University of Kentucky
UKnowledge

Theses and Dissertations--Statistics

Statistics

2012

Analysis of Binary Data via Spatial-Temporal Autologistic Regression Models

Zilong Wang
University of Kentucky, zilong.wang@uky.edu

Recommended Citation
Wang, Zilong, "Analysis of Binary Data via Spatial-Temporal Autologistic Regression Models" (2012). *Theses and Dissertations--Statistics*. Paper 3.
http://uknowledge.uky.edu/statistics_etds/3

This Doctoral Dissertation is brought to you for free and open access by the Statistics at UKnowledge. It has been accepted for inclusion in Theses and Dissertations--Statistics by an authorized administrator of UKnowledge. For more information, please contact UKnowledge@lsv.uky.edu.

STUDENT AGREEMENT:

I represent that my thesis or dissertation and abstract are my original work. Proper attribution has been given to all outside sources. I understand that I am solely responsible for obtaining any needed copyright permissions. I have obtained and attached hereto needed written permission statements(s) from the owner(s) of each third-party copyrighted matter to be included in my work, allowing electronic distribution (if such use is not permitted by the fair use doctrine).

I hereby grant to The University of Kentucky and its agents the non-exclusive license to archive and make accessible my work in whole or in part in all forms of media, now or hereafter known. I agree that the document mentioned above may be made available immediately for worldwide access unless a preapproved embargo applies.

I retain all other ownership rights to the copyright of my work. I also retain the right to use in future works (such as articles or books) all or part of my work. I understand that I am free to register the copyright to my work.

REVIEW, APPROVAL AND ACCEPTANCE

The document mentioned above has been reviewed and accepted by the student's advisor, on behalf of the advisory committee, and by the Director of Graduate Studies (DGS), on behalf of the program; we verify that this is the final, approved version of the student's dissertation including all changes required by the advisory committee. The undersigned agree to abide by the statements above.

Zilong Wang, Student

Dr. Yanbing Zheng, Major Professor

Dr. Constance Wood, Director of Graduate Studies

Analysis of Binary Data via Spatial-Temporal Autologistic Regression Models

---

DISSERTATION

A dissertation submitted in partial fulfillment of the
requirements for the degree of Doctor of Philosophy in the
College of Arts and Sciences
at the University of Kentucky

By
Zilong Wang
Lexington, Kentucky

Director: Dr. Yanbing Zheng and Dr. Richard Kryscio, Professor of Statistics
Lexington, Kentucky

2012

Copyright© Zilong Wang 2012

ABSTRACT OF DISSERTATION

Analysis of Binary Data via Spatial-Temporal Autologistic Regression Models

Spatial-temporal autologistic models are useful models for binary data that are measured repeatedly over time on a spatial lattice. However, the traditional parametrization presents difficulties in interpreting model parameters across varying levels of statistical dependence.

In order to overcome interpretable parameters, a centered spatial-temporal autologistic regression model has been developed. Two efficient statistical inference approaches, expectation-maximization pseudo-likelihood approach (EMPL) and Monte Carlo expectation-maximization likelihood approach (MCEML), have been proposed. Also, Bayesian inference is considered and studied. In addition, We consider the imputation of missing values is for spatial-temporal autologistic regression models. Most existing imputation methods are not admissible to impute spatial-temporal missing values, because they can disrupt the inherent structure of the data and lead to a serious bias during the inference or computing efficient issue. Two imputation methods, iteration-KNN imputation and maximum entropy imputation, are proposed, both of them are relatively simple and can yield reasonable results.

In summary, the main contributions of this dissertation are the development of a spatial-temporal autologistic regression model with centered parameterization, and proposal of EMPL, MCEML, and Bayesian inference to obtain the estimations of model parameters. Also, iteration-KNN and maximum entropy imputation methods have been presented for spatial-temporal missing data.

KEYWORDS: Autologistic regression models, Binary data, Imputation, Spatial-temporal process

Author's Signature: Zilong Wang

Date: December 10, 2012

Analysis of Binary Data via Spatial-Temporal Autologistic Regression Models

By
Zilong Wang

Director of Dissertation: Dr. Yanbing Zheng and Dr. Richard Kryscio

Director of Graduate Studies: Dr. Constance Wood

Date: December 10, 2012

ACKNOWLEDGMENTS

I would like to express my deep appreciations and gratitude to my advisors, Dr. Yanbing Zheng and Dr. Richard Kryscio, for their time, support, and mentorship during my research. Dr. Yanbing Zheng helped me to develop my research ability, scientific thinking, and analytical skills to a great extent.

I would like to acknowledge my committee members and former Directors of Graduate Studies, Dr. William Griffith, who recruited me in the Department of Statistics, and Dr. Arne Bathke, for their friendly guidance, thoughtful provoking suggestions during the whole Ph.D. life in University of Kentucky.

I would also like to thank other committee members, Dr. Svetla Slavova and Dr. Thomas Mueller for their valuable comments on my dissertation.

Many thanks to Dr. Arnold J. Stromberg, Dr. Mai Zhou, and Dr. Cidambi Srinivasan for the excellent statistical background they provided in teaching and work.

I am very grateful to my friends, Xiang Zhang, Yushun Lin, and Yifan Yang, for the help they provided during my dissertation research. Sincere thanks to all of my friends who have made my life full of joy. It is impossible to name all of you but you are always on my mind.

My appreciations also go to the Department of Statistics, which provided me with a good environment to study and research.

I would like to thank my parents, my father, who is now live happily in heaven, for your love, encourage, and belief in me; my mother, for your love, patience and support during my whole life.

Most importantly, I would like to thank my beloved wife, Lijuan Wu, who is always standing with me through good times and bad, for your support and love in our ten-year marriage. You are always my constant source of love, happiness, and strength all the years.

TABLE OF CONTENTS

Acknowledgments . . . iii

Table of Contents . . . v

List of Figures . . . vi

List of Tables . . . vii

Chapter 1 Introduction . . . 1
  1.1 Overview . . . 1
  1.2 Spatial-temporal autologistic regression models on Lattice . . . 4
  1.3 Statistical inference for autologistic models . . . 9
  1.4 Imputation methods for massive spatial-temporal missing data . . . 17
  1.5 Outline of the dissertation . . . 20

Chapter 2 Centered Spatial-temporal Autologistic Regression Model . . . 22
  2.1 Centered Spatial-Temporal Autologistic Regression Model . . . 24
  2.2 Parameter estimation and statistical inference . . . 26
  2.3 Simulation Studies . . . 31
  2.4 Application to the southern pine beetle data . . . 40

Chapter 3 Imputation Methods for Spatial-Temporal Data . . . 50
  3.1 Imputation methods . . . 50
  3.2 Simulation Study and Application . . . 57

Chapter 4 Summary and Discussion . . . 69

Appendices . . . 72
  A. Tables for centered spatial-temporal autologistic regression model . . . 72
  B. Tables for missing data imputation . . . 75
  C. R code for centered spatial-temporal autologistic regression model . . . 76
  D. R code for Imputation . . . 91

Bibliography . . . 100

Vita . . . 104

LIST OF FIGURES

1.1 Map of the 100 counties of North Carolina 5

2.1 Comparison of Marginal Means 34
2.2 Comparison of Monte Carlo Means 35
2.3 EMPLE of the intercept $\boldsymbol{\theta}_0$ 38
2.4 EMPLE of the coefficient of the covariate $\boldsymbol{\theta}_1$ 39
2.5 EMPLE of the spatial correlation coefficient $\boldsymbol{\theta}_2$ 40
2.6 EMPLE of the temporal correlation coefficient $\boldsymbol{\theta}_3$ 41
2.7 MCEML of the intercept $\boldsymbol{\theta}_0$ 42
2.8 MCEML of the coefficient of the covariate $\boldsymbol{\theta}_1$ 43
2.9 MCEML of the spatial correlation coefficient $\boldsymbol{\theta}_2$ 44
2.10 MCEML of the temporal correlation coefficient $\boldsymbol{\theta}_3$ 45
2.11 Annual southern pine beetle outbreaks from 1960 to 1996 in North Carolina 47
2.12 Map of mean fall precipitation in North Carolina 48

3.1 Comparison of imputation performance by imputation methods 63
3.2 Comparison of imputation performance by large scale probability 64
3.3 Map of the study area in the Chilcotin (Canada) 65
3.4 Map of the outbreaks MBP by year 66
3.5 Comparison of error rates 67
3.6 Comparison of imputation time 68

LIST OF TABLES

2.1 Critical Values for $\boldsymbol{\theta}_2$ and $\boldsymbol{\theta}_3$ 46
2.2 Comparison of computation time for centered model 46
2.3 Comparison of model parameter estimation for centered model 46
2.4 Comparison of model parameter estimation for traditional model 47
2.5 Comparison of the prediction performance 49

1 Simulation in 5*5 Lattice and 5 time points 72
2 Simulation in 10*10 Lattice and 5 time points 72
3 Simulation in 20*20 Lattice and 5 time points 73
4 Simulation in 5*5 Lattice and 10 time points 73
5 Simulation in 10*10 Lattice and 10 time points 73
6 Simulation in 20*20 Lattice and 10 time points 74
7 Simulation in 5*5 Lattice and 20 time points 74
8 Simulation in 10*10 Lattice and 20 time points 74
9 Simulation in 20*20 Lattice and 20 time points 75
10 Error Rates for Beta=(1,0.5,0.1,0.5) 75
11 Error Rates for Beta=(1,0.5,0.3,0.5) 75
12 Error Rates for Beta=(1,0.5,0.5,0.5) 75
13 APD for Beta=(1,0.5,0.1,0.5) 76
14 APD for Beta=(1,0.5,0.3,0.5) 76
15 APD for Beta=(1,0.5,0.5,0.5) 76
16 Error Rates for MPB Data 76
17 Computing Time for MPB Data (Unit in hour) 76

Chapter 1 Introduction

1.1 Overview

The values of spatial-temporal binary data are either 0 or 1, which are measured repeatedly over time on a spatial lattice, and derived from the presence or absence of a characteristic in the study. Binary data are more and more appeared in agriculture, biology, ecology, geography, epidemiology, finance, and image analysis disciplines recently. This dissertation focuses on spatial-temporal binary data observed on a lattice over time.

Autologistic regression models are useful tools for analyzing spatial-temporal binary data on a Markov random field, which account for effects of potential covariates and spatial-temporal dependence among the data simultaneously. For binary data on a spatial lattice, the traditional autologistic regression model was first introduced by Besag (1972, 1974), and it was extended to account for the effects of covariates by Gumpertz et al. (1997) and Huffer and Wu (1998). For binary data that are measured repeatedly over time on a spatial lattice, Zhu et al. (2005) and Zheng and Zhu (2008) generalized the traditional spatial-temporal autologistic regression model to account for covariates, spatial dependence, and temporal dependence simultaneously.

For statistical inference of spatail-temporal autologistic regression model, there are three widely used statistical approaches: maximum pseudo-likelihood approach (PL), Monte Carlo maximum likelihood approach (ML), and Bayesian inference. Maximum pseudo-likelihood approach was first proposed by Besag (1975). It is the fastest and most straightforward approach to obtian statistical inference, but it is statistical inefficient. Zhu et al. (2005) considered MPLE for statistical inference, but it may be statistically inefficient especially when spatial and/or temporal dependence is strong. Monte Carlo maximum likelihood approach and Bayesian inference were presented by Geyer (1974) and Møller et al. (2006), respectively. They are statistically efficient but require more computation demand. A fully Bayesian approach for both model parameter inference and prediction at future time points is proposed by Zheng and Zhu (2008).

However, in the presence of positive spatial and temporal dependence, the traditional models have non-zero spatial and/or temporal neighbors and the conditional expectation of binary response observation never decreases. This is unreasonable if most of the neighbors are zeros and could bias the realizations towards 1. Hence, the interpretation of regular parameterizations of a traditional spatial-temporal atuologistic regression model may not be straightforward and is difficult across varying levels of statistical dependence.

In order to solve the above problem, a centered spatial-temporal autologistic regression model has been proposed to analyze spatial-temporal binary data observed on a lattice over time in this dissertation. Expectation-maximization pseudolikelihood (EMPL) and Monte Carlo expectation-maximization likelihood approaches (MCEML) are developed for statistical inference. Also, Baysian inference is considered and studied. Furthermore, the statistical efficiency of the three approaches for various sizes of sampling lattices and numbers of sampling time points has been compared. Monte Carlo is used to obtain predictive distributions at future time points in term of prediction, and compared with the performance of the the traditional spatial-temporal autologistic regression model. The methodology is demonstrated via simulation studies and a real data example concerning southern pine beetle outbreak in North Carolina.

Missing data arise in the modern massive data analysis, and the problems lie in incorrect measurements, faulty equipment, and manual data entry errors, etc. For statistical analysis of missing data, the simplest way is to delete the data points with any missing values, but this is only valid for data missing completely at random (MCAR) cases when the data contains relatively small numbers of missing values by Little and Rubin (1987). The other ways are imputation methods to estimate the missing values based on learning algorithms for missing at random cases. Here we consider missing at random (MAR) cases and impute missing values to count for spatial and temporal effects in statistical analysis.

Spatial-temporal regression models are time-consuming because spatial and temporal effects are accounted for the statistical analysis process. The nearest neighbor and mean substitution are the simplest and commonly suggested ways to deal with this issue. However, these two imputation methods can disrupt the inherent structure of the data, and lead to a serious bias during the inference. On the other hand, most complex imputation methods, such as multiple imputation and Bayesian imputation, are not computationally efficient. Considering efficiency and accuracy, two new imputation methods are presented: iteration-KNN imputation and maximum entropy imputation. Both of them are relatively simple and can yield reasonable results.

Depending on research interests, both centered spatial-temporal autologistic regression model and new imputation methods are studied in this dissertation. In the following sections, autologistic regression models and imputation methods will be introduced, previous research on these topics will be reviewed, and our special goal and the final outline of the dissertation will be presented.

1.2 Spatial-temporal autologistic regression models on Lattice

In this section the autologistic regression models will be introduced, especially for the traditional autologistic model based on Besag (1972, 1974) and the traditional spatial-temporal autologistic regression model from Zheng and Zhu (2008). Before the introduction of these models, a brief description of spatial data, lattice and neighborhood will be provided.

1.2.1 Spatial data

Spatial data consist of measurements or observations taken at specific locations or within specific regions. The locations or regions can be in 1, 2 or 3 dimensions. For example, a segment along a lake is 1 dimensional, the surface of a lake is 2 dimensional, and the entire lake is 3 dimensional. In this dissertation, this terminology is specified to 2 dimensional space. According to Cressie (1993), spatial data can be categorized to three main types: geostatistical data, lattice data, and spatial point patterns data. The goals and approaches for the three types of spatial data in data analysis are a little different.

Geostatistical data are measurements taken at fixed locations. Usually, the locations are spatially continuous in the region. One example is the rainfall recorded at weather stations. Summarizing the spatial correlation and drawing inferences are the main goals of geostatistical data analysis. Kriging is a famous interpolation method based on linear least squares estimation algorithms, which is a fundamental tool and widely used in geostatistcal data analysis from 1970s (Cressie, 1993).

Before introducing lattice data, it is better to define and overview the terminology "lattice". Here the terminology lattice building on the spatial analysis refers to a countable collection of regular or irregular spatial sites, and is linked to the spatial neighborhood information by Cressie (1993). For a regular spatial lattice, the first order neighborhood contains the four nearest neighbors, the second order neighborhood contains the four second nearest neighbors, and so on. For example, figure 1.1 shows a map of the 100 counties of North Carolina numbered in alphabetical order; their seats form a lattice.

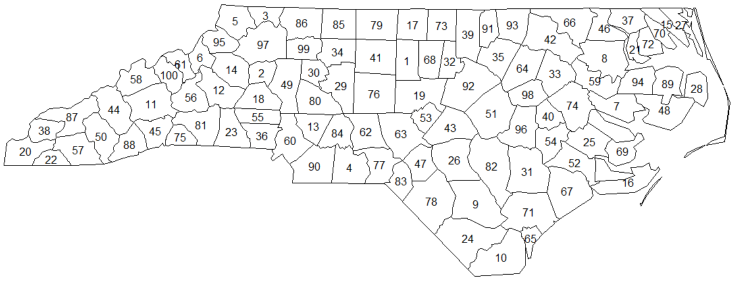

Figure 1.1: Map of the 100 counties of North Carolina numbered in alphabetical order.

The spatial lattice $\boldsymbol{D}_N$ of the ith county, and the neighborhood set N_i can be specified,

$$\boldsymbol{D}_N \equiv \{(i : N_i) : i = 1, ..., 100\}$$

$$N_i \equiv \{k : k \text{ is a spatial neighbor of } i\}, i = 1, ..., 100$$

For example, The county Alamance in North Carolina has site number 1 and its spatial neighbors (adjacent neighbors on a lattice) are {17, 19, 32, 41, 76}. Although the spatial lattice $\boldsymbol{D}_N$ does not contain exact site-location information, it is enough to build a model of spatial dependence between counties.

Lattice data are observations associated with spatial regions, where the regions can be regularly or irregularly spaced. The purpose of the analysis is to draw inferences and identify relationships among adjacent neighbors. Unlike geostatistical data, there is no possibility of a response between data locations. A typical example is southern pine beetle outbreaks (SPB), which shows presence or absence of a particular beetle in North Carolina.

Spatial point pattern data arise when locations themselves are the variables of interest. Spatial point patterns consist of a finite number of locations observed in a spatial region. For example, locations of lung caner cases in relation to the location of an incinerator. The objectives of a spatial point pattern are to identify, quantify and model the inherent spatial pattern among the data.

This dissertation focuses on spatial-temporal binary data that are measured repeatedly over time on a spatial lattice.

1.2.2 Autologistic models

Autologistic regression models will be introduced here. For traditional autologistic regression model and traditional spatial-temporal autologistic regression model, the previous research and development will be reviewed, and special studies for this dissertation will be outlined.

Traditional autologistic models

The traditional autologistic model was proposed by Besage (1972, 1974), unlike hierarchical models, it models spatial dependence among random variables directly and conditionally for binary data. In the last forty years it has proved to be a very useful model in many disciplines, particularly in ecology, environment, and epidemiology.

With $i = 1, ..., n$, let $\boldsymbol{s}_i$ denote the ith representative site on a spatial lattice, and let $\{N_i = j : \boldsymbol{s}_j \text{ is a neighbor of } \boldsymbol{s}_i\}$ denote the collection of sites that are spatial neighbors of $\boldsymbol{s}_i$ for a given neighborhood structure. Let $Y_i = Y(\boldsymbol{s}_i)$ denote the binary response variable at ith site such that $Y_i = 0$ or 1. Let $X_{0,i} \equiv 1$ and $X_{k,i} = X_k(\boldsymbol{s}_i)$ denote the kth explanatory variable at the ith site, where p denotes the number of explanatory variables. The full conditional distribution for the traditional autologistic model is given by,

$$\frac{p(Y_i = 1|Y_j : j \neq i)}{p(Y_i = 0|Y_j : j \neq i)} = \exp\{\sum_{k=0}^{p} \theta_k X_{k,i} Y_i + \theta_{p+1} \sum_{j \in N_i} Y_i Y_j\}$$

Then,

$$p(Y_i|Y_j : j \neq i) = p(Y_i|Y_j : j \in N_i) = \frac{\exp\{\sum_{k=0}^{p} \theta_k X_{k,i} Y_i + \sum_{j \in N_i} \theta_{p+1} Y_i Y_j\}}{1 + \exp\{\sum_{k=0}^{p} \theta_k X_k + \sum_{j \in N_i} \theta_{p+1} Y_j\}}$$

Where θ_0 is an intercept, θ_k is a slope for the kth covariate $X_{k,i}$, and θ_{p+1} is a spatial autoregressive coefficient.

Let $\boldsymbol{\theta} = (\theta_0, ..., \theta_p, \theta_{p+1})'$ denote the parameter vector of the model. Then the corresponding joint distribution is,

$$\begin{aligned}\mathcal{L}(\boldsymbol{\theta}) &= p(Y_1, \ldots, Y_n|\boldsymbol{\theta}) \\ &= c(\boldsymbol{\theta})^{-1} \exp\{\sum_{i=1}^{n} \sum_{k=0}^{p} \theta_k X_{k,i} Y_i + \frac{1}{2} \sum_{i=1}^{n} \sum_{j \in N_i} \theta_{p+1} Y_i Y_j\}\end{aligned}$$

Where $c(\boldsymbol{\theta})$ is a normalizing constant, which does not have an analytical form and usually creates a computational challenge for when using either maximum likelihood or Bayesian inference.

Traditional Spatial-temporal autologistic models

For binary data measured repeatedly over time on a spatial lattice, Zhu et al. (2005) generalized the autologistic regression models to account for covariates, spatial dependence, and temporal dependence simultaneously. Let $t \in \mathcal{Z}$ denote a set of time points, let $Y_{i,t} = Y(s_i, t)$ denote the binary response variable at the ith site s_i and the tth time point such that $Y_{i,t} = 0$ or 1. Let $X_{0,i,t} \equiv 1$ and $X_{k,i,t} = X_k(s_i, t)$ denote the kth explanatory variable at the ith site and tth time point. The traditional spatial-temporal autologistic regression model in Zheng and Zhu (2008) is defined via the following full conditional distribution,

$$\begin{aligned} & p(Y_{i,t} | Y_{i',t'} : (i',t') \neq (i,t)) = p(Y_{i,t} | Y_{i',t'} : (i',t') \in N_{i,t}) \\ & = \frac{\exp\{\sum_{k=0}^{p} \theta_k X_{k,i,t} Y_{i,t} + \sum_{j \in N_i} \theta_{p+1} Y_{i,t} Y_{j,t} + \theta_{p+2} Y_{i,t}(Y_{i,t-1} + Y_{i,t+1})\}}{1 + \exp\{\sum_{k=0}^{p} \theta_k X_{k,i,t} + \sum_{j \in N_i} \theta_{p+1} Y_{j,t} + \theta_{p+2}(Y_{i,t-1} + Y_{i,t+1})\}} \end{aligned} \tag{1.1}$$

Where $N_i = \{(j,t) : s_j \text{ is a spatial neighbor of } s_i\}$ and $N_{i,t} = \{(j,t) : j \in N_i\} \cup \{(i, t-1), (i, t+1)\}$ denote the spatial neighborhood and spatial-temporal neighborhood for the ith site and tth time point, respectively. Compared to the traditional autologistic model, one additional parameter θ_{p+2}, a temporal autoregressive coefficient, is included.

Let $\boldsymbol{Y}_t = (Y_{1,t}, ..., Y_{n,t})'$ denote the binary response on the entire spatial lattice for a given time point t and $\boldsymbol{Y}_1, ..., \boldsymbol{Y}_T$ denote binary responses measured at T time points. According to Hammersley-Clifford Theorem, the joint distribution of $\boldsymbol{Y}_2, ..., \boldsymbol{Y}_{T-1}$

conditioned on $\boldsymbol{Y}_1$ and $\boldsymbol{Y}_T$ is,

$$\begin{aligned}\mathcal{L}(\boldsymbol{\theta}) &= p(\boldsymbol{Y}_2, ..., \boldsymbol{Y}_{T-1}|\boldsymbol{Y}_1, \boldsymbol{Y}_T; \boldsymbol{\theta})\\ &= c(\boldsymbol{\theta})^{-1}\exp\{\sum_{t=2}^{T-1}(\sum_{i=1}^{n}\sum_{k=0}^{p}\theta_k X_{k,i,t}Y_{i,t} + \frac{1}{2}\sum_{i=1}^{n}\sum_{j\in N_i}\theta_{p+1}Y_{i,t}Y_{j,t})\\ &+ \sum_{t=2}^{T}\sum_{i=1}^{n}\theta_{p+2}Y_{i,t-1}\}\end{aligned}$$

Similarly to the traditional autologistic model, $c(\boldsymbol{\theta})$ is a normalizing constant which does not have an analytical form.

1.3 Statistical inference for autologistic models

Since the joint distribution of the autologistic regression model has a normalizing constant which involves the model parameters and does not have an analytical form, direct maximization of the likelihood function is not straightforward. There has been much research on statistical inference for autologistic models and such work is generally based on pseudo-likelihood, Markov chain Monte Carlo (MCMC) approximation of likelihood, and Bayesian hierarchical models. In particular, Besag (1975) proposed to maximize pseudo-likelihood functions. Huffer and Wu (1998) used MCMC to approximate the unknown normalizing constant and maximum likelihood estimates (MLE) for spatial autologistic models. Huang and Ogata (2002) generalized the pseduo-likelihood and proposed maximum generalized pseudo-likelihood estimates, which connect maximum pseudo-likelihood estimates (MPLE) and MLE and show better performance than MPLE in terms of standard errors and efficiencies relative to MLE. Zheng and Zhu (2008) cast the inference problem under a Bayesian hierarchical modeling framework and compared the performance of maximum pseudo-likelihood, MCMC maximum likelihood, and Bayesian inference. They demonstrated that parameter inference via maximum pseudo-likelihood is statistically inefficient especially when spatial and/or temporal dependence is strong, whereas the performance of the MCMC maximum likelihood is comparable to the Bayesian approach.

In addition to autologistic regression models, an alternative approach to analyze spatial-temporal binary data is marginal models using quasi-likelihood (QL) estimating equations for statistical inference, which allows separate modeling of regression and dependence of the response variables. Lin et al. (2009) developed an QL estimating equation for non-separable spatial-temporal binary data and compared the efficiencies of the QL estimates with MPLEs. Lin (2010) developed an QL estimating equation for separable spatial-temporal binary data.

In this section, a brief review of the maximum pseudo-likelihood approach, Monte Carlo maximum likelihood approach, and Bayesian inference will be presented. Since Monte Carlo samples are widely used in these statistical approaches, we will start with a basic introduction to two common Monte Carlo sampling methods: Gibbs sampling and perfect sampling.

1.3.1 Monte Carlo sampling methods

In this dissertation, Monte Carlo samples are generated using three types of sampling: Gibbs sampler after burn-in (BGS), perfect simulation (PS), and Gibbs sampler but start at a perfect simulation sample (PGS). They are developed based on Gibbs sampling and perfect sampling methods.

Gibbs sampling

To estimate the unknown normalizing constant in the joint distribution of autologistic regression models, we need to generate Monte Carlo random samples from the full conditional distribution. One of the important approaches is Gibbs sampling. The Gibbs sampling is a special case of the Metropolis-Hastings sampling, whose distinct feature is using conditional distributions to construct Markov chain moves at each iteration, instead of joint distribution. Thus, it is a powerful tool especially when the joint distribution is unknown or difficult to sample directly, but the conditional distribution of each variable is known and easy or easier to sample.

The Gibbs sampling algorithm generates an instance from the distribution of each variable in turn, conditional on the current values of the other variables. Thus, one simulates k random variables sequentially from the k conditionals compared to generate a single k-dimensional vector using the full joint distribution. The sequence of samples from Gibbs sampler consists of a Markov chain, and the stationary distribution of that Markov chain converges to the target joint distribution.

Same as other MCMC algorithms, Gibbs sampling generates a Markov chain of samples, each of which is correlated with nearby samples. As a result, every mth sample is taken to form an independent sample. In addition, samples from the beginning of the chain may not accurately represent the desired distribution, and must be thrown away ("burn-in"). In this dissertation, BGS are repeatedly used to generate Monte Carlo samples.

Perfect sampling

Perfect sampling (PS) is another important approach to generate Monte Carlo random samples in autologistic regression models. According to Propp and Wilson (1996) and Møller (1999), a perfect sampler for an autologistic model can be constructed as follows.

Let $\boldsymbol{L}_T(t,i)$ and $\boldsymbol{U}_T(t,i)$ denote the ith observations at time t of the lower and upper chains, respectively. These chains started at time T in the past with same simulation seeds. Fix $T < 0$ and set the lower chain $\boldsymbol{L}_T(t,*) = 0$ and upper chain $\boldsymbol{U}_T(t,*) = 1$. Update the chains according to

$$\boldsymbol{L}_T(t,i) = F_i^{-1}(\boldsymbol{R}(t,i))|\boldsymbol{L}_T(t,1:i-1), \boldsymbol{L}_T(t-1,(i+1):n)$$

$$\boldsymbol{U}_T(t,i) = F_i^{-1}(\boldsymbol{R}(t,i))|\boldsymbol{U}_T(t,1:i-1), \boldsymbol{U}_T(t-1,(i+1):n)$$

Where the $\boldsymbol{R}(t,i)$ are independent standard uniform variates and

$$F_i^{-1}(p) = \begin{cases} 1, \text{if}\, p > 1 - p_i \\ \\ 0, \text{if}\, p \leq 1 - p_i \end{cases}$$

with

$$p_i = p(Y_{i,t} = 1|\boldsymbol{\theta})$$

If $\boldsymbol{L}_T$ and $\boldsymbol{U}_T$ coalesce at time $t_0 \leq 0$, return $\boldsymbol{L}_T(0,*)$ as one sample from the joint distribution. Otherwise, double time T and start over. Use new uniform variates from $T, T+1, ..., \frac{T}{2} - 1$, but reuse the previously generated variates for time points $\frac{T}{2}, \frac{T}{2} + 1, ..., -1$.

Although perfect sampling requires more computational time than Gibbs sampling, it can guarantee that the sample is drawn from the exact target distribution during each iteration. Unlike Gibbs sampling, the sequential samples based on PS are from the target distribution and do not need to “burn-in”.

Based on the advantages and disadvantages of Gibbs sampling and perfect sampling, one combination of Monto Carlo sampling method is Gibbs sampler started at a perfect simulation sample (PGS). Its first sample is drawn from PS to guarantee that it is from the target distribution. Start from this sample, using Gibbs sampler to generate the other independent samples. The subsequent samples are also from the target distribution exactly. In this dissertation, some Monte Carlo samples are generated from PGS, especially when the spatial and/or temporal dependence is strong such that BGS is not working well.

After reviewing Monto Carlo sampling methods, the next goal is to investigate the parameter estimation and statistical inference approaches of autologistic regression models.

1.3.2 Maximum pseudolikelihood

Maximum pseudo-likelihood approach, first introduced by Besag (1975), is a popular and convenient way to obtain statistical inferences of autologistic regression models. The maximum pseudo-likelihood estimate (MPLE) is the value of $\boldsymbol{\theta}$ that maximizes the product of the conditional likelihoods. Based on the above traditional spatial-temporal regression model, it is as following,

$$\tilde{\boldsymbol{\theta}} = \text{argmax}\mathcal{L}_{\text{PL}}(\boldsymbol{\theta})$$

Where,

$$\begin{aligned}\mathcal{L}_{\text{PL}}(\boldsymbol{\theta}) &= \log\{\prod_{i,t} p(Y_{i,t}|Y_{i',t'} : (i',t') \neq (i,t))\}\\ &= \sum_{i,t} \log\{\frac{\exp\{\sum_{k=0}^{p} \theta_k X_{k,i,t} Y_{i,t} + \sum_{j\in N_i} \theta_{p+1} Y_{i,t} Y_{j,t} + \theta_{p+2} Y_{i,t}(Y_{i,t-1} + Y_{i,t+1})\}}{1 + \exp\{\sum_{k=0}^{p} \theta_k X_{k,i,t} + \sum_{j\in N_i} \theta_{p+1} Y_{j,t} + \theta_{p+2}(Y_{i,t-1} + Y_{i,t+1})\}}\}\end{aligned}$$

Although pseudo-likelihood is not the true likelihood except in the trivial case of independence, Besag (1975) showed that the MPLE converges almost surely to the

MLE as the lattice size goes to ∞.

To maximize the pseudo-likelihood function and obtain the maximum pseudo-likelihood estimate (MPLE) of $\boldsymbol{\theta}$, the easiest way is to use a standard logistic regression software function such as *proc logistic* in SAS or *glm* in R. Also, the standard error and approximate confidence intervals using a parametric bootstrap can be computed. That is, first generate M Monte Carlo samples from the target distribution using BGS, PS, or PGS, and compute the MPLE for each sample. After that, the M bootstrap samples are used to obtain the approximate variance of the MPLE, and construct corresponding approximate confidence interval, where parallel parametric bootstrap can greatly increase the efficiency of resampling process.

This statistical inference approach is the most efficient way in computation, but it is well known it may be statistical inefficient, especially when the spatial and/or temporal dependence is strong.

1.3.3 Monte Carlo maximum likelihood

Although, MPLE is straightforward and computationally efficient, it may be statistically inefficient especially in the existence of strong spatial and/or temporal dependence. An alternative approach is Monte Carlo maximum likelihood (MCML), which is direct maximization of likelihood function using Markov chain Monte Carlo (MCMC). It is statistically efficient but requires more computational time to simulate Monte Carlo samples.

Based on the above traditional spatial-temporal regression model, the likelihood function is,

$$\begin{aligned}
\mathcal{L}(\boldsymbol{\theta}) &= p(\boldsymbol{Y}_2, ..., \boldsymbol{Y}_{T-1}|\boldsymbol{Y}_1, \boldsymbol{Y}_T; \boldsymbol{\theta}) \\
&= c(\boldsymbol{\theta})^{-1} \exp\{\sum_{t=2}^{T-1}[\sum_{i=1}^{n}\sum_{k=0}^{p} \theta_k X_{k,i,t} Y_{i,t} + \frac{1}{2}\sum_{i=1}^{n}\sum_{j\in N_i} \theta_{p+1} Y_{i,t} Y_{j,t}] \\
&+ \sum_{t=2}^{T}\sum_{i=1}^{n} \theta_{p+1} Y_{i,t} Y_{i,t-1}\} \\
&= c(\boldsymbol{\theta})^{-1} \exp(\boldsymbol{\theta}' \boldsymbol{Z})
\end{aligned}$$

where,

$$\begin{aligned}
\boldsymbol{Z} =&(\sum_{t=2}^{T-1}\sum_{i=1}^{n} Y_{i,t}, \sum_{t=2}^{T-1}\sum_{i=1}^{n} X_{1,i,t} Y_{i,t}, \ldots, \sum_{t=2}^{T-1}\sum_{i=1}^{n} X_{p,i,t} Y_{i,t}, \\
&\sum_{t=2}^{T-1}\sum_{i=1}^{n} \frac{1}{2}\sum_{j\in N_i} Y_{i,t} Y_{j,t}, \sum_{t=2}^{T}\sum_{i=1}^{n} \theta_{p+1} Y_{i,t} Y_{i,t-1})'
\end{aligned}$$

$c(\boldsymbol{\theta})$ is an unknown normalizing constant in the sense that it can only be computed analytically for small lattice sizes.

Based on a preselected parameter vector $\boldsymbol{\psi} = (\psi_0, ..., \psi_{p+2})'$, generate M Monte Carlo samples from the joint distribution. Then the approximation of the following ratio of two normalizing constant is,

$$\begin{aligned}
\frac{c(\boldsymbol{\theta})}{c(\boldsymbol{\psi})} &= E_{\psi}[\frac{\exp(\boldsymbol{\theta}' \boldsymbol{Z})}{\exp(\boldsymbol{\psi}' \boldsymbol{Z})}] \\
&\approx M^{-1} \sum_{m=1}^{M} \frac{\exp(\boldsymbol{\theta}' \boldsymbol{Z}^m)}{\exp(\boldsymbol{\psi}' \boldsymbol{Z}^m)} \\
&= M^{-1} \sum_{m=1}^{M} \exp((\boldsymbol{\theta} - \boldsymbol{\psi})' \boldsymbol{Z}^m)
\end{aligned}$$

Where $\boldsymbol{Z}^m$ is $\boldsymbol{Z}$ evaluated at the mth Monte Carlo sample of $\boldsymbol{Y}$; $m = 1, ..., M$. Therefore MLE can be approximated by maximizing a rescaled version of the likelihood function,

$$c(\boldsymbol{\psi})\mathcal{L}(\boldsymbol{\theta}) = \frac{c(\boldsymbol{\psi})}{c(\boldsymbol{\theta})} \exp(\boldsymbol{\theta}' \boldsymbol{Z})$$

Because $c(\boldsymbol{\theta})$ is free of $\boldsymbol{\psi}$,

$$c(\boldsymbol{\psi})\mathcal{L}(\boldsymbol{\theta}) = [M^{-1}\sum_{m=1}^{M} \exp((\boldsymbol{\theta} - \boldsymbol{\psi})^{'}\boldsymbol{Z}^{m})]^{-1} \exp(\boldsymbol{\theta}^{'}\boldsymbol{Z})$$

Following Huffer and Wu (1998) and Geyer (1994), The variances can be approximated by using the diagonal elements of the observed Fisher information matrix.

1.3.4 Bayesian inference

Møller et al. (2006) presented an auxiliary-variable MCMC algorithm that allows us to construct a proposal distribution so that the normalizing constant cancels out in the Metropolis-Hastings ratio. Recently, Zheng and Zhu (2008) proposed a Bayesian approach for both model parameter inference and prediction at future time points using Markov chain Monte Carlo (MCMC). Here we describe this method following Zheng and Zhu (2008) for the traditional spatial-temporal regression model.

Let $P(\boldsymbol{\theta}|\boldsymbol{Y})$ denote the posterior distribution of $\boldsymbol{\theta}$ with a prior distribution $\pi(\boldsymbol{\theta})$, where $\boldsymbol{Y}$ denotes all the data. Consider Metropolis-Hastings (MH) algorithm to generate Monte Carlo samples for the parameter vector $\boldsymbol{\theta}$. Let $\boldsymbol{\theta}^{(0)}$ be a pre-selected initial parameter vector, $\boldsymbol{\theta} = \boldsymbol{\theta}^{l}$ be lth step parameter, and $\boldsymbol{\theta}^{*}$ be a new candidate for $(l+1)$th step parameter which is generated according to a proposal distribution $q(\boldsymbol{\theta}^{*}|\boldsymbol{\theta})$. Then the metropolis-Hastings random walk acceptance probability for the algorithm of Zheng and Zhu (2008) is given by,

$$\alpha(\boldsymbol{\theta}^{*}|\boldsymbol{\theta}) = \min\{\frac{\pi(\boldsymbol{\theta}^{*})p(\boldsymbol{Y}_{2},...,\boldsymbol{Y}_{T-1}|\boldsymbol{Y}_{1},\boldsymbol{Y}_{T};\boldsymbol{\theta}^{*})q(\boldsymbol{\theta}|\boldsymbol{\theta}^{*})}{\pi(\boldsymbol{\theta})p(\boldsymbol{Y}_{2},...,\boldsymbol{Y}_{T-1}|\boldsymbol{Y}_{1},\boldsymbol{Y}_{T};\boldsymbol{\theta})q(\boldsymbol{\theta}^{*}|\boldsymbol{\theta})}, 1\}$$

Now drawing a random number $U \sim$ Uniform$[0,1]$. Then at the $(l+1)$th step, $\theta_{l+1} = \theta^{*}$ if $\alpha \geqslant U$. Otherwise, $\theta_{l+1} = \theta$.

Now

$$\frac{p(\boldsymbol{Y}_2, ..., \boldsymbol{Y}_{T-1}|\boldsymbol{Y}_1, \boldsymbol{Y}_T; \boldsymbol{\theta}^*)}{p(\boldsymbol{Y}_2, ..., \boldsymbol{Y}_{T-1}|\boldsymbol{Y}_1, \boldsymbol{Y}_T; \boldsymbol{\theta})} = \frac{\frac{1}{c(\boldsymbol{\theta}^*)} \exp\{\boldsymbol{\theta}^{*'} \boldsymbol{Z}\}}{\frac{1}{c(\boldsymbol{\theta})} \exp\{\boldsymbol{\theta}' \boldsymbol{Z}\}}$$

$$= \frac{\exp\{\boldsymbol{\theta}^{*'} \boldsymbol{Z}\}}{\exp\{\boldsymbol{\theta}' \boldsymbol{Z}\}} \times \frac{c(\boldsymbol{\theta})}{c(\boldsymbol{\theta}^*)}$$

Consider a preselected parameter vector $\boldsymbol{\psi} = (\psi_0, ..., \psi_p + 2)'$, and generate M Monte Carlo samples from the joint distribution $p(\boldsymbol{Y}_2, ..., \boldsymbol{Y}_{T-1}|\boldsymbol{Y}_1, \boldsymbol{Y}_T; \boldsymbol{\psi})$,

$$\frac{p(\boldsymbol{Y}_2, ..., \boldsymbol{Y}_{T-1}|\boldsymbol{Y}_1, \boldsymbol{Y}_T; \boldsymbol{\theta}^*)}{p(\boldsymbol{Y}_2, \ldots, \boldsymbol{Y}_{T-1}|\boldsymbol{Y}_1, \boldsymbol{Y}_T; \boldsymbol{\theta})} = \frac{\exp\{\boldsymbol{\theta}^{*'} \boldsymbol{Z}\}}{\exp\{\boldsymbol{\theta}' \boldsymbol{Z}\}} \times \frac{\frac{c(\boldsymbol{\theta})}{c(\boldsymbol{\psi})}}{\frac{c(\boldsymbol{\theta}^*)}{c(\boldsymbol{\psi})}}$$

$$\approx \exp\{(\boldsymbol{\theta}^* - \boldsymbol{\theta})\boldsymbol{Z}\} \times \frac{\sum_{m=1}^{M} \exp((\boldsymbol{\theta} - \boldsymbol{\psi})' \boldsymbol{Z}^m)}{\sum_{m=1}^{M} \exp((\boldsymbol{\theta}^* - \boldsymbol{\psi})' \boldsymbol{Z}^m)}$$

For the MH algorithm, a good choice of the parameter vector $\boldsymbol{\psi}$ would speed up the convergence process. As $\boldsymbol{\psi}$ is closer to the posterior mode of $\boldsymbol{\theta}$, the results is better. Usually, the MPLE is a first choice for $\boldsymbol{\psi}$. However when the MPLE is far away from the true value of $\boldsymbol{\theta}$, the MH algorithm requires a large number of Monte Carlo samples to approximate the likelihood ratio by Sun (2004). Another way to obtain $\boldsymbol{\psi}$ is by a stochastic approximation algorithm from Gu and Zhu (2001). Furthermore, we need to adjust the variance of the proposal distribution to get a reasonable acceptance probability in the MH algorithm, if this acceptance probability is too low or high, the posterior distribution may not be proper.

1.4 Imputation methods for massive spatial-temporal missing data

Missing data arise in the modern massive spatial-temporal data analysis, and the problems lie in incorrect measurements, faulty equipment, and manual data entry errors, etc. For statistical analysis of missing data, the simplest way is to delete the data points with any missing values. However, this strategy maybe invalid to spatial-temporal data analysis.

In order to choose proper imputation methods for spatial-temporal data, it is important to understand why the data are missing.

1.4.1 Missing data mechanisms

For spatial-temporal data, assume that a sequence of measurements $Y_{i,t}$ are designed to be meansured at site $i = 1, ..., n$ over time point $t = 1, ..., T$. Let $\boldsymbol{Y} = (Y_{1,1}, ..., Y_{1,T}, Y_{2,1}, ..., Y_{n,T})^{'}$. Also partition $Y_{i,t}$ into observed and missing categories as $Y_{i,t}^{o}$ and $Y_{i,t}^{m}$,

$$Y_{i,t} = \begin{cases} \text{observed data}\, Y_{i,t}^{o} \\ \\ \text{missing data}\, Y_{i,t}^{m} \end{cases}$$

In addition, the missing data indicator $R_{i,t}$ is defined by,

$$R_{i,t} = \begin{cases} 1, \text{if}\, Y_{i,t}\, \text{is observed} \\ \\ 0, \text{otherwise} \end{cases}$$

Let $\boldsymbol{R} = (R_{1,1}, ..., R_{1,T}, R_{2,1}, ..., R_{n,T})^{'}$ Therefore, the full data is $(\boldsymbol{Y}, \boldsymbol{R})$ which is the complete data together with the missing indicators.

There are three types of missing data by Rubin (1976) and Little and Rubin (1987): missing completely at random (MCAR), missing at random (MAR), and missing not at random (MNAR). MCAR exists when missing is independent of both the unobserved (missing) and observed measurements. It is ignorable missing, and the model of $R_{i,t}$ (missing data indicator) does not contain information about parameters of interest. MAR exists when missing is only independent of unobserved measurements, but depends on observed measurements. It is ignorable missing, but the pattern of missing is traceable or predictable. MNAR exists when missing depends on both the unobserved and observed measurements. It is non-ignorable missing with not traceable or predictable, and model for $R_{i,t}$ does contain information about parameters of interest.

1.4.2 Imputation overview

For statistical analysis of missing data, the simplest way is to delete the data points with any missing values. However, according to Litter and Rubin (1987) this method is only valid under MCAR when the data contain relatively small numbers of missing values. Alternative ways are using imputation methods to estimate the missing values based on learning algorithms for MAR case, for example, expectation-maximization (EM) imputation by Dempster, Laird and Rubin (1977), mean imputation, conditional mean imputation (Buck's Method), hot deck imputation, and multiple imputation by Little and Rubin (1987), sequential imputations and Bayesian imputation by Kong, Liu and Wong (1994), K-nearest neighbor (KNN) imputation by Batista and Monard (2003), support vector machine (SVM) imputation by Pelckmansa and Brabanter (2005) ect.

Here we consider the MAR case for spatial-temporal data and impute missing values to estimate spatial and temporal effects in statistical analysis. The nearest neighbor or mean substitution are the simplest and commonly suggested ways to deal with this issue and are still used in many statistical software packages. However, these two imputation methods can disrupt the inherent structure of the data, and lead to a serious bias during the inference (Kim et al 2004). Moreover, most complex imputation methods, such as multiple imputation and Bayesian imputation, are not admissible for computing efficiency issue. Recently, several imputation methods have been applied to the imputation of spatial-temporal massive missing data, including EM imputation Smith, Kolenikov, and Cox (2003) and KNN-based imputation (KNN imputation, weighted KNN imputation, Sequential KNN imputation, etc.) by Crookston and Finley (2008) and Meesad and Hengpraprohm (2008). In EM imputation method, the procedure consists of iterations of the model based EM algorithm where the conditional means and covariance matrices are estimated iteratively. KNN-based imputation method is developed from hot deck imputation method, and uses K nearest neighbor observations and KNN-based algorithms to estimate missing values. In general, the recently developed KNN-based imputation method is most efficient, and EM imputation method is most accurate by Weeks (2001).

Considering efficiency and accuracy, we propose two new imputation methods, iteration-KNN imputation and maximum entropy imputation, for spatial-temporal massive missing data in this dissertation. Iteration-KNN imputation uses a KNN imputation repeatedly with EM-style algorithm to improve accuracy with high computing speed. Maximum entropy imputation estimates missing values based on regression model with maximum entropy. Both of them are relatively simple and can yield reasonable results. We evaluate the efficiency and accuracy of these mthods through comparison with mean substitution, KNN, and EM imputation across both different missing rates and large scale probability in simulation data.

1.5 Outline of the dissertation

The remainder of this dissertation is divided into two major parts: centered spatial-temporal autologistic regression model and missing data imputation methods, which are organized as follows. In chapter 2, a centered spatial-temporal autologistic regression model is developed in section 1. In section 2, new estimation and statistical inference approaches are proposed. And a simulation study is conducted in section 3, followed by a real data example in section 4. In Chapter 3, the iteration-KNN and maximum entropy imputation methods, as well as KNN and EM imputation methods are introduced in section 1. Simulation study is designed to investigate the efficiency and accuracy across both different missing rates and large scale probability in section 2. Imputation methods are applied to a real data example in section 3. Finally, conclusion and discussion are presented in chapter 4.

Copyright© Zilong Wang, 2012.

Chapter 2 Centered Spatial-temporal Autologistic Regression Model

Parameters interpretation problem

the interpretation of model parameters for a traditional spatial-temporal autologistic may not be straightforward when incorporating regression.

In the presence of positive spatial and temporal dependence, under the parameterizations in the traditional spatial-temporal autologistic regression model, the conditional expectation of $Y_{i,t}$ given its neighbors is,

$$\begin{aligned} &E(Y_{i,t}|Y_{i',t'} : (i^{'}, t^{'} \in N_{i,t})) \\ &= \frac{\exp\{\sum_{k=0}^{p} \theta_k X_{k,i,t} + \sum_{j \in N_i} \theta_{p+1} Y_{j,t} + \theta_{p+2}(Y_{i,t-1} + Y_{i,t+1})\}}{1 + \exp\{\sum_{k=0}^{p} \theta_k X_{k,i,t} + \sum_{j \in N_i} \theta_{p+1} Y_{j,t} + \theta_{p+2}(Y_{i,t-1} + Y_{i,t+1})\}} \end{aligned}$$

which increases over,

$$\frac{\exp\{\sum_{k=0}^{p} \theta_k X_{k,i}\}}{1 + \exp\{\sum_{k=0}^{p} \theta_k X_{k,i}\}}$$

the expectation of $Y_{i,t}$ under independence, as long as $Y_{i,t}$ has non-zero spatial and/or temporal neighbors and never decreases. This is unreasonable if most of the neighbors are zeros and could bias the realizations towards 1. Hence, the interpretation of parameters is difficult across varying levels of statistical dependence.

For non-Gaussian Markov random field models of spatial lattice data, the idea of centered parameterization was first proposed by Kaiser and Cressie (1997), who considered a Winsorized Poisson conditional model. Recently, Kaiser and Caregea (2009) explored the centered parameterization for general exponential family of Markow random field models. In particular, Caragea and Kaiser (2009) studied the centered parameterization for spatial atuologistic regression models and showed that the centered parameterization overcomes the interpretation difficulties.

To solve this parameter interpretation problem, a centered spatial-temporal autologistic model is developed for analyzing spatial-temporal binary data observed on a lattice over time in this dissertation. Moreover, expectation-maximization pseudo-likelihood (EMPL) and Monte Carlo expectation-maximization likelihood (MCEML) have been proposed for statistical inference of model parameters. Also, Bayesian inference is considered and studied.

Recently, Huges et al. (2011) explored the performance of these inference approaches under the centered parameterization of spatial-only autologistic regression models and showed that when the spatial lattice is large enough, maximum pseudo-likelihood provides reliable inference for moderate spatial dependence.

Here we propose Expectation maximization pseudo-likelihood (EMPL) and Monte Carlo expectation-maximization likelihood (MCEML) for statistical inference of model parameters. The performance of Bayesian inference and further comparison of the efficiency of these inference approaches for various sizes of sampling lattices and numbers of sampling time points through both a simulation study and a real data example is also studied. Furthermore, for spatial-temporal data, prediction into the future is of interest. We use Monte Carlo to obtain predictive distributions at future time points and compare the forecasting performance between the models with uncentered and centered parameterization.

2.1 Centered Spatial-Temporal Autologistic Regression Model

Under a regularity condition of pairwise-only dependence, a centered spatial-temporal autologistic regression model is defined with the full conditional distribution,

$$\begin{aligned} &p(Y_{i,t}|Y_{i',t'} : (i',t') \neq (i,t)) = p(Y_{i,t}|Y_{i',t'} : (i',t') \in N_{i,t}) \\ &= \frac{\exp\{\sum_{k=0}^{p} \theta_k X_{k,i,t} Y_{i,t} + \sum_{j\in N_i} \theta_{p+1} Y_{i,t} Y^*_{j,t} + \theta_{p+2} Y_{i,t}(Y^*_{i,t-1} + Y^*_{i,t+1})\}}{1 + \exp\{\sum_{k=0}^{p} \theta_k X_{k,i,t} + \sum_{j\in N_i} \theta_{p+1} Y^*_{j,t} + \theta_{p+2}(Y^*_{i,t-1} + Y^*_{i,t+1})\}} \end{aligned} \tag{2.1}$$

where $Y^*_{i,t}$ denotes the centered response for the ith site and tth time point,

$$Y^*_{i,t} = Y_{i,t} - p_{i,t}$$

and the center $p_{i,t}$ is the probability of $Y_{i,t} = 1$ under independence,

$$p_{i,t} = \frac{\exp\{\sum_{k=0}^{p} \theta_k X_{k,i,t}\}}{1 + \exp\{\sum_{k=0}^{p} \theta_k X_{k,i,t}\}}. \tag{2.2}$$

Thus, the conditional expectation of $Y_{i,t}$ given its neighbors is,

$$\begin{aligned} &E(Y_{i,t}|Y_{i',t'} : (i',t') \in N_{i,t}) \\ &= \frac{\exp\{\sum_{k=0}^{p} \theta_k X_{k,i,t} + \sum_{j\in N_i} \theta_{p+1} Y^*_{j,t} + \theta_{p+2}(Y^*_{i,t-1} + Y^*_{i,t+1})\}}{1 + \exp\{\sum_{k=0}^{p} \theta_k X_{k,i,t} + \sum_{j\in N_i} \theta_{p+1} Y^*_{j,t} + \theta_{p+2}(Y^*_{i,t-1} + Y^*_{i,t+1})\}} \end{aligned}$$

Suppose both the spatial autoregressive coefficient θ_{p+1} and the temporal autoregressive coefficient θ_{p+2} are positive. Compare the conditional expectation of the centered model with the expectation of the independence model, as following,

$$E(Y_{i,t}|Y_{i',t'} : (i',t') \in N_{i,t}) > p_{i,t}$$

when

$$\theta_{p+1} \sum_{j\in N_i} Y_{j,t} + \theta_{p+2}(Y_{i,t-1} + Y_{i,t+1}) > \theta_{p+1} \sum_{j\in N_i} p_{j,t} + \theta_{p+2}(p_{i,t-1} + p_{i,t+1})$$

where $\sum_{j\in N_i} p_{j,t}$ and $p_{i,t-1} + p_{i,t+1}$ are the expected numbers of non-zero spatial and temporal neighbors under the independence model (2.2), respectively.

Specifically, if $\theta_{p+2} = 0$, under the situation that the observed number of non-zero spatial neighbors is greater than the expected number of non-zero spatial neighbors under independence, i.e., $\sum_{j \in N_i} Y_{j,t} > \sum_{j \in N_i} p_{j,t}$, the conditional expectation of $Y_{i,t}$ increases over $p_{i,t}$, the expectation under independence. Similarly, if $\theta_{p+1} = 0$, then the conditional expectation of $Y_{i,t}$ increases over $p_{i,t}$ only when the observed number of non-zero temporal neighbors is greater than the expected number of non-zero temporal neighbors under independence, i.e. $Y_{i,t-1} + Y_{i,t+1} > p_{i,t-1} + p_{i,t+1}$.

By HammersleyClifford Theorem (Cressie 1993), the joint distribution of the spatial-temporal process $\{Y_{i,t}\}$ specified by the conditional distributions (1) is well-defined, the joint likelihood function of $Y_2, ..., Y_{T-1}$ conditioned on Y_1 and Y_T is defined as following,

$$\begin{aligned}
&\mathcal{L}(\boldsymbol{\theta}) = p(\boldsymbol{Y}_2, ..., \boldsymbol{Y}_{T-1} | \boldsymbol{Y}_1, \boldsymbol{Y}_T; \boldsymbol{\theta}^*) \\
&= c^*(\boldsymbol{\theta})^{-1} \exp\{\sum_{t=2}^{T-1} [\sum_{i=1}^{n} \sum_{k=0}^{p} \theta_k X_{k,i,t} Y^*_{i,t} + \frac{1}{2} \sum_{i=1}^{n} \sum_{j \in N_i} \theta_{p+1} Y^*_{i,t} Y^*_{j,t}] \\
&+ \sum_{t=2}^{T} \sum_{i=1}^{n} \theta_{p+1} Y^*_{i,t} Y^*_{i,t-1}\} \\
&= c^*(\boldsymbol{\theta})^{-1} \exp(\boldsymbol{\theta}' \boldsymbol{Z}^*_\theta)
\end{aligned} \tag{2.3}$$

where,

$$\begin{aligned}
\boldsymbol{Z}^*_\theta = \{(&\sum_{t=2}^{T-1} \sum_{i=1}^{n} Y^*_{j,t}, \sum_{t=2}^{T-1} \sum_{i=1}^{n} X_{1,i,t} Y^*_{i,t}, \ldots, \sum_{t=2}^{T-1} \sum_{i=1}^{n} X_{p,i,t} Y^*_{i,t}, \\
&\sum_{t=2}^{T-1} \sum_{i=1}^{n} \frac{1}{2} \sum_{j \in N_i} Y^*_{i,t} Y_{j,t}, \sum_{t=2}^{T} \sum_{i=1}^{n} Y^*_{i,t} Y^*_{i,t-1})\}'
\end{aligned}$$

Similar to the uncentered parameterization, where $c^*_{\boldsymbol{\theta}}$ is a normalizing constant without a closed form.

2.2 Parameter estimation and statistical inference

For statistical inference in the centered model, expectation-maximization pseudolikelihood (EMPL) and Monte Carlo expectation-maximization likelihood approaches (MCEML)are proposed to estimate maximum pseudo-likelihood and MCMC maximum likelihood, respectively. Also, Bayesian inference is considered and studied. On the other hand, because the model parameters $\boldsymbol{\theta}$ is involved in the equations (2.1) and (2.3) for the centered model, the parameter inference is computationally more intensive than that for the model with uncentered parameterization. For the prediction, a predictive distribution is defined similarly to Zheng and Zhu (2008).

2.2.1 Expectation-maximization pseudo-likelihood estimator

The pseudo-likelihood function is the product of the full conditional distributions (2.1), and MPLE is the estimate of $\boldsymbol{\theta}$ that maximizes the pseudo-likelihood function. Guyon (1995) pointed out that MPLE are consistent and asymptotically normal under suitable regularity conditions. However, MPLE may be statistically inefficient when the spatial dependence and/or temporal dependence is strong (see, e.g., Gumpertz et al. 1997; Wu and Huffer 1997; Zheng and Zhu 2008; Huges et al. 2011). EMPL is proposed to obtain the MPLE for the centered model, which is a combination of an expectation-maximization (EM) algorithm and a NewtonRaphson algorithm. The EMPL algorithm proceeds as follows.

- **Step 0:** Start from a preselected $\boldsymbol{\theta}_0$ and set $\hat{\boldsymbol{\theta}}^0 = \boldsymbol{\theta}_0$.

- **E (expectation) step:** Given $\hat{\boldsymbol{\theta}}^{l-1}$

(1) Compute $p_{i,t}^{l-1}$, the expectation of $Y_{i,t}$ under the independent logistic regression model.

(2) Compute $Y_{i,t}^{*(l-1)} = Y_{i,t} - p_{i,t}^{l-1}$, the centered responses for lth iteration.

- **M (Maximization) step:** Obtain $\hat{\boldsymbol{\theta}}^{l}$ by maximizing,

$$\log\{\prod_{i,t} p(Y_{i,t}|Y_{i^{'},t^{'}} : (i^{'},t^{'}) \in N_{i,t}; \hat{\boldsymbol{\theta}}^{l-1})\}$$

$$= \sum_{i,t} \log\{\frac{\exp\{\sum_{k=0}^{p} \theta_k X_{k,i,t} Y_{i,t} + \sum_{j \in N_i} \theta_{p+1} Y_{i,t} Y_{j,t}^{*(l-1)} + \theta_{p+2} Y_{i,t}(Y_{i,t-1}^{*(l-1)} + Y_{i,t+1}^{*(l-1)})\}}{1 + \exp\{\sum_{k=0}^{p} \theta_k X_{k,i,t} + \sum_{j \in N_i} \theta_{p+1} Y_{j,t}^{*(l-1)} + \theta_{p+2}(Y_{i,t-1}^{*(l-1)} + Y_{i,t+1}^{*(l-1)})\}}\}$$

This step can be carried out by a Newton-Raphson algorithm using standard logistic regression software function.

- **Convergence criteria**

Repeat step **E** and **M** until $|\hat{\boldsymbol{\theta}}^{l} - \hat{\boldsymbol{\theta}}^{l-1}| < \delta$, then $\hat{\boldsymbol{\theta}} = \hat{\boldsymbol{\theta}}^{l}$.

where $\hat{\boldsymbol{\theta}}$ is the EMPL estimate (EMPLE) of $\boldsymbol{\theta}$, and δ is a preselected precision parameter.

The standard error of EMPLE can be computed using a parametric bootstrap. That is, M Monte Carlo samples of spatial-temporal binary responses are generated from the likelihood function evaluated at the EMPLE using a Gibbs sampler starting at a perfect simulation sample (PGS). Then, the EMPLE of each sample $\tilde{\boldsymbol{\theta}}^{m}, m = 1, ..., M$ can be computed. These resampled EMPLEs consist of the bootstrap sample $\tilde{\boldsymbol{\theta}} = \{\tilde{\boldsymbol{\theta}}^{1}, ..., \tilde{\boldsymbol{\theta}}^{M}\}$ and are used to approximate the standard error of the EMPLE based on the original data. Also, the approximate confidence interval of EMPLE based on the original data can be obtained from the quantiles of the bootstrap sample $\tilde{\boldsymbol{\theta}}$. For PGS, PS is used to generate the first Monte Carlo sample and guarantee that it is from the target monotone centered spatial-temporal autologistic regression model exactly (e.g., Propp and Wilson 1996; Møller 1999). By starting a Gibbs sampler at a perfect simulation sample, the chain starts at the equilibrium and then the subsequent samples are also from the target distribution exactly (Zheng and Zhu 2008).

For the starting value $\boldsymbol{\theta}_0$ at step 0, maximum pseudo-likelihood estimate from the traditional spatial-temporal autologistic regression model would be a natural choice. Different starting points would affect the time of convergence, data inherent structure, and data size. The computing time rapidly increases as the distance between the staring point and true value increases. For the initial value $\boldsymbol{\theta}_0$ at step 0, the estimate of the model parameter under the independent logistic regression model would be a good choice.

2.2.2 Monte Carlo expectation-maximization likelihood estimator

Let $\boldsymbol{\psi} = (\psi_0, ..., \psi_{p+2})'$ be a reference parameter for the centered model, and $\boldsymbol{Z}^*_{\boldsymbol{\psi}}$ is $\boldsymbol{Z}^*$ with centers evaluated at $\boldsymbol{\psi}$. The rescaled version of the likelihood function is as following,

$$c^*(\boldsymbol{\psi})\mathcal{L}(\boldsymbol{\theta}) = \frac{c^*(\boldsymbol{\psi})}{c^*(\boldsymbol{\theta})}\exp(\boldsymbol{\theta}'\boldsymbol{Z}^*_{\boldsymbol{\theta}}) = \exp(\boldsymbol{\theta}'\boldsymbol{Z}^*_{\boldsymbol{\theta}})\{E_{\boldsymbol{\psi}}[\frac{\exp(\boldsymbol{\theta}'\boldsymbol{Z}^*_{\boldsymbol{\theta}})}{\exp(\boldsymbol{\theta}'\boldsymbol{Z}^*_{\boldsymbol{\psi}})}]\}^{-1} \tag{2.4}$$

By generating M Monte Carlo samples of $\boldsymbol{Y}$ from the likelihood function evaluated at $\boldsymbol{\psi}$, we have,

$$E_{\boldsymbol{\psi}}[\frac{\exp(\boldsymbol{\theta}'\boldsymbol{Z}^*_{\boldsymbol{\theta}})}{\exp(\boldsymbol{\theta}'\boldsymbol{Z}^*_{\boldsymbol{\theta}})}] \approx M^{-1}\sum_{m=1}^{M}\exp(\boldsymbol{\theta}'\boldsymbol{Z}^{*(m)}_{\boldsymbol{\theta}} - \boldsymbol{\psi}'\boldsymbol{Z}^{*(m)}_{\boldsymbol{\psi}}) \tag{2.5}$$

By (2.4) and (2.5), an MCMC approximate of the rescaled version of likelihood is as following,

$$c^*(\boldsymbol{\psi})\mathcal{L}(\boldsymbol{\theta}) \approx \exp(\boldsymbol{\theta}'\boldsymbol{Z}^*_{\boldsymbol{\theta}})[M^{-1}\sum_{m=1}^{M}\exp(\boldsymbol{\theta}'\boldsymbol{Z}^{*(m)}_{\boldsymbol{\theta}} - \boldsymbol{\psi}'\boldsymbol{Z}^{*(m)}_{\boldsymbol{\psi}})]^{-1} \tag{2.6}$$

Based on (2.6), MCEML estimator by combining an EM algorithm and a Newton-Raphson algorithm is developed as following.

- **Step 0:** Start from a preselected $\boldsymbol{\theta}_0$ and set $\hat{\boldsymbol{\theta}}^0 = \boldsymbol{\theta}_0$.

(1) Choose a reference parameter vector $\boldsymbol{\psi}$, and generate M Monte Carlo samples of $\boldsymbol{Y}$ from the likelihood function evaluated at $\boldsymbol{\psi}$.

(2) Compute $p_{i,t,\boldsymbol{\psi}}$, the expectation of $Y_{i,t}$ under the independent logistic regression

model evaluated at $\boldsymbol{\psi}$.

(3) Compute $Y^*_{i,t,m} = Y_{i,t,m} - p_{i,t,\boldsymbol{\psi}}, m = 1, ..., M$, the centered responses for M Monte Carlo samples evaluated at $\boldsymbol{\psi}$.

- **E (expectation) step:** Given $\hat{\boldsymbol{\theta}}^{l-1}$

(1) Compute $p^{l-1}_{i,t}$, the expectation of $Y_{i,t}$ under the independent logistic regression model.

(2) Compute $Y^{*(l-1)}_{i,t} = Y_{i,t} - p^{l-1}_{i,t}$, the centered responses for lth iteration.

(3) Compute $Y^{*(l-1)}_{i,t,m} = Y_{i,t,m} - p^{l-1}_{i,t}$, the centered responses for M Monte Carlo samples (generated at step 0) at lth iteration.

- **M (Maximization) step:** Obtain $\hat{\boldsymbol{\theta}}^{l}$ by maximizing the following function,

$$\exp(\boldsymbol{\theta}' \boldsymbol{Z}^*_{\hat{\boldsymbol{\theta}}^{l-1}})[M^{-1} \sum_{m=1}^{M} \exp(\boldsymbol{\theta}' \boldsymbol{Z}^{*(m)}_{\hat{\boldsymbol{\theta}}^{l-1}} - \boldsymbol{\psi}' \boldsymbol{Z}^{*(m)}_{\boldsymbol{\psi}})]^{-1}$$

Where $\boldsymbol{Z}^*_{\hat{\boldsymbol{\theta}}^{l-1}}$ is $\boldsymbol{Z}^*$ with centered responses $Y^{*(l-1)}_{i,t}$, and $\boldsymbol{Z}^{*(m)}_{\hat{\boldsymbol{\theta}}^{l-1}}$ and $\boldsymbol{Z}^{*(m)}_{\boldsymbol{\psi}}$ are $\boldsymbol{Z}^*$ evaluated at the mth Monte Carlo sample of $\boldsymbol{Y}$ (generated at step 0) with centered responses $Y^{*(l-1)}_{i,t,m}$ and $Y^*_{i,t,m}$, respectively.

This step can be carried out using a Newton-Raphson algorithm.

- **Convergence criteria**

Repeat step **E** and **M** until $\left|\hat{\boldsymbol{\theta}}^{l} - \hat{\boldsymbol{\theta}}^{l-1}\right| < \delta$, then $\hat{\boldsymbol{\theta}} = \hat{\boldsymbol{\theta}}^{l}$.

where $\hat{\boldsymbol{\theta}}$ is the MCEML estimate (MCEMLE) of $\boldsymbol{\theta}$, and δ is a preselected precision parameter.

The Fisher information matrix of the original data is approximated as the by product of the MCEML estimation, the standard error of MCEMLE is obtained from the diagonal of the matrix.

The MCEMLE provides a good approximation to the maximum likelihood estimate (MLE) of model parameters when the reference parameter $\boldsymbol{\psi}$ is close to the true value

(Geyer and Thompson 1992). The EMPLE is a natural choice for the reference parameter. However, when the spatial/temporal dependence is strong, EMPLE can be far away from the true value. Under this situation, EMPLE is not a proper reference parameter, the iteration leads to a sequence of estimates that drift off to infinity. Alterative reference parameter can be an approximation obtained by a stochastic approximation algorithm (see, e.g., Gu and Zhu 2001; Zheng and Zhu 2008). For the initial value $\boldsymbol{\theta}_0$ at step 0, EMPLE would be a good choice.

2.2.3 Bayesian inference

For Bayesian inference, Monte Carlo samples of $\boldsymbol{\theta}$ are generated from the posterior distribution $p(\boldsymbol{\theta}|\boldsymbol{Y})$ using MetropolisHastings (MH) algorithm. The metropolis-Hastings random walk acceptance probability is computed as,

$$\alpha(\boldsymbol{\theta}^*|\boldsymbol{\theta}) = \min\{\frac{\pi(\boldsymbol{\theta}^*)p(\boldsymbol{Y}_2, ..., \boldsymbol{Y}_{T-1}|\boldsymbol{Y}_1, \boldsymbol{Y}_T; \boldsymbol{\theta}^*)q(\boldsymbol{\theta}|\boldsymbol{\theta}^*}{\pi(\boldsymbol{\theta})p(\boldsymbol{Y}_2, ..., \boldsymbol{Y}_{T-1}|\boldsymbol{Y}_1, \boldsymbol{Y}_T; \boldsymbol{\theta})q(\boldsymbol{\theta}^*|\boldsymbol{\theta}}, 1\}$$

where $\pi(\boldsymbol{\theta})$ denotes a prior distribution for $\boldsymbol{\theta}$ and $q(\boldsymbol{\theta}|\boldsymbol{\theta}^*)$ denotes a proposal distribution , which is set to be a normal distribution with mean $\boldsymbol{\theta}$ and and diagonal variance matrix $\Sigma = \text{diag}\{\sigma_0^2, \sigma_1^2, ..., \sigma_p^2, \sigma_{p+1}^2, \sigma_{p+2}^2\}$ in the analysis. With a preselected reference parameter $\boldsymbol{\psi}$, M Monte Carlo samples of $\boldsymbol{Y}$ are generated from the likelihood function evaluated at $\boldsymbol{\psi}$. Then the likelihood ratio in $\alpha(\boldsymbol{\theta}^*|\boldsymbol{\theta})$ can be approximated as,

$$\begin{aligned}
\frac{p(\boldsymbol{Y}_2, ..., \boldsymbol{Y}_{T-1}|\boldsymbol{Y}_1, \boldsymbol{Y}_T; \boldsymbol{\theta}^*)}{p(\boldsymbol{Y}_2, ..., \boldsymbol{Y}_{T-1}|\boldsymbol{Y}_1, \boldsymbol{Y}_T; \boldsymbol{\theta})} &= \frac{\exp(\boldsymbol{\theta}^{*'}\boldsymbol{Z}^*_{\boldsymbol{\theta}^*})}{\exp(\boldsymbol{\theta}'\boldsymbol{Z}^*_{\boldsymbol{\theta}})} \times \frac{\frac{c^*(\boldsymbol{\theta})}{c^*(\boldsymbol{\psi})}}{\frac{c^*(\boldsymbol{\theta}^*)}{c^*(\boldsymbol{\psi})}} \\
&\approx \frac{\exp(\boldsymbol{\theta}^{*'}\boldsymbol{Z}^*_{\boldsymbol{\theta}^*})}{\exp(\boldsymbol{\theta}^{*'}\boldsymbol{Z}^*_{\boldsymbol{\theta}})} \times \frac{\sum_{m=1}^{M}\exp(\boldsymbol{\theta}'\boldsymbol{Z}^{*(m)}_{\boldsymbol{\theta}} - \boldsymbol{\psi}'\boldsymbol{Z}^{*(m)}_{\boldsymbol{\psi}})}{\sum_{m=1}^{M}\exp(\boldsymbol{\theta}^{*'}\boldsymbol{Z}^{*(m)}_{\boldsymbol{\theta}^*} - \boldsymbol{\psi}'\boldsymbol{Z}^{*(m)}_{\boldsymbol{\psi}})}
\end{aligned}$$

where $\boldsymbol{Z}^{*(m)}_{\boldsymbol{\theta}}$, $\boldsymbol{Z}^{*(m)}_{\boldsymbol{\theta}^*}$ and $\boldsymbol{Z}^{*(m)}_{\boldsymbol{\psi}}$ are $\boldsymbol{Z}^*$ evaluated at the mth Monte Carlo sample of $\boldsymbol{Y}$ with centers computed based on $\boldsymbol{\theta}$, $\boldsymbol{\theta}^*$ and $\boldsymbol{\psi}$, respectively.

2.2.4 Prediction

Let $\tilde{\boldsymbol{Y}} = (\boldsymbol{Y}_{T+1}, ..., \boldsymbol{Y}_{T+T^*}{}')$ denote the responses at future time points $\boldsymbol{Y}_{T+1}, ..., \boldsymbol{Y}_{T+T^*}$ with $T^* \geq 1$. For prediction of $\tilde{\boldsymbol{Y}}$ based on model parameter inference from EMPL and MCEML, we use Gibbs samplers to obtain Monte Carlo samples of $\tilde{\boldsymbol{Y}}$ from,

$$\begin{aligned} & p(\tilde{\boldsymbol{Y}}|\boldsymbol{Y}_T, \boldsymbol{Y}_{T+T^*+1}; \boldsymbol{\theta}) \\ & \propto \exp\{\sum_{t=T+1}^{T+T^*}(\sum_{i=1}^{n}\sum_{k=0}^{p}\theta_k X_{k,i,t}Y^*_{i,t} + \frac{1}{2}\sum_{i=1}^{n}\sum_{j\in N_i}\theta_{p+1}Y^*_{i,t}Y^*_{j,t}) + \sum_{t=T+1}^{T+T^*}\sum_{i=1}^{n}\theta_{p+2}Y^*_{i,t}Y^*_{i,t-1}\} \end{aligned}$$

That is, generate $Y_{i,t}$ from the full conditional distribution (2.1) evaluated at the EMPLE or the MCEMLE for $i = 1, ..., n$ and $t = T+1, ..., T+T^*$. For prediction of $\tilde{\boldsymbol{Y}}$ under the Bayesian framework, the posterior predictive distribution of $\tilde{\boldsymbol{Y}}$ is as following,

$$p(\tilde{\boldsymbol{Y}}|\boldsymbol{Y}_T, \boldsymbol{Y}_{T+T^*+1}) = \int p(\tilde{\boldsymbol{Y}}|\boldsymbol{Y}_T, \boldsymbol{Y}_{T+T^*+1}; \boldsymbol{\theta})p(\boldsymbol{\theta}|Y)d\boldsymbol{\theta} \tag{2.7}$$

To draw Monte Carlo samples $\tilde{\boldsymbol{Y}}$ from (2.7), first draw $\boldsymbol{\theta}$ from its posterior distribution $p(\boldsymbol{\theta}|\boldsymbol{Y})$, then draw $\tilde{\boldsymbol{Y}}$ from $p(\tilde{\boldsymbol{Y}}|\boldsymbol{Y}_T, \boldsymbol{Y}_{T+T^*+1}; \boldsymbol{\theta})$ for each given $\boldsymbol{\theta}$ using a Gibbs sampler (Zheng and Zhu 2008).

2.3 Simulation Studies

In this section, a simulation study is conducted to evaluate the performance of statistical inference approaches for the centered spatial-temporal autologistic regression model.

2.3.1 Data simulation and model specification

In the simulation, the size of the sampling grid $r \times r$ is varied by letting $r = 5, 10, \text{or } 20$. The number of sampling time points T is varied by letting $T = 7, 12, \text{or } 22$, which includes boundary time points. One covariate is considered in the model with $X_{i,t} \sim N(3,1)$. For spatial dependence, only the first order neighbors are considered. Thus the centered model is defined via the following full conditional distribution,

$$
\begin{aligned}
&p(Y_{i,t}|Y_{i',t'} : (i', t') \in N_{i,t}) \\
&= \frac{\exp\{\theta_0 Y_{i,t} + \theta_1 X_{i,t} Y_{i,t} + \sum_{j \in N_i} \theta_2 Y_{i,t} Y^*_{j,t} + \theta_3 Y_{i,t}(Y^*_{i,t-1} + Y^*_{i,t+1})\}}{1 + \exp\{\sum_{k=0}^{p} \theta_k X_{i,t} + \sum_{j \in N_i} \theta_2 Y^*_{j,t} + \theta_3 (Y^*_{i,t-1} + Y^*_{i,t+1})\}}
\end{aligned}
\tag{2.8}
$$

where

$$Y^*_{i,t} = Y_{i,t} - p_{i,t}$$

$Y^*_{i,t}$ is the centered response, N_i denotes the first order neighborhood for the ith site, θ_0 is an intercept, θ_1 is a slope for the covariate, θ_2 and θ_3 are spatial and temporal autoregressive coefficients, respectively. And $p_{i,t}$ is the expectation of $Y_{i,j}$ under the independent logistic regression model,

$$p_{i,t} = \frac{\exp\{\theta_0 + \theta_1 X_{i,t}\}}{1 + \exp\{\theta_0 + \theta_1 X_{i,t}\}} \tag{2.9}$$

Let $\boldsymbol{\theta} = (\theta_0, \theta_1, \theta_2, \theta_3)'$ denote the vector of all the model parameters. The corresponding joint distribution is,

$$
\begin{aligned}
&\mathcal{L}(\boldsymbol{\theta}) = p(\boldsymbol{Y}_2, ..., \boldsymbol{Y}_{T-1}|\boldsymbol{Y}_1, \boldsymbol{Y}_T; \boldsymbol{\theta}^*) \\
&= c^*(\boldsymbol{\theta})^{-1} \exp\{\sum_{t=2}^{T-1}[\sum_{i=1}^{n} \theta_0 Y^*_{i,t} + \sum_{i=1}^{n} \theta_1 X_{i,t} Y^*_{i,t} + \frac{1}{2}\sum_{i=1}^{n}\sum_{j \in N_i} \theta_2 Y^*_{i,t} Y^*_{j,t}] \\
&+ \sum_{t=2}^{T}\sum_{i=1}^{n} \theta_3 Y^*_{i,t} Y^*_{i,t-1}\}
\end{aligned}
$$

The spatial and temporal autoregressive coefficients are generally positive in real cases. When spatial and temporal autoregression coefficients exceed some critical values, MCEMLE would fail to exist because almost all values of data are same. According to Huffer and Wu (1998), the critical values may vary with different values of the coefficients of the covariates. In our simulation study, we consider θ_2 and θ_3 to be from 0 to 1, for which MCEMLE exists for all samples using the proposed method. In applications, the spatial and temporal autoregressive coefficients are generally positive. We fix the intercept to be $\theta_0 = 1$ and the slop to be $\theta_1 = -0.5$, but vary θ_2 and θ_3 to be 0.1, 0.5 and 0.9 to reflect different degrees of spatial and temporal dependence. For each combination of $\boldsymbol{\theta}$, r, and T, a perfect simulation sample is generated from the centered spatial-temporal autologistic regression model.

2.3.2 Centered parameterization versus uncentered parameterization

The first study is designed to identify the difficulties of uncentered parameterization in interpreting model parameters across varying levels of spatial and/or temporal dependence, and to demonstrate that centered parameterization can provide meaningful interpretation.

For each combination of $\boldsymbol{\theta}$, $M = 1,000$ data sets are generated from both the centered and traditional model using Gibbs sampler. Then, the expectation of $Y_{i,t}$ can be approximated to the marginal data means of corresponding simulated data set. For a simulated data set indexed by m and denoted by $\boldsymbol{Y}_m = \{Y_{i,t,m} : i = 1, ..., n; t = 1, ..., T\}$, the marginal data mean is defined as,

$$D_E\{\boldsymbol{Y}_m\} = \frac{1}{n}\sum_{i,t} Y_{i,t,m}$$

The expectation of $\boldsymbol{Y}_{i,j}$ under the independent logistic regression model is computed from (2.8). Figure 2.1 displays a comparison of the expectation of $\boldsymbol{Y}_{i,j}$ among the centered, traditional and independent models across different spatial-temporal dependence. As expected, with small spatial-temporal dependence ($\theta_2 = \theta_3 = 0.1$), there is a very high agreement between them, and there is only a tiny difference between centered and traditional model. Moreover, as spatial parameter θ_2 and/or temporal parameter θ_3 increases, the marginal mean values for the centered model remain similar to the expectation value of independent model, however, the marginal means of traditional model increases the realizations towards 1.

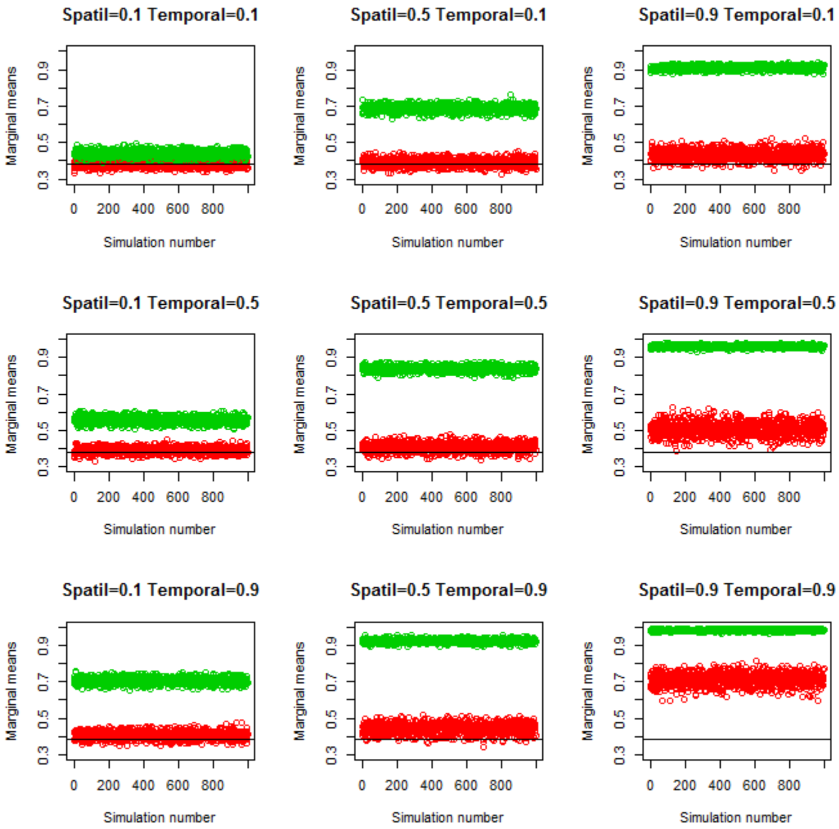

Figure 2.1: Comparison of the expectation of $Y_{i,t}$ among centered, traditional and independent models for all combinations of $\boldsymbol{\theta}$.

Let $\boldsymbol{Ys}$ denote M Monte Carlo simulated data sets. The Monte Carlo estimate of the expected average marginal data structure is defined as,

$$E_M\{\boldsymbol{Ys}\} = \frac{1}{M}\sum_{m=1}^{M} D_E\{\boldsymbol{Y}_m\}$$

95% confidence Monte Carlo confidence intervals are computed from the quantiles. Figure 2.2 presents a comparison of the Monte Carlo expectation between centered and traditional model. It points out that the parameters of traditional model increases dramatically as the strength of the spatial-temporal dependence increases, while the parameters of the centered model give a reasonable interpretation across varies levels of dependence. It can be seen from Figure 2.2, when both spatial and temporal dependence are strong, the performance of the centered model decreases, the main reason is that EMPL is statistically inefficient in this case (EMPL is used to compute the center in generating Monto Carlo samples).

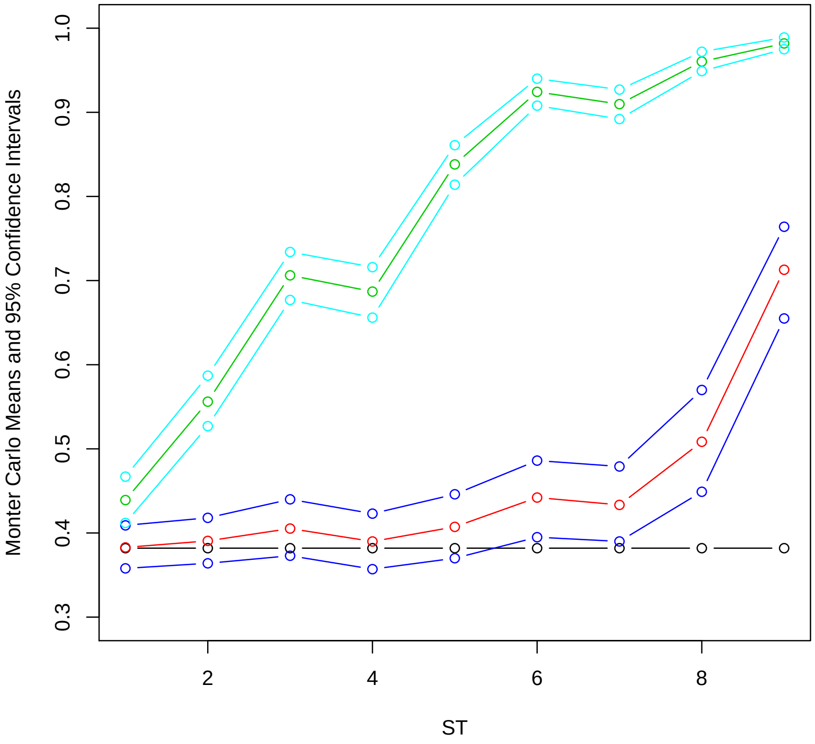

Figure 2.2:

Comparison of Monte Carlo means among centered, traditional and independent models for all combinations of $\boldsymbol{\theta}$

The points from *left* to *right* are presenting cases with spatial and temporal coefficients $(\boldsymbol{\theta}_2, \boldsymbol{\theta}_3)$ = (0.1, 0.1), (0.1, 0.5), (0.1, 0.9), (0.5, 0.1), (0.5, 0.5), (0.5, 0.9), (0.9, 0.1), (0.9, 0.5), (0.9, 0.9), respectively. Points with *red*, and *blue* color are representing Monte Carlo means and corresponding 95% confidence intervals for the centered models, points with *red*, and *teal* color are representing them for the traditional means, and point with *black* color are representing the expectation of $Y_{i,t}$ of under independent model.

2.3.3 Statistical efficiency and performance comparison

EMPLE

For each simulated data set, let the initial value $\boldsymbol{\theta}_0$ be the estimate under the independent logistic regression model, the EMPLEs are computed following the EMPL algorithm, and the standard error of each EMPLE is computed from its corresponding 100 resampled EMPLEs. Figures 2.3, 2.4, 2.5 and 2.6 show the EMPLEs and their 95% confidence intervals for all the simulated data sets. The results show that both the size of sampling lattices and the number of sampling time points have a significant effect on the performance of EMPLEs. First, as grid size increases, the general performance of EMPLEs improves with decreasing estimation error in terms of both bias and standard errors. For example, when $r = 5$, $T = 7$ and the spatial and temporal dependence are relatively weak $\boldsymbol{\theta}_2 = \boldsymbol{\theta}_3 = 0.1$, the EMPLEs of $\boldsymbol{\theta}$ are $(0.432, 0.367, 0.221, 0.601)$ with standard errors $(0.696, 0.236, 0.364, 0.449)$. Fix $T = 7$ and let the size of lattice increase to $r = 20$, the EMPLEs are $(0.813, 0.455, 0.149, 0.181)$ with standard errors reduced to $(0.155, 0.049, 0.067, 0.093)$. Second, as the number of sampling time points increases, the performance of EMPLEs also gets better with decreasing estimation errors. For instance, fix $r = 5$ and let the number of sampling time points increase to $T = 22$, the EMPLEs of model parameters are $(0.904, 0.472, 0.228, 0.055)$ with smaller standard errors $(0.297, 0.095, 0.147, 0.196)$. For the other cases with different spatial and/or temporal autoregressive coefficients, similarly, as grid size and the number of time points increase, the performance of EMPLEs improves with decreasing estimation errors. On the other hand, when both spatial and temporal autoregressive coefficients are large, the realization of data tends to same values so that the EMPLE of the intercept is not accurate.

EMPLE and Bayesian

The MCEMLEs are also computed for each simulated data set. Figures 2.7, 2.8, 2.9 and 2.10 give the MCEMLEs and the corresponding 95% confidence intervals for all simulated data sets. Here the size of Monte Carlo samples generated at Step 0 in the algorithm is 100 using PGS. MCEMLEs are more accurate than EMPLEs with smaller standard errors. Even when both spatial and temporal autoregressive coefficients are large, MCEMLEs are good estimators for all model parameters since they have small bias and standard errors. The results of the Bayesian approach agree well with MCEMLEs, which are not shown here.

Critical values of converge

Huffer and Wu (1998) pointed out that the critical values may vary with different values of the coefficients of the covariates. In this simulation study, with fixed intercept $\boldsymbol{\theta}_0$ and coefficient of covariate $\boldsymbol{\theta}_1$, for a 20 sampling grid with the number of sampling time points $T = 22$, we check the range of the spatial autoregressive coefficient $\boldsymbol{\theta}_2$ and the temporal autoregressive coefficient $\boldsymbol{\theta}_3$ for which MCEMLE exists for the samples using the proposed method. The results show when $\boldsymbol{\theta}_2$ exceeds 1 and $\boldsymbol{\theta}_3$ exceeds 1.2, MCEMLE would fail to exist. Furthermore, the ranges of $\boldsymbol{\theta}_2$ and $\boldsymbol{\theta}_3$ where EMPLE serves as a good reference point for the MCEML has been studied, and the results are given in Table 2.1.

Computation demand

In Table 2, it shows the time that it takes to compute the statistical inference results for EMPL, MCEML, and Bayesian inference approach for partial combinations of θ, r, and T. For Bayesian inference, a total of 1,000 Monte Carlo samples are generated with the first 1,000 samples discarded for burn-in. The computing time is based on R

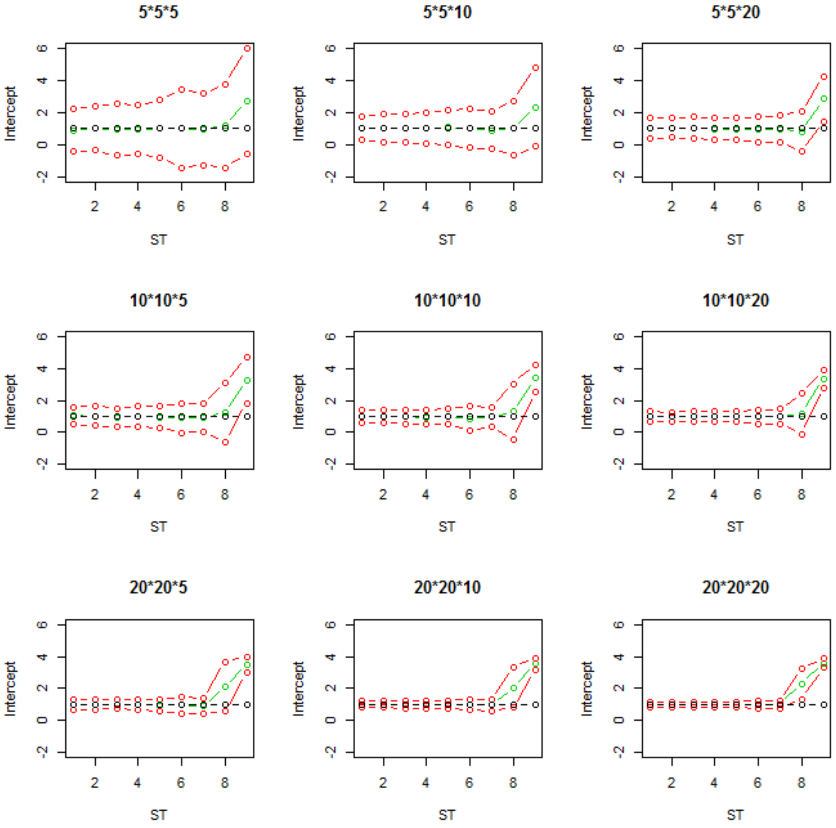

Figure 2.3: EMPLE of the intercept $\boldsymbol{\theta}_0$ in the simulation study.
In each subplot, the points from *left* to *right* are presenting cases with spatial and temporal coefficients $(\boldsymbol{\theta}_2, \boldsymbol{\theta}_3) =$ (0.1, 0.1), (0.1, 0.5), (0.1, 0.9), (0.5, 0.1), (0.5, 0.5), (0.5, 0.9), (0.9, 0.1), (0.9, 0.5), (0.9, 0.9), respectively. Points with *black*, *green*, and *red* color are representing true values, estimates, and 95% confidence intervals, respectively. The title of each subplot indicates the size of the spatial lattice and the number of sampling time points r × r × (T - 2).

and C programs written by the authors and run on an AMD Phenom II X personal computer. The results show that the computing time for EMPL and MCEML are comparable. For EMPL, the parameter estimation part is fast and most of the computing time is spent on computing the standard errors. For MCEML, although the standard errors based on Fishers information are quickly computed, the parameter estimation part is more time-consuming than EMPL since we need to generate Monte Carlo samples from the reference point and at the E step we need to update centered

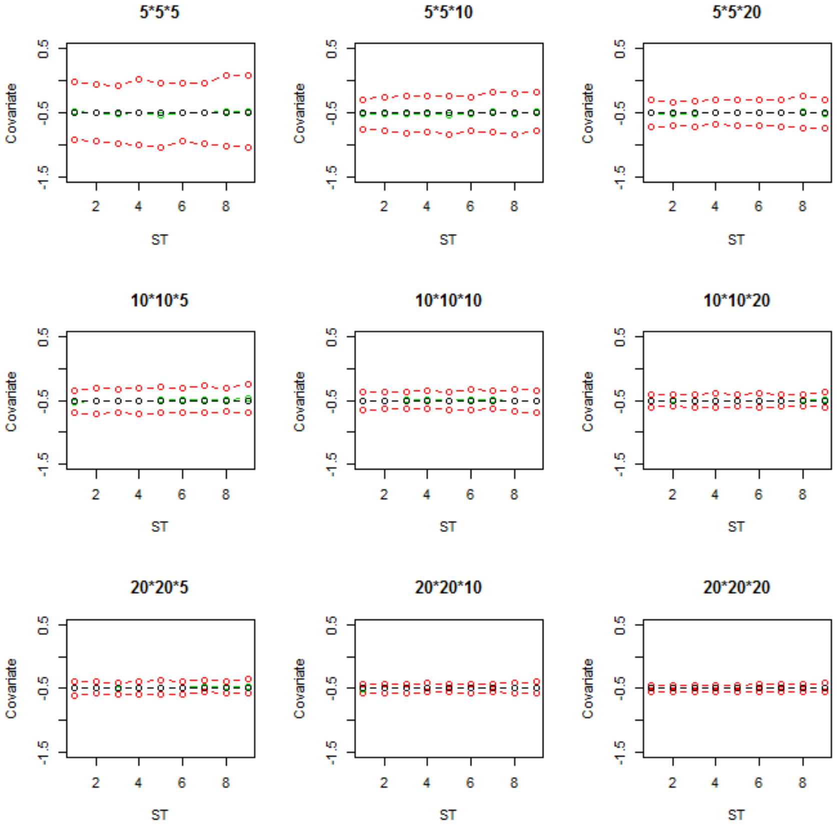

Figure 2.4: EMPLE of the coefficient of the covariate $\boldsymbol{\theta}_1$ in the simulation study. In each subplot, the points from *left* to *right* are presenting cases with spatial and temporal coefficients $(\boldsymbol{\theta}_2, \boldsymbol{\theta}_3) = (0.1,\ 0.1), (0.1,\ 0.5), (0.1,\ 0.9), (0.5,\ 0.1), (0.5,\ 0.5), (0.5,\ 0.9), (0.9,\ 0.1), (0.9,\ 0.5), (0.9,\ 0.9)$, respectively. Points with *black*, *green*, and *red* color are representing true values, estimates, and 95% confidence intervals, respectively. The title of each subplot indicates the size of the spatial lattice and the number of sampling time points r × r × (T 2).

responses for each Monte Carlo sample. Bayesian approach is the most computationally intensive, which is not recommended for the centered parameterization. It is interesting to note that additional computing time is spent for EMPL and MCEML as spatial and temporal autoregression coefficients get larger. The reason is that it takes more time for the coupled chains to achieve coalescence in perfect simulation.

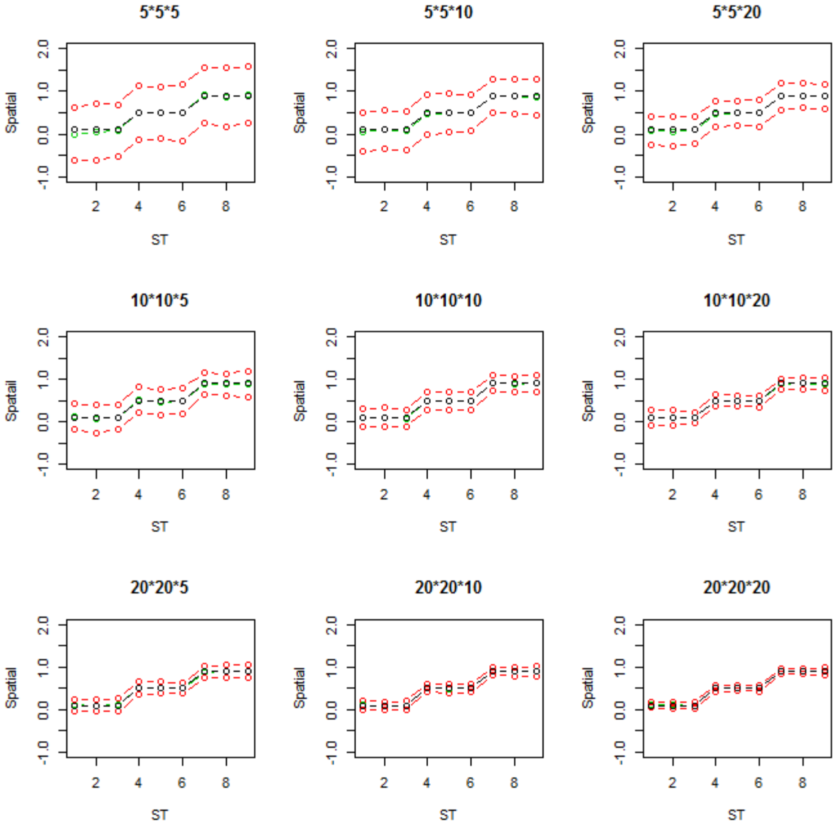

Figure 2.5: EMPLE of the spatial correlation coefficient $\boldsymbol{\theta}_2$ in the simulation study. In each subplot, the points from *left* to *right* are presenting cases with spatial and temporal coefficients $(\boldsymbol{\theta}_2, \boldsymbol{\theta}_3)$ = (0.1, 0.1), (0.1, 0.5), (0.1, 0.9), (0.5, 0.1), (0.5, 0.5), (0.5, 0.9), (0.9, 0.1), (0.9, 0.5), (0.9, 0.9), respectively. Points with *black*, *green*, and *red* color are representing true values, estimates, and 95% confidence intervals, respectively. The title of each subplot indicates the size of the spatial lattice and the number of sampling time points r × r × (T 2).

2.4 Application to the southern pine beetle data

The southern pine beetle is the most destructive insect to pines in the southern United States, it will attack all Southern Yellow Pines, especially loblolly, shortleaf, and pitch pines. The southern pine beetle (SPB) data consist of SPB outbreak (0 = no outbreak; 1 = outbreak) in the 100 counties of North Carolina from 1960 to 1996. Figure 2.11 is a time-series map of the outbreak. To make comparison with results in Zheng and Zhu (2008), the average precipitation in the fall (in cm) is considered

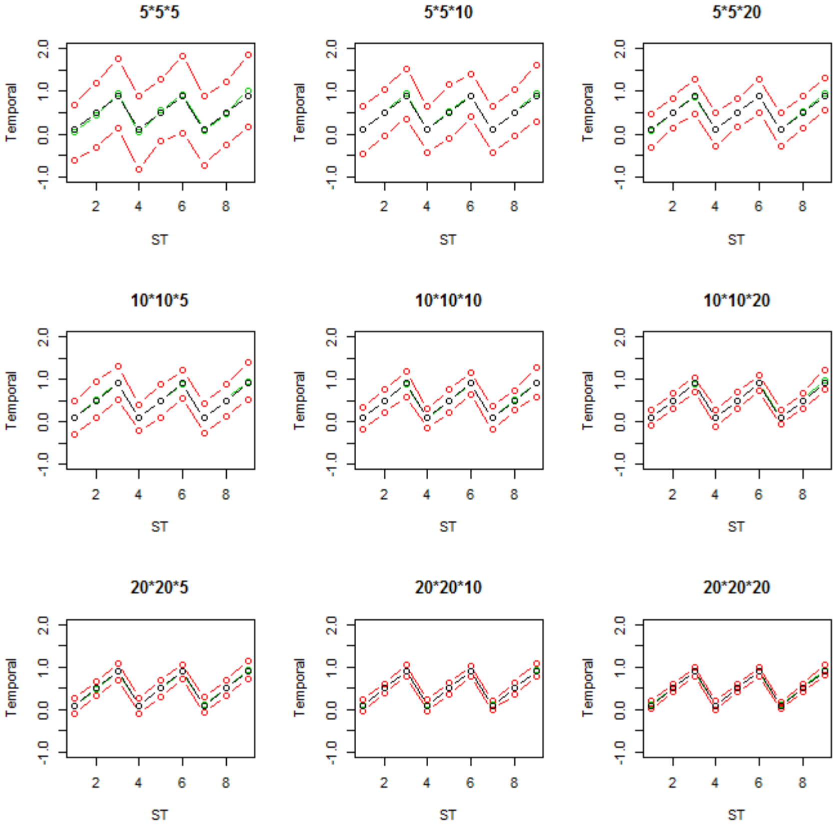

Figure 2.6: EMPLE of the temporal correlation coefficient $\boldsymbol{\theta}_3$ in the simulation study. In each subplot, the points from *left* to *right* are presenting cases with spatial and temporal coefficients $(\boldsymbol{\theta}_2, \boldsymbol{\theta}_3)$ = (0.1, 0.1), (0.1, 0.5), (0.1, 0.9), (0.5, 0.1), (0.5, 0.5), (0.5, 0.9), (0.9, 0.1), (0.9, 0.5), (0.9, 0.9), respectively. Points with *black*, *green*, and *red* color are representing true values, estimates, and 95% confidence intervals, respectively. The title of each subplot indicates the size of the spatial lattice and the number of sampling time points r × r × (T 2).

as the only covariate in the model (Fig. 2.12). Data from 1960 to 1991 are used for model parameter inference, and data from 1992 to 1996 for model validation. Two counties were considered to be neighbors if the corresponding county seats are within 30 miles of each other. The likelihood function of the centered model is same as that in the simulation study as given in (2.8).

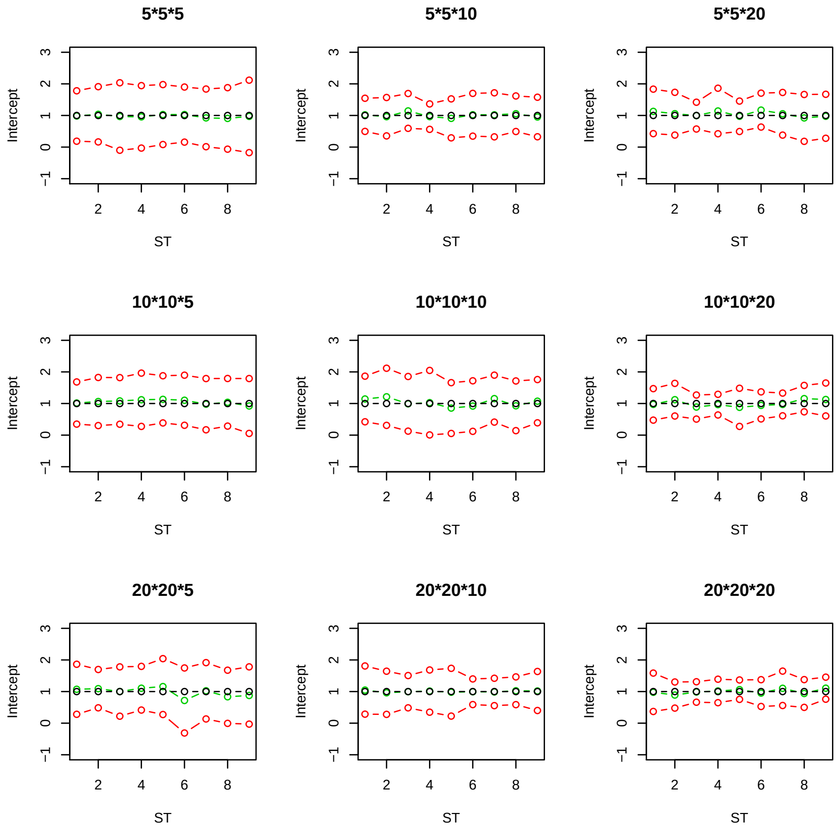

Figure 2.7: MCEML of the intercept $\boldsymbol{\theta}_0$ in the simulation study.
In each subplot, the points from *left* to *right* are presenting cases with spatial and temporal coefficients $(\boldsymbol{\theta}_2, \boldsymbol{\theta}_3)$ = (0.1, 0.1), (0.1, 0.5), (0.1, 0.9), (0.5, 0.1), (0.5, 0.5), (0.5, 0.9), (0.9, 0.1), (0.9, 0.5), (0.9, 0.9), respectively. Points with *black*, *green*, and *red* color are representing true values, estimates, and 95% confidence intervals, respectively. The title of each subplot indicates the size of the spatial lattice and the number of sampling time points r × r × (T 2).

Table 2.3 gives the model parameter inference using EMPL, MCEML, and the Bayesian hierarchical model. For EM pseudo-likelihood, the EMPLEs and their standard errors obtained by parametric bootstrap are reported. For each EMPLE, 1,000 resampled EMPLEs are used to compute the standard error. For MCEM likelihood, the reference parameter is from the MCMC stochastic approximation algorithm, and both the MCEMLEs and their standard errors obtained from the empirical Fisher information are reported. Furthermore, for the Bayesian inference, set the prior distribution to

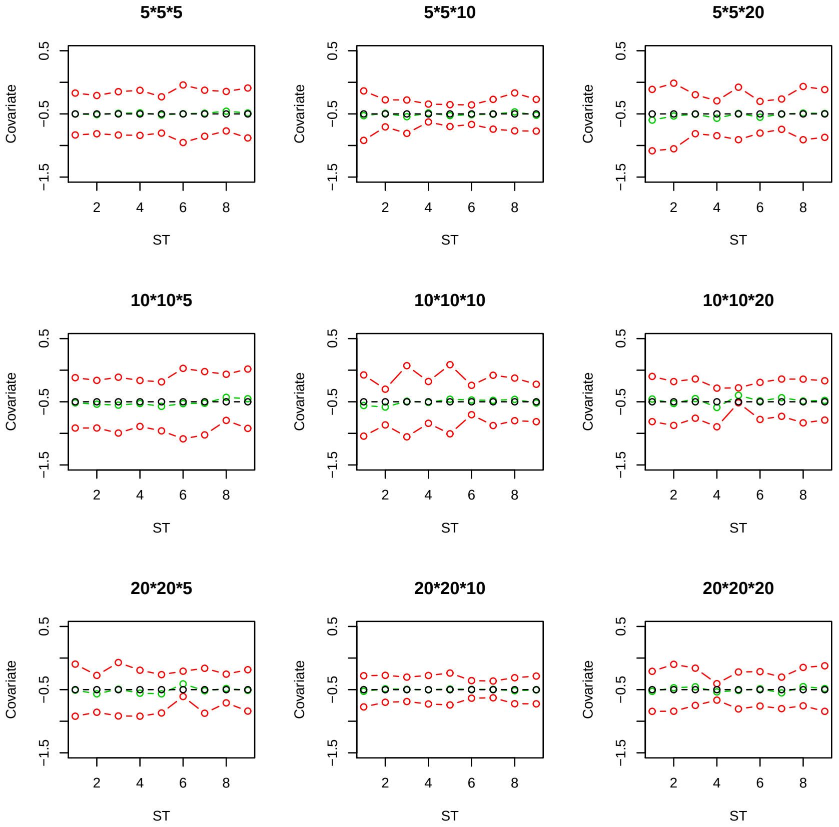

Figure 2.8: MCEML of the coefficient of the covariate $\boldsymbol{\theta}_1$ in the simulation study. In each subplot, the points from *left* to *right* are presenting cases with spatial and temporal coefficients $(\boldsymbol{\theta}_2, \boldsymbol{\theta}_3) = (0.1,\ 0.1), (0.1,\ 0.5), (0.1,\ 0.9), (0.5,\ 0.1), (0.5,\ 0.5), (0.5,\ 0.9), (0.9,\ 0.1), (0.9,\ 0.5), (0.9,\ 0.9)$, respectively. Points with *black*, *green*, and *red* color are representing true values, estimates, and 95% confidence intervals, respectively. The title of each subplot indicates the size of the spatial lattice and the number of sampling time points r × r × (T 2).

be uniform on (10, 10) for all model parameters and variance components in the proposal distribution to be $\sigma_0^2 = \sigma_1^2 = \sigma_2^2 = \sigma_3^2 = 0.012^2$. A total of 200,000 Monte Carlo samples are generated with the first 1,000 samples discarded for burn-in and the means with the standard deviation of the posterior samples of the model parameters are reported.

The results suggest that the inference for the model parameters under centered pa-

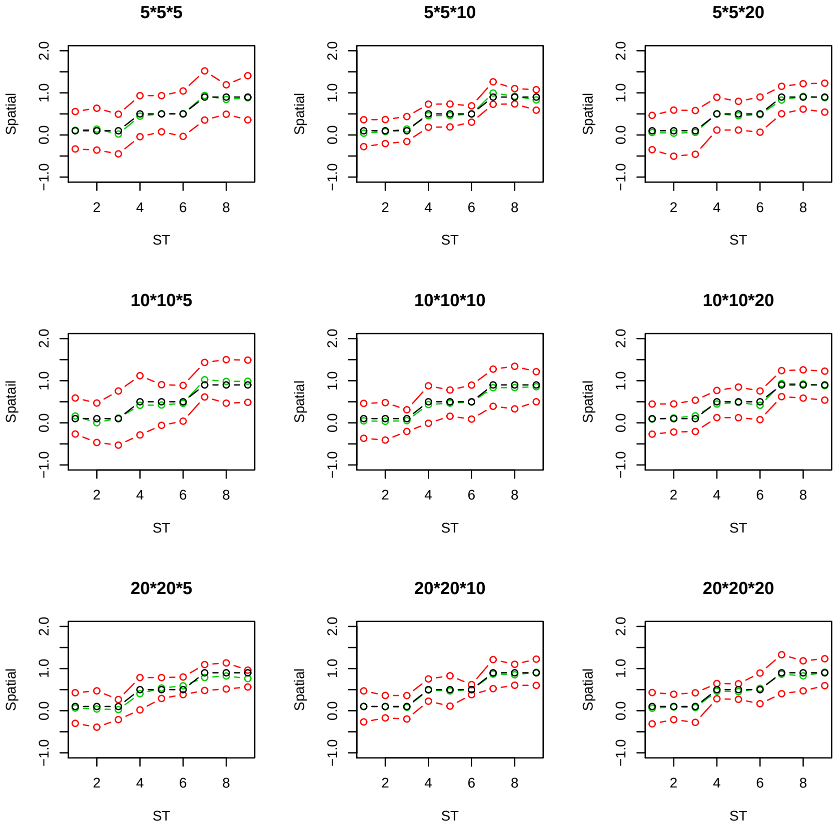

Figure 2.9: MCEML of the spatial correlation coefficient $\boldsymbol{\theta}_2$ in the simulation study. In each subplot, the points from *left* to *right* are presenting cases with spatial and temporal coefficients $(\boldsymbol{\theta}_2, \boldsymbol{\theta}_3) =$ (0.1, 0.1), (0.1, 0.5), (0.1, 0.9), (0.5, 0.1), (0.5, 0.5), (0.5, 0.9), (0.9, 0.1), (0.9, 0.5), (0.9, 0.9), respectively. Points with *black*, *green*, and *red* color are representing true values, estimates, and 95% confidence intervals, respectively. The title of each subplot indicates the size of the spatial lattice and the number of sampling time points r × r × (T 2).

rameterization using the posterior distribution matches well with MCEML, but the inference from EM pseudo-likelihood is very different from both Bayesian inference and MCEM likelihood. Also, estimation based on EM pseudo-likelihood results in higher variance than Bayesian inference and MCEM likelihood. MCEML and Bayesian inference show that there is a negative relation between SPB outbreaks and the mean precipitation in the fall, while EMPL gives an opposite result. All the three approaches suggest that there is significant evidence of positive spatial and temporal

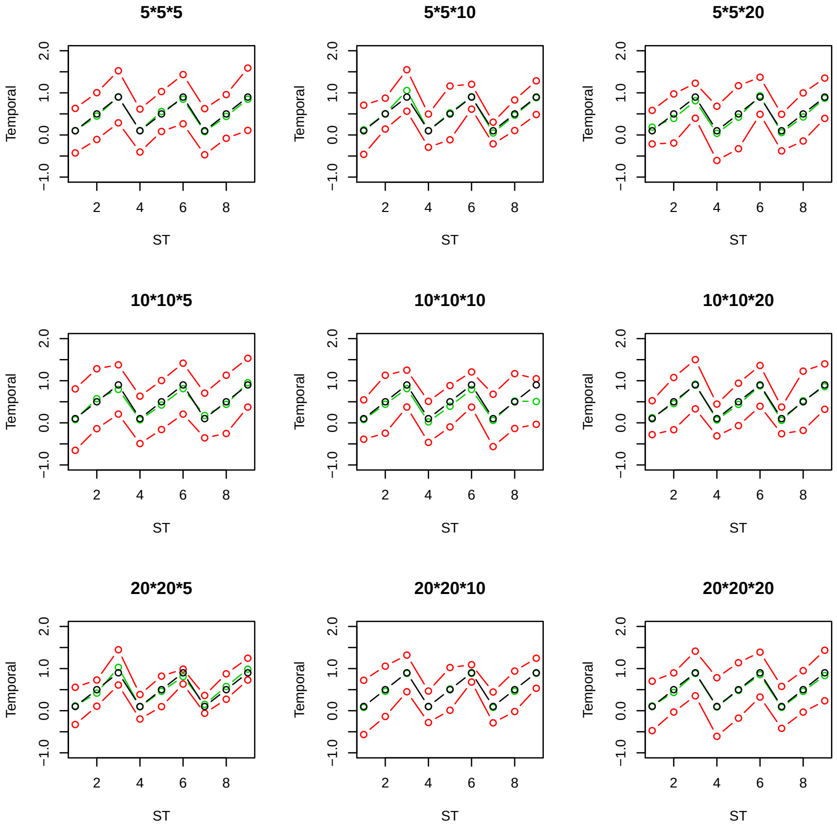

Figure 2.10: MCEML of the temporal correlation coefficient $\boldsymbol{\theta}_3$ in the simulation study.
In each subplot, the points from *left* to *right* are presenting cases with spatial and temporal coefficients $(\boldsymbol{\theta}_2, \boldsymbol{\theta}_3) = (0.1,\ 0.1)$, (0.1, 0.5), (0.1, 0.9), (0.5, 0.1), (0.5, 0.5), (0.5, 0.9), (0.9, 0.1), (0.9, 0.5), (0.9, 0.9), respectively. Points with *black*, *green*, and *red* color are representing true values, estimates, and 95% confidence intervals, respectively. The title of each subplot indicates the size of the spatial lattice and the number of sampling time points r × r × (T - 2).

dependence for SPB outbreak. Furthermore, for comparison, the results from the uncentered spatial-temporal autologistic regression model (Zheng and Zhu 2008) are presented in Table 2.4. The parameter estimates and the corresponding standard errors for the centered model are very close to those for the uncentered model using all three inference approaches. The possible reason for this is that the influence of the center is very small for this example. The average of the centers $p_{i,t}$ evaluated at the MCEMLE is only 0.05 and the spatial and temporal autoregressive terms dominate

Table 2.1: Largest values for the spatial autoregressive coefficient $\boldsymbol{\theta}_2$ and the temporal autoregressive coefficient $\boldsymbol{\theta}_3$ for which MCEMLE exists when using the EMPLE as the reference point

Lattice $r \times r$	Time points $T-2$	Spatial $\boldsymbol{\theta}_2$	Temporal $\boldsymbol{\theta}_3$
20*20	20	0.1	1.9
20*20	20	0.2	1.8
20*20	20	0.3	1.6
20*20	20	0.4	1.4
20*20	20	0.5	1.2
20*20	20	0.6	1
20*20	20	0.7	0.8
20*20	20	0.8	0.7
20*20	20	0.9	0.5
20*20	20	1	0.3

Table 2.2: Comparison of model parameter estimation for the centered spatial-temporal autologistic model using expectation-maximization pseudo-likelihood (EMPL), Monte Carlo expectation-maximization likelihood (MCEML), and Bayesian inference. (Unit in second)

Lattice $r \times r$	Time $T-2$	Spatial $\boldsymbol{\theta}_2$	Temporal $\boldsymbol{\theta}_3$	EMPL	MCEML	Bayesian
5*5	5	0.1	0.1	4.67	3.67	626
5*5	5	0.9	0.9	34.42	29.42	654
5*5	10	0.1	0.1	6.22	4.23	778
5*5	10	0.9	0.9	48.76	39.26	765
10*10	5	0.1	0.1	9.46	8.48	1057
10*10	5	0.9	0.9	82.25	66.94	1074

Table 2.3: Comparison of model parameter estimation for the centered spatial-temporal autologistic model using expectation-maximization pseudolikelihood (EMPL), Monte Carlo expectation-maximization likelihood (MCEML), and Bayesian inference

Parameters	EMPL		Bayesian		MCEML	
	Estimate	SE	Estimate	SE	Estimate	SE
Intercept	-4.9648	0.3199	-2.8577	0.2911	-2.4043	0.1554
Slope	0.2131	0.0796	-0.1345	0.0753	-0.1298	0.0505
Spatial	1.4706	0.1307	0.9504	0.0579	0.9534	0.0483
Temporal	1.7502	0.1774	0.8918	0.1024	0.8903	0.0728

the model, which makes the difference between the centered and uncentered parameterization not evident.

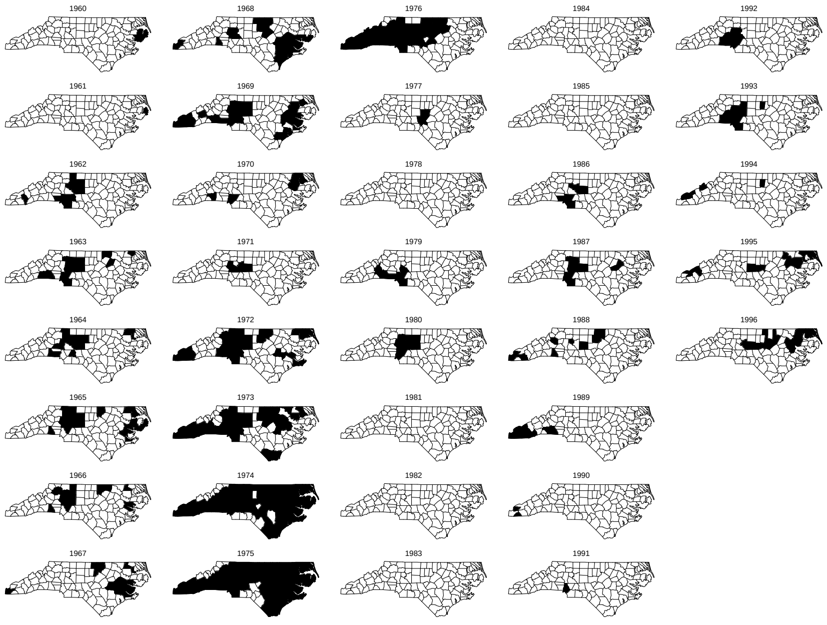

Figure 2.11: Map of southern pine beetle outbreaks from 1960 to 1996 in North Carolina. (For each county, black color implies an outbreak)

Table 2.4: Comparison of model parameter estimation for the uncentered spatial-temporal autologistic model using maximum pseudolikelihood (MPL), Monte Carlo maximum likelihood (MCML), and Bayesian inference (Zheng and Zhu 2008)

Parameters	MPL		Bayesian		ML	
	Estimate	SE	Estimate	SE	Estimate	SE
Intercept	-5.1600	0.6606	-2.7075	0.2074	-2.7079	0.2033
Slope	0.2459	-0.1760	-0.1433	0.0546	-0.1433	0.0524
Spatial	1.4503	0.1379	0.9075	0.0583	0.9114	0.0537
Temporal	1.7135	0.2372	1.0257	0.1282	1.0198	0.1174

For prediction and model validation, the SPB outbreak from Year 1992 to 2001 is predicted. The responses at the ending time point, i.e. $Y_{i,2002}{}^{'}s$, are generated from independent Bernoulli trials with probability of outbreak $\sum_{t=1960}^{1991} \frac{Y_{i,t}}{31}, i = 1, ..., 100$. For model parameter inference based on EMPL and MCEML, Gibbs samplers are used to generate 1,000,000 Monte Carlo samples starting at a perfect simulated sample. Then every 50th of the 1,000,000 samples are used to form an approximately independent Monte Carlo sample of size 20,000. For model parameter inference based

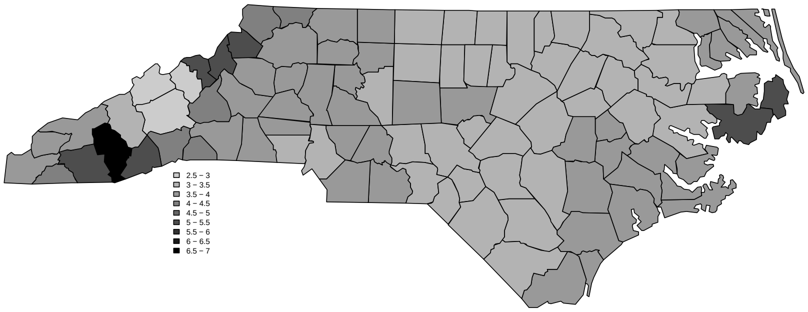

Figure 2.12: Map of mean fall precipitation in North Carolina

on the Bayesian hierarchical model, every 10th of the 200,000 Monte Carlo samples of the model parameters is taken to form an approximately independent Monte Carlo sample of size 20,000. Then a Gibbs sampler with burn-in is used to generate a prediction for each $\boldsymbol{\theta}$. Under each type of inference, the mean of the predicted values is used to predict whether that county has an outbreak for that year. For each year between 1992 and 1996 where the data are available, a prediction error rate is computed as the proportion of counties that are with outbreaks predicted differently from the actual observation. The corresponding prediction error rates are reported in Table 2.5. Again, the prediction results are close for the Bayesian approach and MCEML, but the prediction is very poor using EMPL.

The prediction performance based on the centered spatial-temporal autologistic regression model and traditional model are comparable (see Table 2.5). Since the statistical inference based on the centered parameterization is much more computationally

Table 2.5: Comparison of the prediction performance between the centered model and the uncentered model

Year	Centered Model			Traditional Model		
	EMPL	Bayesian	MCEML	MPL	Bayesian	MCML
1992	0.65	0.14	0.18	0.66	0.09	0.09
1993	0.72	0.12	0.19	0.65	0.13	0.13
1994	0.70	0.14	0.20	0.74	0.08	0.16
1995	0.63	0.13	0.23	0.68	0.14	0.13
1996	0.62	0.09	0.24	0.61	0.16	0.17

intensive, it appears that one can simply use the uncentered model if prediction is of primary interest, although further investigation will be needed. If the focus is on the interpretation of the model parameters, the centered parameterization would be recommended.

Copyright© Zilong Wang, 2012.

Chapter 3 Imputation Methods for Spatial-Temporal Data

Missing data, i.e. incomplete data matrices, are important problems that are repeatedly encountered in spatial-temporal studies. Generally, the spatial-temporal study is required to use complete data matrices, otherwise it could significantly distort statistical conclusions (Kim et al 2004). There are a large number of imputation techniques available, but most are invalid based on the efficiency and accuracy of imputation. The main reason is that the missing data in spatial-temporal study is related to location and time information. The main contribution of this chapter is algorithm development for iteration-KNN and maximum entropy imputation on spatial-temporal data. It should be pointed out that these two new imputation methods are not limited to spatial-temporal data, they can be applied to any missing data under the MAR assumption.

3.1 Imputation methods

In this section, the general schemes of iteration-KNN and maximum entropy imputation methods are sketched, as well as those of KNN and EM imputation methods. Let $\boldsymbol{Y}$ denote a data matrix for response variable. With $i = 1, ..., n$, let y_i denote the ith response observation. Let R_i be the missing data indicator, then $R_i = 1$ if y_i is observed, otherwise $R_i = 0$. Let $\boldsymbol{Y}^m$ denote all missing data and $\boldsymbol{Y}^o$ denote all observed data, then the full data is $\boldsymbol{Y} = (\boldsymbol{Y}^o, \boldsymbol{Y}^m)$.

3.1.1 KNN imputation

k-nearest neighbor (kNN) imputation is one of the most important and fastest imputation methods in incomplete data discovery, which has been developed with great success on industrial data. There are two R packages, "Imputation" and "YaImpute", which can carry out in KNN imputation. KNN imputation is developed from hot-dect imputation under using K nearest neighbor observations (Meesad and Hengpraprohm 2008). Hot-dect imputation is that the imputed values should be achieved from the same data set where the missing values are from. Same as hot-deck imputation, KNN imputation is preferred in the situation that it preserves the distribution of item values and thus can mostly keep the data properties as if they are not missing (Rao and Shao, 1992).

In order to estimate a missing value y_i, first, K references of non-missing values whose contribution values are most similar to y_i are selected from the whole data set. Next, the imputed value of $y_{i,j}$ is estimated as the average value of them,

$$\tilde{y}_i = \frac{1}{K} \sum_{j \in N_i} (y_j) \tag{3.1}$$

where N_i is the index set of non-missing K-nearest neighbor observations for ith missing response observation y_i. On the other hand, it should be pointed out that there is no theoretical criteria for selecting the best K-value. Generally the K-value is determined by the experience of researchers from similar studies. For spatial-temporal data, K values can be determined by the spatial and temporal neighborhood structures.

3.1.2 EM imputation

EM imputation is a kind of regression-based imputation method, which is a general framework for solving maximum likelihood/pseudo-likelihood problems when an observable model is derived from an underlying latent model. Based on the study of Dempster, Laird, and Rubin (1977), EM imputation algorithm only requires a weaker MAR assumption. This imputation method is based on estimated regression models between missing data and observed data with combining EM algorithm. The imputation procedure consists of iterations of EM algorithm where the expectation values and covariance matrices of the incomplete data are estimated (Bolotin 2001). The algorithm of EM imputation includes the following steps:

- **Step 0:** Start from a preselected $\boldsymbol{\theta}_0$ and set $\hat{\boldsymbol{\theta}}^0 = \boldsymbol{\theta}_0$.
- **E (expectation) step:** Given $\hat{\boldsymbol{\theta}}^{l-1}$

  Replace missing values with estimated values $\tilde{\boldsymbol{Y}}^{m(l)}$.

  Where estimated values $\tilde{\boldsymbol{Y}}^{m(l)}$ are based on the expectations of the missing data, which is conditional on the current stage parameter $\hat{\boldsymbol{\theta}}^{l-1}$ and the observed data $\boldsymbol{Y}^o$.
- **M (Maximization) step:** Given $\tilde{\boldsymbol{Y}}^{m(l)}$

  Obtain $\hat{\boldsymbol{\theta}}^l$ by maximizing the likelihood/pseudo-likelihood function $\mathcal{L}(\boldsymbol{\theta}|\{\boldsymbol{Y}^o, \tilde{\boldsymbol{Y}}^{m(l)}\})$.
- **Convergence criteria**

  Repeat step **E** and **M** until $\mathcal{L}(\boldsymbol{\theta}^l; \boldsymbol{Y}^o) < \mathcal{L}(\boldsymbol{\theta}^{l-1}; \boldsymbol{Y}^o)$, then $\tilde{\boldsymbol{Y}}^m = \tilde{\boldsymbol{Y}}^{m(l-1)}$.

Where a preselected $\boldsymbol{\theta}_0$ can be obtained by maximizing the likelihood/pseudo-likelihood function $\mathcal{L}(\boldsymbol{\theta}|\{\boldsymbol{Y}^o, \tilde{\boldsymbol{Y}}^{m(0)}\})$, $\tilde{\boldsymbol{Y}}^{m(0)}$ are imputed values of missing data using KNN imputation or mean substitution.

Same as EM, the EM imputation algorithm converges monotonically in that the likelihood/pseudolikelihood of the available data increases monotonically from iteration to iteration. However, EM imputation algorithm converges only linearly, and the rate of convergence depends on the fraction of values that are missing in the data set, and so it may need many iterations to converge, i.e. EM iteration is time intensive requiring more computation. For spatial-temporal missing, EM is a good choice if both the size of data is relatively small and the spatial-temporal model is relatively simple.

3.1.3 Iteration-KNN imputation

When the missing rate is in high level or the non-missing data are biased and can not keep the properties of whole data set, the performance of KNN is very poor and it should lead to a serious bias during the inference. In this situation, iteration-KNN imputation is developed based on KNN, which can improve accuracy but still keep same level of computational demand compared to KNN (Caruana 2008).

Iteration-KNN is an EM style non-parametric imputing method, which uses an iterative KNN for imputing missing values. The algorithm is similar to EM with using KNN instead of parametric regression models. However, iteration-KNN combines E and M steps into a single step because it updates the fill-in imputed values and the model at the same time. It first estimates missing values from observed data by KNN imputation and cuts the data into q unjoint subsets, then piecewise improves accuracy of fill-in values through recursive process for all subsets. Compared to EM imputation, iteration-KNN imputation is more efficient with acceptable accuracy. Furthermore, the performance of iteration-KNN is better when regression models are unknown or cannot fit the data well. The algorithm of iteration-KNN imputation is developed for missing data as following.

- **Step 1:**

  (1) Impute and fill in all missing values $\tilde{\boldsymbol{Y}}^{m(0)}$ by KNN imputation.

  (2) Divide whole data set to q unjoint subsets $\{U_1, ..., U_q\}$,

$$\{y_1, ..., y_{j_{U_1}}\} \in U_1$$

$$\{y_{j_{U_1}+1}, ..., y_{j_{U_2}}\} \in U_2$$

$$...$$

$$\{y_{j_{U_{q-1}}+1}, ..., y_{j_{U_q}}\} \in U_3$$

such that for each missing data $y_i \in U_c$, its K nearest neighbors can be found in the joint set $\{U_{c-1} \cup U_c \cup U_{c+1}\}$, where $q \in \mathcal{Z}$ and $q > 3$.

- **step 2:**

(1) Impute and fill in missing values for subset U_1 by KNN imputation, treating the other subset $\{U_2, ..., U_q\}$ non-missing.

(2) Impute and fill in missing values for subset U_2 by KNN imputation, treating the other subset $\{U_1, U_3, ..., U_q\}$ non-missing.

...

(q) Impute and fill in missing values for subset U_q by KNN imputation, treating the other subset $\{U_1, U_3, ..., U_{q-1}\}$ non-missing.

In the end of first iteration, all fill-in imputed values $\tilde{\boldsymbol{Y}}^{m(1)}$ are obtained.

- **Convergence criteria**

Repeat step 2 until $sup\{|\tilde{\boldsymbol{Y}}^{m(l)} - \tilde{\boldsymbol{Y}}^{m(1-1)}|\} < \delta$ in lth iteration, then $\tilde{\boldsymbol{Y}}^{m} = \tilde{\boldsymbol{Y}}^{m(l-1)}$. Where δ is a preselected precision parameter for checking convergence.

Unlike EM imputation, iteration-KNN imputation has a fast rate of convergence, usually the iteration number is less than 10 if the number of subsets q is not too large. It should be pointed out that no theoretical criteria for selecting the best q number, which is determined by the size and the inherent structure of the data. For spatial-temporal data, the number of time points is a nature choice to determine a reasonable q number.

3.1.4 Maximum entropy imputation

The principle of maximum entropy was introduced by Bishop and Ulrych (1975) and Guiasu and Shenitzer (1985). In statistics, a maximum entropy probability distribution is a probability distribution whose entropy is at least as great as that of all other members of a specified class of distributions. That is, if nothing is known about a distribution except that it belongs to a certain class, then the distribution with the largest entropy should be chosen as default. For example, under specified mean μ and standard deviation σ, the normal distribution $N(\mu, \sigma^2)$ has maximum entropy among all real-valued distributions.

Maximum entropy imputation is an imputation method which is based on the maximum entropy framework, the main idea is that the probability distribution with the maximum entropy subject to additional constrains should be chosen, where these constrains are based on what is known (Uffink 1995). Generally, constrains can be achieved from the results of similar studies, statistical inference from a small training data set, or even research background knowledge. The performance of maximum entropy imputation depends on the additional constrains, i.e., quantity and quality of external or internal information (Uffink 1996). But when missing rate is high, maximum entropy imputation has the best performance. The reason is that observed data may not reserve enough information to discover the statistical inference under high missing levels, the maximum entropy distribution is the only reasonable probability distribution for producing proper imputation.

For the continuous variable, the entropy of the ith observation Y_i is defined as,

$$H(Y_i) = -\int p(Y_i) \log p(Y_i) dY_i \tag{3.2}$$

Here $p(Y_i)\log p(Y_i) = 0$ if $p(Y_i) = 0$. For discrete variable, the entropy of the ith observation y_i is defined as,

$$H(Y_i) = -\sum_{j=1}^{k} p(Y_{ij}) \log p(Y_{ij}) \tag{3.3}$$

Here $\sum_{j=1}^{k} p(Y_i = Y_{ij}) = 1$ and again $p(Y_i) \log pY_i = 0$ if $p(Y_i) = 0$.

When there are no missing values, the ith observation Y_i would be known to be equal to its observed values y_i. In this situation, the entropy of Y_i is,

$$H(Y_i) = -p(y_i) \log p(y_i) = 0$$

In contrast, suppose the variable Y_i has missing values, and the missing belongs to MAR. The missing observation Y_i would be known to be suited within the confidence interval from regression substitution. Let Y_i^L and Y_i^U be the lower and upper boundaries of the confidence interval for continuous variable, respectively. Then the entropy of Y_i would be,

$$H(Y_i) = -\int_{Y_i^L}^{Y_i^U} p(Y_i) \log p(Y_i) dY_i$$

or L and U are the indexes of lower and upper boundary for discrete variable,

$$H(Y_i) = -\sum_{j=L}^{U} p(Y_{ij}) \log p(Y_{ij})$$

Hence, it can be seen that the maximum entropy converges to its maximum values allowed by those limitations, i.e., by our background knowledge about Y.

The algorithm of the maximum entropy is more depended on additional constrains. Suppose there exists m constrains $c_1, ..., c_m$. Based on the entropy framework and these constrains, the imputed value of y_i is estimated as,

$$\text{Max}\, H(y_i) = -\sum_{j=1}^{k} p(y_{ij}) \log p(y_{ij})$$
$$\text{such that} \begin{cases} \sum_{j=1}^{k} p\left(y_i = y_{\text{ij}}\right) = 1 \\ \\ \text{satisfy constrians}\, c_1, ..., c_m \end{cases} \tag{3.4}$$

For spatial-temporal missing data, most of them belong to missing at random (MAR) cases. Therefore, some information of the missing values would be known from regression models, the confidence intervals from corresponding models are a good choice as one reasonable additional constrain. With this model based constrain, the performance of maximum entropy imputation should be similar to EM imputation in small or median missing rates, and better than EM imputation in high missing rates.

3.2 Simulation Study and Application

To evaluate the efficiency and accuracy of iteration-KNN and maximum entropy imputation methods for spatial-temporal data under various missing rates, a comparison among KNN, iteration-KNN, EM, and maximum entropy (with model based confidence interval constrain) imputation methods is designed for this purpose. Furthermore, because the response variable is binary data with values 0 or 1, the influence of the large scale probabilities is also studied.

3.2.1 Data simulation and imputation methods specification

In the study, a traditional spatial-temporal autologistic regression model with only one covariate is considered, which is defined in (1.1). the conditional expectation of

$Y_{i,t}$ given its neighbors is,

$$\begin{aligned}
&E(Y_{i,t}|Y_{i',t'} : (i', t') \in N_{i,t}) \\
&= \frac{\exp\{\theta_0 + \theta_1 X_{i,t} + \sum_{j \in N_i} \theta_2 Y_{j,t} + \theta_3 (Y_{i,t-1} + Y_{i,t+1})\}}{1 + \exp\{\theta_0 + \theta_1 X_{i,t} + \sum_{j \in N_i} \theta_2 Y_{j,t} + \theta_3 (Y_{i,t-1} + Y_{i,t+1})\}}
\end{aligned}$$

Set the size of the sampling grid to be $r \times r = 10 \times 10$, and the time points to be $T = 12$. Here the observations of the first and last time points are the boundaries with no missing values. One covariate is considered in the model with $X_{i,t} \sim N(3, 1)$. For spatial dependence, only the first order neighbors are considered, let N_i denote the first order neighborhood for the ith site. For model parameters $\boldsymbol{\theta}$, fix intercept θ_0, slope θ_1 and temporal autoregressive coefficient θ_3 to be 1, -0.5, and 0.5, respectively, but vary θ_2 from 0.1, 0.3 to 0.5 to reflect different large scale probabilities P_L, which are defined as,

$$\begin{aligned}
P_L &= \frac{1}{r \times r \times (T-2)} \sum_{i=1}^{n} \sum_{t=2}^{T-1} E[Y_{i,t}] \\
&= \frac{1}{r \times r \times (T-2)} \sum_{i=1}^{n} \sum_{t=2}^{T-1} \times \frac{\exp\{\theta_0 + \theta_1 X_{i,t} + \sum_{j \in N_i} \theta_2 Y_{j,t} + \theta_3 (Y_{i,t-1} + Y_{i,t+1})\}}{1 + \exp\{\theta_0 + \theta_1 X_{i,t} + \sum_{j \in N_i} \theta_2 Y_{j,t} + \theta_3 (Y_{i,t-1} + Y_{i,t+1})\}}
\end{aligned}$$

The three simulation data sets are generated from the traditional spatial-temporal autologistic regression model using a perfect simulation sampler. Then generate random missing values at the rate $0.05, 0.1, 0.15, 0.2, 0.3, 0.4, 0.5, 0.6$. Let $R_{i,t}$ denote the missing data indicator for ith site and tth time point. For simulated data sets, P_L can be approximated by marginal data mean,

$$P_L \approx \frac{1}{r \times r \times (T-2)} \sum_{i=1}^{n} \sum_{t=1}^{T-1} R_{i,t} Y_{i,t}$$

With missing values, P_L would be approximated as,

$$P_L \approx \frac{1}{\sum_{i=1}^{n} \sum_{t=1}^{T-1} R_{i,t}} \sum_{i=1}^{n} \sum_{t=1}^{T-1} R_{i,t} Y_{i,t} R_{i,t}$$

By 3.4, the approximate values of large scale probabilities P_L are 0.56, 0.7, and 0.86 for $\theta_2 = 0.1$, 0.3, and 0.5, respectively. It is difficult to impute missing values $Y_{i,t}$

directly for binary response variable. An alternative way is using the expectation of $Y_{i,j}$ instead of itself. Let $P_{i,t}$ denote the the expectation of $Y_{i,t}$, i.e.,

$$P_{i,t} = E[Y_{i,t}] = p(Y_{i,t} = 1)$$

The absolute difference between $P_{i,t}$ and its imputed values is considered. Let $D_{i,t}$ denote the absolute different between $P_{i,t}$ and its imputed values in ith site and tth time point.

$$D_{i,t} = |P_{i,t} - \tilde{P}_{i,t}|, , i = 1, ..., n; t = 2, ..., T-1$$

To measure accuracy, except error rate, the average of absolute probability difference, APD, is another good measurement.

$$APD = \frac{1}{n \times (T-2)} \sum_{i=1}^{n} \sum_{t=2}^{T-1} D_{i,t}, i = 1, ..., n; t = 2, ..., T-1$$

In additional, it is not suitable to randomly generate 0 or 1 based on imputed $P_{i,j}$, since the error rate will increase rapidly by adding extra variance $P_{i,t}(1 - P_{i,t})$. Thus, the missing value $Y_{i,t}$ is imputed as,

$$\tilde{Y}_{i,t} = \begin{cases} 1, \text{if } P_{i,t} \geqslant 0.5 \\ \\ 0, \text{otherwise} \end{cases}$$

Following the above definitions and conditions, we will describe the additional details of the algorithms for KNN, EM, iteration-KNN, and maximum entropy imputation methods. First, for KNN imputation, the nearest neighbors are selected same as the spatial and temporal neighbors used in the model (1.1). That is,

$$N_i^{KNN} = N_i \cup \{Y_{i,t-1}, Y_{i,t+1}\}$$

Then the imputed value $\tilde{p}_{i,t}$ is computed as the following,

$$\tilde{p}_{i,t} = \begin{cases} \frac{Y_{i,t-1}R_{i,t-1}+Y_{i,t+1}R_{i,t+1}+\sum_{j\in N_i} Y_{j,t}R_{j,t}}{R_{i,t-1}+R_{i,t+1}+\sum_{j\in N_i} R_{j,t}}, \text{if } R_{i,t-1} + R_{i,t+1} + \sum_{j\in N_i} R_{j,t} \neq 0 \\ \\ \frac{\sum_{j=1}^{n}\sum_{t=1}^{T} Y_{j,t}R_{j,t}}{\sum_{j=1}^{n}\sum_{t=1}^{T} R_{j,t}}, \text{otherwise} \end{cases}$$

Second, iteration-KNN imputation uses the same nearest neighbors as KNN. Based on its algorithm, the imputed value $\tilde{Y}_{i,t}^{l}$ in lth iteration is computed by,

$$\begin{aligned} \tilde{Y}_{i,t}^{l} =& \frac{1}{\sum_{j\in N_i} R_{j,t} + 2}\{[Y_{i,t-1}R_{i,t-1} + Y_{i,t+1}R_{i,t+1} + \sum_{j\in N_i} Y_{j,t}R_{j,t}] \\ &+ [\tilde{p}_{i,t-1}^{l}(1 - R_{i,t-1}) + \tilde{p}_{i,t+1}^{l-1}(1 - R_{i,t+1}) + \sum_{j\in N_i} \tilde{p}_{j,t}^{l-1}(1 - R_{j,t})]\} \end{aligned}$$

Where

$$\tilde{p}_{i,t}^{l} = E[Y_{i,j}|\{\boldsymbol{Y}^{o}, \boldsymbol{Y}^{m(l-1)}\}]$$

$\tilde{p}_{i,t}^{l}$ denotes the lth imputed value of $p_{i,t}$ under all observed values and $(l-1)$th imputed expectation values. The convergence criteria is as following,

$$sup\{|\tilde{p}_{i,t}^{m(l)} - \tilde{p}_{i,t}^{m(1-1)}|\} < \delta$$

Third, for EM imputation, pseudo-likelihood is considered and the parameters estimation can be carried out by standard logistic regression functions for the full data with imputed missing values. In step 0, all missing values are first filled by KNN imputation $Y^{m(0)}$, then initial parameters $\boldsymbol{\theta}_0$ can be computed from the full data with imputed missing values. Last, 95% confidence interval for missing values $P_{i,t}$ is used as the only constrain in maximum entropy imputation. For lth iteration, the corresponding 95% confidence interval $[p_{i,t}^{L(l)}, p_{i,t}^{U(l)}]$ can be approximated from the quantiles of the parametric bootstrap sample. That is, 100 Monte Carlo samples of binary responses are drawn from the pseudo-likelihood function with $\boldsymbol{\theta}^{(l-1)}$ using PGS, then compute $\tilde{p}_{i,t}^{l}$ for each Monte Carlo samples and $\tilde{p}_{i,t}^{l(b)}, b = 1, ..., 100$ constructs the parametric bootstrap sample. The steps of the maximum entropy are as following,

- **Step 1:** Replace missing values with imputed values $\tilde{Y}^{m(0)}$ by KNN imputation.
- **Step 2:** Compute MPLE $\boldsymbol{\theta}^0$ from the data $\{Y^o, \tilde{Y}^{m(0)}\}$.
- **Step 3:** Compute approximately 95% confidence interval $[p_{i,t}^{L(1)}, p_{i,t}^{U(1)}]$ for each missing values.
- **Step 4:** Replace missing values with imputed values $\tilde{Y}^{m(1)}$ by 3.4 with the constrain $p_{i,t} \in [p_{i,t}^{L(1)}, p_{i,t}^{U(1)}]$.
- **Step 5:** Compute MPLE $\boldsymbol{\theta}^1$ from the data $\{Y^o, \tilde{Y}^{m(1)}\}$.
- **Step 5:** Repeat step 3 to 5 until $sup|p_{i,t}^l - \tilde{p}_{i,t}^{l-1}| < \delta$, then $\tilde{p}_{i,t} = \tilde{p}_{i,t}^l$.

3.2.2 Imputation accuracy and efficiency comparison

For each simulated date set under various missing rate, let the initial imputed missing values be estimated from KNN imputation. Then start from observed data and KNN imputed data, impute the missing values from EM, iteration-KNN, and maximum entropy imputation methods.

The results clearly show the performance of KNN, iteration-KNN, EM, and maximum entropy imputation methods are significantly different. Figure 2.1 shows that their performance are same as predicted both in error rates and APD. That is, under the large scale probability 0.56 ($\theta_2 = 0.1$), first, KNN is the fastest imputation methods, but it has the worst performance in any missing rates and large scale probability. For example, when the missing rate is 0.15, KNN has 0.3967 error rate, but the error rates are 0.3133, 0.2933 and 0.3022 for iteration-KNN, EM, and maximum entropy imputation methods, respectively. Second, Iteration-KNN has faster convergent speed and the number of iterations is from 3 to 8 in our study. Also it has better performance than KNN, and the imputation results are stabler as the missing rates increase. For instance, when the error rates of iteration-KNN jump 0.082 (from 0.31 to 0.392) as the missing rates jump from 0.05 to 0.5, but KNN and EM imputation methods jump 0.17 (from 0.31 to 0.48), and 0.212 (from 0.26 to 0.472) at the same time. Third, EM has the best performance with missing rates under around 0.3, but it need more computing time than others, especially when the missing rate is large. For the same reason, the performance deteriorates rapidly as the missing rates higher than 0.3, since the regression model will be invalid with higher missing rate. When missing rate is less than 0,3, the error rates of EM is less than 0.296, it is the smallest compared to 0.41, 0.335 and 0.3 for KNN, iteration-KNN and maximum entropy, respectively. But when the missing rate is 0.3 or higher, the error rates increase rapidly and EM has the worse performance than iteration-KNN and maximum entropy. Last, maximum entropy has better performance than KNN and iteration-KNN in any situation, but worse than EM if regression model is valid. When the missing rate is higher than 0.3, i.e., the regression model is invalid, maximum entropy imputation can still keep a reasonable error rate under the properties of entropy itself. Also, the rate of convergence for maximum entropy is faster than EM, but slower than iteration KNN. Furthermore, the error rate is not a good criterion to show the imputation performance for binary data; APD is a better choice. For example, under large scale probability 0.86 ($\theta_2 = 0.5$), iteration-KNN has worst performance than EM and maximum entropy; the APDs are 0.1225, 0.0596 and 0.0702 under missing rate 0.2. respectively. But they are in same error rate level, 0.135, 0.13, and 0.135, respectively.

Both missing rates and large scale probability are factors which has significant effect on imputation. Figure 3.1 shows that the error rates and APDs increase as missing rates increases. Figure 3.2 shows that large scale probability is also needed to consider in imputation if the response variable is binary data, or categorical data

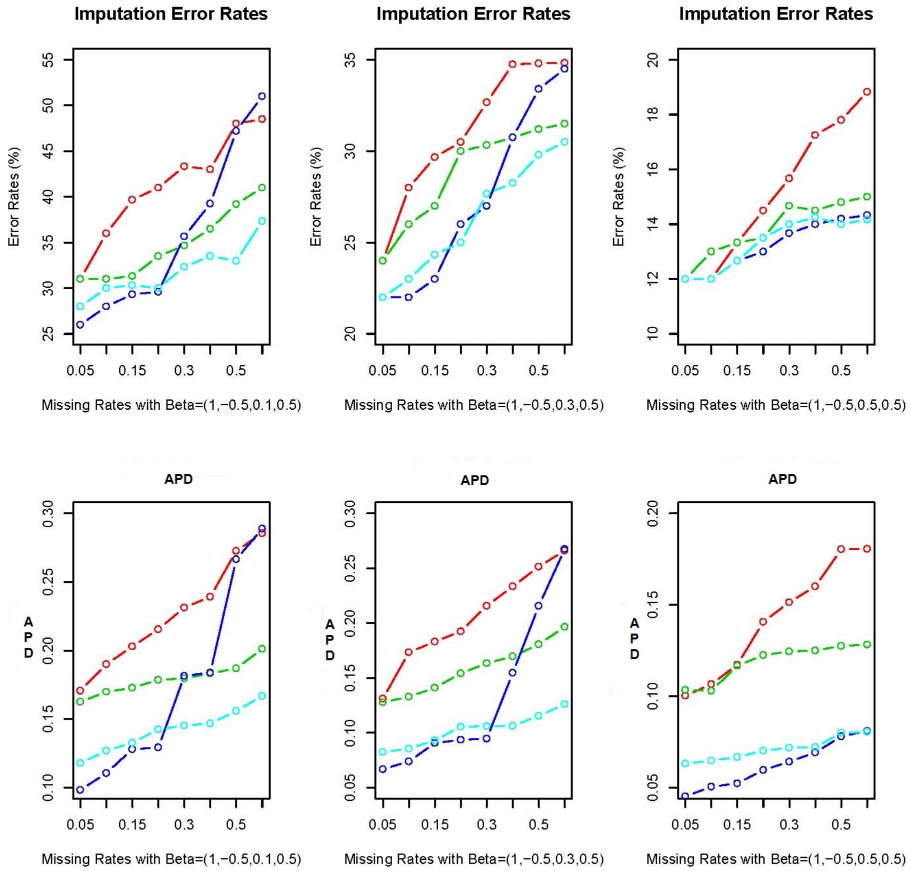

Figure 3.1: Comparison imputation performance by imputation methods.
The upper 3 subplots are error rates plots with the large scale probabilities 0.56, 0.7 and 0.86, respectively. Points with *red*, *green*, *blue*, and *teal* color represent KNN, iteration KNN, EM and maximum entropy imputation methods, respectively. The lower 3 subplots are APD plots with the large scale probabilities 0.56, 0.7 and 0.86, respectively. Points with *red*, *green*, *blue*, and *teal* color are representing KNN, iteration KNN, EM and maximum entropy imputation methods, respectively.

with small levels. For example, if the large scale probability is close to 0.5 (average probability of all possible levels), the inherent structure of original data is difficult to be discovered, i.e., even small amount of missing values will disrupt the inherent structure. In this case, imputation methods have more power to affect the results of imputation. If the large scale probability close to 1, i.e., the extreme situation, the inherent structure of original data can be discovered by small proportion data. So these imputation methods tend to have similar performance.

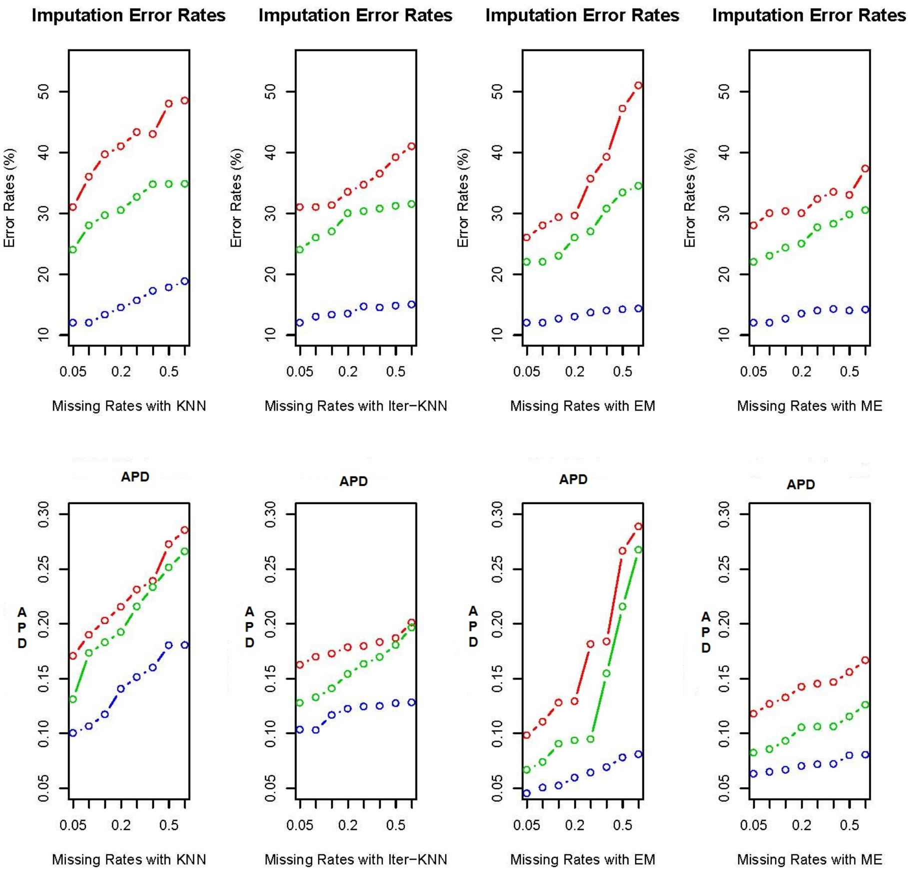

Figure 3.2: Comparison imputation performance by large scale probability.
The upper 4 subplots are KNN, iteration-KNN, EM and maximum entorpy imputation methods, respectively. Points with *red*, *green*, and *blue*, color represent the large scale probabilities 0.56, 0.7 and 0.86, respectively. The lower 4 subplots are KNN, iteration-KNN, EM and maximum entorpy imputation methods, respectively. Points with *red*, *green*, and *blue*, color represent the large scale probabilities 0.56, 0.7 and 0.86, respectively.

3.2.3 Application to the mountain pine beetle data

The mountain pine beetle (MPB)is a species of bark beetle native to the forests of western North America, which attacks pine trees by laying eggs under the bark. Usually, in dry summers and mild winters the population of MPB increases and spreads quickly so that huge areas of pine trees will be killed. The mountain pine beetle

(MPB) data consist of MPB outbreaks (0 = no outbreak; 1 = outbreak) in the Chilcotin area of the central British Columbia (Canada) from 1998 to 2006, and the data set is spatial-temporal binary data set with the size of study areas 17063. Figure 3.3 is a map of the study area in the Chilcotin. Because temperature plays a vital role in the MBP outbreaks, the mean temperature of each year (in Celsius degree) is considered as the only covariate in the model. Data from 1999 to 2005 are used for imputation methods validation, and data in Year 1998 and 2006 are boundaries for temporal part. Assume that the missing rates are 0.05, 0.1, 0.15, 0.2, 0.3, 0.4, 0.5 and 0.6. For any study area, the four nearest study areas were considered to be neighbors based on the distance. Figure 3.4 is a plot of outbreaks by year. We assume that the performance of every imputation methods is good in that the large scale probability

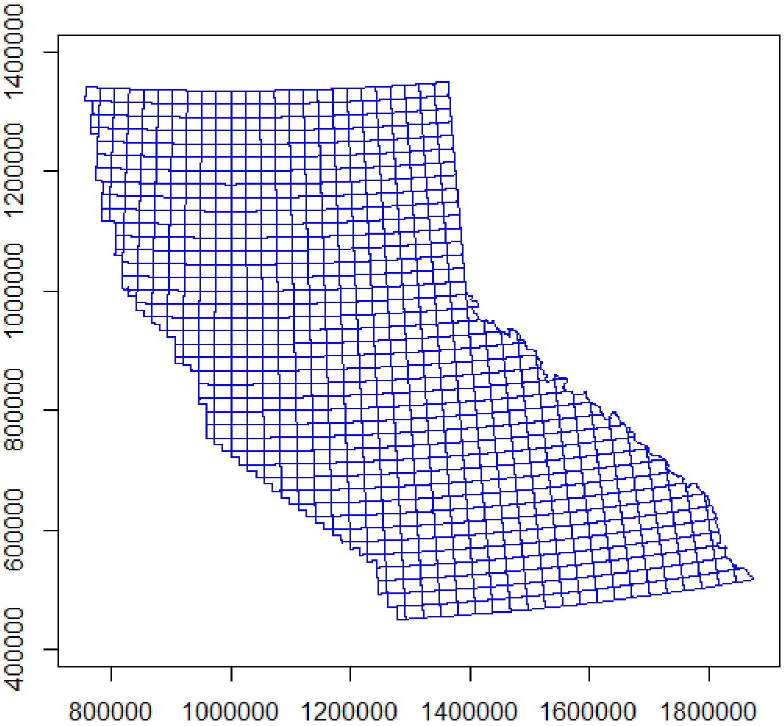

Figure 3.3: Map of the study area in the Chilcotin (Canada).

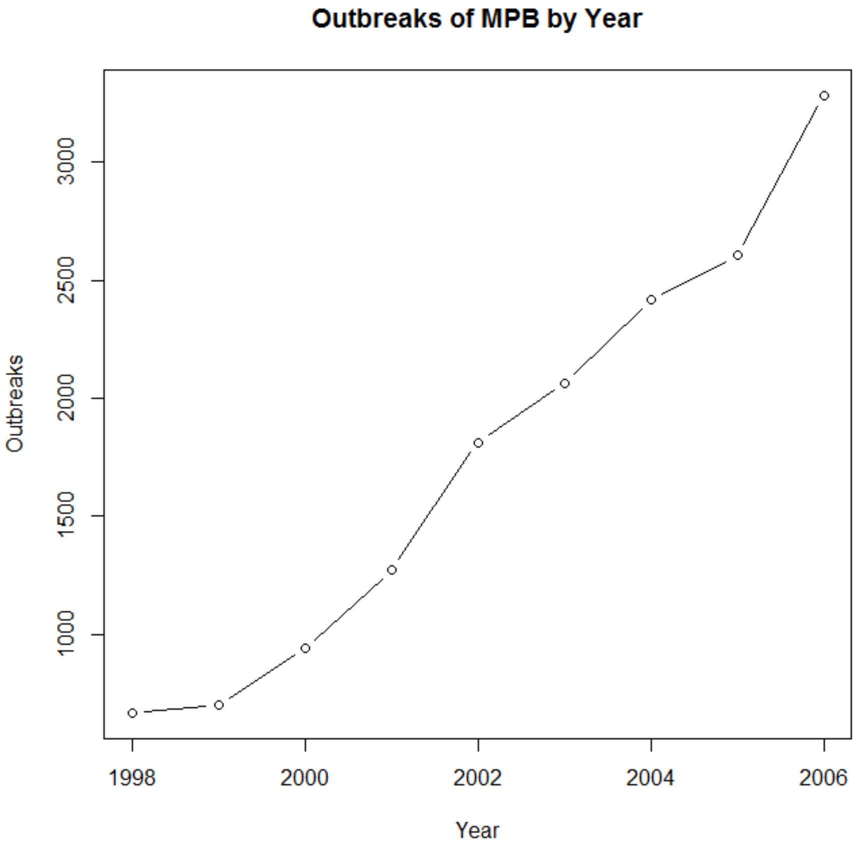

Figure 3.4: Map of the outbreaks MBP by year).

of MPB is approximated to 0.1026.

Figure 3.5 gives error rates using KNN, iteration-KNN, EM, and maximum entropy for missing rate 0.05, 0.1, 0.15, 0.2, 0.3, 0.4, 0.5 and 0.6. Same as predicted, all of them have low error rates, the order of the performance from highest to lowest is EM, maximum entropy, iteration-KNN and KNN. For example, with the largest missing rate 0.6, the error rates are 0.0533, 0.0429, 0.0334, and 0.0392 for KNN, iteration-KNN, maximum entropy and EM, respectively. In our research (not shown here), we have also studied average of absolute probability difference (APD)for the performance of imputation methods. Both the simulation study and the real data example show that the performance of EM is best under a low missing level, when missing increases, iteration-KNN and maximum entropy are good alternatives.

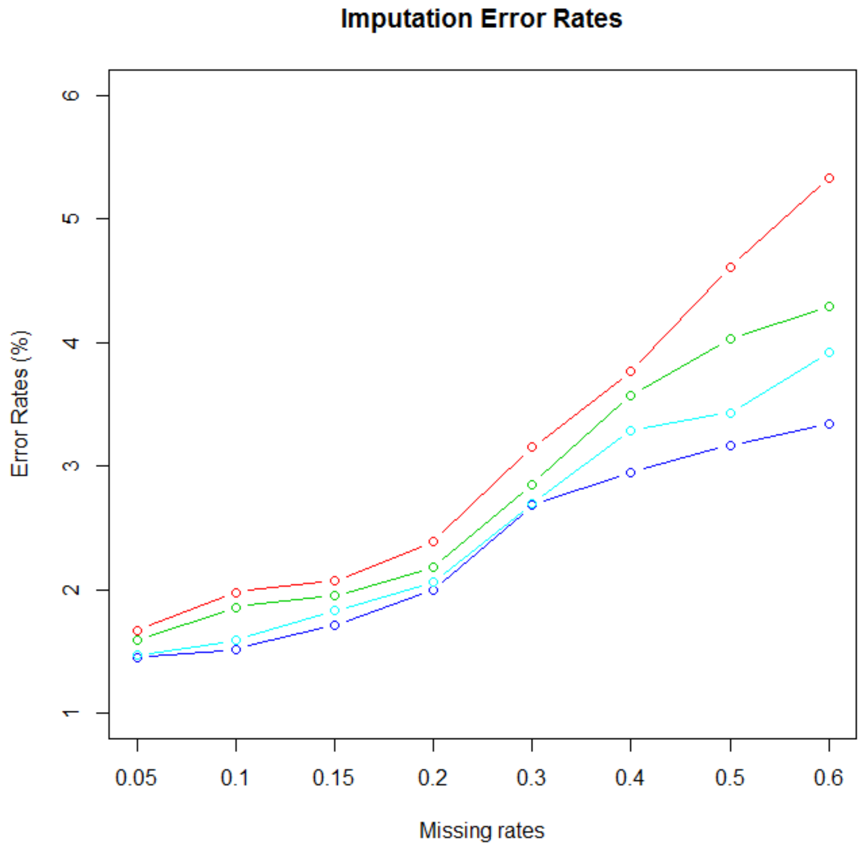

Figure 3.5: Comparison of error rates for KNN, iteration-KNN, maximum entropy and EM imputation methods.
Points with *red*, *green*, *blue*, and *teal* color represent KNN, iteration KNN, EM and maximum entropy imputation methods, respectively.

On the other hand, they have significant differences in computer time. Figure 3.6 gives the time that it takes to impute missing values for various missing rate. For instant, EM imputation is time-consuming for large data set, it requires 62.74 hours to impute missing values for missing rate 0.6 compared to 7.57, 21.92 and 32.29 hours for maximum entropy, iteration-KNN, and KNN imputation methods. As a conclusion, both iteration-KNN and maximum entropy can yield acceptable error rates with reasonable computation demand for massive missing data.

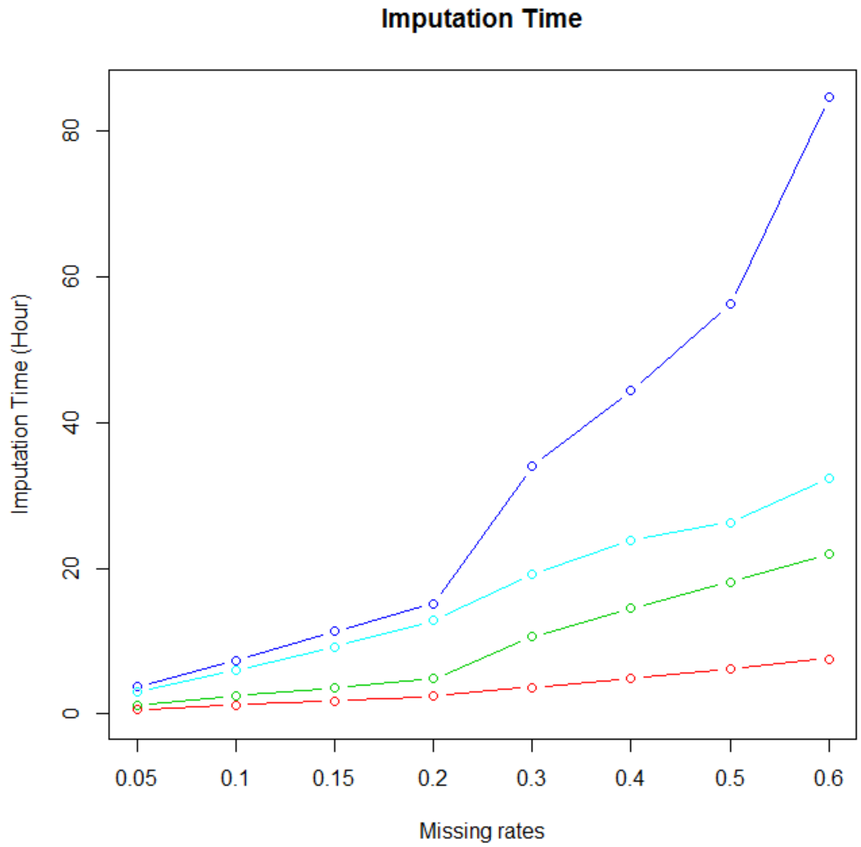

Figure 3.6: Comparison of imputation time for KNN, iteration-KNN, maximum entropy and EM imputation methods (Unit in hour).
Points with *red*, *green*, *blue*, and *teal* color represent KNN, iteration KNN, EM and maximum entropy imputation methods, respectively.

Copyright© Zilong Wang, 2012.

Chapter 4 Summary and Discussion

This dissertation is devoted to the analysis of binary data via spatial-temporal autologitic regression models. Specifically, we have carefully examined the traditional spatial-temporal model and developed the centered spatial-temporal autologistic regression model, where the centered model solves the parameters interpretation problems in the traditional model. We also propose two new imputation methods, iteration-KNN and maximum entropy imputation, which are both effective ways to impute spatial-temporal missing values considering efficiency and accuracy for spatial-temporal missing data.

The centered spatial-temporal autologistic regression model is developed to obtain reasonable parameter interpretations across varying levels of spatial and temporal statistical dependence. The traditional spatial-temporal autologistic regression model by Zheng and Zhu (2008) is an important and widely used model to analyze binary data measured repeatedly over time on a spatial lattice, which can account for covariates, spatial dependence, and temporal dependence simultaneously. However, it has been presented that the traditional spatial-temporal autologistic model fails to provide meaningful interpretations in chapter 2., the traditional model's non-negative autocovariate could bias the realizations towards 1. To overcome this interpretation problem, we have considered a spatial-temporal autologistic regression model with centered parameterization, which is an alternative parameterization that can help to alleviate this difficulty.

For centered model, we have developed statistical inference based on expectation-maximization pseudo-likelihood (EMPL), Monte Carlo expectation-maximization likelihood (MCEML), and studied the performance of Bayesian inference for the centered model. Both simulation study and real data example show that the performance of the MCEML is comparable to the Bayesian approach and these two approaches are more statistically efficient than EMPL. We compare the prediction performance of the centered spatial-temporal autologistic regression model with the traditional one using a real data example. It has been shown that these two models generate comparable predictions. Since the statistical inference based on the centered parameterization is much more computationally intensive for the centered model, we suggest to just use the traditional model if prediction is of primary interest. In our simulation study (not shown here), we have also studied the edge effect of the spatial lattice on inference. The analysis shows that statistical inference is not sensitive to the shape of the lattice.

Spatial-temporal missing data are MAR cases required to impute missing values to count for spatial and temporal effects in statistical analysis. Considering efficiency and accuracy, we have proposed two new imputation methods: iteration-KNN imputation and maximum entropy imputation. Iteration-KNN imputation is an iterative non-parametric algorithm for imputing missing values, which uses a KNN imputation repeatedly to improve accuracy with high computing speed. Also, it can suffer from the negative effects of model failure, so that it has more stable performance when observed data can not reserve the properties of original data set. Iteration-KNN imputation is a combination of point estimates by non-parametric KNN and distribution estimates by EM, which estimates sequential multiple values for each missing value. Moreover, we have proposed a maximum entropy imputation for spatial-temporal incomplete data, which follows maximum entropy distribution with additional constrains. When missing rate is high, maximum entropy imputation is the only reasonable way to estimate missing values. As is shown in chapter 3, both simulation and real data application present iteration-KNN and maximum entropy imputation methods are effective ways to deal with missing values, which can yield smaller error rates than KNN and need less computation time than EM for missing data.

The final purpose of every theoretical research is to be applied in the real world. For the centered spatial-temporal autologistic regression model, future work focuses on creating corresponding R package for spatial-temporal data researcher/user for solving research/application problems. For missing data, future research can focus on extending iteration-KNN and maximum entropy imputation to high-dimensional space data, and discussion on more efficient and more accurate imputation methods for spatial-temporal data.

Copyright© Zilong Wang, 2012.

Appendices

A. Tables for centered spatial-temporal autologistic regression model

Table 1: Simulation in 5*5 Lattice and 5 time points

True	MPLE	SE
(1, -0.5, 0.1, 0.1)	(0.906, -0.471, -0.006, 0.038)	(0.697, 0.231, 0.315, 0.334)
(1, -0.5, 0.1, 0.5)	(1.001, -0.502, 0.049, 0.440)	(0.695, 0.228, 0.343, 0.393)
(1, -0.5, 0.1, 0.9)	(0.959, -0.526, 0.081, 0.947)	(0.832, 0.229, 0.313, 0.417)
(1, -0.5, 0.5, 0.1)	(0.924, -0.497, 0.489, 0.033)	(0.778, 0.262, 0.324, 0.434)
(1, -0.5, 0.5, 0.5)	(1.001, -0.539, 0.492, 0.548)	(0.916, 0.259, 0.307, 0.370)
(1, -0.5, 0.5, 0.9)	(0.999, -0.499, 0.495, 0.934)	(1.247, 0.232, 0.343, 0.463)
(1, -0.5, 0.9, 0.1)	(0.957, -0.510, 0.915, 0.080)	(1.137, 0.243, 0.329, 0.412)
(1, -0.5, 0.9, 0.5)	(1.185, -0.475, 0.863, 0.478)	(1.334, 0.281, 0.354, 0.375)
(1, -0.5, 0.9, 0.9)	(2.699, -0.479, 0.915, 1.025)	(1.684, 0.284, 0.337, 0.434)

Table 2: Simulation in 10*10 Lattice and 5 time points

True	MPLE	SE
(1, -0.5, 0.1, 0.1)	(1.062, -0.522, 0.120, 0.085)	(0.276, 0.089, 0.155, 0.199)
(1, -0.5, 0.1, 0.5)	(1.034, -0.509, 0.071, 0.523)	(0.323, 0.105, 0.165, 0.214)
(1, -0.5, 0.1, 0.9)	(0.939, -0.499, 0.111, 0.906)	(0.296, 0.093, 0.151, 0.198)
(1, -0.5, 0.5, 0.1)	(1.019, -0.512, 0.507, 0.096)	(0.321, 0.103, 0.154, 0.160)
(1, -0.5, 0.5, 0.5)	(0.946, -0.494, 0.460, 0.495)	(0.351, 0.103, 0.148, 0.197)
(1, -0.5, 0.5, 0.9)	(0.908, -0.491, 0.489, 0.882)	(0.466, 0.096, 0.158, 0.166)
(1, -0.5, 0.9, 0.1)	(0.899, -0.483, 0.895, 0.088)	(0.449, 0.107, 0.130, 0.181)
(1, -0.5, 0.9, 0.5)	(1.266, -0.494, 0.878, 0.500)	(0.953, 0.092, 0.130, 0.188)
(1, -0.5, 0.9, 0.9)	(3.266, -0.464, 0.888, 0.948)	(0.728, 0.117, 0.158, 0.223)

Table 3: Simulation in 20*20 Lattice and 5 time points

True	MPLE	SE
(1, -0.5, 0.1, 0.1)	(1.005, -0.503, 0.113, 0.084)	(0.158, 0.052, 0.069, 0.088)
(1, -0.5, 0.1, 0.5)	(0.970, -0.490, 0.092, 0.486)	(0.152, 0.048, 0.069, 0.088)
(1, -0.5, 0.1, 0.9)	(1.015, -0.507, 0.108, 0.893)	(0.138, 0.047, 0.079, 0.101)
(1, -0.5, 0.5, 0.1)	(0.957, -0.485, 0.498, 0.098)	(0.155, 0.051, 0.076, 0.092)
(1, -0.5, 0.5, 0.5)	(0.934, -0.488, 0.512, 0.497)	(0.188, 0.056, 0.070, 0.099)
(1, -0.5, 0.5, 0.9)	(0.948, -0.490, 0.500, 0.894)	(0.273, 0.053, 0.060, 0.087)
(1, -0.5, 0.9, 0.1)	(0.891, -0.465, 0.879, 0.108)	(0.247, 0.047, 0.070, 0.092)
(1, -0.5, 0.9, 0.5)	(2.099, -0.479, 0.897, 0.506)	(0.787, 0.048, 0.075, 0.093)
(1, -0.5, 0.9, 0.9)	(3.485, -0.468, 0.899, 0.929)	(0.249, 0.056, 0.075, 0.109)

Table 4: Simulation in 5*5 Lattice and 10 time points

True	MPLE	SE
(1, -0.5, 0.1, 0.1)	(1.041, -0.523, 0.048, 0.092)	(0.382, 0.118, 0.237, 0.287)
(1, -0.5, 0.1, 0.5)	(1.052, -0.518, 0.098, 0.511)	(0.445, 0.135, 0.231, 0.276)
(1, -0.5, 0.1, 0.9)	(1.021, -0.530, 0.071, 0.949)	(0.466, 0.148, 0.235, 0.300)
(1, -0.5, 0.5, 0.1)	(1.031, -0.520, 0.458, 0.103)	(0.484, 0.141, 0.245, 0.272)
(1, -0.5, 0.5, 0.5)	(1.077, -0.534, 0.491, 0.525)	(0.569, 0.153, 0.234, 0.322)
(1, -0.5, 0.5, 0.9)	(1.003, -0.519, 0.493, 0.906)	(0.618, 0.136, 0.212, 0.256)
(1, -0.5, 0.9, 0.1)	(0.908, -0.489, 0.895, 0.106)	(0.601, 0.163, 0.194, 0.278)
(1, -0.5, 0.9, 0.5)	(1.034, -0.517, 0.883, 0.505)	(0.876, 0.166, 0.204, 0.281)
(1, -0.5, 0.9, 0.9)	(2.359, -0.478, 0.859, 0.953)	(1.259, 0.154, 0.213, 0.336)

Table 5: Simulation in 10*10 Lattice and 10 time points

True	MPLE	SE
(1, -0.5, 0.1, 0.1)	(1.009, -0.507, 0.106, 0.088)	(0.212, 0.071, 0.106, 0.132)
(1, -0.5, 0.1, 0.5)	(0.997, -0.502, 0.109, 0.488)	(0.212, 0.068, 0.112, 0.134)
(1, -0.5, 0.1, 0.9)	(0.977, -0.495, 0.077, 0.878)	(0.216, 0.070, 0.104, 0.150)
(1, -0.5, 0.5, 0.1)	(0.950, -0.483, 0.496, 0.083)	(0.224, 0.071, 0.108, 0.113)
(1, -0.5, 0.5, 0.5)	(0.994, -0.502, 0.498, 0.480)	(0.251, 0.071, 0.108, 0.138)
(1, -0.5, 0.5, 0.9)	(0.881, -0.482, 0.503, 0.899)	(0.387, 0.083, 0.109, 0.133)
(1, -0.5, 0.9, 0.1)	(0.965, -0.492, 0.906, 0.094)	(0.315, 0.072, 0.096, 0.138)
(1, -0.5, 0.9, 0.5)	(1.288, -0.502, 0.894, 0.507)	(0.897, 0.087, 0.093, 0.122)
(1, -0.5, 0.9, 0.9)	(3.404, -0.514, 0.902, 0.928)	(0.434, 0.090, 0.104, 0.173)

Table 6: Simulation in 20*20 Lattice and 10 time points

True	MPLE	SE
(1, -0.5, 0.1, 0.1)	(1.013, -0.504, 0.104, 0.103)	(0.107, 0.034, 0.051, 0.066)
(1, -0.5, 0.1, 0.5)	(1.004, -0.500, 0.090, 0.499)	(0.105, 0.034, 0.049, 0.059)
(1, -0.5, 0.1, 0.9)	(0.995, -0.499, 0.096, 0.909)	(0.123, 0.037, 0.051, 0.070)
(1, -0.5, 0.5, 0.1)	(0.972, -0.493, 0.507, 0.105)	(0.115, 0.036, 0.047, 0.066)
(1, -0.5, 0.5, 0.5)	(0.979, -0.493, 0.494, 0.498)	(0.120, 0.034, 0.051, 0.066)
(1, -0.5, 0.5, 0.9)	(0.990, -0.499, 0.499, 0.903)	(0.178, 0.037, 0.047, 0.063)
(1, -0.5, 0.9, 0.1)	(0.950, -0.493, 0.897, 0.107)	(0.181, 0.035, 0.048, 0.058)
(1, -0.5, 0.9, 0.5)	(2.055, -0.487, 0.893, 0.505)	(0.644, 0.040, 0.049, 0.071)
(1, -0.5, 0.9, 0.9)	(3.530, -0.486, 0.893, 0.937)	(0.191, 0.044, 0.063, 0.083)

Table 7: Simulation in 5*5 Lattice and 20 time points

True	MPLE	SE
(1, -0.5, 0.1, 0.1)	(1.029, -0.509, 0.081, 0.085)	(0.316, 0.108, 0.168, 0.202)
(1, -0.5, 0.1, 0.5)	(1.049, -0.521, 0.059, 0.493)	(0.316, 0.096, 0.180, 0.179)
(1, -0.5, 0.1, 0.9)	(1.059, -0.514, 0.094, 0.875)	(0.340, 0.101, 0.163, 0.210)
(1, -0.5, 0.5, 0.1)	(0.968, -0.491, 0.463, 0.111)	(0.343, 0.096, 0.151, 0.200)
(1, -0.5, 0.5, 0.5)	(0.981, -0.503, 0.493, 0.499)	(0.351, 0.103, 0.146, 0.170)
(1, -0.5, 0.5, 0.9)	(0.937, -0.499, 0.495, 0.892)	(0.408, 0.104, 0.161, 0.206)
(1, -0.5, 0.9, 0.1)	(0.986, -0.507, 0.885, 0.105)	(0.418, 0.106, 0.164, 0.197)
(1, -0.5, 0.9, 0.5)	(0.814, -0.484, 0.901, 0.520)	(0.631, 0.129, 0.150, 0.196)
(1, -0.5, 0.9, 0.9)	(2.868, -0.513, 0.880, 0.942)	(0.712, 0.114, 0.148, 0.195)

Table 8: Simulation in 10*10 Lattice and 20 time points

True	MPLE	SE
(1, -0.5, 0.1, 0.1)	(1.013, -0.506, 0.094, 0.098)	(0.152, 0.048, 0.090, 0.091)
(1, -0.5, 0.1, 0.5)	(0.986, -0.496, 0.097, 0.495)	(0.145, 0.049, 0.090, 0.096)
(1, -0.5, 0.1, 0.9)	(1.011, -0.508, 0.095, 0.880)	(0.170, 0.050, 0.066, 0.085)
(1, -0.5, 0.5, 0.1)	(1.000, -0.499, 0.496, 0.089)	(0.170, 0.058, 0.070, 0.101)
(1, -0.5, 0.5, 0.5)	(0.996, -0.503, 0.502, 0.497)	(0.167, 0.047, 0.063, 0.101)
(1, -0.5, 0.5, 0.9)	(0.993, -0.497, 0.482, 0.912)	(0.227, 0.057, 0.067, 0.098)
(1, -0.5, 0.9, 0.1)	(1.005, -0.500, 0.895, 0.105)	(0.227, 0.049, 0.061, 0.086)
(1, -0.5, 0.9, 0.5)	(1.187, -0.496, 0.901, 0.498)	(0.657, 0.048, 0.075, 0.091)
(1, -0.5, 0.9, 0.9)	(3.357, -0.486, 0.888, 0.985)	(0.282, 0.059, 0.073, 0.121)

Table 9: Simulation in 20*20 Lattice and 20 time points

True	MPLE	SE
(1, -0.5, 0.1, 0.1)	(1.003, -0.502, 0.107, 0.108)	(0.080, 0.027, 0.033, 0.049)
(1, -0.5, 0.1, 0.5)	(0.994, -0.498, 0.102, 0.499)	(0.074, 0.024, 0.033, 0.047)
(1, -0.5, 0.1, 0.9)	(1.008, -0.501, 0.101, 0.894)	(0.087, 0.029, 0.033, 0.053)
(1, -0.5, 0.5, 0.1)	(0.992, -0.498, 0.501, 0.101)	(0.082, 0.026, 0.038, 0.055)
(1, -0.5, 0.5, 0.5)	(0.978, -0.495, 0.501, 0.502)	(0.088, 0.025, 0.033, 0.049)
(1, -0.5, 0.5, 0.9)	(0.985, -0.496, 0.502, 0.890)	(0.125, 0.029, 0.035, 0.051)
(1, -0.5, 0.9, 0.1)	(0.991, -0.496, 0.897, 0.103)	(0.123, 0.029, 0.031, 0.044)
(1, -0.5, 0.9, 0.5)	(2.286, -0.495, 0.899, 0.513)	(0.487, 0.029, 0.033, 0.048)
(1, -0.5, 0.9, 0.9)	(3.571, -0.492, 0.899, 0.946)	(0.143, 0.036, 0.042, 0.062)

B. Tables for missing data imputation

Table 10: Error Rates for Beta=(1,0.5,0.1,0.5)

Imputation	0.05	0.1	0.15	0.2	0.3	0.4	0.5	0.6
KNN	0.31	0.36	0.3967	0.41	0.4333	0.43	0.48	0.485
Iteration-KNN	0.31	0.31	0.3133	0.335	0.3467	0.365	0.392	0.41
EM	0.26	0.28	0.2933	0.296	0.3567	0.3925	0.472	0.51
Maximum Entropy	0.28	0.3	0.3033	0.3	0.3233	0.335	0.33	0.3733

Table 11: Error Rates for Beta=(1,0.5,0.3,0.5)

Imputation	0.05	0.1	0.15	0.2	0.3	0.4	0.5	0.6
KNN	0.24	0.28	0.2967	0.305	0.3267	0.3475	0.348	0.3483
Iteration-KNN	0.24	0.26	0.27	0.3	0.3033	0.3075	0.312	0.315
EM	0.22	0.22	0.23	0.26	0.27	0.3075	0.334	0.345
Maximum Entropy	0.22	0.23	0.2433	0.25	0.2767	0.2825	0.298	0.305

Table 12: Error Rates for Beta=(1,0.5,0.5,0.5)

Imputation	0.05	0.1	0.15	0.2	0.3	0.4	0.5	0.6
KNN	0.12	0.12	0.1333	0.145	0.1567	0.1425	0.178	0.1883
Iteration-KNN	0.12	0.13	0.1333	0.135	0.1467	0.145	0.148	0.15
EM	0.12	0.12	0.1267	0.13	0.1367	0.14	0.142	0.1433
Maximum Entropy	0.12	0.12	0.1267	0.135	0.14	0.1425	0.14	0.1417

Table 13: APD for Beta=(1,0.5,0.1,0.5)

Imputation	0.05	0.1	0.15	0.2	0.3	0.4	0.5	0.6
KNN	0.1707	0.19	0.203	0.2155	0.2313	0.2392	0.2727	0.2856
Iteration-KNN	0.1626	0.1699	0.1728	0.1786	0.1797	0.1833	0.187	0.2012
EM	0.0983	0.1106	0.128	0.1293	0.1815	0.1839	0.2666	0.2889
Maximum Entropy	0.1179	0.1269	0.1327	0.1425	0.1453	0.1468	0.1559	0.1668

Table 14: APD for Beta=(1,0.5,0.3,0.5)

Imputation	0.05	0.1	0.15	0.2	0.3	0.4	0.5	0.6
KNN	0.1311	0.1734	0.1831	0.1925	0.2158	0.2335	0.2515	0.2662
Iteration-KNN	0.1278	0.1329	0.141	0.1542	0.1634	0.1697	0.1807	0.1966
EM	0.0668	0.0739	0.0906	0.0936	0.0947	0.1547	0.2157	0.2676
Maximum Entropy	0.0824	0.0855	0.0931	0.1055	0.1061	0.1063	0.1154	0.1261

Table 15: APD for Beta=(1,0.5,0.5,0.5)

Imputation	0.05	0.1	0.15	0.2	0.3	0.4	0.5	0.6
KNN	0.1003	0.1066	0.1173	0.1407	0.1514	0.1601	0.1804	0.1806
Iteration-KNN	0.1034	0.103	0.1167	0.1225	0.1245	0.125	0.1275	0.1283
EM	0.0452	0.0505	0.0523	0.0596	0.0642	0.0692	0.078	0.081
Maximum Entropy	0.0631	0.0648	0.0667	0.0702	0.0718	0.0721	0.08	0.0805

Table 16: Error Rates for MPB Data

Imputation	0.05	0.1	0.15	0.2	0.3	0.4	0.5	0.6
KNN	0.0167	0.0198	0.0207	0.0239	0.0315	0.0377	0.0461	0.0533
Iteration-KNN	0.0159	0.0186	0.0195	0.0218	0.0285	0.0357	0.0403	0.0429
EM	0.0145	0.0152	0.0171	0.020	0.0268	0.0295	0.0317	0.0334
Maximum Entropy	0.0147	0.0159	0.0183	0.0206	0.0270	0.0329	0.0349	0.0392

Table 17: Computing Time for MPB Data (Unit in hour)

Imputation	0.05	0.1	0.15	0.2	0.3	0.4	0.5	0.6
KNN	0.58	1.19	1.81	2.43	3.65	4.92	6.24	7.57
Iteration-KNN	1.18	2.41	3.59	4.82	10.56	14.55	18.07	21.92
EM	3.74	7.28	11.34	15.12	33.98	44.35	56.27	84.74
Maximum Entropy	3.02	5.98	9.15	12.82	19.23	23.86	26.3	32.29

C. R code for centered spatial-temporal autologistic regression model

```
### BGS

sample.gibbs = function(yt, xt, sidsloc, sidsnboor,

                        tt.tb, tt.te, n.sample, beta)
```

```
{
   y.sample = matrix(0, ncol=n.sample, nrow=nt)
   pt.sample = matrix(0, ncol=n.sample, nrow=nt)
   center = matrix(0, ncol=1, nrow=nt)
   for (i in 1:nt)
   {
      temp = exp(beta0 + beta1*xt[i])
      center[i] = temp/(1+temp)
   }
   order = sample(nt)
   for (i in (1:(n.sample+1000))){
    ttt=.C("gibbs_sim_100_4dim",
     nt=as.integer(nt), m=as.integer(m),
     tt=as.integer(tt), sidsloc=as.integer(sidsloc),
     sidsnboor=as.integer(sidsnboor),
     order=as.integer(order),
     yt=as.double(yt), ytte=as.double(tt.te),
     yttb=as.double(tt.tb), cp=as.double(center),
     xtsp=as.double(matrix(0, ncol=1, nrow=nt)),
     xtte=as.double(matrix(0, ncol=1, nrow=nt)),
     beta0=as.double(beta0), beta1=as.double(beta1),
     beta2=as.double(beta2), beta3=as.double(beta3),
     pp=as.double(matrix(0,ncol=1, nrow=nt)),
     xt=as.double(xt) )
    yt = ttt$yt
    y.sample[,i] = ttt$yt
    pt.sample[,i] = ttt$pp
   }
   return(y.sample)
}
### PS
yt.ub.ini=matrix(1, ncol=1, nrow=nt)
yt.lb.ini=matrix(0, ncol=1, nrow=nt)
perfect.sim=function(yt.ub.ini,yt.lb.ini,xt,
                    sidsloc,sidsnboor,
                    tt.tb,tt.te,beta){
   center = matrix(0, ncol=1, nrow=n)
   for (i in 1:n){
      temp = exp(beta0 + beta1*xt[i*tt])
      center[i] = temp/(1+temp)
   }
  mt = 0
  repeat{
   mt = mt+1
   yt.ub = yt.ub.ini
   yt.lb = yt.lb.ini
   for (xxx in -(2^(mt-1)):-1)
   {
    for (jt in 1:nt)
    {
      ttt=.C("gibbs_PS_Test",
         j=jt, m=as.integer(m), tt=as.integer(tt),
         sidsloc=as.integer(sidsloc),
         sidsnboor=as.integer(sidsnboor),
      order=as.integer(order), yt=as.double(yt.ub),
      ytte=as.double(tt.te), yttb=as.double(tt.tb),
      cp=as.double(center), xt=as.double(xt),
      xtte=as.double(matrix(0, ncol=1, nrow=nt)),
      xtsp=as.double(matrix(0, ncol=1, nrow=nt)),
      beta0=as.double(beta0), beta1=as.double(beta1),
      beta2=as.double(beta2), beta3=as.double(beta3),
      py = 0 )$py
   for (te in 1:mt){
      if (xxx == -2^(te-1) && jt == 1){
         set.seed(seeds[te])
      }
   }
   yt.ub[jt] = rbinom(1,1,ttt)
 }
}
for (yyy in -(2^(mt-1)):-1)
{
 for (wt in 1:nt)
 {
  sss=.C("gibbs_PS_Test", j=wt, m=as.integer(m),
  tt=as.integer(tt),
  sidsloc=as.integer(sidsloc),
  sidsnboor=as.integer(sidsnboor),
  order=as.integer(order), yt=as.double(yt.lb),
  ytte=as.double(tt.te), yttb=as.double(tt.tb),
  cp=as.double(center), xt=as.double(xt),
      xtte=as.double(matrix(0, ncol=1, nrow=nt)),
      xtsp=as.double(matrix(0, ncol=1, nrow=nt)),
      beta0=as.double(beta0), beta1=as.double(beta1),
      beta2=as.double(beta2), beta3=as.double(beta3),
      py = 0 )$py
       for (de in 1:mt){
          if (yyy == -2^(de-1) && wt == 1){
             set.seed(seeds[de])
          }
       }
       yt.lb[wt] = rbinom(1,1,sss)
    }
   }
   if ( sum(abs(yt.ub - yt.lb)) == 0 ){
      cat ("UB & LB Match", fill=T)
      break
   }
   else if (mt >= nps){
      cat ("Exceeding the max MT", fill=T)
      break
   }
  }
   return(data.frame(yt.ub, yt.lb))
}
### EMPLs
pseudo=function(yt,xt,sidsloc,sidsnboor,
       tt.tb,tt.te,niter,tol.low)
```

```
{
   res.0 = glm(yt ~ xt, family=binomial("logit"))
   beta0 = as.numeric(res.0$coefficients[1])
   iter = 0
   repeat{
      iter= iter+1
      center = matrix(0, ncol=1, nrow=nt)
      for (i in 1:nt)
      {
         temp = exp(beta0 + beta1*xt[i])
         center[i] = temp/(1+temp)
      }
   datanew=.C("centerdata_sim_100_4dim",
      n=as.integer(n), tt=as.integer(tt),
      m=as.integer(m), yt=as.double(yt),
      yttb=as.double(tt.tb),
      ytte=as.double(tt.te),
      xtsp=as.double(matrix(0,nrow=nt,ncol=1)),
      xtte=as.double(matrix(0, nrow=nt, ncol=1)),
      sidsloc=as.integer(sidsloc),
      sidsnboor=as.integer(sidsnboor),
      center=as.double(center) )
   res=glm(yt~xt+as.matrix(datanew$xtsp)+
             as.matrix(datanew$xtte),
             family=binomial("logit"))
  if (iter>=niter){
  cat ("### exceed the maximum iteration number", fill=T)
```

```
         break
  }
  else if (
  (abs(as.numeric(res$coefficients[1])-beta0)<tol.low) &
  (abs(as.numeric(res$coefficients[2])-beta1)<tol.low)){
         break
      }
      else      {
         beta0 = as.numeric(res$coefficients[1])
         beta1 = as.numeric(res$coefficients[2])
      }
   }
   out.beta = res$coefficients
   return(out.beta)
}
### MCEML Function
MCEML=function(yi,xt,sidsloc,sidsnboor,
       tt.tb,tt.te,niter,ini,base,ys)
{
   center.base = matrix(0, ncol=1, nrow=nt)
   for (i in 1:nt) {
      temp = exp(base0 + base1*xt[i])
      center.base[i] = temp/(1+temp)
   }
   z.base = matrix(nrow=mt, ncol=4, -999)
   for (i in 1:mt)
   {
```

```
z.base.temp = .C("centerdata_sim_100_4dim", n=as.integer(n),
tt=as.integer(tt), m=as.integer(m),
yt=as.double(ys[i,]), yttb=as.double(tt.tb),
ytte=as.double(tt.te),xtsp=as.double(matrix(0,nrow=nt,ncol=1)),
xtte=as.double(matrix(0, nrow=nt, ncol=1)),
sidsloc=as.integer(sidsloc),sidsnboor=as.integer(sidsnboor),
center=as.double(center.base) )
y.center = ys[i,] - center.base
z.base[i,1] = sum(y.center)
z.base[i,2] = sum(xt*y.center)
z.base[i,3] = 0.5*sum(y.center*as.matrix(z.base.temp$xtsp))
temp.z4 = 0
temp.z4.board = 0 #for board points in temporal part
for (j in 1:n){
 for (k in 2:tt)
 {
 temp.z4=temp.z4+y.center[(j-1)*tt+k]*y.center[(j-1)*tt+k-1]
 }
 temp.z4.board=temp.z4.board +
 (tt.te[j]-center.base[j*tt-1])*y.center[j*tt-1] +
  y.center[(j-1)*tt+1]*(tt.tb[j]-center.base[(j-1)*tt+1])
  }
  z.base[i,4] = temp.z4 + temp.z4.board
}
base.core = as.vector(t(as.matrix(base))%*%t(z.base))
iter = 0
center = matrix(0, ncol=1, nrow=nt)
```

```
z.beta = matrix(nrow=mt, ncol=4, -999)
scale.newton = 1
repeat{
   iter= iter+1
 if (iter==1) {
   for (i in 1:nt){
      temp = exp(beta0 + beta1*xt[i])
      center[i] = temp/(1+temp)
   }
   for (i in 1:mt) {
  z.beta.temp = .C("centerdata_sim_100_4dim", n=as.integer(n),
  tt=as.integer(tt), m=as.integer(m),
  yt=as.double(ys[i,]), yttb=as.double(tt.tb),
  ytte=as.double(tt.te),xtsp=as.double(matrix(0,nrow=nt,ncol=1)),
  xtte=as.double(matrix(0, nrow=nt, ncol=1)),
  sidsloc=as.integer(sidsloc),sidsnboor=as.integer(sidsnboor),
  center=as.double(center.base) )
   y.center = ys[i,] - center
   z.beta[i,1] = sum(y.center)
   z.beta[i,2] = sum(xt*y.center)
   z.beta[i,3] = 0.5*sum(y.center*as.matrix(z.beta.temp$xtsp))
   temp.z4 = 0
   temp.z4.board = 0
   for (j in 1:n){
    for (k in 2:tt){
       temp.z4=temp.z4+y.center[(j-1)*tt+k]*y.center[(j-1)*tt+k-1]
         }
    temp.z4.board=temp.z4.board +
    (tt.te[j]-center[j*tt-1])*y.center[j*tt-1] +
    y.center[(j-1)*tt+1]*(tt.tb[j]-center[(j-1)*tt+1])
      }
     z.beta[i,4] = temp.z4 + temp.z4.board
  }
  z.true = z.beta[yi,]
  beta.core.temp = matrix(nrow=mt, ncol=4, -999)
  for (i in 1:mt){
      beta.core.temp[i,] = z.beta[i,] - z.true
   }
beta.core= s.vector(t(c(beta0,beta1,beta2,beta3))
       %*%t(beta.core.temp))
   wi = exp(beta.core - base.core)
   w = sum(wi)
   mle = log(mt) - log(w)
   if (mle > 10^10 || mle < 10^(-10)){
      cat ("!!!Initial MLE too small or big!!!", fill=T)
      break
   }
   beta = c(beta0, beta1, beta2, beta3)
 }
 else {
    if ( mle.new <= mle ) {
       if (iter > 2 || scale.newton < 0.15){
          # cat ("***Succeess***", fill=T)
          break
```

```
            }
            else{
               scale.newton = scale.newton/2
               iter = 0
            }
         }
         else {
            mle = mle.new
            beta = c(beta0, beta1, beta2, beta3)
            scale.newton = 1
         }
      }
   der.1st = -t(as.matrix(wi/w))%*%beta.core.temp
   der.2nd = matrix(nrow=4, ncol=4, 0)
   der.2nd[1,1] = (der.1st[1,1])^2 - sum(wi/w*(beta.core.temp[,1])^2)
   der.2nd[1,2] = der.1st[1,1]*der.1st[1,2] -
     sum(wi/w*beta.core.temp[,1]*beta.core.temp[,2])
   der.2nd[1,3] = der.1st[1,1]*der.1st[1,3] -
    sum(wi/w*beta.core.temp[,1]*beta.core.temp[,3])
   der.2nd[1,4] = der.1st[1,1]*der.1st[1,4] -
    sum(wi/w*beta.core.temp[,1]*beta.core.temp[,4])
   der.2nd[2,1] = der.2nd[1,2]
   der.2nd[2,2] = (der.1st[1,2])^2 - sum(wi/w*(beta.core.temp[,2])^2)
   der.2nd[2,3] = der.1st[1,2]*der.1st[1,3] -
    sum(wi/w*beta.core.temp[,2]*beta.core.temp[,3])
   der.2nd[2,4] = der.1st[1,2]*der.1st[1,4] -
    sum(wi/w*beta.core.temp[,2]*beta.core.temp[,4])
```

```
der.2nd[3,1] = der.2nd[1,3]
der.2nd[3,2] = der.2nd[2,3]
der.2nd[3,3] = (der.1st[1,3])^2 - sum(wi/w*(beta.core.temp[,3])^2)
der.2nd[3,4] = der.1st[1,3]*der.1st[1,4] -
 sum(wi/w*beta.core.temp[,3]*beta.core.temp[,4])
der.2nd[4,1] = der.2nd[1,4]
der.2nd[4,2] = der.2nd[2,4]
der.2nd[4,3] = der.2nd[3,4]
der.2nd[4,4] = (der.1st[1,4])^2 - sum(wi/w*(beta.core.temp[,4])^2)
     se.1 = 1/sqrt(abs(der.2nd[1,1]))
     se.2 = 1/sqrt(abs(der.2nd[2,2]))
     se.3 = 1/sqrt(abs(der.2nd[3,3]))
     se.4 = 1/sqrt(abs(der.2nd[4,4]))
if( sum(is.nan(der.2nd)) > 0 || sum(is.infinite(der.2nd)) > 0){
 cat ("@@@ inverse failed, der.2nd=", der.2nd, fill=T)
 break
     }
 else{
  inv.der.2nd = solve(der.2nd)
  beta.new = beta- t(scale.newton*(inv.der.2nd%*%t(der.1st)))
  if (iter>=niter){
   cat ("### exceed the maximum iteration number", fill=T)
    break
        }
    else{
           beta0 = as.numeric(beta.new[1])
           beta1 = as.numeric(beta.new[2])
        beta2 = as.numeric(beta.new[3])
        beta3 = as.numeric(beta.new[4])
        for (i in 1:nt){
           temp = exp(beta0 + beta1*xt[i])
           center[i] = temp/(1+temp)
        }
for (i in 1:mt){
 z.beta.temp = .C("centerdata_sim_100_4dim", n=as.integer(n),
 tt=as.integer(tt), m=as.integer(m),
 yt=as.double(ys[i,]), yttb=as.double(tt.tb),
 ytte=as.double(tt.te),xtsp=as.double(matrix(0,nrow=nt,ncol=1)),
 xtte=as.double(matrix(0, nrow=nt, ncol=1)),
 sidsloc=as.integer(sidsloc),sidsnboor=as.integer(sidsnboor),
 center=as.double(center.base) )
 y.center = ys[i,] - center
 z.beta[i,1] = sum(y.center)
 z.beta[i,2] = sum(xt*y.center)
 z.beta[i,3] = 0.5*sum(y.center*as.matrix(z.beta.temp$xtsp))
 temp.z4 = 0
 temp.z4.board = 0
 for (j in 1:n){
  for (k in 2:tt){
   temp.z4=temp.z4+y.center[(j-1)*tt+k]*y.center[(j-1)*tt+k-1]
   }
    temp.z4.board=temp.z4.board +
     (tt.te[j]-center[j*tt-1])*y.center[j*tt-1] +
     y.center[(j-1)*tt+1]*(tt.tb[j]-center[(j-1)*tt+1])
```

```
         }
         z.beta[i,4] = temp.z4 + temp.z4.board
         }
         z.true = z.beta[yi,]
         beta.core.temp = matrix(nrow=mt, ncol=4, -999)
         for (i in 1:mt){
         beta.core.temp[i,] = z.beta[i,] - z.true
            }
         beta.core = as.vector(t(c(beta0, beta1, beta2, beta3))
                        %*%t(beta.core.temp))
            wi = exp(beta.core - base.core)
            w = sum(wi)
            mle.new = log(mt) - log(w)
            cat ("mle.new = ", mle.new, fill=T)
        if (mle.new > 10^10 || mle.new < 10^(-10)){
        beta = c(beta0, beta1, beta2, beta3)
        cat ("!!!New MLE too small or big!!!", fill=T)
        break
           }
         }
      }
   }
   out.beta = beta
   betase = c(beta, se.1, se.2, se.3, se.4)
   return(betase)
}
### Main ###
```

```
dyn.load("gibbs_sim_100_4dim.dll")
dyn.load("centerdata_sim_100_4dim.dll")
MCEML = MCEML(yi = i, xt = as.numeric(xt.0), sidsloc = sidsloc,
        sidsnboor=sidsnboor, tt.tb = tt.tb, tt.te = tt.te,
        ini=beta.0, base=beta.0, ys=ys)
PL=pseudo(yt=as.numeric(ys[i,]),xt = as.numeric(xt.0),
  sidsloc = sidsloc, sidsnboor=sidsnboor,
  tt.tb = tt.tb,tt.te = tt.te)
```

D. R code for Imputation

```
### KNN Imputation

imputation.knn = function(data.knn, count.m, sidsloc,

                 sidsnboor, loc.m, tt.tb, tt.te){

   for (i.knn in (1:count.m)) {

      temp.knn = 0

      temp.i = 0

      for (j.knn in (1:length(sidsloc))){

   if((loc.m[i.knn]%%n!=0)&(sidsloc[j.knn]==loc.m[i.knn]%%n)){

    vvv = sidsnboor[j.knn] + (loc.m[i.knn]%/%n)*n

    temp.knn = temp.knn + data.knn[vvv,1]*data.knn[vvv,3]

    temp.i = temp.i +  data.knn[vvv,3]

         }

    if ( (loc.m[i.knn]%%n == 0) & (sidsloc[j.knn] == n) ){

     uuu = sidsnboor[j.knn] + ((loc.m[i.knn]%/%n)-1)*n

      temp.knn = temp.knn + data.knn[uuu,1]*data.knn[uuu,3]

    temp.i = temp.i +  data.knn[uuu,3]

         }

      }

    if ( loc.m[i.knn] <= n ){

      temp.knn = temp.knn + tt.tb[loc.m[i.knn]] +

      data.knn[(loc.m[i.knn]+n),1]*data.knn[(loc.m[i.knn]+n),3]

      temp.i = temp.i + 1 + data.knn[(loc.m[i.knn] + n), 3]

      }

      else if ( loc.m[i.knn] > (nt-n) ){

      temp.knn = temp.knn +

       data.knn[(loc.m[i.knn]-n), 1]*data.knn[(loc.m[i.knn] -n),3]+
       tt.te[loc.m[i.knn]+n-nt]
      temp.i = temp.i + data.knn[(loc.m[i.knn] - n), 3] + 1
      }
      else{
      temp.knn = temp.knn +
      data.knn[(loc.m[i.knn]-n),1]*data.knn[(loc.m[i.knn]-n),3]+
      data.knn[(loc.m[i.knn]+n),1]*data.knn[(loc.m[i.knn]+n),3]
      temp.i = temp.i + data.knn[(loc.m[i.knn] - n), 3] +
       data.knn[(loc.m[i.knn] + n), 3]
      }
      if (temp.i == 0){
         data.knn[loc.m[i.knn],1] = rbinom(1,1,0.5)
      }
      else{
         data.knn[loc.m[i.knn],2] = temp.knn/temp.i
         if ( data.knn[loc.m[i.knn],2] >= 0.5) {
            data.knn[loc.m[i.knn],1] = 1
         }
         else {
            data.knn[loc.m[i.knn],1] = 0
         }
      }
   }
   return (data.knn)
}
### EM-KNN Imputation
imputation.EMknn = function(data.EMknn, count.m, sidsloc,
  sidsnboor, loc.m, tt.tb, tt.te, niter, tol.low){
   iter = 0
   repeat{
      iter = iter + 1
      old = data.EMknn
      new = imputation.knn.v2(data.knn = old, count.m=count.m,
             sidsloc=sidsloc, sidsnboor=sidsnboor,
            loc.m=loc.m, tt.tb=tt.tb, tt.te=tt.te)
      D = sum(abs(new[,2] - old[,2]))
      if (iter >= niter){
      cat ("### exceed the maximum iteration number", fill=T)
         break
      }
      else if ( D < tol.low ){
         break
      }
      data.EMknn = new
 }
 return (data.EMknn)
}
### EM Imputation
imputation.EM = function(data.EM, xt.0, count.m, sidsloc,
   sidsnboor, loc.m, tt.tb, tt.te, niter, tol.low){
   iter = 0
   PL.EM=pseudo(yt=as.numeric(data.EM[,1]),xt=as.numeric(xt.0),
                  sidsloc=sidsloc,sidsnboor=sidsnboor,
                  tt.tb = tt.tb, tt.te = tt.te)
```

```
beta.EM = PL.EM[1:4]
MLE.log.old = -0.5*PL.EM[5]
old.data = data.EM
repeat{
   iter = iter + 1
   for (i.EM in 1:count.m){
   temp.s = 0
   for (j.EM in 1:length(sidsloc)){
   if((loc.m[i.EM]%%n!=0)&(sidsloc[j.EM]==loc.m[i.EM]%%n)){
            vvv = sidsnboor[j.EM] + (loc.m[i.EM]%/%n)*n
            temp.s = temp.s + data.EM[vvv,1]
         }
   if ( (loc.m[i.EM]%%n == 0) & (sidsloc[j.EM] == n) ){
            uuu = sidsnboor[j.EM] + ((loc.m[i.EM]%/%n)-1)*n
            temp.s = temp.s + data.EM[uuu,1]
         }
      }
      temp.t = 0
   if ( loc.m[i.EM] <= n ){
   temp.t=temp.t+tt.tb[loc.m[i.EM]]+data.EM[(loc.m[i.EM]+n),1]
   }
   else if ( loc.m[i.EM] > (nt-n) ){
         temp.t = temp.t + data.EM[(loc.m[i.EM] - n), 1]
                  + tt.te[loc.m[i.EM]+n-nt]
      }
   else{
   temp.t = temp.t + data.EM[(loc.m[i.EM] - n), 1] +
                  data.EM[(loc.m[i.EM] + n), 1]
   }


   temp.EM = beta.EM[1] + beta.EM[2]*xt.0[loc.m[i.EM]] +
     beta.EM[3]*temp.s + beta.EM[4]*temp.t
   data.EM[loc.m[i.EM],2] = exp(temp.EM)/(1+exp(temp.EM))
   if ( data.EM[loc.m[i.EM],2] >= 0.5){
         data.EM[loc.m[i.EM],1] = 1
      }
   else{
         data.EM[loc.m[i.EM],1] = 0
      }
   }
   test = sum(abs(data.EM[,1] - old.data[,1]))
   PL.EM=pseudo(yt=as.numeric(data.EM[,1]),
            xt=as.numeric(xt.0),
            sidsloc = sidsloc, sidsnboor=sidsnboor,
                  tt.tb = tt.tb, tt.te = tt.te)
   beta.EM = PL.EM[1:4]
   MLE.log.new = -0.5*PL.EM[5]
   if (iter >= niter){
   cat ("### exceed the maximum iteration number", fill=T)
   break
   }
   else if ( MLE.log.new <= MLE.log.old + tol.low ){
      break
   }
```

```
      MLE.log.old = MLE.log.new
      old.data = data.EM
 }
 return (data.EM)
}
### ME Imputation
imputation.ME=function(data.ME,xt.0,count.m,sidsloc,sidsnboor,
    loc.m, tt.tb, tt.te, niter, tol.low){
   iter = 0
   PL.ME=pseudo(yt=as.numeric(data.ME[,1]),xt=as.numeric(xt.0),
                  sidsloc = sidsloc, sidsnboor=sidsnboor,
                  tt.tb = tt.tb, tt.te = tt.te)
   beta.ME = PL.ME[1:4]
   old.ME = data.ME
   sample.ME = matrix(-999, nrow = nrow(data.ME), ncol = 100)
   cip.ME = matrix(-999, nrow = nrow(data.ME), ncol = 3)
   repeat {
      iter = iter + 1
      for (i.se in 1:100){
       sample.ME[,i.se]=ME.gibbs(yt=data.ME[,1], xt = xt.0,
       sidsloc=sidsloc,sidsnboor=sidsnboor,tt.tb=tt.tb,
       tt.te=tt.te, n.sample=10, beta=beta.ME)[,2]
      }
      for (i.ME in 1:count.m){
       se.temp = sd(sample.ME[loc.m[i.ME],])
       upper = min (1, data.ME[loc.m[i.ME], 2] + 1.96*se.temp)
       lower = max (0, data.ME[loc.m[i.ME], 2] - 1.96*se.temp)
      if (key.ME > upper){
           data.ME[loc.m[i.ME], 2] = upper
        }
        else if (key.ME < lower){
           data.ME[loc.m[i.ME], 2] = lower
        }

        if (data.ME[loc.m[i.ME],2] >= 0.5){
           data.ME[loc.m[i.ME],1] = 1
        }
        else{
           data.ME[loc.m[i.ME],1] = 0
        }
     }
     test = sum(abs(data.ME[,1] - old.ME[,1]))
PL.ME=pseudo(yt=as.numeric(data.ME[,1]),xt=as.numeric(xt.0),
           sidsloc = sidsloc, sidsnboor=sidsnboor,
           tt.tb = tt.tb, tt.te = tt.te)

     beta.ME = PL.ME[1:4]
     diff = sum(abs(data.ME[,2] - old.ME[,2]))/count.m
     if (iter >= niter{
      cat ("###exceed the maximum iteration number",fill=T)
      break
     }
     else if ( diff <= tol.low ){
        break
      }
      old.ME = data.ME
 }
 return (data.ME)
}
### Main ###
ptm <- proc.time()
res.imp.knn = imputation.knn (data.knn = as.matrix(imp.knn),
  count.m=count.m, sidsloc=sidsloc, sidsnboor=sidsnboor,
  loc.m=loc.m, tt.tb=tt.tb, tt.te=tt.te)
proc.time() - ptm
ptm <- proc.time()
imp.EMknn = imputation.knn (data.knn = res.spi, count.m=count.m,
 sidsloc=sidsloc, sidsnboor=sidsnboor, loc.m=loc.m,
 tt.tb=tt.tb, tt.te=tt.te)
res.imp.EMknn = imputation.EMknn (data.EMknn = imp.EMknn,
 count.m=count.m, sidsloc=sidsloc, sidsnboor=sidsnboor,
 loc.m=loc.m,tt.tb=tt.tb,tt.te=tt.te,niter=100,tol.low=0.0001)
proc.time() - ptm
ptm <- proc.time()
imp.EM = imputation.knn (data.knn = res.spi, count.m=count.m,
 sidsloc=sidsloc, sidsnboor=sidsnboor, loc.m=loc.m,
 tt.tb=tt.tb, tt.te=tt.te)
res.imp.EM = imputation.EM (data.EM = imp.EM, xt.0=xt.0,
 count.m=count.m, sidsloc=sidsloc, sidsnboor=sidsnboor,
 loc.m=loc.m,tt.tb=tt.tb,tt.te=tt.te,niter=20,tol.low=0.0001)
proc.time() - ptm
```

```
ptm <- proc.time()
imp.ME = imputation.knn (data.knn = res.spi, count.m=count.m,
 sidsloc=sidsloc, sidsnboor=sidsnboor, loc.m=loc.m,
 tt.tb=tt.tb, tt.te=tt.te)
res.imp.ME = imputation.ME (data.ME = imp.ME, xt.0=xt.0,
 count.m=count.m, sidsloc=sidsloc, sidsnboor=sidsnboor,
 loc.m=loc.m,tt.tb=tt.tb,tt.te=tt.te,niter=20,tol.low=0.0001)
proc.time() - ptm
```

Copyright© Zilong Wang, 2012.

Bibliography

[1] G.E. Batista and M.C. Monard. An analysis of four missing data treatment methods for supervised learning. *Applied Artificial Intelligence*, 17(5-6):519–533, 2003.

[2] J. Besag. Nearest-neighbour systems and the auto-logistic model for binary data. *Journal of the Royal Statistical Society Series B*, 34:75–83, 1972.

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

[4] J. Besag. Statistical analysis of non-lattice data. *The Statistician*, 24:179–195, 1975.

[5] T. Bishop and T. Ulrych. Maximum entropy spectral analysis and autoregressive decomposition. *Reviews of Geophysics and Space Physics*, 1975.

[6] A. Bolotin. A new method of multiple imputation for completely (or almost completely) missing data. In *Proceedings of the 12th WSEAS international conference on Mathematical and computational methods in science and engineering*, pages 34–45. World Scientific and Engineering Academy and Society (WSEAS).

[7] P. C. Caragea and M. S. Kaiser. Autologistic models with interpretable parameters. *Journal of Agricultural, Biological, and Environmental Statistics*, 14:281–300, 2009.

[8] R. Caruana. A non-parametric em-style algorithm for imputing missing values. In *Proceedings of Artificial Intelligence and Statistics.*

[9] N. Cressie. *Statistics for Spatial Data, Revised Edition.* Wiley, New York, 1993.

[10] N.L. Crookston and A.O. Finley. yaimpute: An r package for knn imputation. *Journal of Statistical Software*, 23(10):1–16, 2008.

[11] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 1–38, 1977.

[12] A.O. Finley and R.E. McRoberts. Efficient k nearest neighbor searches for multisource forest attribute mapping. *Remote Sensing of Environment*, 112(5):2203–2211, 2008.

[13] C. J. Geyer and E. A. Thompson. Constrained Monte Carlo maximum likelihood for dependent data (with discussion). *Journal of the Royal Statistical Society Series B*, 54:657–699, 1992.

[14] M. G. Gu and H. T. Zhu. Maximum likelihood estimation for spatial models by Markov chain Monte Carlo stochastic approximation. *Journal of the Royal Statistical Society Series B*, 63:339–355, 2001.

[15] S. Guiasu and A. Shenitzer. The principle of maximum entropy. *The mathematical intelligencer*, 7(1):42–48, 1985.

[16] M. L. Gumpertz, J. M. Graham, and J. B. Ristaino. Autologistic models of spatial pattern of phytophthora epidemic in bell pepper: effects of soil variables on disease presence. *Journal of Agricultural, Biological, and Environmental Statistics*, 2:131–156, 1997.

[17] X. Guyon. *Random Fields on a Network: Modeling, Statistics, and Applications.* Springer, New York, 1995.

[18] F. Huang and Y. Ogata. Generalized pseudo-likelihood estimates for Markov random fields on lattice. *Annals of Institutes of Statistical Mathematics*, 54:1–18, 2002.

[19] F. W. Huffer and H. Wu. Markov chain Monte Carlo for autologistic regression models with application to the distribution of plant speicies. *Biometrics*, 54:509–524, 1998.

[20] J. P. Huges, M. Haran, and P. C. Caragea. Autologistic models for binary data on a lattice. *Environmetrics*, 22:857–871, 2011.

[21] M. S. Kaiser and P. C. Caregea. Exploring dependence with data on spatial lattice. *Biometrics*, 65:857–865, 2009.

[22] K.Y. Kim, B.J. Kim, and G.S. Yi. Reuse of imputed data in microarray analysis increases imputation efficiency. *BMC bioinformatics*, 5(1):160, 2004.

[23] A. Kong, J.S. Liu, and W.H. Wong. Sequential imputations and bayesian missing data problems. *Journal of the American Statistical Association*, 89(425):278–288, 1994.

[24] P.-S. Lin. Estimating equations for separable spatial-temporal binary data. *Environmental and Ecological Statistics*, 17:543–557, 2010.

[25] P.-S. Lin, H.-Y. Lee, and M. Clayton. A comparison of efficiencies between quasi-likelihood and pseudo-likelihood estimates in non-separable spatial-temporal binary data. *Journal of Statistical Planning and Inference*, 139:3310–3318, 2009.

[26] R.J.A. Little and D.B. Rubin. *Statistical analysis with missing data*, volume 4. Wiley New York, 1987.

[27] J. Luengo, S. Garca, and F. Herrera. A study on the use of imputation methods for experimentation with radial basis function network classifiers handling missing attribute values: The good synergy between rbfns and eventcovering method. *Neural Networks*, 23(3):406–418, 2010.

[28] P. Meesad and K. Hengpraprohm. Combination of knn-based feature selection and knnbased missing-value imputation of microarray data. In *Innovative Computing Information and Control, 2008. ICICIC'08. 3rd International Conference on*, pages 341–341. IEEE.

[29] J. Møller. Perfect simulation of conditionally specified models. *Journal of the Royal Statistical Society, Series B*, 61:251–264, 1999.

[30] K. Pelckmans, J. De Brabanter, JAK Suykens, and B. De Moor. Handling missing values in support vector machine classifiers. *Neural Networks*, 18(5):684–692, 2005.

[31] J. G. Propp and D. B. Wilson. Exact sampling with coupled markov chains and applications to statistical mechanics. *Random Structures and Algorithms*, 9:223–252, 1996.

[32] C.C. Rodriguez. Entropic priors for discrete probabilistic networks and for mixtures of gaussians models. *arXiv preprint physics/0201016*, 2002.

[33] D.B. Rubin. Inference and missing data. *Biometrika*, 63:581–592, 1976.

[34] T. Schneider. Analysis of incomplete climate data: Estimation of mean values and covariance matrices and imputation of missing values. *Journal of Climate*, 14(5):853–871, 2001.

[35] R.L. Smith, S. Kolenikov, and L.H. Cox. Spatiotemporal modeling of pm2. 5 data with missing values. *Journal of Geophysical Research-Atmospheres*, 108(D24):9004, 2003.

[36] J. Uffink. Can the maximum entropy principle be explained as a consistency requirement? *Studies in History and Philosophy of Science Part B: Studies in History and Philosophy of Modern Physics*, 26(3):223–261, 1995.

[37] J. Uffink. The constraint rule of the maximum entropy principle. *Studies in History and Philosophy of Science Part B: Studies in History and Philosophy of Modern Physics*, 27(1):47–79, 1996.

[38] M. Weeks. Methods of imputation for missing data. Technical report, mimeo, Faculty of Economics and Politics, University of Cambridge.[Links], 2001.

[39] H. Wu and F. W. Huffer. Modeling the distribution of plant species using the autologistic regression model. *Environmental and Ecological Statistics*, 4:31–48, 1997.

[40] Y. Zheng and J. Zhu. Markov chain Monte Carlo for a spatial-temporal autologistic regression model. *Journal of Computational and Graphical Statistics*, 17:123–137, 2008.

[41] J. Zhu, H.-C. Huang, and C.-T. Wu. Modeling spatial-temporal binary data using Markov random fields. *Journal of Agricultural, Biological, and Environmental Statistics*, 10:212–225, 2005.

[1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12] [13] [15] [16] [17] [14] [18] [19] [20] [21] [22] [23] [24] [25] [26] [27] [28] [29] [30] [31] [32] [33] [34] [35] [36] [37] [38] [39] [40] [41]

Copyright© Zilong Wang, 2012.

Vita

Zilong Wang

University of Kentucky Department of Statistics

Date of Birth: Aug 12th, 1977

Place of Birth: Juan-Cheng, Shan-Dong Province, China

Education

Master of Science in Statistics, University of Illinois, 2008

Bachelor of Science in Statistics, University of Illinois, 2007

Employment

Research/Teaching Assistant	*Aug 2009 to Dec 2012*
Statistics Department, University of Kentucky	
Biostatistician	*May 2011 to Aug 2011*
Monsanto, St Louis, MO	
Research Assistant	*Aug 2007 to Jul 2008*
Statistics Department, University of Illinois	

Selected Publications

1. Wang, Z. and Zheng, Y. (2012). Analysis of Binary Data via a Centered Spatial-Temporal Autologistic Regression Model (CSTARM). *Environmental and Ecological Statistics*, In-Press

Copyright© Zilong Wang, 2012.