho) + \left(\frac{\partial b_2}{\partial \alpha^{(c)}}\right)^2 \frac{f(a_2,b_2,b_2,\rho)}{\partial b_2} + \\
& \frac{\partial^2 a_2}{\partial \alpha^{(c_1)^2}} f(a_2,b_2,a_2,\rho) + \left(\frac{\partial a_2}{\partial \alpha^{(c)}}\right)^2 \frac{f(a_2,b_2,a_2,\rho)}{\partial a_2},
\end{aligned}$$

where the partial derivatives of $f(\cdot)$ is given by (3.23).

Chapter 4

Model Selection via Gibbs Sampling

4.1 Introduction

Generalized linear models (GLMs) provide flexible framework to describe how a dependent variable can be explained by a range of explanatory variables (covariates) and are widely used in many fields of science. However, the assumption of independent response has become a major limit for univariate regression models. Gaussian copula models (Xue-Kun Song, 2000) are often considered as a multivariate extension of the GLMs. One principal merit is that the specification of the regression model is separated from the dependence structure. Gaussian copula regression models have been successfully employed in several complex applications arising, for example, in longitudinal data (Song et al., 2013) and spatial statistics (Nikoloulopoulos, 2016; Bai et al., 2014). In this chapter, we consider Gaussian copula log-linear regression models on correlated count data, with Poisson or Negative binomial marginals.

The maximum likelihood approaches for inference are generally considered the best options for estimating the model parameters. However, while likelihood computation for continuous responses are straightforward, the discrete case is considerably more difficult because the likelihood function involves multidimensional Gaussian integrals. In order to reduce the integral dimensionality, we use the pairwise composite likelihood (CL) methods (Varin et al., 2011) as a numerical approximation to the full likelihood function.

Yet fitting a single model is not satisfactory in all circumstances. In many situations, one wants to decide, among all the parameters in the model, which are important in some way to describe the response variable, in other words, which one should be retained, and which one should be dropped. Determining the removal or addition of a given term can be done in several ways. Hypothesis test tools such as t test or LR test, involve specifying a significance threshold for the p values. Since the number of tests is typically high, this poses the problem of choosing a relevant significant level. Alternatively, a more popular approach is the information criterion (IC) based methods, for example the well

known AIC (Akaikei, 1973), which selects models with the best prediction power, and BIC (Schwarz et al., 1978) which is shown to be consistent in many settings. However a crucial assumption for both AIC and BIC is the measure of goodness-of-fit needs to be the full likelihood functions, which is not available under the CL framework. Therefore, we adopt a model selection criterion developed by Gao and Song (2010), the composite likelihood BIC (CL-BIC), that is shown to be consistent under mild regularity conditions and produces satisfactory selection results in our numerical experiments.

A full IC-based selection is to compare all candidate models and rank them based on their IC values. However, because the number of candidate models grows exponentially with the number of parameters, such exhaustive search can easily become infeasible even for a moderate number of parameters. For example, if a full model has 20 parameters, then the number of all candidate models to be evaluated is 2^{20}. One common approach for handling this computational complication is the stepwise selection (see for example Miller (2002)). Apart from its dependence on the starting point and stopping rules, the major drawback is that it does not guarantee convergence, and even if it does converge, the backward and forward approaches are not generally expected to converge to the same model (Venables and Ripley, 2013). Another solution is through the penalized likelihood methods like LASSO (Tibshirani, 1996) and SCAD (Fan and Li, 2001), however these methods rely heavily on the choice of the tuning parameter on penalty strength, besides we encounter convergence problems when fitting composite likelihood with a LASSO penalty to our Gaussian copula models.

In this chapter, we focus on a fast and consistent model selection procedure that uses an MCMC approach, first introduced by Qian and Field (2002) on logistic regression models. The method can handle large candidate model set and the convergence of the MCMC method ensures that the selected model has the lowest IC among all candidate models, provided that the MCMC sample is sufficiently large. Although in this thesis we only focus on selection of the copula-based model, this selection method is so flexible that it can be applied to a wide variety of regression based models as an efficient variable subsetting toolkit, as long as the information criterion is properly chosen.

This chapter is organized as follows. Section 4.2 briefly summarizes the model framework and the estimation of the information criterion, CL-BIC, and describes in detail the procedure and algorithm of the model selection method. Section 4.3 evaluates the performance of the model selection with simulation experiments. Section 4.4 provides an illustrative example on real data.

4.2 Model Selection based on Gibbs sampling

4.2.1 Model framework

Let $\boldsymbol{Y} = (\boldsymbol{Y}_1, \ldots \boldsymbol{Y}_T)$ be a grouped count data set, which can be sensibly assumed to have a Poisson or negative binomial marginal distribution, where $\boldsymbol{Y}_t = (Y_{1,t}, \ldots, Y_{d,t})'$ are d-dimensional vectors of observations. Let $\boldsymbol{\Sigma}$ denote the $d \times d$ correlation matrix of $\boldsymbol{Y}_t$, independent of t. For convenience, we take the same correlation structure as discussed in Chapter 3, yet the methodology can be easily generalized to handle other types of correlation structures. Recall that we assumed the data set is observed on a spatial lattice, or a regular $n_{\mathscr{L}} = n \times n$ grid, observation on each tile is a $n_{\mathscr{C}}$-dimensional vector corresponding to counts of $n_{\mathscr{C}}$ groups/clusters. Thus, $\boldsymbol{Y}_t$ is also written as $(Y_{1,t}^{(1)}, \ldots, Y_{1,t}^{(n_{\mathscr{C}})}, Y_{2,t}^{(1)}, \ldots, Y_{2,t}^{(n_{\mathscr{C}})}, \ldots, Y_{n_{\mathscr{L}},t}^{(1)}, \ldots, Y_{n_{\mathscr{L}},t}^{(n_{\mathscr{C}})})'$ and $d = n_{\mathscr{C}} n_{\mathscr{L}}$. The correlation matrix $\boldsymbol{\Sigma}$ is then parameterized in a straightforward manner:

$$\boldsymbol{\Sigma} = \begin{pmatrix} \boldsymbol{R}_0 & \boldsymbol{R}_1 & \boldsymbol{0} & \boldsymbol{0} & \ldots & \boldsymbol{R}_1 & \boldsymbol{0} & \ldots & \ldots \\ \boldsymbol{R}_1 & \boldsymbol{R}_0 & \boldsymbol{R}_1 & \boldsymbol{0} & \ldots & \ldots & \boldsymbol{R}_1 & \ldots & \ldots \\ \boldsymbol{0} & \boldsymbol{R}_1 & \boldsymbol{R}_0 & \boldsymbol{R}_1 & \ldots & \ldots & \ldots & \ldots & \ldots \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \boldsymbol{R}_0 \end{pmatrix}_{n_{\mathscr{L}} \times n_{\mathscr{L}}},$$

where each entry is a $n_{\mathscr{C}} \times n_{\mathscr{C}}$ matrix representing cross group/cluster correlation in the same tile $\boldsymbol{R}_0$ and neighbouring tiles $\boldsymbol{R}_1$ parametrized as

$$\boldsymbol{R}_0 = \begin{pmatrix} 1 & \rho_0^{(1,2)} & \rho_0^{(1,3)} & \ldots & \rho_0^{(1,n_{\mathscr{C}})} \\ \rho_0^{(1,2)} & 1 & \rho_0^{(2,3)} & \ldots & \rho_0^{(2,n_{\mathscr{C}})} \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ \rho_0^{(1,n_{\mathscr{C}})} & \ldots & \ldots & \ldots & 1 \end{pmatrix}, \quad \boldsymbol{R}_1 = \begin{pmatrix} \rho_1^{(1)} & \rho_1^{(1,2)} & \rho_1^{(1,3)} & \ldots & \rho_1^{(1,n_{\mathscr{C}})} \\ \rho_1^{(1,2)} & \rho_1^{(2)} & \rho_1^{(2,3)} & \ldots & \rho_1^{(2,n_{\mathscr{C}})} \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ \rho_1^{(1,n_{\mathscr{C}})} & \ldots & \ldots & \ldots & \rho_1^{(n_{\mathscr{C}})} \end{pmatrix}.$$

The correlation parameter vector is denoted as $\boldsymbol{\rho} = (\rho^{(1)}, \ldots, \rho^{(n_{\mathscr{C}})}, \rho_0^{(1,2)}, \ldots, \rho_0^{(n_{\mathscr{C}}-1,n_{\mathscr{C}})}, \rho_1^{(1,2)}, \ldots, \rho_1^{(n_{\mathscr{C}}-1,n_{\mathscr{C}})})$ with dimension $p_\rho = n_{\mathscr{C}}^2$.

The dependence structure is combined with the marginal distribution of $Y_{i,t}^{(c)}$ by the Gaussian copula model (Xue-Kun Song, 2000) that can be specified as the joint data cumulative distribution function

$$\boldsymbol{Y}_t | \boldsymbol{Y}_{t-1} = \left(F_{\mu_{1,t}^{(1)}}^{-1} \left[\Phi(Z_1)\right], \ldots, F_{\mu_{1,t}^{(n_{\mathscr{C}})}}^{-1} \left[\Phi(Z_{n_{\mathscr{C}}})\right], F_{\mu_{2,t}^{(1)}}^{-1} \left[\Phi(Z_{n_{\mathscr{C}}+1})\right], \ldots, F_{\mu_{n_{\mathscr{L}},t}^{(n_{\mathscr{C}})}}^{-1} \left[\Phi(Z_d)\right] \right), \tag{4.1}$$

where $\boldsymbol{Z} = (Z_1, \ldots, Z_d)$ has a d-dimensional multivariate standard normal cumulative distribution with correlation matrix $\boldsymbol{\Sigma}$, Φ denotes the univariate standard normal cumulative

distribution function and $F^{-1}_{\mu^{(c)}_{i,t}}(u) = \inf\{y : F_{\mu^{(c)}_{i,t}}(y) \leq u\}$ for $0 \leq u \leq 1$ with $\mu^{(c)}_{i,t}$ being the expectation of $Y^{(c)}_{i,t}$.

The expected values of $Y^{(c)}_{i,t}$, $\mu^{(c)}_{i,t}$ is assumed to depend on a vector of explanatory variables through the relationship

$$g(\mu^{(c)}_{i,t}) = \boldsymbol{X}^{(c)\prime}_{i,t}\boldsymbol{\beta},$$

for a suitable link function $g(\cdot)$ and a p_β-dimensional vector of regression coefficients $\boldsymbol{\beta}$. This setting encompasses a variety of popular model classes, for example the GLMs. In this chapter, we focus on the log link function. Recall that in Chapter 3, we consider a temporal setting where $\boldsymbol{X}^{(c)}_{i,t}$ depends on the past observations of Y_t:

$$\log(\mu^{(c)}_{i,t}) = \beta^{(c)}_0 + \sum_{c'=1}^{n_{\mathscr{C}}} S^{(c')}_{i,t-1}\beta^{(c|c')} \quad \text{and} \quad S^{(c')}_{i,t-1} = \frac{1}{n_i}\sum_{i\sim j}\log\left(1 + Y^{(c')}_{j,t-1}\right), \tag{4.2}$$

where $\sim$ denotes neighbouring locations and n_i denotes the number of neighbouring tiles of tile i. In this case, the regression coefficients is specified as $\boldsymbol{\beta} = (\beta^{(1)}_0, \beta^{(1|1)}, \beta^{(1|2)}, \ldots, \beta^{(2)}_0, \beta^{(2|1)}, \ldots, \beta^{(n_{\mathscr{C}}|n_{\mathscr{C}})})'$ and thus $p_\beta = n_{\mathscr{C}}(n_{\mathscr{C}}+1)$. This marginal model falls into a more general class of observation-driven time series models, the generalised autoregressive moving average model, GARMA(p,q) (Benjamin et al., 2003) with $p = 1, q = 0$.

In this chapter, we consider model selection for the (temporal) model described above. Besides, we also discuss a simpler, non-temporal setting considered by Gao and Song (2010) where covariates are external explanatory variables instead of formed by previous responses, and the vector responses $\boldsymbol{Y}_1, \ldots \boldsymbol{Y}_T$ are considered independent.

4.2.2 Composite likelihood inference

Denote the full parameter as $\boldsymbol{\theta} = (\boldsymbol{\beta}, \boldsymbol{\rho})'$ with dimension $p = p_\beta + p_\rho$, which is estimated by maximizing the pairwise log-composite likelihood (CL):

$$cl(\boldsymbol{\theta}; \mathbf{y}) = \sum_{t=1}^{T}\sum_{i_1,i_2=1}^{n_{\mathscr{L}}}\sum_{c_1,c_2=1}^{n_{\mathscr{C}}} \log\left[\int_{a^{(c_1)}_{i_1,t}}^{b^{(c_1)}_{i_1,t}}\int_{a^{(c_2)}_{i_2,t}}^{b^{(c_2)}_{i_2,t}} \phi_\rho(z_1,z_2)dz_1dz_2\right],$$

where $\phi_\rho(z_1,z_2)$ denotes a bivariate standard normal density function with correlation ρ, and ρ is the correlation between $Y^{(c_1)}_{i_1,t}$ and $Y^{(c_2)}_{i_2,t}$. For computational concern, we compute only those component likelihood pairs that involve observations in neighbouring tiles.

Specifically, let Ω be the collection of pairwise index subsets $\{s = (i_1, i_2, c_1, c_2) : c_1, c_2 = 1, \ldots, n_{\mathscr{C}}, i_1, i_2 = 1, \ldots, n_{\mathscr{L}}, i_1 \sim i_2\}$, and denote $y_{s,t}$ as the pair of observations

$(y_{i_1,t}^{(c_1)}, y_{i_2,t}^{(c_2)})$, then the CL function can be rewritten as

$$cl(\boldsymbol{\theta};\, \boldsymbol{y}) = \sum_{t=1}^{T} \sum_{s \in \Omega} cl(\boldsymbol{\theta}; y_{s,t}). \tag{4.3}$$

4.2.3 Estimation of Model Selection Criterion

Following notations by Gao and Song (2010), Bayes information criterion (BIC) in the CL framework can be specified as

$$\text{CL-BIC } = -2cl(\hat{\boldsymbol{\theta}}; \boldsymbol{y}) + (\log(T) + 2\gamma \log(p))\, d^*, \tag{4.4}$$

where γ is a tuning parameter controlling penalty strength/forcing sparsity and $d^* = \text{trace}[\boldsymbol{H}^{-1}(\hat{\boldsymbol{\theta}})\boldsymbol{K}(\hat{\boldsymbol{\theta}})]$, $\boldsymbol{H}(\boldsymbol{\theta};\, \boldsymbol{y})$ and $\boldsymbol{K}(\boldsymbol{\theta};\, \boldsymbol{y})$ denote negative Hessian matrix and the variance of the first derivative of the CL function in (4.3) respectively:

$$\boldsymbol{H}(\boldsymbol{\theta};\, \boldsymbol{y}) = -\mathbf{E}\left[\frac{\partial^2 cl(\boldsymbol{\theta};\, \boldsymbol{y})}{\partial \boldsymbol{\theta}^2}\right], \quad \boldsymbol{K}(\boldsymbol{\theta};\, \boldsymbol{y}) = \mathbf{Var}\left[\frac{\partial cl(\boldsymbol{\theta};\, \boldsymbol{y})}{\partial \boldsymbol{\theta}}\right].$$

Since the second Bartlett identity remains true for each component likelihood, $\boldsymbol{H}(\hat{\boldsymbol{\theta}}\, \boldsymbol{y})$ can be reasonably estimated as

$$\hat{\boldsymbol{H}}(\hat{\boldsymbol{\theta}};\, \boldsymbol{y}) = \sum_{t=1}^{T} \sum_{s \in \Omega} \boldsymbol{u}(\hat{\boldsymbol{\theta}};\, y_{s,t}) \boldsymbol{u}(\hat{\boldsymbol{\theta}};\, y_{s,t})', \tag{4.5}$$

where $\boldsymbol{u}(\boldsymbol{\theta};\, y_{s,t}) = \partial cl(\boldsymbol{\theta};\, y_{s,t}))/\partial \boldsymbol{\theta}$ denotes the component pair score function.

The estimation of $\boldsymbol{K}(\hat{\boldsymbol{\theta}};\, \boldsymbol{y})$ poses more difficulty. For the non-temporal setting where $\boldsymbol{Y}_1, \ldots \boldsymbol{Y}_T$ are considered independent, we take the sample variance of the composite score function for each t

$$\hat{\boldsymbol{K}}(\hat{\boldsymbol{\theta}};\, \boldsymbol{y}) = \frac{1}{T-1} \sum_{t=1}^{T} \left(\sum_{s} \boldsymbol{u}(\hat{\boldsymbol{\theta}};\, y_{s,t}) - \frac{1}{T} \boldsymbol{u}(\hat{\boldsymbol{\theta}};\, \boldsymbol{y}) \right) \left(\sum_{s} \boldsymbol{u}(\hat{\boldsymbol{\theta}};\, y_{s,t}) - \frac{1}{T} \boldsymbol{u}(\hat{\boldsymbol{\theta}};\, \boldsymbol{y}) \right)', \tag{4.6}$$

since the naive estimator $\left(\sum_{t=1}^{T} \sum_{s} \boldsymbol{u}(\hat{\boldsymbol{\theta}};\, y_{s,t})\right) \left(\sum_{t=1}^{T} \sum_{s} \boldsymbol{u}(\hat{\boldsymbol{\theta}};\, y_{s,t})\right)'$ vanishes when evaluated at the maximum CL estimator, as also stated by Gao and Song (2010).

For the temporal setting, we estimate $\boldsymbol{K}(\boldsymbol{\theta};\, \boldsymbol{y})$ via a parametric bootstrap approach. Specifically, we obtain B independent bootstrap samples $\boldsymbol{y}_{(1)}^*, \ldots, \boldsymbol{y}_{(B)}^*$ by generating data of each $t > 1$ from a multivariate Poisson distribution via the Gaussian copula specified in (4.1), where parameters are taken as the maximised CL estimator $\hat{\boldsymbol{\theta}}$ and observations of the first time point is kept the same as the original data set. The bootstrapped estimator

is obtained as

$$\hat{\boldsymbol{K}}(\boldsymbol{\theta}; \boldsymbol{y}) = \frac{1}{B}\sum_{b=1}^{B} \boldsymbol{u}(\hat{\boldsymbol{\theta}}; \boldsymbol{y}^*_{(b)})\boldsymbol{u}(\hat{\boldsymbol{\theta}}; \boldsymbol{y}^*_{(b)})'.$$

Algorithm 1 describes the estimation of d^* needed for CL-BIC in (4.4) in details.

Algorithm 1 Function to fit specified model and compute according CL-BIC **(ModelFit)**

Input: B: the number of bootstrap sample size,
temporal: logical variable indicating if bootstrap is needed,
γ: tuning parameter for adjusting penalty strength in CL-BIC.
Output: CL-BIC value.

Fit the data into a log-linear regression model for initial parameter estimates $\hat{\boldsymbol{\theta}}$
While not converged **do** $\hat{\boldsymbol{\theta}}$ += $\boldsymbol{H}^{-1}(\hat{\boldsymbol{\theta}})\boldsymbol{u}(\hat{\boldsymbol{\theta}})$ ▷ Estimation on original data
Declare $\boldsymbol{H}$ and $\boldsymbol{K}$ as $p \times p$ matrices
if temporal **then**
    Initialize $\boldsymbol{H}$ and $\boldsymbol{K}$ zero matrices
    for b = 1 to B **do** ▷ Start bootstrapping
        Take $\boldsymbol{y}_0$ from the original data, $\hat{\boldsymbol{\theta}}^*_{(b)} = \hat{\boldsymbol{\theta}}$
        for t = 1 to T **do** ▷ Genrate bootstrap samples
            Generate $\boldsymbol{y}_t$ using Model described in (4.1) and (4.2) with $\hat{\boldsymbol{\theta}}$ substituted
        end for
        while not converged **do** ▷ Estimation on bootstrapped sample
            Declare and initialize $\boldsymbol{H}(\hat{\boldsymbol{\theta}}^*_{(b)})$ and $\boldsymbol{u}(\hat{\boldsymbol{\theta}}^*_{(b)})$ as zero matrix and vector
            for t = 1 to T **do**
                $\boldsymbol{H}(\hat{\boldsymbol{\theta}}^*_{(b)})$ += $\boldsymbol{H}_t(\hat{\boldsymbol{\theta}}^*_{(b)})$; $\boldsymbol{u}(\hat{\boldsymbol{\theta}}^*_{(b)})$ += $\boldsymbol{u}_t(\hat{\boldsymbol{\theta}}^*_{(b)})$;
                $\boldsymbol{K}$ += $\boldsymbol{u}_t(\hat{\boldsymbol{\theta}}^*_{(b)}) \cdot \boldsymbol{u}_t(\hat{\boldsymbol{\theta}}^*_{(b)})'$;
            end for
            $\hat{\boldsymbol{\theta}}^*_{(b)}$ += $\boldsymbol{H}^{-1}(\hat{\boldsymbol{\theta}}^*_{(b)})\boldsymbol{u}(\hat{\boldsymbol{\theta}}^*_{(b)})$;
        end while
        $\boldsymbol{H}$ += $\boldsymbol{H}(\hat{\boldsymbol{\theta}}^*_{(b)})$
    end for
    $\boldsymbol{H}$ /= B; $\boldsymbol{K}$ /= B;
else
    Initialize $\boldsymbol{u}(\hat{\boldsymbol{\theta}})$ as a zero vector; Declare ***tmpK*** as a $p \times T$ matrix.
    for t = 0 to $T-1$ **do**
        $\boldsymbol{H}(\hat{\boldsymbol{\theta}})$ += $\boldsymbol{H}_t(\hat{\boldsymbol{\theta}})$; ***tmpK***[, t] = $\boldsymbol{u}_t(\hat{\boldsymbol{\theta}})$; $\boldsymbol{u}(\hat{\boldsymbol{\theta}})$ += $\boldsymbol{u}_t(\hat{\boldsymbol{\theta}})$;
    end for
    $\boldsymbol{H} = \boldsymbol{H}(\hat{\boldsymbol{\theta}})/T$; ***tmpK***.eachcol() -= $\boldsymbol{u}(\hat{\boldsymbol{\theta}})/T$; $\boldsymbol{K} = \boldsymbol{tmpK} \cdot \boldsymbol{tmpK}'$;
end if
Substitute $d^* = \text{trace}(\boldsymbol{H}^{-1}\boldsymbol{K})$ into Equation (4.4) for CL-BIC output.

4.2.4 Model Selection Procedure

It is natural to search through all possible models and find the one that minimizes the criterion. However, since the number of candidate models grows exponentially with the dimension of the parameter, an exhaustive search quickly becomes computationally infeasible even for a moderate number of parameters. For example, in the temporal setting (i.e. the copSTM model proposed in Chapter 3), suppose $n_{\mathscr{C}} = 3, n_{\mathscr{L}} = 10 \times 10$ and $T = 10$, parameter estimation takes up to 45s, apart from time consumed in the bootstrapping procedure. With this model setup, the number of parameters to be selected is 18, thus the computational time required just to estimate all possible models would be $2^{18} \times 45$s (over 30000 hours), which is well beyond manageable.

Therefore, we propose to carry out the model selection via an MCMC approach, which was originally introduced by Qian and Field (2002) in the context of variable selection of logistic regression models. The proposed methodology can handle large candidate model set, in fact, our numerical experiences suggest that it takes less than two hours to find the best model under the setup mentioned above (with bootstrap sample size $B = 100$).

The key idea is to convert the models selection into a problem of random sample generation from a finite population. The finite population is defined as the set of all candidate models, on which a discrete probability distribution is induced from the CL-BIC, such that the model with a lower CL-BIC has a higher probability. The induced distribution does not need to have a closed form in order to carry out the MCMC method. The convergence of the MCMC method ensures that the selected model has the lowest IC among all candidate models, provided that the MCMC sample is sufficiently large.

Following notation of Qian and Field (2002), a model α can be denoted as a p-dimensional binary vector $\boldsymbol{v}_\alpha = (v_1, \ldots, v_p)$, where each entry is an indicator of whether a parameter is included in the model, i.e. $v_j = I(j \in \alpha), j = 1, \ldots, p$.

The goal is to generate a sample of candidate models (or binary vectors) from a distribution such that the model that minimizes the selection criterion (denoted as α_0) has the highest probability, and therefore appearing most frequently in the sample.

In particular, define a probability distribution as

$$P(\alpha) = \frac{\exp\left[-S(\alpha;Y)\right]}{\sum_{\alpha' \in \mathscr{A}} \exp\left[-S(\alpha';Y)\right]},$$

where $\mathscr{A}$ is the set of all possible candidate models and $S(\cdot)$ is a model selection criterion, which in this case is the CL-BIC specified in (4.4). It is easy to see from the expression that the smaller $S(\alpha;Y)$ is, the higher $P(\alpha)$ gets, which ensures α_0 has the highest probability.

But direct generation from $P(\alpha)$ is still computationally intractable when p is large, since the denominator of $P(\alpha)$ has $|\mathscr{A}| = 2^p$ terms. In this case, the Gibbs sampling is

adopted to avoid this computational difficulty, since instead of the full probability distribution $P(\alpha)$, all that is needed is conditional distributions

$$P(v_j|v_{-j}) = \frac{P(v_j, v_{-j})}{P(v_j = 0, v_{-j}) + P(v_j = 1, v_{-j})}, \\ = \frac{\exp\left[-S(v_j, v_{-j}; Y)\right]}{\exp\left[-S(v_j = 0, v_{-j}; Y)\right] + \exp\left[-S(v_j = 1, v_{-j}; Y)\right]}, \tag{4.7}$$

which cancels out the denominator of $P(\alpha)$. The conditional distribution is simply a Bernoulli distribution, where $v_{-j} = \{v_i, i \neq j,\ i = 1, \ldots, p\},\ j = 1, \ldots, p$.

A brief sketch of the Gibbs sampling procedure is summarized as follows, with pseudocode shown in Algorithm 2:

- Step1: Start with $\boldsymbol{v} = (v_1^{(0)}, \ldots, v_p^{(0)}) = (1, \ldots, 1)$ and $k = 0$;
- Step2: For $j = 1, \ldots, p$, generate $v_j^{(k)}$ from $P(v_j|v_1^{(k)}, \ldots, v_{j-1}^{(k)}, v_{j+1}^{(k-1)}, \ldots, v_p^{(k-1)})$;
- Step3: Repeat Step2 N times to generate a sequence of models: $\{\boldsymbol{v}^{(1)}, \ldots, \boldsymbol{v}^{(N)}\}$.

Apparently, Step2 (same as Line 4 in Algorithm 2) is the crucial step for model generation. We describe this step in detail by pseudo C++ code in Algorithm 3.

Algorithm 2 Main Model Selection Procedure

```
#include<map>
#include<vector>
```

Input: N: the number of models to be generated.
Output: Selected best model

1: Call function **ModelFit** (Algorithm 1) on the full model, i.e. $\boldsymbol{v} = (1, \ldots, 1)$.
double OldCrt = obtained CL-BIC

2: **Initializations:**
Crts: map<vector<int>, double>; **Crts**.insert(make_pair(v, *old_crt*));
▷ To keep criterion values for all evaluated models
Mods: map<vector<int>, int>;
▷ To save generated models and their frequencies

3: **for** count = 1 to N **do**
4: Call function **GenModel**(&**Crts**, &**Mods**, &$\boldsymbol{v}$, &OldCrt, ...) ▷ Override inputs.
5: **end for**

6: Swap the keys and values of **Mods**, and output the last 5 pairs.

4.2.5 Computational aspects

In particular, we keep record of all models evaluated in history to avoid repetitive computation. Records are kept in associative containers (i.e. `map`'s in our case) that are known to

be more efficient than sequential containers (e.g. `vector`'s) when performing lookup and retrieval tasks. The **Crts** map stores CL-BIC values of every model evaluated in the past, while the **Mods** map stores only the generated ones, thus, **Mods** is updated only N times but **Crts** could potentially be updated pN times. The elements of a `map` are pairs of `key` and `value`, where `key` plays a role similar to index. A `map` automatically sorts its `pair`'s according to their `key`'s in an increasing order, and search operations are carried out through the `key`'s. To our knowledge, the `key`'s of a `map` does not (currently) accept RcppArmadillo defined data types, therefore, instead of `arma::vec`, we use `std::vector` with binary elements for `key`'s in both **Crts** and **Mods**. The `values`'s of **Crts** are the criterion values and those for **Mods** are the model frequencies.

The function in Algorithm 3 does not return any value, instead it rewrites $\boldsymbol{\nu}$, **Crts** and **Mods** every time it is called. By doing this, we avoid copying the maps that might be large (depending on the situations), thus saving a lot of time. When the model generation is done, we swap the `key`'s and `value`'s in **Mods** into a `multimap` (since `map` requires unique `key`'s, but it is very likely that there are models with exactly the same frequencies), so that the frequency of each model is automatically ordered (in increasing order by default), from which the last k `pair`'s would correspond to the k models that appeared most frequently. The small template function for swapping `pair`'s in a `map` is not relevant to the model selection procedure concerned in this chapter, thus not shown here.

4.3 Simulation Studies

To study the performance of the Gibbs sampling methods in selecting significant parameters in the setting of high-dimensional count data, we conduct Monte Carlo simulation experiments under two model setups: the non-temporal setup, where we assume that $\boldsymbol{Y}_1,\ldots,\boldsymbol{Y}_T$ are independent and the temporal setup which corresponds to the copSTM model proposed in Chapter 3.

We set $n_{\mathscr{L}} = 10 \times 10$ for both settings, larger $n_{\mathscr{L}}$ is expected to lead to better results due to larger sample size, while our numerical results show that a 10×10 grid already produces satisfactory outcomes. However, it is worth noting that in the temporal setting, it is not recommended to set n smaller than 6. Because if the number of tiles is too small, most of the neighbourhood area would be overlapping. Recall that the covariates are computed out of response of the previous time point in neighbourhood tiles, this will lead to highly correlated covariates. For example, in an extreme case where $n = 2$, the neighbourhoods of all 4 tiles are exactly the same, which means the covariate vectors would be the same as well.

Algorithm 3 void Function to generate Model via Gibbs sampling (**GenModel**)

Input: **Crts**: A map with evaluated models and according criterion,
Mods: A map with generated models and the times that it has been generated,
$\boldsymbol{v}$: A binary vector specifying model to be evaluated,
OldCrt: CL-BIC value of the last generated model.
(Note: All the inputs above are passed by reference)

for pp = 1 to $p-1$ **do** ▷ Evaluating the ppth parameter (Always skip intercept)

$\boldsymbol{v}[pp]$ = 1 - $\boldsymbol{v}[pp]$;

auto iterv = **Crts**.find($\boldsymbol{v}$); ▷ look for $\boldsymbol{v}$ in the record
if iterv == **Crts**.end() **then** ▷ if $\boldsymbol{v}$ is not in the record
Call function **ModelFit** (Algorithm 1) on model $\boldsymbol{v}$ for CL-BIC: NewCrt
Crts.insert(make_pair($\boldsymbol{v}$, NewCrt)); ▷ Write new record
else
NewCrt = iterv→second; ▷ Extract CL-BIC from record
end if

if NewCrt is finite **then**
double s = exp(NewCrt - OldCrt);
double prob = v[pp] ? 1/(1 + s) : s/(1 + s); ▷ Conditional probability in (4.7)
int vj = R::rbinom(1, prob); ▷ Generate v_j
if vj == $v[pp]$ **then** OldCrt = NewCrt; **else** $\boldsymbol{v}[pp]$ = 1 - $\boldsymbol{v}[pp]$;
else
$\boldsymbol{v}[pp]$ = 1 - $\boldsymbol{v}[pp]$; ▷ Very bad model
end if
end for

++**Mods**[$\boldsymbol{v}$]; ▷ Record generated model

4.3.1 Non-temporal setting

In this setting, we assume that $\boldsymbol{Y}_1,\ldots,\boldsymbol{Y}_T$ are independent with no temporal dependence. Thus, the model can be seen as a "multivariate" Poisson regression model, where each $\boldsymbol{Y}_t = (Y_1,\ldots,Y_d)'$ has the correlation structure $\boldsymbol{\Sigma}$ specified in Section 4.2. Due the independence assumption, the estimation of d^* of the penalty term in CL-BIC does not need bootstrapping. Details are shown in (4.6) and (4.5). We consider cases of $n_{\mathscr{C}} = 2$ and 3 for different parameter dimensions, and $T = 100$ and 5 for different sample sizes. The regression coefficients are the same for all settings: $\boldsymbol{\beta} = (1, 0.8, 0, -0.6, 0, 0.4, 0, -0.2)'$. We set the correlation parameters to be:

		$\rho_0^{(1,2)}$	$\rho_0^{(1,3)}$	$\rho_0^{(2,3)}$	$\rho_1^{(1)}$	$\rho_1^{(2)}$	$\rho_1^{(3)}$	$\rho_1^{(1,2)}$	$\rho_1^{(1,3)}$	$\rho_1^{(2,3)}$
$n_{\mathscr{C}} = 2$	$\boldsymbol{\rho}_a$	0	–	–	-0.3	0.3	–	0	–	–
	$\boldsymbol{\rho}_b$	-0.3	–	–	0	0.3	–	0	–	–
$n_{\mathscr{C}} = 3$	$\boldsymbol{\rho}_a$	0	0	0	-0.6	0.3	-0.2	0	0	0
	$\boldsymbol{\rho}_b$	-0.6	0	0.3	-0.2	0	0	-0.1	0	0

For each value of $n_{\mathscr{C}}$, $\boldsymbol{\rho}_a$ naturally produces a positive definite block correlation matrix $\boldsymbol{\Sigma}$, while $\boldsymbol{\rho}_b$ does not. The working correlation matrix used in the case of $\boldsymbol{\rho}_b$ is the nearest positive definite matrix of the original block matrix, calculated by `nearPD` function in the R package `Matrix`. As a result, the zeros in the parameter setup may become non-zeros but very small numbers, these effects are considered to be not useful and not be used to compute the positive selection rates. This setup can help us evaluate the performance of the model selection procedure when parameters have little effect.

For each setting, 100 simulated data sets are generated. Table 4.1 and 4.2 summarize the performance of the model selection method with Poisson and Negative binomial marginals respectively, where we gradually increase the tuning parameter in the penalty term $\gamma = 0, 0.5, 1$. We report our results in the same kind of measurements as Gao and Song (2010): the positive selection rate (PSR) and the false discovery rate (FDR). Specifically, PSR denotes the ratio of identified significant parameters among all significant parameters, while FDR denotes that of the falsely identified significant parameters among all significant parameters.

For both marginals, we observe satisfactory selection performance with very little effects from the "noisy" correlation setup $\boldsymbol{\rho}_b$ in most cases. For the larger sample size case where $T = 100$, selection of regression coefficients has very good control of the FDR rate even for $\gamma = 0$. With increasing γ, both regression coefficients and correlation parameters show a steady decrease in FDR without disturbing the PSR rate, approaching (almost) perfection when $\gamma = 1$. For the smaller sample size case $T = 5$, the approximation of penalty in Cl-BIC is expected to be less stable, since T can be seen as the bootstrap sample size B in the estimation of $\boldsymbol{H}(\hat{\boldsymbol{\theta}})$ and $\boldsymbol{K}(\hat{\boldsymbol{\theta}})$. In this case, $\gamma = 0$ does not adequately control the FDR rate, yet $\gamma = 1$ seems too harsh, especially for correlation parameters

when $n_{\mathscr{C}} = 3$. It seems that $\gamma = 0.5$ results in a relatively good balance of sensitivity and selectivity in this case.

			$T = 100$					
			Coefficients			Correlations		
		$\gamma =$	0	0.5	1	0	0.5	1
$n_{\mathscr{C}} = 2$	$\boldsymbol{\rho}_a$	PSR	1.000	1.000	1.000	1.000	1.000	1.000
		FDR	0.017	0.000	0.000	0.340	0.115	0.010
	$\boldsymbol{\rho}_b$	PSR	1.000	1.000	1.000	1.000	1.000	1.000
		FDR	0.045	0.012	0.000	0.351	0.142	0.015
$n_{\mathscr{C}} = 3$	$\boldsymbol{\rho}_a$	PSR	1.000	1.000	1.000	1.000	1.000	1.000
		FDR	0.022	0.000	0.000	0.057	0.000	0.000
	$\boldsymbol{\rho}_b$	PSR	1.000	1.000	1.000	1.000	1.000	1.000
		FDR	0.029	0.007	0.005	0.145	0.000	0.000
			$T = 5$					
			Coefficients			Correlations		
		$\gamma =$	0	0.5	1	0	0.5	1
$n_{\mathscr{C}} = 2$	$\boldsymbol{\rho}_a$	PSR	1.000	1.000	1.000	1.000	1.000	0.945
		FDR	0.170	0.063	0.022	0.310	0.005	0.000
	$\boldsymbol{\rho}_b$	PSR	1.000	1.000	1.000	1.000	1.000	0.935
		FDR	0.202	0.072	0.038	0.320	0.010	0.000
$n_{\mathscr{C}} = 3$	$\boldsymbol{\rho}_a$	PSR	1.000	1.000	1.000	1.000	0.857	0.643
		FDR	0.145	0.087	0.038	0.302	0.015	0.000
	$\boldsymbol{\rho}_b$	PSR	1.000	1.000	1.000	0.931	0.523	0.503
		FDR	0.170	0.093	0.050	0.417	0.014	0.000

Table 4.1: Positive selection rates (PSR) and false discovery rates (FDR) on correlated Poisson regression model with different number of groups ($n_{\mathscr{C}}$), penalty tuning parameter (γ) and sample size (T).

Table 4.3 show computational time required for model selection on data sets of various sizes. The timings are carried out on an Apple iMac computer with a 2.7 GHz Intel Core i5 processor and 8 GB 1600 MHz DDR3 memory. Computational effort is required mainly in the number of candidate models evaluated in the model selection process. As a result, although sample size T does affect run time, it is not the predominating factor. In fact, when increase T from 5 to 100, computation time increases much less than 20 times, in most cases around four to five times. Besides, for the same sample size, increasing γ from 0 to 0.5 dramatically reduces required time (most obvious for $n_{\mathscr{C}} = 3$). This is all because larger sample size or heavier penalty makes the CL-BIC values more distinguished, making the optimal or selected model more "stand out" among other candidate models,

			$T = 100$					
			Coefficients			Correlations		
		$\gamma =$	0	0.5	1	0	0.5	1
$n_{\mathscr{C}} = 2$	$\boldsymbol{\rho}_a$	PSR	1.000	1.000	1.000	1.000	1.000	1.000
		FDR	0.038	0.005	0.000	0.322	0.142	0.000
	$\boldsymbol{\rho}_b$	PSR	1.000	1.000	1.000	1.000	1.000	1.000
		FDR	0.031	0.002	0.000	0.329	0.074	0.000
$n_{\mathscr{C}} = 3$	$\boldsymbol{\rho}_a$	PSR	1.000	1.000	1.000	1.000	1.000	1.000
		FDR	0.024	0.002	0.000	0.091	0.000	0.000
	$\boldsymbol{\rho}_b$	PSR	1.000	1.000	1.000	1.000	1.000	0.995
		FDR	0.020	0.002	0.000	0.132	0.000	0.000
			$T = 5$					
			Coefficients			Correlations		
		$\gamma =$	0	0.5	1	0	0.5	1
$n_{\mathscr{C}} = 2$	$\boldsymbol{\rho}_a$	PSR	1.000	1.000	1.000	1.000	0.995	0.924
		FDR	0.190	0.054	0.027	0.278	0.005	0.000
	$\boldsymbol{\rho}_b$	PSR	1.000	1.000	1.000	1.000	0.995	0.895
		FDR	0.197	0.076	0.034	0.285	0.010	0.000
$n_{\mathscr{C}} = 3$	$\boldsymbol{\rho}_a$	PSR	1.000	1.000	1.000	1.000	0.840	0.583
		FDR	0.180	0.076	0.022	0.408	0.004	0.000
	$\boldsymbol{\rho}_b$	PSR	1.000	1.000	1.000	0.929	0.518	0.503
		FDR	0.208	0.080	0.031	0.346	0.000	0.000

Table 4.2: Positive selection rates (PSR) and false discovery rates (FDR) on correlated Negative binomial regression model with different number of groups ($n_{\mathscr{C}}$), penalty tuning parameter (γ) and sample size (T).

thus fewer different models are generated or evaluated in the Gibbs sampling process.

4.3.2 Temporal setting

In this setting, model selection is carried out on the copSTM model proposed in Chapter 3. The correlation parameters $\boldsymbol{\rho}$ are set the same as Section 4.3.1 shown in Table 4.3.1. The regression coefficients $\boldsymbol{\beta}$, which are interpreted as the impacts on growth between groups, are collected in the $n_{\mathscr{C}} \times n_{\mathscr{C}}$ matrix

$$\mathscr{B} = \begin{pmatrix} 1 & 0 \\ 0.5 & 1 \end{pmatrix} \text{ for } n_{\mathscr{C}} = 2 \text{ and } \mathscr{B} = \begin{pmatrix} 0.8 & 0.3 & 0 \\ 0.5 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \text{ for } n_{\mathscr{C}} = 3,$$

		$T=100$			$T=5$		
	$\gamma=$	0	0.5	1	0	0.5	1
$n_{\mathscr{C}}=2$	$\boldsymbol{\rho}_a$	6 min	5 min	4 min	44 s	25 s	22 s
	$\boldsymbol{\rho}_b$	5 min	5 min	5 min	66 s	29 s	23 s
$n_{\mathscr{C}}=3$	$\boldsymbol{\rho}_a$	47 min	14 min	14 min	9 min	74 s	76 s
	$\boldsymbol{\rho}_b$	51 min	15 min	15 min	14 min	86 s	66 s

Table 4.3: Computational time to run model selection under the non-temporal setting with different number of groups ($n_{\mathscr{C}}$), penalty tuning parameter (γ) and sample size (T).

in which the entry in the cth row and c'th column corresponds to value of $\beta^{(c|c')}$. Recall from Chapter 2 that this is interpreted as the impact of group c' to group c. Intercepts are -1 for all groups in both settings. In this thesis, we always assume the model includes intercept terms. Thus the number of parameter to be selected (regression coefficients + correlation parameters) is $4+4=8$ for $n_{\mathscr{C}}=2$ and $9+9=18$ for $n_{\mathscr{C}}=3$.

The $\boldsymbol{K}(\boldsymbol{\theta})$ required for CL-BIC is estimated using the parametric bootstrap approach described in Section 4.2.3. The performance of model selection depends mainly on the size of the bootstrap samples B, instead of that of the original data set. Thus, in Table 4.4, we fix T at 10 and show selection results of the Poisson copSTM model with increasing B. Specifically we show PSR and FDR for $\boldsymbol{\beta}$ and $\boldsymbol{\rho}$ separately, as well as computation time. As expected, selection results are generally better for large B, however, on consideration of balancing between statistical and computational efficiency, we choose to use $B=500$ for the rest of this section.

	$B=$	100	200	500	1000
$n_{\mathscr{C}}=2$	PSR_β	1.000	1.000	1.000	1.000
	FDR_β	0.083	0.077	0.051	0.036
	PSR_ρ	1.000	1.000	1.000	1.000
	FDR_ρ	0.355	0.349	0.373	0.344
	Time (min)	18	39	67	138
$n_{\mathscr{C}}=3$	PSR_β	0.994	0.994	0.992	0.988
	FDR_β	0.193	0.168	0.134	0.090
	PSR_ρ	0.997	0.998	1.000	1.000
	FDR_ρ	0.600	0.570	0.545	0.510
	Time (min)	58	113	242	500

Table 4.4: Positive selection rates (PSR), false discovery rates (FDR) for $\boldsymbol{\beta}$ and $\boldsymbol{\rho}$ separately and computational time required for the copSTModel selection with different bootstrap sample sizes (B). Correlation parameters are the same as $\boldsymbol{\rho}_a$ shown in Table 4.3.1, $n, T=10$.

In Table 4.5 (for Poisson marginal) and 4.6 (for Negative binomial marginal), we show

PSR and DFR for $\boldsymbol{\beta}$ and $\boldsymbol{\rho}$ separately with different $n_{\mathscr{C}}$ and γ, while $T = 10$ and $B = 500$ are fixed. The selection of regression coefficients seems almost always easier than correlation parameters, with very low FDR for $n_{\mathscr{C}} = 2$ even when $\gamma = 0$, while correlation parameters require γ to be at least 0.5 for a satisfactory control of FDR. Besides, with this moderate number of parameters, the selection result is barely affected by the noise induced in $\boldsymbol{\rho}_b$. However, for $n_{\mathscr{C}} = 3$, where the number of parameters more than doubled compared to $n_{\mathscr{C}} = 2$, and with a higher dimension of Y_t, such impact becomes more obvious. While we still observe fairly good result for $\boldsymbol{\rho}_a$ ($\gamma = 0$ to 0.5), the noise in $\boldsymbol{\rho}_b$ seems so strong that it becomes hard to distinguish between a genuine zero correlation and a small non-zero value, as indicated by the PSR of the correlation parameters.

			Coefficients			Correlations		
		$\gamma =$	0	0.5	1	0	0.5	1
$n_{\mathscr{C}} = 2$	$\boldsymbol{\rho}_a$	PSR	1.000	1.000	1.000	1.000	1.000	0.990
		FDR	0.051	0.020	0.003	0.373	0.296	0.258
	$\boldsymbol{\rho}_b$	PSR	1.000	1.000	1.000	1.000	1.000	0.995
		FDR	0.051	0.013	0.003	0.399	0.283	0.260
$n_{\mathscr{C}} = 3$	$\boldsymbol{\rho}_a$	PSR	0.992	0.976	0.938	1.000	0.923	0.820
		FDR	0.134	0.024	0.006	0.545	0.490	0.479
	$\boldsymbol{\rho}_b$	PSR	0.976	0.945	0.886	0.698	0.542	0.488
		FDR	0.159	0.070	0.049	0.499	0.439	0.372

Table 4.5: Positive selection rates (PSR) and false discovery rates (FDR) for $\boldsymbol{\beta}$ and $\boldsymbol{\rho}$ separately under the temporal setting with Poisson marginals, with different number of groups ($n_{\mathscr{C}}$), setting of correlation parameters ($\boldsymbol{\rho}_a$ and $\boldsymbol{\rho}_b$) and penalty tuning parameter (γ). Data sets are generated on a 10×10 lattice, with $T = 10$ time points. Bootstrap sample size used is 500.

4.4 Real data analysis

To examine the performance of the proposed model selection method, we reanalyze data of the cancer cell-fibroblast co-culture experiment in Chapter 2 and 3. Recall that in this data, fibroblasts (F) are non-fluorescent whereas cancer cells fluoresce either in the red (R) or green (G) channels due to the experimental expression of mCherry or GFP proteins respectively. Each image was subsequently tiled using a 25×25 regular grid.

The data set include $n_{\mathscr{C}} = 3$ cell populations, thus corresponding to 9 regression coefficients $\boldsymbol{\beta}$ and 9 correlation parameters $\boldsymbol{\rho}$, the same as in our simulation studies with the temporal setting. We carry out the model selection by sampling models from the distribution specified in (4.7) where the model criterion $S(\cdot)$ being the CL-BIC in (4.4)

			Coefficients			Correlations		
		$\gamma =$	0	0.5	1	0	0.5	1
$n_{\mathscr{C}} = 2$	$\boldsymbol{\rho}_a$	PSR	1.000	1.000	1.000	0.975	0.960	0.980
		FDR	0.039	0.008	0.007	0.373	0.306	0.260
	$\boldsymbol{\rho}_b$	PSR	1.000	1.000	1.000	0.950	0.926	0.936
		FDR	0.056	0.019	0.000	0.347	0.323	0.293
$n_{\mathscr{C}} = 3$	$\boldsymbol{\rho}_a$	PSR	1.000	0.975	0.969	0.957	0.867	0.813
		FDR	0.114	0.035	0.006	0.560	0.492	0.490
	$\boldsymbol{\rho}_b$	PSR	0.975	0.969	0.857	0.656	0.550	0.407
		FDR	0.176	0.035	0.020	0.528	0.471	0.447

Table 4.6: Positive selection rates (PSR) and false discovery rates (FDR) for $\boldsymbol{\beta}$ and $\boldsymbol{\rho}$ separately under the temporal setting with Negative binomial marginals, with different number of groups ($n_{\mathscr{C}}$), setting of correlation parameters ($\boldsymbol{\rho}_a$ and $\boldsymbol{\rho}_b$) and penalty tuning parameter (γ). Data sets are generated on a 10×10 lattice, with $T = 10$ time points. Bootstrap sample size used is 500.

with tuning parameter $\gamma = 0.5$. Penalty terms are estimated based on 500 bootstrap samples. Table 4.7 shows the frequencies of candidate models generated via Gibbs sampling with runs $N = 500$ and 200, as well as according CL-BIC values. The relevant frequencies of sub-models seem quite stable for the two different N's, with the model in the first row standing out among all candidate models. The binary model expression indicates whether or not a parameter is selected in the model. Parameters are in the order: $\beta^{(G|G)}, \beta^{(G|R)}, \beta^{(G|F)}, \beta^{(R|G)}, \beta^{(R|R)}, \beta^{(R|F)}, \beta^{(F|G)}, \beta^{(F|R)}, \beta^{(F|F)}$, $\rho_1^{(G)}, \rho_1^{(R)}, \rho_1^{(F)}$, $\rho_0^{(G,R)}, \rho_0^{(G,F)}, \rho_0^{(R,F)}$, $\rho_1^{(G,R)}$, $\rho_1^{(G,F)}$, $\rho_1^{(R,F)}$. Table 4.4 shows estimates of parameters in the selected best model with estimated 95% confidence intervals in the parenthesis. Selection results agree with those in previous chapters. Specifically, positive impacts and correlation between Green and Red cancer cells (i.e. $\beta^{(G|R)}, \beta^{(R|G)}, \rho_1^{(G,R)}$), negative spatial correlation but no significant impact between cancer cells and Fibroblasts (i.e. $\rho_1^{(G,F)}, \rho_1^{(R,F)}$ and $\beta^{(c|F)}, \beta^{(F|c)}$, $c = R, G$). Thus confirming the correctness of the model selection procedure.

4.5 Discussions

Gaussian copula regression models naturally induce more parameters than their GLM siblings by taking into consideration not only for regression coefficients but also for correlation parameters. Thus in order to avoid an over-parametrized model and retain the most meaningful parameters, variable selection in applications of Gaussian copula models is an extremely necessary but relatively new topic.

model	$N = 500$	$N = 200$	CL-BIC
110 110 001 000 111 001	362	151	2441
110 110 001 110 111 011	63	25	2458
110 110 001 100 111 001	58	21	2461
110 110 001 000 110 001	11	3	2540
110 110 001 110 110 001	2	–	3083
110 110 001 000 001 001	1	–	4129
110 100 001 100 111 001	1	–	4965

Table 4.7: Frequencies of candidate models generated via Gibbs sampling with runs $N = 500$ and 200, as well as according CL-BIC value.

$c =$	G	R	F
$\widehat{\beta}_0^{(c)}$	-1.00 (-1.08, -0.91)	-0.60 (-0.67, -0.52)	-0.42 (-0.52, -0.31)
$\widehat{\beta}^{(G\|c)}$	1.46 (1.33, 1.59)	0.22 (0.12, 0.32)	–
$\widehat{\beta}^{(R\|c)}$	0.32 (0.21, 0.42)	1.26 (1.37, 1.67)	–
$\widehat{\beta}^{(F\|c)}$	–	–	1.09 (1.18, 1.34)
$\widehat{\rho_0}^{(G,c)}$	1	0.05(0.02, 0.08)	-0.03 (-0.04, -0.02)
$\widehat{\rho_0}^{(R,c)}$	–	1	-0.02 (-0.03, -0.01)
$\widehat{\rho_0}^{(F,c)}$	–	–	1
$\widehat{\rho_1}^{(G,c)}$	–	–	–
$\widehat{\rho_1}^{(R,c)}$	–	–	-0.02(-0.03, -0.01)
$\widehat{\rho_1}^{(F,c)}$	–	–	–

Table 4.8: Parameter estimates in the selected model via Gibbs sampler using the cancer cell growth data, with bootstrap 95% confidence intervals based on 500 bootstrap samples in parenthesis.

Due to the complication that already exists in fitting copula models, an exhaustive screening of all candidate models is apparently infeasible. Therefore, in this chapter, we propose to perform the model selection using the Gibbs sampling method. This method is originally introduced by Qian and Field (2002) in the context of logistic linear regression model. It provides a very effective and reliable approach that is especially appealing when dealing with large candidate model set. It guarantees to converge to the best model with the lowest criterion value without having to exhaustively search through all possible models. With a well developed information criterion, the CL-BIC (Gao and Song, 2010), we can obtain very good selection results in both temporal and non-temporal settings, and for both regression coefficients and correlation parameters within a reasonable amount of time.

Although in this chapter we only focus on selection of the copula-based model, the

proposed methodology is extremely flexible and can be applied to a wide variety of regression based models as an efficient variable subsetting toolkit. In the next chapter, we implement this selection method as an R package function for both the Gaussian copula models and standard log-linear regression models.

Chapter 5

copSTM: An R package for the analysis of spatio-temporal count lattice data with model selection tools

5.1 Introduction

Recently, there has been an increasing interest in models for spatio-temporal count data and a considerable number of publications on this subject has appeared in the literature, with different space-time structures depending on the goals of the analysis, see for example Rushworth et al. (2014), Bradley et al. (2015) and Quick et al. (2017).

Following Cox et al. (1981), temporal models can be loosely characterised as either parameter driven or observation driven. In parameter driven models, conditional expectation is modelled by a latent process, which cannot be observed directly and which evolves independently of the past and present values of the observed process. On the other hand, for observation-driven models, conditional expectation of the outcome depends explicitly on the past observations. While latent models require a computationally expensive Markov chain Monte Carlo (MCMC) algorithm, observation-driven models can be easily fitted with likelihood-based methods and are straightforward to apply to a rich toolkit available for this class of models, for example, variable selection. In this chapter, we consider a GARMA (generalised autoregressive moving average) subclass of observation driven models, first introduced by Benjamin et al. (2003). The CRAN archive contains several R (R Core Team 2017) packages devoted to univariate observation-driven time series models that can handle count data, including **glarma** (Dunsmuir et al., 2015), **gamlss** (Rigby and Stasinopoulos, 2005) and **tscount** (Liboschik et al., 2017).

On top of temporal dependency, we extend the GARMA model to handle multivariate spatio-temporal data by capturing also the spatial as well as cross variable correlations. Specifically, we consider a Gaussian copula model (Gao and Song, 2010) to extend the

univariate temporal regression models, with the merit that the specification of the regression model is separate from the dependence structure, which is allowed to be very flexible with both positive and negative correlations. Many recent R packages implement Gaussian copula models on spatially correlated data, although only a few consider copulas for regression modelling: the **gcmr** (Masarotto et al., 2017) for general copula regression models with a few specified correlation structures, **gcKrig** (Han and De Oliveira, 2018) for geo-statistical data and **copCAR** (Goren and Hughes, 2017) for areal data. However, packages for spatio-temporal modelling is much less well developed and mostly are latent process models that require MCMC methods, for example **spBayes** (Finley et al., 2015), **spTimer** (Bakar et al., 2015) and **CARBayes** (Lee et al., 2018).

We review in more detail the relevant package functions and the corresponding model classes in Section 5.5 and compare them to **copSTM**. Because all packages have different focus and specialisations, there is no package fitting exactly the same kind of model as ours. But it is possible to compare some special cases of our model with those from other packages. We show that our package functions reach very similar results but within a shorter running time, thus confirming the reliability and efficiency of our package. For the same reason, we provide some features that are not offered by other packages and vice vera, for example, **tscount** provides moving average parameters while we consider only autoregressive terms, **gcKrig** provides spatial prediction that our package does not. On the other hand, we allow the spatio-temporal data to be multivariate, and we provide model selection tools, which are not implemented in the above mentioned packages.

In addition, to model fitting, **copSTM** also provides functions for fast model selection of the copSTM model as well as standard log-linear regression models. Many R packages have been created in the past years to carry out automated model selection. The function `step` now available in the builtin package **stats** (Venables and Ripley, 2013) uses a stepwise selection procedure, which nonetheless suffers from the convergence problem of stepwise methods. **leaps** (Lumley and Miller, 2009) uses a branch-and-bound algorithm for quicker search of the best subsets but can only handle linear regressions. **bestglm** (McLeod and Xu, 2010) takes advantage of **leaps**, however due to its dependence of **leaps**, the optimization on search is still limited to the Gaussian case, for non-Gaussian GLM, a simple exhaustive enumeration approach is used. **subselect** (Cerdeira et al., 2009) dose variable selection under the context of principle component and **glmulti** (Calcagno et al., 2010) implement genetic algorithms and is built for the purpose of fast computation. In later sections, we compare one of our model selection functions for log-linear regression models with the main function in **glmulti**.

The **copSTM** package is not currently on the Comprehensive R Archive Network, but is available for download from Github using the package **devtools** (Wickham and Chang, 2016) by typing command `devtools::install_github("pqiao29/copSTM")`. It carries out tasks of scientific interest for the analysis of spatio-temporal grouped count data

on lattice with Gaussian copula models. First, the package computes maximum composite likelihood estimation and standard error for the model parameters of three model specifications: (i) a simpler temporal model that does not address spatial correlations (proposed in Chapter 2); (ii) a spatial Gaussian copula model with external explanatory variables (discussed in Chapter 4); (iii) a spatio-temporal model combining (i) and (ii) (proposed in Chapter 3). Second, it provides automated model selection tools for the three models mentioned above, as well as the general log-linear regression models. Finally, we offer a Web application for visualization of the data, parameter estimates and selection results on the temporal model considered in Chapter 2, which is appealing for an exploratory analysis especially for non-R users.

The models available in **copSTM** can be fitted to Poisson and Negative binomial data, Section 5.2 summarises the estimation and selection of models that are implemented. Section 5.3 provides an overview of the package and its functionality. Section 5.4 illustrates the use of the package by reproducing some of the simulation and real data results presented in Chapter 2, 3 and 4. Section 5.5 reviews other R packages which fit either temporal or Gaussian copula model and compare them with our package. Finally, Section 5.6 discusses the limitations of our package and gives an outlook on possible future extensions. In the Appendix we give a step by step instruction of the usage of the Web application.

5.2 Gaussian copula Spatio-temporal model

Let $\boldsymbol{Y} = (\boldsymbol{Y}_1, \ldots \boldsymbol{Y}_T)$ be a grouped count data set observed on a regular $n_{\mathscr{L}} = n \times n$ spatial lattice at T consecutive time points. Observation in each location/tile is a $n_{\mathscr{C}}$-dimensional vector corresponding to counts of $n_{\mathscr{C}}$ groups/clusters. Denote i, t and c as indices for tile, time point and group respectively. Thus, $\boldsymbol{Y}_t$ is written as a d-dimensional vector $(Y_{1,t}^{(1)}$, $\ldots$, $Y_{1,t}^{(n_{\mathscr{C}})}$, $Y_{2,t}^{(1)}$, $\ldots$, $Y_{2,t}^{(n_{\mathscr{C}})}$, $\ldots$, $Y_{n_{\mathscr{L}},t}^{(1)}$, $\ldots$, $Y_{n_{\mathscr{L}},t}^{(n_{\mathscr{C}})})'$, where $d = n_{\mathscr{C}} n_{\mathscr{L}}$. We assume that the marginal distribution functions of $Y_{i,t}^{(c)}$ is parameterized in terms of its expected value $\mathrm{E}(Y_{i,t}^{(c)}) = \mu_{i,t}^{(c)}$ which depends on a vector of explanatory variables through the relationship $g(\mu_{i,t}^{(c)}) = \boldsymbol{X}_{i,t}^{(c)\prime}\boldsymbol{\beta}$, for a suitable link function $g(\cdot)$ and a p_β-dimensional vector of regression coefficients $\boldsymbol{\beta}$. This setting encompasses a variety of popular model classes, for example the GLMs. In this chapter, we focus on Poisson and Negative binomial marginals with the log link function.

For temporal data, we let $\boldsymbol{X}_{i,t}^{(c)}$ take the form of an autoregressive term that depends on the past observations of Y_t in neighbouring locations. Specifically,

$$\log(\mu_{i,t}^{(c)}) = \beta_0^{(c)} + \sum_{c'=1}^{n_{\mathscr{C}}} S_{i,t-1}^{(c')} \beta^{(c|c')} \quad \text{and} \quad S_{i,t-1}^{(c')} = \frac{1}{n_i} \sum_{i \sim j} \log\left(1 + Y_{j,t-1}^{(c')}\right), \tag{5.1}$$

where $\sim$ denotes neighbouring tiles and n_i denotes the number of neighbouring tiles of tile i. In this case, the regression coefficients is specified as $\boldsymbol{\beta} = (\beta_0^{(1)}, \beta^{(1|1)}, \beta^{(1|2)}, \ldots, \beta_0^{(2)}, \beta^{(2|1)}, \ldots, \beta^{(n_{\mathscr{C}}|n_{\mathscr{C}})})'$ and thus $p_\beta = n_{\mathscr{C}}(n_{\mathscr{C}}+1)$. This marginal model falls into a more general class of observation-driven time series models, the generalised autoregressive moving average model, GARMA(p,q) (Benjamin et al., 2003) with $p = 1, q = 0$.

Let $\boldsymbol{\Sigma}$ denote the $d \times d$ correlation matrix of $\boldsymbol{Y}_t$ identical at different time points. We assume the correlation matrix to be a block adjacency matrix

$$\boldsymbol{\Sigma} = \begin{pmatrix} \boldsymbol{R}_0 & \boldsymbol{R}_1 & \mathbf{0} & \mathbf{0} & \ldots & \boldsymbol{R}_1 & \mathbf{0} & \ldots & \ldots \\ \boldsymbol{R}_1 & \boldsymbol{R}_0 & \boldsymbol{R}_1 & \mathbf{0} & \ldots & \ldots & \boldsymbol{R}_1 & \ldots & \ldots \\ \mathbf{0} & \boldsymbol{R}_1 & \boldsymbol{R}_0 & \boldsymbol{R}_1 & \ldots & \ldots & \ldots & \ldots & \ldots \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \boldsymbol{R}_0 \end{pmatrix}_{n_{\mathscr{L}} \times n_{\mathscr{L}}},$$

where each entry is a $n_{\mathscr{C}} \times n_{\mathscr{C}}$ matrix representing cross group/cluster correlation in the same tile $\boldsymbol{R}_0$ and neighbouring tiles $\boldsymbol{R}_1$ parametrized as

$$\boldsymbol{R}_0 = \begin{pmatrix} 1 & \rho_0^{(1,2)} & \rho_0^{(1,3)} & \ldots & \rho_0^{(1,n_{\mathscr{C}})} \\ \rho_0^{(1,2)} & 1 & \rho_0^{(2,3)} & \ldots & \rho_0^{(2,n_{\mathscr{C}})} \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ \rho_0^{(1,n_{\mathscr{C}})} & \ldots & \ldots & \ldots & 1 \end{pmatrix}, \quad \boldsymbol{R}_1 = \begin{pmatrix} \rho_1^{(1)} & \rho_1^{(1,2)} & \rho_1^{(1,3)} & \ldots & \rho_1^{(1,n_{\mathscr{C}})} \\ \rho_1^{(1,2)} & \rho_1^{(2)} & \rho_1^{(2,3)} & \ldots & \rho_1^{(2,n_{\mathscr{C}})} \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ \rho_1^{(1,n_{\mathscr{C}})} & \ldots & \ldots & \ldots & \rho_1^{(n_{\mathscr{C}})} \end{pmatrix}.$$

Note that it is possible that $\boldsymbol{\Sigma}$ is not positive definite, in this case, the working correlation matrix is taken as the nearest positive definite matrix of $\boldsymbol{\Sigma}$, computed with code modified from function `nearPD` in R package **Matrix** (Bates and Maechler, 2019).

Marginal distributions of $Y_{i,t}^{(c)}$ and dependence structure $\boldsymbol{\Sigma}$ are combined via a Gaussian copula (Joe, 2014), and the joint data cumulative distribution function is given by

$$\boldsymbol{Y}_t | \boldsymbol{Y}_{t-1} = \left(F^{-1}_{\mu_{1,t}^{(1)}}\left[\Phi(Z_1)\right], \ldots, F^{-1}_{\mu_{1,t}^{(n_{\mathscr{C}})}}\left[\Phi(Z_{n_{\mathscr{C}}})\right], F^{-1}_{\mu_{2,t}^{(1)}}\left[\Phi(Z_{n_{\mathscr{C}}+1})\right], \ldots, F^{-1}_{\mu_{n_{\mathscr{L}},t}^{(n_{\mathscr{C}})}}\left[\Phi(Z_d)\right] \right), \tag{5.2}$$

where $\boldsymbol{Z} = (Z_1, \ldots, Z_d)$ has a d-dimensional multivariate standard normal cumulative distribution with correlation matrix $\boldsymbol{\Sigma}$, Φ denotes the univariate standard normal cumulative distribution function and $F^{-1}_{\mu_{i,t}^{(c)}}(u) = \inf\{y : F_{\mu_{i,t}^{(c)}}(y) \leq u\}$ for $0 \leq u \leq 1$ with $\mu_{i,t}^{(c)}$ being the expectation of $Y_{i,t}^{(c)}$.

5.2.1 Parameter estimation and likelihood inference

The **copSTM** package implements maximum pairwise composite likelihood for the Gaussian copula regression models. Denote $\boldsymbol{\theta} = (\boldsymbol{\beta}, \boldsymbol{\rho})'$ as the p-dimensional vector of all

model parameters, where $\boldsymbol{\rho}$ is the vector of correlation parameters in $\boldsymbol{\Sigma}$ with dimension p_ρ and $p = p_\beta + p_\rho$. The pairwise composite likelihood function is

$$cl(\boldsymbol{\theta};\boldsymbol{y}) = \sum_{t=1}^{T}\sum_{i_1,i_2=1}^{n_{\mathscr{L}}}\sum_{c_1,c_2=1}^{n_{\mathscr{C}}} \log\left[\int_{a_{i_1,t}^{(c_1)}}^{b_{i_1,t}^{(c_1)}}\int_{a_{i_2,t}^{(c_2)}}^{b_{i_2,t}^{(c_2)}} \phi_\rho(z_1,z_2)dz_1dz_2\right],$$

where $\phi_\rho(z_1,z_2)$ denotes a 2-dimensional multivariate standard normal density function with correlation ρ, which corresponds to the correlation between $Y_{i_1,t}^{(c_1)}$ and $Y_{i_2,t}^{(c_2)}$. For computational concern, implemented package functions compute only those component likelihood pairs that involve observations in neighbouring tiles. Specifically, let Ω be a collection of pairwise index subsets $\{s=(i_1,i_2,c_1,c_2): c_1,c_2=1,\ldots,n_{\mathscr{C}}, i_1,i_2=1,\ldots,n_{\mathscr{L}}, i_1 \sim i_2\}$, and denote $y_{s,t}$ as the pair of observations $(y_{i_1,t}^{(c_1)}, y_{i_2,t}^{(c_2)})$, then the composite log-likelihood is written as

$$cl(\boldsymbol{\theta};\boldsymbol{y}) = \sum_{t=1}^{T}\sum_{s\in\Omega} cl(\hat{\boldsymbol{\theta}};y_{s,t}). \tag{5.3}$$

Optimization of (5.3) is done through the Fish-Scoring algorithm. Source code for the approximation of the bivariate normal cummulative function required for (5.3) is modified from the recursive method proposed by Meyer (2010).

5.2.2 Variable selection

The **copSTM** package also offers information criterion-based model selection tools for all models implemented. We take the Bayes information criterion (BIC) in the composite likelihood framework developed by Gao and Song (2010), specified as

$$\text{CL-BIC}(\hat{\boldsymbol{\theta}};\boldsymbol{y}) = -2cl(\hat{\boldsymbol{\theta}};\boldsymbol{y}) + (\log(T) + 2\gamma\log(p))\,d^*, \tag{5.4}$$

where γ is a tuning parameter for enforcing sparsity and $d^* = \text{trace}[\boldsymbol{H}^{-1}(\hat{\boldsymbol{\theta}};\boldsymbol{y})\boldsymbol{K}(\hat{\boldsymbol{\theta}};\boldsymbol{y})]$, $\boldsymbol{H}(\boldsymbol{\theta};\boldsymbol{y})$ and $\boldsymbol{K}(\boldsymbol{\theta};\boldsymbol{y})$ denote negative Hessian matrix and the variance of the first derivative of the composite log-likelihood in (5.3) respectively:

$$\boldsymbol{H}(\boldsymbol{\theta};\boldsymbol{y}) = -\mathbf{E}\left[\frac{\partial^2 cl(\boldsymbol{\theta};\boldsymbol{y})}{\partial\boldsymbol{\theta}^2}\right],\quad \boldsymbol{K}(\boldsymbol{\theta};\boldsymbol{y}) = \mathbf{Var}\left[\frac{\partial cl(\boldsymbol{\theta};\boldsymbol{y})}{\partial\boldsymbol{\theta}}\right].$$

Both matrices could be estimated using a parametric bootstrap method, see Chapter 4 for details.

The search through candidate models is carried out via an MCMC approach originally introduced by Qian and Field (2002). Following notation of Qian and Field (2002), define

a probability distribution as

$$P(\alpha) = \frac{\exp\left[-\text{CL-BIC}(\alpha;\boldsymbol{y})\right]}{\sum_{\alpha' \in \mathscr{A}} \exp\left[-\text{CL-BIC}(\alpha';\boldsymbol{y})\right]},$$

where α denote a sub-model of interest or equivalently a sub-vector of $\boldsymbol{\theta}$ with only selected variables, $\mathscr{A}$ is the set of all possible candidate models, and CL-BIC is the model selection criterion specified in (5.4). It is easy to see from the expression that the model with the smallest CL-BIC value has the highest probability. Thus, by generating a sufficiently large number of models from this distribution, one can easily identify the model that appears most frequently as the best model (with the specified criterion). In our implementations, we always assume intercepts and the overdispersion parameter for negative binomial response are in the selected model. The procedure for generating candidate models is through a Gibbs sampler, described in detail in Chapter 4.

5.3 Package functionality

In Table 5.1, the first column shows a list of exported functions in **copSTM**. The coloured cells in the table indicate the functionality of each function, in particular, three major features are covered: (i) Data organization: Organizes raw data into the form required as input for model fitting and selection functions; (ii) full Model estimation: Fit the Gaussian copula spatio-temporal model; (iii) Model selection: Select the model with the lowest criterion. The rows with blue/purple cells correspond to functions for the independent models, while those with pink cells are for the Gaussian copula models. Functions starting with "`idp`" fits a special case of the proposed model, where responses are considered independent i.e. the correlation matrix $\boldsymbol{\Sigma}$ is an identity matrix. If only regression coefficients are of interest, these functions are much faster than those starting with "`cop`".

Functions	Data Organization	full Model Estimation	Model Selection
`idpSTM()`			
`idpSTMSelect()`			
`logGLMselect()`			
`sim_data()`			
`make_data()`			
`copSTM()`			
`copSTMSelect()`			

Table 5.1: List of exported functions in package **copSTM**, with their functionalities indicated with coloured cells. Blue/purple correspond to functions for independent models while pink ones are those for Gaussian copula models.

The function `make_data(data, n)` takes an integer `n` (the number of tiles is $n \times n$)

and a matrix consisting of four columns in the order of time points, spatial coordinate (x and y) and group, the first and last column need to be consecutive integers starting from 0 and 1 respectively, the coordinate columns can take values of any real numbers, see the illustrative data in the package as an example: `data("cell_growth_data")`. The function tiles the data according to the coordinate information and organizes it as a response vector and a covariate matrix with the temporal relationship specified in (5.1) that can be passed to the functions for model estimation and selection.

5.3.1 Model fitting

The core function for parameter estimation is `copSTM()` that provides maximum composite likelihood estimators of the Gaussian copula models:

```
copSTM(x, y, K, n, marginal, cor_type = "both", temporal = TRUE,
  maxit = 100, eps = 0.1, std_err = FALSE, B = 100, Message_prog = FALSE)
```

The argument `y` is a vector containing the response counts, `x` is a matrix containing covariates, `K` and `n` represent $n_{\mathscr{C}}$ and $n = \sqrt{n_{\mathscr{L}}}$ respectively. `marginal` specifies marginal distribution of the counts: `"pois"` for Poisson and `"nbinom"` for negative binomial. Argument `cor_type` specifies the correlation structure $\boldsymbol{\Sigma}$. Users may choose to ignore some correlation parameters for fast computation, if `cor_type = "sp"`, only spatial correlation is taken into account, which means the model only estimates $\rho_1^{(1)}, \ldots, \rho_1^{(n_{\mathscr{C}})}$; if `cor_type = "mv"`, the function consider only the cross group correlation at the same location, i.e. only parameter in $\boldsymbol{R}_0$ are computed while $\boldsymbol{R}_1$ is taken as a zero matrix; if `cor_type = "both"`, all correlation parameters are estimated and if `cor_type = "ind"`, the model is treated as independent and obtain the same results with `idpSTM()` but slightly slower. The logical argument `temporal` tells the function whether the input data should be treated as temporal data or $\boldsymbol{Y}_t$'s are independent. If `temporal = FALSE`, the model becomes a spatial only Gaussian copula regression model where the covariate matrix is allowed to be anything, however if `temporal = TRUE`, covariates at time point t are assumed to be formed by response at time $t-1$ specified in (5.1). The required covariates as well as response can be obtained either by the function `make_data()`, which organizes raw data into the required form for `copSTM`, or by function `sim_data()`, which simulate datasets with given model. Standard errors are not returned by default, because in the temporal setting, estimation of standard errors requires bootstrapping, which is a very time consuming procedure compared to parameter estimation. If standard errors are required, the argument `std_err` needs to set to be `TRUE`, and if `temporal` is also `TRUE`, the user needs to choose the bootstrap sample size `B` (by default 100). In this case, it is recommended to set `Message_prog` to be `TRUE` as well, which prints messages informing the progress of the bootstrap procedure, making the wait less boring. The last two arguments `maxit` and `eps`

are tuning parameters controlling the convergence criterion of the likelihood optimization process, of which the default values should be sufficient in most cases. The function `copSTM()` returns a list of pairwise composite log-likelihood, parameter estimates and standard error (if required).

Function `idpSTM(data, n, marginal, maxit = 50, fit = FALSE)` can be seen as a `glm` wrapper, which combines `make_data()` with the fitting of log-linear regression models for convenient use. Except that `idpSTM()` uses maximum likelihood estimation while the standard `glm()` in package **stats** adopts the least square estimation, coefficient estimates are mostly the same: always `TRUE` for `all.equal()`, sometimes for `identical()`. This function as well as its model selection tools are implemented as a more user-friendly online interface, a Shiny application, where users can visualize their data, estimate model parameters and perform model selection without having R on the local computer, all that is needed is a browser. See the appendix of this chapter for a detailed instruction and web link.

5.3.2 Variable selection

The package offers three models selection functions for different model assumptions:

- `copSTMSelect(x, y, K, n, temporal, cor_type, marginal = "pois", ModelCnt = 100, B = 0, maxit1 = 50, maxit2 = 10, add_penalty = 0, Message_prog = TRUE, Message_res = TRUE, eps = 0.1)`

  The `copSTMSelect()` does model selection on the model fitted by `copSTM()`. Apart from arguments needed for `copSTM()`, it also takes arguments: `add_penalty`, which is γ in (5.4); and `ModelCnt`, which is the number of models to be generated via the Gibbs sampling, same as N in Section 5.2.

- `logGLMselect(y, x, marginal, maxit = 50, skip = NULL, ModelCnt = 100, Message = T)`

  The `logGLMselect()` does variable selection on log-linear regression models based on traditional BIC. It allows user to force some parameters in the selected model by specifying the indices of those parameters in the argument `skip`. If not specified, only the intercept and overdispersion parameter (if `marginal = nbinom`) are skipped.

- `idpSTMSelect(data, n, marginal, ModelCnt = 100, maxit = 50, Message = F)`

  The `idpSTMSelect()` does selection on the model fitted by `idpSTM()`. This function is also included in the Shiny application for accessibility for non-R users.

All model selection functions return (composite) log-likelihood, a binary vector indicating the selected model and estimated parameters and their standard errors in the selected model. Besides, all selection functions print the top 5 most frequently generated models with the according frequencies, if the argument `Message_res` is `TRUE`. This helps users to decide whether the Gibbs sampling procedure has converged.

5.4 Usage and examples

Most numerical results in this thesis are reproducible with the **copSTM** package. We show the usage of the package functions by "reproducing" some of the simulation and real data results in previous chapters.

5.4.1 Model fitting

The first example show estimation of model parameters on a simulated data set. The following code simulates and estimates a spatio-temporal data from the Gaussian copula model, with Poisson marginals ($n_{\mathscr{C}} = 3$, $n_{\mathscr{L}} = 25 \times 25$ and $T = 10$).

```
true_beta <- c(-0.8, 1, 0.5, -0.6, -0.5, 0.4, 1, -0.2, -0.3, 0.2, 0.1, 1)
true_rho <- c(0,  0,  0, -0.6,  0.2, -0.3, 0,  0, 0)
sim_dat <- sim_data(n = 25, K = 3, temporal = TRUE, t_size = 10,
                    marginal = "pois", true_beta, true_rho,
                    cor_type = "both")
res <- copSTM(sim_dat$covariates, sim_dat$response, K = 3, n = 25,
              marginal = "pois", cor_type = "both", temporal = TRUE,
              std_err = FALSE)$coefficients
```

This gives the absolute values of estimated bias ($\times 10^2$) of the composite likelihood estimators for the Gaussian copula model, presented in Table 3.1 in Chapter 3.

```
est <- c(rbind(res$intercept, res$main_effects), res$correlations)
round(abs(est - c(true_beta, true_rho))*100, 2)
[1] 6.94 4.02 6.72 0.11 5.17 1.38 0.82 2.13 2.74 0.25 0.97 0.62 1.44
[14] 0.07 0.07 1.97 0.10 1.39 0.57 0.22 0.67
```

If not all correlation parameters are of interest, the running time can be dramatically reduced by changing the argument `cor_type`:

```
f <- function(cor_type){
  invisible(copSTM(sim_dat$covariates, sim_dat$response, K = 3, n = 10,
         marginal = "pois", cor_type = cor_type, temporal = TRUE,
```

```
         std_err = FALSE))}
> library(microbenchmark)
> microbenchmark(f("both"), f("sp"), f("mv"), times = 10)
Unit: milliseconds
      expr       min        lq      mean   median        uq       max neval
 f("both") 5904.1930 5909.0389 6091.6362 6079.720 6128.4491 6487.7348    10
   f("sp") 1456.4515 1457.8807 1491.8328 1458.905 1481.4369 1662.0524    10
   f("mv")  879.4631  881.0292  886.9255  882.310  887.0951  917.7225    10
```

Next, we show an example with the real data, the cell growth data analyzed in this thesis. The code below loads the data and estimate the regression coefficients in the independent model.

```
data("cell_growth_data", package = "copSTM")
res <- idpSTM(cell_growth_data, n = 25, marginal = "pois")
```

This provides parameter estimates and according standard errors used for calculating the 95%confidence intervals shown in Table 2.4.1 (full model) in Chapter 2. Slight variation with the exact numbers in Table 2.4.1 is expected due to different computational techniques.

```
> res$coefficients
$intercept
          [,1]       [,2]       [,3]
[1,] -1.066618 -0.6053747 -0.4159173

$main_effects
          [,1]        [,2]       [,3]
[1,] 1.4366726 0.225782854 0.06485456
[2,] 0.3156149 1.258519114 0.02417806
[3,] 0.1024003 0.006440251 1.11977708

> res$standard_error
$intercept
[1] 0.05839237 0.04715967 0.04511669

$main_effects
           [,1]       [,2]       [,3]
[1,] 0.05468199 0.04574958 0.04450948
[2,] 0.04756699 0.03833244 0.03714000
[3,] 0.05180622 0.04251341 0.03951879
```

Finally, we fit the cell growth data into the Gaussian copula model to get a full collection of both regression coefficients and correlation parameters:

```
dat <- make_data(cell_growth_data, n =25)
K <- dat$K
res <- copSTM(dat$covariates, dat$response, dat$K,
              n =25, marginal = "pois", cor_type = "both",
              temporal = TRUE, std_err = FALSE)$coefficients
```

The estimation takes about 40 seconds. If standard errors are required, set `std_err = TRUE` and argument `B` (bootstrap sample size). This reproduces results shown in Table 3.6 in Chapter 3.

```
# Regression coefficients
> res$intercept
[1] -1.0608978 -0.6003824 -0.4209282

> res$main_effects
            [,1]        [,2]       [,3]
[1,] 1.45191195 0.220123781 0.06271111
[2,] 0.31450152 1.257046444 0.05245478
[3,] 0.08080938 0.004787924 1.08122867

# Correlation parameters
R0 <- matrix(NA, K, K)
R1 <- matrix(NA, K, K)
diag(R0) <- 1
diag(R1) <- res$correlations[1:K]
ind_r = K + 1
for(i in 1:(K - 1)){
  for(j in (i + 1):K){
    R0[i, j] <- res$correlations[ind_r]
    ind_r = ind_r + 1
  }
}
for(i in 1:(K - 1)){
  for(j in (i + 1):K){
    R1[i, j] <- res$correlations[ind_r]
    ind_r = ind_r + 1
  }
```

```
}

## correlation in the same tile:
print(round(R0, 3))
     [,1]  [,2]   [,3]
[1,]    1 0.048 -0.025
[2,]   NA 1 -0.016
[3,]   NA    NA  1
## correlation in neighbouring tiles:
print(round(R1, 3))
       [,1]   [,2]   [,3]
[1,] -0.012  0.007 -0.009
[2,]     NA -0.003 -0.017
[3,]     NA     NA  0.011
```

5.4.2 Variable selection

In the first example, we reproduce the model selection for the independent model on the cell growth data with BIC as criterion. This reproduces the result in Table 2.4.1 (BIC model) in Chapter 2. Without consideration of correlations, the model selection runs very fast, taking only about 0.2 seconds.

```
res <- idpSTMSelect(cell_growth_data, n = 25, marginal = "pois",
                    ModelCnt = 200)
> res$coefficients
$intercept
[1] -0.9811441 -0.6000859 -0.3726855

$main_effects
          [,1]      [,2]     [,3]
[1,] 1.4525131 0.2264492 0.000000
[2,] 0.3157527 1.2588421 0.000000
[3,] 0.0000000 0.0000000 1.133632
```

Model selection for the Gaussian copula model can be very time consuming, depending on the choice of `n` (number of tiles), `K` (the number of groups) and `B` (bootstrap sample size). Thus we show a toy example on simulated data with 2 groups, 10×10 lattice and `B = 20`. The following code takes about 1min 34 sec. Selection performance can be improved by increasing `B` and maybe adjusting `add_penalty`.

```
## Simulate data
```

```
set.seed(444)
true_beta <- c(-1, 1, 0.5, -0.5, 0, 1)
true_rho <- c(-0.3, 0, 0.3, 0)
sim_dat <- sim_data(n = 10, K  = 2, t_size = 10, true_beta,
                    marginal = "pois", temporal = TRUE,
                    rho = true_rho, cor_type = "both")
## Model selection
res <- copSTMSelect(sim_dat$covariates, sim_dat$response, K = 2,
                    n = 10, marginal = "pois", temporal = TRUE,
                    cor_type = "both", ModelCnt = 500, B = 20,
                    add_penalty = 0.5, Message_prog = FALSE,
                    Message_res = TRUE)
Model 1111011110 appeared 499 times
Model 1111011010 appeared 1 times

> res$selected_model                          # selected model
 [1] 1 1 1 1 0 1 1 1 1 0
> as.numeric(as.logical(c(true_beta, true_rho)))  # true model
 [1] 1 1 1 1 0 1 1 0 1 0
```

5.5 Comparison with other packages

In this section, we provide a comparison with other R packages which can be employed for count spatio-temporal data analysis. We consider a large number of somehow related packages which makes this comparison quite extensive yet interesting for readers who want some guidance on choosing the most appropriate package for their data. Since all packages focus on different aspacts, we are only able to compare some special cases of our model with each one of them. Apart from packages that focus on modelling, we also compare separately our model selection function with another model selection package. In the following subsections, we discuss in detail how these packages differ from our package **copSTM**.

5.5.1 For independent data ($n = n_{\mathscr{C}} = 1$)

For the special case where there is only one location and one group, each time point has only one observation, the copSTM model becomes a simple univariate AR(1) time series model, or a standard GLM. We compare this special case with the following three packages:

- The `glm` function in package **stats** and the `glm.nb` function in **MASS** (for negative binomial distribution) (Venables and Ripley, 2013) fit standard GLMs with the iteratively reweighted least squares algorithm.
- The **gamlss** package implement the generalised additive models for location, scale and shape (GAMLSS) introduced by Rigby and Stasinopoulos (2005), as an extension of the generalised additive model. Apart from other parameters, the overdispersion coefficient of the negative binomial distribution changes with time: $\alpha_t = \exp(\beta_0 + \beta_1 \log(Y_{t-1}) + 1)$.
- The **tscount** package by Liboschik et al. (2017) consider a spacial case of the GARMA(p, q) (Benjamin et al., 2003) on count time series data, and provide methods of parameter estimation, model assessment and intervention analysis.

For comparison, we use the campylobacter infection data that contains the number of campylobacterosis cases (reported every 28 days) in the North of Québec in Canada. The data is first reported by Ferland et al. (2006) and is available in the **tscount** package by command: `data("campy", package = "tscount")`. The code below fits an AR(1) model to the data with functions from the three packages mentioned above, as well as the copSTM. The conditional distribution is negative binomial with a log link.

```
library(gamlss, tscount, copSTM)
y <- campy[-1]
x <- log(campy[-length(campy)] + 1)
glm_res <- MASS::glm.nb(y~x)
gamlss_res <- gamlss(y ~ x,
                     family = NBII(mu.link = "log", sigma.link = "log"))
tscount_res <- tsglm(y, model = list(past_obs = 1),
                     link = "log", distr = "nbinom")
xx <- cbind(rep(1, length(x)), x)
copSTM_res <- copSTM(xx, y, K = 1, n = 1, marginal = "nbinom",
                     cor_type = "ind", temporal = TRUE)
```

Table 5.2 collects the results. Both the estimates and their standard errors (in parenthesis) of regression coefficients are quite similar in all packages. For the overdispersion parameter α, only **copSTM** and the standard `glm.nb` from **MASS** reach similar results. The running time of **tscount** is substantially larger than those of the others, because among the four packages, this is the only one written purely with R, apart from the fact that estimation is done by substituting the likelihood directly into a numerical optimisation function. It is not expected though, that the time required for our package is less than **MASS**, maybe because `glm.nb` provides a more complete result that fits into the GLM class.

parameters	**MASS**	**gamlss**	**tscount**	**copSTM**
β_0	0.723 (0.182)	0.754 (0.181)	0.672 (0.208)	0.723 (0.206)
β_1	0.694 (0.072)	0.683 (0.069)	0.713 (0.082)	0.694 (0.080)
α	11.214 (2.566)	18.713 (0.231)	9.100 (NA)	11.207 (2.154)
Time (Milliseconds)	10.931	27.369	222.343	2.522

Table 5.2: Parameter estimates with estimated standard errors in parenthesis using different packages, as well as the corresponding computational times.

5.5.2 For spatially correlated data ($T = n_{\mathscr{C}} = 1$)

In the case there is only one time point and one group, the copSTM model becomes a Gaussian copula model on univariate lattice count data. We compare this special case with the following two packages using Gaussian copulas:

- The **gcKrig** package written by Han and De Oliveira (2018) provides model estimation and spatial prediction on geostatistical count data, with three families of popular isotropic correlation functions: the Matérn family, the power exponential family and the spherical family.

- The **gcmr** package written by Masarotto et al. (2017) provides tools for fitting a general Gaussian copula regression model that can by applied to a wide variety of non-independent data sets. For spatial data, the Matérn family correlation function is implemented.

Since our package assumes a different correlation structure with the others, we compare the packages by fitting two simulated datasets with different correlation structures. Both datasets are generated on a 20×20 grid with Poisson marginals. The first one follows our correlation structure, that is, only observations in the same neighbourhood are correlated with specified correlation, which is 0.3 in this case. Note that the correlation families considered by **gcKrig** does not handle negative correlation, therefore we have to choose a positive value for ρ. The other dataset is simulated by the `simgc` function in the **gcKrig** package with the Matérn correlation with range $\phi = 0.8$ and no nugget.

The following code generates the first dataset:

```
K <- 1  ## The number of groups
n <- 20 ## n*n lattice
t_size <- 1 ## The number of time points
true_beta <- c(2, -1.5, 1)
true_rho <- 0.3
sim_dat1 <- sim_data(n, K, t_size = t_size, beta = true_beta,
                     marginal = "pois", cor_type = "sp",
                     rho = true_rho, temporal = FALSE)
```

The code below fits the Gaussian copula model to the first simulated data `sim_dat1` using **gcKrig, gcmr** and **copSTM**.

```
library(gcKrig, gcmr)
copSTM_res <- copSTM(sim_dat1$covariates, sim_dat1$response, K, n,
                     marginal = "pois", cor_type = "sp",
                     temporal = FALSE,  std_err = FALSE)
xloc <- rep(c(1:n), n)
yloc <- rep(c(1:n), each = n)
gcKrig_res <-  mlegc(y = sim_dat1$response, x = sim_dat1$covariates[, -1],
               locs = cbind(xloc, yloc), corr = matern.gc(nugget = FALSE),
               marginal = poisson.gc(link = "log"))
D <- sp::spDists(cbind(xloc, yloc))
gcmr_res <- gcmr(sim_dat1$response ~ sim_dat1$covariates[, -1],
                 marginal = poisson.marg, cormat = matern.cormat(D))
```

The following code generates the second dataset `sim_dat2`:

```
xloc <- rep(c(1:n), n)
yloc <- rep(c(1:n), each = n)
x1 <- rnorm(n*n)
x2 <- rnorm(n*n)
sim_dat2 <- simgc(locs = cbind(xloc, yloc), sim.n = 1,
                     marginal = poisson.gc(lambda = exp(2 - 1.5*x1 + x2)),
                     corr = matern.gc(range = 0.8, nugget = 0))
```

Code for fitting the Gaussian copula model on the second dataset is similar to that of the first, we thus omit it to avoid repetition.

Due to the design of the models, our package does not provide correlation parameters that are directly comparable to those of the other two packages. Therefore, we transfer between our estimated correlation at distance 1 (ρ) and the Matérn range parameter (ϕ) estimated in the other packages by functions in R package **fields**: The `Matern.cor.to.range` calculates the according range parameter with given correlation at distance `d`, while `Matern` does the opposite.

```
 Matern.cor.to.range(d = 1, cor.target = 0.275, nu = .5)
 [1] 0.7746029
 Matern(d = 1, range = 0.879)
[1] 0.3205694
```

Table 5.3 shows results of both datasets using the three packages. The second column shows true parameter values, where ρ is used only in the first data set while ϕ only in the second. For each package, the correlation parameter (either ρ or ϕ) that is computed with the **fields** functions is shown in Italic font, while the one estimated as model parameters as in the same font as other numbers. All three packages show satisfactory results in both data sets, suggesting a stable performance under misspecified correlation structures. On the other hand, the computational time required for **gcmr** is more than 1000 times of that needed for **copSTM**, for the same reason as **tscount**.

parameters		simData1			simData2		
		gcKrig	**gcmr**	**copSTM**	**gcKrig**	**gcmr**	**copSTM**
β_0	2	2.039	2.039	2.044	1.975	1.974	1.941
β_1	-1.5	-1.481	-1.481	-1.488	-1.516	-1.517	-1.526
β_2	1	0.990	0.990	0.985	0.996	0.997	1.003
ρ	0.3	*0.316*	*0.321*	0.275	*0.335*	*0.331*	0.271
ϕ	0.8	0.868	0.879	*0.775*	0.915	0.905	0.766
Time (seconds)		86.256	191.711	0.137	96.696	254.114	*0.169*

Table 5.3: Parameter estimates and corresponding computational times using different packages on two simulated datasets, where the first is generated with correlation between neighbouring locations (ρ), and the second is simulated using a Matérn correlation with range parameter (ϕ).

5.5.3 For univariate spatio-temporal data ($n_{\mathscr{C}} = 1$)

Although autoregressive temporal parameters are treated as regression coefficients, it is not appropriate to include the information of the previous time points directly as regression covariates in the Gaussian copula models discussed above. A possible reason is that these models do not distinguish observations at the same location but different time points. We demonstrate this with the first example on a simulated data with a 20×20 lattice at 10 time points.

```
K <- 1         ## The number of groups
n <- 20        ## n*n lattice
t_size <- 10   ## The number of time points
true_beta <- c(1, -0.8)
true_rho <- 0.3
sim_dat <- sim_data(n, K, t_size = t_size, beta = true_beta,
                    marginal = "pois", cor_type = "sp",
                    rho = true_rho, temporal = TRUE)
copSTM_res <- copSTM(sim_dat$covariates, sim_dat$response, K, n,
                     marginal = "pois", cor_type = "sp",
```

```
                    temporal = TRUE,  std_err = TRUE, B = 100,
                    Message_prog = TRUE)
xloc <- rep(c(1:n), n)
yloc <- rep(c(1:n), each = n)
gcKrig_res <-  mlegc(y = sim_dat$response, x = sim_dat$covariates[, -1],
                     locs = cbind(xloc, yloc),
                     corr = matern.gc(nugget = FALSE),
                     marginal = poisson.gc(link = "log"))
### Estimates
> unlist(copSTM_res$coefficients)
   intercept main_effects correlations
   0.9518744   -0.7577795    0.3201917
> t(summary(gcKrig_res)$coefficients$parest[, 1])
     Intercept         x     range
    0.9738393 -0.690831 0.8905873
### Standard errors
> unlist(copSTM_res$standard_error)
   intercept main_effects correlations
  0.05499572   0.06767610   0.01294451
t(summary(gcKrig_res)$coefficients$parest[, 2])
     Intercept     x    range
    0.000000 0.000000 0.140061
```

In this simulated data, **gcKrig** does not estimate well the only regression coefficient, whose true value is -0.8 but estimated as -0.69. Besides, the standard errors look unreasonably small, indicating that the model is not fitted on an appropriate type of data.

In the second example, we consider also another popular package: the **CARBayesST** by Lee et al. (2018), that can be applied to (univariate) spatio-temporal data. The model however, is built under a very different framework, specifically a latent process model extended from the conditional autoregressive (CAR) model and estimated under a Bayesian setting using Markov chain Monte Carlo (MCMC) simulation.

In this case, we consider function `ST.CARar` with Poisson responses in **CARBayesST**, which resembles our model the most. Specifically, $Y_{i,t}|\lambda_{i,t} \sim \text{Pois}(\lambda_{i,t})$, where $\log(\lambda_{i,t}) = \boldsymbol{X}'\boldsymbol{\beta} + \phi_{i,t}$, i and t indices for location and time respectively. The vector of random effects at time point t, $\boldsymbol{\phi}_t = (\phi_{1,t}, \ldots, \phi_{n_{\mathscr{L}},t})$ is modelled by $\boldsymbol{\phi}_t|\boldsymbol{\phi}_{t-1} \sim N(\rho_T\boldsymbol{\phi}_{t-1}, \tau^2\boldsymbol{Q}(\boldsymbol{W},\rho_S)^{-1})$ (Rushworth et al., 2014), where $\boldsymbol{Q}(\boldsymbol{W},\rho_S) = \rho_S[\text{diag}(\boldsymbol{W}\boldsymbol{1})] + (1-\rho_S)\boldsymbol{I}$ (Leroux et al., 2000), $\boldsymbol{W}$ is a symmetric adjacency matrix, ρ_S and ρ_T are spatial and temporal dependence parameter respectively taking values in the unit interval. Thus temporal autocorrelation is captured by mean $\rho_T\boldsymbol{\phi}_{t-1}$, while spatial autocorrelation is induced by the variance

$\tau^2 \boldsymbol{Q}(\boldsymbol{W}, \rho_S)^{-1}$.

We use data from a cloned cancer cell experiment that is similar to the one analysed in Chapter 2 and 3, except that there is only a single cell population resulting in a univariate spatio-temporal data.

parameters	β_0	β_1	ϕ	ρ	Time (seconds)
CARBayesST	0.216 (0.030)	0.997 (0.011)	–	–	85.56
copSTM	0.197 (0.043)	1.001 (0.013)	*0.447*	0.107 (0.017)	11.63
gcKrig	0.306 (0.221)	0.940 (0.117)	0.540 (0.069)	*0.157*	21.40

Table 5.4: Parameter estimates with standard error in the parenthesis, as well as computational times for running three functions from different packages on univariate spatio-temporal data.

Results, especially standard error estimates from **CARBayesST** and **copSTM** are very similar to each other, while those from **gcKrig** look quite different. Thus, we tend to believe results from the first two packages are more reliable in this case, which also confirms that a spatial Gaussian copula model should not be used for temporal data by simply adding some temporal covariates. On the other hand, although both **CARBayesST** and **copSTM** can fit spatio-temporal data and reach similar regression coefficient estimates, it is not possible to compare results of the correlation parameters, due to the fact that they are fundamentally different in correlation structure designs. **CARBayesST** returns estimates of τ^2, ρ_S, ρ_T as 0.000 (0.000), 0.980 (0.012), 0.526 (0.240) respectively, where τ^2 seems to suggest little, if any covariance for $\boldsymbol{\phi}_t$ and ρ_T is not very informative since the 95% confidence interval is (0.056, 0.996), almost covering (0, 1). It is also hard to compare the speed of the two packages because they both depend heavily on their tuning parameters: `n.sample` for **CARBayesST** (the number of MCMC samples) and `B` for **copSTM** (the number of bootstrap samples). For the results displayed in Table 5.4, these parameters are `n.sample = 50000` and `B = 100`.

Unfortunately, **CARBayesST** does not consider multivariate case, while its sister package, the **CARBayes** by Lee (2013) concerns multivariate, but only spatial data. The model considered is also a CAR-based mixed effect model with spatial and cross variable autocorrelations captured in random effects. However similar to the example with **CARBayesST**, although we can fit a multivariate data to functions in **CARBayes**, correlation parameters can not be interpreted in the straightforward manner like copula-based models. Besides, the MCMC-based estimation for latent process models almost always takes longer to run than likelihood-based methods even for very elegantly coded packages like the ones discussed above.

5.5.4 Comparing model selection functions

Finally, we compare one of our model selection functions, the `logGLMSelect` with the package **glmulti** (Calcagno et al., 2010) that specialised in model selection on GLM using generic algorithms. The **glmulti** package offers flexible model selection tools on GLM models, allowing not just main effects but also interaction terms, as well as constraints on model complexities.

Here we compare the two package functions on a simulated dataset, since in this case we know the what the true models is. The following code generates data from a standard Poisson regression model.

```
true_beta <- c(1, 0, 0, 2, 3, 0, 0, 1, 0, 0, 0)
                   ## true model: 110011001000
intercept <- 1
p <- length(true_beta)
nn <- 1000      ## sample size
x <- matrix(rnorm(p*nn), nn, p)
mu <- exp(x %*% true_beta + intercept)
y <- sapply(mu, rpois, n = 1)
```

We first perform model selection on the generated data on `logGLMselect` from **copSTM**. Note that every run of `logGLMselect` may give a different number of frequencies of generated models, but the best selected model stays the same.

```
copSTM_res <- logGLMselect(y, x, "pois", ModelCnt = 100, Message = TRUE)
Model 110011001000 appeared 93 times
Model 111011001010 appeared 4 times
Model 111011001000 appeared 2 times
Model 110011001001 appeared 1 times
```

Already we see that the true model appears the most frequently (93 times), the according time is 0.18 seconds. Then we feed the same data into the `glmulti` function in **glmulti**:

```
dt <- data.frame(y, x)
ini <- glm(y ~., data = dt, family = poisson)
glmulti_res <- glmulti::glmulti(ini, level = 1, crit = bic, method = "h")
summary(glmulti_res)$bestmodel
[1] "y ~ 1 + X4 + X5"
```

It is quite surprising that even with the method of exhaustive screening, which runs for 9.33 seconds, `glmulti` does not pick up the true model (with `X8` missing) from which the data is generated. Finally, we check the BIC value calculated by the standard `glm` function in **stats** with both selected models:

```
selected <- which(copSTM_res$selected_model == 1)
glma <- glm(y ~ ., data = dt[, selected], family = poisson)
glmb <- glm(summary(glmulti_res)$bestmodel, data = dt, family = poisson)
> BIC(glma) ## from copSTM
[1] 4039.017
> BIC(glmb) ## from glmulti
[1] 2115970
```

This confirms that **copSTM** dose pick up the model with a smaller BIC value than **glmulti** and more than 50 times faster.

Admittedly, **glmulti** provides many functionalities that **copSTM** does not, for example, it can fit all standard response distributions in the exponential family while **copSTM** (currently) can handle only Poisson and negative binomial, but can possibly include it in future versions.

5.6 Discussions

This chapter is a vignette of an R package **copSTM** for the analysis of spatio-temporal count data on lattice using Gaussian copula regression models. The package, together with its supplementary web application (built with R package **Shiny** (Chang et al., 2018)), implements most of the main tasks commonly required in the analysis of this kind of data: simulation, visualization, parameter estimation and model selection. The web application is built mostly for non-R users, since it does not impose any requirement on the local computer, all that is needed is a browser. It produces visualized data summary as well as parameter estimation and variable selection in a straightforward and interpretable manner. All outputs are available for download as pdf plots or csv tables. But due to computational restrictions, this application is limited to independent regression models where the correlation matrix $\mathbf{\Sigma}$ is an identity matrix. On the other hand, the R package provides the computational intensive evaluation of the copula models.

All package functions are implemented in C++, linked to R through **RcppArmadillo** (Eddelbuettel and Sanderson, 2014). Apart from RcppArmadillo, the **copSTM** does not rely on any other R package, thus avoiding the cost of repeatedly calling R functions from C++. There are, however, functions modified from other R packages and translated to C++, these functions are not exported. Our numerical examples show that the package produces similar results in the special cases that are comparable to other existing packages, confirming the correctness of our calculations. On the other hand, our package offers very competitive computational speed, almost always several times faster than its competitors.

The goal of **copSTM** is not to replace any other R package already available, but rather

to provide investigators with an alternative option of modelling and fast criterion based variable selection. In the future, we plan to enhance our package in the following directions. First, the model functions can be extended to handle not just regular lattice data but geographical data as well, by allowing a user specified adjacency or distance matrix as input, and implementing other correlation structures including the Matérn, the power exponential and the spherical families. Second, the model selection function `logGLMselect` can be extended to a more complete variable selection toolkit for generalized linear regression models. For example, allow other exponential family distributions for the response variables, apart from the Poisson and Negative binomial distribution that are currently implemented, or provide options for more complicated formulas such as interaction terms instead only linear terms in the current version. Finally, it is worthy to write the package in the S3 object specification scheme with standard methods such as `summary` and `print` and enable functions to pass and return objects following the general framework of the standard `glm` that is familiar to most R users.

5.7 Appendix. An R Shiny application: Interactive Analysis of Spatio-temporal Cell Growth Data

5.7.1 Installation

- **Option 1: Web page**.

  The Shiny application is available as a web page with link: `https://pqiao.shinyapps.io/STModelling/`.

  Pros: Very easy access. Does not impose any requirement on the local computer.

  Cons: Slower computation. Computations of this web page version is run on the server hosted by Shinyapps.io, due to memory limitations and other reasons, it is expected to be around 10 times slower than a local computer. This drawback is not obvious for most functions that run relatively fast, typically within seconds. The only exception is the last part (model selection) where massive computation is involved. For example, if a typical wait on local computer is 15 seconds, then a 2-minute run time is usually expected on the web page.

- **Option2: Download source on local computer**.

  Download source code from a git repository on GitHub and run on local computer. This could be done with a single R command:

`shiny::runGitHub('STModelling','pqiao29')`.

Pros: Fast, therefore better user experience.

Cons: The local computer needs to have the following installed:

– R (>= 2.10.0),

– A few R packages:

i). For launching the application: Shiny, dplyr, ggplot2.

Install by command: `install.packages("...")` and replace ... with the

package name, for example `install.packages("Shiny")`.

ii) For installing package copSTM (in iii)): devtools, RcppArmadillo.

Install in the same way as i).

iii) For all computations in this application: copSTM.

Install by command: `devtools::install_github("pqiao29/copSTM")`.

– C++ compiller: Rtools (for Windows), Xcode (for Mac),

`sudoapt-getinstallr-base-dev` or similar (for Linux).

5.7.2 A Step-by-Step Workflow

All graphs presented in this application are interactive with input tuning parameters and are available for download as .pdf files, some of the data results can also be dowloaded as .csv files. The initial interface is the same as shown in Figure 5.1, where we refer the left half of the page (the grey area) as the side panel for user inputs, and the right half as the main panel for displaying outputs, which is composed of four tabs: Data, Summary, Estimation and Selection.

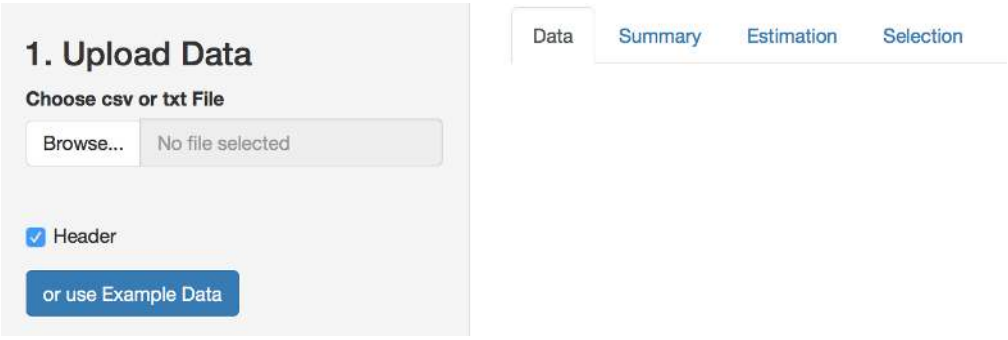


Figure 5.1: Initial interface

- **Step1. Upload Data**

  Data can be uploaded from a local .txt or .csv file. Select the local file via the white button "Browse..." (See Figure 5.1). Uploaded data is required to contain four columns in the following order (separated by space for .txt file):

  1. Time: The time point in which the data point is observed. Required to be continuous integer numbers starting from 0, for example, 0, 1, 2, ...
  2. X: Spatial coordinate x, any real number.
  3. Y: Spatial coordinate y, any real number.
  4. Group: A categorial variable indicating which group or cell population this point or cell belongs to. Required to be continuous integer numbers starting from 1, for example, 1, 2, 3...

  It is not required that the uploaded data contain headers (column names), but if it does, the "Header" box needs to be checked.

  Alternatively, click the blue button "or use Example Data" to explore the functionalities with the built-in data for illustrative purpose.

- **Step2. View Data**

  When data is successfully uploaded or the example data is chosen, the "Data" tab in the main panel will display the number of time points and groups from the input data, as well as the data itself (See Figure 5.2). This is to confirm that the read-in file is uploaded as intended.

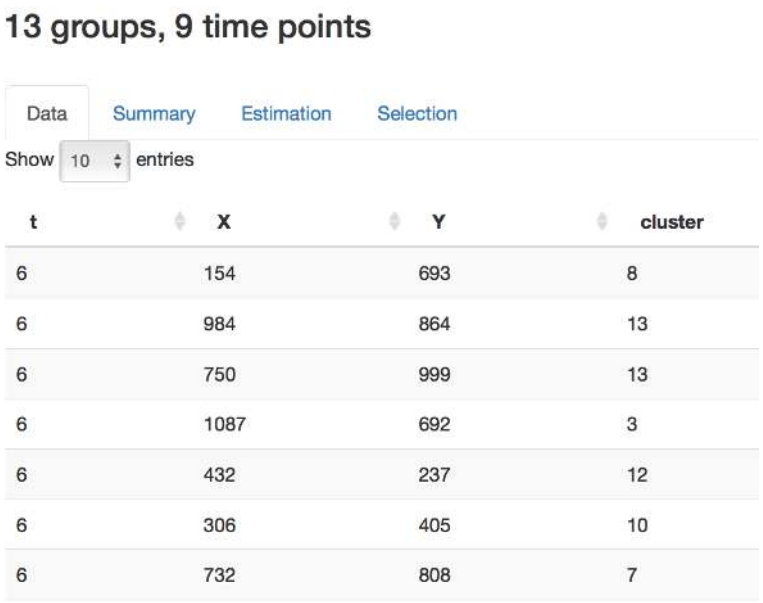


Figure 5.2: Step2. view data

At the same time, some slide bars for adjusting tuning parameters will appear on

the side panel, which will be discussed in later steps.

- **Step3. Data Summary**

  To get a visualised data summary, first go to the "Summary" tab in the main panel and click the blue button "Show Data Summary" in the side panel (See Figure 5.3).

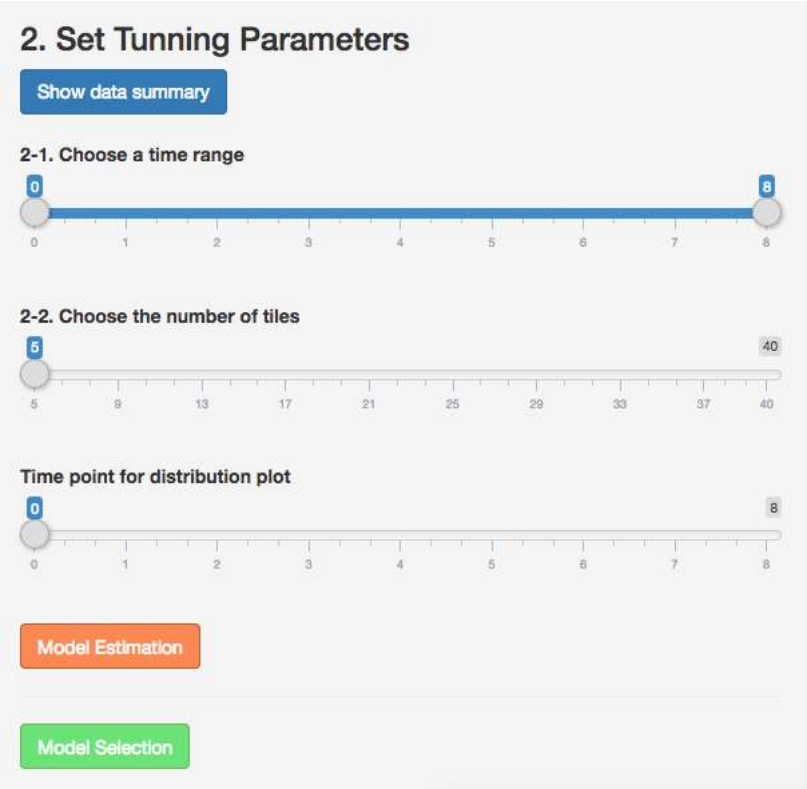


Figure 5.3: Tuning parameters

This part provides two kinds of plots for exploring the temporal and spatial trend of the data separately:

The Growth Curves show the change of the total cell count of each group in the whole image across time, where the x-axis is the time points and y-axis shows the count. Groups are distinguished by colours. The first and last time point can be specified by the slide bar for time range in order to exclude unwanted images (time points), the plot changes interactively with the input time range.

For example, in the example data, the growth curves of all time points are shown in Figure 5.7.2 (left), but because the last time point (i.e. 8) looks like a low quality image with unreasonably low count of all groups, one can adjust the time range to exclude this time point and obtain the plot on the right.

The Distribution heatmap summarises the spatial distribution of cell count (regardless of the group) in the tiled image. By selecting through the slide bar "2-2", one can specify the number of tiles n, and the image will be tiled into an $n \times n$

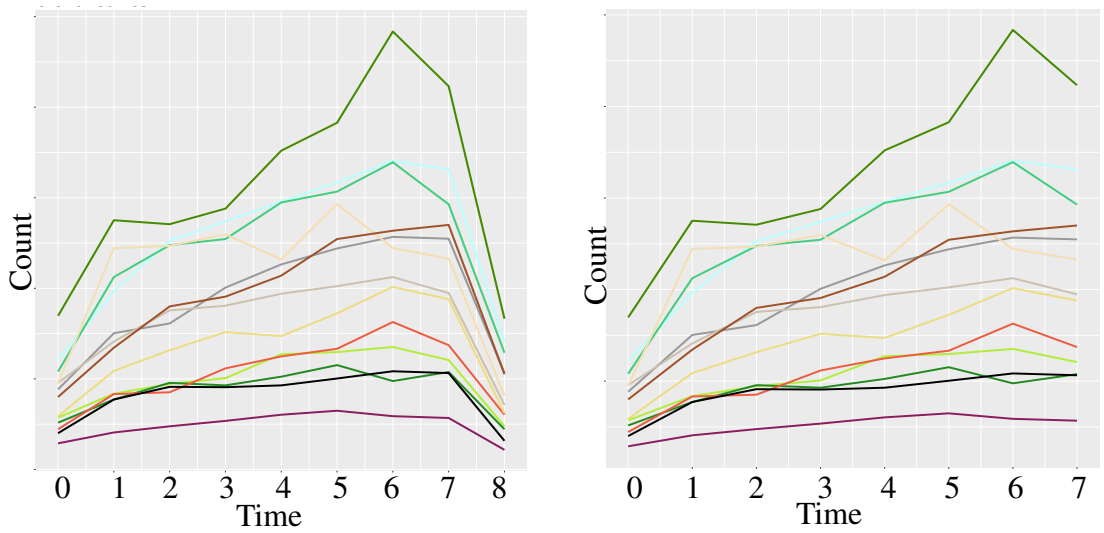


Figure 5.4: Growth curves of total count of 13 groups across time. Left: plot of all time points. Right: the last time point is omitted.

lattice. Accordingly, an $n \times n$ heat map is plotted, where the numbers shown in the plot are the cell counts in the tiles, with darker colours indicating higher counts. Each heat map shows the spatial distribution of only one time point, therefore, one can change the option on the "Time point for distribution plot" to view the change at different time points. For example, by sliding this bar, we obtain in Figure 5.5 a series of heat maps gradually getting darker indicating the growth over time.

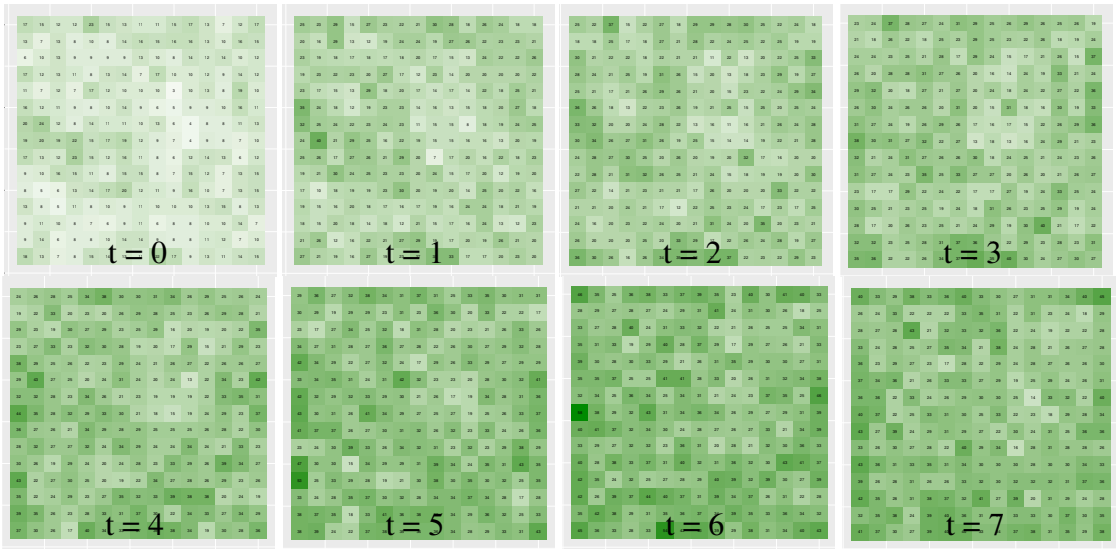


Figure 5.5: Spatial distribution of cell counts in a 15×15 lattice across time.

Finally, note that the choice of the first two tuning parameters also affect later analysis, the model estimation and selection.

- **Step4. Model Estimation**

  Click the orange button "Model Estimation" in the side penal and go to the "Es-

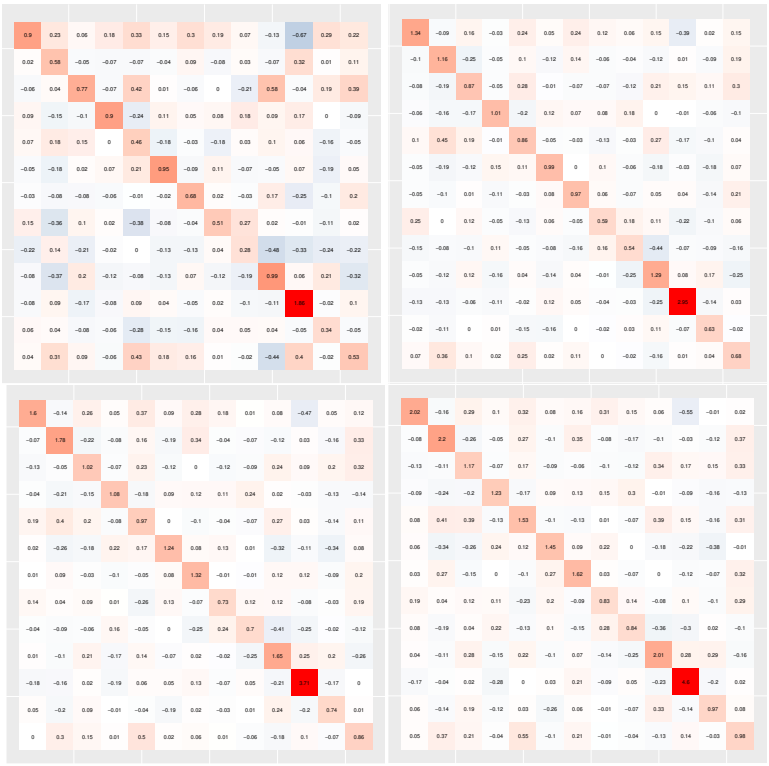
timation" tab in the main panel. Two plots are shown in this section: the model estimation and assessment.

The Model Estimation shows estimated autoregressive parameters representing the impacts on growth between groups in a $n_{\mathscr{C}} \times n_{\mathscr{C}}$ table where $n_{\mathscr{C}}$ is the number of groups. The number shown in the ith row and jth column is interpreted as change in the average cell count of group j per tile, due to interactions with cells of group i in neighbouring tiles. For "neighbouring" tiles, we take the Moore neighbourhood that is composed of a central tile and the eight tiles surrounding it.

A positive (or a negative) sign of the number indicates that the presence of cells of group i in neighbouring tiles promotes (or inhibits) the growth of cells of group j. Thus, by looking at each row of the table, we see the influence of each group on other groups while each column represents the sensitivity. The table is coloured as a heat map to give a convenient overview of the impacts in general, where red for positive white for zero and steel blue for negative.

After changing the time range and/or the number of tiles n in the side panel, click the orange button again, the estimations change accordingly. In general, a larger n leads to finer grided image and therefore smaller tiles and neighbourhoods, resulting in more local impacts. Estimates are generally stable with mild changes in n. For example, Figure 5.6 shows the estimates of the example data (the last time point excluded) with n goes from 10 to 25.

$n = 10$ $n = 15$



$n = 20$ $n = 25$

Figure 5.6: Estimated impacts with $n = 10, 15, 20, 25$

The Goodness-of-Fit Curves are produced at the same time of estimation, where each colour corresponds to one group (same as the growth curves), with the solid lines being the observed total cell counts in the whole image at different time points, while the dashed line being the fitted values. Figure 5.7 shows the goodness-of-fit curves at $n = 25$ as an example.

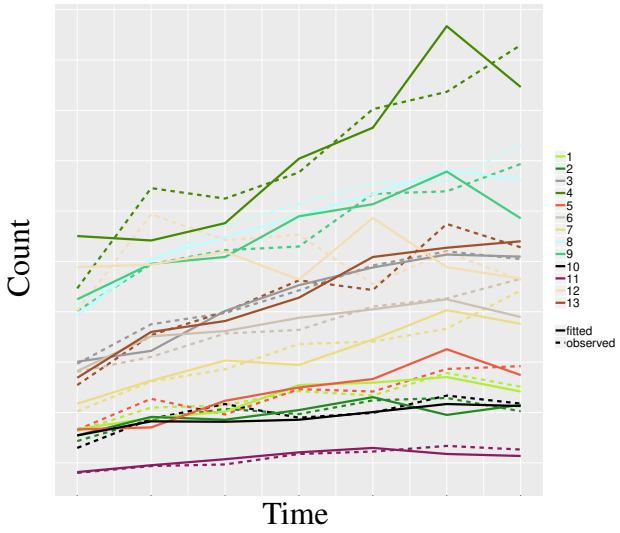


Figure 5.7: Goodness-of-fit curves with $n = 15$.

- **Step5. Model Selection**

  Model section is activated by clicking the green button in the side panel, and results are shown in the "Selection" tab in the main panel.

  The goal of model selection is to retain only the meaningful spatio-temporal impacts between groups. Selection is based on the traditional Bayesian Information Criterion (BIC). For a small number of parameters, p, one could do it by brute force, that is, try all the 2^p candidate models and select the one with the lowest criterion value. However, this is not computationally effective or even feasible when p is large, for example, in this case 169.

  Thus, model selection in this application is carried out in a more elegant way, with a Gibbs sampling method introduced by Qian and Field (2002). All functions are implemented in C++, which are 30 to 40 times faster than their R equivalences. The resulting computational time with this data set for example, is around 15 seconds for $n = 15$ if run on a local computer and less than 2 minutes on web page. Note that the time elapse depends mostly on the value of n, therefore it is recommended to start with a small n and move to the intended value when ready for the wait. If wish to change tuning parameter values, need to click the green button again to create an updated result.

  Result of the model selection is shown in the same format as the estimation heat map, except that only the significant parameters are coloured and numbered. A pair

of results of the full model (left) and selected model (right) are shown in Figure 5.8 with $n = 15$. The graph on the right filters out the "noisy" insignificant parameters thus empathising more on the selected ones. A mild variation in numbers is expected when the model is fitted on only a subset of the predictors.

Figure 5.8: Full model estimation (left) and model selection (right) with $n = 15$.

Chapter 6

Discussion and future work

6.1 Summary and final remarks

Spatio-temporal data arises from many scientific disciplines such as environmental sciences, ecology and pathology among others. Our work in this thesis is originally motivated by data obtained from a longitudinal image data collected from live-cell growth experiment based on flourescent proteins. The RGB marking technique introduces three lentiviral vectors in individual cells encoding the basic colours red, green and blue. Using a high-content imaging system (Operetta, Perkin Elmer), characteristics for each individual cell can be observed at subsequent times, including spatial coordinates, morphological features and measurements on quantities of three different flourescent proteins: Cerulean (blue), Venus (green) and mCherry (red). Through data analysis, scientists are interested in assessing how interactions between cell types may affect their growth.

The first question of interest arises from the identification of cell types. We propose a semi-supervised regression clustering approach, that takes into consideration not only the three-dimensional vector response indicating colours, but also other morphology parameters such as cell area and roundness. The regression clustering is performed iteratively starting from the initial partition given by the robust K-means (Kondo et al., 2016). The prediction strength (Tibshirani and Walther, 2005) obtained via cross validation confirms that the regression clustering result by taking cell area as an explanatory variable produces much higher predictive power than a simple robust K-means.

With cells clustered into $n_{\mathscr{C}}$ groups, we propose to quantify the spatial distribution for different cell groups by dividing the images into a number of contiguous regions (tiles) to form an $n \times n$ regular lattice structure and count the frequency of each cell group inside each tile at subsequent T time points. This yields an array of $n \times n \times T$ spatial-temporal observations, in which each entry is a $n_{\mathscr{C}}$ dimensional vector of counts. Different strategies have been proposed to model spatio-temporal data, depending on the goals of the analysis. In this thesis, we are primarily concerned with the impacts between cell groups

on growth patterns.

In Chapter 2, we propose a conditional temporal model with Poisson responses. The model falls into a general class of the GARMA model (Benjamin et al., 2003), which can be classified as observation-driven models in the context of time series analysis, termed by Cox et al. (1981). Model parameters are directly interpreted as impacts on growth of each cell group due to interaction with other cell groups from the previous time point within a spatial neighbourhood. Applications on the illustrative real data examples confirm that the conclusions drawn by model parameter estimates are biologically meaningful.

Because this model is relatively simple and fast to fit, it allows us to build a web application making it accessible to non-R users: `https://pqiao.shinyapps.io/STModelling/`. The application is built with the R package `shiny` (Chang et al., 2018), it is very user friendly and straighforward to use. It provides tools for model visualization, estimation and selection carried out interactively with the user's choice of tuning parameters such as n. Users may choose to upload and analyze their own data or explore the functionalities with the built-in data example. All analysis results are available for download as pdf plots and csv table.

Yet the most challenging aspect for modelling spatio-temporal data is the autocorrelations that occur temporally, spatially and cross-variables (i.e. between groups in our case). Therefore in Chapter 3, we develop a Gaussian copula regression model for the analysis of multivariate spatio-temporal data, the copSTM, extending the model proposed in Chapter 2. Gaussian copulas are a powerful and flexible method for modelling multivariate data, it combines the simplicity of interpretation in marginal modelling with the flexibility in the specification of the dependence structure. The marginal response is modelled similarly with the GARMA model structure, while a correlation matrix is designed for capturing correlations between cell groups and between neighbouring tiles. Since we are primarily concerned with count data, we only consider Poisson and Negative binomial marginal distributions, but the model can easily be applied to any distribution in the exponential family.

However, the likelihood function of the copula model includes high-dimensional integrals, making approximation and optimization extremely slow, taking up to four to five hours to our experience. To solve this problem, we adopt the pairwise composite likelihood methods (Varin et al., 2011), which greatly reduces the dimensions of the integrals. We also derive the closed forms of the first and second derivatives of the composite log-likelihood function. This makes our computation much faster compared to "dumping" the approximated likelihood function into a numerical optimization function for general purposes (like `optim` in package `stats`) as many other R packages do. Finally, this computational difficulty also motivates us to rewrite our R code into C++. By carefully avoiding repetitive computations or copying big matrices (this is possible thanks to the C++ functionality for passing variables by reference), we manage to reduce the time elapse for the

same estimation task to 45 seconds. This is part of the work presented in the R package described in Chapter 5.

Another concern of copula models is the risk of over-parametrization, due to the fact that they naturally induce more parameters (the correlation parameters) than the simpler generalized linear regression models. Thus, in order to select the sub-model that contains only the meaningful parameters, we implement an information criterion-based model selection on the proposed copSTM model in Chapter 4. We adopt the CL-BIC developed by Gao and Song (2010) as selection criterion, which can be seen as a sister of the traditional BIC but under the composite likelihood framework. The search through the candidate model set borrows strength from the Gibbs sampling technique. The methodology guarantees to converge to the model with the lowest criterion value, but without searching through all possible models exhaustively. It is originally introduced by Qian and Field (2002) in the context of logistic linear regression models, but has the potential to be applied to any regression-based models as long as the information criterion is properly chosen or designed.

Finally, the estimation and variable selection of the copSTM model proposed in Chapter 2, 3 and 4 are wrapped in an R package with the same name `copSTM`. Chapter 5 serves as a vignette of this package, all exported functions are described in details with examples. This chapter also shows how most of the numerical results on both simulated and real data in this thesis can be reproduced with this package. This is important because *"Reproducibility is the only thing that an investigator can guarantee about a study."* – Roger Peng.

Besides, we also show a number of examples comparing with other available R packages, confirming the correctness and efficiency of our package. Although this package is currently only available from GitHub, we believe it provides R users a competitive toolkit for the analysis spatio-temporal count data on lattice, and hope it can eventually make its way to CRAN in the near future.

6.2 Future Research

In this section, we outline some future research directions that deserve investigation.

- **Joint regression models.** Several authors exploited Gaussian and t-copulas to construct joint regression models for responses of mixed types. (Frees and Valdez, 2008; Song et al., 2009; Wu and de Leon, 2014; Jiryaie et al., 2016) Implementation of methods for handling responses with multiple distributions is a possible extension for future versions of the package `copSTM`.

- **Copula vines.** The use of copulas based on graphical models called vines, along the lines described by Panagiotelis et al. (2012). Gräler (2014) and Erhardt et al.

(2015a) and Erhardt et al. (2015b) use this approach for modelling geostatistical continuous data, which is implemented in the R package `spcopula` (Gräler, 2017). The application of copula vines to the modelling of geostatistical count data seems a promising topic of future research.

- **Maximization by parts** Maximization by parts is a numerical iterative algorithm proposed by Song et al. (2005) to optimize complex log-likelihoods that can be partitioned into a manageable "working log-likelihood" plus a more complex "remainder log-likelihood". The algorithm aims to enhance numerical stability relative to the direct numerical optimization of the likelihood function. Among other applications, maximization by parts has been proposed for fitting continuous Gaussian copula regression models

Bibliography

H. Akaikei. Information theory and an extension of maximum likelihood principle. In *Proc. 2nd Int. Symp. on Information Theory*, pages 267–281, 1973.

Y. Bai, P. X.-K. Song, and T. Raghunathan. Joint composite estimating functions in spatiotemporal models. *J R Stat Soc Series B Stat Methodol*, 74(5):799–824, 2012.

Y. Bai, J. Kang, and P. X.-K. Song. Efficient pairwise composite likelihood estimation for spatial-clustered data. *Biometrics*, 70(3):661–670, 2014.

K. S. Bakar, S. K. Sahu, et al. sptimer: Spatio-temporal bayesian modelling using r. *Journal of Statistical Software*, 63(15):1–32, 2015.

D. Bates and M. Maechler. *Matrix: Sparse and Dense Matrix Classes and Methods*, 2019. URL `https://CRAN.R-project.org/package=Matrix`. R package version 1.2-17.

M. A. Benjamin, R. A. Rigby, and D. M. Stasinopoulos. Generalized autoregressive moving average models. *Journal of the American Statistical association*, 98(461):214–223, 2003.

J. Besag. Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 192–236, 1974.

J. Bradic, J. Fan, and W. Wang. Penalized composite quasi-likelihood for ultrahigh dimensional variable selection. *J R Stat Soc Series B Stat Methodol*, 73(3):325–349, 2011.

J. R. Bradley, S. H. Holan, C. K. Wikle, et al. Multivariate spatio-temporal models for high-dimensional areal data with application to longitudinal employer-household dynamics. *The Annals of Applied Statistics*, 9(4):1761–1791, 2015.

J. R. Bradley, S. H. Holan, and C. K. Wikle. Multivariate spatio-temporal survey fusion with application to the american community survey and local area unemployment statistics. *Stat*, 5(1):224–233, 2016.

J. R. Bradley, S. H. Holan, C. K. Wikle, et al. Computationally efficient multivariate spatio-temporal models for high-dimensional count-valued data. *Bayesian Analysis*, 2017.

V. Calcagno, C. de Mazancourt, et al. glmulti: an r package for easy automated model selection with (generalized) linear models. *Journal of statistical software*, 34(12):1–29, 2010.

B. P. Carlin, A. E. Gelfand, and S. Banerjee. *Hierarchical modeling and analysis for spatial data*. Chapman and Hall/CRC, 2014.

M. Cattelan and N. Sartori. Empirical and simulated adjustments of composite likelihood ratio statistics. *Journal of Statistical Computation and Simulation*, 86(5):1056–1067, 2016.

J. O. Cerdeira, P. Silva, J. Cadima, and M. Minhoto. subselect: Selecting variable subsets. *R package version 0.9-9993, URL http://CRAN. R-project. org/package= subselect*, 2009.

W. Chang, J. Cheng, J. Allaire, Y. Xie, and J. McPherson. *shiny: Web Application Framework for R*, 2018. URL `https://CRAN.R-project.org/package=shiny`. R package version 1.2.0.

V. Christou and K. Fokianos. Quasi-likelihood inference for negative binomial time series models. *Journal of Time Series Analysis*, 35(1):55–78, 2014.

D. R. Cox, G. Gudmundsson, G. Lindgren, L. Bondesson, E. Harsaae, P. Laake, K. Juselius, and S. L. Lauritzen. Statistical analysis of time series: Some recent developments [with discussion and reply]. *Scandinavian Journal of Statistics*, pages 93–115, 1981.

N. Cressie and C. K. Wikle. *Statistics for spatio-temporal data*. John Wiley & Sons, 2015.

R. A. Davis, W. T. Dunsmuir, and S. B. Streett. Observation-driven models for poisson counts. *Biometrika*, 90(4):777–790, 2003.

P. Diggle, P. J. Diggle, P. Heagerty, P. J. Heagerty, K.-Y. Liang, S. Zeger, et al. *Analysis of longitudinal data*. Oxford University Press, 2002.

W. T. Dunsmuir, D. J. Scott, et al. The glarma package for observation driven time series regression of counts. *Journal of Statistical Software*, 67(7):1–36, 2015.

D. Eddelbuettel and C. Sanderson. Rcpparmadillo: Accelerating r with high-performance c++ linear algebra. *Computational Statistics & Data Analysis*, 71:1054–1063, 2014.

T. M. Erhardt, C. Czado, and U. Schepsmeier. R-vine models for spatial time series with an application to daily mean temperature. *Biometrics*, 71(2):323–332, 2015a.

T. M. Erhardt, C. Czado, and U. Schepsmeier. Spatial composite likelihood inference using local c-vines. *Journal of Multivariate Analysis*, 138:74–88, 2015b.

J. Fan and R. Li. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American statistical Association*, 96(456):1348–1360, 2001.

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

D. Ferrari and D. L. Vecchia. On robust estimation via pseudo-additive information. *Biometrika*, 99(1):238–244, 2011.

A. O. Finley, S. Banerjee, and A. E.Gelfand. spBayes for large univariate and multivariate point-referenced spatio-temporal data models. *Journal of Statistical Software*, 63(13): 1–28, 2015. URL `http://www.jstatsoft.org/v63/i13/`.

K. Fokianos and D. Tjøstheim. Log-linear poisson autoregression. *Journal of Multivariate Analysis*, 102(3):563–578, 2011.

K. Fokianos, A. Rahbek, and D. Tjøstheim. Poisson autoregression. *Journal of the American Statistical Association*, 104(488):1430–1439, 2009.

E. W. Frees and E. A. Valdez. Hierarchical insurance claims modeling. *Journal of the American Statistical Association*, 103(484):1457–1469, 2008.

X. Gao and P. X.-K. Song. Composite likelihood bayesian information criteria for model selection in high-dimensional data. *Journal of the American Statistical Association*, 105(492):1531–1540, 2010.

G. A. Garden and A. R. La Spada. Intercellular (mis) communication in neurodegenerative disease. *Neuron*, 73(5):886–901, 2012.

A. E. Gelfand and P. Vounatsou. Proper multivariate conditional autoregressive models for spatial data analysis. *Biostatistics*, 4(1):11–15, 2003.

H. Geys, G. Molenberghs, and L. M. Ryan. Pseudolikelihood modeling of multivariate outcomes in developmental toxicology. *Journal of the American Statistical Association*, 94(447):734–745, 1999.

V. P. Godambe. An optimum property of regular maximum likelihood estimation. *The Annals of Mathematical Statistics*, 31(4):1208–1211, 1960.

E. Goren and J. Hughes. copcar: Fitting the copcar regression model for discrete areal data. *Denver, CO. R package version*, pages 2–0, 2017.

B. Gräler. Modelling skewed spatial random fields through the spatial vine copula. *Spatial Statistics*, 10:87–102, 2014.

B. Gräler. spcopula: Modelling spatial and spatio-temporal dependence with copulas in r. 2017. URL `https://R-forge.R-project.org/projects/spcopula/`. R package version 0.2.4.

Z. Han and V. De Oliveira. gckrig: An r package for the analysis of geostatistical count data using gaussian copulas. *Journal of Statistical Software*, 87(1):1–32, 2018.

P. J. Heagerty and S. R. Lele. A composite likelihood approach to binary spatial data. *Journal of the American Statistical Association*, 93(443):1099–1111, 1998.

L. Held, M. Höhle, and M. Hofmann. A statistical framework for the analysis of multivariate infectious disease surveillance counts. *Statistical modelling*, 5(3):187–199, 2005.

S. Holan and C. Wikle. Hierarchical dynamic generalized linear mixed models for discrete-valued spatio-temporal data. *Handbook of Discrete–Valued Time Series*, 2015.

N. Jenish and I. R. Prucha. Central limit theorems and uniform laws of large numbers for arrays of random fields. *J Econom*, 150(1):86–98, 2009.

F. Jiryaie, N. Withanage, B. Wu, and A. De Leon. Gaussian copula distributions for mixed data, with application in discrimination. *Journal of Statistical Computation and Simulation*, 86(9):1643–1659, 2016.

H. Joe. *Dependence modeling with copulas*. Chapman and Hall/CRC, 2014.

R. Kalluri and M. Zeisberg. Fibroblasts in cancer. *Nat Rev Cancer*, 6(5):392–401, 2006.

H. Kazianka and J. Pilz. Copula-based geostatistical modeling of continuous and discrete data including covariates. *Stochastic Environmental Research and Risk Assessment*, 24 (5):661–673, 2010.

B. Kedem and K. Fokianos. *Regression models for time series analysis*, volume 488. John Wiley & Sons, 2005.

L. Knorr-Held. Bayesian modelling of inseparable space-time variation in disease risk. *Statistics in medicine*, 19(17-18):2555–2567, 2000.

L. Knorr-Held and S. Richardson. A hierarchical model for space–time surveillance data on meningococcal disease incidence. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 52(2):169–183, 2003.

R. Koenker. Quantile regression for longitudinal data. *Journal of Multivariate Analysis*, 91(1):74–89, 2004.

Y. Kondo, M. Salibian-Barrera, R. Zamar, et al. Rskc: an r package for a robust and sparse k-means clustering algorithm. *Journal of Statistical Software*, 72(5):1–26, 2016.

D. La Vecchia, L. Camponovo, and D. Ferrari. Robust heart rate variability analysis by generalized entropy minimization. *Computational Statistics & Data Analysis*, 82:137–151, 2015.

D. Lee. Carbayes: An r package for bayesian spatial modeling with conditional autoregressive priors. *Journal of Statistical Software*, 55(13):1–24, 2013.

D. Lee, A. Rushworth, and G. Napier. Spatio-temporal areal unit modelling in r with conditional autoregressive priors using the carbayesst package. *Journal of Statistical Software*, 84(9), 2018.

G. Leoni, P. Neumann, R. Sumagin, et al. Wound repair: role of immune–epithelial interactions. *Mucosal Immunol*, 8(5):959–968, 2015.

B. G. Leroux, X. Lei, and N. Breslow. Estimation of disease rates in small areas: a new mixed model for spatial dependence. In *Statistical models in epidemiology, the environment, and clinical trials*, pages 179–191. Springer, 2000.

K.-Y. Liang and S. L. Zeger. Longitudinal data analysis using generalized linear models. *Biometrika*, 73(1):13–22, 1986.

T. Liboschik, K. Fokianos, and R. Fried. tscount: An r package for analysis of count time series following generalized linear models. *Journal of Statistical Software, Articles*, 82 (5):1–51, 2017.

B. G. Lindsay. Composite likelihood methods. *Contemporary mathematics*, 80(1):221–239, 1988.

T. Lumley and A. Miller. Leaps: regression subset selection. r package version 2.9. *See http://CRAN. R-project. org/package= leaps*, 2009.

G. Masarotto, C. Varin, et al. Gaussian copula marginal regression. *Electronic Journal of Statistics*, 6:1517–1549, 2012.

G. Masarotto, C. Varin, et al. Gaussian copula regression in r. *Journal of Statistical Software*, 77(8):1–26, 2017.

A. McLeod and C. Xu. bestglm: Best subset glm. *URL http://CRAN. R-project. org/package= bestglm*, 2010.

J. P. Medema and L. Vermeulen. Microenvironmental regulation of stem cells in intestinal homeostasis and cancer. *Nature*, 474(7351):318, 2011.

C. Meyer. Recursive numerical evaluation of the cumulative bivariate normal distribution. *arXiv preprint arXiv:1004.3616*, 2010.

A. Miller. *Subset selection in regression*. Chapman and Hall/CRC, 2002.

M. Mohme, C. L. Maire, K. Riecken, S. Zapf, T. Aranyossy, M. Westphal, K. Lamszus, and B. Fehse. Optical barcoding for single-clone tracking to study tumor heterogeneity. *Molecular Therapy*, 2017.

A. S. Mugglin, N. Cressie, and I. Gemmell. Hierarchical statistical modelling of influenza epidemic dynamics in space and time. *Statistics in medicine*, 21(18):2703–2721, 2002.

V. N. Nair. Qq plots with confidence bands for comparing several populations. *Scandinavian Journal of Statistics*, pages 193–200, 1982.

A. K. Nikoloulopoulos. On the estimation of normal copula discrete regression models using the continuous extension and simulated likelihood. *Journal of Statistical Planning and Inference*, 143(11):1923–1937, 2013.

A. K. Nikoloulopoulos. Efficient estimation of high-dimensional multivariate normal copula models with discrete spatial responses. *Stochastic environmental research and risk assessment*, 30(2):493–505, 2016.

A. Panagiotelis, C. Czado, and H. Joe. Pair copula constructions for multivariate discrete data. *Journal of the American Statistical Association*, 107(499):1063–1072, 2012.

M. Paul and L. Held. Predictive assessment of a non-linear random effects model for multivariate time series of infectious disease counts. *Statistics in Medicine*, 30(10): 1118–1136, 2011.

M. Paul, L. Held, and A. M. Toschke. Multivariate modelling of infectious disease surveillance data. *Statistics in medicine*, 27(29):6250–6267, 2008.

G. Qian and C. Field. Using mcmc for logistic regression model selection involving large number of candidate models. In *Monte Carlo and Quasi-Monte Carlo Methods 2000*, pages 460–474. Springer, 2002.

G. Qian, y. Wu, D. Ferrari, Q. Puxue, and F. Hollande. Semisupervised clustering by iterative partition and regression with neuroscience applications. *Computational Intelligence and Neuroscience*, 2016.

P. Qiao, C. Mølck, D. Ferrari, and F. Hollande. A spatio-temporal model and inference tools for longitudinal count data on multicolor cell growth. *The International Journal of Biostatistics*, 2018.

H. Quick, B. P. Carlin, and S. Banerjee. Heteroscedastic conditional auto-regression models for areally referenced temporal processes for analysing california asthma hospitalization data. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 64(5):799–813, 2015.

H. Quick, L. A. Waller, and M. Casper. Hierarchical multivariate space-time methods for modeling counts with an application to stroke mortality data. *arXiv preprint arXiv:1602.04528*, 2016.

H. Quick, L. A. Waller, and M. Casper. A multivariate space-time model for analysing county level heart disease death rates by race and sex. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 2017.

C. R. Rao, Y. Wu, and Q. Shao. An m-estimation-based procedure for determining the number of regression models in regression clustering. *Advances in Decision Sciences*, 2007, 2007.

R. A. Rigby and D. M. Stasinopoulos. Generalized additive models for location, scale and shape. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 54 (3):507–554, 2005.

A. Rushworth, D. Lee, and R. Mitchell. A spatio-temporal model for estimating the long-term effects of air pollution on respiratory hospital admissions in greater london. *Spatial and spatio-temporal epidemiology*, 10:29–38, 2014.

Ó. Sans, A. M. Schmidt, A. A. Nobre, et al. Bayesian spatio-temporal models based on discrete convolutions. *Canadian Journal of Statistics*, 36(2):239–258, 2008.

F. E. Satterthwaite. An approximate distribution of estimates of variance components. *Biometrics bulletin*, 2(6):110–114, 1946.

B. Schrödle, L. Held, and H. Rue. Assessing the impact of a movement network on the spatiotemporal spread of infectious diseases. *Biometrics*, 68(3):736–744, 2012.

G. Schwarz et al. Estimating the dimension of a model. *The annals of statistics*, 6(2): 461–464, 1978.

G. Shaddick and J. Wakefield. Modelling daily multivariate pollutant data at multiple sites. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 51(3): 351–372, 2002.

Q. Shao and Y. Wu. A consistent procedure for determining the number of clusters in regression clustering. *Journal of Statistical Planning and Inference*, 135(2):461–476, 2005.

M. Sklar. Fonctions de repartition an dimensions et leurs marges. *Publ. inst. statist. univ. Paris*, 8:229–231, 1959.

P. X.-K. Song, Y. Fan, and J. D. Kalbfleisch. Maximization by parts in likelihood inference. *Journal of the American Statistical Association*, 100(472):1145–1158, 2005.

P. X.-K. Song, M. Li, and Y. Yuan. Joint regression analysis of correlated data using gaussian copulas. *Biometrics*, 65(1):60–68, 2009.

P. X.-K. Song, M. Li, and P. Zhang. Vector generalized linear models: A gaussian copula approach. In *Copulae in Mathematical and Quantitative Finance*, pages 251–276. Springer, 2013.

D. P. Tabassum and K. Polyak. Tumorigenesis: it takes a village. *Nat Rev Cancer*, 15(8): 473–483, 2015.

C. Y. Tang, W. Zhang, and C. Leng. Discrete longitudinal data modeling with a mean-correlation regression approach. *Statistica Sinica*, 29(2):853–876, 2019.

R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.

R. Tibshirani and G. Walther. Cluster validation by prediction strength. *Journal of Computational and Graphical Statistics*, 14(3):511–528, 2005.

C. Varin and P. Vidoni. A note on composite likelihood inference and model selection. *Biometrika*, 92(3):519–528, 2005.

C. Varin, N. Reid, and D. Firth. An overview of composite likelihood methods. *Statistica Sinica*, pages 5–42, 2011.

W. N. Venables and B. D. Ripley. *Modern applied statistics with S-PLUS*. Springer Science & Business Media, 2013.

L. A. Waller, B. P. Carlin, H. Xia, and A. E. Gelfand. Hierarchical spatio-temporal mapping of disease rates. *Journal of the American Statistical association*, 92(438):607–617, 1997.

H. Wickham and W. Chang. devtools: Tools to make developing r packages easier; 2016. *URL https://CRAN. R-project. org/package= devtools. R package version*, 1(4):381, 2016.

C. K. Wikle and C. J. Anderson. Climatological analysis of tornado report counts using a hierarchical bayesian spatiotemporal model. *Journal of Geophysical Research: Atmospheres*, 108(D24), 2003.

C. K. Wikle, L. M. Berliner, and N. Cressie. Hierarchical bayesian space-time models. *Environmental and Ecological Statistics*, 5(2):117–154, 1998.

B. Wu and A. R. de Leon. Gaussian copula mixed models for clustered mixed outcomes, with application in developmental toxicology. *Journal of agricultural, biological, and environmental statistics*, 19(1):39–56, 2014.

P. Xue-Kun Song. Multivariate dispersion models generated from gaussian copula. *Scandinavian Journal of Statistics*, 27(2):305–320, 2000.

S. L. Zeger and B. Qaqish. Markov regression models for time series: a quasi-likelihood approach. *Biometrics*, pages 1019–1031, 1988.