Estimation of Conditional Auto-regressive Models



Zhe Sha

Department of Statistics

University of Oxford

This dissertation is submitted for the degree of

Doctor of Philosophy

St Cross College October 2016

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements. This dissertation contains fewer than 65,000 words including appendices, bibliography, footnotes, tables and equations and has fewer than 150 figures.

Zhe Sha

October 2016

Acknowledgements

First and foremost, I would like to express my sincere gratitude to my supervisor, Prof. Brian D. Ripley. This thesis would not exist without his expert guidance and continuous support. Perhaps more important than completing the thesis, I have learnt from Brian the good qualities of a statistician and researcher, as well as some wisdom for life. I would also like to thank Dr Ruth Ripley for her insightful comments and kind patience when going through all the technical details and prose of this thesis.

Thanks to Dr Nick Golding, Prof. Andrew Hector and Matteo Tanadini for providing the British plant data set and the Sabah biodiversity experiment data set. Big thanks to Prof. Roger S. Bivand for his encouragement and help on using `R` and GIS software for analysing spatial data. Also thanks to my examiners, Prof. Geoff Nicholls and Dr. Duncan Lee for all the helpful advices provided during and after my VIVA.

I thank my dearest friends in Oxford and from home in China, without whom my life would certainly be much less lively and colourful. Special thanks go to Luke Kelly and Yunqi Wang who have always been so supportive for my work and brought me a lot of fun in exciting activities and delicious food.

Last but not least, I would like to thank my parents: I am proud and lucky to be your daughter. Wherever I go and no matter what I do, I know you are always there supporting me, praying for me and loving me; and I very much love you, too.

Abstract

Conditional auto-regressive (CAR) models are frequently used with spatial data. However, the likelihood of such a model is expensive to compute even for a moderately sized data set of around 1000 sites. For models involving latent variables, the likelihood is not usually available in closed form.

In this thesis we use a Monte Carlo approximation to the likelihood (extending the approach of Geyer and Thompson (1992)), and develop two strategies for maximising this. One strategy is to limit the step size by defining an experimental region using a Monte Carlo approximation to the variance of the estimates. The other is to use response surface methodology. The iterative procedures are fully automatic, with user-specified options to control the simulation and convergence criteria. Both strategies are implemented in our R package `mclcar`.

We demonstrate aspects of the algorithms on simulated data on a torus, and achieve similar results to others in a short computational time on two datasets from the literature. We then use the methods on a challenging problem concerning forest restoration with data from around 7000 trees arranged in transects within study plots. We modelled the growth rate of the trees by a linear mixed effects model with CAR spatial error and CAR random effects for study plots in an acceptable computational time.

Our proposed methods can be used for similar models to provide a clearly defined framework for maximising Monte Carlo approximations to likelihoods and reconstructing likelihood surfaces near the maximum.

Table of contents

List of figures **xiii**

List of tables **xvii**

1 Introduction **1**

1.1 Overview of the thesis 3

1.2 Conditional auto-regressive models 4

1.2.1 Conditional independence 4

1.2.2 Definition of CAR models 5

1.3 Parameter Estimation 9

1.3.1 Maximum likelihood estimation 10

1.3.2 Variations of the likelihood methods 11

1.3.3 Bayesian inference 13

2 Monte Carlo likelihood methods **15**

2.1 The Monte Carlo maximum likelihood estimator 15

2.2 Asymptotic normality of the MC-MLE 17

2.2.1 Convergence to the exact MLE 18

2.2.2 Convergence to the true value 19

2.3 Quantifying the variance 19

2.3.1 Finite variance 20

2.3.2 Variance estimator 22

2.3.3 Example: the CAR model on torus 23
2.4 Conclusion 34

3 Implementation of Monte Carlo likelihood methods 35
3.1 Choosing the importance sampler 35
3.1.1 Finding the initial value of ψ 36
3.1.2 Examples 37
3.2 Optimisation 43
3.2.1 Experimental region and updating $\psi^{(k)}$ 44
3.2.2 Convergence criteria 47
3.2.3 The likelihood function 52
3.3 Sampling algorithm and improvements 58
3.3.1 Sampling from the importance distribution 58
3.3.2 Variance reduction 63
3.3.3 Other optimisation algorithms 70
3.4 Conclusion 73

4 Response surface method 75
4.1 Methodology 76
4.2 Implementation 77
4.2.1 Design region and model fitting 79
4.2.2 Lack of fit and variance estimate 84
4.2.3 Profiling 86
4.2.4 Update and convergence criteria 87
4.3 Examples 88
4.3.1 Direct CAR model 89
4.3.2 GLM with latent CAR variables 97
4.4 Conclusion 101

5 Applications **103**

5.1 Scottish lip cancer 103

5.2 British native plant species 111

5.2.1 Data description 112

5.2.2 Model and estimation 115

5.3 Conclusion 119

6 The Sabah biodiversity experiment **121**

6.1 Data description 122

6.2 Defining the spatial structure 126

6.2.1 Parameter estimation 130

6.3 Estimation results 131

6.3.1 Analysis of separate plots 131

6.3.2 Mixed effect analysis 133

6.4 Conclusion 137

7 Discussion **141**

7.1 Major contributions 141

7.2 Problems and future improvements 143

References **145**

Appendix A Some details and examples of CAR models **151**

A.1 The unimodality of the likelihood for the direct CAR likelihood . . 151

A.2 Valid parameter space 157

A.2.1 Range of ρ 157

A.2.2 General constraints on parameters 157

A.3 Equivalent SAR and CAR models 158

A.3.1 MPLE and LSE 160

Appendix B Details for implementation **163**

B.1 The gradient and Hessian 163

Appendix C Manual for the package "mclcar" **165**

List of figures

1.1 Lattice and torus 6

2.1 Monte Carlo error v.s. Monte Carlo sample size 24

2.2 Monte Carlo error v.s. evaluation points 25

2.3 Monte Carlo error v.s. data size 27

2.4 Variance of the Monte Carlo variance estimates 28

2.5 Finite variance region of a direct CAR model on a 20×20 torus . . 29

2.6 Comparison of the exact and estimated Monte Carlo variance . . . 30

2.7 Comparison of the exact and Monte Carlo likelihood with variance estimates 31

2.8 Variability of the MC-MLE 33

3.1 Monte Carlo likelihood with MPLE as importance sampler parameter 38

3.2 Monte Carlo likelihood with different value of ρ for the importance sampler 40

3.3 Density plots of the MC-MLEs using different importance samplers 42

3.4 Examples of the experimental regions 46

3.5 Examples of the experimental regions with variance constraints . . . 47

3.6 Examples of the iterative optimization procedure 51

3.7 Likelihood information v.s. data size 58

3.8 Likelihood information v.s. number of trials in the binomial observations 59
3.9 Diagnostic plots for the adaptive MCMC algorithm 63

4.1 The RSM procedure 78
4.2 Examples of the Central Composite designs 79
4.3 Fitted RSM of different design regions 84
4.4 Monte Carlo variance estimates within the design regions 85
4.5 Evolutions of the response surface (direct CAR model) 91
4.6 Mean square error of the MC-MLEs 94
4.7 Mean square error of the MC-MLEs 95
4.8 Box plot of the Monte Carlo MLEs 96
4.9 Box plot of the exact MLEs 97
4.10 Evolutions of the response surfaces (latent CAR model) 99

5.1 Scottish lip cancer: SMR Map 107
5.2 British plants: Map of the presence of the first species 112
5.3 British plants: Images of the environmental variables 113
5.4 British plants: Map of species diversity 114
5.5 British plants: scatter plots 116
5.6 British plants: Contours of the fitted response surfaces in final iterations 118

6.1 SBE: Design map of the study plots 123
6.2 SBE: Map of the estimated random plot effects 125
6.3 SBE: Map of the estimated within-plot variances after adjusting for the within-plot spatial error 132
6.4 SBE: Map of the estimated spatial coefficients ρ 132
6.5 SBE: Map of the estimated CAR plot effects $\tilde{\boldsymbol{b}}$ 137

6.6 SBE: Map of the predicted growth rate of *Hopea sangal* 138

A.1 The shape of the likelihood function and its derivatives with different values of $\bar{\lambda}$ 156

A.2 CAR v.s. SAR: Spatial schemes on rectangular grids 159

List of tables

3.1 Results of the iterative optimisation procedure 50

4.1 Parameter limits of the design region 83
4.2 Parameter values and esitamtes 89
4.3 Parameter values for the simulation study 93

5.1 The Scottish lip cancer data 105
5.2 Estimation results of the Scottish lip cancer data 110
5.3 Estimation results of species diversity map 116

6.1 Biodiversity treatments 122
6.2 Estimation results of the mixed effect model for the Sabah biodiversity experiment data set 135
6.3 Estimation results of $\boldsymbol{\beta}$ 136

B.1 Gradient and Hessian for Approximation II & III 164

Chapter 1

Introduction

One major branch in spatial statistics is modelling the spatial variation of some variable of interest $\mathbf{Y}$ within a given region $\mathcal{D}$. Suppose observations of the variable, denoted by $y(s_i)$, are given for a fixed set of locations $\mathcal{S} = \{s_i : s_i \subset \mathcal{D}, i = 1, \ldots, N\}$, then the spatial variation can be modelled as either continuous or discrete according to the positional property of s_i. Continuous spatial variation is often used to describe the *point referenced data* where s_i represents a point that varies continuously over $\mathcal{S}$; while discrete spatial variation is used to model the *lattice data* or *areal data* where s_i corresponds to a point on a grid or an areal unit with well-defined boundaries.

Statistical inference of continuous spatial variation has its origins in the mining industry and was later developed into *geostatistics* (Mathéron, 1963; Diggle and Ribeiro, 2007). The goals are usually interpolation and prediction based on the stationary of the continuous stochastic spatial process. For the discrete spatial variation, inference focuses more on smoothing, and explaining the dispersion that is not adequately explained by the available covariates through modelling the spatial correlation. It has a wide application in image analysis (Besag et al., 1991), disease mapping (Clayton and Kaldor, 1987; Waller and Gotway, 2004; Lee and Mitchell, 2012), ecology (Banerjee et al., 2004; Fortin and Dale, 2005) and

econometrics (Anselin, 1988; Getis et al., 2004; Anselin, 2010) where the data are usually aggregated and only available at the resolution of a pre-defined lattice or areal units.

This thesis mainly discusses the modelling and estimation of the latter. In particularly, we are more interested in studying the spatial correlation than making predictions. *Conditional auto-regressive (CAR)* models (Whittle, 1954; Besag, 1974; Cressie, 1993) provide a simple and effective framework for modelling lattice and areal data that appear to have spatial interactions; however parameter estimation of such models is non-trivial. Bayesian inference is often used in practice as they are fast and easy to implement for commonly used models and priors; however, the methodology has its own limitation such as subjectivity of choosing the priors and tuning hyper parameters. For the frequentist approach, exact likelihood inference is only available for trivial cases; while there are several variations of likelihood based inference, the asymptotics results are usually not applicable for the CAR models for reasons discussed in Section 2.2.2.

In this thesis, we use a Monte Carlo approximation (Geyer and Thompson, 1992) to the likelihood of CAR models and develop two strategies for maximising this. One strategy is to limit the step size by defining an experimental region using a Monte Carlo approximation to the variance of the estimates. The other is to use response surface methodology. The strategies are iterative and fully automatic, with user-specified options to control the simulation and convergence criteria. Both appear to be stable and converge very fast to the optimal region of the likelihood function.

1.1 Overview of the thesis

In following sections of this chapter, we first provide the definition of CAR models and then review some existing estimation methods. The remainder of the thesis is organised as follows.

Chapter 2 describes Monte Carlo likelihood methods and related asymptotic results, and proposes estimators of the Monte Carlo variance for later use in finding the Monte Carlo maximum likelihood estimator. Then in Chapter 3 we discuss and resolve a few issues in the implementation and develop an automatic iterative procedure for directly maximising the Monte Carlo likelihood. In Chapter 4 we consider the situation where careful design of the evaluating scheme is needed due to limited computational budgets and introduce the response surface methodology (RSM) for locating and reconstructing the optimal region of the Monte Carlo likelihood.

In Chapter 5 our proposed procedures are applied on two data sets, one small and one large, modelled by generalised linear models with CAR latent variables. The small data set is frequently used in literature. We use it as an example for demonstrating the use of our `R` (R Core Team, 2016) package `mclcar` and then compare our results with those from existing estimations. With the large data set, we show that our proposed algorithms are able to provide maximum likelihood estimation within reasonable amount of time. Chapter 6 applies the algorithms on a new example with two layers of CAR structures, the CAR error and CAR random effects.

Finally we conclude in Chapter 7 with discussion on possible improvements and future developments.

Major contribution of this work include the followings. First, we have successfully implemented the Monte Carlo likelihood methods on CAR models which has not studied carefully in any existing literature. Then based on a simulation

study of our proposed estimators for the Monte Carlo variance, we are able to combine the variance estimator into the maximisation of the Monte Carlo likelihood and thus propose a fully automatic iterative procedure of the Maximum Monte Carlo likelihood estimation. We further propose to use the RSM method as an approximation to the Monte Carlo likelihood when the gradient of the function is not available or when the computational budget is limited. Finally, both these two procedures are implemented in our `R` package `mclcar`, which is available on **CRAN**(https://cran.r-project.org) for public use.

The Monte Carlo likelihood methods have been studied in the literature for more than twenty years but none of the existing methods have actually used the variance information in the Monte Carlo samples. An iterative procedure was recommended but has not been implemented in any fully automatic procedure. A direct maximization of the Monte Carlo likelihood can be numerically unstable or slow to converge when the parameter value is closes to its limits or a bad initial value is used. Our proposed RSM method provides stable and fast approximation when direct maximisation fails. Both the iterative procedure and RSM procedure provide general framework for maximising functions with Monte Carlo variance that is not homogeneous over the parameter space and therefore can be easily extended to problems other than estimating the parameters of the CAR models.

1.2 Conditional auto-regressive models

1.2.1 Conditional independence

We simplify the spatial index $s_i \in \mathcal{S}$ to be $i \in \mathcal{I}$ where $\mathcal{I} = \{1, 2, \ldots, N\}$. Denote by Y_i, $i \in \mathcal{I}$, the random variable at the spatial unit i which takes a value in $\mathcal{Y} \subset \mathbb{R}$. It is often easier to specify the conditional distribution of Y_i given the rest than to describe the joint distribution of $\mathbf{Y} = (Y_1, \ldots, Y_N)^T$. The *Hammersley-Clifford*

Theorem (Hammersley and Clifford, 1971) justifies the construction of the joint distribution through the *full conditionals* $\boldsymbol{\pi}(Y_i|\mathbf{Y}_{-i})$, $i = 1, \ldots, N$. We usually make the assumption of the Markov property that each y_i depends only on its neighbours $y_{j\sim i}$,

$$\boldsymbol{\pi}(Y_i|\mathbf{Y}_{-i}) = \boldsymbol{\pi}(Y_i|\mathbf{Y}_{j\sim i}) \tag{1.1}$$

where $j \sim i = \{j|j \neq i, j \text{ is a neighbour of } i\}$ is the neighbourhood set of location i. If we assume $\boldsymbol{\pi}(\mathbf{Y}) > 0$, $\forall \mathbf{Y} \in \mathcal{Y}^N$, the *Hammersley-Clifford Theorem* provides the necessary and sufficient conditions for the above full conditionals to define the Markov random field that equivalently represents the joint distribution of $\mathbf{Y}$. Based on this approach, Besag (1974) presents several such conditional specifications for models within the exponential families, including *auto-normal*, *auto-logistic* and *auto-Poisson* models and the corresponding constraints on parameters for each distribution.

1.2.2 Definition of CAR models

In spatial statistics, conditional auto-regressive models refer to a class of regression models with error terms or covariance structure of random effects specified as the following Gaussian full conditionals

$$y_i|y_{j\sim i} \sim \mathcal{N}(\mathbf{X}_i\beta + \sum_j \rho\, w_{ij}(y_j - \mathbf{X}_j\beta),\, \sigma_i^2) \tag{1.2}$$

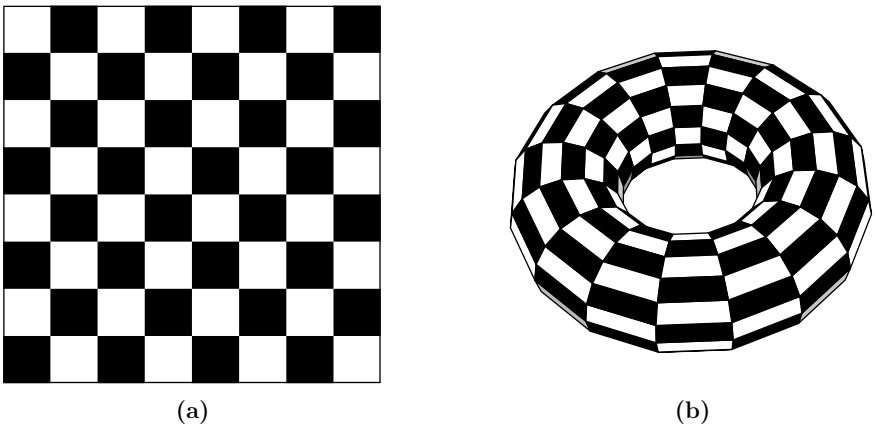
where $\mathbf{X}_j$ is the j^{th} row of the design matrix $\mathbf{X}$, $\boldsymbol{\beta}$ is a vector of the linear coefficients, ρ is the spatial coefficient showing the global strength of the spatial effect and $\{w_{ij}\}$ are the elements of the spatial weight matrix $\mathbf{W}$ reflecting the local spatial

effect. The joint distribution of $\mathbf{Y}$ is the multivariate Gaussian distribution:

$$\mathbf{Y} \sim \mathcal{N}(\mathbf{X}\boldsymbol{\beta}, \boldsymbol{\Sigma}) \tag{1.3}$$

where $\boldsymbol{\Sigma} = \mathbf{M}(\mathbf{I} - \rho\mathbf{W})^{-1}$ is the variance-covariance matrix, $\mathbf{M} = \text{diag}(\sigma_1^2, \ldots, \sigma_N^2)$ is a diagonal matrix with elements within $\text{diag}(\cdot)$ on its diagonal and $\mathbf{I}$ is an $N \times N$ identity matrix. The distribution is well defined if and only if $\boldsymbol{\Sigma}$ is symmetric and positive definite, which requires $w_{ij}/\sigma_i^2 = w_{ji}/\sigma_j^2$ and ρ within the range $(1/\lambda_1, 1/\lambda_N)$ where $\lambda_1 < \lambda_2 < \cdots < \lambda_N$ are the ordered eigenvalues of $\mathbf{W}$.

Finding the range of ρ requires computing the eigenvalues of $\mathbf{W}$. A special case is the CAR model with "rook-style" neighbourhood scheme on a torus as shown in Figure 1.1. The spatial weight matrix is symmetric and "block-circulant" as defined later in equation (1.11). The eigenvalues of such matrix can be proved to be within $(-4, 4)$ and thus the range of ρ is $(-0.25, 0.25)$. In practice, $\mathbf{W}$ is usually irregular; a simple and sufficient constraint is to require ρ to be within $(-1, 1)$ if $\mathbf{W}$ is row-standardised (proof see e.g., (Banerjee et al., 2004, Chap. 3)) and scale the marginal variances correspondingly to keep the covariance matrix symmetric.



(a) **(b)**

Fig. 1.1 The "rook-style" neighbourhood scheme on (a) an 8×8 lattice and (b) a 16×16 torus. Given a square, say in black, its "rook-style" neighbours are the white squares nearest to it as in the move of a rook in a chess game.

When ρ equals to either λ_1 or λ_N, we have the so-called *intrinsic* CAR model (Besag and Kooperberg, 1995), where the Gaussian distribution is improper as the covariance matrices becomes singular. General covariance matrix $\boldsymbol{\Sigma_\theta}$ with more parameters can be defined but the constraints on $\boldsymbol{\theta}$ are usually difficult to describe. Some practical results on the constraints can be found in Rue and Held (2005, Section 2.7) and details in the appendix A.2.

The spatial weight matrix is usually defined by prior knowledge of the spatial relationships of the study areas and it can be formulated from either a contiguity or connectivity binary matrix with

$$w_{ij} = \begin{cases} 1 & \text{if } i \neq j \text{ are neighbours or connected} \\ 0 & \text{otherwise} \end{cases} \tag{1.4}$$

or can be a general matrix based on some dissimilarity metrics such as distances, length of common boundaries, etc. Haining (2003, pp.78-85) provides a comprehensive list of commonly used systems of weights. There may be a few parameters in $\mathbf{W}$ to be estimated, as well as $\boldsymbol{\rho}$.

Latent CAR models

The conditional independence approach can be used directly to define the covariance structure of the observed variable $\mathbf{Y}$, but in practice it is often more natural to use a two-step model that assumes $\mathbf{Y}$ is driven by some latent CAR variable $\mathbf{Z} = \{z_i : i \in \mathcal{I}\}$. Suppose the latent variable $\mathbf{Z}$ follows a CAR model with some parameter $\boldsymbol{\theta}_1$ and the observed data $\mathbf{Y}$ is related to $\mathbf{Z}$ with distribution $\boldsymbol{\pi}(\mathbf{Y}|\mathbf{Z}, \boldsymbol{\theta_2})$. Denote all the parameters by $\boldsymbol{\theta} = (\boldsymbol{\theta}_1^T, \boldsymbol{\theta}_2^T)^T$. Given $\mathbf{Z}$ and $\boldsymbol{\theta}$, denote the likelihood by $\text{Lik}(\mathbf{Y}; \mathbf{Z}, \boldsymbol{\theta})$. If we assume some prior $\boldsymbol{\pi}(\boldsymbol{\theta})$ on the parameters, then the posterior

distribution can be represented as

$$\boldsymbol{\pi}(\mathbf{Z}, \boldsymbol{\theta}|\mathbf{Y}) \propto \text{Lik}(\mathbf{Y}; \mathbf{Z}, \boldsymbol{\theta})\boldsymbol{\pi}(\mathbf{Z}, \boldsymbol{\theta}) \propto \left(\prod_{i\in\mathcal{I}} \boldsymbol{\pi}(y_i|z_i, \boldsymbol{\theta})\right) \boldsymbol{\pi}(\mathbf{Z}|\boldsymbol{\theta})\boldsymbol{\pi}(\boldsymbol{\theta}) \tag{1.5}$$

For example, assume we have zero mean observations $\mathbf{y}$ in a linear regression model and we also know that the spatial correlations shown in the observations are actually from an unobserved effect $\mathbf{Z}$ that follows the CAR model, so the model can be written as

$$\mathbf{Y} = \mathbf{Z} + \boldsymbol{\epsilon}$$

where $\mathbf{Z} \sim \mathcal{N}(\mathbf{0}, (\mathbf{I} - \rho\mathbf{W})^{-1}\tau^2)$. Now if we assume the three parameters in the model have independent priors $\boldsymbol{\pi}(\rho, \sigma^2, \tau^2) = \text{U}(1/\lambda_1, 1/\lambda_n) \times \text{IG}(a_1, b_1) \times \text{IG}(a_2, b_2)$, where U stands for uniform distribution within the range of the parameter ρ given by the smallest and largest eigenvalues of $\mathbf{W}$ and IG stands for inverse Gaussian distribution with $a_i, b_i, i = 1, 2$ as the hyper-parameters. Then the posterior is

$$\begin{aligned}\boldsymbol{\pi}(\mathbf{Z}, \boldsymbol{\theta}|\mathbf{Y}) \propto \exp\left\{-\frac{(\mathbf{Y}-\mathbf{Z})^T(\mathbf{Y}-\mathbf{Z})}{2\sigma^2} - \frac{\mathbf{Z}^T(\mathbf{I}-\rho\mathbf{W})\mathbf{Z}}{2\tau^2}\right\} \\ \times \text{U}(1/\lambda_1, 1/\lambda_n) \times \text{IG}(a_1, b_1) \times \text{IG}(a_2, b_2)\end{aligned}$$

where the normalising constants is not in a closed form.

Our interest is usually the distribution of $\boldsymbol{\pi}(\mathbf{Z}|\mathbf{Y})$ and some marginals $\boldsymbol{\pi}(\boldsymbol{\theta}_j|\mathbf{Y})$. The latent CAR variables can be easily accommodated into a wide range of useful models, such as the generalised linear models, that have more flexibility than a direct conditional specification of the spatial relationship. For example, the auto-Poisson model requires the spatial coefficient $\rho \leq 0$ to have a valid joint distribution of $\mathbf{Y}$.

A subclass of CAR models

It is worth mentioning a subclass of CAR models, the *simultaneous auto-regressive (SAR)* models (Whittle, 1954)

$$y_i = \mathbf{X}_i\boldsymbol{\beta} + \sum_j \rho\, w_{ij}(y_j - \mathbf{X}_j\boldsymbol{\beta}) + \xi_i \tag{1.6}$$

where ξ_i are independent with normal distribution $\mathcal{N}(0, \tau_i^2)$ and other parameters are the same as those in a CAR model, except that to have a well defined joint distribution, $\mathbf{W}$ does not have to be symmetric as long as $\mathbf{I} - \rho\mathbf{W}$ is invertible. The joint distribution derived from (1.6) is

$$\mathbf{Y} \sim \mathcal{N}(\mathbf{X}\boldsymbol{\beta}, (\mathbf{I} - \rho\mathbf{W})^{-1}\boldsymbol{\Lambda}\left((\mathbf{I} - \rho\mathbf{W})^{-1}\right)^T) \tag{1.7}$$

where $\boldsymbol{\Lambda} = \text{diag}(\tau_1^2, \ldots, \tau_N^2)$ is a diagonal matrix.

The simultaneous specification in (1.6) is another way to define the spatial auto-regression and it has a similar form to the auto-regressive (AR) model in time series; however the spatial dependency defined by the SAR models are not directly analogous to the AR process in times series since ξ_i may still be dependent on the neighbours of y_i. More comparisons of the SAR and CAR models can be found in appendix A.3.

1.3 Parameter Estimation

In this section we discuss both the frequentist and Bayesian approaches to estimating the parameters in CAR models, address the difficulties therein and review some existing solutions.

1.3.1 Maximum likelihood estimation

The likelihood of the direct CAR models defined by (1.2) and (1.3) follows the density function of a multivariate normal distribution and, assuming $\sigma_i^2 = \sigma^2$ for all i, the likelihood function is

$$\ell(\boldsymbol{\theta}) \propto -\frac{N}{2}\log(\sigma^2) + \frac{1}{2}\log|\mathbf{I} - \rho\mathbf{W}| - \frac{(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})^T(\mathbf{I} - \rho\mathbf{W})(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})}{2\sigma^2} \tag{1.8}$$

where $\boldsymbol{\theta} = (\boldsymbol{\beta}^T, \sigma^2, \rho)^T$ are parameters to be estimated and $\mathbf{W}$ is a given spatial weight matrix. The evaluation of the determinant $|\mathbf{I} - \rho\mathbf{W}|$ of a $N \times N$ matrix usually has a complexity of $\mathcal{O}(N^3)$; hence as N increases, both evaluation and optimisation of the likelihood can be computationally expensive. Ord (1975) pointed out that the above determinant can be written as

$$|\mathbf{I} - \rho\mathbf{W}| = \prod_i^N (1 - \rho\lambda_i) \tag{1.9}$$

with λ_i being the eigenvalues of $\mathbf{W}$. Thus by supplying the eigenvalues, the maximum likelihood estimator (MLE) can by found by a line search on ρ with the profile log-likelihood of ρ

$$\ell_{\hat{\boldsymbol{\beta}},\hat{\sigma}^2}(\rho) = -N\log\hat{\sigma}^2 + \sum_i^N \log(1 - \rho\lambda_i) + \text{const} \tag{1.10}$$

where

$$\begin{cases} \hat{\boldsymbol{\beta}} = (\mathbf{X}^T(\mathbf{I} - \hat{\rho}\mathbf{W})\mathbf{X})^{-1}\mathbf{X}^T(\mathbf{I} - \hat{\rho}\mathbf{W})\mathbf{Y} \\ \hat{\sigma}^2 = \dfrac{1}{N}(\mathbf{Y} - \mathbf{X}\hat{\boldsymbol{\beta}})^T(\mathbf{I} - \hat{\rho}\mathbf{W})(\mathbf{Y} - \mathbf{X}\hat{\boldsymbol{\beta}}) \end{cases}$$

Now the computational cost depends on the complexity of finding the eigenvalues for $\mathbf{W}$, which in general case is also of $\mathcal{O}(N^3)$. For CAR models with "rook-style" neighbours on a $m \times n$ torus as shown by Figure 1.1 (b), the precision matrix is an

$mn \times mn$ block-circulant matrix with $m \times m$ blocks

$$\mathbf{C} = \begin{bmatrix} \mathbf{C}_0 & \mathbf{C}_{m-1} & \dots & \mathbf{C}_2 & \mathbf{C}_1 \\ \mathbf{C}_1 & \mathbf{C}_0 & \mathbf{C}_{m-1} & & \mathbf{C}_2 \\ \vdots & \mathbf{C}_1 & \mathbf{C}_0 & \ddots & \vdots \\ \mathbf{C}_{n-2} & & \ddots & \ddots & \mathbf{C}_{m-1} \\ \mathbf{C}_{m-1} & \mathbf{C}_{m-2} & \dots & \mathbf{C}_1 & \mathbf{C}_0 \end{bmatrix} \tag{1.11}$$

where each block $\mathbf{C}_i$ is an $n \times n$ circulant matrix with entries of the same arrangement as in (1.11). The eigenvalues of such matrices are in closed form and can be computed by discrete Fourier transformation with $\mathcal{O}(N \log N)$ flops. Some early work using the above property for approximating the determinant for spatial process is given by Whittle (1954) and details about using circulant matrix and Fourier transformation in CAR models can be found in Rue and Held (2005). For general weight schemes for CAR processes on discrete support, Rue and Held (2005) also proves that for large N the precision matrix can be permuted to have a form similar to the Toeplitz matrix, which then can be approximated by a block circulant matrix and therefore the above fast computation can be used.

1.3.2 Variations of the likelihood methods

The likelihood of the full data is usually difficult to evaluate, which has led to the development of alternative approaches such as replacing the likelihood by the conditional likelihood of a subset of the data. This was an early example of the *composite likelihood* Lindsay (1988), which is defined to be a weighted product of a collection of likelihoods associated with a set of marginal or conditional events. Varin et al. (2011) provides a recent overview of this topic. The "coding" scheme (Besag, 1974) and "pseudo-likelihood" (Besag, 1972) can both be considered as composite likelihood methods.

The "pseudo-likelihood" method uses the product of the conditional probabilities of each y_i

$$\mathcal{PL}(\mathbf{Y}; \sigma^2, \boldsymbol{\beta}) = \prod_i^N p_i(y_i; \sigma^2, \boldsymbol{\beta}) \tag{1.12}$$

where p_i are the full conditionals, i.e., the probability density functions of y_i given all its neighbours. Then, the maximum pseudo-likelihood estimator (MPLE) is found by iteratively solving the following system from some initial values

$$\begin{cases} \hat{\boldsymbol{\beta}} = (\mathbf{X}^T(\mathbf{I} - \hat{\rho}\mathbf{W})^2\mathbf{X})^{-1}\mathbf{X}^T(\mathbf{I} - \hat{\rho}\mathbf{W})^2\mathbf{Y} \\ \hat{\rho} = \dfrac{(\mathbf{Y} - \mathbf{X}\hat{\boldsymbol{\beta}})^T\mathbf{W}(\mathbf{Y} - \mathbf{X}\hat{\boldsymbol{\beta}})}{(\mathbf{Y} - \mathbf{X}\hat{\boldsymbol{\beta}})^T\mathbf{W}^2(\mathbf{Y} - \mathbf{X}\hat{\boldsymbol{\beta}})} = \dfrac{\boldsymbol{e}^T\mathbf{W}\boldsymbol{e}}{\boldsymbol{e}^T\mathbf{W}^2\boldsymbol{e}} \\ \hat{\sigma}^2 = \dfrac{1}{N}(\mathbf{Y} - \mathbf{X}\hat{\boldsymbol{\beta}})^T(\mathbf{I} - \hat{\rho}\mathbf{W})^2(\mathbf{Y} - \mathbf{X}\hat{\boldsymbol{\beta}}) \end{cases} \tag{1.13}$$

where we simplify the multiplication of a general square matrix $\boldsymbol{A}^T\boldsymbol{A}$ to be $\boldsymbol{A}^2$. The MPLE happens to be the same as the least squares estimators [1]. Similar methods for continous spatial processes can be found in Vecchia (1988) and Stein et al. (2004).

When the CAR variable is unobserved, the likelihood usually involves a high-dimensional integral without a closed form. There are two general approaches for likelihood-based estimation. The first approach is to approximate the integral in the likelihood function and then use the maximum of the approximated likelihood as the estimator. The integral can be approximated via either a deterministic method such as quadrature or the Laplace approximation (Evans and Swartz, 2000), or by Monte Carlo simulation (Ripley, 1987; Evans and Swartz, 2000; Geyer and Thompson, 1992). The second approach searches for an approximation to the MLE by iteratively solving estimation equations which are usually given by some

[1] More details on the least squares estimator and the different asymptotic properties for CAR and SAR models can be found in A.3.

variation on the likelihood function such as the hierarchical likelihood (h-likelihood, (Lee and Nelder, 1996)) and the penalised quasi-likelihood (PQL(Schall, 1991; Breslow and Clayton, 1993)).

1.3.3 Bayesian inference

Bayesian inference on the parameters are based on the posterior distribution $\boldsymbol{\pi}(\boldsymbol{\theta}|\mathbf{Y})$ and major difficulty often comes from the unknown normalising constant or posterior moments, which also reduces to the problem of approximating integrals. The approximation can be done either deterministically or by Monte Carlo simulation.

Among the deterministic methods, the Laplace approximation to the posterior marginals (Tierney and Kadane, 1986) has been widely used and was elaborated by Rue et al. (2009) to become *integrated nested Laplace approximations* (INLA). The idea is to approximate the posterior of the parameter and the latent variable by the Laplace approximation (Tierney and Kadane, 1986)

$$\tilde{\boldsymbol{\pi}}(\boldsymbol{\theta}|\mathbf{Y}) \propto \left.\frac{\boldsymbol{\pi}(\mathbf{Z},\boldsymbol{\theta}|\mathbf{Y})}{\tilde{\boldsymbol{\pi}}_G(\mathbf{Z}|\boldsymbol{\theta},\mathbf{Y})}\right|_{\mathbf{Z}=\mathbf{Z}^*(\boldsymbol{\theta})} \tag{1.14}$$

$$\tilde{\boldsymbol{\pi}}(z_i|\boldsymbol{\theta},\mathbf{Y}) \propto \left.\frac{\boldsymbol{\pi}(\mathbf{Z},\boldsymbol{\theta},\mathbf{Y})}{\tilde{\boldsymbol{\pi}}_{GG}(\mathbf{Z}_{-i}|z_i,\boldsymbol{\theta},\mathbf{Y})}\right|_{\mathbf{Z}_{-i}=\mathbf{Z}^*_{-i}(z_i,\boldsymbol{\theta})} \tag{1.15}$$

where $\tilde{\boldsymbol{\pi}}_G$ and $\tilde{\boldsymbol{\pi}}_{GG}$ denote the corresponding Gaussian distribution with the same modes and curvatures at the modes as the targets. The method is "nested" in the sense that in (1.15) instead of deriving the marginal distributions directly from $\tilde{\boldsymbol{\pi}}_G(\mathbf{Z}|\boldsymbol{\theta},\mathbf{Y})$, a new Gaussian distribution $\tilde{\boldsymbol{\pi}}_{GG}(\mathbf{Z}_{-i}|z_i,\boldsymbol{\theta},\mathbf{Y})$ is fitted for each $\mathbf{Z}_{-i}|z_i,\boldsymbol{\theta},\mathbf{Y}$. Applications to spatial modelling can be found in Lindgren and Rue (2015) and Blangiardo et al. (2013).

The posterior distribution can also be approximated by a family of distributions other than the Gaussian as in the variational Bayesian methods (Attias, 2000). The above can all produce fairly fast approximations to certain quantities with

respect to the posterior distributions but they only provide restricted information on the posterior.

A general solution is to approximate the posterior distributions by its samples, which usually requires the use of Markov Chain Monte Carlo (MCMC) algorithms (Ripley, 1987; Robert and Casella, 2005) and a recent application to the CAR models is given by Lee (2013). The posterior samples are often generated by the Gibbs Sampler with Metropolis steps and in order to achieve a faster mixing rate the algorithm updates in blocks of parameters and latent variables that are conditionally independent. Details of the implementation can be found in Chapter 3.

Approximate Bayesian Computation (ABC) methods (Rubin, 1984; Tavaré et al., 1997; Pritchard et al., 1999; Marin et al., 2012) provide another way to generate posterior samples. Although the ABC method can be less efficient than using MCMC algorithms, it provides approximate samples from both the prior and posterior and thus the likelihood distribution, therefore it can also be used for maximum likelihood estimation.

Chapter 2

Monte Carlo likelihood methods

The method of *Monte Carlo likelihood* was introduced in Geyer and Thompson (1992) and the name was first used in the discussion by Green (1992). The *Monte Carlo maximum likelihood estimator* (MC-MLE) is used to approximate the MLE for intractable likelihood functions, particularly when the likelihood function involves evaluating an integral not known in closed form. Here, we review the methods and some asymptotic results with illustrations from the CAR models using simulated data on the torus; the implementation of finding MC-MLE for the CAR models will be discussed in the next chapter.

2.1 The Monte Carlo maximum likelihood estimator

Consider a family of probability densities $\{f_{\boldsymbol{\theta}}\}$ with respect to some measure μ and assume the density indexed by the parameter $\boldsymbol{\theta}$, can be written as

$$L(\boldsymbol{\theta}; \mathbf{x}) = f_{\boldsymbol{\theta}}(\mathbf{x}) = \frac{1}{c(\boldsymbol{\theta})} h_{\boldsymbol{\theta}}(\mathbf{x}) \tag{2.1}$$

where $c(\boldsymbol{\theta}) = \int_{\mathcal{X}} h_{\boldsymbol{\theta}}(\mathbf{x})\,\mathrm{d}\mu(\mathbf{x})$ is the normalising constant. When $c(\boldsymbol{\theta})$ is unknown analytically, it can be approximated by a weighted average of s samples from an importance sampler distribution g,

$$c_s(\boldsymbol{\theta}) = \frac{1}{s}\sum_{i=1}^{s} \frac{h_{\boldsymbol{\theta}}(\mathbf{X}^{(i)})}{g(\mathbf{X}^{(i)})}, \;\; \mathbf{X}^{(i)} \sim g \tag{2.2}$$

where g can have any density function with respect to a dominating measure ν such that $\int_{\mathcal{A}} g\,\mathrm{d}\nu = 0$ implies $\int_{\mathcal{A}} \mathrm{d}\mu = 0$ for any measurable set $\mathcal{A} \subset \mathcal{X}$. If g is chosen to be $f_{\boldsymbol{\psi}}$ from the same family as $f_{\boldsymbol{\theta}}$, then we have the following importance sampling identity for the ratio of the normalising constants

$$\frac{c(\boldsymbol{\theta})}{c(\boldsymbol{\psi})} = \mathbb{E}_{\boldsymbol{\psi}}\left[\frac{h_{\boldsymbol{\theta}}(\mathbf{X})}{h_{\boldsymbol{\psi}}(\mathbf{X})}\right] = \int \frac{h_{\boldsymbol{\theta}}(\mathbf{x})}{h_{\boldsymbol{\psi}}(\mathbf{X})} f_{\boldsymbol{\psi}}(\mathbf{X})\,\mathrm{d}\mathbf{X} \tag{2.3}$$

Similarly, for the marginal density $f_{\boldsymbol{\theta}}(\mathbf{y})$ of $\mathbf{Y} \in \mathcal{Y}$ given by the joint density $f_{\boldsymbol{\theta}}(\mathbf{y}, \mathbf{z})$ of the random variables $(\mathbf{Y}, \mathbf{Z}) \in \mathcal{Y} \times \mathcal{Z}$, we have the following identity

$$\frac{f_{\boldsymbol{\theta}}(\mathbf{y})}{f_{\boldsymbol{\psi}}(\mathbf{y})} = \mathbb{E}_{\boldsymbol{\psi}}\left[\frac{f_{\boldsymbol{\theta}}(\mathbf{Y}, \mathbf{Z})}{f_{\boldsymbol{\psi}}(\mathbf{Y}, \mathbf{Z})}\middle| \mathbf{Y} = \mathbf{y}\right] = \int \frac{f_{\boldsymbol{\theta}}(\mathbf{y}, \mathbf{Z})}{f_{\boldsymbol{\psi}}(\mathbf{y}, \mathbf{Z})} f_{\boldsymbol{\psi}}(\mathbf{Z}|\mathbf{y})\,\mathrm{d}\mathbf{Z} \tag{2.4}$$

Now assume the data y_i, $i = 1, \ldots, N$ are from the probability model $f_{\boldsymbol{\theta}}$, we can use the identities to give approximations to the log-likelihood ratio $\ell_{\boldsymbol{\psi}}(\boldsymbol{\theta}) = \log \frac{L(\boldsymbol{\theta};\mathbf{y})}{L(\boldsymbol{\psi};\mathbf{y})}$. If the normalising constant is unknown, then we can use s_1 Monte Carlo samples from $h_{\boldsymbol{\psi}}$ to approximate $\ell_{\boldsymbol{\psi}}(\boldsymbol{\theta})$ by

$$\textbf{Approx. I}: \;\; \hat{\ell}^{s_1}_{\boldsymbol{\psi},N}(\boldsymbol{\theta}) = \log \frac{h_{\boldsymbol{\theta}}(\mathbf{y})}{h_{\boldsymbol{\psi}}(\mathbf{y})} - \log \frac{1}{s_1}\sum_{i=1}^{s_1} \frac{h_{\boldsymbol{\theta}}(\mathbf{Y}^{(i)})}{h_{\boldsymbol{\psi}}(\mathbf{Y}^{(i)})} \tag{2.5}$$

from the identity in (2.3). When there are unobserved variables $\mathbf{Z}$ in the model, the marginal log-likelihood ratio can be approximated by s_2 samples from $f_{\boldsymbol{\psi}}(\mathbf{Z}|\mathbf{Y} = \mathbf{y})$

as

$$\textbf{Approx. II :} \quad \hat{\ell}^{s_2}_{\boldsymbol{\psi},N}(\boldsymbol{\theta}) = \log \frac{1}{s_2} \sum_{i=1}^{s_2} \frac{f_{\boldsymbol{\theta}}(\mathbf{y}, \mathbf{Z}^{*(i)})}{f_{\boldsymbol{\psi}}(\mathbf{y}, \mathbf{Z}^{*(i)})} \tag{2.6}$$

using identity (2.4). If the extended likelihood $L(\boldsymbol{\theta}; \mathbf{y}, \mathbf{z})$ includes both the unknown normalising constant and the integral over latent variables, then the approximation becomes

$$\textbf{Approx. III :} \quad \hat{\ell}^{s_1,s_2}_{\boldsymbol{\psi},N}(\boldsymbol{\theta}) = \log \frac{1}{s_1} \sum_{i=1}^{s_1} \frac{h_{\boldsymbol{\theta}}(\mathbf{y}, \mathbf{Z}^{*(i)})}{h_{\boldsymbol{\psi}}(\mathbf{y}, \mathbf{Z}^{*(i)})} - \log \frac{1}{s_2} \sum_{j=1}^{s_2} \frac{h_{\boldsymbol{\theta}}(\mathbf{Y}^{(j)}, \mathbf{Z}^{(j)})}{h_{\boldsymbol{\psi}}(\mathbf{Y}^{(j)}, \mathbf{Z}^{(j)})} \tag{2.7}$$

where $\mathbf{Y}^{(j)}$ and $\mathbf{Z}^{(j)}$ are sampled from $f_{\boldsymbol{\psi}}(\mathbf{Y}, \mathbf{Z})$ to approximate the $c(\boldsymbol{\theta})$ in $f_{\boldsymbol{\theta}} = \frac{1}{c(\boldsymbol{\theta})} h_{\boldsymbol{\theta}}(\mathbf{Y}, \mathbf{Z})$.

Given a fixed value of $\boldsymbol{\psi}$, the log likelihood ratio can be used in place of the log likelihood since they differ by a constant that does not depend on the parameter $\boldsymbol{\theta}$. Therefore, the above approximations are all called the *Monte Carlo likelihood* (Green, 1992) and in the remainder we denote it by $\hat{\ell}^{s}_{\boldsymbol{\psi},N}$ or $\hat{\ell}_{\boldsymbol{\psi}}$. The Monte Carlo maximum likelihood estimator (MC-MLE) is defined to be the value of $\boldsymbol{\theta}$ corresponding to the maximum of $\hat{\ell}^{s}_{\boldsymbol{\psi},N}$

$$\hat{\boldsymbol{\theta}}^{s}_{\boldsymbol{\psi},N} = \arg\max_{\boldsymbol{\theta} \in \Theta} \hat{\ell}^{s}_{\boldsymbol{\psi},N}(\boldsymbol{\theta}) \tag{2.8}$$

2.2 Asymptotic normality of the MC-MLE

Convergence results can be found in Geyer (1994) and Cappé et al. (2002). Geyer (1994) proved that as the Monte Carlo sample size increases, the Monte Carlo likelihood converges to the exact likelihood with probability one under some mild continuity conditions on the density function, based on which, the Monte Carlo

MLE, profile likelihood and the confidence level sets also converge in probability to the corresponding exact values. Geyer (1994) also gives the convergence rate of the Monte Carlo MLE to the exact MLE $\hat{\boldsymbol{\theta}}$ while Cappé et al. (2002) discuss the rate of the MC-MLE converging to the true parameter value $\boldsymbol{\theta}^*$.

2.2.1 Convergence to the exact MLE

We now omit the sample size of the data, N, and the importance sampler value, $\boldsymbol{\psi}$, in the notation and use ℓ^s and $\hat{\boldsymbol{\theta}}^s$ to denote the Monte Carlo log-likelihood ratio and the MC-MLE. Under some regularity conditions described in Geyer (1994), we have the following result for the gradient of the Monte Carlo likelihood $\boldsymbol{\nabla}\hat{\boldsymbol{\ell}}^{\boldsymbol{s}}$ at the MLE $\hat{\boldsymbol{\theta}}$

$$N^{1/2}\boldsymbol{\nabla}\hat{\boldsymbol{\ell}}^{\boldsymbol{s}}(\hat{\boldsymbol{\theta}}) \xrightarrow{\mathcal{L}} \mathcal{N}(\mathbf{0}, \boldsymbol{A}) \tag{2.9}$$

for some covariance matrix $\boldsymbol{A}$ given by

$$\boldsymbol{A} = \frac{c(\boldsymbol{\psi})^2}{c(\boldsymbol{\theta})^2} \sum_{t=-\infty}^{+\infty} \boldsymbol{\gamma}(\boldsymbol{t}). \tag{2.10}$$

where $\boldsymbol{\gamma}(\boldsymbol{t}) = \mathrm{cov}(\mathbf{z}_{\boldsymbol{\theta}}(\mathbf{X}_0), \mathbf{z}_{\boldsymbol{\theta}}(\mathbf{X}_t))$ and $\mathbf{Z}_{\boldsymbol{\theta}}(\mathbf{X}) = \left[\frac{\nabla h_{\boldsymbol{\theta}}(\mathbf{x})}{h_{\boldsymbol{\theta}}(\mathbf{x})} - \frac{\nabla h_{\boldsymbol{\theta}}(\mathbf{X})}{h_{\boldsymbol{\theta}}(\mathbf{X})}\right] \frac{h_{\boldsymbol{\theta}}(\mathbf{X})}{h_{\boldsymbol{\psi}}(\mathbf{X})}$ is a functional of the importance samples. If we assume further that $\boldsymbol{B} = -\boldsymbol{\nabla^2}\boldsymbol{\ell}(\hat{\boldsymbol{\theta}})$ and $\boldsymbol{\nabla^3}\boldsymbol{\ell}^{\boldsymbol{s}}(\boldsymbol{\theta})$ is bounded in probability uniformly in a neighbourhood of $\hat{\boldsymbol{\theta}}$, then

$$-\boldsymbol{\nabla^2}\hat{\boldsymbol{\ell}}^{\boldsymbol{s}}(\hat{\boldsymbol{\theta}}^s) \xrightarrow{\mathcal{P}} \boldsymbol{B} \tag{2.11}$$

$$s^{1/2}(\hat{\boldsymbol{\theta}}^s - \hat{\boldsymbol{\theta}}) \xrightarrow{\mathcal{L}} \mathcal{N}(\mathbf{0}, \boldsymbol{B^{-1}AB^{-1}}). \tag{2.12}$$

This means that the MC-MLE $\hat{\boldsymbol{\theta}}^s$ converges to the MLE $\hat{\boldsymbol{\theta}}$ with probability one at the rate of $1/\sqrt{s}$, regardless the value of N and $\boldsymbol{\psi}$.

The above results can be used to assess the variability and approximate confidence regions of the MC-MLE, if the covariance matrix $V(\hat{\boldsymbol{\theta}}^s) = \boldsymbol{B^{-1}AB^{-1}}$ in (2.12) is given. Estimation of the variances will be discussed in Section 2.3, particularly for the CAR models on the torus.

2.2.2 Convergence to the true value

The asymptotic results are non-trivial to derive for spatial data as this involves considerations of how the data grow to infinity (whether to use infill asymptotics for a fixed domain or to assume an increasing domain?) and what conditions are at the boundary (whether the data are modelled as wrapped on a torus or embedded within a large enough grid?). Discussion on the boundary conditions and the corresponding asymptotic behaviour can be found in Ripley (1991, chap. 1) and Rue and Held (2005, chap. 5).

In this thesis and most of the time in practice, the CAR model is defined on a finite set of locations within a bounded area and therefore the asymptotic results may not apply, especially when the data size is small. We focus on comparing the inference based on Monte Carlo likelihood and the exact MLE, hence the $1/\sqrt{s}-$convergence to the exact MLE will be sufficient for us.

2.3 Quantifying the variance

The importance sampling estimator converges almost surely to the target by the Strong Law of Large Numbers, regardless of the choice of the importance distribution g as long as $\text{supp}(g) \supset \text{supp}(f)$ (see Evans and Swartz, 2000, p.166), which is apparently true in the setting of the Monte Carlo likelihood method; however the convergence rate depends on the Monte Carlo variance and it is desirable that the importance sampling estimator has a finite variance as small as possible. In most of the literature of importance sampling, a single target is given and the aim is

to find an appropriate importance sampler distribution. However, for the Monte Carlo likelihood, the importance sampler distribution is often chosen by the user at the beginning and the target is the likelihood ratio that varies with the parameter $\boldsymbol{\theta}$; therefore, it is crucial to decide for a given f_{ψ}, the range of $\boldsymbol{\theta}$ such that the likelihood ratio can be approximated with finite variance.

In the following, we first derive the finite variance region in Θ for a given importance sampler at parameter $\boldsymbol{\psi}$ for the CAR models and then provide two estimates of the variance. The variance estimates will be used in optimising the Monte Carlo likelihood. Then we show some properties of the variance estimators with examples of simulated CAR data on a torus.

2.3.1 Finite variance

In the setting of the Monte Carlo likelihood, we can write the likelihood ratio with a given importance sampler value $\boldsymbol{\psi}$ in the following generic form

$$\ell_{\boldsymbol{\psi}}(\boldsymbol{\theta}) = \log r_{\boldsymbol{\psi},\boldsymbol{\theta}}(\mathbf{x}) - \log \mathbb{E}_{\boldsymbol{\psi}}[r_{\boldsymbol{\psi},\boldsymbol{\theta}}(\mathbf{X})] \tag{2.13}$$

where $r_{\boldsymbol{\psi},\boldsymbol{\theta}}$ is the ratio of the un-normalised density functions, $\mathbf{x}$ is the observed data and $\mathbf{X}$ is a random variable from the importance distribution. Denote by $R(\boldsymbol{\psi},\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{\psi}}[r_{\boldsymbol{\psi},\boldsymbol{\theta}}(\mathbf{X})]$, the expectation on the right hand side of (2.13). The variance of the importance sampling approximation $\hat{R}(\boldsymbol{\psi},\boldsymbol{\theta})$ can be written as

$$\mathrm{Var}(\hat{R}) = \frac{1}{s}\mathbb{E}_{\boldsymbol{\psi}}\left[r_{\boldsymbol{\psi},\boldsymbol{\theta}}(\mathbf{Y})^2\right] - \frac{1}{s}R^2 \tag{2.14}$$

The variance is finite only if $\mathbb{E}_{\boldsymbol{\psi}}\left[r_{\boldsymbol{\psi},\boldsymbol{\theta}}(\mathbf{Y})^2\right] < \infty$.

Consider this variance for the CAR models. Without loss of generality, we assume the data has mean zero and thus $\boldsymbol{\theta}$ contains only parameters in the covariance matrix. This is justified by the fact that in practice the estimates of

the linear coefficients $\boldsymbol{\beta}$ do not change much for different covariance matrices and thus the data can always be scaled to have mean zero. We also only focus on Monte Carlo likelihood for the direct CAR model with **Approx. I** in equation (2.5). Denote by $\boldsymbol{\Sigma}_{\boldsymbol{\psi}}, \boldsymbol{\Sigma}_{\boldsymbol{\theta}}$ the covariance matrices of the importance sampler and target CAR models. We can write

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{\psi}}\left[r_{\boldsymbol{\psi},\boldsymbol{\theta}}(\mathbf{Y})^2\right] &= \int \left(\frac{q_{\boldsymbol{\theta}}(\mathbf{Y})}{q_{\boldsymbol{\psi}}(\mathbf{Y})}\right)^2 \frac{q_{\boldsymbol{\psi}}(\mathbf{Y})}{c(\boldsymbol{\psi})}\,\mathrm{d}\mathbf{Y} \\
&= (2\pi)^{-N/2}|\boldsymbol{\Sigma}_{\boldsymbol{\psi}}|^{-1/2} \int \exp\left\{-\mathbf{Y}^T\boldsymbol{\Sigma}_{\boldsymbol{\theta}}^{-1}\mathbf{Y} + \frac{\mathbf{Y}^T\boldsymbol{\Sigma}_{\boldsymbol{\psi}}^{-1}\mathbf{Y}}{2}\right\}\mathrm{d}\mathbf{Y} \\
&= |\boldsymbol{\Sigma}_{\boldsymbol{\psi}}^{-1}\boldsymbol{\Sigma}^*|^{1/2} \qquad (2.15)
\end{aligned}$$

where $\boldsymbol{\Sigma}^* = (2\boldsymbol{\Sigma}_{\boldsymbol{\theta}}^{-1} + \boldsymbol{\Sigma}_{\boldsymbol{\psi}}^{-1})^{-1}$. From this, we can also write the variance of the $\hat{R}(\boldsymbol{\psi}, \boldsymbol{\theta})$ in the form

$$\mathrm{Var}(\hat{R}) = \frac{1}{s}\left(|\boldsymbol{\Sigma}_{\boldsymbol{\psi}}^{-1}\boldsymbol{\Sigma}^*|^{1/2} - |\boldsymbol{\Sigma}_{\boldsymbol{\psi}}^{-1}\boldsymbol{\Sigma}_{\boldsymbol{\theta}}|\right) \qquad (2.16)$$

For the CAR covariance specified as $\boldsymbol{\Sigma} = \sigma^2(\mathbf{I} - \rho\mathbf{W})^{-1}$, equation (2.15) reduces to

$$\mathbb{E}_{\boldsymbol{\psi}}\left[r_{\boldsymbol{\psi},\boldsymbol{\theta}}(\mathbf{Y})^2\right] = |\boldsymbol{\Sigma}_{\boldsymbol{\psi}}^{-1}\boldsymbol{\Sigma}^*|^{1/2} = \left(\frac{\sigma^{*2}}{\sigma_{\boldsymbol{\psi}}^2}\right)^{N/2}\left|(\mathbf{I} - \rho_{\boldsymbol{\psi}}\mathbf{W})(\mathbf{I} - \rho^*\mathbf{W})^{-1}\right|^{1/2} \qquad (2.17)$$

where

$$\begin{cases} \sigma^{*2} = \dfrac{\sigma_{\boldsymbol{\theta}}^2\sigma_{\boldsymbol{\psi}}^2}{2\sigma_{\boldsymbol{\psi}}^2 - \sigma_{\boldsymbol{\theta}}^2} \\ \rho^* = \dfrac{2\rho_{\boldsymbol{\theta}}\sigma_{\boldsymbol{\psi}}^2 - \rho_{\boldsymbol{\psi}}\sigma_{\boldsymbol{\theta}}^2}{2\sigma_{\boldsymbol{\psi}}^2 - \sigma_{\boldsymbol{\theta}}^2} \end{cases} \qquad (2.18)$$

A sufficient (but not necessary) condition for the expectation to be finite is that $\mathbf{I} - \rho^*\mathbf{W}$ be positive definite. This requires ρ^* to be within $(1/\lambda_1, 1/\lambda_N)$ and $\sigma^{*2} > 0$, where λ_1 and λ_N are the smallest and largest eigenvalues of $\mathbf{W}$ as defined below equation (1.3). Hence, we get the following finite-variance region constraints

for $\boldsymbol{\theta}$

$$\begin{cases} 0 < \dfrac{\sigma^2_{\boldsymbol{\theta}}}{\sigma^2_{\boldsymbol{\psi}}} < 2; \\ 1/\lambda_1 < \dfrac{2\rho_{\boldsymbol{\theta}}\sigma^2_{\boldsymbol{\psi}} - \rho_{\boldsymbol{\psi}}\sigma^2_{\boldsymbol{\theta}}}{2\sigma^2_{\boldsymbol{\psi}} - \sigma^2_{\boldsymbol{\theta}}} < 1/\lambda_N \end{cases} \tag{2.19}$$

For the generalised linear model with CAR latent variables, the rule is similar when using **Approx. II** in (2.6) for approximating the marginal likelihood.

2.3.2 Variance estimator

The Monte Carlo variance is not usually available in closed form. One way to derive an estimator is by using the central limit theorem. Assume that $g_{\boldsymbol{\psi},\boldsymbol{\theta}}$ is continuous and that $\nabla g_{\boldsymbol{\psi},\boldsymbol{\theta}}(\mu)$ exists and is finite, and under some regularity conditions, then we can use the continuous mapping theorem to show that

$$\sqrt{s}(\hat{R}(\boldsymbol{\theta},\boldsymbol{\psi}) - R(\boldsymbol{\theta},\boldsymbol{\psi})) \xrightarrow{d} \mathcal{N}(0, \mathrm{Var}(\hat{R})) \tag{2.20}$$

Then by applying the delta method we have

$$\sqrt{s}(\hat{\ell}_{\boldsymbol{\psi}}(\boldsymbol{\theta}) - \ell_{\boldsymbol{\psi}}(\boldsymbol{\theta})) \xrightarrow{d} \mathcal{N}\left(0, \frac{\mathrm{Var}(\hat{R})}{R^2(\boldsymbol{\psi},\boldsymbol{\theta})}\right) \tag{2.21}$$

Both $\mathrm{Var}(\hat{R})$ and R are unknown but we can usually find some estimates for them. For R we can use the importance sampling approximation $\hat{R}$, whereas for $\mathrm{Var}(\hat{R})$, there are two obvious ways: the plug-in estimate $\hat{V}$ and the Monte Carlo estimate $\tilde{V}_{\mathtt{mc}}$. The plug-in estimate uses the sample variance of the Monte Carlo samples or a functional of the Monte Carlo samples to approximate $\hat{R}$, while the Monte Carlo estimate uses the sample variance of M replications of $\hat{R}$. Similarly the asymptotic variances for the gradient of the Monte Carlo likelihood in expression (2.9) and the MC-MLE in expression (2.12) can be estimated in either of the two ways.

The plug-in estimator may be less reliable than the Monte Carlo one as it uses only one realization of the importance sampler approximation but it is much cheaper to compute and can be easily combined into the likelihood evaluation as a crude estimate of the variance. Hence in the following, we will always use the plug-in estimator for the variance if not stated otherwise.

2.3.3 Example: the CAR model on torus

In this example, we study the variance estimators using simulated data on toruses. We first illustrate some basic properties of the variance of the Monte Carlo likelihood by using Monte Carlo replications; then we plot the finite variance region and compare the exact asymptotic variance with the plug-in estimator, finally we study the variance estimate of the MC-MLE and the corresponding approximated confidence intervals.

In the following simulation studies, we focus on data generated on a $n \times n$ torus with "rook-style" neighbourhood scheme for three reasons: firstly, such specification avoids possible model mis-specification and estimation bias caused by the boundary conditions; secondly, exact likelihood for such model is available and fast to compute and therefore can be used for comparison with the Monte Carlo likelihood method; finally, the simple structure of the precision matrix is convenient for analysing the Monte Carlo likelihood function and the Monte Carlo variance (see e.g., section 3.2.3 and section 4.2).

Variability of the Monte Carlo likelihood

In the following we show how the Monte Carlo variance varies according to the data size, the Monte Carlo sample size and the distance of the evaluating point to the importance sampler value. Here data were simulated on toruses using zero mean and CAR covariance with $\rho = 0.1, \sigma^2 = 1$, and the weight matrix was set to

be the binary "rook-style" adjacency matrix. The number of Monte Carlo samples and the importance sampler value vary according to the different purposes in the study. The Monte Carlo likelihood estimates are replicated 100 times in all cases to produce $\tilde{V}_{\mathtt{mc}}$.

Monte Carlo sample size From previous analysis we know that the Monte Carlo error scales at the rate of $1/\sqrt{s}$ where s is the size of the Monte Carlo samples. Figure 2.1 shows the Monte Carlo error for an evaluation at a chosen point with Monte Carlo sample size increasing from 10 to 10^6. The data is simulated on a 20×20 torus and the point chosen to be evaluated is $\hat{\rho}_p - 0.05$ where $\hat{\rho}_p$ is the MPLE. The figure is on log-log scale and the fitted line has a slope about -0.5.

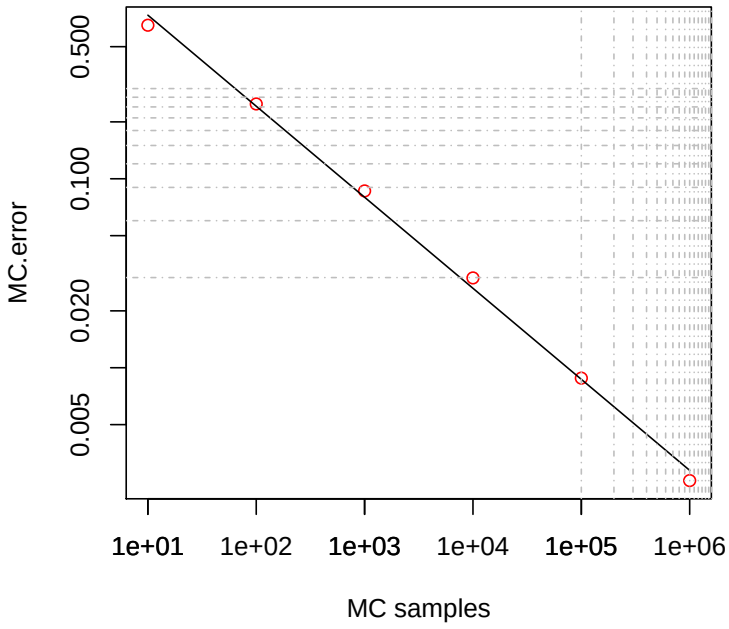


Fig. 2.1 Monte Carlo error v.s. Monte Carlo sample size plotted on log-log scale. The Monte Carlo variance is estimated from 100 replications of the Monte Carlo likelihood.

Next we show in Figure 2.2 how the Monte Carlo variance depends on the distance between the evaluation point and the importance sampling parameter

value. The Monte Carlo variance is zero at the importance sampler value and as the evaluation point moves away, the variance increases. This is most obvious from the bottom panel of Figure 2.2 when the number of Monte Carlo samples is the smallest. This means that when evaluating for parameter values near that of the importance sampler, we can use fewer samples in the Monte Carlo likelihood than for those far away.

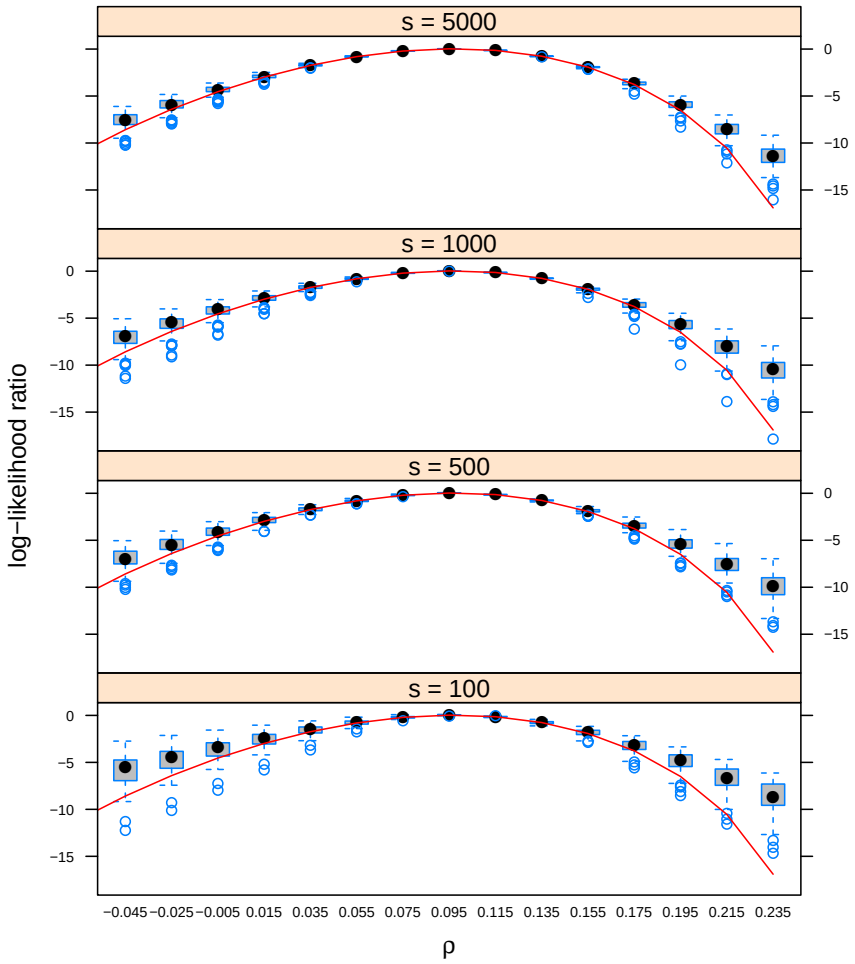


Fig. 2.2 Box plots of Monte Carlo likelihood for different evaluation points and increasing Monte Carlo sample size. The box plot is produced by 100 replications of the Monte Carlo likelihood at each evaluation point and the red lines imposed on the box plots are the exact log-likelihood ratio.

Data size Now we study how the number of observations may affect the Monte Carlo variance. In Figure 2.3 we plot the Monte Carlo likelihood for data on an $n \times n$ torus for $n = 10, 20$ and 30. The Monte Carlo likelihood is evaluated for σ^2 fixed at its MPLE value, and a set of 25 values of ρ equally spaced within its range, 100 times.

The red solid lines in the plots are the smoothed mean value of the 100 replications and the red and blue dashed lines are the smoothed curves of the lower and upper 5% quantiles of the replications. We also plot the exact value by the thick solid green lines in each plot. The left panels show the plots for the entire range of ρ while the right panels shows the corresponding zoom-in part around the maximum where the range of the log-likelihood ratio is set to be $(-2, 0)$ and the range of ρ is set to be $(-0.05, 0.2)$.

As the torus size increases, the Monte Carlo error becomes larger for the same ρ. This can be most obviously seen from the left panels where the left end of the curves seem to diverge with a similar amount in the plots but the scales of the plots are vastly different. However in practice this may not be a severe problem if we are only interested in a small region around the maximum. The zoom-in plots on the right panel show that the likelihood become more concentrated around the maximum for larger toruses. Hence as the size of the data set increases, the likelihood usually become more informative which reduces the size of the region of interest and we do not need to spend too much effort on exploring regions where the Monte Carlo variance may explode.

Variability of $\hat{V}$

The previous examples all used Monte Carlo replications to assess the variability of the Monte Carlo likelihood, however this can be expensive to compute. In practice, we would like to use the plug-in estimate, $\hat{V}$, as a cheap approximation and we study the variability of $\hat{V}$ by comparing it with $\tilde{V}_{\texttt{mc}}$ in this example. Here we use

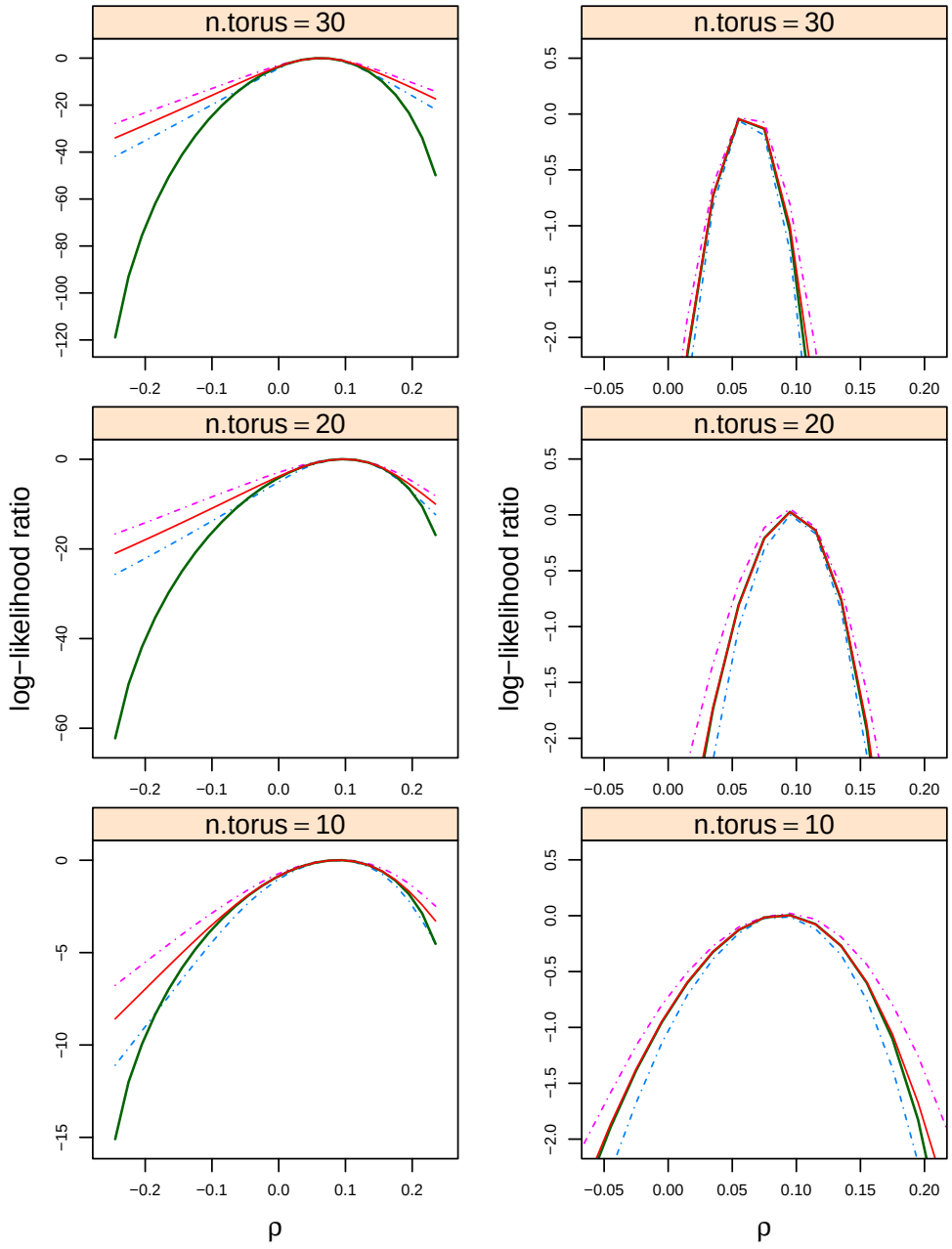


Fig. 2.3 Monte Carlo and exact likelihood ratio data simulated on increasing torus size. Plots on left panels show the curve for the entire parameter space and the right panel show the zoom-in around the maxima of the left. Details can be found in text.

simulated data on a 20×20 torus, with the same parameter settings as before. The importance sampler value is again set to the MPLE and the Monte Carlo sample size is fixed to be 500.

Figure 2.4 shows the box plots of 100 evaluations of $\hat{V}(\rho)$ at the same set of ρ values used in Figure 2.3. The value of $\hat{V}$ is centred by $\tilde{V}_{\texttt{mc}}(\rho)$. In the figure, the importance sampler value of ρ is shown by the red dashed line and the red solid line shows the exact MLE. The Monte Carlo variance estimate $\tilde{V}_{\texttt{mc}}$ is an unbiased estimate of the Monte Carlo variance and is usually more reliable than the plug-in estimate; therefore $\hat{V}(\rho)$ centred around $\tilde{V}_{\texttt{mc}}(\rho)$ with small variability are considered as good estimates.

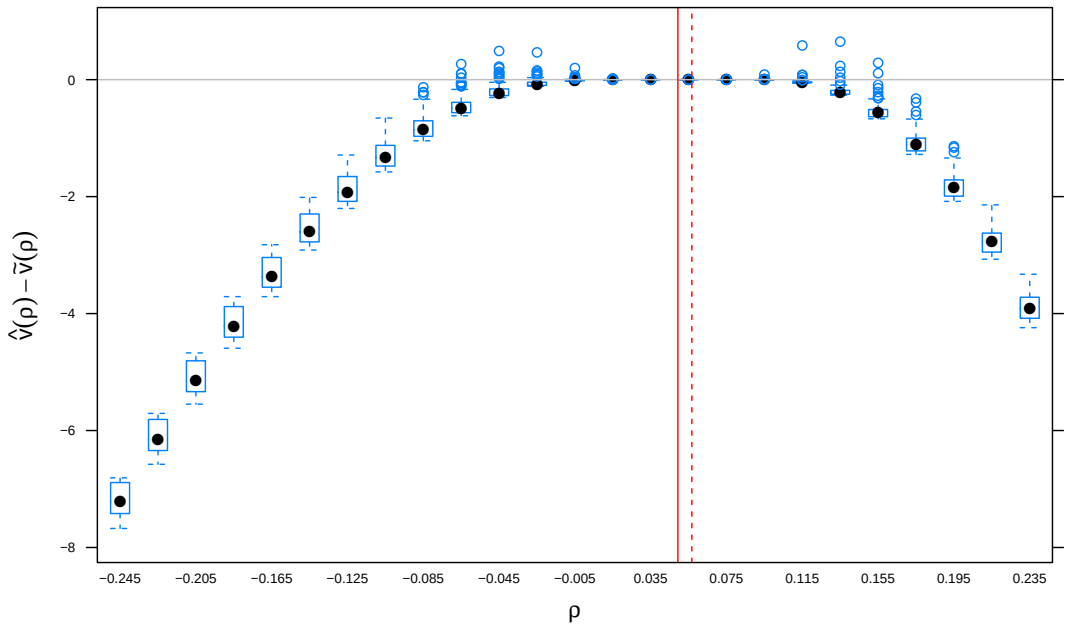


Fig. 2.4 Box plots $\hat{V}(\rho)$ centred at $\tilde{V}_{\texttt{mc}}(\rho)$. The solid red line shows the value of ρ at the exact MLE and the dashed red line is the MPLE used as ρ_ψ. $\hat{V}$ is less biased with small variability for ρ near ρ_ψ but it becomes much more variable for values far away and tends to under-estimate the true Monte Carlo variance.

As expected, the figure conveys two important properties of $\hat{V}$: the variance tends to be under-estimated as the evaluation point moves away from the importance sampler value and the variability of $\hat{V}$ increases substantially at the same time.

For points near the importance sampler value, $\hat{V}$ is almost identical to $\tilde{V}_{\text{mc}}$ with a very small variance. Therefore, we may conclude that $\hat{V}$ can be used as a reliable estimate of the Monte Carlo variance for a small region around the importance sampler value but otherwise its value can be misleading.

Finite variance region and the use of $\hat{V}$

We summarise a few rules of thumb in using $\hat{V}$ by comparing it with the asymptotic variance for direct CAR data on a torus. Figure 2.5 illustrates the finite variance region given by the constraints in expression (2.19) for the same setting as the previous example. The finite variance region is within the red triangle.

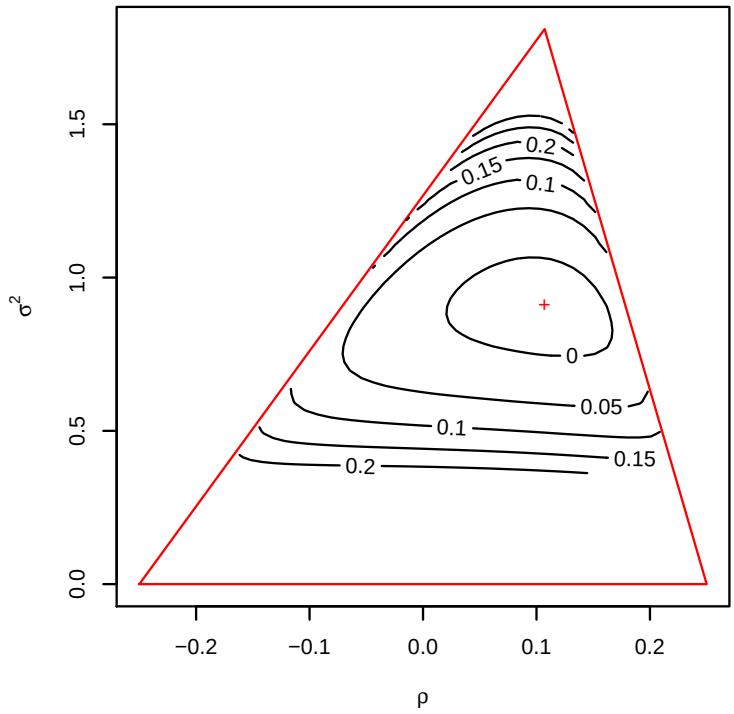


Fig. 2.5 Finite variance region for a direct CAR model on a 20×20 torus with "rook-style" spatial weight matrix. The importance sample value $\boldsymbol{\psi}$ is the MPLE. The finite variance region is defined by the red triangle and the contour lines are the asymptotic variance on the log scale using $s = 500$.

In Figure 2.5 we also plot the contours of $V(\hat{\ell})$, which is the asymptotic variance of the Monte Carlo likelihood derived in formula (2.21). The asymptotic variance can be written in the following closed form for the direct CAR model on the torus

$$V(\hat{\ell}) = \frac{V(\hat{R})}{R^2(\boldsymbol{\psi}, \boldsymbol{\theta})} = \frac{1}{s}\left(\frac{|\boldsymbol{\Sigma}_{\boldsymbol{\psi}}\boldsymbol{\Sigma}^*|^{1/2}}{|\boldsymbol{\Sigma}_{\boldsymbol{\theta}}|} - 1\right) = \frac{1}{s}\left[\left(\frac{\sigma_{\boldsymbol{\psi}}\sigma^*}{\sigma^2_{\boldsymbol{\theta}}}\right)^N \left(\prod_{i=1}^{N} \frac{\sqrt{(1-\rho_{\boldsymbol{\psi}}\lambda_i)(1-\rho^*\lambda_i)}}{1-\rho_{\boldsymbol{\theta}}\lambda_i}\right) - 1\right]$$

with $\boldsymbol{\Sigma}^*, \sigma^{*2}, \rho^*$ defined as in equations (2.15) and (2.18). The variance increases roughly at exponential speed in the number of sites N and this can be seen from the evenly spaced contours in Figure 2.5 where the asymptotic variance is log-transformed and scaled by N.

Now we compare $\hat{V}$ with the asymptotic variance. In Figure 2.6 the contour lines of $\hat{V}$ are superimposed on the image of the asymptotic variance with the same setting as in the previous example.

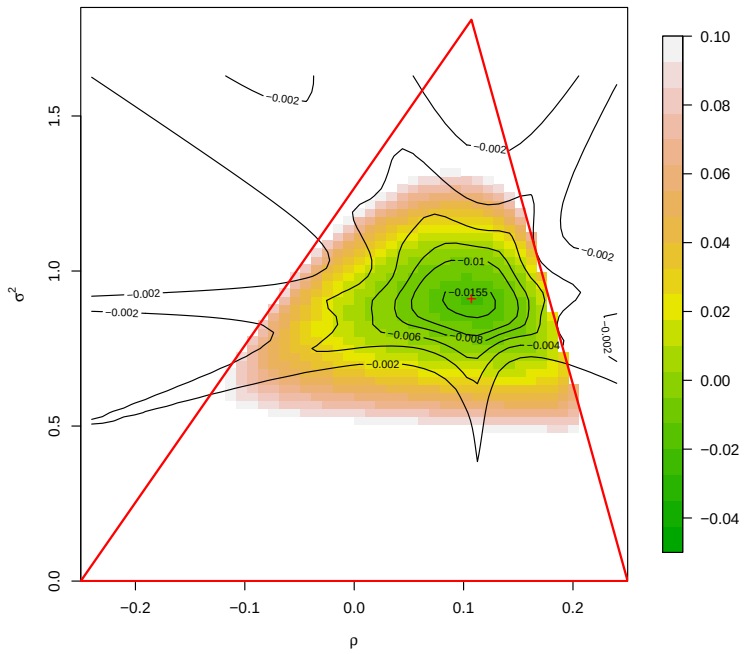


Fig. 2.6 Comparison of the exact and estimated variance using the same data in Figure 2.5. Contours of $\hat{V}$ are overlaid on the level plot of the asymptotic variance.

From the plot, the variance estimate is very close to exact within the level $\exp(-0.0155 \times 400) = \exp(-6) \approx 0.002$ and beyond that level the estimate became smaller than the exact value. Meanwhile, the Monte Carlo error should scale as $1/s$ and in this example we used $s = 500$; this gives $1/500 \approx 0.002$ which is close to the contour level of $\exp(-6)$. Hence we may conclude that in general the variance estimate is accurate within $1/s$ and beyond this level, it more and more under-estimates the true variability. One useful rule for showing the accuracy of the variance estimates is to plot the growth rate of variance on a given path and check when it stops growing exponentially fast; this rule is used in the steepest-ascent analysis in Chapter 4.

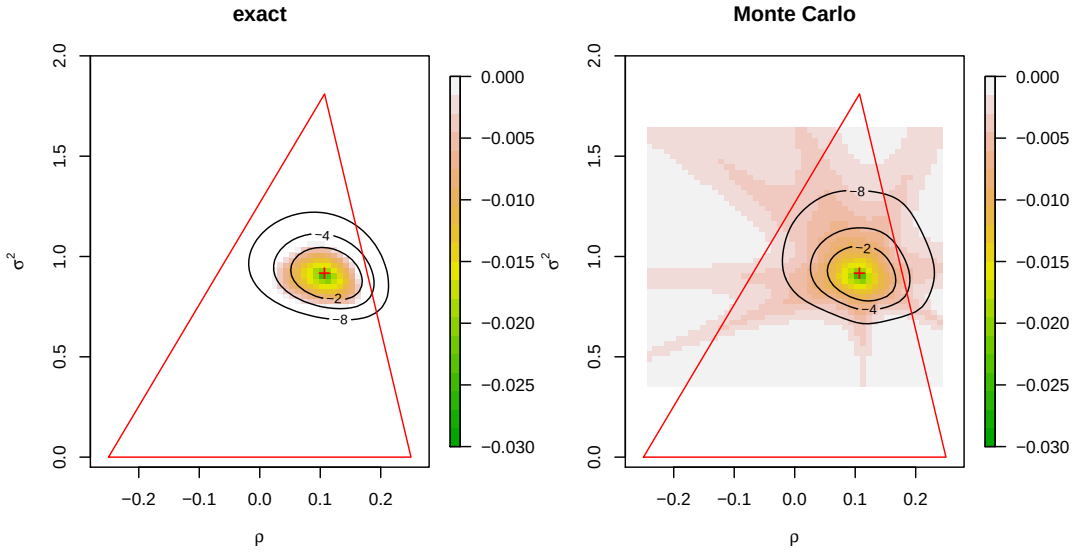


Fig. 2.7 Comparison of the exact and estimated values using the data from Figure 2.5. Contours of the log-likelihood ratios are overlaid on the level plot of the variance in each plot.

Next we add the information given by the Monte Carlo likelihood to the variance estimate. Figure 2.7 includes the contour plots of the exact and Monte Carlo likelihood, superimposed on the images of the asymptotic variance and of $\hat{V}$ correspondingly. Again the variance is scaled in the same way as the previous

examples. The contours of the exact and Monte Carlo log-likelihood look very similar within the level set of -4. The variances also tend to follow the pattern of the likelihood surface especially around the level set of -2. From the Monte Carlo estimate on the right panel, $\hat{V}$ substantially under-estimates the exact variance beyond the level set -4. This critical level can be used as a guideline for the reliability of the Monte Carlo likelihood. One explanation is that the 95% level set for $\boldsymbol{\psi}$ based on a likelihood ratio test with two degrees of freedom is all the $\boldsymbol{\theta}$ such that $\ell(\boldsymbol{\theta}) - \ell(\boldsymbol{\psi}) < 3$ and this is well within the accurate region of the Monte Carlo approximation shown by the figures.

Variance estimates of the MC-MLE

Finally, we study the variability of the MC-MLE by producing the confidence interval of the MLE based on the asymptotic distribution in (2.12). In Figure 2.8, we plot the 100 replications of the 95% confidence interval using $\hat{V}$ and compare it with the interval given by $\tilde{V}_{\texttt{mc}}$.

For each replication, the MC-MLE is obtained by directly optimising the Monte Carlo likelihood using $\boldsymbol{\psi}$ equal to the MPLE. Such MC-MLEs are replicated 100 times then plotted in ascending order with the corresponding 95% CI using $\hat{V}$. The exact MLE is drawn as the red solid line and the 95% confidence interval produced by the exact MLE is shown as the shaded band within the red dashed lines. The MC-MLEs are apparently centred around the exact value. The confidence intervals produced by MC-MLEs and $\hat{V}$ show some variability but there is no obvious pattern except some very wide CIs appears when the MC-MLE are biased below. In the plot, we also show the confidence interval given by $\tilde{V}_{\texttt{mc}}$ as the blue dashed lines and this is very close to the exact values.

In the 100 replications of the Monte Carlo MLE and the corresponding 95% confidence intervals, there are 90 of the intervals covering the exact MLE, so the coverage of the intervals are very close to the expected value and therefore we

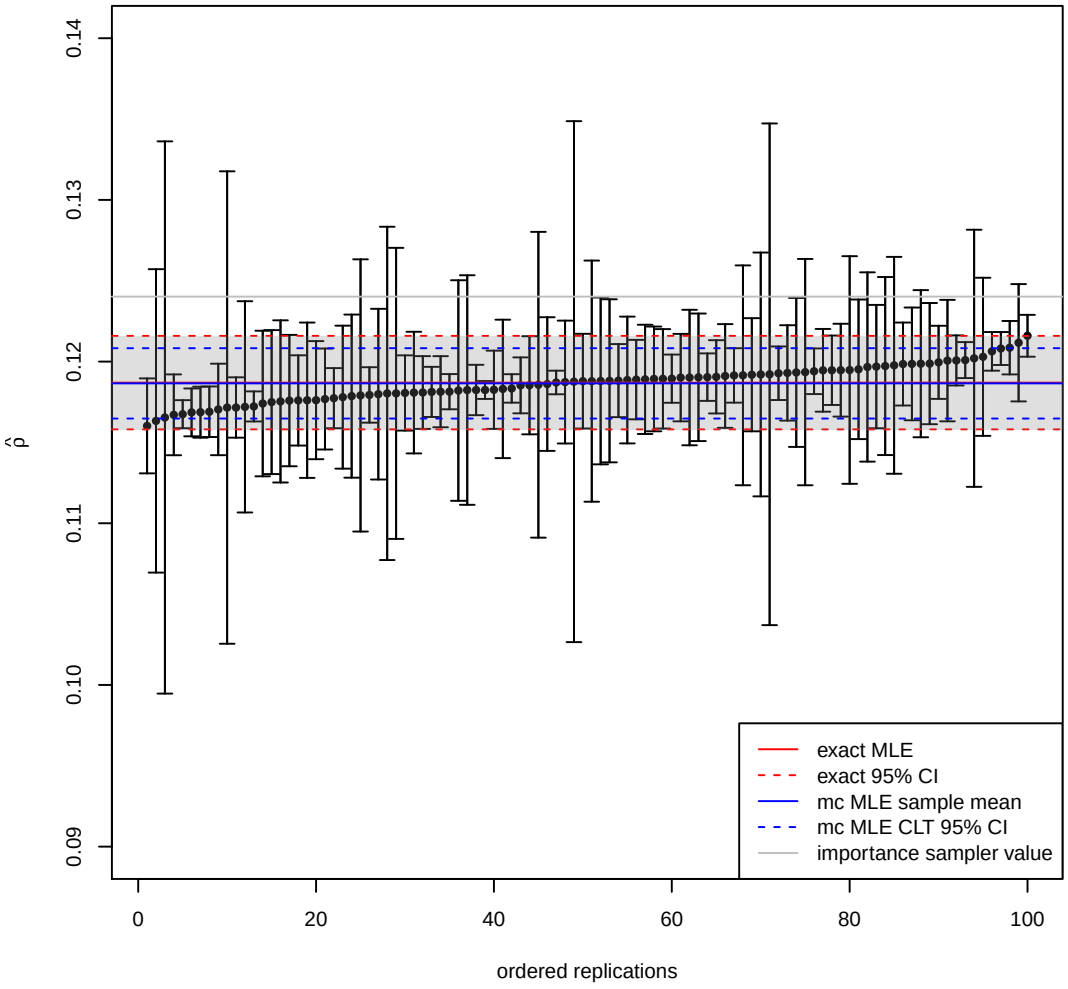


Fig. 2.8 Comparing the confidence intervals of MLE given by $\hat{V}$, $\tilde{V}_{\mathtt{mc}}$ and the asymptotic variance. Details are in the text.

can use these approximated confidence intervals to describe the variability of the Monte Carlo MLE. For those intervals that do not cover the exact MLE, the Monte Carlo MLEs are either far away from the importance sampler value and thus underestimated both the exact MLE and the Monte Carlo variance, or they are too close to the importance sampler with small variance estimates and overestimated MLEs.

2.4 Conclusion

In this section, we have reviewed the Monte Carlo likelihood method and the corresponding asymptotic results in the literature. Based on these results, we studied the finite variance region for the CAR models on the torus which can be useful for controlling the step size in later optimisation procedures.

We have also proposed two variance estimates, the plug-in variance estimator $\hat{V}$ and the Monte Carlo estimator $\tilde{V}_{\mathrm{mc}}$, and conducted simulation experiments in studying the property of these two estimates. Our simulation results show that in general the plug-in variance estimator $\hat{V}$ provides a good approximation to the Monte Carlo variance for both the Monte Carlo likelihood and the MC-MLE when using a importance sampler value that is close to the target. This can be useful in monitoring the accuracy of the Monte Carlo approximation and the convergence of the maximisation procedure in later chapters.

Chapter 3

Implementation of Monte Carlo likelihood methods

Elegant as the idea of Monte Carlo likelihood may seem, there are three significant issues in its implementation. First of all we need to choose an appropriate importance sampler for approximating the likelihood; then we will need to be able to generate samples from the importance distribution and finally, since the function is stochastic, we may need to iterate the procedure until the MC-MLE converges according to some criteria. In this chapter we discuss these problems and provide some useful rules for later use.

3.1 Choosing the importance sampler

In its original form, the Monte Carlo likelihood used an importance sampler with the same parametric form as the likelihood function and thus, the approximation is to the log-likelihood ratio between two different parameter values $\boldsymbol{\psi}$ and $\boldsymbol{\theta}$. This has the advantage that the target can be used in place of the log-likelihood since they only differ by a constant that depends on $\boldsymbol{\psi}$ and that the Monte Carlo variance is relatively easier to analyse by monitoring the distance between $\boldsymbol{\theta}$ and $\boldsymbol{\psi}$. The

Monte Carlo variance is minimised when $\boldsymbol{\psi}$ is chosen to be the MLE $\hat{\boldsymbol{\theta}}$ but this is contradictory to our purpose of finding the MLE. Therefore, we wish to find a $\boldsymbol{\psi}$ that may be close to the MLE as a starting point and this can usually be found by fitting a simple model to the data or use some other estimator for the parameter. Instead of using the likelihood distribution as the importance sampler, some other sampling methods (Geyer and Møller, 1994; Gelman and Meng, 1998; Chen and Shao, 1997) have been proposed for reducing the Monte Carlo variance which will be briefly reviewed at the end of this chapter. In the following, we restrict ourselves to the likelihood distribution and discuss how to find the initial value of $\boldsymbol{\psi}$ and then how to update its value for CAR models.

3.1.1 Finding the initial value of ψ

In Chapter 1 we reviewed several methods for estimating CAR models, which can be used for generating an initial $\boldsymbol{\psi}$. For direct CAR models, the MPLE can be used as it is fast to compute. For the generalised linear model with CAR latent variables, we can transform the data to the corresponding linear response and then fit a direct CAR model by maximum pseudo-likelihood.

The MPLE $\hat{\boldsymbol{\theta}}_{PL}$ was proven to be consistent to the true value $\boldsymbol{\theta}^*$ for the CAR processes as early as by Whittle (1954),[1] but the asymptotic results are non-trivial to derive and may not applicable to the CAR model in this study for reasons discussed in section 2.2.2. Nevertheless, we quote the following asymptotic result of the composite likelihood from Varin et al. (2011) as a rough guidance, since the pseudo-likelihood can be regarded as a special case of the composite likelihood. The MPLE is asymptotically normally distributed:

$$\sqrt{n}(\hat{\boldsymbol{\theta}}_{PL} - \boldsymbol{\theta}) \xrightarrow{d} \mathcal{N}_p(\mathbf{0}, \boldsymbol{G}^{-1}(\boldsymbol{\theta})) \tag{3.1}$$

[1] Other related literature includes Besag (1975), Mase (1995).

where $\boldsymbol{G}(\boldsymbol{\theta})$ is the Godambe information matrix defined by

$$\boldsymbol{G}(\boldsymbol{\theta}) = \boldsymbol{H}(\boldsymbol{\theta})\boldsymbol{J}(\boldsymbol{\theta})^{-1}\boldsymbol{H}(\boldsymbol{\theta}) \tag{3.2}$$

$$\boldsymbol{H}(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{\theta}}\{-\boldsymbol{\nabla}_{\boldsymbol{\theta}}\boldsymbol{u}(\boldsymbol{\theta}, \mathbf{Y})\} \tag{3.3}$$

$$\boldsymbol{J}(\boldsymbol{\theta}) = \mathbf{Var}_{\boldsymbol{\theta}}(\boldsymbol{u}(\boldsymbol{\theta}, \mathbf{Y})), \quad \boldsymbol{u}_{\boldsymbol{\theta},\mathbf{Y}} = \boldsymbol{\nabla}_{\boldsymbol{\theta}}\boldsymbol{\mathcal{P}}\boldsymbol{\mathcal{L}}(\boldsymbol{\theta}, \mathbf{Y})$$

In practice $\boldsymbol{H}(\boldsymbol{\theta})$ and $\boldsymbol{J}(\boldsymbol{\theta})$ need to be estimated from the data. For the GLM with CAR latent variable, the asymptotic behaviour of the proposed $\boldsymbol{\psi}^{(0)}$ is even more complicated but we make use of the same result on the MPLE given by the transformed data.

3.1.2 Examples

Here we give three examples of direct CAR models to show how different choices of $\boldsymbol{\psi}$ may affect the approximation of the likelihood and the distribution of the MC-MLE.

Example 1: MPLE as the $\boldsymbol{\psi}$

We show in Figure 3.1 the performance of using the MPLE as $\boldsymbol{\psi}$ for the importance sampler by comparing the Monte Carlo likelihood to the exact. All the data were generated on the 20×20 torus with mean zero, $\sigma^2 = 1$ and we used the values $\rho = \{-0.24, -0.15, -0.05, 0.02, 0.1, 0.2, 0.24\}$ that cover the entire range to represent positive, negative, strong and weak spatial effects. For each case, the importance sampler value $\boldsymbol{\psi}$ is chosen to be the MPLE estimated from the data. The corresponding Monte Carlo likelihoods, approximated by using 500 Monte Carlo samples, are shown as the red solid lines, along which the dashed lines show the lower and upper 5% quantiles from 100 replications of the Monte Carlo likelihood to reflect the variability. In all plots, the likelihood is projected on

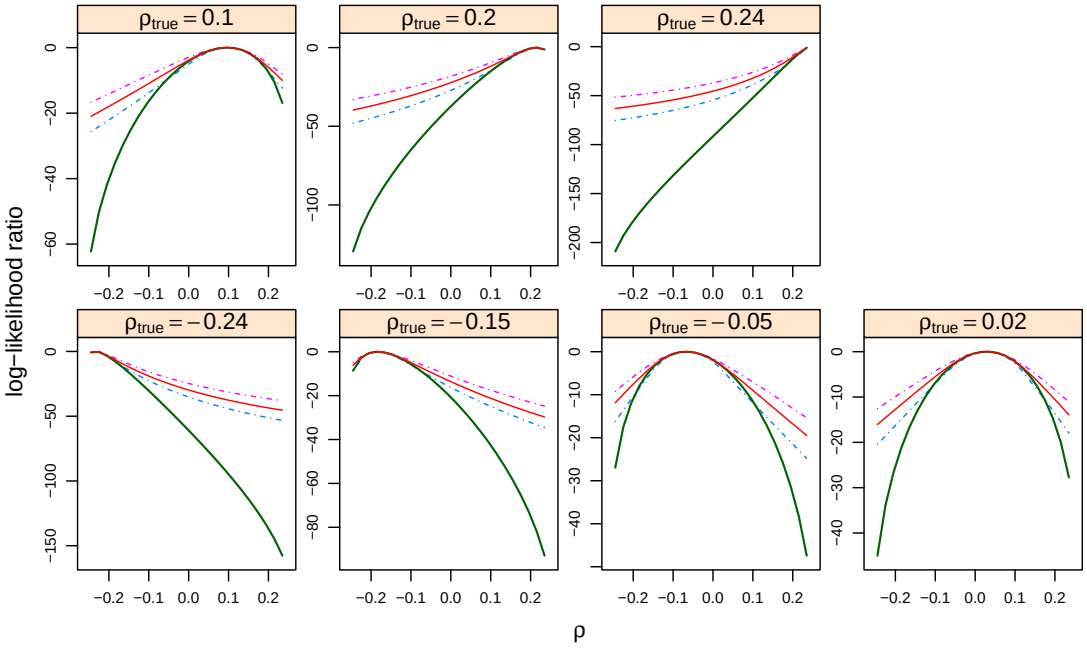


Fig. 3.1 Comparing the exact (green solid lines) and the Monte Carlo likelihood (red solid lines) using MPLE as $\boldsymbol{\psi}$ using simulated data on the 20×20 torus with different values of ρ. The dashed lines along the Monte Carlo likelihood are the lower and upper 5% quantile of the Monte Carlo likelihood from 100 replications.

the ρ coordinate with all other parameter values fixed at the MPLEs. The exact likelihood are shown as the green solid lines.

In each plot the MPLE is very close to the true value, although it usually tends to over-estimate the spatial effect which can be seen from the fact that the point at which the log-likelihood ratio equals to zero always indicates a stronger spatial effect than the true value. The Monte Carlo likelihood seems able to capture the second-order property of the likelihood function very well near the importance sampler value but for points far away, it only shows very weak first-order information. The approximated 95% limit bands also show that the variance is severely under-estimated for parameter values far away from $\boldsymbol{\psi}$.

We may conclude that the MPLE is usually a good choice for $\boldsymbol{\psi}$ for approximating the likelihood function around its mode, which is the region of our interest; however at the same time it is important to bear in mind that the performance

of the Monte Carlo approximation deteriorates with the distance between the evaluation point and the importance sampler value $\boldsymbol{\psi}$.

The huge difference in scales for the plots also show that the distance between two distributions indexed by different values of ρ is non-euclidean in $\Delta\rho$. The distributions with weak spatial effect (ρ close to zero) are similar to each other; while for stronger spatial correlation the difference between two distributions can be drastic with only a small change in ρ. This suggests that a transformation of ρ could be useful in controlling variance.

Example 2: $\boldsymbol{\psi}$ and the shape of the Monte Carlo likelihood

Now we have a closer look at how the choice of $\boldsymbol{\psi}$ may affect the shape of the Monte Carlo likelihood. Same as before, we use the data simulated on the 20×20 torus, with true parameter value $\rho = 0.1, \sigma^2 = 1$ and mean zero. Then we vary the value of ρ_{ψ} for the importance sampler to be $(-0.2, 0, 0.1, 0.15)$ with other parameter values fixed at the true value.

The left panel of Figure 3.2 shows the contours of the Monte Carlo likelihood projected on the space of ρ and σ^2 overlaid on the image of the exact log-likelihood ratios. The log-likelihood ratios in all plots are $\log \ell(\boldsymbol{\theta}) - \log \ell(\hat{\boldsymbol{\theta}})$, where $\hat{\boldsymbol{\theta}}$ is the exact MLE, for comparison. The right panels show the corresponding profile likelihoods with the the red and black solid lines being the Monte Carlo approximation and the exact values. The blue dashed lines are the $\pm 2 s.e.(\hat{\ell}_{\hat{\sigma}^2_{\psi}}(\rho))$.

The Monte Carlo profile likelihood shows different shapes with different $\boldsymbol{\psi}$. From both left and right panels, the Monte Carlo likelihood is very close to the exact value at a small neighbourhood of $\boldsymbol{\psi}$ but as the evaluation point gets further away from $\boldsymbol{\psi}$, the approximation can no longer capture the second-order information of the likelihood function and thus extend linearly toward each ends. The plug-in variance estimates also largely under-estimate variability where the approximation is poor.

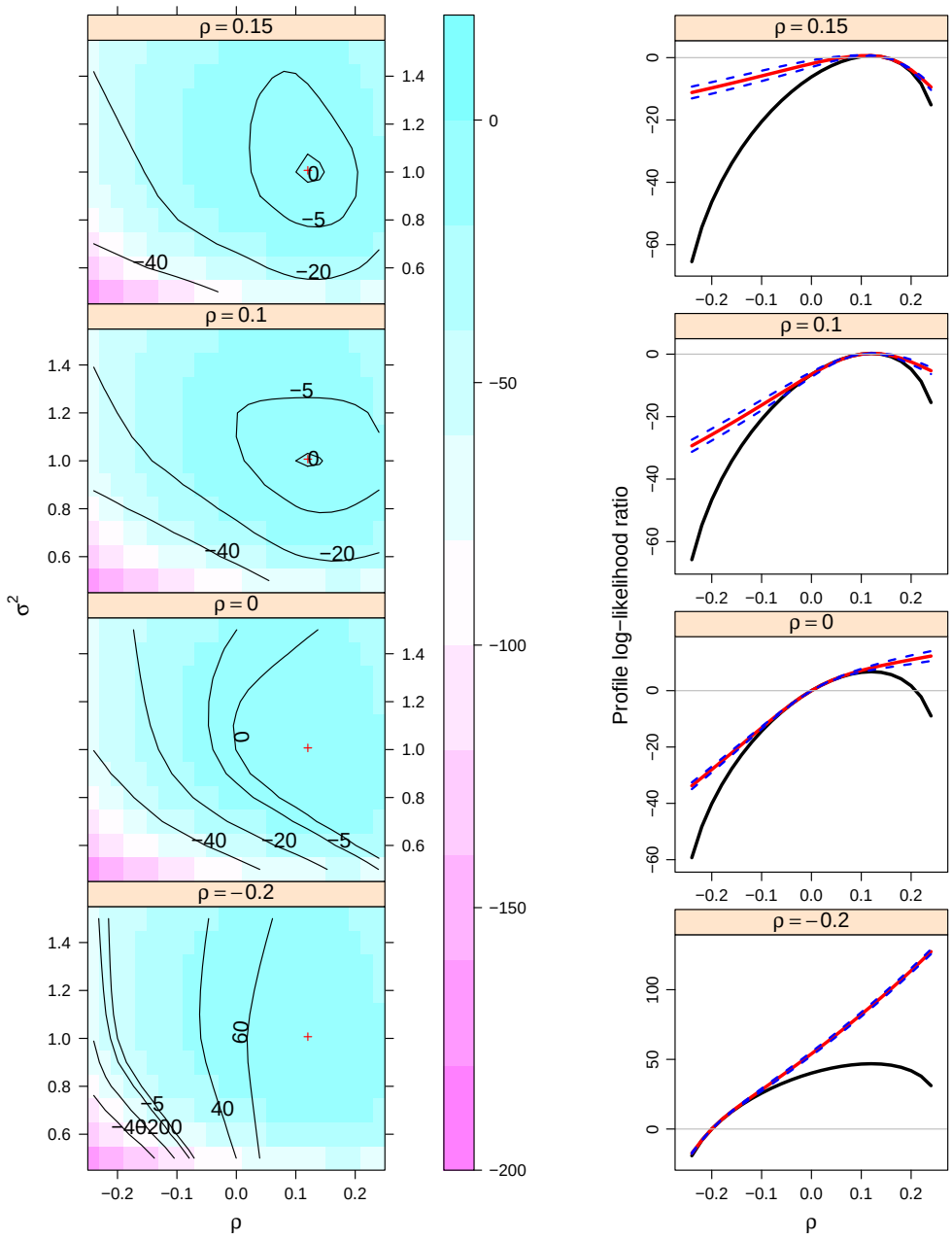


Fig. 3.2 Left panels: contour plots of the Monte Carlo likelihood using different ρ_ψ values, overlaid on top of the exact log-likelihood ratio image; the red cross in each plot is the exact MLE $\hat{\boldsymbol{\theta}}$ and all values are re-centred by $\log \ell(\hat{\boldsymbol{\theta}})$. Right panels: the corresponding Monte Carlo profile likelihood of ρ (red solid lines) v.s. the exact (black solid lines); the blue dashed lines show the estimated $\pm 2 s.e.(\hat{\ell}_{\hat{\sigma}^2_\psi}(\rho))$ limits.

This can be especially problematic when the importance sampler value $\boldsymbol{\psi}$ is too far away from the MLE $\hat{\boldsymbol{\theta}}$ as shown by the bottom row in the figure. The Monte Carlo likelihood gives only very vague gradient information toward the target and the second-order information is non-existent. This may lead to an MC-MLE at the boundary of the parameter space if directly optimising the Monte Carlo likelihood function. In practice if we use a poor starting value for $\boldsymbol{\psi}$ and iterate the optimisation by using the MC-MLE of the previous step as current step $\boldsymbol{\psi}$ then the procedure might converge very slowly or even just bounce back and forward on the parameter boundaries without converging. Therefore, for both one-step or iterative optimisation of the Monte Carlo likelihood function, the choice of $\boldsymbol{\psi}$ is crucial in finding the MC-MLE.

A poor starting $\boldsymbol{\psi}$ is not entirely useless as it at least provides accurate approximations near $\boldsymbol{\psi}$ where the first-order information usually dominates the shape of the function. As the $\boldsymbol{\psi}$ gets closer to the maximum point of the function, the gradient gets closer to zero and the second-order properties become more obvious. Therefore, the shape information of the Monte Carlo likelihood can be used as guidance to how close the current $\boldsymbol{\psi}$ is to the maximum; this idea will be elaborated in Chapter 4 with the example continued there.

Example 3: ψ and the distribution of MC-MLE

In the third example, we compare the distribution of the MC-MLE by replicating the one-step optimisation of the Monte Carlo likelihood with two different values of $\boldsymbol{\psi}$. The data is the same as the previous example and the the procedure is repeated 100 times in each case. Figure 3.3 shows the density plots of the 100 MC-MLEs for $\boldsymbol{\psi}$ equal to the MPLE in the left panel and $\rho_{\psi} = 0$ with other value being fixed at the truth on the right.

When using the MPLE for $\boldsymbol{\psi}$ which is near to the exact MLE, the MC-MLE tends to concentrate around the exact value with small variability, while using

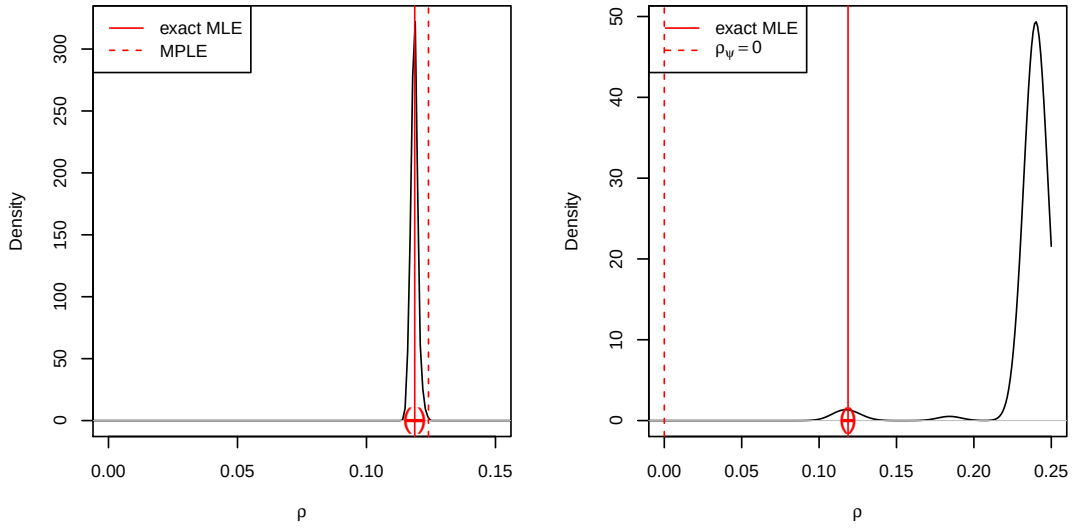


Fig. 3.3 Density plot of the MC-MLE replicates using different values of ρ_{ψ}. The left panel uses the MPLE while the right panel uses $\rho_{\psi} = 0$, and other parameter values fixed at the truth for $\boldsymbol{\psi}$. The red solid lines show the exact MLE and the dashed lines are the value of ρ_{ψ}. The red thick lines within the brackets at the bottom show the 95% CIs of the exact MLE.

$\rho_{\psi} = 0$, which is far from the exact as in the right panel of the figure, the largest mode of the MC-MLE appears at the right end of the range of ρ and totally misses the exact MLE. Although in some of the simulations, the MC-MLEs given by the poor $\boldsymbol{\psi}$ are also close to the exact value as shown by the flat mode around the red line in the right panel, the variability is too large to provide precise location of the MLE. Nevertheless the estimates do provide information about the right direction toward the target value. This shows the choice of $\boldsymbol{\psi}$ can greatly affect the performance of the MC-MLE and a $\boldsymbol{\psi}$ value near to the exact maximum point is more likely to lead to a reliable estimate of the MLE from the Monte Carlo likelihood.

All the examples illustrate the effect of $\boldsymbol{\psi}$ when approximating the likelihood and finding the MC-MLE. In general, when $\boldsymbol{\psi}$ is far away from the maximum, the shape of the Monte Carlo likelihood is dominated by the gradient information and

a one-step optimisation can lead to unreliable MC-MLE's with large variance. This is due to the fact that the Monte Carlo variance is non-constant over the parameter space; therefore, to improve the results we need to reduce the Monte Carlo variance for the area around the maximum, which can be done either by increasing the Monte Carlo sample size or, by moving $\boldsymbol{\psi}$ close to the target. This naturally suggests that an iterative optimisation procedure should be used when starting from knowing nothing about the closeness of $\boldsymbol{\psi}$ and $\hat{\boldsymbol{\theta}}$; moreover the convergence of the iterative procedure can also help improve the accuracy of the MC-MLE as it should ensure that the current $\boldsymbol{\psi}$, which is also the MC-MLE from the previous step, is close to the exact MLE.

An iterative optimisation procedure for Monte Carlo approximation was used in early literature (Ripley, 1987; Geyer and Thompson, 1992; Geyer and Møller, 1994); recent applications include Christensen and Ribeiro (2002); Giorgi and Diggle (2015) with corresponding `R` packages `geoRglm` and `PrevMap`, where the Monte Carlo likelihood is used in geo-statistical models and the iteration needs to be done manually. The package `glmm` (Knudson, 2015) implements an iterative optimisation procedure based on the package `trust` (Geyer, 2015). Most existing work assumes a satisfactory initial value of $\boldsymbol{\psi}$ is available and the iterative procedures update $\boldsymbol{\psi}$ by simply using the MC-MLE from previous step without considering the Monte Carlo variance; this may be problematic as shown in the above examples. The procedure might waste a lot of time bouncing around the boundary of the parameter space and converge very slowly. In the next section we consider how to avoid this situation by using estimates of the variance.

3.2 Optimisation

The Monte Carlo likelihood function can be maximised by using any appropriate numerical optimisation algorithms. We first discuss how to update the importance

sampler value in the iterative procedure and the convergence criteria for the final output of the MC-MLE. Then based on these rules, we demonstrate the iterative procedure with an example using simulated data on the torus. Finally, we comment on the shape of the likelihood function for the CAR models.

3.2.1 Experimental region and updating $\psi^{(k)}$

Given the starting value $\boldsymbol{\psi}^{(0)}$ for the importance sampler, the most natural update scheme for $\boldsymbol{\psi}^{(k+1)}$ would be to use the MC-MLE $\hat{\boldsymbol{\theta}}^{s}_{\boldsymbol{\psi}^{(k)}}$ obtained from the kth step; however as was pointed out in the previous section, this MC-MLE might be misleading and even further away from the target than $\boldsymbol{\psi}^{(k)}$ if the $\boldsymbol{\psi}^{(k)}$ itself is far away from the exact MLE.

Increasing the Monte Carlo sample size s will help by reducing the variance but at the same time this will increase the computational burden which is not desirable at the early stages of the iterative procedure. Hence instead of always moving to the MC-MLE, we define an experimental region $\mathbf{E}^{(k)}$ for updating $\boldsymbol{\psi}^{(k)}$. The new $\boldsymbol{\psi}^{(k+1)}$ is set to $\hat{\boldsymbol{\theta}}^{s}_{\boldsymbol{\psi}^{(k)}}$, the MC-MLE found at the kth step if it is within the experimental region; otherwise we set $\boldsymbol{\psi}^{(k+1)}$ to some $\tilde{\boldsymbol{\theta}}_{\boldsymbol{\psi}^{(k)}}$ between $\boldsymbol{\psi}^{(k)}$ and the MC-MLE:

$$\boldsymbol{\psi}^{(k+1)} = \begin{cases} \hat{\boldsymbol{\theta}}^{s}_{\boldsymbol{\psi}^{(k)}}, & \hat{\boldsymbol{\theta}}^{s}_{\boldsymbol{\psi}^{(k)}} \in \mathbf{E}^{(k)} \\ \tilde{\boldsymbol{\theta}}_{\boldsymbol{\psi}^{(k)}}, & \text{otherwise.} \end{cases} \tag{3.4}$$

One possible value for $\tilde{\boldsymbol{\theta}}_{\boldsymbol{\psi}^{(k)}}$ is the intersection of the boundary of $\mathbf{E}^{(k)}$ and the segment between $\boldsymbol{\psi}^{(k)}$ and $\hat{\boldsymbol{\theta}}^{s}_{\boldsymbol{\psi}^{(k)}}$.

Ideally we would like $\mathbf{E}^{(k)}$ to be the region within which $\mathbb{P}(\hat{l}_{\boldsymbol{\psi}^{(k)}}(\boldsymbol{\theta}) > 0)$ is large. This means that any point within this region will increase the likelihood with large probability. Then by using the asymptotic result derived for the Monte Carlo

likelihood in expression (2.21), we can define this region to be

$$\mathbf{E}^{(k)} = \left\{ \boldsymbol{\theta} \in \boldsymbol{\theta} : \hat{\ell}_{\boldsymbol{\psi}^{(k)}}(\boldsymbol{\theta}) - 2s.d(\hat{\ell}_{\boldsymbol{\psi}^{(k)}}(\boldsymbol{\theta})) > 0 \right\} \tag{3.5}$$

Unfortunately, the exact variance in the the asymptotic distribution is unknown and the result is only an approximation for $\boldsymbol{\theta} \neq \boldsymbol{\psi}^{(k)}$ near $\boldsymbol{\psi}^{(k)}$. The plug-in variance estimate $\hat{V}(\hat{\ell}_{\boldsymbol{\psi}^{(k)}})$ can be used but it also tends to under-estimate the true variability for $\boldsymbol{\theta}$ far from $\boldsymbol{\psi}^{(k)}$. Nevertheless, we can still use the method to find a coarse approximation for the experimental region.

Figure 3.4 shows the experimental region as defined in equation (3.5) for two different importance sampler values. We used the same simulated data on the 20×20 torus in the previous examples. The exact log-likelihood ratio $\ell_{\boldsymbol{\psi}}$ is shown as the black curve in each plot and the exact MLE is marked by the black bar along the line of $\ell_{\boldsymbol{\psi}} = 0$. The corresponding Monte Carlo likelihood is plotted as the red solid line enveloped by the blue dashed lines of the approximated 95% limits. The red triangles mark the locations of the importance sampler values for ρ and the red bar show the corresponding MC-MLEs. The experimental regions are shown by the thick blue intervals.

In both plots the experimental regions cover the exact MLE but the interval is not very useful as the Monte Carlo variance is hugely underestimated at the right end. Meanwhile, finding the boundaries of the experimental region is also non-trivial. In the above example, we found the end points of the intervals numerically by interpolating the Monte Carlo likelihood; this requires a number of evaluations of the Monte Carlo likelihood and as the dimension of the parameter increases, the boundary of $\mathbf{E}^{(\mathbf{k})}$ becomes more difficult to define.

A more practical approach would be to check the variance estimate at $\hat{\boldsymbol{\theta}}^{s}_{\boldsymbol{\psi}^{(k)}}$, the MC-MLE given by $\boldsymbol{\psi}^{(k)}$. According to our previous analysis, the Monte Carlo

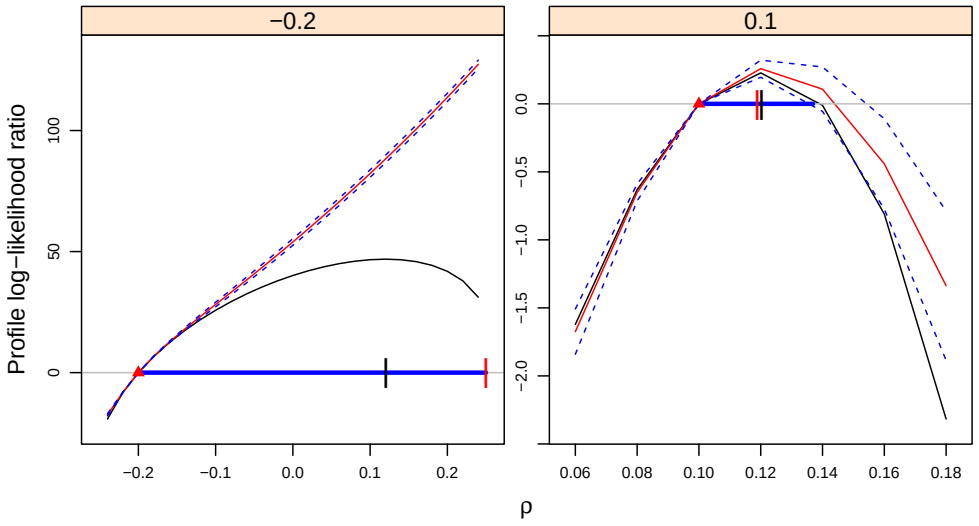


Fig. 3.4 Experimental regions using two different importance sampler values, shown as the blue thick intervals on the level of $\ell_{\psi} = 0$. Details to be found in text.

variance should scale at $1/s$ therefore we define the following update rule

$$\boldsymbol{\psi}^{(k+1)} = \begin{cases} \hat{\boldsymbol{\theta}}^{s}_{\boldsymbol{\psi}^{(k)}}, & \hat{V}\left(\hat{\ell}(\boldsymbol{\theta}^{s}_{\boldsymbol{\psi}^{(k)}})\right) \leq a/s \\ (1-b)\,\boldsymbol{\psi}^{(k)} + b\,\hat{\boldsymbol{\theta}}^{s}_{\boldsymbol{\psi}^{(k)}} & \text{otherwise.} \end{cases} \tag{3.6}$$

where $a > 0$ and $0 < b < 1$ are tuning parameters to control the accuracy and penalty on the variance. Optimal values for a and b will generally be difficult to derive. From the asymptotic result in (2.21), we have

$$\mathrm{Var}(\hat{\ell}(\boldsymbol{\theta}^{s}_{\boldsymbol{\psi}^{(k)}}) \approx \frac{\mathrm{Var}(\hat{R})}{sR^2(\boldsymbol{\psi}, \boldsymbol{\theta})},$$

so ideally we would like to choose a approximately equal to $\dfrac{\mathrm{Var}(\hat{R})}{R^2(\boldsymbol{\psi}, \boldsymbol{\theta})}$ which is the squared coefficient of variation (CV) of $\hat{R}$ and measure the dispersion of the distribution of $\hat{R}$. In practice we recommend to use $a = 1$ as distribution with CV < 1 is considered to be low-varianced. The value of b controls the updating step size in the convex optimisation problem so it does not affect the convergence

of the algorithm. In the following implementation, we simply choose $b = 0.5$. In our package `mclcar`, these two values can be specified by users.

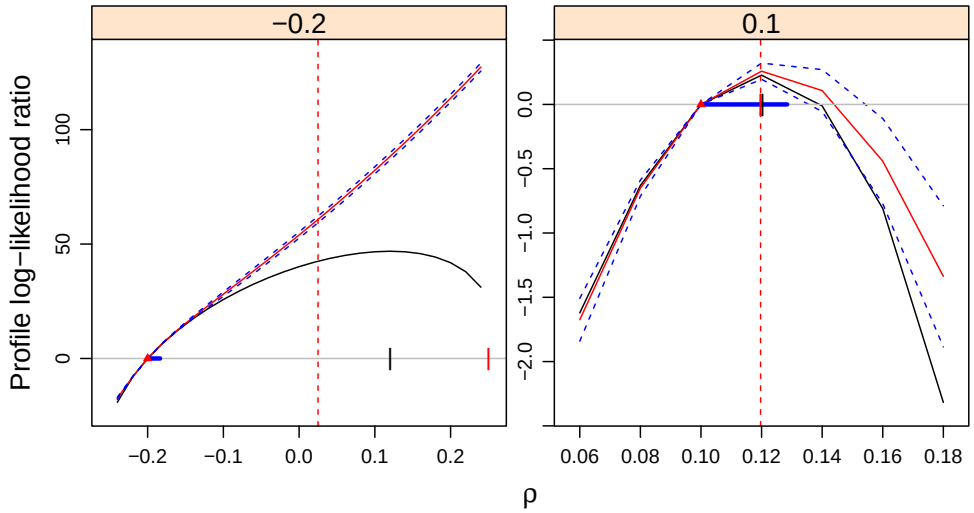


Fig. 3.5 Experimental regions with variance constraints. Same as Figure 3.4. The red dashed line shows the location of $\boldsymbol{\psi}^{(k+1)}$.

Figure 3.5 shows the above updating rule for the same example as in Figure 3.4. The dashed vertical line in each plot shows the location of $\boldsymbol{\psi}^{(k+1)}$. The experimental regions are smaller than before especially the one in the left panel. The new rule successfully moves closer to the target value when $\boldsymbol{\psi}^{(k)}$ is far away and when the importance sampler value is near to the maximum point, the update is not much affected by the variance constraints.

3.2.2 Convergence criteria

In most existing implementations, the MC-MLE is found by the result of a single step of maximising the Monte Carlo likelihood, although it is recommended the optimisation be repeated until convergence. There is a lack of literature on defining the convergence criteria and thus little has been done to automate the iterative procedure. We propose two simple tests for convergence and with the updating rule

proposed in the previous section, we demonstrate an automatic iterative procedure for maximising the Monte Carlo likelihood.

There are two natural criteria for monitoring convergence:

1. The improvement in the likelihood is small enough.

2. The difference between the MC-MLEs output from two consecutive iterations is small enough.

For the first criterion, we can monitor the value of the maximum Monte Carlo likelihood in each iteration. Since by the end of the kth iteration, the Monte Carlo likelihood evaluation at $\boldsymbol{\psi}^{(k+1)}$ is actually an approximation to the log likelihood ratio $\ell(\boldsymbol{\psi}^{(k+1)}) - \ell(\boldsymbol{\psi}^{(k)})$, we would expect the procedure converges when $\hat{\ell}_{\boldsymbol{\psi}^{(k)}}(\boldsymbol{\psi}^{(k+1)}) < \varepsilon_l^{(k)}$ for some small $\varepsilon_l^{(k)} > 0$. In general, we would set $\varepsilon_l^{(k)}$ to scale as

$$\varepsilon_l^{(k)} \approx 2\sqrt{\mathrm{Var}\left(\hat{\ell}_{\boldsymbol{\psi}^{(k)}}(\boldsymbol{\psi}^{(k+1)})\right)}$$

This is based on the asymptotic distribution of the Monte Carlo likelihood that if $\hat{\ell}_{\boldsymbol{\psi}^{(k)}}$ is smaller than the $\varepsilon_l^{(k)}$ defined above, it is hard to tell whether the change in Monte Carlo likelihood is non-zero due to the improvement or pure error, without increasing the Monte Carlo sample size s. In practice, we can use the $\hat{V}\left(\hat{\ell}_{\boldsymbol{\psi}^{(k)}}(\boldsymbol{\psi}^{(k+1)})\right)$ to estimate $\varepsilon_l^{(k)}$.

For the second criterion, Euclidean distance can be used to assess the difference between the MC-MLEs but again the Monte Carlo error needs to be taken into consideration; therefore in practice, it is more straightforward to consider the marginal error bands of the MC-MLE. The 95% error bands of the MC-MLE can also be approximated by using $\hat{V}$. With these bands for $\boldsymbol{\psi}^{(k)} = \hat{\boldsymbol{\theta}}^s_{\boldsymbol{\psi}^{(k-1)}}$, if the MC-MLE $\hat{\boldsymbol{\theta}}^s_{\boldsymbol{\psi}^{(k)}}$ is within the error bands then without increasing the Monte Carlo sample size s, we can not tell if $\hat{\boldsymbol{\theta}}^s_{\boldsymbol{\psi}^{(k)}}$ is significantly different from $\hat{\boldsymbol{\theta}}^s_{\boldsymbol{\psi}^{(k-1)}}$ and therefore we might conclude that the iteration converges at step k.

Theoretically we can always achieve better accuracy by increasing the Monte Carlo sample size but this is constrained by the available computing resources; hence in practice, we recommend using a small s for the first few iterations and when the procedure achieves convergence, we increase s according to the size of the problem and repeat the iterative procedure till it converges once more. At this stage we may output the final result or if resources allow we may repeat the procedures with even more samples. In our experience, for a given s the entire procedure usually takes less than 5 iterations to converge.

Although both criteria are easy to apply in most situations, the first one is preferred as it only requires the comparison of two scalars and is fast to compute while the second criterion needs to compare all the marginals and approximate the Hessian matrix at the MC-MLE for multiple comparisons when $\boldsymbol{\theta}$ is a vector. This can be computationally expensive.

An example of the iterative procedure

Here we use the same data simulated on the 20×20 torus as in previous examples to demonstrate the automatic implementation of the iterative procedure. In the example, we try to optimise the Monte Carlo profile likelihood of ρ and to show the effect of a poor importance sampler value $\rho_{\boldsymbol{\psi}^{(0)}} = -0.2$ as the initial value instead of the MPLE. The Monte Carlo sample size is set to be $s = 500$ and after the first convergence, we increase it to $s = 1000$.

The automatic iterative procedure achieved convergence within 5 steps and the total computing time was about 138 seconds. Table 3.1 summarises the results of the 5 iterations and the exact MLE. The results from the last two iterations are both very close to the exact.

In this procedure we used the convergence rule based on the value of the Monte Carlo likelihood. In Figure 3.6, we plot for the same example the evolution of $\boldsymbol{\psi}^{(k)}$ and also illustrate the convergence rule based on the error band of MC-MLE. In

Table 3.1 Results of the iterative optimisation procedure

iteration	cumulated time	ρ	σ^2	convergent
0	0	-0.2000	1.1711	–
1	14.6	0.0245	1.0487	FALSE
2	29.1	0.1368	0.9876	FALSE
3	60.1	0.1162	0.9988	FALSE
4	88.0	0.1188	0.9974	TRUE
5	137.8	0.1183	0.9976	TRUE
exact	–	0.1187	0.9974	–

each plot, the Monte Carlo profile likelihood is shown as the solid curve enveloped by the $2s.e$ error bands using $\hat{V}$; the exact log-likelihood ratio function is plotted as the black solid line for comparison. The triangles and the underlying shaded areas shows exact MLE, current importance value $\boldsymbol{\psi}^{(k)}$, the Monte Carlo MLE and the corresponding 95% confidence intervals, experimental region and the $2s.e$ error bands. The red dashed line shows the location of the the updated $\boldsymbol{\psi}^{(0)}$ according to the rule in equation (3.6).

In the first iteration $\boldsymbol{\psi}^{(0)}$ is apparently too far away from the MLE and the Monte Carlo likelihood totally fail to capture the second order property. The MC-MLE lies at the boundary of the parameter space since Monte Carlo likelihood is increasing for the entire range of ρ; therefore we set $\boldsymbol{\psi}^{(1)}$ to be the mid-point of $\boldsymbol{\psi}^{(0)}$ and $\hat{\boldsymbol{\theta}}_{\boldsymbol{\psi}^{(0)}}$. The same situation happens in the second iteration but the $\boldsymbol{\psi}^{(1)}$ has already become closer to the exact MLE as shown by the Monte Carlo likelihood getting close to zero. The Monte Carlo likelihood starts to capture the second-order property in the third and fourth iterations and $\boldsymbol{\psi}^{(k+1)}$ is set to be the MC-MLE $\hat{\boldsymbol{\theta}}_{\boldsymbol{\psi}^{(k)}}$. The experimental region shown by the blue shaded area begins to cover both the 95% confidence interval of the exact MLE (black shaded area) and the $2s.e$ error bands of the MC-MLE (red shaded area). In the fourth iteration, the procedure converges and therefore in the fifth iteration we increase the sample size and the procedure also appear to converge under both convergence criteria.

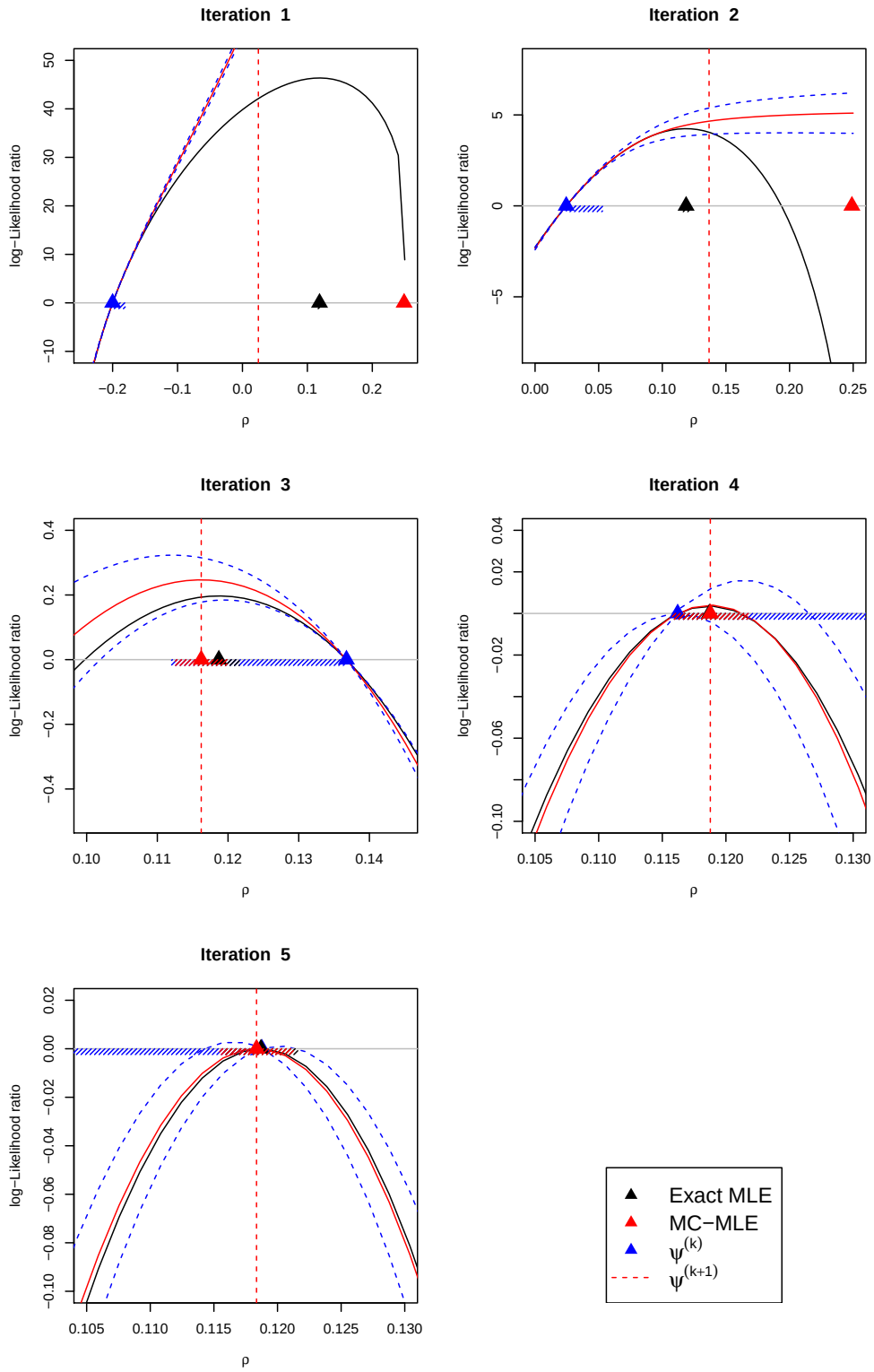


Fig. 3.6 Iterative optimization of the Monte Carlo likelihood. Details to be found in the text.

3.2.3 The likelihood function

In this section we discuss some details of the likelihood function which will affect the performance of the optimisation algorithm. Throughout this thesis the R package `maxLik` (Henningsen and Toomet, 2011) is used as it provides convenient tools for likelihood analysis. The gradient is provided if available to help convergence of the "BFGS" algorithm and the Hessian is provided for fast evaluations in the final output. We first list the general form of the gradient and Hessian function for the CAR models; then we try to analyse the shape of the likelihood function and form some general idea about under what circumstances the optimisation algorithm might fail.

The Gradient and Hessian

Chapter 2 gave the Monte Carlo approximations in equations (2.5), (2.6) and (2.7) for three situations which are essentially using the same form. Denote by $h(\boldsymbol{\theta})$ the un-normalised likelihood function in general, then the corresponding gradient of the Monte Carlo likelihood function for **Approx. I** is

$$\begin{aligned}\boldsymbol{\nabla}\hat{\boldsymbol{\ell}}^{\boldsymbol{s}}_{\boldsymbol{\psi}}(\boldsymbol{\theta}) &= \frac{1}{\hat{\mathbf{I}}(\boldsymbol{\theta})}\frac{1}{s}\sum_{i=1}^{s}\frac{h_i(\boldsymbol{\theta})}{h_i(\boldsymbol{\psi})}\boldsymbol{\nabla}\log \boldsymbol{h_i}(\boldsymbol{\theta}) = \frac{1}{C(\boldsymbol{\theta})}\sum_{i=1}^{s}c_i(\boldsymbol{\theta})\boldsymbol{\nabla}\log \boldsymbol{h_i}(\boldsymbol{\theta}) \\ &= \sum_{i=1}^{s}\omega_i(\boldsymbol{\theta})\boldsymbol{\nabla}\log \boldsymbol{h_i}(\boldsymbol{\theta}) \end{aligned} \tag{3.7}$$

where $C(\boldsymbol{\theta}) = \sum_{i=1}^{s} c_i(\boldsymbol{\theta})$, $c_i(\boldsymbol{\theta}) = \dfrac{h_i(\boldsymbol{\theta})}{h_i(\boldsymbol{\psi})}, \omega_i(\boldsymbol{\theta}) = \dfrac{c_i(\boldsymbol{\theta})}{C(\boldsymbol{\theta})}$. The gradient of the Monte Carlo likelihood function is proved to be an unbiased estimator of the exact likelihood function with asymptotic normality as shown by (2.9). The Hessian function is given by

$$\boldsymbol{\nabla^2}\hat{\boldsymbol{\ell}}^{\boldsymbol{s}}_{\boldsymbol{\psi}}(\boldsymbol{\theta}) = -\boldsymbol{\nabla}\hat{\boldsymbol{\ell}}^{\boldsymbol{s}}_{\boldsymbol{\psi}}(\boldsymbol{\theta})\boldsymbol{\nabla^T}\hat{\boldsymbol{\ell}}^{\boldsymbol{s}}_{\boldsymbol{\psi}}(\boldsymbol{\theta}) + \frac{1}{C(\boldsymbol{\theta})}\sum_{i=1}^{s}c_i(\boldsymbol{\theta})[\boldsymbol{\nabla}\log \boldsymbol{h_i}(\boldsymbol{\theta})\boldsymbol{\nabla^T}\log \boldsymbol{h_i}(\boldsymbol{\theta}) + \boldsymbol{\nabla^2}\log \boldsymbol{h_i}(\boldsymbol{\theta})] \tag{3.8}$$

which requires the gradient and Hessian of $\log h(\boldsymbol{\theta})$. For the CAR models the Hessian matrix of $\log h(\boldsymbol{\theta})$ is given by

$$\boldsymbol{\nabla^2 \log h(\theta)} = \begin{array}{c} \\ \rho \\ \sigma^2 \\ \boldsymbol{\beta} \end{array} \begin{array}{c} \begin{array}{ccc} \rho & \sigma^2 & \boldsymbol{\beta} \end{array} \\ \begin{pmatrix} \dfrac{\partial^2 \log h(\boldsymbol{\theta})}{\partial \rho^2} & \dfrac{\partial^2 \log h(\boldsymbol{\theta})}{\partial \rho \partial \sigma^2} & \mathbf{0} \\ \dfrac{\partial^2 \log h(\boldsymbol{\theta})}{\partial \sigma^2 \partial \rho} & \dfrac{\partial^2 \log h(\boldsymbol{\theta})}{\partial (\sigma^2)^2} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \dfrac{\partial^2 \log h(\boldsymbol{\theta})}{\partial \boldsymbol{\beta}^2} \end{pmatrix} \end{array} \tag{3.9}$$

Entries of the matrix elements for a few CAR models used in this thesis is given in the Appendix B.1. Similar results can be derived for **Approx. II** and **Approx. III**.

Shape of the Likelihood function

In the previous discussion, we assumed that the likelihood function satisfies the regularity conditions and has a unique maximum at $\hat{\boldsymbol{\theta}}_n$; however this might not be true even for the exact likelihood function of some CAR models.

The unimodality of the likelihood of direct CAR models has been discussed by Ripley (see 1991, chap. 2), where the profile likelihood of the parameter ρ is proved to be unimodal when the mean is known; when there is a parametrized mean to be estimated, it usually can be converted to a problem with zero mean. The book also gives an example that for some CAR models, certain optimization methods might fail when the likelihood function has its maximum close to the boundary of the parameter space and then plummets to $-\infty$ at some point beyond the maximum before reaching the boundary. Such situations are very likely to happen in practice as shown in the Appendix A.1.

Another problem comes from models with the CAR latent process when the observed data does not show much spatial information. Here we illustrate the effect

for the logistic regression with a CAR latent variable. Suppose we have binomial observations $\mathbf{Y}$ with CAR latent variables $\mathbf{Z}$, then the model is

$$\begin{cases} \mathbf{Y} \sim \text{Binomial}(\boldsymbol{m}, \boldsymbol{p}), \\ \text{logit}(\boldsymbol{p}) = \mathbf{X}\boldsymbol{\beta} + \mathbf{Z} \\ \mathbf{Z} \sim \mathcal{N}(\mathbf{0},\ \boldsymbol{Q}),\ \ \boldsymbol{Q} = \sigma^{-2}(\mathbf{I} - \rho\mathbf{W}) \end{cases} \tag{3.10}$$

and the likelihood function can be written as

$$\begin{aligned} L(\boldsymbol{\theta}; \mathbf{y}) = (2\pi)^{-N/2}|\boldsymbol{Q}|^{1/2} \prod_{i=1}^{N} \binom{m}{y_i} \exp(\mathbf{y}^T\mathbf{X}\boldsymbol{\beta}) \\ \int \exp\left\{\mathbf{y}^T\mathbf{Z} - \sum_{k=1}^{N} m_k \log(1 + \exp(\mathbf{x}_k^T\boldsymbol{\beta} + z_k)) - \frac{\mathbf{Z}^T\boldsymbol{Q}\mathbf{Z}}{2}\right\} \mathrm{d}\mathbf{Z} \end{aligned} \tag{3.11}$$

Now we analyse how the number of observations and the number of trials in each unit may affect the shape of the profile likelihood of ρ. Assume all $m_k = m$ and denote by $\exp\{F(\mathbf{Z})\}$ the integrand for the latent variable $\mathbf{Z}$ in the above likelihood function, then

$$F(\mathbf{Z}) = \mathbf{y}^T\mathbf{Z} - m\sum_{k=1}^{N} \log(1 + \exp(\mathbf{x}_k^T\boldsymbol{\beta} + z_k)) - \frac{\mathbf{Z}^T\boldsymbol{Q}\mathbf{Z}}{2} \tag{3.12}$$

The Laplace approximation to the integral uses a Gaussian distribution to approximate $F(\mathbf{Z})$ with its maximum $\mathbf{Z}^*$ as the mean and the curvature at the maximum for the variance. The maximum $\mathbf{Z}^*$ can be found by numerically solving

$$F'(\mathbf{Z}) = -m\left(\frac{\exp(\mathbf{x}_1^T\boldsymbol{\beta} + z_1)}{(1 + \exp(\mathbf{x}_1^T\boldsymbol{\beta} + z_1))}, \ldots, \frac{\exp(\mathbf{x}_N^T\boldsymbol{\beta} + z_N)}{(1 + \exp(\mathbf{x}_N^T\boldsymbol{\beta} + z_N))}\right)^T + \mathbf{y} - \boldsymbol{Q}\mathbf{Z} = \mathbf{0}$$

Then at $\mathbf{Z}^*$ we evaluate the negative second derivative and use it as the precision matrix of the Gaussian distribution

$$\boldsymbol{Q}^* = -F''(\mathbf{Z}^*) = m\,\text{diag}\left(\frac{\exp(\mathbf{x}_1^T\boldsymbol{\beta} + z_1^*)}{(1 + \exp(\mathbf{x}_1^T\boldsymbol{\beta} + z_1^*))^2}, \ldots, \frac{\exp(\mathbf{x}_N^T\boldsymbol{\beta} + z_N^*)}{(1 + \exp(\mathbf{x}_N^T\boldsymbol{\beta} + z_N^*))^2}\right) + \boldsymbol{Q}$$

Thus the approximated log marginal likelihood becomes

$$\begin{aligned}\tilde{\ell} &\propto \mathbf{y}^T\mathbf{X}\boldsymbol{\beta} + F(\mathbf{Z}^*) + \frac{1}{2}(\log|\boldsymbol{Q}| - \log|\boldsymbol{Q}^*|) \\ &= \mathbf{y}^T(\mathbf{X}\boldsymbol{\beta} + \mathbf{Z}^*) - m\sum_{k=1}^{N}\log(1+\exp(\mathbf{x}_k^T\boldsymbol{\beta} + z_k^*)) - \frac{\mathbf{Z}^{*T}\boldsymbol{Q}\mathbf{Z}^*}{2} + \frac{1}{2}(\log|\boldsymbol{Q}| - \log|\boldsymbol{Q}^*|)\end{aligned}$$

This function is still complicated to analyse in terms of ρ and m. For simplicity, we analyse when $\boldsymbol{\beta} = \mathbf{0}$. To make $\mathbf{Z}^*$ not depend on ρ and σ^2 we approximate the integral by expanding $F(\mathbf{x})$ at $\mathbf{Z}^* = \mathbf{0}$ which is reasonable as $\mathbf{Z}$ is the latent spatial error assumed to have mean zero. Then the approximated log likelihood becomes

$$\begin{aligned}\tilde{\ell} &\propto F'(0)^T\mathbf{Z} + \frac{1}{2}(\log|\boldsymbol{Q}| - \log|\boldsymbol{Q}^*|_{\mathbf{Z}^*=0}) \\ &\propto \frac{(\mathbf{y}-\frac{m}{2})^T(\mathbf{I}-\rho\mathbf{W})(\mathbf{y}-\frac{m}{2})}{2\sigma^2} - \frac{N}{2}\log\sigma^2 + \frac{1}{2}\sum_{i=1}^{N}\left[\log(1-\rho\lambda_i) - \log\left(1-\rho\lambda_i + \frac{m\sigma^2}{4}\right)\right]\end{aligned}$$

and the gradient is

$$\begin{aligned}\frac{\partial\tilde{\ell}}{\partial\sigma^2} &= \frac{\mathbf{b}^T(\mathbf{I}-\rho\mathbf{W})\mathbf{b}}{2(\sigma^2)^2} - \frac{N}{2\sigma^2} + \frac{m}{8}\sum_{i=1}^{N}\frac{1}{1-\rho\lambda_i + m\sigma^2/4} \\ \frac{\partial\tilde{\ell}}{\partial\rho} &= -\frac{\mathbf{b}^T\mathbf{W}\mathbf{b}}{2\sigma^2} - \frac{1}{2}\sum_{i=1}^{N}\lambda_i\left(\frac{1}{1-\rho\lambda_i} - \frac{1}{1-\rho\lambda_i+m\sigma^2/4}\right)\end{aligned}$$

where $\mathbf{b} = \mathbf{y} - m/2$.

To find the profile likelihood of ρ, we need to solve $\partial\tilde{\ell}/\partial\sigma^2 = 0$, which gives

$$\sum_{i=1}^{n}\frac{1}{(1-\rho\lambda_i)/(m\hat{\sigma}^2) + 1/4} + \frac{4\mathbf{b}^T(\mathbf{I}-\rho\mathbf{W})\mathbf{b}}{\hat{\sigma}^2} = 4N \tag{3.13}$$

The second derivative of the profile likelihood of ρ will have the form

$$\frac{\partial^2\tilde{\ell}_p}{\partial\rho^2} = \frac{1}{2}\sum\lambda_i^2\left(\frac{1}{(1-\rho\lambda_i+m\hat{\sigma}^2/4)^2} - \frac{1}{(1-\rho\lambda_i)^2}\right) < 0 \tag{3.14}$$

This shows that $\tilde{\ell}_p(\rho)$ is concave on the entire range of ρ and hence unimodal if the mode exists.

How do m and N affect the shape of $\tilde{\ell}_p(\rho)$? When N increases, we observe that in (3.14) there are more negative terms cumulated in the curvature expression and the absolute value of each of these terms is of the form

$$\frac{1}{1/\lambda_i - \rho} - \frac{1}{1/\lambda_i - \rho + c/|\lambda_i|} \geq \frac{1}{1/\lambda_i - \rho} - \frac{1}{1/\lambda_i - \rho + c/\lambda^*} \geq e(\lambda_1, \lambda_N) > 0$$

where $\lambda^* = \max|\lambda_i|$, $c = m\hat{\sigma}^2$ since $\hat{\sigma}^2 \to \sigma^2$ as $n \to \infty$. Hence for fixed m, $m\hat{\sigma}^2$ tends to some constant $c > 0$; and $e(\lambda_1, \lambda_N)$ is a function of λ_1, λ_N which is roughly determined by the number of neighbours in the spatial weight structures which is independent of N. Therefore, larger N will provide a more concentrated likelihood for ρ.

When N is fixed, equation (3.13) can be written as

$$\hat{\sigma}^2 = \frac{C_1}{N - \sum_{i=1}^{N} \frac{m\hat{\sigma}^2}{4(1-\rho\lambda_i) + m\hat{\sigma}^2}} > \frac{C_1}{N - 2 - \sum_{i=2}^{N-1} \frac{m\hat{\sigma}^2}{4(1-\rho\lambda_i) + m\hat{\sigma}^2}}$$

$$> \frac{C_1}{(n-2)(1 - \frac{m\hat{\sigma}^2}{C_2 + m\hat{\sigma}^2})}$$

where $C_1 = (\mathbf{y} - m/2)^T \mathbf{W} (\mathbf{y} - m/2)$ and $C_2 = \min\{4(1 - \lambda_2/\lambda_1), 4(1 - \lambda_{N-1}/\lambda_N)\}$. Therefore from the above inequality, it is obvious that $\hat{\sigma}^2$ is always larger than some constant, say $C_3 > 0$. Thus when m increases, $m\hat{\sigma}^2 > mC_3$ also increases which makes $|\partial^2 \tilde{\ell}_p / \partial \rho^2|$ become larger and the likelihood more informative for ρ. As $m \to \infty$, $|\partial^2 \tilde{\ell}_p / \partial \rho^2|$ is upper bounded by $\frac{1}{2} \sum (\lambda_i / (1 - \rho\lambda_i))^2$. This means that by increasing the number of trials in each unit, the spatial information contained in the likelihood can be estimated more accurately but it will not increase as m only

controls the randomness introduced by the binomial realization and when $m \to \infty$, we expect $\hat{p}_i \to p_i$ so the problem can be simplified to estimating the covariance parameter for a direct CAR model.

Example

In the following, we demonstrate the results by using the above approximation with weight matrix generated on 10×10 torus with "rook-style" neighbourhood structure. First we show how the width of a 95% Wald-type interval changes when we fix m and increase N by the torus size n. Here we assume m is very large and the $|\partial^2 \tilde{\ell}_p / \partial \rho^2|$ is upper-bounded by $\frac{1}{2} \sum (\lambda_i / (1 - \rho \lambda_i))^2$.

Figure 3.7 shows plots of the half-width of the Wald-type interval obtained by $1.96 \times s.e.(\hat{\rho})$ for different value of ρ and n. The left panel shows that for fixed torus size $n = 10$, the width increases as the spatial effect decreases; the right panel shows given ρ, the confidence width becomes smaller when the number of observed spatial units increases.

Figure 3.8 shows for a given length d of the interval, how many number of trials m in each unit is needed for different torus sizes. Here we choose the width to be $d = 0.014$ which is slightly larger than the maximum width shown in the left panel of Figure 3.7.

On the left panel, we plot the m needed to achieve the CI width against value of ρ and it shows that more trials are needed to confirm that there is no or little spatial correlation. On the right panel, we plot, for $\rho = 0.1$, the number of trials needed in each unit against the size of torus. It shows that when there are more spatial units observed, fewer repeated experiments are needed in each unit to get a tight CI for ρ.

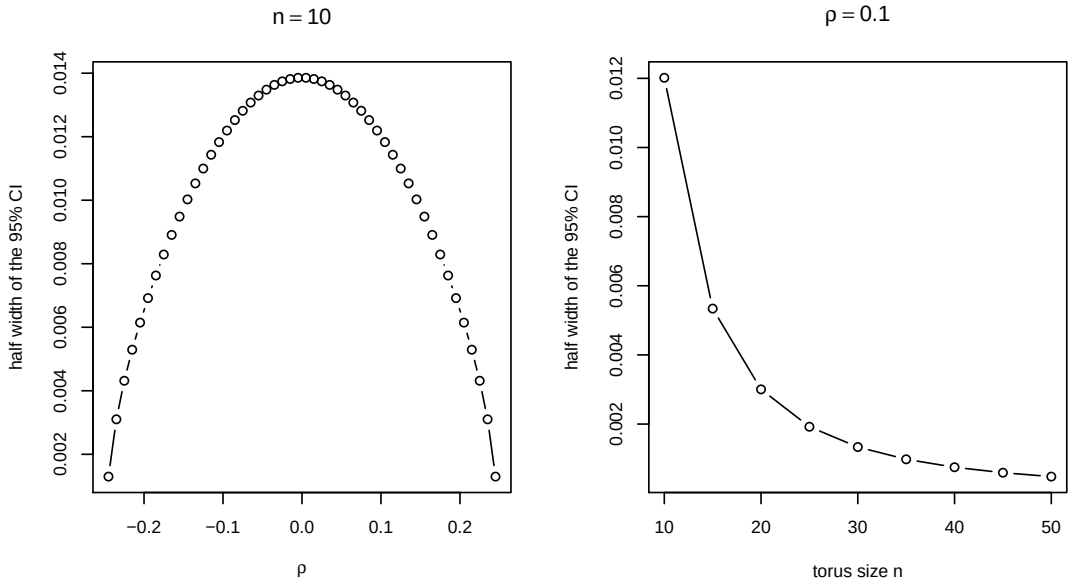


Fig. 3.7 Widths of the 95% confidence interval against the spatial coefficient ρ and torus size. The left panel shows CI width varies according to the value of ρ for data on 10×10 torus and the right panels shows the widths against the size of torus for fixing $\rho = 0.1$.

3.3 Sampling algorithm and improvements

Finally, we discuss some technical issues and alternative algorithms in the implementation of the Monte Carlo likelihood methods. We first describe the sampling algorithms used to generate the importance samples in the CAR models then we review some variance reduction techniques and alternative algorithms for optimising stochastic functions.

3.3.1 Sampling from the importance distribution

Sampling from the direct CAR model is relatively easy as it just requires the generation of multivariate normal variables. Fast algorithms that exploit the sparsity of the precision matrix are described in the book Rue and Held (2005) and several related papers such as Rue (2001) and Rue et al. (2002). For the latent CAR models, we will need to sample from the posterior CAR latent field $P_{\psi}(\mathbf{Z}|\mathbf{y})$

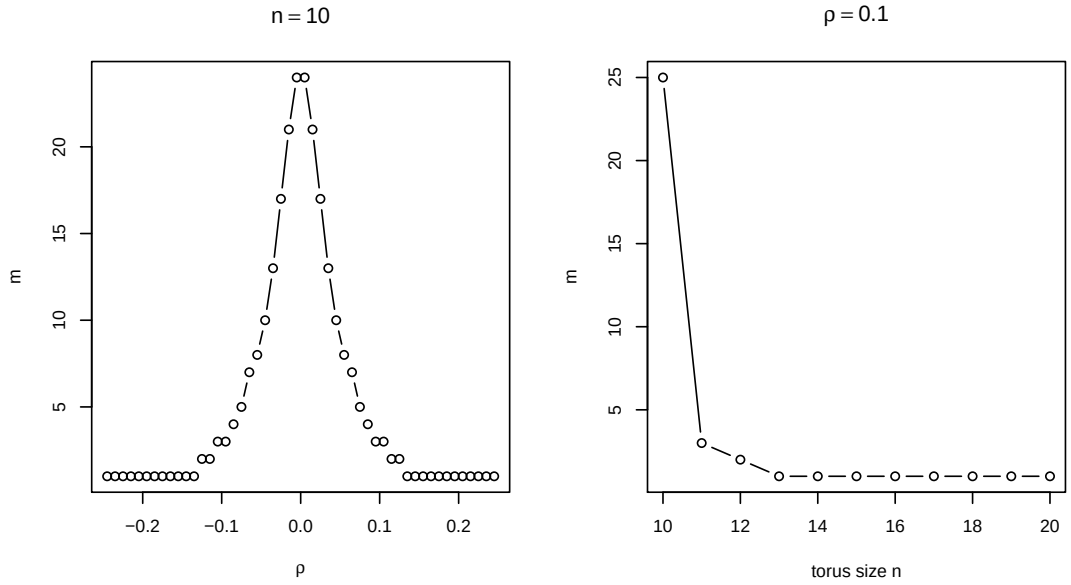


Fig. 3.8 Number of trials needed in each unit for fixing widths of the 95% CI to be 0.014. The left panels show for data on a 10×10 torus, the smallest number of trials needed when varying the value of ρ. The right panel plot when fixing $\rho = 0.1$, the smallest number of trials needed when varying the torus size. When there are more than 200 spatial units, one observation per unit is sufficient.

which is usually not available in closed form. For example, the logistic regression model with latent CAR error requires sampling from

$$\begin{aligned} P(\mathbf{Z}|\mathbf{Y}) &= \frac{P(\mathbf{Z},\mathbf{Y})}{P(\mathbf{Y})} \propto P(\mathbf{Y}|\mathbf{Z})P(\mathbf{Z}) \\ &\propto \left(\prod_{i=1}^{N} \frac{\exp(y_i(\mathbf{x}_i^T\boldsymbol{\beta}+z_i))}{(1+\exp(\mathbf{x}_i^T\boldsymbol{\beta}+z_i))^n}\right) \exp\left(-\frac{\mathbf{Z}^T(\mathbf{I}-\rho\mathbf{W})\mathbf{Z}}{2\sigma^2}\right) \end{aligned} \tag{3.15}$$

The posterior is known up to some normalising constants and usually some MCMC algorithms can be used to generate correlated samples. In the following we denote by $h(\mathbf{Y},\mathbf{Z})$ the un-normalised joint density of $\mathbf{Y},\mathbf{Z}$ and we briefly describe two algorithms used in this thesis: the random-walk Metropolis algorithm (RWMA) and the Metropolis adjusted Langevin diffusion algorithm (MALA). The latter is used as default since it is more efficient in exploring the sample space while the former is included as a comparison.

Two basic MCMC Algorithms

Random-walk Metropolis Given the current state $\mathbf{Z}$, and $\mathbf{Z}^*$ from a multivariate normal proposal, the random-walk Metropolis algorithm (RWMA) updates the sample according to the acceptance probability

$$\alpha = \min\left\{\frac{h_{\psi}(\mathbf{Y},\mathbf{Z}^*)}{h_{\psi}(\mathbf{Y},\mathbf{Z})}, 1\right\} \tag{3.16}$$

Instead of using existing functions for general problems, such as `rwmetrop()` in package `LearnBayes` (Albert, 2014), we wrote similar code that exploits the sparsity of the precision matrix of a CAR model to speed up the computation. By pivoting in the Cholesky decomposition of the precision matrix, the sparsity can be preserved in the Cholesky triangular matrix and thus less computation is required to generate

the samples. The improvement is more obvious when the size of the dataset becomes large.

MALA The Metropolis adjusted Langevin diffusion algorithm (MALA, Roberts and Tweedie (1996)) is based on the Langevin diffusion

$$\mathrm{d}\mathbf{X}_t = \frac{1}{2}\nabla \log \boldsymbol{\pi}(\mathbf{X}_t)\,\mathrm{d}t + \mathrm{d}\mathbf{W}_0,\ \mathbf{X}_0 = \mathbf{x}_0 \tag{3.17}$$

where $\mathbf{W} = (\mathbf{W}_t)_{t\geq 0}$ is a standard $d-$dimensional Wiener process. By using numerical methods such as the Euler discretization (also known as the Euler-Maruyama method (Kloeden and Platen, 1992)) the sequence of random variables $\mathbf{X}_h, \mathbf{X}_{2h}, \ldots, \mathbf{X}_{nh}$, can be approximated by

$$\mathbf{x}_{(i+1)h} = \mathbf{x}_{ih} + h\mathbf{b}(\mathbf{x}_{ih}) + \sigma(\mathbf{x}_{ih})\varepsilon \tag{3.18}$$

given the initial condition $\mathbf{X}_0 = \mathbf{x}_0$ and the step-size h, where $\varepsilon \sim \mathcal{N}(0, hI_{d\times d})$, $\mathbf{b}(\mathbf{x})$ is the drift vector and $\sigma(\mathbf{x})$ is the volatility matrix. Setting $\mathbf{b}(\mathbf{x}) = \frac{1}{2}\nabla \log \pi(\mathbf{x})$ and $\sigma(\mathbf{x}) = \mathbf{I}$ as in (3.17), the Langevin diffusion has an invariant distribution $\boldsymbol{\pi}$ and under certain conditions, the diffusion converges to $\boldsymbol{\pi}$ at an exponential rate from any starting point.

Thus, the discretization as in (3.18) of the Langevin diffusion can be used to form the basis of a Metropolis-Hastings algorithm by setting the proposal to be

$$\mathbf{X}^* \sim \mathcal{N}_d\left(\mathbf{x} + \frac{h}{2}\boldsymbol{\nabla}\mathbf{log}(\boldsymbol{x}),\ h\mathbf{I}_{d\times d}\right) \tag{3.19}$$

As for the scale parameter in RWMA, the step size h needs to be tuned and the corresponding acceptance probability is

$$\alpha = \min\left\{\frac{h_\psi(\mathbf{Y}, \mathbf{Z}^*)q(\mathbf{Z}|\mathbf{Z}^*)}{h_\psi(\mathbf{Y}, \mathbf{Z})q(\mathbf{Z}^*|\mathbf{Z})}, 1\right\} \tag{3.20}$$

where $q(\cdot|\cdot)$ is given by the proposal in expression (3.19).

For both algorithms, we draw the initial values of $\mathbf{Z}$ from the marginal distribution $P_\psi(\mathbf{Z})$. Pilot runs need to be conducted to decide the convergence of the Markov chains. The obtained MCMC samples are thinned to achieve the desired effective sample size while keeping the storage and computational budget.

Adaptive MCMC

To slightly automate the MCMC tuning step, we adopted an adaptive sampling strategy similar to that used in the R package `PrevMap` (Giorgi and Diggle, 2015) for both RWMA and MALA. In the code the scale parameter is tuned according to a short pilot run using an adaptive MCMC scheme and then the chain starts over and runs with the parameters fixed at the chosen values for the given number of iterations.

Based on the result of Roberts and Rosenthal (2001), the optimal scale σ^2_{rw} for the RWMA should be $\mathcal{O}(N^{-1})$ and the optimal scale σ^2_{ma} (corresponding to the step size h in previous section) for MALA is $\mathcal{O}(N^{-1/3})$, where N is the dimension of the multivariate distribution. These scales can be used as the starting point in the adaptive algorithm. Then at the ith iteration, the scale is tuned to achieve the optimal acceptance rate α^* by using the following update rule based on the stochastic approximation method[2],

$$\sigma_i = \sigma_{i-1} + c_1 i^{-c_2}(\alpha_i - \alpha^*) \tag{3.21}$$

for some constants c_1, c_2 also to be tuned according to the constraints given in section 3.3.3. In practice, users can adapt these values by monitoring the trace plots of σ_i and α_i in a pilot run. The optimal acceptance rate for RWMA is about 0.234 for $N > 1$ and for MALA is about 0.574 (Roberts and Rosenthal, 2001).

[2]More details on the stochastic approximation method are reviewed in section 3.3.3

In our implementation, diagnostic plots include the plot of acceptance rate versus number of iteration, scale parameter versus iteration, ACF plots for each marginals and trace plots for five randomly selected marginals. They are displayed automatically by default setting "`pilot.run = TRUE`". Figure 3.9 shows an example of sampling $P(\mathbf{Z}|\mathbf{y})$ for simulated data on a 20×20 torus using MALA.

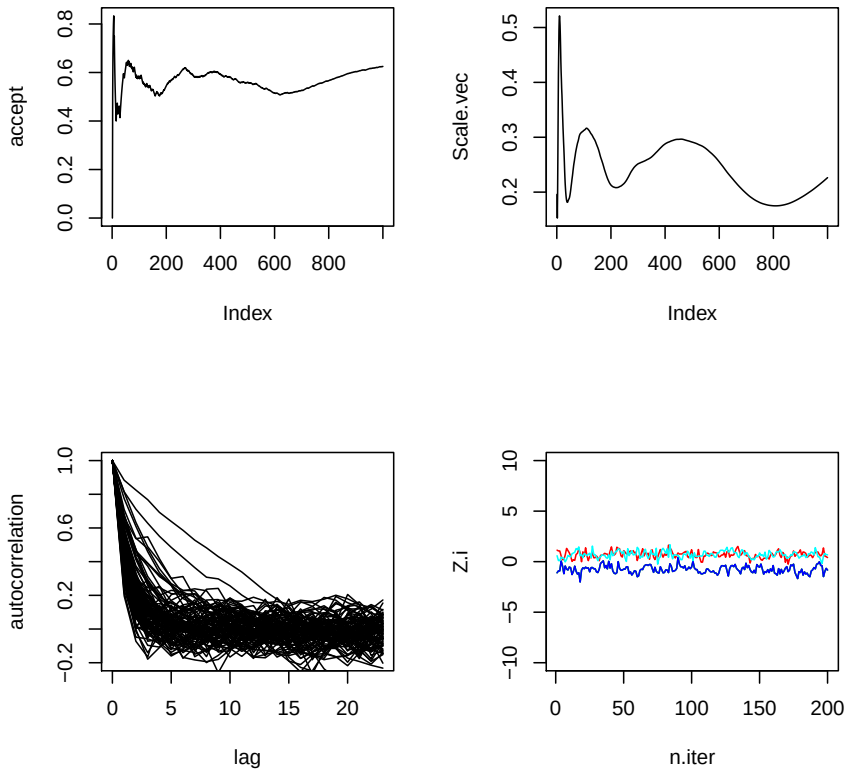


Fig. 3.9 Diagnostic plots for the adaptive MCMC algorithm. Top left: trace plot of the acceptance rate; top right: trace plot of the scale parameter; bottom left: ACF plots of the each posterior marginals; bottom right: trace plot of five randomly selected posterior marginals.

3.3.2 Variance reduction

The performance of the Monte Carlo likelihood methods largely depends on the variance of the importance sampling approximation of the log-likelihood ratio

between $\ell(\boldsymbol{\theta})$ and $\ell(\boldsymbol{\psi})$; therefore we could improve the accuracy of the method by reducing the Monte Carlo variance.

One most obvious preference is to only use the approximation when there is no exact solution or the computing the exact is not feasible with current resources. For example when both the normalizing constant of the likelihood and the integral of the latent field is difficult to compute, it is recommended in Geyer (1994) to use the **Approx. II** of equation (2.6) with the exact value of the normalising constant whenever possible. If both parts are approximated by the Monte Carlo method as in the **Approx III** of equation (2.7), the variances from the two sources adds up and may cover the true signal.

The Monte Carlo likelihood can be seen as a special case of estimating ratios of normalising constants which has always been a challenging computational problem. Using similar notation as for the Monte Carlo likelihood, we denote by $h_t(\mathbf{x})$ the un-normalised density function of $f_t(\mathbf{x}) = h_t(\mathbf{x})/c_t$ for some distribution indexed by t and c_t is the normalising constant. The aim is to approximate the ratio of the normalising constants between two distributions of interest. Without loss of generality, we denote the two distributions of interest by indices 0 and 1 and hence we want to estimate $r = c_1/c_0$.

Apart from the basic techniques such as the Monte Carlo integration and importance sampling, various simulation based methods have been proposed to improve the accuracy of the approximation while keeping the computational cost low, especially when f_1 and f_0 have very little overlap. In the following we review some successful methods that are widely used in the literature under two categories: the first one tries to find a good importance sampler that can "cover" both f_1 and f_0 and the second is to use some density to "connect" samples from f_1 and f_0.

Finding a better importance sampler

As pointted out by Green (1992) in the discussion of Geyer and Thompson (1992), no distribution within the likelihood model can serve as a good importance sampler for the purpose of approximating the likelihood for a wide range of $\boldsymbol{\theta}$. This leads to the search of better importance samplers outside the parametric family given by the likelihood model, among which are the method of *re-weighting mixtures by reverse logistic regression* (Geyer and Thompson, 1992; Geyer and Møller, 1994) and *umbrella sampling* (Chen and Shao, 1997); the former uses a mixture of distributions from the likelihood model with different values of $\boldsymbol{\psi}$ instead of a single one and the latter tries to construct an optimal density to "cover" both ends of $\boldsymbol{\theta}$ and $\boldsymbol{\psi}$.

Re-weighting mixtures by reverse logistic regression In the Monte Carlo likelihood, we use the importance sampler $f_{\boldsymbol{\psi}}$; now instead of using samples from a single distribution, we generate n_j samples from $j = 1, \ldots, m$ distributions f_j of the same parametric form but with j indexing for different values of $\boldsymbol{\psi}$ and use the mixtures

$$f_{\text{mix}} = \sum_{j=1}^{m} \frac{n_j}{n} \frac{1}{c_j} h_j(\mathbf{x}) \tag{3.22}$$

where $n = \sum_{j=1}^{m} n_j$, as the importance distribution. The choice of $n_j, j = 1, \ldots m$ is determined by the computational budget and the user's prior knowledge of the distribution of interest. In practice, most implementations use $n_j = n_0 = n/m$. The unknown normalising constants in equation (3.22) can be estimated up to a constant of proportionality by fitting a reverse logistic regression to the above mixture density with the generated samples $\mathbf{X}_{ij}$. The reverse logistic regression

"forgets" which distribution the $\mathbf{X}_{ij}$ are from and maximise the quasi-likelihood

$$\ell_n(\eta) = \sum_{j=1}^{m_j} \sum_{i=1}^{n_j} \log p_j(\mathbf{X}_{ij}, \eta)$$

where $\eta = (\eta_1, \ldots, \eta_m)^T$ and

$$\eta_j = -\log c_j + \log \frac{n_j}{n}$$

if we write $h_{\text{mix}}(\mathbf{x}) = \sum_{i=1}^{m} h_j(\mathbf{x}) e^{\eta_j}$.

It is shown in Geyer and Møller (1994) that the method provide consistent estimates that are asymptotically normal under some general conditions and it is also stabler than using a single distribution to obtain the MC-MLE when no iteration procedure is used. As in the iterative procedure, the method still requires samples from several different distributions; the sampling procedure might be accelerated by for example, using parallel computing or coupling the Markov chains in the MCMC algorithm. In fact, our iterative procedure can also be adapted to construct a mixture density with the samples from each iteration if likelihood ratio of two far away parameters need to be compared.

Umbrella sampling The method of umbrella sampling was first developed in Torrie and Valleau (1977) and later it is termed as *ratio importance sampling* by Chen and Shao (1997). This method tries to find a middle density $f_{1/2}$ such that it "covers" both ends, then the target ratio can be written as

$$r = \frac{c_1}{c_0} = \frac{c_1/c_{1/2}}{c_0/c_{1/2}} = \frac{E_{1/2}[q_1/q_{1/2}]}{E_{1/2}[q_0/q_{1/2}]} \tag{3.23}$$

By minimising the asymptotic relative mean-square error defined in the paper Chen and Shao (1997), and under the assumption that $\int_{\mathcal{X}} |f_1(\mathbf{x}) - a f_0(\mathbf{x})| \, \mathrm{d}\mathbf{x} > 0, \forall a > 0$,

the optimal $f_{1/2}$ is given by

$$f_{\text{opt}}(\mathbf{X}) = \frac{|f_1(\mathbf{X}) - r f_0(\mathbf{X})|}{\int_{\mathbf{X}} |f_1(\mathbf{x}) - r f_0(\mathbf{x})| \, \mathrm{d}\mathbf{x}} \tag{3.24}$$

which also depends on the ratio r. A two-stage algorithm is aprovided in Chen and Shao (1997) to get the optimal density but sampling from this is usually non-trivial and the optimal density has to change whenever we use a different f_1 or f_0.

Bridge Sampling and path sampling

Similar to using samples from a mixture distribution, bridge sampling and path sampling use samples from both f_1 and f_0 or any densities with parameter along a path indexed by $t \in [0, 1]$ in the parameter space to approximate the ratio. Similar to the spirit of umbrella sampling, bridge sampling and path sampling try to find the optimal bridge or path by minimising the Monte Carlo variance.

Bridge sampling Based on the assumption that if for an arbitrary function $\alpha(\mathbf{x})$ defined on the common support $\Omega_0 \cap \Omega_1$ of the parametric family $f_t(\mathbf{x})$ indexed by $t \in [0, 1]$, we have $0 < \left| \int_{\Omega_0 \cap \Omega_1} \alpha(\mathbf{x}) f_0(\mathbf{x}) f_1(\mathbf{x}) \, \mu(\mathrm{d}\mathbf{x}) \right| < \infty$, then the ratio can be written as

$$r = \frac{c_1}{c_0} = \frac{E_0[h_1(\mathbf{X})\alpha(\mathbf{X})]}{E_1[h_0(\mathbf{X})\alpha(\mathbf{X})]} \tag{3.25}$$

Suppose there is a distribution $p_{1/2}$ between p_0 and p_1 with support $\Omega_0 \cap \Omega_1$, and let $\alpha = h_{1/2}/(h_0 h_1)$, then (3.25) becomes

$$r = \frac{c_1}{c_0} = \frac{c_{1/2}/c_0}{c_{1/2}/c_1} = \frac{E_0[h_{1/2}/h_0]}{E_1[h_{1/2}/h_1]} \tag{3.26}$$

where $f_{1/2}$ serves as a *bridge* between f_0 and f_1 and the optimal bridge is given by the weighted harmonic mean of f_0 and f_1

$$f_{1/2} = (s_1 f_0^{-1} + s_0 f_1^{-1})^{-1} \tag{3.27}$$

where $s_t = n_t/(n_0+n_1), t = 0, 1$ and n_t is the number of samples from f_t. Again the optimal bridge still contains the unknown normalising constants and an iterative procedure is give in Gelman and Meng (1998) to estimate r.

Path sampling The idea of bridge sampling can be extended so that there is more than one "bridge" between f_1 and f_0 if their supports are far apart or completely separate. Suppose between f_1 and f_0 there are $l = 1, 2, \ldots, L-1$ intermediate unnormalized densities denoted by h_l, and between these $L+1$ densities there are $l = 0, 1, \ldots, L$ unnormalized "bridge" densities $h_{l+1/2}$, then apply the identity in (3.25), we have

$$r = \frac{c_1}{c_0} = \prod_{l=0}^{L} \frac{c_{l-1/2}/c_{l-1}}{c_{l-1/2}/c_l} = \prod_{l=1}^{L} \frac{E_{l-1}(h_{l-1/2}(\mathbf{X})/h_{l-1}(\mathbf{X}))}{E_l(h_{l-1/2}(\mathbf{X})/h_l(\mathbf{X}))} \tag{3.28}$$

Here we slightly abuse the notation of the l to represent the index $\boldsymbol{\theta}_l \in [0, 1]$. If we take the limit of $L \to \infty$ in the log of (3.28), it becomes the basic identity of *path sampling* (Gelman and Meng, 1998)

$$\lambda = \log r = \log \frac{c_1}{c_0} = \int_0^1 E_{\boldsymbol{\theta}}[U(\mathbf{X}, \boldsymbol{\theta})] \, \mathrm{d}\boldsymbol{\theta} \tag{3.29}$$

where $U(\mathbf{X}, \boldsymbol{\theta}) = \frac{d}{d\boldsymbol{\theta}} \log q_{\boldsymbol{\theta}}(\mathbf{X})$. If $\boldsymbol{\theta}$ is regarded as a random variable from a prior $p(\boldsymbol{\theta})$, then (3.29) can be interpreted as the expectation of $U(\mathbf{X}, \boldsymbol{\theta})/p(\boldsymbol{\theta})$ over the joint of $(\mathbf{X}, \boldsymbol{\theta})$ and we can write (3.29) as

$$\lambda = E\left[\frac{U(\mathbf{X}, \boldsymbol{\theta})}{p(\boldsymbol{\theta})}\right] \tag{3.30}$$

By minimising the Monte Carlo variance, the optimal prior for $\boldsymbol{\theta}$ is given by

$$p(\boldsymbol{\theta}) = \frac{\sqrt{E_{\boldsymbol{\theta}}[U^2(\mathbf{X}, \boldsymbol{\theta})]}}{\int_0^1 \sqrt{E_{\vartheta}[U^2(\mathbf{X}, \boldsymbol{\vartheta})]}\, \mathrm{d}\boldsymbol{\vartheta}} \tag{3.31}$$

The optimal path is in general difficult to find but some sub-optimal choices are discussed the same paper.

Warp bridge sampling Another way to deal with the situation where p_1 and p_0 are far away or separate from each other is to use *warp bridge sampling* (Meng and Schilling, 2002). It transforms both the densities to increase the overlap. Suppose $T_i, i = 1, 2$ are two one-to-one transformations in $\mathbb{R}^d$ that aim to *warp* p_0 and p_1 into similar shapes and denote $p_i^{(T_i)}(\mathbf{x})$ to be the transformed density and has the density function

$$p_i^{(T_i)}(\mathbf{x}) = \frac{q_i(T_i^{-1}(\mathbf{x}))/|J_i(T_i(\mathbf{x}))|}{c_i} \tag{3.32}$$

where $J_i(T_i(\mathbf{x}))$ is the Jacobian of the transformation T_i. With warped distribution, (3.25) can be used again and by transforming the samples back to those from the original distribution, the ratio can be estimated by using

$$\hat{r}_{\alpha}^{(T_1,T_2)} = \frac{\frac{1}{n_2}\sum_{j=1}^{n_2} \alpha(T_2(\mathbf{x}_{2j}))q_1(T_1^{-1}(T_2(\mathbf{x}_{2j})))\,/|J_1(T_2(\mathbf{x}_{2j}))|}{\frac{1}{n_1}\sum_{j=1}^{n_1} \alpha(T_1(\mathbf{x}_{1j}))q_2(T_2^{-1}(T_1(\mathbf{x}_{1j})))\,/|J_2(T_1(\mathbf{x}_{1j}))|} \tag{3.33}$$

General schemes for finding the suitable transformation are described in Meng and Schilling (2002) by the three stages of "warping" the "centres", "spread" and skewness, which are usually represented by matching the corresponding sample moments.

The above methods are all better at approximating the ratio when f_1 and f_0 are far away from each other but they all bring the costs of finding some optimal distribution that also involves the ratio r of interest and thus, either some iterative

procedure or sub-optimal choice has to be used in practice. In this thesis, we simply use the initial form of the Monte Carlo likelihood with f_ψ from the same likelihood family as the importance sampler. For purposes other than finding the MC-MLE, the methods reviewed above might have better performance.

3.3.3 Other optimisation algorithms

In a more general context, the Monte Carlo likelihood methods can be viewed as a special problem of optimising an unknown function $f(\boldsymbol{\theta}), \boldsymbol{\theta} \in \boldsymbol{\theta}$ when a noisy version $F(\boldsymbol{\theta}, \mathbf{X})$ is available and has the property that

$$\min_{\boldsymbol{\theta}\in\boldsymbol{\theta}} f(\boldsymbol{\theta}) = \min_{\boldsymbol{\theta}\in\boldsymbol{\theta}} \mathbb{E}\left[F(\boldsymbol{\theta}, \mathbf{X})\right] \tag{3.34}$$

where $\mathbf{X}$ is some random variable from a given probability measure μ. This can usually be converted to solving the equation

$$g(\boldsymbol{\theta}) = \mathbb{E}\left[G(\boldsymbol{\theta}, \mathbf{X})\right] = 0 \tag{3.35}$$

where $g(\boldsymbol{\theta})$ and $G(\boldsymbol{\theta}, \mathbf{X})$ are the corresponding gradient functions.

Stochastic Approximation methods

One of the prototypical algorithm is the Robbins-Monro stochastic approximation algorithm (SA) introduced by Robbins and Monro (1951). Assume that the above equation has unique root at $\boldsymbol{\theta} = \boldsymbol{\theta}^*$, the SA algorithm generates a sequence of $\{\boldsymbol{\theta}_n\}$ by

$$\boldsymbol{\theta}_{n+1} = \boldsymbol{\theta}_n - \gamma_n G(\boldsymbol{\theta}_n, \mathbf{x}_n) \tag{3.36}$$

for positive step size $\{\gamma_n\}$. Then for uniformly bounded $f(\boldsymbol{\theta})$ with non-decreasing $g(\boldsymbol{\theta})$ and positive $g'(\boldsymbol{\theta})$, $\boldsymbol{\theta}_n \xrightarrow{P} \boldsymbol{\theta}^*$ if the sequence of γ_n satisfies

$$\sum_{n=0}^{\infty} \gamma_n = \infty, \qquad \sum_{n=0}^{\infty} \gamma_n^2 < \infty.$$

Under the assumption that $f(\boldsymbol{\theta})$ is twice continuously differentiable and strongly convex, Chung (1954) and Sacks (1958) proved that the step size γ_n can be chosen to achieve the optimal convergence rate $\mathcal{O}(1/n)$.

Based on the SA algorithm, several algorithms have been proposed to deal with problems in various contexts. The stochastic gradient descent (SGD) algorithm in the machine learning literature (Bottou, 1998) uses a random sub-sample of the entire dataset to approximate the gradient of the empirical loss function for large scale problem; similarly in the Bayesian learning of network parameters Welling and Teh (2011) proposed the method of the stochastic gradient Langevin dynamics (SGLD) which uses the stochastic gradient in the proposal of the MALA.

Another widely used algorithm for maximum-likelihood estimation is the EM algorithm (Dempster et al., 1977) which iterates between the following two step until converges:

1. **E-step**: Compute $\boldsymbol{Q}(\boldsymbol{\theta}|\boldsymbol{\theta}_n) = \mathbb{E}_{\boldsymbol{\theta}_n}\left[f(\boldsymbol{\theta}; \mathbf{X})\right]$

2. **M-step**: Update $\boldsymbol{\theta}_n$ by $\boldsymbol{\theta}_{n+1} = \max_{\boldsymbol{\theta}\in\boldsymbol{\theta}} \boldsymbol{Q}(\boldsymbol{\theta}|\boldsymbol{\theta}_n)$.

Usually $\boldsymbol{Q}(\boldsymbol{\theta}|\boldsymbol{\theta}_n)$ may appear in closed form and is easy to compute but when the expectation is intractable, one possible solution is to use simulation based approxiamtion. The Monte Carlo EM algorithm (MCEM, Wei and Tanner (1990)) was introduced to replace the expectation by a Monte Carlo integration. Later on similar algorithms to the MCEM have been proposed, including the stochastic EM

algorithm (SEM, Celeux and Diebolt (1992)) which use

$$\tilde{\boldsymbol{Q}}_n(\boldsymbol{\theta}|\boldsymbol{\theta}_n) = f(\boldsymbol{\theta}; \mathbf{x})$$

with a single sample $\mathbf{x}$ from $\mu_{\boldsymbol{\theta}_n}$ in the "E-step". It can be viewed as a special case of the stochastic approximation EM algorithm (SAEM, Delyon et al. (1999)) where the expectation is approximated according to the SA algorithm

$$\hat{\boldsymbol{Q}}_n(\boldsymbol{\theta}|\boldsymbol{\theta}_n) = \hat{\boldsymbol{Q}}_{n-1}(\boldsymbol{\theta}) + \gamma_n \left(\frac{1}{m(n)} \sum_{j=1}^{m(n)} f(\boldsymbol{\theta}; \mathbf{x}_n^{(j)} - \hat{\boldsymbol{Q}}_{n-1}(\boldsymbol{\theta})) \right)$$

Convergence results are established in the corresponding papers and for SAEM it also depends on the choice of step size γ_n and the Monte Carlo sample size $m(n)$.

Design-based methods

Instead of using the gradient information only, some design-based methods try to build an emulator (or surrogate function) by using evaluations at a small set of design points then use the emulator to locate the optimum. The design points are chosen to achieve variance optimality criteria and the accuracy of the emulator can be improved by sequentially adding more design points. Such methods are often studied under the topic of computer design and response surface estimation. Santner et al. (2003), Khuri and Mukhopadhyay (2010) and Box and Draper (2007) provide comprehensive reviews and in Chapter 4 we combine the response surface methodology with the Monte Carlo likelihood method; more details are included therein.

3.4 Conclusion

In this chapter, we proposed an iterative procedure for maximising the Monte Carlo likelihood. The procedure is fully automatic and it tackles problems, including choosing the initial values, the update step size and convergence criterion which have not been discussed or solved in existing literature.

We also discussed in details about the implementation of the procedure which is given by the function `OptimMCL` in the package `mclcar`. Alternative methods of approximating and maximising the likelihood are briefly reviewed at the end to provide as improvement or generalisation of the Monte Carlo likelihood method whenever suitable. However these methods are only discussed for completeness but not used in our implementation since for the CAR models, they usually achieve relatively small improvement in accuracy with more computational cost.

Chapter 4

Response surface method

The Monte Carlo likelihood can be maximised directly but the estimate is only reliable when it is near the importance sampler value. A direct optimization may be infeasible when the starting point is too far away from the target and methods based on the stochastic approximation algorithm can be inefficient as it requires sampling from a new importance distribution in each iteration to get only the gradient information.

In this chapter, we introduce the *Response surface methodology (RSM)*[1] to tackle the above problems in optimising the Monte Carlo likelihood. In the following sections, we provide a brief review of the general methodology, address several issues in the implementation regarding to the Monte Carlo likelihood and provide our current solutions correspondingly. Finally we demonstrate the entire procedure with two examples using simulated CAR data on the torus.

The examples show that the RSM provides an accurate estimate of the MLE and is more efficient when the initial importance distribution is far different from the target, compared to the result from direct optimization of the Monte Carlo likelihood function. Apart from the point estimate of the MLE, the RSM also

[1] In the following text, we also use the abbreviation RSM to stand for the response surface method that we used to optimise the Monte Carlo likelihood and readers should be able to differentiate the two from the context.

reconstructs the likelihood surface around the MLE and thus provides information about the confidence level sets.

4.1 Methodology

The response surface methodology provides a series of statistical techniques for building and exploiting empirical models by conducting only an economic numbers of experimental runs. It was originally developed by Box and Wilson (1951) to find the optimal condition for chemical reactions where the exact functional relationship between the input variables and the response is unknown. The method explores the functional relationship locally by fitting low-order polynomials to the measurements from a few evaluations at given design points. Based on the gradient information of the fitted models, the system gradually moves along the *steepest ascent path* towards the optimum point. As the system gets close enough to the optimum point, the stationary point of fitted second-order polynomials is usually a good estimate of the optimum.

Details about the methodology can be found in works such as Box and Draper (2007), and recent reviews can be found in Myers et al. (2009), Khuri and Mukhopadhyay (2010), Khuri (2006), and Lenth (2009). In the following we provide a brief description of the methodology by using a flow chart in Figure 4.1.

At the beginning of the procedure, the input variables are selected and transformed based on the current understanding of the problem; then the initial centre point $\boldsymbol{\xi}_c^{(0)}$ and the corresponding experimental region are chosen according to prior knowledge of the response. Evaluations are then taken at the design points to fit a first-order model. The next centre point is found by exploring along the steepest ascent path given by the gradient of the fitted model. This is iterated until the first-order model can no longer fit the data well, at which stage the model is updated by adding second-order terms. When the system is close enough to

the optimum point, the fitted quadratic model usually provides a stationary point which can be output as the optimum point; otherwise *ridge analysis* (Box and Draper, 2007, chap.9-12) can be used to locate the next centre point.

The central composite design (CCD) is used most often in generating the design points for running the experiments. It has the useful property that the design points for fitting the first and second models can be assembled sequentially. The design consists of three groups of points: the central points for estimating the pure error; the cube points lying on the corner of the cube (or hypercube) of the experimental region for estimating first order effect; and the star points on each coordinate for estimating the quadratic effect. Figure 4.2 shows the design points for two and three input variables. The distance of the star point from the centre are chosen to make the design have additional properties such as rotatability and orthogonally blocking if possible. Details on the design aspect and a table of second-order composite designs can be found in Box and Draper (2007).

The RSM can also be used when optimising intractable functions which can be approximated by computer experiments. In this thesis on parameter estimation of the CAR models, we are more interested in parameters in the covariance matrix, so our input variables are ρ and σ^2. The initial value of th centre point is recommended to be close to the maximum when prior information is available, otherwise we can use any value within the valid parameter range. In the next section, we first discuss how to find a suitable transformation of the inputs and define the experimental region; then we present how to adapt the RSM to optimize the Monte Carlo likelihood of the CAR models.

4.2 Implementation

In this section, we try to resolve three major issues in implementing the RSM on the Monte Carlo likelihood. In the first subsection we deal with the error of

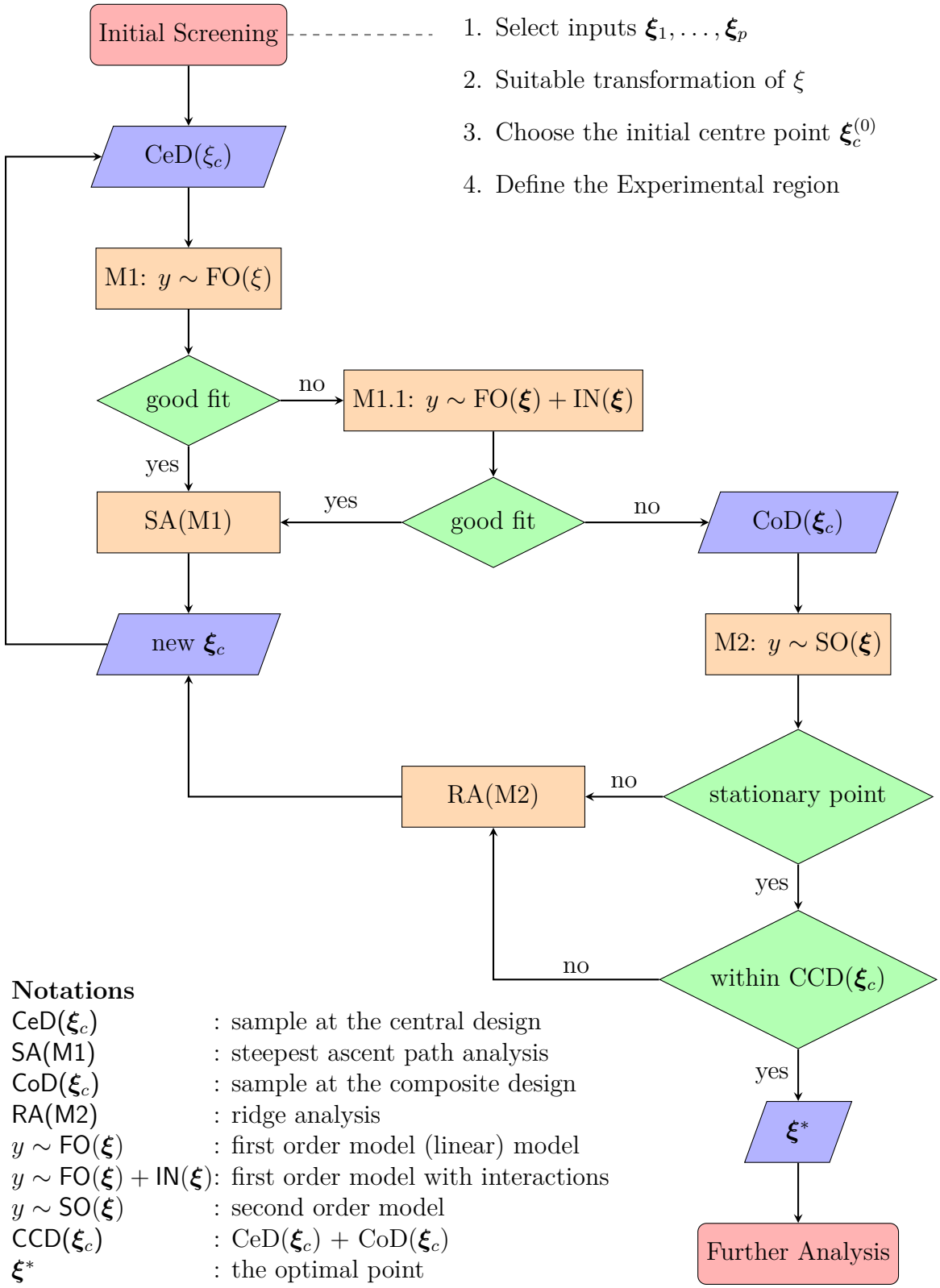


Fig. 4.1 The RSM procedure
The RSM procedure for finding the optimum

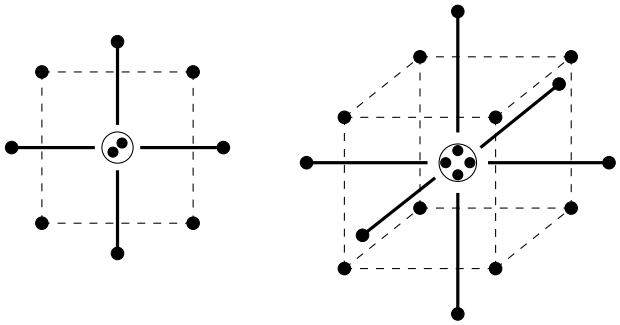


Fig. 4.2 Central composite designs for two and three input variables.

the Monte Carlo likelihood that affects the design region and the fitting of the model; then we try to reduce the dimensionality of the input variables and finally we discuss the the updating and convergence criteria.

4.2.1 Design region and model fitting

To generate the design points for evaluation, we need to define the design region within which the low-order polynomial would be sufficient to approximate the target function and the target function should vary equally in scales for each unit change of the input variables. For the Monte Carlo likelihood function, there is an additional constraint that the design region should be within or largely within the finite variance region of the Monte Carlo approximation.

In the following, we provide a general rule of defining the design region for the Monte Carlo likelihood of the CAR models based on our previous analysis of the finite variance region in section 2.3.1. Although the region is derived based on results of the direct CAR model on the torus, it should suffice also for the latent CAR models and other spatial structures as they share similar finite variance properties. We consider only an RSM design for ρ and σ^2, since given the covariance matrix, the linear coefficient $\boldsymbol{\beta}$ is relatively easy to estimate.

First we transform the parameter to a space that can easily accommodate the finite variance inequality (2.19)

$$\begin{cases} 0 < \dfrac{\sigma_{\boldsymbol{\theta}}^2}{\sigma_{\boldsymbol{\psi}}^2} < 2; \\ 1/\lambda_1 < \dfrac{2\rho_{\boldsymbol{\theta}}\sigma_{\boldsymbol{\psi}}^2 - \rho_{\boldsymbol{\psi}}\sigma_{\boldsymbol{\theta}}^2}{2\sigma_{\boldsymbol{\psi}}^2 - \sigma_{\boldsymbol{\theta}}^2} < 1/\lambda_n \end{cases}$$

Hence, a natural way to transform the input variables would be

$$\begin{cases} t_s = \log \dfrac{\sigma_{\boldsymbol{\theta}}^2}{\sigma_{\boldsymbol{\psi}}^2}; \\ t_r = \rho_{\boldsymbol{\theta}} - \rho_{\boldsymbol{\psi}} \end{cases}$$

where t_s reflects the difference between the variances as a ratio and is independent of ρ and t_r shows the distance between the values of ρ when assuming $\sigma_{\boldsymbol{\theta}}^2 = \sigma_{\boldsymbol{\psi}}^2$. Both t_s and t_r are centred at zero when $\boldsymbol{\theta} = \boldsymbol{\psi}$.

Next we find a suitable amount of change in t_s and t_r for the design region, which should be large enough for efficient exploration of surface but small enough that the magnitude of the Monte Carlo variance does not affect too much the value of the approximated function. Usually we are interested in the region within some critical value k for the log-likelihood ratio $\ell(\boldsymbol{\psi}, \boldsymbol{\theta})$, therefore for such $\boldsymbol{\theta}$ that $\mathbb{E}(\hat{\ell}(\boldsymbol{\psi}, \boldsymbol{\theta})) = \ell(\boldsymbol{\psi}, \boldsymbol{\theta}) = k$, we would like to have

$$\mathbb{P}(|\hat{\ell}(\boldsymbol{\psi}, \boldsymbol{\theta}) - k| > \epsilon) < \alpha \tag{4.1}$$

for some small value $\epsilon > 0$ and critical level α. By using the asymptotic normality in (2.21) we have the following condition on the Monte Carlo variance

$$Z_{1-\alpha/2} s.d.(\hat{\ell}(\boldsymbol{\theta}, \boldsymbol{\psi})) = Z_{1-\alpha/2} \sqrt{\frac{V(\boldsymbol{\psi}, \boldsymbol{\theta})}{sI^2(\boldsymbol{\psi}, \boldsymbol{\theta})}} < \epsilon \tag{4.2}$$

where $Z_{1-\alpha/2}$ is the α quantile of a standard normal distribution. We can choose α to be commonly used significance level such as $\alpha = 0.05$. The value of ϵ can be specified by user to change the size of the experimental region. Since we are less interested in the region where the likelihood-ratio is smaller than zero, a reasonable choice is $\epsilon = k$ which gives

$$\mathbb{P}\left(\hat{\ell}(\boldsymbol{\psi}, \boldsymbol{\theta}) < 0 \cup \hat{\ell}(\boldsymbol{\psi}, \boldsymbol{\theta}) > 2k\right) < \alpha$$

Thus, for the direct CAR model on the torus the above reduces to

$$\frac{|\boldsymbol{\Sigma}_{\boldsymbol{\psi}}||\boldsymbol{\Sigma}^*|}{|\boldsymbol{\Sigma}_{\boldsymbol{\theta}}||\boldsymbol{\Sigma}_{\boldsymbol{\theta}}|} < \left(\left(\frac{k}{2}\right)^2 s + 1\right)^2 \tag{4.3}$$

where $\boldsymbol{\Sigma}_{\boldsymbol{\psi}}, \boldsymbol{\Sigma}_{\boldsymbol{\theta}}$ and $\boldsymbol{\Sigma}^*$ are defined as in Section 2.3.1.

The boundary of the parameter space given by this inequality is still non-trivial to specify; hence we simplify it by finding the limits of ρ and σ^2 given the other being held at the value of $\boldsymbol{\psi}$. For $\rho_{\boldsymbol{\theta}} = \rho_{\boldsymbol{\psi}}$ we have

$$\left(\frac{\sigma^2_{\boldsymbol{\psi}}\sigma^{*2}}{\sigma^2_{\boldsymbol{\theta}}\sigma^2_{\boldsymbol{\theta}}}\right)^n = [\exp(t_s)(2 - \exp(t_s))]^{-n} < \left(\left(\frac{k}{2}\right)^2 s + 1\right)^2$$

which gives

$$|t_s| < \sqrt{1 - \left(\left(\frac{k}{2}\right)^2 s + 1\right)^{-2/n}} \tag{4.4}$$

Similarly for $\sigma^2_{\boldsymbol{\psi}} = \sigma^2_{\boldsymbol{\theta}}$, we have

$$\prod \frac{(1 - \rho_{\boldsymbol{\theta}}\lambda_i)^2}{(1 - \rho_{\boldsymbol{\psi}}\lambda_i)(1 - 2(\rho_{\boldsymbol{\theta}} - \rho_{\boldsymbol{\psi}})\lambda_i)} = \prod \frac{(1 - \rho_{\boldsymbol{\theta}}\lambda_i)^2}{(1 - \rho_{\boldsymbol{\theta}}\lambda_i)^2 - (\lambda_i t_r)^2} < \left(\left(\frac{k}{2}\right)^2 s + 1\right)^2 \tag{4.5}$$

A sufficient condition is to require that

$$\max\left\{\frac{(1-\rho_{\boldsymbol{\theta}}\lambda_i)^2}{(1-\rho_{\boldsymbol{\theta}}\lambda_i)^2-(\lambda_i t_r)^2}\right\} \leq \left(\left(\frac{k}{2}\right)^2 s+1\right)^{\frac{2}{n}}$$

which is equivalent to

$$\max\left|\frac{t_r}{1/\lambda_i-\rho_{\boldsymbol{\theta}}}\right| \leq \sqrt{1-\left(\left(\frac{k}{2}\right)^2 s+1\right)^{-2/n}}$$

Since $\rho_{\boldsymbol{\theta}} \in (1/\lambda_1, 1/\lambda_n)$, we have the maximum value on the left-hand side equal to $\frac{|t_r|}{\rho_{\boldsymbol{\theta}}-1/\lambda_1}$ or $\frac{|t_r|}{1/\lambda_n-\rho_{\boldsymbol{\theta}}}$, depending on whether $\rho_{\boldsymbol{\theta}}$ is closer to $1/\lambda_1$ or $1/\lambda_n$. Rewriting $\rho_{\boldsymbol{\theta}}$ as $\rho_{\boldsymbol{\psi}} - t_r$, we have the following solutions for the range of t_r

$$\begin{cases} \dfrac{b(1/\lambda_1-\rho_{\boldsymbol{\psi}})}{1+b} < t_r < -\dfrac{b(1/\lambda_1-\rho_{\boldsymbol{\psi}})}{1-b}, & |\rho_{\boldsymbol{\theta}}-\lambda_1| < |\rho_{\boldsymbol{\theta}}-\lambda_n|; \\ -\dfrac{b(1/\lambda_n-\rho_{\boldsymbol{\psi}})}{1-b} < t_r < \dfrac{b(1/\lambda_n-\rho_{\boldsymbol{\psi}})}{1+b}, & \text{otherwise.} \end{cases} \tag{4.6}$$

where $b = \sqrt{1-\left(\left(\frac{k}{2}\right)^2 s+1\right)^{-2/n}}$.

In practice, this bound will be too tight for t_r as most of the terms in the product of (4.5) above are close to 1 except for a few that have $1/\lambda_i$ close enough to $\rho_{\boldsymbol{\theta}}$; therefore, we only need to consider a small fraction, say $\gamma = 0.2$, of the entire n terms that have $\left|\frac{t_r}{1/\lambda_i-\rho_{\boldsymbol{\theta}}}\right| > 1$. Now the maximum range of t_r without the finite variance constraint would be $\rho_{\boldsymbol{\psi}} - 1/\lambda_1$ or $1/\lambda_n - \rho_{\boldsymbol{\psi}}$, so for the terms within the fraction we have the following sufficient conditions on the λ_i

$$\begin{cases} \lambda_1 < \lambda_i < 1/(2/\lambda_1-\rho_{\boldsymbol{\psi}}), & |\rho_{\boldsymbol{\theta}}-\lambda_1| < |\rho_{\boldsymbol{\theta}}-\lambda_n|; \\ 1/(2/\lambda_n-\rho_{\boldsymbol{\psi}}) < \lambda_i < \lambda_n, & \text{otherwise.} \end{cases}$$

In most applications, the spatial weight matrix can be rearranged to be approximated by a Toeplitz matrix, of which the eigenvalues are asymptotically uniformly

distributed on (λ_1, λ_n) as the size of the matrix $n \to \infty$; therefore, the fraction γ can be estimated as

$$\gamma = \begin{cases} \dfrac{-\lambda_1(1-\rho_{\boldsymbol{\psi}}\lambda_1)}{(2-\rho_{\boldsymbol{\psi}}\lambda_1)(\lambda_n-\lambda_1)} & |\rho_{\boldsymbol{\theta}}-\lambda_1| < |\rho_{\boldsymbol{\theta}}-\lambda_n|; \\ \dfrac{\lambda_n(1-\rho_{\boldsymbol{\psi}}\lambda_n)}{(2-\rho_{\boldsymbol{\psi}}\lambda_n)(\lambda_n-\lambda_1)} & \text{otherwise.} \end{cases}$$

With the above fraction γ, the constant b of the sufficient condition in the inequality (4.6) becomes $b = \sqrt{1-\left(\left(\frac{k}{2}\right)^2 s+1\right)^{-\frac{2}{\gamma n}}}$.

Example of design region

As an illustration, we show this design region for the example in 2.3.1. In this example, we have $n = 400$ observations on the 20×20 torus, and use $s = 500$ Monte Carlo samples. The importance sampler value $\boldsymbol{\psi} = (0.1072, 0.9052)$ is chosen to be the maximum pseudo likelihood estimate. Table 4.1 shows the range of parameters for two different choices of k.

Table 4.1 Parameter limits of the design region

	$k=-3$		$k=-12$	
t_s	0.186		0.219	
t_r	−0.103	0.042	−0.135	0.047
$\sigma^2_{\boldsymbol{\theta}}$	0.752	1.090	0.727	1.126
ρ	0.004	0.149	−0.028	0.154

Figure 4.3 shows the fitted second order model using the above two design regions overlaid on the Monte Carlo likelihood function in the left and right panel respectively. The red cross is the importance sampler value and the design centre is set to be the geometric centre of the box. This allows the repeated sampling at the centre point for estimating the pure error term by using only a fraction of the entire Monte Carlo samples in one evaluation.

Comparing the plots, the rsm approximations are similar to each other although the one with smaller design region captures the shape of the Monte Carlo likelihood

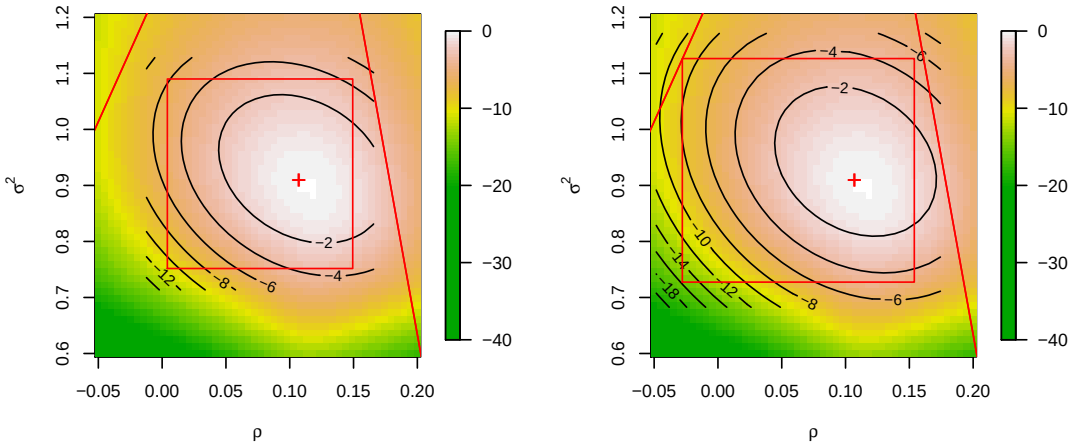


Fig. 4.3 Comparing the fitted RSM of different design regions: the left panel shows the smaller design region given by using $k = -3$ in formulae (4.4) and (4.6) and the right panel shows the larger design region with $k = -12$.

better than the other. Also it can be seen from the plots that the design regions are generally smaller than the region of interest in practice. We can increase the design area by 10% as long as it stays within the parameter space.

Several other transformations that accommodate the variance constraint have been tried but none appears as efficient as the one above.

4.2.2 Lack of fit and variance estimate

The RSM usually assumes that the pure error from the evaluation is constant all over the parameter space which is not true for Monte Carlo likelihood. An obvious remedy is to use weighted least squares in fitting the model with weights set to be inverse of the variance estimate. As shown in section 2.3, for each evaluation of the Monte Carlo likelihood, we can also obtain a variance estimate which is reasonably accurate within the region around the importance sampler value; when the Monte Carlo approximation is substantially different from zero, it tends to

under-estimate the exact variance. Hence, within a suitable design region, it can be used to improve the goodness of fit of the rsm model.

Continuing the previous example, we show in Figure 4.4 the variance estimate within the smaller design region compared to the exact variance. The variance estimate is quite accurate within the design region but it under-estimates the true variance by a factor of 10 near the boundary of the design region. This may explain why the fitted response is very close to the target function but still appears to lack fit. When judging the fitting of the rsm for the Monte Carlo likelihood, we should not pay too much attention to the classical lack of fit test in the RSM framework as the variance of the target function can not always be accurately estimated at any given design point.

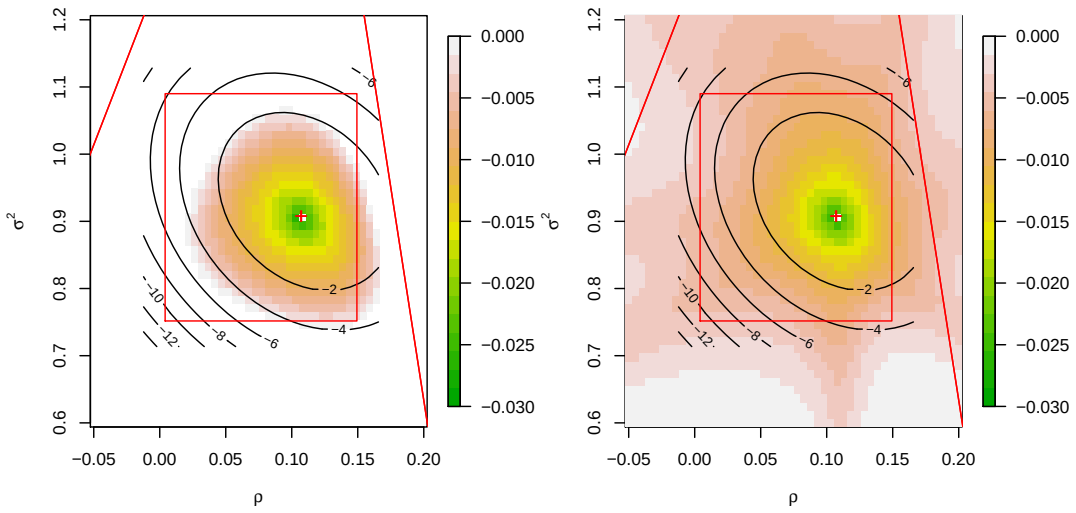


Fig. 4.4 The Monte Carlo variances within the smaller design region as in the left panel of Figure 4.3. The left panel show the contour image of the exact asymptotic variance of the Monte Carlo likelihood and the right panel shows the estimated variance from the Monte Carlo samples with sample size equal to 500. The contours are from the fitted second order model by the response surface method.

4.2.3 Profiling

One limitation of RSM is that the central composite design generates more design points as the dimension of the inputs increases. This will not only make the total cost of evaluation more expensive but also the response difficult to visualise. The number of inputs can be reduced by controlling variables as often done in real experiments; when optimising a function, nuisance parameters can be profiled out or conditioned on a set of fixed values. In our application of the Monte Carlo likelihood of the CAR model, we are interested in the ρ and σ^2 that specify the spatial correlations and as discussed in section 2.3.1, the estimates of the linear coefficients $\boldsymbol{\beta}$ in the GLM do not change too much for a small change in the covariance structure, so we can focus on the profile likelihood $\ell_{\hat{\boldsymbol{\beta}}}(\rho, \sigma^2)$ or the conditional likelihood $\ell(\rho, \sigma, \boldsymbol{\beta} = \boldsymbol{\beta}_{\boldsymbol{\psi}})$ instead.

The Monte Carlo profile likelihood $\hat{\ell}_{\hat{\boldsymbol{\beta}}}(\rho, \sigma^2)$ can be obtained by solving the non-linear system

$$\frac{\partial \hat{\ell}(\rho, \sigma, \boldsymbol{\beta})}{\partial \boldsymbol{\beta}} = \mathbf{0}$$

where the left hand side is readily available from the Monte Carlo approximation.

The above non-linear system can be solved by standard numerical methods such as the Newton-Raphson's method; however it should be noted that iterations must be terminated early before the solution get too far away from the importance sampler value $\boldsymbol{\beta}_{\boldsymbol{\psi}}$. There are two reasons for the early stopping of the algorithm. Firstly when the data size increases each iteration can be computationally expensive; and secondly since the system itself incurs Monte Carlo errors that increases with the distance between $\boldsymbol{\beta}$ and $\boldsymbol{\beta}_{\boldsymbol{\psi}}$, a very "accurate" solution may not be trustworthy if the variance at that point is too big. In practice, we simply set the tolerance of the function value from zero to be 1 and the maximum number of iterations to be

2. In our experience, such settings are usually sufficient for the algorithm to find a reasonable solution.

4.2.4 Update and convergence criteria

With the fitted response surface, we can either move the system to a region closer to the optimum point or output the final result if the procedure appears to converge. In the following, we discuss how to choose the step size for updating the system and set the convergence criteria for the final output.

Step size of the steepest ascent

For a first-order model, the steepest ascent analysis explores a path orthogonal to the fitted plane. For a second-order model, the ridge analysis explores along the *canonical path*. Details of finding these paths can be found in works such as Box and Draper (2007). The exploration simply takes a few more evaluations along the path and choose a step size which takes it to the point showing the biggest improvement. The points to be evaluated are often chosen to be equally spaced along the path with distance equal to half or one unit of the coded data.

In the traditional context of RSM where the evaluation of the target function is assumed to have constant error, the exploration along the steepest ascent path should always provide the biggest improvement at a point within a finite distance from the current centre; however, for the Monte Carlo likelihood function, the steepest ascent analysis may suggest a point that is infinitely far away from current centre due to the increasing Monte Carlo error; in such a situation we should set a maximum update step size within which the steepest-ascent analysis is still to be trusted.

Based on our previous analysis of the Monte Carlo variance and its estimate, a readily available maximum update step size is the distance to the point at which

the estimated Monte Carlo variance is no longer growing exponentially fast when $\boldsymbol{\theta}$ moves away from the importance sampler value, as shown in the example of Figure 2.5. In our implementation when the steepest path does not have a improvement point within finite distance, we plot the log of variance estimate $\log(\mathrm{Var}(\hat{\ell}))$, along the steepest ascent path and choose the first point at which $\log(\mathrm{Var}(\hat{\ell}))$ begins to decrease as the next design centre.

Convergence criteria

The stationary point of a fitted second-order response model can be regarded as the optimum point for the current iteration; however, this point may be still far away from the target point especially when it is outside the design region. Once a stationary point is found within the design region, we can output it as the optimal point, yet it still might carry substantial Monte Carlo variance. Therefore, we use the following convergence criteria to decide the final iteration.

After the stationary point is first found to be within the design region, we repeat the experiment with the same design and an increased number of Monte Carlo samples, say double the size. If the new stationary point is still within the design region, we conclude the algorithm has converged for current Monte Carlo accuracy and RSM model. The stationary point can be used as the final output.

If more accuracy is required for the RSM model, we can decrease the design region so that the RSM provides a better fit for the Monte Carlo approximation. This can be useful when we are interested in the contour shape of the likelihood around the MLE.

4.3 Examples

This section presents two examples of using the RSM to maximise the Monte Carlo likelihood. The first example uses a poor starting point to show the entire evolution

of the response and the second example starts from an estimate close to the target point. Both examples show that the RSM gives a robust and accurate estimate of the optimum point compared to the direct optimisation of the Monte Carlo likelihood, especially when the MLE is near the boundary of the parameter space. The R package `rsm` (Lenth, 2009) is used for generating the designs, fitting and analysing the response surface model.

4.3.1 Direct CAR model

In this example, we use data simulated from a direct CAR model on a 20×20 torus. The exact likelihood function and MLE is available in this setting so we can compare the estimate with the exact values. Table 4.2 shows the true value we used to simulate the data, the exact MLE and approximated estimates. In practice, we can use the MPLE as the initial importance sampler value; here we used $\boldsymbol{\psi}_0 = (-0.202, 1.492)$, which is far away from the maximum, to demonstrate the evolution of the fitted response. All the estimates are similar to each other and close to the true parameter value used to simulate the data.

Table 4.2 Parameter values and esitamtes

	true value	exact MLE	MPLE	$\boldsymbol{\psi}_0$	MC.MLE	RSM.MLE
ρ	0.1200	0.1373	0.1288	-0.2020	0.1382	0.1334
σ^2	1.0000	0.8982	0.9033	1.4920	0.8951	0.9192

In the table, the MC.MLE denotes the parameters estimated by the iterative procedure of finding the MC-MLE described in section 3.2 and RSM.MLE the parameters estimated by the response surface method. Both algorithms started from the poor $\boldsymbol{\psi}_0$. Note that if using a direct optimisation without considering the Monte Carlo variance, the algorithm did not converge within 50 iterations and the estimates kept oscillating between the boundaries of the parameter space; but with the MPLE as the starting point, the algorithm usually quickly converged in less than 3 iterations. The procedure for MC.MLE converged in nine iterations while

the response surface method found the optimal region at iteration 3 and converged within five iterations. For both algorithms, one additional iteration with doubled Monte Carlo samples was done after the convergence to improve the accuracy and the results in the table are from this final iteration. The Monte Carlo MLE obtained by direct optimization can be more precise than the RSM based estimate when the initial value of $\boldsymbol{\psi}_0$ is close enough to the optimum but the method is less efficient when we do not have a good enough starting point; this is also shown by simulation results in the end of this section.

In the above experiments, each evaluation at the central point uses a subset of the entire Monte Carlo samples for estimating the "pure error" while at the cube and star points all Monte Carlo samples are used. The subsets used by the central points are of equal size and form a random partition of the entire Monte Carlo samples. In the following, we demonstrate the use of response surface method in Figure 4.5.

Figure 4.5 includes the results of the first three iterations and the steepest ascent analysis plots below respectively in each column. Initially, we set the critical value of the log-likelihood ratio k to be -6 for a larger design region. At this stage, the response is dominated by the gradient information near the importance sampler point and, within the design region the fitted surface is very close to the the exact value shown as the background image.

The middle row of the figure show plots of exploration along the steepest ascent path and the first two plots indicate that the optimum point is infinitely far away from current design centre. However, these plots can not be trusted due to the increasing Monte Carlo errors as shown in the bottom row where we plotted the estimated Monte Carlo variance. According to previous analysis, the Monte Carlo variance should grow exponentially fast in the distance from the importance sampler value if the approximation were accurate enough; therefore we should expect to see straight lines within the trustworthy distance in the variance plots.

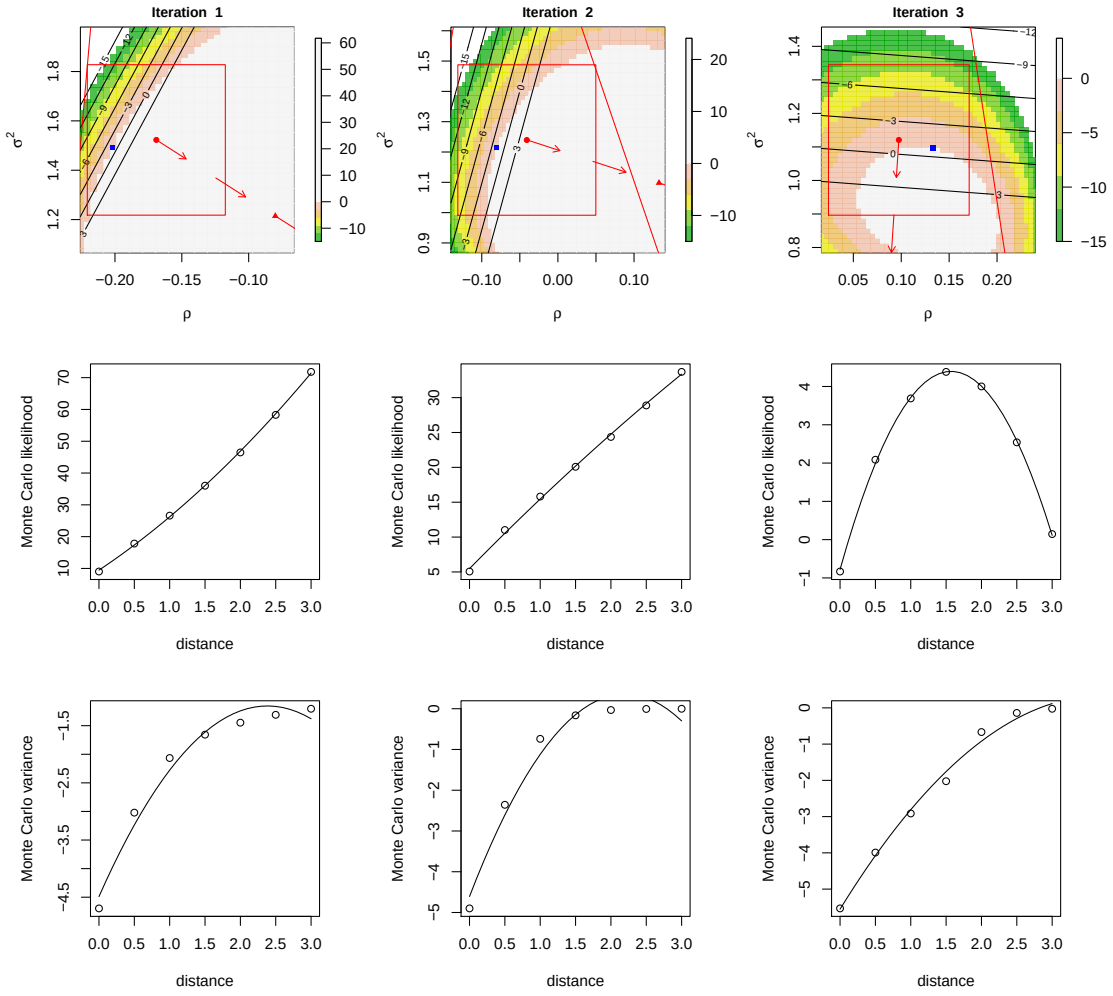


Fig. 4.5 Evolutions of the response surfaces for a direct CAR model on the torus: Iteration 1 to 3. Top panels show contours of the fitted response surfaces superimposed on the images of the exact likelihood. The red rectangle defines the design region and the two red lines show part of the finite variance region. The blue square point is the importance sampler value. The red point is the design centre, starting from which the red arrow shows the steepest ascent path. The red triangle on the path is the next importance sampler value. The middle panels are the evaluations along the steepest path and the bottom panels show the log variance estimates along the steepest path. Details are in the text.

For the first two plots in the bottom row, the estimated variances stop growing fast enough around 1 to 1.5 units away from the design centre, which means the approximation is no longer accurate beyond these points. By fitting a quadratic curve to the log variance, we choose the step size of the steepest ascent analysis to be at the maximum of the fitted curve and the updated importance sampler value for the next iteration is shown as the red triangle in the contour plots in the first row.

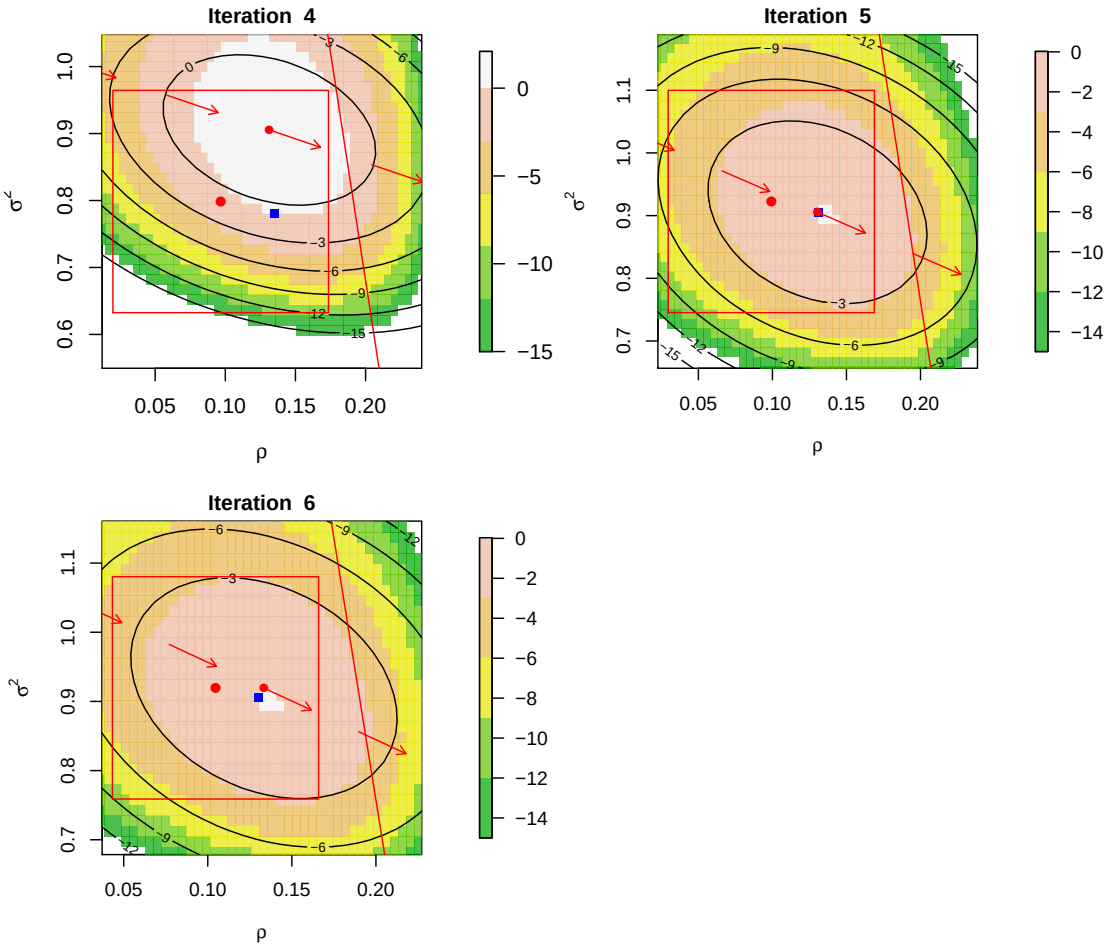


Fig. 4.5 (cont.) Evolutions of the response surfaces for a direct CAR model on the torus: Iteration 4 to 6. In all three iterations, the approximated systems have stationary points within the corresponding design region hence no steepest ascent analysis was needed. The larger red point in each plot is the design centre while the smaller one is the stationary point and the red arrows show the canonical path of the stationary second order system.

Figure 4.5 (cont.) shows the results from iterations 4 to 6. In the fourth iteration, the design region has already moved close to the target region and the fitted response has a stationary point within the design region. At this stage we may conclude that the system has found the optimal region. Once the stationary point was found we reduced k by half for a smaller design region and doubled the Monte Carlo sample size. Then at iteration 5 the procedure converges according to the current tolerance on the difference of the optimum points. In the last iteration, k is reduced by half again to improve the accuracy of the response surface approximation. The contours of the fitted response are very close to the exact value in the last two iterations.

One benefit of the RSM is that apart from the optimum point, it also shows the level sets of the approximated likelihood ratio function. This can be used as a guidance for likelihood ratio tests. For a two parameter test, the true parameter value and the exact MLE are both within the contour level of -3, which means they are not significantly different from the Monte Carlo MLE at the 5% level.

A simulation comparison of the direct optimisation and RSM optimisation

In this section, we compare the Monte Carlo MLE obtained by the iterative optimisation procedure and the RSM procedure to the exact MLE using simulated data with zero mean from direct CAR models on the 20×20 torus. We have chosen to following five groups of parameter values in table 4.3 for (ρ, σ), which covers most range of $\rho \in (-0.25, 0.25)$ and also reflect the change in the scale parameter σ^2.

Table 4.3 Parameter values for the simulation study

	1	2	3	4	5
ρ	-0.2	-0.05	0.1	0.24	0.24
σ^2	1	1	1	1	2

Then for each group of values, we simulate a data set and compute the corresponding exact MLE, the Monte Carlo MLE from the iterative optimisation procedure (MC.MLE), the Monte Carlo MLE from the RSM procedure (RSM.MLE) and the corresponding Monte Carlo errors using 100 Monte Carlo samples for iterations and 200 samples for the final step. These five groups of experiments are replicated 100 times and we used the results to calculate the mean square error of the Monte Carlo MLE to the exact MLE and the coverage of the approximated 95% confidence interval of the Monte Carlo MLE as shown in Figure 4.6 and Figure 4.7.

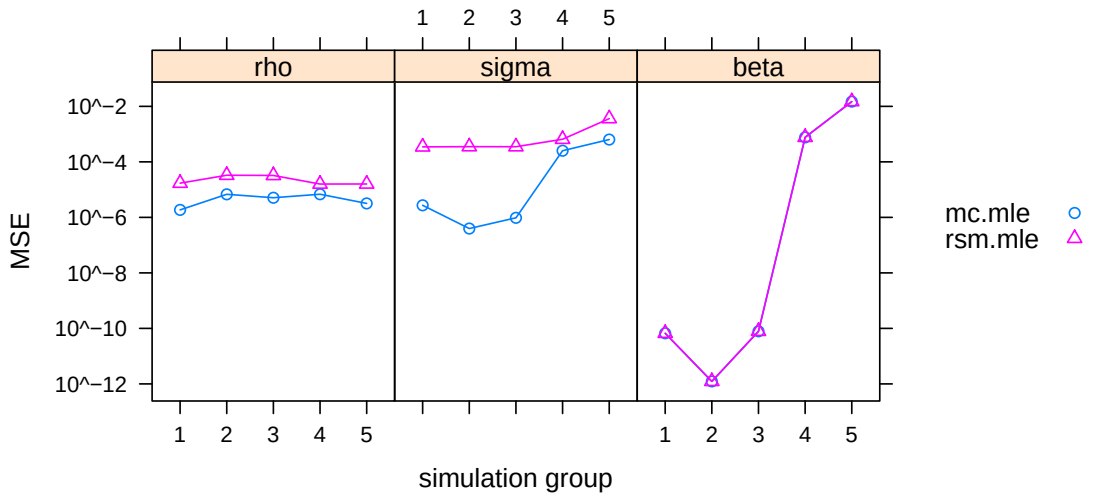


Fig. 4.6 Mean square errors of the Monte Carlo MLEs to the exact MLEs in log 10 scale.

In Figure 4.6, the mean square errors for all parameters and all experiments are below or around 10^{-2} which is roughly the reciprocal of the Monte Carlo sample size. The coverages of the approximated 95% CIs are around 0.95 except for the RSM.MLE of ρ when ρ is close to zero and the upper bound. This is because the rsm method only uses an approximation to the Monte Carlo likelihood function in searching the maximum and hence introduces more bias and error. In general, the

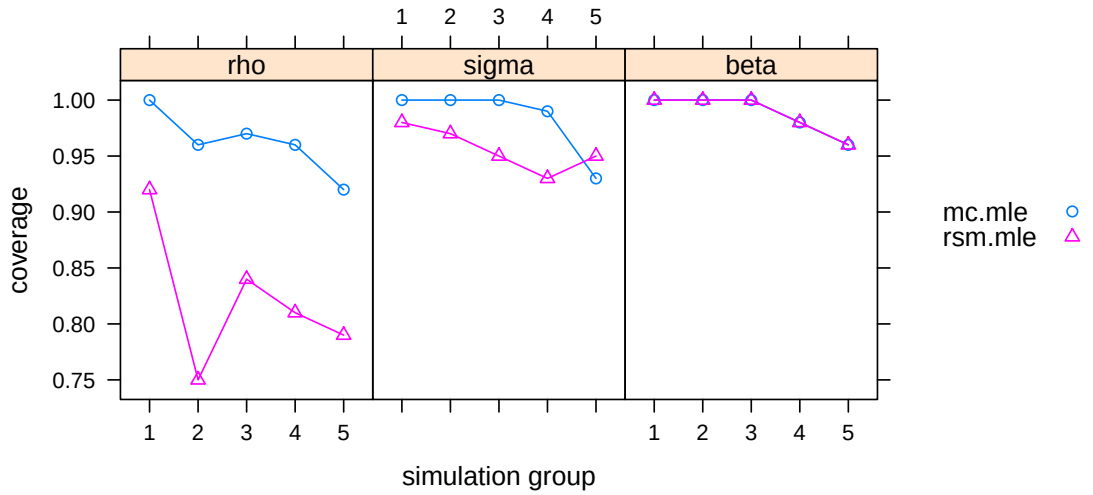


Fig. 4.7 Mean square errors of the Monte Carlo MLEs to the exact MLEs in log 10 scale.

iterative procedure of directly maximise the Monte Carlo likelihood provides more accurate estimates fo the MLE.

The trends in both figures also show that the Monte Carlo methods become less accurate as ρ get close to its limits and σ^2 become larger. This is due to the fact that the difference between the distributions changes more drastically for ρ near its limits and therefore more Monte Carlo samples are needed for reducing the error.

It should be noted that the above analysis focuses on the accuracy of the Monte Carlo MLE in approximating the exact MLE, so the mean square error and coverage are both calculated for the Monte Carlo MLEs to the exact MLE, which is different from the usual sense of mean square error and coverage for an estimator of a parameter. The mean square error of the Monte Carlo MLE to the true value should be the sum of the two parts.

Figure 4.8 shows the box plots of the Monte Carlo MLEs centred by the exact MLEs and Figure 4.9 shows the box plots of the exact MLEs centred by the true parameter values in the 100 replications. The errors of the former are much smaller than the latter and can be controlled by the Monte Carlo sample size.

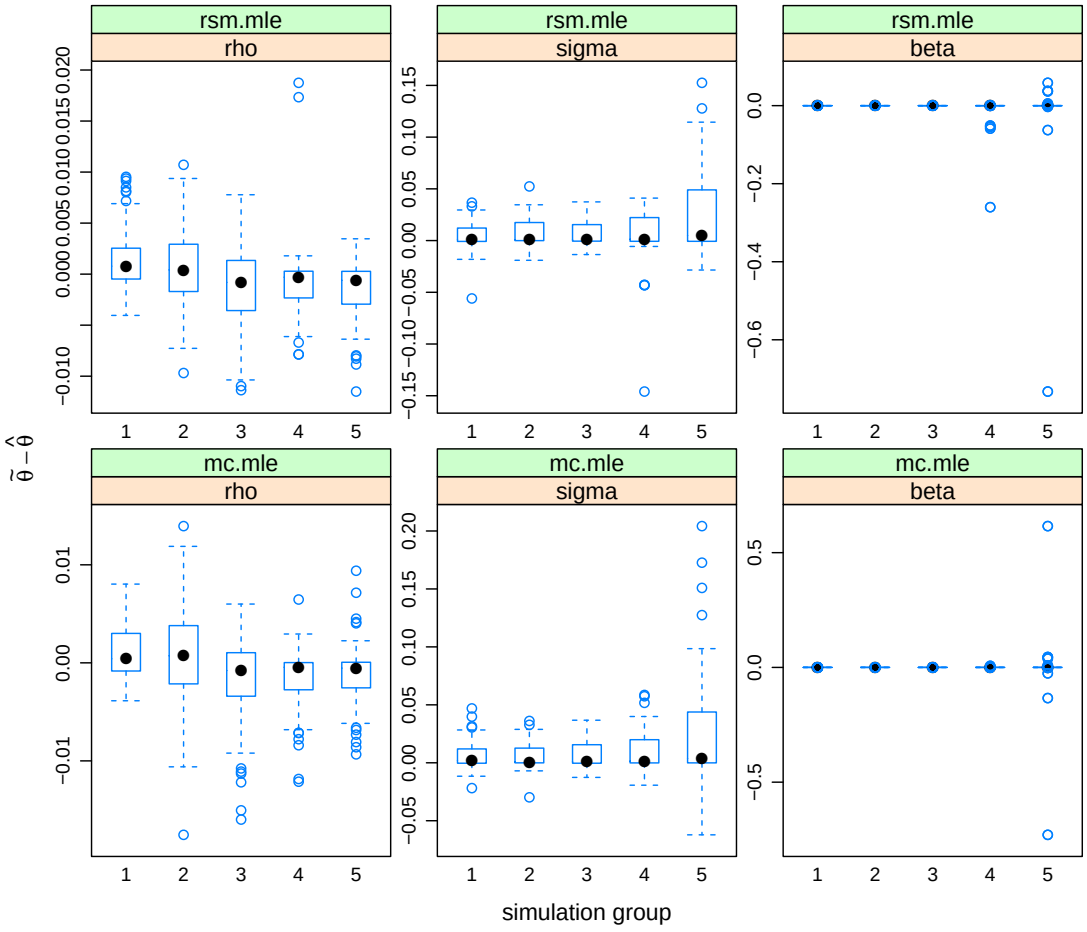


Fig. 4.8 Box plots of the Monte Carlo MLEs ($\tilde{\theta}$) centred by the exact MLEs ($\hat{\theta}$).

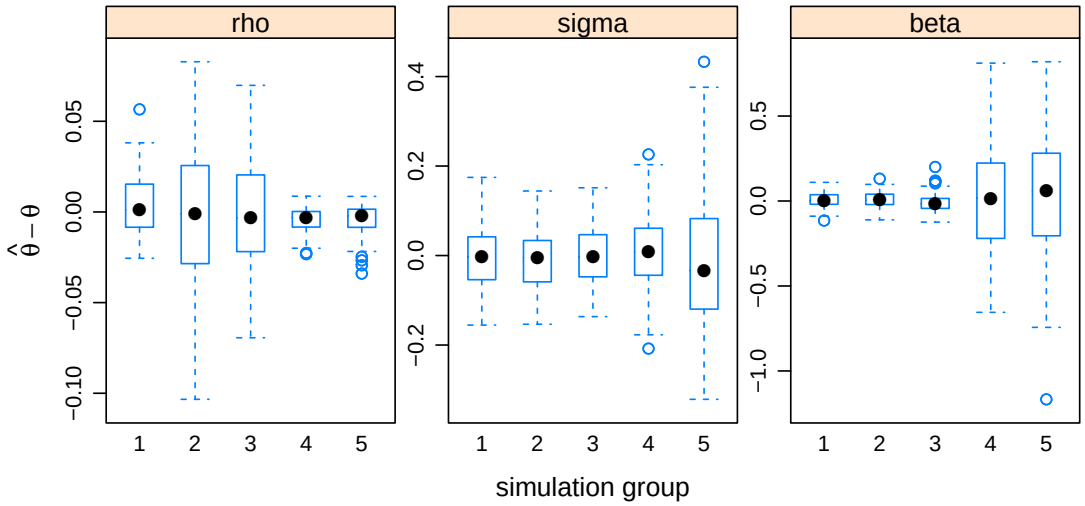


Fig. 4.9 Box plots of the exact MLEs ($\hat{\theta}$) centred by the true parameter values (θ).

The simulation study in this section shows that the two proposed methods both provide good approximations to the exact MLE for almost the entire range of ρ and the Monte Carlo variance is roughly proportional to the reciprocal of the Monte Carlo sample size. When ρ is close to its limits, we had a few failures in the simulation experiment due to numerical singularity. This happened even when calculating the exact MLEs. In these "unsuccessful" experiments, the iterative procedure may also fail for the same reason but the RSM method can always produce a reasonable estimate. Therefore, when strong spatial correlation is detected in a primarily analysis, it is recommended to use more Monte Carlo samples and restrict the searching space. It is also helpful to use the RSM method to find an initial value.

4.3.2 GLM with latent CAR variables

In this example, we apply the RSM to the data of binomial observations with CAR latent variables. The model is defined in (3.10) and the likelihood in equation (3.11). To simulate the data, we set $m = 30, \boldsymbol{\beta} = (1, 1)^T, \rho = 0.2, \sigma^2 = 1.5$ on a

20×20 torus. The likelihood ratio is approximated by using equation (2.6) with the log-determinant of the covariance matrix computed exactly.

The initial values for $\boldsymbol{\beta}$ are estimated by a GLM assuming no spatial correlations among the observations; and ρ, σ^2 are estimated by fitting a direct CAR model to the residuals from the fitted GLM response. In all iterations except for the final one, the Monte Carlo sample size is chosen to be 1000; in the last iteration, the Monte Carlo sample size is increased to 5000. The repeated evaluation at the centre points use a quarter of the entire samples and this subset is generated by sampling without replacement for each evaluation. Figure 4.10 and 4.10 (cont.) show the evolution of the response surface.

The first two iterations are shown in Figure 4.10. The design regions are both close to the target area and the response model begins to detect the second order effect; however the fitted response appears to be a rising ridge in both iterations which means the stationary point is far away from the current design region. Therefore the steepest-ascent analysis was used to decide the importance sampler value as shown in the right panel of the figure.

Figure 4.10 (cont.) shows the last three iterations. In iteration 3, the stationary point is found to be within the design region, hence we increase the Monte Carlo sample size in iteration 4. The stationary point is again found to be within the design region which confirms the result in the last iteration that the algorithm has converged to the target region. To improve the accuracy of the fitted response, we use a smaller design region and the result in iteration 5 is consistent with the previous two. Although the direct optimization achieved similar results to the RSM with less computing cost in this example, the RSM is still a more robust method when the direct optimization might fail.

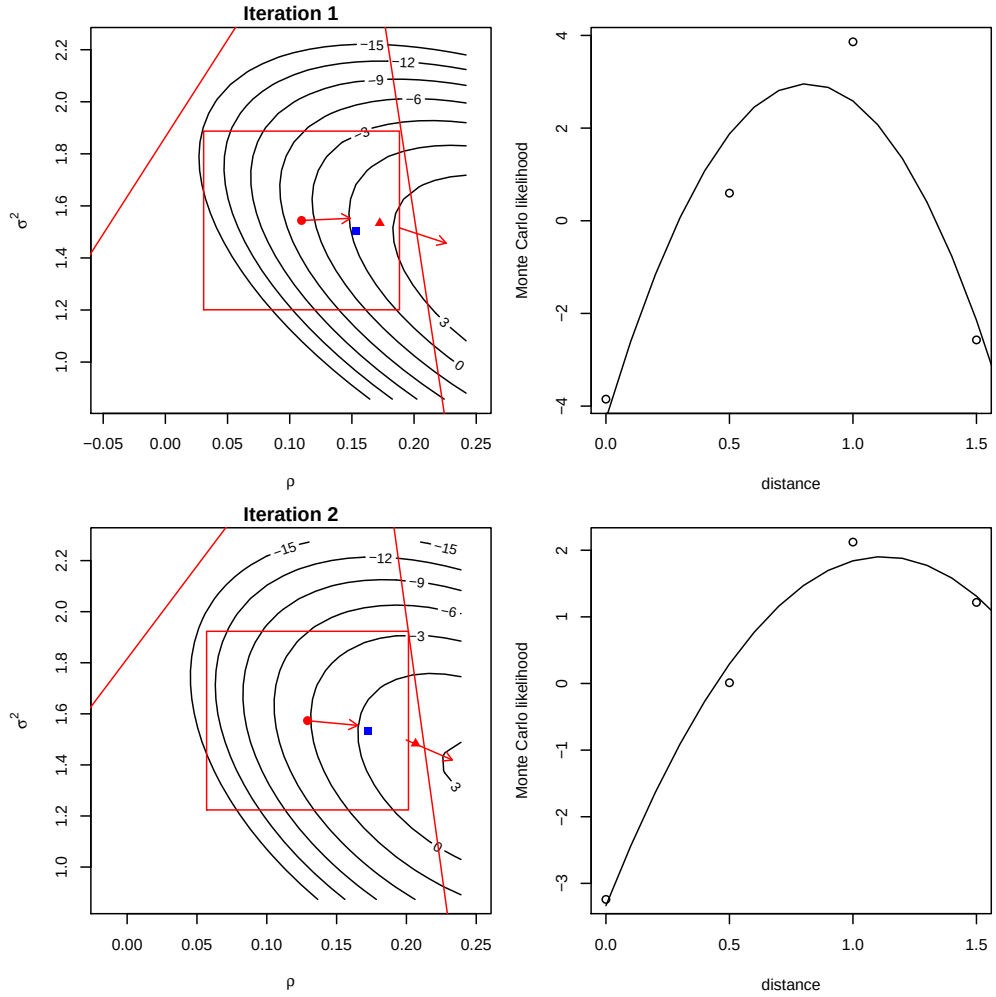


Fig. 4.10 Evolutions of the response surfaces for a latent CAR model on the torus: Iteration 1 and 2. The left panels show the contours of the fitted response in each plot and the right panels are the corresponding steepest analysis plots. The design region is within square and the finite variance region is partly shown within the two intersecting lines. The red dot is the design centre, the blue square is the importance sampler value and the red arrows starts from the design centre show the steepest ascent path. The red triangle on the steepest path is the updated importance sampler value for next iteration. More details to be found in text.

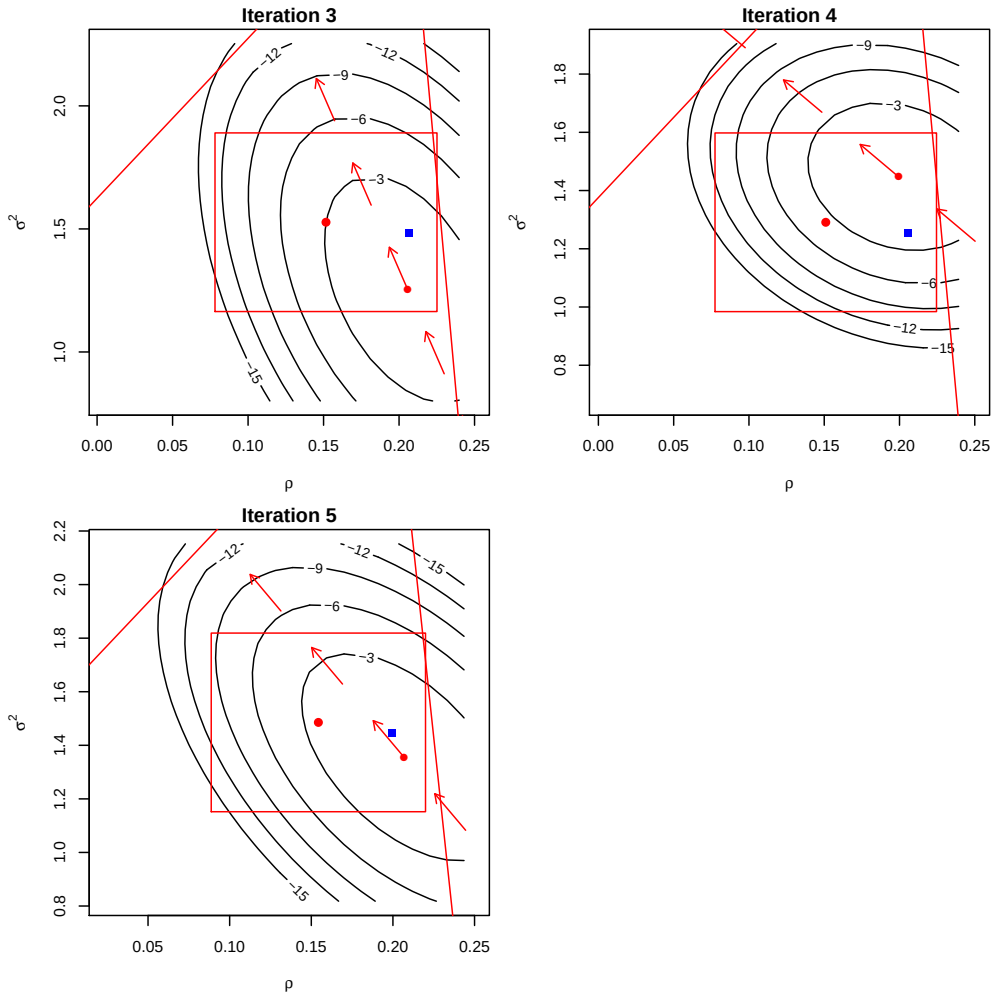


Fig. 4.10 (cont.) Evolutions of the response surfaces for a latent CAR model on the torus: Iteration 3 to 5. The system has stationary point within the design region in Iteration 3. For better accuracy, in Iteration 4 the Monte Carlo sample size is doubled and in Iteration 5, the design region is decreased. More details to be found in text.

4.4 Conclusion

In this chapter, we adapted the response surface methodology in maximising the Monte Carlo likelihood. The RSM have been used in maximising unknown functions with stochastic errors but the method was designed for homogeneous errors which can be estimated by repeated measurements at the central points. We are the first to apply this method on a function approximated by computer simulations with inhomogeneous simulation errors. Our proposed RSM procedure generalises the classical RSM method by first combining the variance estimates in fitting the lower order approximations to the target function and then by monitoring the growth trend of the variance estimates in the steepest analysis.

The proposed method is also implemented in the function `rsmMCL` in the package `mclcar` and we provide examples for applying this procedure on both direct and latent CAR models. For the direct CAR models where exact MLE can be computed, we compared the RSM procedure and the iterative procedure proposed in Chapter 3 on 100 simulated data sets for five groups of different parameter values that represent different spatial correlations. Both the RSM and iterative procedure provide good coverage of the exact MLE. The iterative procedure is usually more accurate than the RSM since it maximises the Monte Carlo likelihood function directly; however, when the value of ρ is close to its limits, direct optimisation might fail but RSM can still provide reasonable approximation. Therefore, it is recommended to combine the RSM and the iterative procedure when little information is known of the spatial correlation parameter.

Chapter 5

Applications

Spatial information is often available with data from areas such as epidemiology, environmental studies, etc. In this chapter, we use the Monte Carlo likelihood method on two data sets in the CAR literature. In our first application, we study the Scottish lip cancer data set (Breslow and Clayton, 1993) and show that estimation results from the Monte Carlo likelihood method are consistent with those from previous methods. Then we study the species distribution of British native plants, a more challenging data set with about 3000 observations. Both data sets are modelled by a generalised linear model with an unobserved spatial error term in the linear trend. Another application about a more complicated problem is described in Chapter 6.

5.1 Scottish lip cancer

The Scottish Lip Cancer data set is frequently used as an example of estimating CAR parameters (Breslow and Clayton, 1993; Clayton and Kaldor, 1987; Rousset and Ferdy, 2014; Ronnegard et al., 2010). The data set is available in several R packages (such as `hglm`, `CARBayes`). We have extract the data and map information from these packages and included them as `ScotCancer` and `Scotplot` in our package

`mclcar`. The following lines show the R code for loading the package and the data.

```
library(mclcar)
str(ScotCancer)

## List of 10
##  $ rho   : NULL
##  $ sigma : NULL
##  $ beta  : NULL
##  $ y     : num [1:56] 6.43 4.48 3.67 3.6 3.49 ...
##  $ covX  : num [1:56, 1:2] 1 1 1 1 1 1 1 1 1 1 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:56] "1" "2" "3" "4" ...
##   .. ..$ : chr [1:2] "(Intercept)" "Paff"
##   ..- attr(*, "assign")= int [1:2] 0 1
##  $ W     : num [1:56, 1:56] 0 0 0 0 1 0 0 0 1 0 ...
##  $ Z.true: NULL
##  $ eta   : NULL
##  $ Emean : NULL
##  $ psi   : num [1:4] 0.165 0.184 0.55 0.023

class(scotplot)

## [1] "SpatialPolygonsDataFrame"
## attr(,"package")
## [1] "sp"
```

The data object `ScotCancer` is assemble in the format required by the functions for Monte Carlo likelihood estimation and `scotplot` is a `SpatialPolygonsDataFrame`

that can be used to map the data by using packages such as `sp` (Bivand et al., 2008). Table 5.1 summarises the variables of interests in this data set.

Table 5.1 The Scottish lip cancer data

variables	description
`y`	count of observed incidents of lip cancer in 56 Scottish counties;
`expec`	expected number of incidents based on the standardised mortality ratios
`p.aff`	percentage of the work force in agriculture, fishing or forestry;
`W.Ad`	neighbourhood matrix defined by counties sharing common borders.

The `ScotCancer$psi` value in the data set is the maximum pseudo-likelihood estimate (MPLE) of the log response to be used as the initial importance sampler value in the Monte Carlo likelihood methods and it is obtained by the following codes.

```
obs.response <- log(ScotCancer$y + 0.5) # added 0.5 in case obs=0
Z.data <- list(data.vec = data.frame(y=obs.response,
                                     covX=ScotCancer$covX[,-1]),
               W = ScotCancer$W)
Z.mple <-mple.dCAR(Z.data)
psi <- Z.mple
ScotCancer$psi <- psi
```

A preliminary screening of the spatial correlation can be done by calculating the Moran's I for testing the global strength of spatial correlation reflected by the parameter ρ. We have done this test on the standardised mortality ratios (SMR) and it provides a p-value much smaller than 0.01, indicating strong evidence against the null hypothesis that there is no spatial correlations.

```
library(spdep)
scot.smr <- probmap(n=scotplot$observed,
                    x = scotplot$expected)$relRisk/100
```

```
moran.test(scot.smr, listw = mat2listw(ScotCancer$W))

##
##  Moran I test under randomisation
##
## data:  scot.smr
## weights: mat2listw(ScotCancer$W)
##
## Moran I statistic standard deviate = 6.7111, p-value = 9.66e-12
## alternative hypothesis: greater
## sample estimates:
## Moran I statistic       Expectation          Variance
##       0.521234276      -0.018181818       0.006460472
```

We can visualise the data on a map and the following code shows how to map the SMRs. Figure 5.1 shows the plot and the SMRs of counties near to each other also tend to be similar.

```
library(RColorBrewer)
pal <- colorRampPalette(brewer.pal(7, "Oranges"))
at_FE <- seq(0, 7, length.out=21)
scotplot$smr <- scot.smr
spplot(scotplot, "smr", col.regions=pal(20), at=at_FE)
```

The purpose is to predict the counts of observed incidents from the given demographic and spatial information; hence a Poisson model with CAR latent

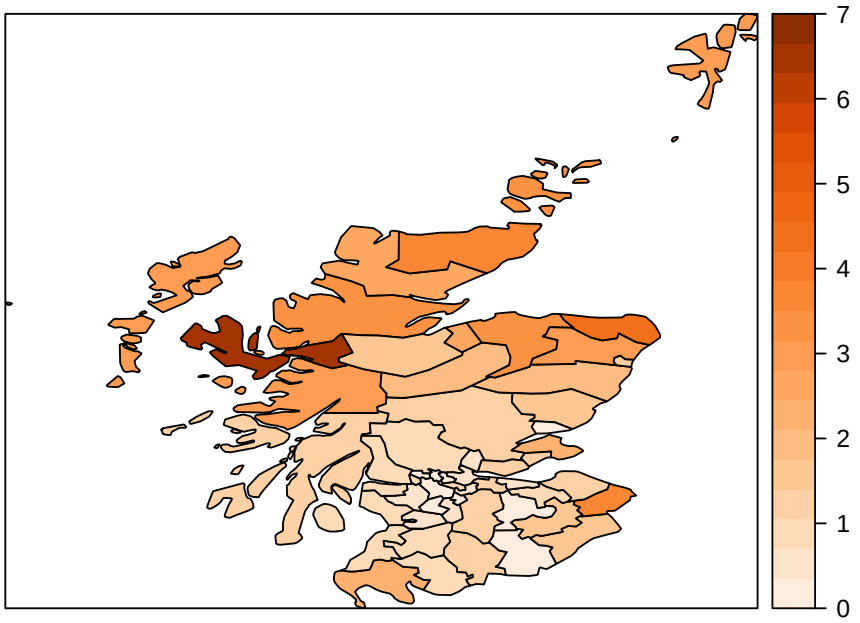


Fig. 5.1 Map of the standard mortality rate for each county in Scotland.

variables was used

$$\begin{cases} \mathbf{Y} \sim \text{Poisson}(\boldsymbol{\mu}) \\ \log(\boldsymbol{\mu}) = \boldsymbol{\eta} \\ \boldsymbol{\eta} = \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}, \;\; \mathbf{Z} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}) \end{cases} \tag{5.1}$$

where $\mathbf{X}$ contains the percentage of the work force in agriculture, fishing and forestry as the explanatory variable and log of the expected number of incidents as the offset, and the covariance matrix of the CAR variable is defined to be $\boldsymbol{\Sigma} = \sigma^2(I - \rho\,\texttt{W.Ad})^{-1}$.

Before using the function `OptimMCL` and `rsmMCL` in the package, we need to specify some parameters for controlling the procedures. The following codes provide as an example of setting these parameters; both function provides default values for use in general except for the `psi.range` which specifies the range of ρ and σ^2 and should be given by user according to the data set. Detailed descriptions of the control list can be found at the user's manual in Appendix C.

```
ew <- eigen(ScotCancer$W, only.values = TRUE)$values
b1 <- 1/min(ew) + 1e-5
b2 <- 1/max(ew) - 1e-5
con1 <- list(n.iter = 20, mc.var = TRUE,
             psi.range = c(b1, b2, 0.1, 10))
mc.con1 <- list(N.Zy = 2e4, Scale = 1.65/(56^(2/6)),
                thin = 50, burns = 1.5e4,
                method = "mala", scale.fixed = FALSE)
```

Then with the above setting we can estimate the model by using the following codes for the iterative procedure and RSM procedure for maximising the Monte Carlo likelihood.

```
mcmle1 <- OptimMCL(data = ScotCancer, psi0 = psi,
                   family = "poisson", control = con1,
                   mc.control = mc.con1 )
res1 <- summary(mcmcle1, family = "poisson", mc.covar=TRUE)
```

```
con2 <- list(n.iter = 20, mc.var = TRUE,
             psi.lim = list(r.lim = c(b1, b2),
                            s.lim = c(0.01, 5)))
mcmle2 <- rsmMCL(data = ScotCancer, psi0 = psi, family = "poisson",
                  control = con2,
                  mc.control = mc.con1)
res2 <- summary(mcmle2, family = "poisson")
```

Table 5.2 compares the estimates of the Monte Carlo likelihood methods with estimates from various methods. The `glm` model is a Poisson model without spatial effects and the `glmmPQL` and `INLA` provide esitamtes for an intrinsic CAR model where ρ is fixed at its maximum 0.175. Estimation results for the two rows are from Clayton and Kaldor (1987) and the result of `INLA` is calculated by using the default options for the priors. By assuming an intrinsic CAR prior for the spatial random effects, the estimates of the fixed effects greatly decrease. It should be noted that INLA produces different estimates when changing the priors.

Then the model specified in (5.1) is fitted by the packages `hglm` and `spaMM`, and the Monte Carlo likelihood method, where the Monte Carlo likelihood method uses the MPLE of the linear part in model (5.1) as the starting point for the importance sampler. The `hglm` package fits the hierarchical generalised linear models using the extended quasi likelihood method (Ronnegard et al., 2010) by default. The spatial mixed models of `spaMM` (Rousset and Ferdy, 2014) use the Laplace approximation to the marginal likelihood function.

For the Monte Carlo likelihood method, we use only 100 samples generated by MALA in the approximation as the dimensionality of the latent variable is relatively small. The Monte Carlo maximum likelihood estimator MC-MLE1 was obtained by directly optimising the Monte Carlo likelihood function for one iteration and the MC-MLEn is obtained by the iterative procedure described in 3. The MC-MLEn converges after 6 iterations. In the table we also show the value of 2× Monte Carlo error for the MC-MLE in the brackets after the estimates.

Table 5.2 Estimation results of the Scottish lip cancer data

	Estimated coefficients			
	Intercept	`p.aff`/10	ρ	σ^2
`glm`	-0.54	0.74	– –	– –
`glmmPQL*`	-0.18	0.35	0.175	0.73
`INLA*`	-0.11	0.263	0.175	0.416
`hglm`	0.23	0.38	0.174	0.156
`spaMM`	0.24	0.38	0.174	0.155
MPLE	0.55	0.23	0.165	0.184
MC-MLE1	0.24(0.20)	0.43(0.13)	0.174 (0.0008)	0.118 (0.13)
MC-MLEn	0.50(0.02)	0.33(0.01)	0.175 (0.0001)	0.102 (0.03)

* The spatial random effects follow the intrinsic CAR model, where $\rho = 0.175$ is fixed at the maximum of the range of ρ.

The likelihood based methods (`hglm`, `spaMM`, `MC-MLE`) all give similar results for the Poisson model with CAR latent effect. The data show very strong spatial correlation as the estimated ρ is close to the maximum value 0.175 for a valid CAR model, which may support the use a an intrinsic prior for the random effects. It should be noted that none of these methods provides the exact MLE but MC-MLE method provides the error bars for the MC-MLE to cover the exact MLE. Comparing the last two row of table 5.2, the error bars become much narrower by using the iterative procedure since in the final iteration the importance sampler value and the maximum are very close to each other.

5.2 British native plant species

The British native plant species data were used by Golding (2013) to compare different species distribution models (SDMs). The goal of SDMs is to predict the distribution of the species at given locations using environmental covariates. Records $\mathbf{Y}$ of the presence or absence of the species of interest are collected on a gridded map and the corresponding environmental predictors $\mathbf{X}$ are also available (usually from satellite images) with the same or higher resolution. Then $\mathbf{Y}$ is assumed to be binary outcomes of the species distribution with parameter $\mathbf{p}$ and under the framework of generalised linear models (GLM), we can write

$$\begin{cases} \mathbf{Y} \sim \text{Bin}(\mathbf{1}, \mathbf{p}) \\ \text{logit}(\mathbf{p}) = \boldsymbol{\eta} \\ \boldsymbol{\eta} = \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}, \;\; \mathbf{Z} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}) \end{cases} \tag{5.2}$$

The simple linear relationship between the response $\boldsymbol{\eta}$ and the predictors can be modified to non-linear models such as boosted regression trees (BRT), generalised additive models (GAM) and Gaussian processes (GP), etc., in which cases, the spatial information is usually represented by the projected coordinates of the point that can be used as predictors. The models are fitted using a small training set of the data and the rest are used for model selection.

These non-linear models are more flexible than the GLM and tend to have better performance in terms of prediction, however they also have the drawback that the explanatory power of the model may be obscured. In the following study, instead of comparing prediction performance, we focus on learning the spatial correlation of the data by using the CAR model.

5.2.1 Data description

The British native plant species data is a processed dataset provided by Golding (2013) who extracted it from Preston et al. (2002), which contains the distributions of 1335 vascular plants of Britain, Ireland, the Isle of Man and the Channel Islands[1]. In the processed dataset, only 227 species were selected according to rules explained in Golding (2013).

For each of the 227 species, presence or absence records are collected at a square grid with resolution of 10km, which are assumed to represent known presence or absence instead of presence only. As an illustration, Figure 5.2 shows the record map of the first species, *Hypericum maculatum*, in the data set.

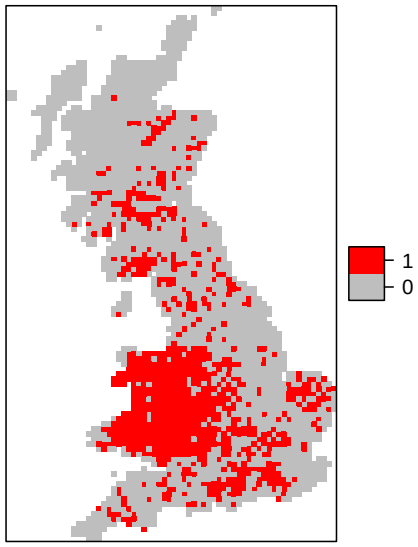


Fig. 5.2 Map of the first species. The red squares have presence records.

For each cell of the grid, predictors are given as 10 principal components derived from an elevation map and more than 800 satellite images. These images are 7-year

[1]The map in 5.2 and the followings do not show the entire region since the we only have a processed data set that includes the regions as shown in the maps.

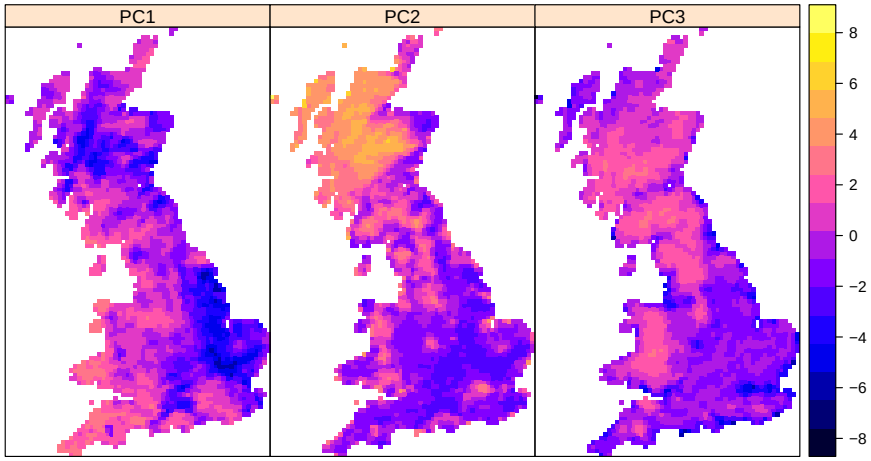


Fig. 5.3 Images of the first three PCs of the environmental variables.

time series of maps in three bands [2] that represent the spatio-temporal variation of environmental conditions. The resolution available is about 250m per pixel. The satellite images were first reduced to 30 components by a temporal Fourier analysis, then a principal component analysis was done to the 30 satellite signals plus the elevation map to obtain the 10 principal components. These 10 PCs are claimed to explain 90% of the variations but the biological and environmental interpretation are not clear. The first three PCs are claimed to correspond to gradients from arable to pasture, lowlands to uplands and urban areas to rural areas and they are shown as plots in Figure 5.3. Finally, to have the same level of resolution as the species records, pixels from these 10 images were re-sampled from 250m resolution to 10km resolution. Thus, both the plant distribution data and the environmental data are given for a total of 2774 squares with a 10km resolution in the data set.

Apart from the distribution map of individual plants, we create the species diversity map shown in the left panel of Figure 5.4 by aggregating the presence of the 227 species in each square. The right panel shows the histogram of the counts

[2]The three bands are mid infra-red(MIR; output of wavelengths, $3-5\mu$m); enhanced vegetation index(EVI; a metric of vegetation cover) and normalized vegetation index(NDVI; a metric of vegetation cover complementary to EVI).

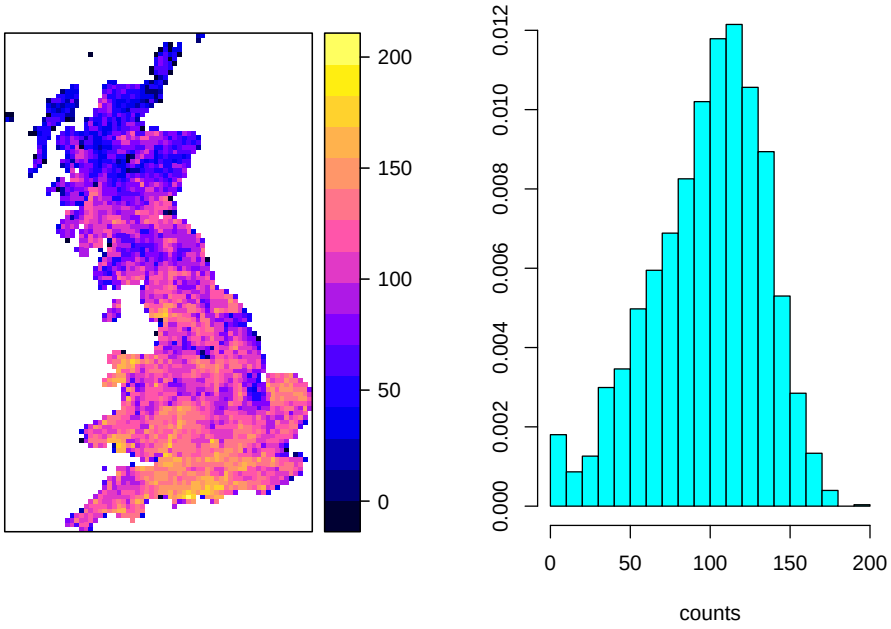


Fig. 5.4 Species diversity map. Left: the counts are the number of species presence in each surveyed square. Right: histogram of the counts.

and the shape looks similar to a normal distribution, except that there are a few counts equal to zero. Hence it is suitable to assume the counts as either binomial or Poisson distribution. The Moran's I test on the standardised counts produce a p-value less than 0.01, which means there is strong evidence for spatial correlation.

Figure 5.5 shows the scatter plots of the transformed response variable, the counts, against the the first two covariates. The top panel shows the log transformed response for the Poisson model and the bottom panel shows the logit transformed response for the binomial model. Both panels show similar linear trends. Most of the observations of small counts tend to have PCs with values close to zero.

5.2.2 Model and estimation

We fit the model (5.2) with the covariance matrix $\mathbf{\Sigma}$ specified by a CAR model. The spatial weight matrix is generated using the "rook-style" neighbours of the survey grid. We fit the aggregated species diversity map where the response can be modelled either as outcomes of a binomial or a Poisson distribution. We again use the MPLE of the linear response model as the starting value for the importance sampler value and directly optimise the obtained Monte Carlo function assuming the starting point is close to the target. The estimates are given in the table 5.3 as "MC-MLE1". Based on this estimate, we reconstruct the likelihood surface by the response surface method; however, the RSM suggests that the optimum point is far outside of the current design region, we therefore conduct the RSM starting from the MPLE for a few iterations until it converges. Then we optimise the Monte Carlo likelihood based on the last RSM output and the optimal point is reported in the row "MC-MLE2" of the table.

We tried to apply the hglm and spaMM methods used in the previous section on this data set, but neither neither provides satisfactory answers: the algorithms either failed due to numerical issues or did not converge after reaching maximum

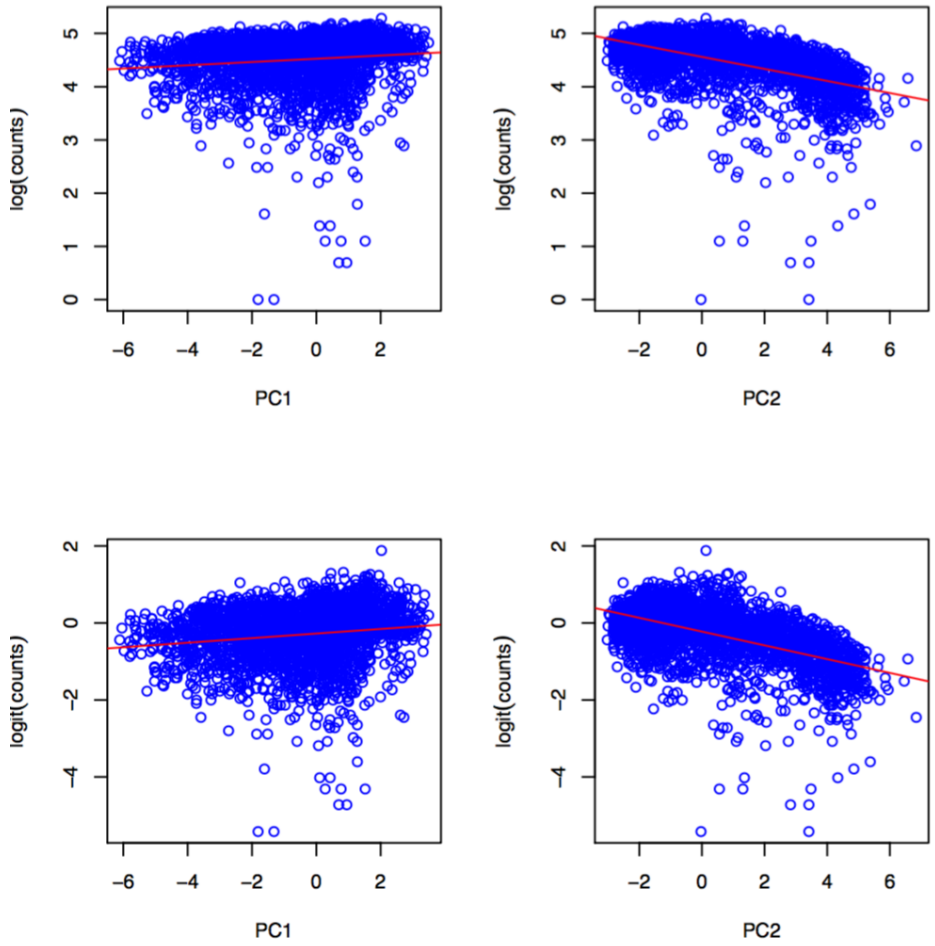


Fig. 5.5 Scatter plots of the response against the first two PCs with fitted lines from linear regression. Top: log transformed response; bottom: logit transformed response.

Table 5.3 Estimation results of species diversity map

	Binomial		Poisson	
	ρ	σ^2	ρ	σ^2
MPLE	0.2216	0.5170	0.1950	0.3838
MC-MLE1	0.2443	0.2022	0.1692	0.1971
RSM final	0.2497	0.2037	0.2492	0.0823
MC-MLE2	0.2489	0.2034	0.2491	0.0823

number of iteration. We have also conduct a Bayesian analysis by using INLA with intrinsic CAR prior on the spatial random effects, and the estimates of the fixed effects are very similar. However, as discussed in the previous section the result for the random effects are not comparable; hence here we omitted the results of INLA and focus on the Monte Carlo likelihood method only.

The MPLE was quite close to the target but it underestimated the spatial correlation and overestimated the variance; after direct optimization, the result was not always closer to the target point as shown by the result of the Poisson model. For the binomial model, the RSM converged after 12 iterations and the total computing time for the RSM was 10.11 CPU hours with the first 10 iterations using 100 Monte Carlo samples and the last two iterations using 600; the Poisson model converged within 9 iterations and the computing time was also around 10 CPU hours. For the first four iterations we used 100 Monte Carlo samples and the for last five we used 600. The RSM procedure automatically increases the Monte Carlo sample size s, when the difference between two consecutive outputs are smaller than $1/s$ or when the the stationary point is found to be within the design region and thus the RSM approximation becomes more accurate when we get closer to the target region.

Comparing the results of the binomial and the Poisson model, the estimates of the spatial coefficient ρ are both very close to the upper bound of the parameter space, and thus indicate very strong spatial correlation. The Poisson model has much smaller estimates σ^2 than the binomial model and this is due to the fact that the logit transform has larger variance than the log transform.

Since the estimate of $\boldsymbol{\beta}$ does not change much and profiling out $\boldsymbol{\beta}$ involves solving a huge non-linear system which can be slow and inaccurate, we simply fix $\boldsymbol{\beta}$ to the importance sampler value and only approximate the two-parameter system of ρ and σ^2. We have also compared the above results with those given by

Monte Carlo profile likelihood, and they are very similar although the latter for both binomial and Poisson models took more than 50 CPU hours.

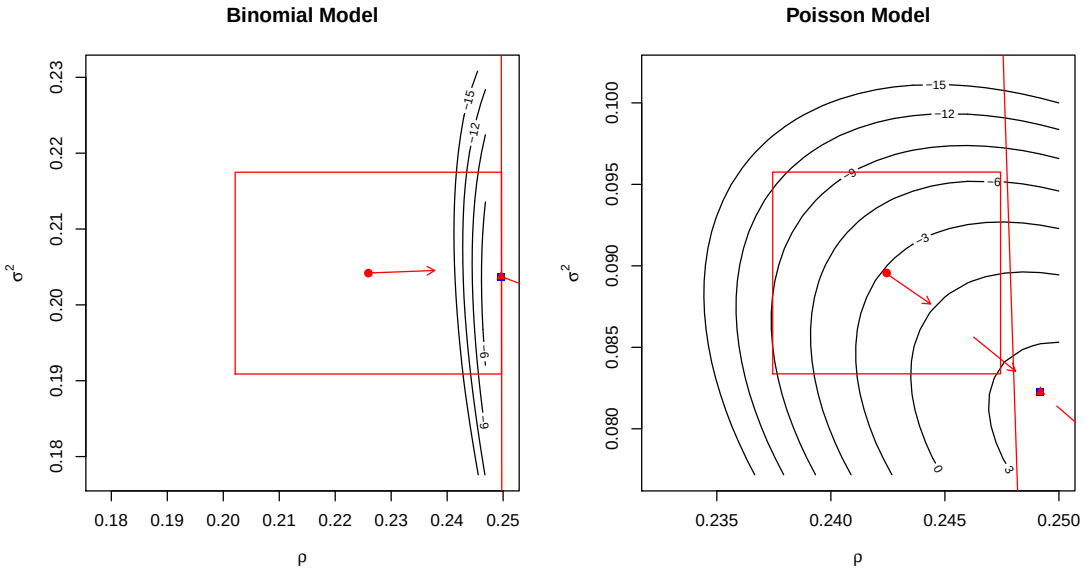


Fig. 5.6 Contours of the fitted response of the Monte Carlo likelihood function for ρ and σ^2 in the final RSM iteration. Th red boxes show the design region generated from the importance sampler value and the red lines on the right are part of the right boundary of the finite variance region. The red dots in the middle of the boxes are the design centres and the red triangle shows the updated optimum point given by the current system and the blue square dots(both covered by the red triangle) are the optima given by the previous iterations. The red arrows show the steepest paths.

Figure 5.6 shows the approximated responses in the final iterations for both binomial and Poisson model. In both plots, the systems are very close to the optimal region but the stationary points are not within the design region; however, the final two iterations gave almost identical outputs. This can been seen from both plots that the final optimal points, shown as the red triangles, cover the previous optimal points shown by the blue squares. The reason that the stationary points are not in the design region here is because the design region is constructed such that it does not exceed that parameter space and we have chosen a relatively

large design region; but even this coarse approximation sufficed to provide accurate results.

Both the binomial and Poisson models suggest that there is a very strong spatial correlation of the species diversity between adjacent survey squares. The selected species are more likely to be observed at a species abundant place and most species tend to cluster around the same area that perhaps have suitable environmental conditions for them all to grow.

5.3 Conclusion

In this chapter, we have applied the Monte Carlo likelihood on two different data sets and compared the results with those using other existing methods. The Monte Carlo likelihood provides very similar results to the others when they do not fail. With a good starting value, the Monte Carlo MLEs are usually very close to the exact maximum, and with our proposed iterative procedure, we can usually find a better approximation with a few addition steps.

In the second example we combined the RSM procedure and the iterative procedure for direct optimisation to show that when the gradient and Hessian of the target function is expensive to calculate or not available, RSM can be used to reduce the computational complexity.

It should be emphasized that the Monte Carlo MLE is an approximate to the MLE and the standard errors are with respect to the exact MLE instead of the "true" parameter values. This should be differentiate with the standard errors given by most other existing methods. This is not a disadvantage: since as has discussed earlier, asympotics results usually does not apply to CAR models on data set with small number of spatial units and none existing method can provide the exact MLE.

Chapter 6

The Sabah biodiversity experiment

Rapid deforestation can lead to various environmental problems such as drought, carbon loss and air pollution; meanwhile trees are being replanted for forest restoration. Thus come the questions from the ecologists on how the arrangement of the planting would affect the functioning of the forest ecosystems? The Sabah Biodiversity Experiment (SEB, http://www.sabahbiodiversityexperiment.org), located at the Royal Society's Danum Valley research station in Sabah, Malaysia, is one of the largest on-going scientific projects to address such questions.

Several studies (Saner, 2009; Hector et al., 2011; Saner et al., 2010), have examined how different factors may affect the forest restoration rate. In this study, we want to investigate the relationship between the growth rate of replanted trees and the biodiversity of the growing environment by considering the spatial effect of the planting scheme.

6.1 Data description

The design of the Sabah Biodiversity Experiment is shown in Figure 6.1 below. The 500 hectares of forest land were divided into 124 four-hectare study plots and these study plots are naturally divided into two blocks by the road between. 16 species of trees were planted according to the study design summarised in Table 6.1. In the table, the three levels of species diversity, low, medium and high, correspond to monoculture (only one species), four-species and all 16 species. Among the study plots with 16 sepcies, 16 are designed to have a enchanced cut (labelled as "ECC" in Figure 6.1). For the study plots with 4 species, 16 combinations of the total $\binom{16}{4} = 1820$ possibilities were chosen according to the table of species mixtures in Golding (2013). Therefore, the number of compositions of species diversity is in total

$$16 \text{ individual species} + 16 \text{ mixtures} + 1 \text{ all species} + 1 \text{ no species} = 34.$$

Table 6.1 Biodiversity treatments

Diversity	Cut	number of study plots
Low	ordinary cut	16×2
Medium	ordinary cut	16×2
High	ordinary cut	16×2
	enhanced cut	16
Unplanted	no cut	12
Total		124

Within each planted plot, trees were planted 10 metres apart in columns and 3 metres apart in rows as shown by the zoom-in of the study plot 69 in Figure 6.1; thus there were about $1,320$ tree seedlings in each study plot and $147,840$ seedlings in total. The diameters of the tree trunks were recorded after three and half years and ten years to measure the growth of the trees; but over the years, some trees

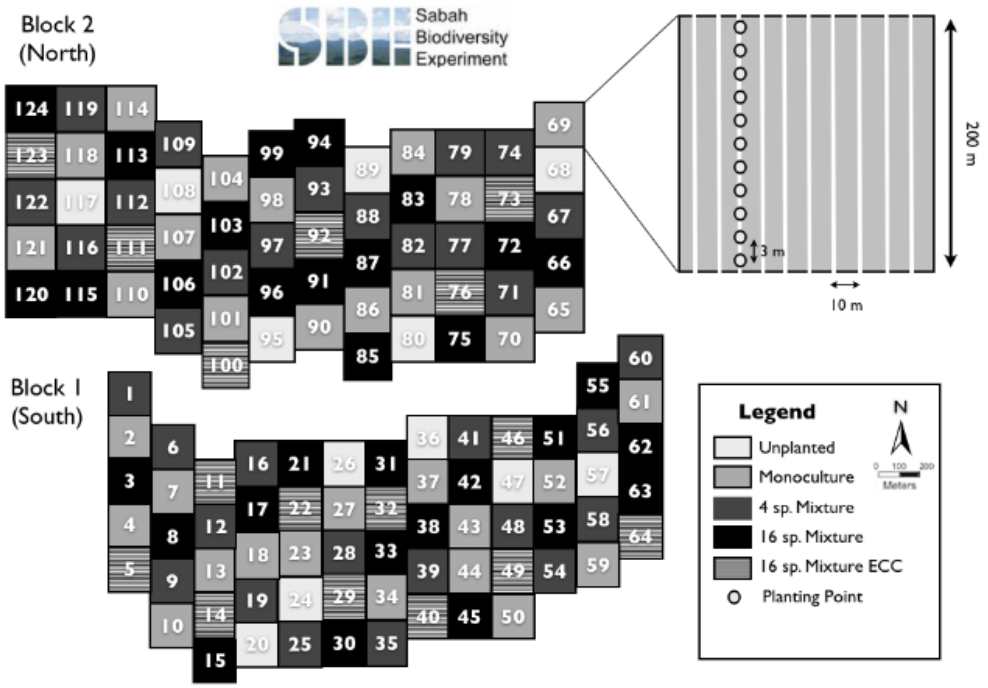


Fig. 6.1 Design map of the study plots from http://www.sabahbiodiversityexperiment.org/the-experiment/a-mega-project.

died and were replanted by new seedlings and therefore in the final data set we have only about 7,000 trees left with both measurements.

With this data set, the researchers are interested in the growth rate of these species under different biodiversity conditions. During the first 10 years, the diameter of the trunk typically grows exponentially fast; hence for each tree the growth rate can be measured as

$$\texttt{growth.rate} = \frac{1}{\texttt{time}} \log(\texttt{diameter.2}/\texttt{diameter.1})$$

where `time` is the time between two measurement in years. This is then fitted by a linear model on the environmental factors. The researchers are particularly interested in answering the following questions,

1. How does increasing biodiversity affect the growth of the trees?

2. Which species or combinations of species provide the best growth?

Previous studies such as Hector et al. (2011) have shown that tree diversity may have some positive effect on forest restoration. In this study, we consider the spatial structure within each study plot and see how that affects the model estimation. In particular,

1. Is there any spatial effect within a study plot and how to account for it?

2. Is the spatial effect same for all the study plots?

Initial screening shows that the following 16 study plots have no records,

$$20, 24, 26, 34, 36, 47, 57, 68, 80, 89, 95, 108, 116, 117, 118, 119$$

among which 12 are designed to be un-planted and 4 are missing for unknown reasons. We simply omit these study plots in our analysis. A simple linear regression model relating the growth rate of the trees to the species of tree and the composition of the plot

$$\mathbb{E}(\texttt{growth.rate}) = \alpha_0 + \sum_{k=1}^{15} \beta_k \texttt{species}_k + \sum_{l=1}^{33} \gamma_l \texttt{treatment}_l$$

was fitted to the observed data and it showed both factors had significant effect on the growth rate, although the model apparently suffers from collinearity due to the design of mono-culture study plots. For simplification, we then merged the levels of `treatment` to have only `mono-culture`, `4-mixtures` and `16-mixtures` and stored it as `treatment2`. The fitted linear model with the mean response equal to the sum and interactions of `species` and `treatment2` shows that the two factors and their interactions have significant effects on the growth rate.

Then we conducted a hierarchical analysis on the data set grouped by study plots. In the hierarchical analysis, the same linear regression model with `species` as the only explanatory variable was fitted for each study plot. The treatment effect

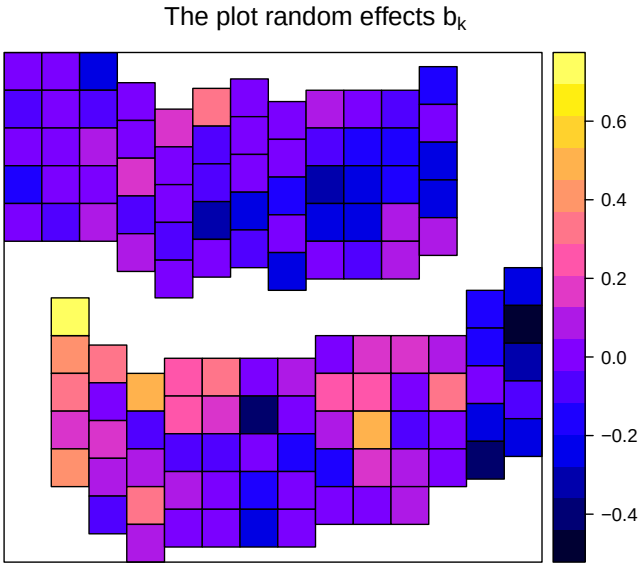


Fig. 6.2 Map of the estimated random plot effects.

is dropped since only one treatment is applied per study plot. For the mono-culture study plots, the model reduced to have only an intercept term. By plotting the residuals on the map, we found that the residuals tend to be similar in magnitude and sign when they are close to each other; this suggests that the growth rate may have spatial correlation based on distance that is similar for all study plots.

We also fitted a linear mixed effect model by adding an independent random effect of study plot b_k, $k = 1, \ldots, 124$ in the previous model that consider the interaction between `species` and `treatment2`. The mixed effect model gives similar estimation results and also improves the fit. The fitted random effects appear to have positive spatial correlation as the estimates tend to be similar for adjacent study plots as shown in Figure 6.2; hence it is natural to consider a spatially correlated random effect for the study plots.

In the following, we model the spatial correlation between trees and study plots. We first define the spatial structure to be used in the model and then estimate the model parameters by hierarchical and mixed effect analysis.

6.2 Defining the spatial structure

Denote by $\mathbf{Y}_k$ the vector of growth rates of trees in the kth study plot, $\mathbf{X}_k, \mathbf{Z}_k$ the corresponding design matrices for fixed and random effects, and $\boldsymbol{\epsilon}_k$ the error term. Then the growth model is

$$\begin{cases} \mathbf{Y}_k = \mathbf{X}_k\boldsymbol{\beta} + \mathbf{Z}_k\boldsymbol{b}_k + \boldsymbol{\epsilon}_k, \quad k = 1, \ldots, K \\ \boldsymbol{\epsilon}_k \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_{\boldsymbol{\epsilon}_k}) \\ \boldsymbol{b}_k \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_{\boldsymbol{b}_k}) \end{cases} \tag{6.1}$$

For a simple linear mixed effect model, both $\boldsymbol{\Sigma}_{\boldsymbol{\epsilon}_k}$ and $\boldsymbol{\Sigma}_{\boldsymbol{b}_k}$ are defined to be diagonal matrices with scale parameters σ^2 and τ^2 and ϵ and $\boldsymbol{b}_k$ are independent of each other. We can write in vector form

$$\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_{\boldsymbol{\epsilon}}), \ \boldsymbol{\Sigma}_{\boldsymbol{\epsilon}} = \mathrm{diag}(\boldsymbol{\Sigma}_{\boldsymbol{\epsilon_1}}, \ldots, \boldsymbol{\Sigma}_{\boldsymbol{\epsilon}_K})$$
$$\boldsymbol{b} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_{\boldsymbol{b}}), \ \boldsymbol{\Sigma}_{\boldsymbol{b}} = \mathrm{diag}(\boldsymbol{\Sigma}_{\boldsymbol{b}_1}, \ldots, \boldsymbol{\Sigma}_{\boldsymbol{b}_K})$$

There are two levels of spatial resolution in this problem: within-plot and between-plot. The within-plot effects describe the spatial correlation between trees. As it reflects the short distance effect between spatial points, we may assume it is the same across all study plots and it can be modelled either on a continuous or discrete spatial support. Meanwhile, the between-plot spatial correlation describes the aggregated effect between each study plot which can be viewed as a spatial polygon and modelled on a discrete support.

Between-plot spatial effect

For the between-plot spatial effect, we considered the 124 study plots as polygons in space based on the plan map. Denote by $\mathbf{W}_p$ the spatial weight matrix for the between plot effects. One natural definition of the weights can be $w_{p_{ij}} = l_{ij}/200$,

where l_{ij} is the length of border in metre that study plot i and j share and 200 is the total length of one side in metres of a study plot. In our analysis since we do not have the exact geographic information of the study region, the polygons are constructed approximately as an illustration; therefore, we simply use a binary weight that indicates whether any two polygons shares common boundaries or not.

Then the plot effect can be assumed to have the following CAR structure

$$\boldsymbol{\Sigma_b} = \boldsymbol{S_b} \otimes \text{diag}(\boldsymbol{\Sigma_{b_1}}, \ldots, \boldsymbol{\Sigma_{b_K}}), \quad \boldsymbol{S_b} = (\mathbf{I} - \rho_p \mathbf{W}_p)^{-1} \tag{6.2}$$

where"$\otimes$" denotes the Kronecker product. Other weight matrices can be considered, but this is one of the simplest.

Within-plot spatial effect

For the within-plot spatial effects, each tree is considered to be a spatial point on a Euclidean plane and has a coordinate according to the planting plan with the origin chosen to be the lower left corner of the study plot 1. The underlying spatial process can be assumed either continuous or discrete.

Continuous support If the underlying spatial process is assumed to be continuous and isotropic, then the covariance function between any two points s_i, s_j can be defined by

$$C(s_i, s_j) = \sigma^2 \rho(u; \phi) \tag{6.3}$$

where $\rho(u; \phi)$ is some function of the Euclidean distance $u = \|s_i - s_j\|$ with parameter ϕ from some standard families. For example, the most commonly used

Matérn family (Diggle and Ribeiro, 2007) has the form

$$\rho(u) = \{2^{\kappa-1}\Gamma(\kappa)\}^{-1}(u/\phi)^{\kappa}K_{\kappa}(u/\phi)$$

where $K_{\kappa}(\cdot)$ is the modified Bessel function of second kind with order κ, $\phi > 0$ is the scale parameter and $\kappa > 0$ is the shape parameter.

If we assume the process is homogeneous over the entire experiment area, then the we can define $\mathbf{\Sigma}_{\epsilon}$ for all the trees. However this will create a huge and dense covariance matrix that can greatly slow down the computation; therefore we may use the following two independence assumptions on defining the within-plot covariance matrices:

1. Block independence: covariance matrices are defined within the two blocks and independent of each other;

2. Plot independence: covariance matrices are defined within study plots and independent of each other.

Then the final estimates of the process parameter would be the average of the block or plot estimates. For the study plot replications, we can also define a hierarchical spatial correlation on the plot parameters using the between-plot spatial structure.

Discrete Support It is also natural to model the spatial process with a discrete support as specified by the design of the experiment. According to the design, trees within each study plot are planted three metres apart in a row and the rows are ten metres apart. Assume the spatial process is isotropic and $\mathbf{W}_e$ is the spatial weight matrix defined by adjacency and standardised by distance, we can define the within-plot error term to have the following covariance matrix

$$\mathbf{\Sigma}_{\epsilon_k}^{(1)} = \sigma^2(\mathbf{I} - \rho_e^{(k)}\mathbf{W}_e^{(k)})^{-1} \tag{6.4}$$

For a non-isotropic process along rows and columns, the covariance matrix can be defined as

$$\boldsymbol{\Sigma}_{\epsilon_k}^{(2)} = \sigma^2(\mathbf{I} - \rho_r^{(k)}\mathbf{W}_r^{(k)} - \rho_c^{(k)}\mathbf{W}_c^{(k)})^{-1} \tag{6.5}$$

where ρ_r, ρ_c are the spatial coefficients along row and column respectively and $\mathbf{W}_r, \mathbf{W}_c$ are the spatial weight matrices defined by row and column adjacency. Then, as the spatial error terms are independent between study plots we have

$$\boldsymbol{\Sigma}_\epsilon = \text{diag}(\boldsymbol{\Sigma}_{\epsilon_1}, \ldots, \boldsymbol{\Sigma}_{\epsilon_K}) \tag{6.6}$$

and by assuming a between-plot spatial effect, we have

$$\boldsymbol{\Sigma}_\epsilon = \begin{pmatrix} \mathbf{S}_{b1,1}\boldsymbol{\Sigma}_{\epsilon_1} & \mathbf{S}_{b1,2}\boldsymbol{\Sigma}_{\epsilon_2} & \cdots & \mathbf{S}_{b1,K}\boldsymbol{\Sigma}_{\epsilon_K} \\ \mathbf{S}_{b2,1}\boldsymbol{\Sigma}_{\epsilon_1} & \mathbf{S}_{b2,2}\boldsymbol{\Sigma}_{\epsilon_2} & \cdots & \mathbf{S}_{b2,K}\boldsymbol{\Sigma}_{\epsilon_K} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{S}_{bK,1}\boldsymbol{\Sigma}_{\epsilon_1} & \mathbf{S}_{bK,2}\boldsymbol{\Sigma}_{\epsilon_2} & \cdots & \mathbf{S}_{bK,K}\boldsymbol{\Sigma}_{\epsilon_K} \end{pmatrix} \tag{6.7}$$

When $\boldsymbol{\Sigma}_{\epsilon_k} = \boldsymbol{\Sigma}_{\epsilon_0}$ for all study plots, the $\boldsymbol{\Sigma}_\epsilon$ simplifies to $\mathbf{S}_b \otimes \boldsymbol{\Sigma}_{\epsilon_0}$.

The processed data set includes only trees with more than two measurements so that most study plots have less than 10% of the trees in the planting scheme. With the incomplete data set, the weight matrices differ between study plots and thus make the spatial coefficients incomparable. Imputation can be used to complete the analysis but with such a small amount of data, the result may not be useful. It seems more sensible to use the error term to describe the short scale spatial correlation over the entire experimental area, rather than allowing it to vary between plots. The covariance matrix of the error term becomes

$$\boldsymbol{\Sigma}_{\boldsymbol{\epsilon}} = \sigma^2(\mathbf{I} - \rho_e\mathbf{W}_e)^{-1} \tag{6.8}$$

where $\mathbf{W}_e$ is the spatial weight matrix for all trees defined according to the description above equation (6.4). Although $\mathbf{W}_e$ is about 7000×7000, computational time can be reduced by using its sparsity. In the following analysis, we use $\boldsymbol{\Sigma_b}$ and $\boldsymbol{\Sigma_\epsilon}$ as defined in equations (6.2) and (6.8).

6.2.1 Parameter estimation

The model parameters are estimated by the MLE and the random effect is given by the BLUP $\mathbb{E}(b|y)$. In the model defined by formula (6.1), $\mathbf{Y} \sim \mathcal{N}(\mathbf{X}\boldsymbol{\beta}, \mathbf{Z}^T\boldsymbol{\Sigma_b}\mathbf{Z}+\boldsymbol{\Sigma_\epsilon})$ and therefore the marginal log-likelihood of the model is

$$\ell(\boldsymbol{\theta}; \mathbf{y}) = -\frac{1}{2}\left(\log|\boldsymbol{\Sigma}_y| + \mathbf{e}^T\boldsymbol{\Sigma}_y^{-1}\mathbf{e}\right) \tag{6.9}$$

where

$$\boldsymbol{\Sigma}_y = \mathbf{Z}^T\boldsymbol{\Sigma_b}\mathbf{Z} + \boldsymbol{\Sigma_\epsilon} \tag{6.10}$$

$$\mathbf{e} = \mathbf{y} - \mathbf{X}\boldsymbol{\beta} \tag{6.11}$$

and thus the corresponding estimate $\tilde{\boldsymbol{b}}$ of the $\boldsymbol{b}$ is

$$\tilde{\boldsymbol{b}} = \mathbb{E}(\boldsymbol{b}|\mathbf{y}) = (\mathbf{Z}^T\boldsymbol{\Sigma}_{\boldsymbol{\epsilon}}^{-1}\mathbf{Z} + \boldsymbol{\Sigma}_{\boldsymbol{b}}^{-1})^{-1}\mathbf{Z}^T\boldsymbol{\Sigma}_{\boldsymbol{\epsilon}}^{-1}\mathbf{e} \tag{6.12}$$

The evaluation of the log-determinant in the likelihood function can be computationally expensive so the Monte Carlo likelihood method is used to estimate this.

6.3 Estimation results

As in the preliminary analysis in section 6.1, we first conduct a within-plot analysis which is fast to compute and then fit a joint model with spatially correlated errors and random plot effects using the estimates from the linear mixed effect model with independent plot effects and errors as a starting point.

6.3.1 Analysis of separate plots

Our preliminary analysis shows that the within plot errors are spatially correlated and therefore, we fit for each study plot independently, a linear model for species with the error term specified by the CAR covariance matrix of equation (6.4).

Figure 6.3 shows the variance estimates for each study plot and Figure 6.4 shows the estimates for the spatial coefficient ρ. Figure 6.4 also shows, for each study plot, the number of links generated by the neighbourhood structure within the plot. The estimated variances do not vary much over the study plots so it would be reasonable to model this as constant over all plots.

The spatial coefficients vary somewhat more: the majority of plots have positive spatial correlation estimates which tends to be similar as the number of links increases. Those with negative correlations are general based on rather few links and therefore the estimates are likely to be unreliable. Therefore, for this data set it is reasonable to model the spatial correlation also as homogeneous over the space.

After adjusting for the spatial correlation within the study plot, we still found the growth rate appears to be similar between neighbouring study plots when plotting the mean growth rate of each study plot by species (figures not included). Hence we consider the model with spatially correlated random effects between the study plots.

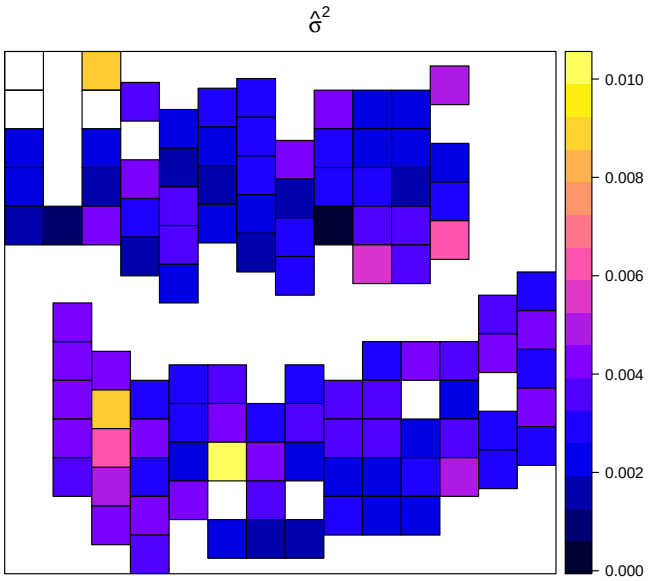


Fig. 6.3 Map of the estimated within-plot variances after adjusting for the within-plot spatial error.

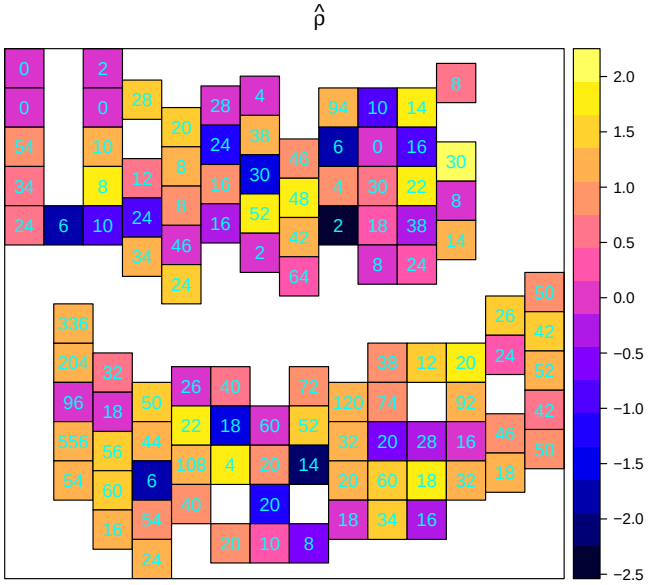


Fig. 6.4 Map of the estimated spatial coefficients ρ. Numbers of links generated by the neighbourhood structures are labelled on the study plot. study plots with zero links are set to have $\rho = 0$.

6.3.2 Mixed effect analysis

As suggested by the above analysis, we now assume our final model to have a spatial error term that is homogeneous all over the experimental area with a spatial weight matrix $\mathbf{W}_e$ defined as in equation (6.8) and a spatially correlated random intercept to account for the plot effect, with covariance matrix defined as in equation (6.2). The likelihood of the model is in closed form as shown in equation (6.9); but the evaluation of the likelihood involves calculating the determinant of $\mathbf{\Sigma}_y$ with size of 7032×7032 and this can be numerically unstable especially when the matrix is close to singular and also is very slow. Even with independence assumptions between the two experiment blocks, which makes $\mathbf{\Sigma}_y$ block diagonal of sizes 4741×4741 and 2291×2291, the amount of time is still large and the optimisation procedure easily breaks down for strong spatial effects. In the following estimation we use the Monte Carlo likelihood as an approximation and optimise it by the response surface design method as described in Chapter 4. The exact result is also calculated for comparison.

Three different RSM schemes were used and the fastest is used for further optimisation. The starting point is chosen to be the estimates from the linear mixed effects model with independent plot effects and errors. All of the three RSM schemes try to update the covariance parameters by profiling out or conditioning on the value of $\boldsymbol{\beta}$. The "**rsm1**" scheme uses the Monte Carlo profile likelihood function and updates the four parameters simultaneously while the "**rsm2**" updates the parameters by approximating two smaller systems of (ρ_e, σ_e^2) and (ρ_p, σ_p^2) with the other parameter values fixed. Although profiling out $\boldsymbol{\beta}$ reduces the dimension of the parameter space, the profile likelihood is much slower to evaluate than the Monte Carlo likelihood and less accurate due to the error of solving the non-linear system for $\boldsymbol{\beta}$. Meanwhile since the value of $\boldsymbol{\beta}$ does not change much for different covariance matrix, we use the conditional Monte Carlo likelihood in "**rsm3**" where

the $\boldsymbol{\beta}$ value are fixed at the importance distribution value for the Monte Carlo likelihood function and the system updates all four covariance parameters in each iteration. The RSM iterations terminate when any of the following events happens:

1. Maximum number of iterations reached.

2. The approximated system has a stationary point within design area.

3. The Monte Carlo likelihood evaluated at the boundary of the design area is close enough to zero. We used a tolerance of 6 for the two parameter system and 10 for the four parameter system since they are close to the 5% critical value for a likelihood ratio test with degrees of freedom equal to 2 and 4.

4. The total time elapsed is more than the maximum allowed.

Table 6.2 shows the estimation results using RSM and Monte Carlo likelihood. We compare the estimates and total elapsed time with those of a direct optimisation of the exact likelihood and profile likelihood functions with starting point the same as the beginning of the RSM.

The first three rows of Table 6.2 shows the results for the three RSM schemes. All systems end up with similar stationary points within only a few iterations and "**rsm3**" is the fastest procedure as the evaluation of the Monte Carlo likelihood is much faster than that of the Monte Carlo profile likelihood. Even with the very coarse approximation of RSM, the stationary point is already very close to the maxima found by directly optimising the exact likelihood functions, as shown in the bottom rows in the table, and the computation time was reduced from more than a day to less than an hour.

The four rows in the middle of Table 6.2 improve the estimates based on "**rsm3**" by directly optimising the Monte Carlo likelihood (mc.l), Monte Carlo profile likelihood (mc.pl), exact likelihood (e.l) and exact profile likelihood (e.pl) function with the "**rms3**" output as the starting point. The Monte Carlo likelihoods are

Table 6.2 Estimation results of the mixed effect model for the Sabah biodiversity experiment data set

	time (hrs)	n.iter	ρ_e	ρ_p	$\sigma_e^2(10^{-3})$	$\sigma_p^2(10^{-4})$
rsm1	2.24	5	1.021	0.2682	4.007	3.257
rsm2	2.22	6	1.024	0.2985	4.015	3.346
rsm3	0.67	12	1.024	0.2977	4.013	3.293
rsm3 + mc.l	1.14	72	1.063	0.2980	4.047	3.298
rsm3 + mc.pl	2.03	23	1.085	0.2977	3.978	3.298
rsm3 + e.l	43.65	22	1.078	0.2988	3.992	3.330
rsm3 + e.pl	11.27	12	1.078	0.2994	3.993	3.307
exact.l	62.35	42	1.078	0.2989	3.993	3.325
exact.pl	27.08	40	1.078	0.2991	3.992	3.336

* Three different updating schemes were used in the response surface design and details to be found in text. With the rsm outputs as starting point, the Monte Carlo likelihood (mc.l), profile likelihood (mc.pl) and the exact likelihood (e.l) and profile likelihood (e.pl) functions were optimised. Results of optimising the exact likelihood (exact.l) and profile likelihood (exact.pl) with initial value assuming no spatial and plot effects are also included for comparison.

iteratively optimised until the increase in likelihood value is smaller than $1/\sqrt{s}$ where s is the Monte Carlo sample size (with $s = 100$). The optimization procedures of both the Monte Carlo likelihoods converged in two steps[1] and provide very close results to the exact ones with little additional time. Also, with the good initial point, the computational time is much reduced for optimising the exact likelihood functions as shown in the last two rows in the middle of the table.

With the estimated covariance parameters from the above table, we now compare the estimates of the linear coefficients $\boldsymbol{\beta}$ in the two models in table 6.3 below. The number in the brackets following the estimates are 2 times of the standard error. The first column are results from the linear mixed effect model with independent plot effects and the second and third columns are results for a mixed effect model with spatial error term and CAR plot effects. The second column are estimates

[1]The number of iterations shown in the table are the sum of iterations used in the two optimisation steps.

from "**rsm3** + mc.l" in table 6.2 and the last column is the exact MLE. In general the Monte Carlo MLE and the exact MLE show very similar results; therefore, the Monte Carlo MLE can be used as a fast approximation of the exact one. When including the spatial correlation, the estimates of $\boldsymbol{\beta}$ stay similar to the linear mixed effect model estimates but the standard errors for all `species` have increased, which makes some levels not so significantly different from the baseline level, Shorea argentifolia; meanwhile the standard errors of the `treatment` decreased which resulted in the growth rate in plots with 16 mixed species being estimated as significantly higher than the baseline level but not the rate in the plots with only 4 mixed species.

Table 6.3 Estimation results of $\boldsymbol{\beta}$

	lme	sp.mc	sp.exact
Intercept	0.169 (0.021)	0.163 (0.022)	0.166 (0.020)
Shorea beccariana	-0.028 (0.021)	-0.025 (0.027)	-0.028 (0.020)
Dipterocarpus conformis	-0.081 (0.020)	-0.079 (0.028)	-0.081 (0.019)
Shorea faguetiana	-0.035 (0.025)	-0.033 (0.026)	-0.036 (0.023)
Hopea ferruginea	-0.012 (0.022)	-0.007 (0.025)	-0.009 (0.021)
Shorea gibbosa	-0.043 (0.022)	-0.039 (0.026)	-0.041 (0.021)
Shorea johorensis	-0.041 (0.021)	-0.039 (0.033)	-0.042 (0.020)
Dryobalanops lanceolata	-0.043 (0.020)	-0.041 (0.029)	-0.043 (0.019)
Shorea leprosula	-0.026 (0.023)	-0.023 (0.033)	-0.026 (0.022)
Shorea macrophylla	-0.052 (0.022)	-0.051 (0.029)	-0.054 (0.021)
Shorea macroptera	-0.062 (0.021)	-0.059 (0.032)	-0.062 (0.020)
Parashorea malaanonan	-0.068 (0.021)	-0.064 (0.023)	-0.067 (0.020)
Shorea ovalis	-0.061 (0.020)	-0.060 (0.030)	-0.062 (0.019)
Shorea parvifolia	-0.025 (0.023)	-0.022 (0.036)	-0.025 (0.022)
Hopea sangal	-0.049 (0.020)	-0.045 (0.030)	-0.048 (0.019)
Parashorea tomentella	-0.077 (0.020)	-0.075 (0.028)	-0.077 (0.019)
treatment.4	0.001 (0.013)	0.004 (0.004)	0.005 (0.010)
treatment.16	0.010 (0.011)	0.012 (0.004)	0.013 (0.009)

Figure 6.5 shows the map of the estimated CAR plot effects. Compared to the independent plot effects estimates in Figure 6.2, the scale is reduced by ten and the map looks smoother. The plot effects account for the variability between the study plots while the fixed effects describe the population mean for each species

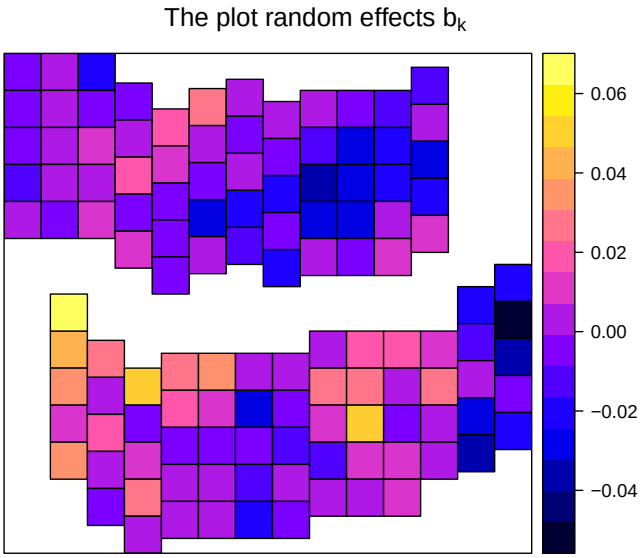


Fig. 6.5 Map of the estimated CAR plot effects $\tilde{\boldsymbol{b}}$.

and treatment; hence the predicted growth rates tend to be more homogeneous compared to those from the within-plot analysis in section 6.3.1. This is shown in Figure 6.6 where we plot the predicted growth rate for one of the species Hopea sangal with and without the random effect. The left panel shows the population mean of the Hopea sangal with different treatments labelled on each square and the right panel shows the prediction after adding the estimated random effects.

6.4 Conclusion

In this study we fitted linear mixed effects models to the growth rate on different species and environmental factors and we compared the fitted results of models with and without spatial error and spatial plot effects. Both models suggest similar conclusions to those in the previous work mentioned at the beginning of the chapter that different species tend to have different growth rates and the growth rate tends to increase when the environment is more bio-diversified.

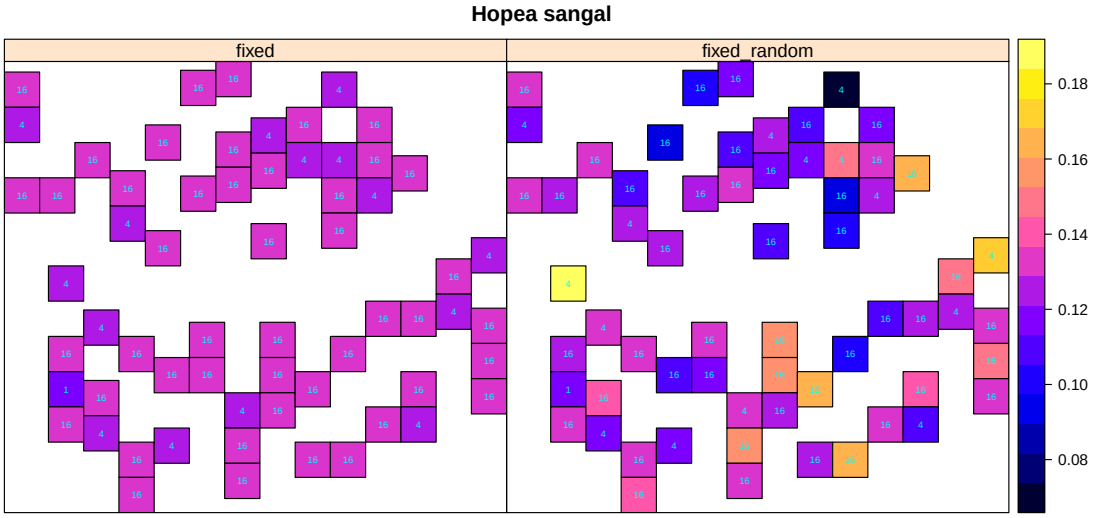


Fig. 6.6 Map of the predicted growth rate of Hopea sangal.

By considering the spatial information of the data set, we also found that there is both strong spatial correlations between the trees close in distance and between plots adjacent to each other. The log-likelihood ratio between the model with and without spatial error and spatial plot effects is as large as 7000 and with only a difference of 2 for a degree of freedom this suggests that the model with spatial effects fits much better than the simple linear mixed effects model. Both the coefficients of the spatial error ρ_e and spatial plot effect ρ_p are positive and close to the largest parameter values for a valid CAR model ($\rho_e \in (-1.45, 1.45)$ and $\rho_p \in (-0.3, 0.3)$) which means the spatial correlation is very strong.

The estimates of the growth rates remain similar to the linear mixed effects model when considering the spatial effects but for some species the difference become less statistically significant as the variances of the some estimates become larger. However for the variable `treatment` that represents the number of species in each study plots, the estimates become slightly larger and the variance smaller; this again confirms our previous conclusion on the effect of biodiversity.

With this application, we also demonstrated that for a CAR model with intractable likelihood due to the large and complicated covariance matrix, the Monte Carlo likelihood can provide a good approximation to exact likelihood and RSM is fast and stable in finding the MC-MLE. The computational time is also reduced greatly by exploiting the sparsity of the CAR covariance matrix.

Chapter 7

Discussion

7.1 Major contributions

In this thesis, we presented two procedures for Monte Carlo maximum likelihood estimation of CAR models with reasonable computational cost. Similar iterative procedures have been mentioned or used in the literature (Geyer and Møller, 1994; Knudson, 2015; Christensen and Ribeiro, 2002; Giorgi and Diggle, 2015) but most assume a good starting point of the parameter is available to ensure a stable convergence and none presents a clearly defined framework for an automatic iterative procedure. A major contribution of this work is that our proposed algorithms are generally stable in converging to the optimum region regardless of the initial point; moreover by defining the convergence criteria according to the asymptotic Monte Carlo variance our iterative procedures are fully automated.

We have produced an R package `mclcar` for the two proposed methods and demonstrating the Monte Carlo likelihood methods for the CAR models. The R package is available on **CRAN** for public use with a vignette explaining the method and showing the usage of the functions. The user's manual is also included in Appendix C.

From our simulation studies in Chapter 4, both our proposed methods provide similar results that are close to the exact MLE. Starting from a point near the optimum, the algorithm based on RSM may be less accurate than directly optimising the Monte Carlo likelihood with the same number of iterations but it is much faster in locating the optimum region from a poor initial value given limited amount of computational resource. Therefore, it is recommended to combine the two procedures such that the RSM based one is used for initial search of the optimum region and then directly optimise the Monte Carlo likelihood.

Most existing frequentist methods for estimating the CAR models are not exactly likelihood based but use the quasi-likelihood, the pseudo-likelihood or the Laplace approximation to the likelihood. These methods may be fast to provide a good estimate but none of them are exact or unbiased approximation to the exact likelihood. The Monte Carlo likelihood method provide an alternative for estimating the CAR model from an frequentist approach and it provides unbiased approximation to the likelihood ratio. When the importance sampler value is close to the target maximum, the method also produces a 95% CI that has a good coverage of the exact MLE.

In a comparison study, We have applied the algorithms on a frequently used data set in literature and demonstrate that our proposed strategies give results consistent with those in the literature. In another example with thousands of spatial locations, we successfully implemented the two proposed procedure on the data set to provide a fast approximation to the MLE, where existing methods either failed to provide a result or did not appear to converge.

We have also used our methods on an unseen example with double layers of CAR structures in the covariance matrix. Estimation of such hierarchical structures are more commonly done by Bayesian approach, although not really included as a standard option in most existing packages; we have not found any work on the frequentist approach for estimating such models. This new example show that

our proposed procedures can be easily adapted to deal with more complicated covariance matrices.

7.2 Problems and future improvements

One problem of our methods is that they are not as fast as most frequentist approach for direct CAR models and small data sets since the methods are based on large number of simulations. For these problems, the Monte Carlo likelihood methods usually provide similar estimates to the others, so they can be used as complimentary to other methods. Moreover, the simulated samples can also be used for constructing Monte Carlo test for the log likelihood ratio statistics since in the last iteration, these samples are generated from a distribution with parameter values that are very close to the exact MLE. For latent CAR models and large data sets, we have shown in the second example in Chapter 5 that our proposed methods were able to provide good estimates in a reasonable amount of time when other methods failed.

Comparing the computational costs of the Monte Carlo likelihood and Bayesian approaches using simulation based methods is non-trivial, since both are simulation based methods that depends on the starting values, the number of iterations and the Monte Carlo sample size. For the Bayesian approaches, the mostly attacked issue is the subjective choice of prior. Inappropriate choice of the prior may not only affect the inference based on the posterior samples but also cause slow convergence of the Markov Chain when using the MCMC algorithms for sampling from the posteriors. For parameter estimation, we are more interested in the mode of the posterior. A full Bayesian approach may waste a lot of time sampling at the tail of the target distribution if not properly implemented, while for the Monte Carlo likelihood, the samples are driven to be drawn from the distribution with parameter close to the MLE. As the data size increases, the Bayesian estimators with uninformative prior

usually provide consistent results as the MLE. However in for the CAR model, one should always bear in mind that the number of data is usually fixed; therefore the prior distribution may have a strong influence on the posterior and the asymptotics do not apply.

The current implementation of the Monte Carlo likelihood methods in our package is not optimised. For example, in computing the Monte Carlo likelihood the gradient and Hessian of the Monte Carlo likelihood, calculations of the same set of statistics are repeated.This is redundant and wastes a lot of time in interchanging values between functions. We can improve it by either using a local environment to store the calculated statistics within the optimisation procedure, or modifying the optimisation function to accommodate the special structure of the Monte Carlo likelihood and its derivatives. Another example is the tuning of the controlling parameters in the iterative and RSM procedures. Further work can be done in finding optimal values or adaptive schemes for these values.

It should be noted that our methods are in fact not restricted to CAR models. We provide a general framework for iterative maximum Monte Carlo likelihood estimation and further work can be done to adapt the algorithms for general public use in practice. Apart from approximating the MLE, the RSM-based method can also be used to reconstruct the likelihood surface around the MLE and thus provide approximations to the confidence level sets. The approximation can be improved by combining all the importance samples from the last few iterations.

The problems discussed above are all beyond the primary goal of this thesis but deserve further study if our proposed methods are to be applied to more general problems.

References

Albert, J. (2014). *`LearnBayes`: Functions for Learning Bayesian Inference*. R package version 2.15, http://CRAN.R-project.org/package=LearnBayes.

Anselin, L. (1988). *Spatial Econometrics: Methods and Models*. NATO ASI Series. Series E, Applied Sciences. Springer.

Anselin, L. (2010). Thirty years of spatial econometrics. *Papers in Regional Science*, 89(1):3–25.

Attias, H. (2000). A variational Bayesian framework for graphical models. In *In Advances in Neural Information Processing Systems 12*, pages 209–215. MIT Press.

Banerjee, S., Carlin, B. P., and Gelfand, A. (2004). *Hierarchical Modeling and Analysis for Spatial Data*. London: Chapman & Hall/CRC.

Besag, J. E. (1972). Nearest-neighbour systems and the auto-logistic model for binary data. *Journal of the Royal Statistical Society. Series B (Methodological)*, 34(1):75–83.

Besag, J. E. (1974). Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society.Series B (Methodological)*, 36:192–236.

Besag, J. E. (1975). Statistical analysis of non-lattice data. *Journal of the Royal Statistical Society. Series D (The Statistician)*, 24(3):179–195.

Besag, J. E. and Kooperberg, C. (1995). On conditional and intrinsic autoregression. *Biometrika*, 82(4):733–746.

Besag, J. E., York, J., and Mollié, A. (1991). Bayesian image restoration, with two applications in spatial statistics. *Annals of the Institute of Statistical Mathematics*, 43(1):1–20.

Bivand, R., Pebesma, E. J., and Rubio, V. (2008). *Applied spatial data: analysis with R*. Use R! New York: Springer.

Blangiardo, M., Cameletti, M., Baio, G., and Rue, H. (2013). Spatial and spatio-temporal models with R-INLA. *Spatial and Spatio-temporal Epidemiology*, 4:33 – 49.

Bottou, L. (1998). Online algorithms and stochastic approximations. In Saad, D., editor, *Online Learning and Neural Networks*. Cambridge University Press, Cambridge, UK.

Box, G. E. P. and Draper, N. R. (2007). *Response Surfaces, Mixtures, and Ridge Analyses*. Wiley, New York.

Box, G. E. P. and Wilson, K. B. (1951). On the experimental attainment of optimum conditions. *Journal of the Royal Statistical Society. Series B (Methodological)*, 13(1):1–45.

Breslow, N. E. and Clayton, D. G. (1993). Approximate inference in generalized linear mixed models. *Journal of the American Statistical Association*, 88(421):9–25.

Cappé, O., Douc, R., Moulines, E., and Robert, C. (2002). On the convergence of the Monte Carlo maximum likelihood method for latent variable models. *Scandinavian Journal of Statistics*, 29(4):615–635.

Celeux, G. and Diebolt, J. (1992). A stochastic approximation type EM algorithm for the mixture problem. *Stochastics and Stochastic Reports*, 41(1-2):119–134.

Chen, M.-H. and Shao, Q.-M. (1997). On Monte Carlo methods for estimating ratios of normalizing constants. *The Annals of Statistics*, 25(4):1563–1594.

Christensen, O. F. and Ribeiro, P. J. (2002). `geoRglm` - a package for generalised linear spatial models. *R-NEWS*, 2(2):26–28. http://cran.R-project.org/doc/Rnews.

Chung, K. L. (1954). On a stochastic approximation method. *The Annals of Mathematical Statistics*, 25(3):463–483.

Clayton, D. G. and Kaldor, J. (1987). Empirical bayes estimates of age-standardized relative risks for use in disease mapping. *Biometrics*, 43(3):671–681.

Cressie, N. A. C. (1993). *Statistics for Spatial Data*. Wiley, revised ed. edition.

Delyon, B., Lavielle, M., and Moulines, E. (1999). Convergence of a stochastic approximation version of the EM algorithm. *The Annals of Statistics*, 27(1):94–128.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B*, 39(1):1–38.

Diggle, P. J. and Ribeiro, P. J. (2007). *Model-based Geostatistics* . Springer series in statistics. Springer.

Evans, M. and Swartz, T. (2000). *Approximating Integrals via Monte Carlo and Deterministic Methods*. OUP Oxford.

Fortin, M. and Dale, M. (2005). *Spatial Analysis: A Guide for Ecologists*. Cambridge University Press.

Gelman, A. and Meng, X.-L. (1998). Simulating normalizing constants: From importance sampling to bridge sampling to path sampling. *Statistical Science*, 13(2):163–185.

Getis, A., Mur, J., and Zoller, H., editors (2004). *Spatial Econometrics and Spatial Statistics*. Applied Econometrics Association Series. Palgrave Macmillan.

Geyer, C. J. (1994). On the convergence of Monte Carlo maximum likelihood calculations. *Journal of the Royal Statistical Society. Series B (Methodological)*, 56(1):261–274.

Geyer, C. J. (2015). *glmm: Trust Region Optimiztion*. http://CRAN.R-project.org/package=trust.

Geyer, C. J. and Møller, J. (1994). Simulation procedures and likelihood inference for spatial point processes. *Scandinavian Journal of Statistics*, 21(4):359–373.

Geyer, C. J. and Thompson, E. A. (1992). Constrained Monte Carlo maximum likelihood for dependent data. *Journal of the Royal Statistical Society. Series B (Methodological)*, 54(3):657–699.

Giorgi, E. and Diggle, P. J. (2015). *PrevMap: Geostatistical Modelling of Spatially Referenced Prevalence Data*. R package version 1.1.4, http://CRAN.R-project.org/package=PrevMap.

Golding, N. (2013). *Mapping and understanding the distributions of potential vector mosquitoes in the UK: New methods and applications*. PhD thesis, University of Oxford.

Green, P. J. (1992). Discussion on "Constrained Monte Carlo maximum likelihood for dependent data" by C. J. Geyer and E. A. Thompson. *Journal of the Royal Statistical Society. Series B (Methodological)*, 54(3):683–684.

Haining, R. (2003). *Spatial Data Analysis: Theory and Practice*. Cambridge University Press.

Hammersley, J. and Clifford, P. (1971). Markov fields on finite graphs and lattices. unpublished manuscript.

Hector, A., Philipson, C., Saner, P., Chamagne, J., Dzulkifli, D., O'Brien, M., Snaddon, J. L., Ulok, P., Weilenmann, M., Reynolds, G., and Godfray, H. C. J. (2011). The Sabah Biodiversity Experiment: a long-term test of the role of tree diversity in restoring tropical forest structure and functioning. *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, 366(1582):3303–3315.

Henningsen, A. and Toomet, O. (2011). `maxLik`: A package for maximum likelihood estimation in R. *Computational Statistics*, 26(3):443–458. http://dx.doi.org/10.1007/s00180-010-0217-1.

Khuri, A. I. (2006). *Response Surface Methodology and Related Topics*. World Scientific.

Khuri, A. I. and Mukhopadhyay, S. (2010). Response surface methodology. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(2):128–149.

Kloeden, P. E. and Platen, E. (1992). *Numerical Solution of Stochastic Differential Equations*. Applications of Mathematics : Stochastic Modelling and Applied Probability. Springer.

Knudson, C. (2015). *glmm: Generalized Linear Mixed Models via Monte Carlo Likelihood Approximation*. http://CRAN.R-project.org/package=glmm.

Lee, D. (2013). `CARBayes`: An R package for Bayesian spatial modeling with conditional autoregressive priors. *Journal of Statistical Software*, 55(13):1–24. http://www.jstatsoft.org/v55/i13/.

Lee, D. and Mitchell, R. (2012). Boundary detection in disease mapping studies. *Biostatistics*, 13:415–426.

Lee, Y. and Nelder, J. A. (1996). Hierarchical generalized linear models. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(4):619–678.

Lenth, R. V. (2009). Response-surface methods in R, using `rsm`. *Journal of Statistical Software*, 32(7):1–17. http://www.jstatsoft.org/v32/i07/.

Lindgren, F. and Rue, H. (2015). Bayesian spatial modelling with R-INLA. *Journal of Statistical Software*, 63(1):1–25.

Lindsay, B. G. (1988). Composite likelihood methods. *Contemporary Mathematics*, 80:220–239.

Marin, J., Pudlo, P., Robert, C. P., and Ryder, R. J. (2012). Approximate Bayesian computational methods. *Statistics and Computing*, 22(6):1167–1180.

Martin, R. J. (1979). A subclass of lattice processes applied to a problem in planar sampling. *Biometrika*, 66(2):209–217.

Mase, S. (1995). Consistency of the maximum pseudo-likelihood estimator of continuous state space Gibbsian processes. *The Annals of Applied Probability*, 5(3):603–612.

Mathéron, G. (1963). Principles of geostatistics. *Economic Geology*, 58(8):1246–1266.

Meng, X.-L. and Schilling, S. (2002). Warp bridge sampling. *Journal of Computational and Graphical Statistics*, 11(3):pp. 552–586.

Myers, R., Montgomery, D., and Anderson-Cook, C. (2009). *Response Surface Methodology: Process and Product Optimization Using Designed Experiments*. Wiley Series in Probability and Statistics. Wiley.

Ord, J. K. (1975). Estimation methods for models of spatial interaction. *Journal of the American Statistical Association*, 70(349):120–126.

Preston, C. D., Pearman, D. A., and Dines, T. D. (2002). *New Atlas of the British and Irish flora*. Oxford University Press.

Pritchard, J. K., Seielstad, M. T., Perez-Lezaun, A., and Feldman, M. W. (1999). Population growth of human Y chromosomes: a study of Y chromosome microsatellites. *Molecular Biology and Evolution*, 16(12):1791–1798.

R Core Team (2016). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Ripley, B. D. (1981). *Spatial Statistics*. Wiley.

Ripley, B. D. (1987). *Stochastic Simulation*. Wiley.

Ripley, B. D. (1991). *Statistical Inference for Spatial Processes*. Cambridge University Press.

Robbins, H. and Monro, S. (1951). A stochastic approximation method. *The Annals of Mathematical Statistics*, 22(3):400–407.

Robert, C. and Casella, G. (2005). *Monte Carlo Statistical Methods*. Springer Texts in Statistics. Springer New York.

Roberts, G. O. and Rosenthal, J. S. (2001). Optimal scaling for various Metropolis-Hastings algorithms. *Statistical Science*, 16(4):351–367.

Roberts, G. O. and Tweedie, R. L. (1996). Exponential convergence of Langevin distributions and their discrete approximations. *Bernoulli*, 2(4):341–363.

Ronnegard, L., Shen, X., and Alam, M. (2010). `hglm`: A package for fitting hierarchical generalized linear models. *The R Journal*, 2(2):20–28. http://journal.r-project.org/archive/2010-2/RJournal_2010-2_Roennegaard~et~al.pdf.

Rousset, F. and Ferdy, J.-B. (2014). Testing environmental and genetic effects in the presence of spatial autocorrelation. *Ecography*, 37(8):781–790.

Rubin, D. B. (1984). Bayesianly justifiable and relevant frequency calculations for the applied statistician. *The Annals of Statistics*, 12(4):1151–1172.

Rue, H. (2001). Fast sampling of Gaussian Markov random fields. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 63(2):325–338.

Rue, H., Follestad, T., Wist, H., and Martino, S. (2002). *`GMRFLib`: a C-library for fast and exact simulation of Gaussian Markov random fields*. https://www.math.ntnu.no/~hrue/GMRFLib/doc/html/main.html.

Rue, H. and Held, L. (2005). *Gaussian Markov Random Fields: Theory and Applications*, volume 104 of *Monographs on Statistics and Applied Probability*. Chapman & Hall, London.

Rue, H., Martino, S., and Chopin, N. (2009). Approximate Bayesian inference for latent Gaussian models by using integrated nested Laplace approximations. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 71(2):319–392.

Sacks, J. (1958). Asymptotic distribution of stochastic approximation procedures. *The Annals of Mathematical Statistics*, 29(2):373–405.

Saner, P., Philipson, C., Ong, R. C., Majalap, N., Egli, S., and Hector, A. (2010). Positive effects of ectomycorrhizal colonization on growth of seedlings of a tropical tree across a range of forest floor light conditions. *Plant and Soil*, 338(1):411–421.

Saner, P. G. (2009). *Ecosystem carbon dynamics in logged forest of Malaysian Borneo*. PhD thesis, University of Zurich.

Santner, T. J., Williams, B., and Notz, W. (2003). *The Design and Analysis of Computer Experiments*. Springer-Verlag.

Schall, R. (1991). Estimation in Generalized Linear Models with Random Effects. *Biometrika*, 78(4):719–727.

Stein, M. L., Chi, Z., and Welty, L. J. (2004). Approximating likelihoods for large spatial data sets. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 66(2):275–296.

Tavaré, S., Balding, D. J., Griffiths, R. C., and Donnelly, P. (1997). Inferring coalescence times from DNA sequence data. *Genetics*, 145(2):505–518.

Tierney, L. and Kadane, J. B. (1986). Accurate approximations for posterior moments and marginal densities. *Journal of the American Statistical Association*, 81(393):82–86.

Torrie, G. M. and Valleau, J. P. (1977). Nonphysical sampling distributions in Monte Carlo free-energy estimation: Umbrella sampling. *Journal of Computational Physics*, 23(2):187–199.

Varin, C., Reid, N., and Firth, D. (2011). An overview of composite likelihood methods. *Statistica Sinica*, 21:5–42.

Vecchia, A. V. (1988). Estimation and model identification for continuous spatial processes. *Journal of the Royal Statistical Society. Series B (Methodological)*, 50(2):297–312.

Waller, L. and Gotway, C. (2004). *Applied Spatial Statistics for Public Health Data*. Wiley Series in Probability and Statistics. Wiley.

Wei, G. C. G. and Tanner, M. A. (1990). A Monte Carlo implementation of the EM algorithm and the poor man's data augmentation algorithms. *Journal of the American Statistical Association*, 85(411):699–704.

Welling, M. and Teh, Y. W. (2011). Bayesian learning via stochastic gradient Langevin dynamics. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 681–688.

Whittle, P. (1954). On stationary processes in the plane. *Biometrika*, 41(3/4):434–449.

Appendix A

Some details and examples of CAR models

A.1 The unimodality of the likelihood for the direct CAR likelihood

We prove the unimodality of the likelihood for the direct CAR models and show an example of simulated data on a torus. We start from a direct CAR model with known mean.

Proposition A.1.1. *The likelihood of the data from a direct CAR model with distribution* $\mathcal{N}(0, \sigma^2(I - \rho\mathbf{W})^{-1})$ *has a single mode.*

Proof. Suppose the observed data $\mathbf{y}$ is from $\mathcal{N}(0, \sigma^2(\mathbf{I}-\rho\mathbf{W})^{-1})$, then the likelihood function is

$$\ell \propto -\frac{N}{2}\log\sigma^2 + \frac{1}{2}\log(1-\rho\lambda_i) - \frac{\mathbf{y}^T(I-\rho\mathbf{W})\mathbf{y}}{2\sigma^2} \tag{A.1}$$

where $\lambda_1 < \cdots < \lambda_i < \cdots < \lambda_N$ are the ordered eigenvalues of $\mathbf{W}$. The gradient $\nabla \ell$ has the following entries

$$\begin{cases} \dfrac{\partial \ell}{\partial \rho} = -\dfrac{1}{2}\displaystyle\sum_{i=1}^{N} \dfrac{\lambda_i}{1-\rho\lambda_i} + \dfrac{\mathbf{y}^T\mathbf{W}\mathbf{y}}{2\sigma^2} \\ \\ \dfrac{\partial \ell}{\partial \sigma^2} = -\dfrac{N}{2\sigma^2} + \dfrac{\mathbf{y}^T(I-\rho\mathbf{W})\mathbf{y}}{2(\sigma^2)^2} \end{cases} \tag{A.2}$$

and the Hessian matrix H has the following entries

$$\begin{cases} \dfrac{\partial^2 \ell}{\partial \rho^2} = -\dfrac{1}{2}\displaystyle\sum_{i=1}^{N} (\dfrac{\lambda_i}{(1-\rho\lambda_i)})^2 \\ \\ \dfrac{\partial^2 \ell}{\partial \rho \partial \sigma^2} = -\dfrac{\mathbf{y}^T\mathbf{W}\mathbf{y}}{2(\sigma^2)^2} \\ \\ \dfrac{\partial^2 \ell}{(\partial \sigma^2)^2} = \dfrac{N}{2(\sigma^2)^2} - \dfrac{\mathbf{y}^T(\mathbf{I}-\rho\mathbf{W})\mathbf{y}}{(\sigma^2)^3} \end{cases} \tag{A.3}$$

By letting $\nabla \ell = 0$ we have

$$\begin{cases} \hat{\sigma}^2 = \dfrac{\mathbf{y}^T(\mathbf{I}-\hat{\rho}\mathbf{W})\mathbf{y}}{N} \\ \\ \displaystyle\sum_{i=1}^{N} \dfrac{\lambda_i}{1-\hat{\rho}\lambda_i} = \dfrac{\mathbf{y}^T\mathbf{W}\mathbf{y}}{\hat{\sigma}^2} \end{cases} \tag{A.4}$$

For the likelihood to have a unique maximum within the given parameter space, we need to prove that

1. The system (A.4) has a unique solution, which requires

$$\sum_{i=1}^{N} \frac{\lambda_i}{1-\rho\lambda_i} = \frac{N\mathbf{y}^T\mathbf{W}\mathbf{y}}{\mathbf{y}^T(\mathbf{I}-\rho\mathbf{W})\mathbf{y}} \tag{A.5}$$

has a unique root within the interval $\rho \in (1/\lambda_1, 1/\lambda_N)$.

2. The negative Hessian matrix $-\boldsymbol{H}$ evaluated at the solution of $\nabla\ell = 0$ is positive definite.

The second is easier to show by using the identities in (A.4). Obviously $|-\boldsymbol{H}_{11}| = \frac{1}{2}\sum_{i=1}^{N}(\frac{\lambda_i}{(1-\hat{\rho}\lambda_i)})^2 > 0$ and

$$|-\boldsymbol{H}| = \frac{N^2}{4(\hat{\sigma}^2)^2}\left[\frac{1}{N}\sum_{i=1}^{N}\left(\frac{\lambda_i}{1-\hat{\rho}\lambda_i}\right)^2 - \left(\frac{1}{N}\sum_{i=1}^{N}\frac{\lambda_i}{1-\lambda_i\hat{\rho}}\right)^2\right] \geq 0 \tag{A.6}$$

by using Jensen's inequality on the convex function x^2. Hence $-\boldsymbol{H}$ is positive definite at $(\hat{\rho}, \hat{\sigma}^2)$ if the solution exists and such solutions are all local maxima.

Now we prove the maximum is unique. This can be done by showing the polynomial function in the equation (A.5)

$$f(\rho) = -\sum_{i=1}^{N}\frac{\lambda_i}{1-\rho\lambda_i} + \frac{N\mathbf{y}^T\mathbf{W}\mathbf{y}}{y^T(\mathbf{I}-\rho\mathbf{W})\mathbf{y}} \tag{A.7}$$

has one root within $(1/\lambda_1, 1/\lambda_N)$. Note that $f(\rho)$ can be written as

$$f(\rho) = -\sum_{i=1}^{N}\frac{\lambda_i}{1-\rho\lambda_i} + N\frac{\bar{\lambda}}{1-\bar{\lambda}\rho} \tag{A.8}$$

where $\bar{\lambda} = \sum_i(m_{ii}^2 y_i^2)\lambda_i / \sum_i y_i^2, \mathbf{W} = \boldsymbol{M}^T\boldsymbol{\Lambda}\boldsymbol{M}$ and $\boldsymbol{M}$ is orthogonal if we assume $\mathbf{W}$ to be real and symmetric. Hence $\alpha_i = m_{ii}^2 y_i^2 / \sum_i y_i^2 \geq 0,\ \sum_i \alpha_i = 1$ can be viewed as weights and $\lambda_1 \leq \bar{\lambda} \leq \lambda_N$. It is easy to check that when $\bar{\lambda} = \lambda_1,\ f(\rho) < 0$ for all $\rho \in (1/\lambda_1, 1/\lambda_N)$ and $f(\rho) > 0$ when $\bar{\lambda} = \lambda_N$; however, either case has probability zero as it requires $\mathbf{y} = (1, 0, \ldots, 0)$ or $\mathbf{y} = (0, \ldots, 0, 1)$. So now we prove that $f(\rho)$ has a unique root in $(1/\lambda_1, 1/\lambda_N)$ for $\lambda_1 < \bar{\lambda} < \lambda_N$. Denote by $t(\rho) = -\sum_{i=1}^{N}\frac{\lambda_i}{1-\rho\lambda_i}$, then

$$t'(\rho) = -\sum_{i=1}^{N}(\frac{\lambda_i}{1-\rho\lambda_i})^2 > 0 \tag{A.9}$$

Since

$$t(0) = \sum_i \lambda_i = \text{trace}(\mathbf{W}) = 0 \tag{A.10}$$

we have $t(\rho) \geq 0$ when $\rho \geq 0$ and $t(\rho) < 0$ when $\rho < 0$. It is also easy to check that

$$\begin{cases} \lim_{\rho \to \frac{1}{\lambda_1}+} t(\rho) = -\infty \\ \lim_{\rho \to \frac{1}{\lambda_N}-} t(\rho) = +\infty \end{cases} \tag{A.11}$$

Since the second term in $f(\rho)$ is bounded, we have

$$\begin{cases} \lim_{\rho \to \frac{1}{\lambda_1}+} f(\rho) = +\infty \\ \lim_{\rho \to \frac{1}{\lambda_N}-} f(\rho) = -\infty \end{cases} \tag{A.12}$$

Since $f(\rho)$ is continuous on $(1/\lambda_1, 1/\lambda_N)$, there is at least one $\rho^* \in (1/\lambda_1, 1/\lambda_N)$ such that $f(\rho^*) = 0$. Next we show such ρ^* is unique within the interval. We wish to show that $f'(\rho)$ does not change sign within this interval; however looking at

$$f'(\rho) = -\sum_i \left(\frac{\lambda_i}{1-\lambda_i\rho}\right)^2 + N\left(\frac{\bar{\lambda}}{1-\bar{\lambda}\rho}\right)^2 \tag{A.13}$$

$$= -\frac{1}{N}\left[\sum_i \left(\frac{\lambda_i}{1-\lambda_i\rho}\right)^2 - \left(\sum_i \frac{\lambda_i}{1-\lambda_i\rho}\right)^2\right] + \frac{1}{N}\left[\left(\frac{N\bar{\lambda}}{1-\bar{\lambda}\rho}\right)^2 - \left(\sum_i \frac{\lambda_i}{1-\lambda_i\rho}\right)^2\right] \tag{A.14}$$

$$< \frac{1}{N} f(\rho)\left[\frac{N\bar{\lambda}}{1-\bar{\lambda}\rho} + \sum_i \frac{\lambda_i}{1-\lambda_i\rho}\right] \tag{A.15}$$

it is possible for $f'(\rho)$ to change sign when ρ is around zero while it being negative for the rest. Therefore, we try to show ρ^* is not in the region where $f'(\rho) \geq 0$.

Now If $f'(\rho) \geq 0$ we have

$$\left(\frac{N\bar{\lambda}}{1-\bar{\lambda}\rho}\right)^2 \geq N\sum\left(\frac{\lambda_i}{1-\lambda_i\rho}\right)^2 > \left(\sum \frac{\lambda_i}{1-\lambda_i\rho}\right)^2 \tag{A.16}$$

by using Jensen's inequality on convex function x^2, hence we have

$$\left|\frac{N\bar{\lambda}}{1-\bar{\lambda}\rho}\right| > \left|\sum \frac{\lambda_i}{1-\lambda_i\rho}\right| \tag{A.17}$$

If $\bar{\lambda} < 0$, then $1/\bar{\lambda} < 1/\lambda_1 < 0$ and $\frac{N\bar{\lambda}}{1-\bar{\lambda}\rho} < 0$. So $f(\rho) < 0$ for $\rho \in [0, \frac{1}{\lambda_N}) \cup \{\rho : f'(\rho) \geq 0\}$; and for $\rho < 0$ we always have $f'(\rho) < 0$ when $f(\rho) > 0$. Therefore we have the unique root $\rho^* \in (1/\lambda_1, 0)$.

Similarly, if $\bar{\lambda} \geq 0$, then $0 < 1/\lambda_1 < 1/\bar{\lambda}$ and $\frac{N\bar{\lambda}}{1-\bar{\lambda}\rho} > 0$. So $f(\rho) > 0$ for $\rho \in (\frac{1}{\lambda_1}, 0) \cup \{\rho : f'(\rho) \geq 0\}$; and for $\rho > 0$ we always have $f'(\rho) < 0$ when $f(\rho) < 0$. Therefore we have the unique root $\rho^* \in (0, 1/\lambda_N)$. □

Figure A.1 illustrates the two situations at the end of the above proof using a W from a 20×20 torus. We used the square root of the absolute values of the functions in the plot for better display without changing much of the shape of the functions.

Proposition A.1.2. *The likelihood of the data from a direct CAR model with distribution* $\mathcal{N}(\mathbf{X}\boldsymbol{\beta}, \sigma^2(\mathbf{I} - \rho\mathbf{W})^{-1})$ *has a single mode.*

When the mean is specified the proof is similar and we provide a sketch below.

Proof. The MLE of $\boldsymbol{\beta}$ should satisfy the following equation

$$\tilde{\boldsymbol{\beta}}(\rho) = (\mathbf{X}^T(\mathbf{I} - \rho\mathbf{W})\mathbf{X})^{-1}\mathbf{X}^T(\mathbf{I} - \rho\mathbf{W})\mathbf{Y} \triangleq \boldsymbol{F}\mathbf{Y} \tag{A.18}$$

Then we have $\mathbf{X}\tilde{\boldsymbol{\beta}} \sim \mathcal{N}(\mathbf{X}\boldsymbol{\beta}, \boldsymbol{\Sigma})$ and

$$\boldsymbol{\Sigma} = \mathbf{X}^T\mathrm{Var}(\tilde{\boldsymbol{\beta}})\mathbf{X} = (\boldsymbol{F}\mathbf{X})^T\mathrm{Var}(\mathbf{Y})(\boldsymbol{F}\mathbf{X}) = \mathrm{Var}(\mathbf{Y}) \tag{A.19}$$

Therefore, $\mathbf{X}\tilde{\boldsymbol{\beta}} \sim \mathcal{N}(\mathbf{X}\boldsymbol{\beta}, \sigma^2(\mathbf{I} - \rho\mathbf{W})^{-1})$. Now let $\mathbf{Z} = \mathbf{Y} - \mathbf{X}\tilde{\boldsymbol{\beta}}$, then $\mathbf{Z} \sim \mathcal{N}(0, 2\sigma^2(\mathbf{I} - \rho\mathbf{W})^{-1})$. Then Proposition A.1.1 follows. □

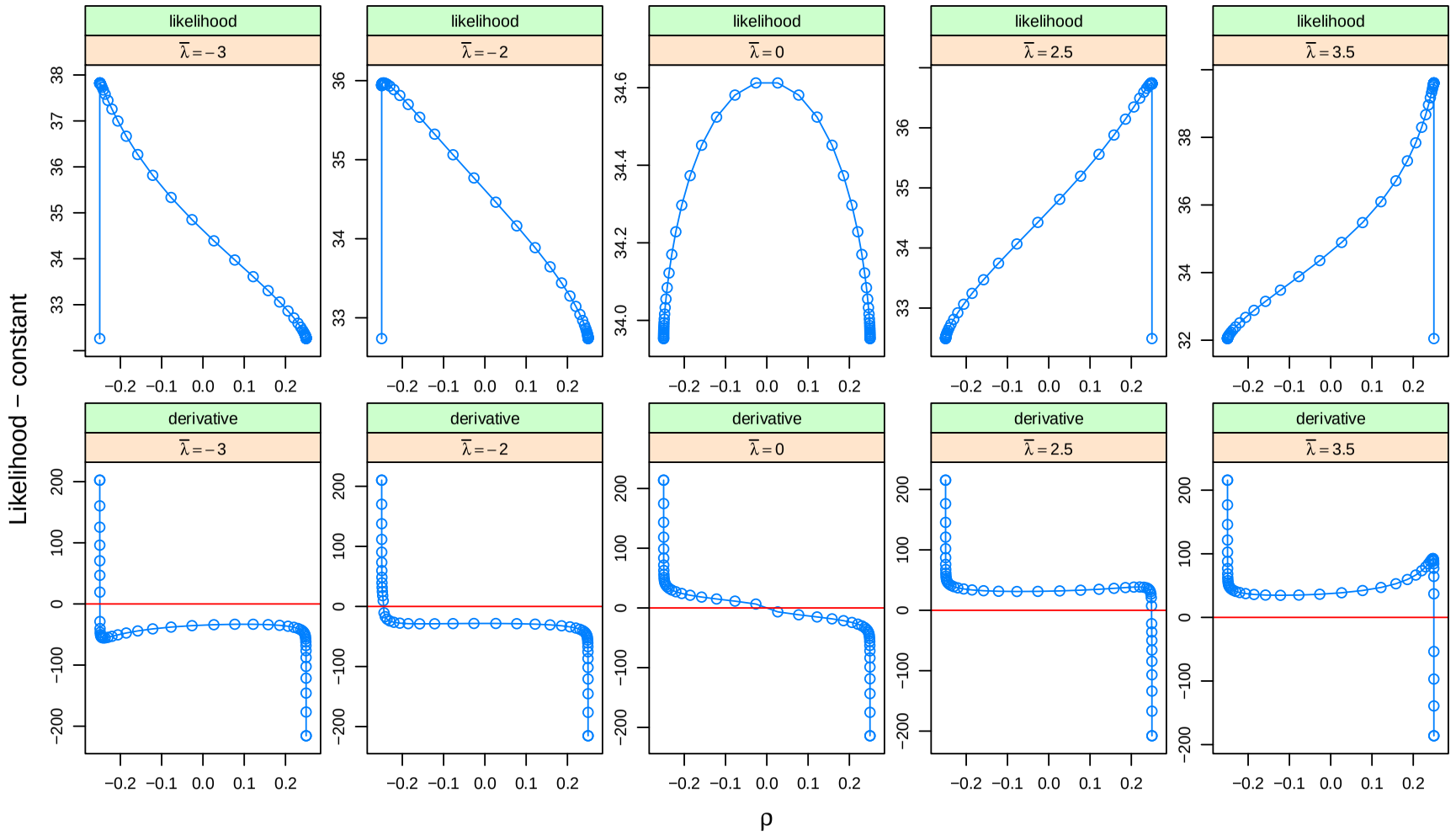


Fig. A.1 The shape of the likelihood function and its derivatives with different values of $\bar{\lambda}$. For $\bar{\lambda}$ close to the smallest and largest eigenvalue of the spatial weight matrix W, the likelihood function is not always concave over the entire parameter space. The absolute values are square-rooted for better display.

A.2 Valid parameter space

A.2.1 Range of ρ

For a direct CAR model with covariance matrix
$vSigma = \sigma^2(\mathbf{I} - \rho\mathbf{W})$, we must have $|\mathbf{\Sigma}| > 0$ which puts constraints on ρ. Since $|\mathbf{I} - \rho\mathbf{W}| = \prod_i^N (1 - \lambda_i\rho)$, the constraints become $1 - \rho\lambda_i > 0$ for every $i = 1, \ldots, N$.

When $\lambda_1\lambda_N < 0$

Suppose the first k ordered eigenvalues of W satisfy $\lambda_1 < \cdots < \lambda_k < 0$, then the range of constraints simplify to $1/\lambda_k < 1/\lambda_{k-1} < \cdots < 1/\lambda_1 < \rho < 0$, which gives $1/\lambda_1 < \rho < 0$. And similarly for $0 \leq \lambda_{k+1} < \cdots < \lambda_N$, we have $0 < \rho < 1/\lambda_N$. When one or a few of the λ_i are zero, the range of ρ is not affected unless all λ_i are zero, when $\rho \in \mathbb{R}$. Hence, when $\lambda_1\lambda_N < 0$, we have $\rho \in (1/\lambda_1, 1/\lambda_N)$.

When $\lambda_1\lambda_N \geq 0$

When $\lambda_i \leq 0$ for all $i = 1, \ldots, N$ this becomes $(1/\lambda_1, \infty)$. Similarly if $\lambda_i > 0$ for all $i = 1, \ldots, N$ we have $\rho \in (-\infty, 1/\lambda_N)$.

A.2.2 General constraints on parameters

In general, denote the precision matrix of the direct CAR model by $\boldsymbol{Q}(\boldsymbol{\theta})$, parametrized by some $\boldsymbol{\theta}$. We require $\boldsymbol{Q}(\boldsymbol{\theta})$ to be always symmetric and strictly positive definite. It is usually difficult to describe the valid parameter space $\mathbf{\Theta}^+$ within which such conditions holds and hence a frequently used sufficient condition is the *diagonal dominance* which restricts $\mathbf{\Theta}$ to some $\mathbf{\Theta}^{++}$ that is a subset of $\mathbf{\Theta}^+$.

A matrix $\boldsymbol{A}$ is called diagonal dominant if $|\boldsymbol{A}_{ii}| > \sum_{j:j\neq i} |\boldsymbol{A}_{ij}|$, $\forall i$.. A diagonal-dominant symmetric matrix with strictly positive diagonal entries is symmetric

positive definite (SPD). The diagonal dominant constraints can be too restrictive in some situations.

There are also general results for the relations between toruses with increasing size. Denote by $\mathbf{n} = (n_1, n_2)$ a vector that define the size of a $n_1 \times n_2$ torus. Roughly speaking, we can have $\mathbf{\Theta}^+_{\mathbf{n}} \supseteq \mathbf{\Theta}^+_{\mathbf{2n}} \supseteq \mathbf{\Theta}^+_{\mathbf{4n}} \supseteq \mathbf{\Theta}^+_{\mathbf{8n}} \dots$ but in general $\mathbf{\Theta}^+_{\mathbf{n}} \not\supseteq \mathbf{\Theta}^+_{\mathbf{n+1}}$. So we can define $\mathbf{\Theta}^+_{\infty} = \cap_{\mathbf{n}>\mathbf{2m}} \mathbf{\Theta}^+_{\mathbf{n}}$ and any configuration in this set is a valid parameter. Here $\mathbf{m}$ is the lag of dependence.

For a non-stationary CAR process on the lattice $\mathcal{I}_{\mathbf{n}'}$ with the same full conditional as the torus but different boundary conditions, we have the result that $\mathbf{\Theta}^+_{\mathbf{n}} \subseteq \mathbf{\Theta}^{+,\mathcal{I}}_{\mathbf{n}'}$ for $\mathbf{n}' < \mathbf{n} - \mathbf{m}$. Details of the results can be found in Section 2.7 of the book Rue and Held (2005).

A.3 Equivalent SAR and CAR models

A direct way to see the spatial relationships defined by the CAR and SAR models is to compare the precision matrices $\boldsymbol{Q}_C$ and $\boldsymbol{Q}_S$ which are the inverses of the covariance matrices in the joint distributions. The precision matrix shows the conditional dependency between each pair of nodes conditional on the rest, with zero entry indicating independent and dependent otherwise. A formal statement and proofs can be found in Rue and Held (2005, p.32). Thus, to have the CAR and SAR models equivalent, we require that their precision matrices to be the same. It can be shown that an SAR distribution can always be represented as a CAR distribution but not necessarily vice versa (Cressie, 1993; Ripley, 1981).

To understand such equivalence between SAR and CAR models, we define higher order neighbourhood structures for a given node on a lattice. In Figure A.2 we show the neighbours of the black node under different schemes. The four red nodes in Figure A.2(b) are the all first-order neighbours; the four blue nodes in Figure A.2(c) and A.2(d) are the second-order neighbours; and the four yellow

nodes in A.2(d) are the third-order neighbours. In general, the first-order CAR models on a lattice do not have their corresponding SAR models (Cressie, 1993, p.409).

In the following we describe two examples of SAR models on the regular grid that are equivalent to a second- and a third-order CAR models; more details of other examples can be found in (Ripley, 1981, p.93).

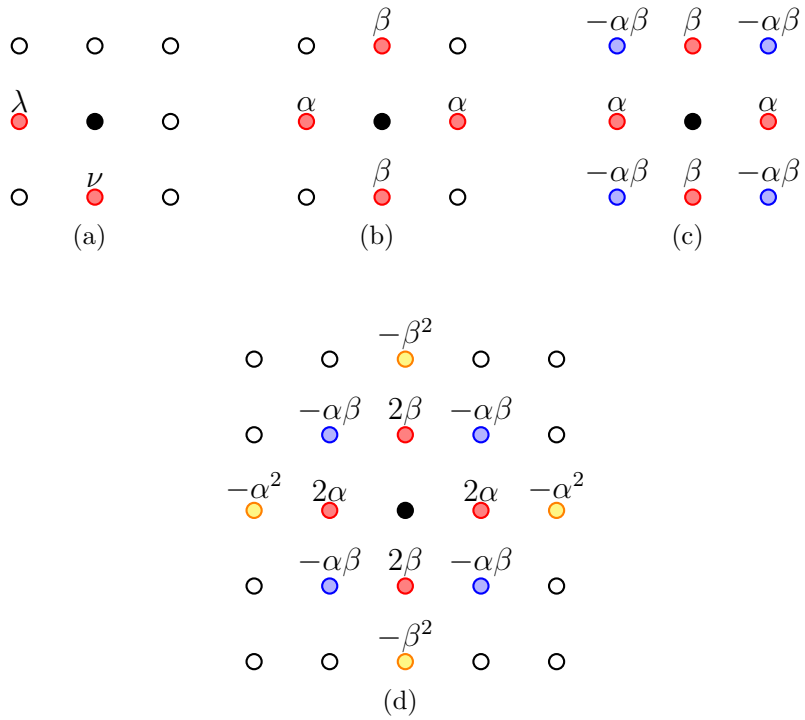


Fig. A.2 Spatial schemes on rectangular grids. The black dot is the location of interest, the grey points are the defined to be the neighbours of the black dots and the labels above are the corresponding spatial parameters.

The first example, also the simplest, is the linear-by-linear process (Martin, 1979). The process is a one-sided SAR model $\mathbf{Z}$ formulated by the product of two Markov processes (Figure A.2(a))

$$Z_{ij} = X_i Y_j, \quad X_i = \lambda X_{i-1} + \epsilon_i^{(x)}, \quad Y_j = \nu Y_{j-1} + \epsilon_j^{(y)} \tag{A.20}$$

where $\epsilon^{(x)}$ and $\epsilon^{(y)}$ are random errors. By comparing the entries of the precision matrices, the SAR process with such a neighbourhood scheme is seen to be equivalent to a second-order CAR process with its first-order parameters being α and β and second-order spatial parameter of $-\alpha\beta$, where $\alpha = \lambda/(1+\lambda^2)$ and $\beta = \nu/(1+\nu^2)$, as in Figure A.2(c).

The second example is the more commonly used first-order SAR model on a lattice where there are two different spatial parameters α and β in the vertical and horizontal directions as shown in Figure A.2(b). Write the precision matrices as $\boldsymbol{Q}_C = \boldsymbol{M}^{-1}(\mathbf{I} - \rho_C \mathbf{W})$ and $\boldsymbol{Q}_S = (\mathbf{I} - \rho_S \mathbf{W})\boldsymbol{\Lambda}^{-1}(\mathbf{I} - \rho_S \mathbf{W})$. If we assume $\boldsymbol{\Lambda} = \mathbf{I}$, then when $\boldsymbol{Q}_C = \boldsymbol{Q}_S$, M is a $N \times N$ diagonal matrix with all entries on the diagonal being $1/(1+2\alpha^2+2\beta^2)$. It actually represents a spatial matrix of a third-order CAR model as shown in Figure A.2(d), where the first-order spatial coefficients are the twice of the SAR model, the second-order spatial parameters are $-\alpha\beta$ and the third-order spatial parameters are $-\alpha^2$ and $-\beta^2$.

A.3.1 MPLE and LSE

It happens that the MPLE of the direct CAR model and the least squares estimators (LSE) of the direct CAR and SAR models are the same. For both CAR and SAR models, the least squares estimation requires to minimise

$$\boldsymbol{\epsilon}^2 = (\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})^T (\mathbf{I} - \rho \mathbf{W})^2 (\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})$$

which gives exactly the same result as the MPLE. It was shown by Whittle (1954) that the LSE is inconsistent for the SAR process on an $n \times n$ torus as $n \to \infty$. In our simple setting with covariance matrix $\boldsymbol{\Sigma} = \sigma^2(\mathbf{I} - \rho \mathbf{W})^{-1}$,

$$\hat{\rho} - \rho = \frac{\boldsymbol{e}^T \mathbf{W} \boldsymbol{e} - \rho \boldsymbol{e}^T \mathbf{W}^2 \boldsymbol{e}}{\boldsymbol{e}^T \mathbf{W}^2 \boldsymbol{e}} = \frac{[(\mathbf{I} - \rho \mathbf{W})^T (\mathbf{Y} - \mathbf{X}\hat{\boldsymbol{\beta}})]^T \mathbf{W} (\mathbf{Y} - \mathbf{X}\hat{\boldsymbol{\beta}})}{\boldsymbol{e}^T \mathbf{W}^2 \boldsymbol{e}}$$

In the CAR models, the numerator of right-hand side has mean zero and, with a suitable $\mathbf{W}$ the least squares estimator is shown to be unbiased and consistent, while for the SAR models, $(\mathbf{I} - \rho\mathbf{W})^T(\mathbf{Y} - \mathbf{X}\hat{\boldsymbol{\beta}})$ is always correlated to $\mathbf{Y} - \mathbf{X}\hat{\boldsymbol{\beta}}$, which makes it impossible for the estimator to be consistent.

Appendix B

Details for implementation

B.1 The gradient and Hessian

Table B.1 includes the gradient and Hessian function for a binomial logit model and a Poisson log linear model with latent CAR variables using Approximation II in (2.6) and Approximation III in (2.7).

Table B.1 Gradient and Hessian for Approximation II & III

	Approx	Binomial	Poisson
$\frac{\partial \log h(\boldsymbol{\theta})}{\partial \rho}$	II	$\frac{\mathbf{Z}^T\mathbf{W}\mathbf{Z}}{2\sigma^2} - \frac{1}{2}\sum_{i=1}^{N}\frac{\lambda_i}{1-\lambda_i\rho}$	
	III	$\frac{\mathbf{Z}^T\mathbf{W}\mathbf{Z}}{2\sigma^2}$	
$\frac{\partial \log h(\boldsymbol{\theta})}{\partial \sigma^2}$	II	$\frac{\mathbf{Z}^T(\mathbf{I}-\rho\mathbf{W})\mathbf{Z}}{2(\sigma^2)^2} - \frac{N}{2\sigma^2}$	
	III	$\frac{\mathbf{Z}^T(\mathbf{I}-\rho\mathbf{W})\mathbf{Z}}{2(\sigma^2)^2}$	
$\frac{\partial \log h(\boldsymbol{\theta})}{\partial \boldsymbol{\beta}}$	II, III	$\sum_{k=1}^{n}\left(\mathbf{Y}_k - \frac{n\exp(x_k^T\boldsymbol{\beta}+z_k)}{1+\exp(x_k^T\boldsymbol{\beta}+z_k)}\right)x_k^T$	$(\mathbf{Y}-\exp(\mathbf{X}\boldsymbol{\beta}+\mathbf{Z}))^T\mathbf{X}$
$\frac{\partial^2 \log h(\boldsymbol{\theta})}{\partial \rho^2}$	II	$-\frac{1}{2}\sum_{i=1}^{N}\frac{\lambda_i^2}{(1-\lambda_i\rho)^2}$	
	III	0	
$\frac{\partial^2 \log h(\boldsymbol{\theta})}{\partial (\sigma^2)^2}$	II	$-\frac{\mathbf{Z}^T(\mathbf{I}-\rho\mathbf{W})\mathbf{Z}}{(\sigma^2)^3} + \frac{N}{2(\sigma^2)^2}$	
	III	$-\frac{\mathbf{Z}^T(\mathbf{I}-\rho\mathbf{W})\mathbf{Z}}{(\sigma^2)^3}$	
$\frac{\partial^2 \log h(\boldsymbol{\theta})}{\partial \rho \partial \sigma^2}$	II, III	$-\frac{\mathbf{Z}^T\mathbf{W}\mathbf{Z}}{2(\sigma^2)^2}$	
$\frac{\partial^2 \log h(\boldsymbol{\theta})}{\partial \boldsymbol{\beta}^2}$	II, III	$n\sum_{k=1}^{N}\frac{\exp(x_k^T\boldsymbol{\beta}+z_k)}{(1+\exp(x_k^T\boldsymbol{\beta}+z_k))^2}x_k^T x_k$	$\mathbf{X}^T\exp(\mathbf{X}\boldsymbol{\beta}+\mathbf{Z})(\exp(\mathbf{X}\boldsymbol{\beta}+\mathbf{Z}))^T\mathbf{X}$

Appendix C

Manual for the package "mclcar"

Package 'mclcar'

Type Package

Title Estimating Conditional Auto-Regressive (CAR) Models using Monte Carlo Likelihood Methods

Version 0.1.6

Date 2016-09-15

Author Zhe Sha [aut, cre]

Maintainer Zhe Sha <zhesha1006@gmail.com>

Description The likelihood of direct CAR models and Binomial and Poisson GLM with latent CAR variables are approximated by the Monte Carlo likelihood. The Maximum Monte Carlo likelihood estimator is found either by an iterative procedure of directly maximising the Monte Carlo approximation or by a response surface design method.

License GPL (>=2)

Depends R (>= 2.10),

LazyData yes

Imports spam, rsm, fields, maxLik, nleqslv, spdep

NeedsCompilation no

Suggests knitr

VignetteBuilder knitr

R topics documented:

mclcar-package 2
CAR.simLM 4
loglik.dCAR 5
mcl.dCAR 7
mcl.glm 9
OptimMCL 12
plot.rsmMCL 14
postZ 15
rsmMCL 17
ScotCancer 20
scotplot 21
summary.OptimMCL 21
summary.rsmMCL 22

`mclcar-package` *Estimating Conditional Auto-Regressive (CAR) Models using Monte Carlo Likelihood Methods*

Description

The likelihood of direct CAR models and Binomial and Poisson GLM with latent CAR variables are approximated by the Monte Carlo likelihood. The Maximum Monte Carlo likelihood estimator is found either by an iterative procedure of directly maximising the Monte Carlo approximation or by a response surface design method.

Details

Package:	mclcar
Type:	Package
Version:	0.1-6
Date:	2016-09-15
License:	GPL (>= 2)
LazyData:	yes
Depends:	R (>= 2.10)

Index of help topics:

`CAR.simLM`	Simulate samples from a CAR model.
`OptimMCL`	Iterative procedure for maximising the Monte Carlo likelihood
`ScotCancer`	Scottish lip cancer dataset from Clayton and Kaldor (1987)
`loglik.dCAR`	Likelihood computing and parameter estimation for a direct CAR model.
`mcl.dCAR`	Monte Carlo likelihood calculation for direct CAR models.
`mcl.glm`	Monte Carlo likelihood calculation for glm with latent CAR variables.
`mclcar-package`	Estimating Conditional Auto-Regressive (CAR) Models using Monte Carlo Likelihood Methods
`plot.rsmMCL`	Plot the fitted response surfaces
`postZ`	Sampling the CAR latent variables given the Binomial or Poisson observations in glm models.
`rsmMCL`	Response surface method for maximising the Monte Carlo likelihood
`scotplot`	Scottish lip cancer dataset from Clayton and Kaldor (1987) with geo information.
`summary.OptimMCL`	Summary the output from the iterative procedure of maximising the Monte Carlo likelihood.
`summary.rsmMCL`	Summary the output from the response surface method of maximising the Monte Carlo likelihood

Author(s)

Zhe Sha Maintainer: Zhe Sha <zhesha1006@gmail.com>

References

- Box, G. E. P. and Draper, N. R. (2007) *Response Surfaces, Mixtures, and Ridge Analyses*, Wiley, New York
- Sha, Z. 2016 *Estimating conditional auto-regression models*, DPhil Thesis, Oxford.

See Also

rsm,maxLik

Examples

```
#### Example 1: iterative optimising the Monte Carlo likelihood
#########################################################################
system.time({
set.seed(33)
n.torus <- 8
nb <- 30
rho <- 0.2
sigma <- 1.5
beta <- c(1, 1)
pars.true <- c(rho, sigma, beta)
X0 <- cbind(rep(1, n.torus^2), sample(log(1:n.torus^2)/5))
mydata2 <- CAR.simGLM(method = "binom", n = c(n.torus, n.torus), pars = pars.true,
                      Xs = as.matrix(X0), n.trial = nb)

## use a glm to find initial values for the importance sampler
data.glm <- data.frame(y=mydata2$y, mydata2$covX[,-1])
fit.glm <- glm(cbind(y, nb-mydata2$y) ~ .,data = data.glm, family=binomial)
library(spdep)
logitp <- log((mydata2$y+0.5)/(mydata2$n.trial - mydata2$y + 0.5))
data.splm <- data.frame(y=logitp, mydata2$covX[,-1])
listW <- mat2listw(mydata2$W)
fit.splm <- spautolm(y~., data = data.splm, listw=listW, family = "CAR")
pars1 <- c(fit.splm$lambda, fit.splm$fit$s2, coef(fit.glm))

 ## Use the iterative procedure to find the MC-MLE
 iter.mcmle <- OptimMCL(data = mydata2, psi0 = pars1, family = "binom",
                        control = list(n.iter = 1, mc.var = TRUE),
                    mc.control = list(N.Zy = 1e3, Scale = 1.65/(n.torus^(2/6)), thin = 5,
                                      burns = 5e2, method = "mala", scale.fixed = TRUE))
summary(iter.mcmle, family = "binom", mc.covar=TRUE)

#### Example 2: RSM optimising the Monte Carlo likelihood
#########################################################################
mydata3 <- CAR.simGLM(method = "poisson", n = c(n.torus, n.torus),
                      pars = pars.true, Xs =as.matrix(X0))
## Fit by rsm
rsm.mcmle <- rsmMCL(data = mydata3, psi0 = c(0, 1, 2, 2), family = "poisson",
                    control = list(n.iter = 1, trace.all = TRUE),
                    mc.control = list(N.Zy = 1e3, Scale = 1.65/(n.torus^(2/6)),
                                      thin = 5, burns = 5e2,
                                      method = "mala", scale.fixed = TRUE))
```

```
summary(rsm.mcmle, family = "poisson",  mc.covar=TRUE)
})
```

`CAR.simLM` *Simulate samples from a CAR model.*

Description

This help page documents several functions for simulate samples from different CAR models.

Usage

```
CAR.simTorus(n1, n2, rho, prec)
CAR.simWmat(rho, prec, W)
CAR.simLM(pars, data)
CAR.simGLM(method = c("binom", "poisson"), W, n = NULL, pars, Xs = NULL,
n.trial = 1)
```

Arguments

`n1, n2, n`	n = c(n1, n2), the size of the torus
`rho`	the spatial coefficient in the CAR precision matrix (inverse of the covariance matrix)
`prec`	the precision in the CAR precision matrix
`W`	the spatial weight matrix
`pars`	parameter values of the direct CAR model
`data`	a `data` object same as described in `loglik.dCAR`
`method`	a character equal to either "binom" or "poisson" that indicate the distribution of the sample to be simulated.
`Xs`	the covariates to be used in the glm
`n.trial`	the number of trials in each unit of the binomial samples

Value

`CAR.simTorus`	returns a list containing the spatial weight matrix `W` and the simulated CAR samples `X`;
`CAR.simWmat,CAR.simLM`	returns an array of the simulated CAR samples;
`CAR.simGLM`	returns a list containing: **rho, sigma, beta,** the parameter values used to generate the samples **y,** the simulated data **covX,** the covariates **W,** the spatial weight matrix **Z.true,** the simulated CAR variables **eta,** the simulated glm linear response **Emean,** the simulated glm mean **n.trial,** the number of trials in each unit for binomial samples

Author(s)

Zhe Sha <sha@stats.ox.ac.uk>

See Also

mcl.glm, mcl.dCAR, mcl.prep.dCAR, mcl.prep.glm

Examples

```
## Simulate CAR data on a torus
set.seed(33)
n.torus <- 10
rho <- 0.2
sigma <- 1.5
prec <- 1/sigma
beta <- c(1, 1)
XX <- cbind(rep(1, n.torus^2), sample(log(1:n.torus^2)/5))
mydata1 <- CAR.simTorus(n1 = n.torus, n2 = n.torus, rho = rho, prec = prec)

## Simulate CAR data for a given spatial weight matrix
Wmat <- mydata1$W
mydata2 <- CAR.simWmat(rho = rho, prec = prec, W = Wmat)

## Simualte data from a linear model with CAR error
y <- XX %*% beta + mydata1$X
mydata1$data.vec <- data.frame(y=y, XX[,-1])
mydata3 <- CAR.simLM(pars = c(0.1, 1, 2, 0.5), data = mydata1)

## Simulate Binomial data with CAR latent variables
mydata4 <- CAR.simGLM(method="binom", n=c(10,10), pars = c(rho, sigma,
                      beta), Xs=XX, n.trial = 5)
```

`loglik.dCAR` *Likelihood computing and parameter estimation for a direct CAR model.*

Description

This help page documents a few functions for exact evaluation of the log-likelihood and profile log-likelihood for a direct CAR model, the maximum pseudo-likelihood estimator, and least square estimators for beta and sigma given the spatial coefficient rho in the CAR covariance matrix. The exact evaluation computes the log determinant in the log-likelihood function with eigen-values of the spatial weight matrix.

Usage

```
loglik.dCAR(pars, data)
ploglik.dCAR(rho, data)

get.beta.lm(rho, data)
sigmabeta(rho, data)

mple.dCAR(data, tol = 1e-06, rho0=0)
```

Arguments

pars	the parameter value to be evaluated
rho	the value of the spatial coefficient in the CAR covariance to be evaluated
data	a list containing the following objects: W the spatial weight matrix, **lambda** the eigenvalues of W (if given), **data.vec** a data frame of the response variable and covariates.
tol	tolerance for the relative difference for two consequtive iterations in finding the MPLE; default set to be = 1e-06
rho0	starting value for iteratively finding the MPLE; default value set to be 0.

Details

The eigen-values can be supplied in the `data` list if the likelihood for the same data is going to be evaluated many times; if not supplied the function use the function `eigen` to find the eigen-values of the weight matrix.

Value

`loglik.dCAR` and `ploglik.dCAR` return a numeric value of the log-likelihood evaluation. `get.beta.lm`, `sigmabeta` and `mple` return a numeric array of the estimates.

Author(s)

Zhe Sha <sha@stats.ox.ac.uk>

See Also

`CAR.simLM`, `get.beta.glm`, `mcl.dCAR`

Examples

```
## Simulate data from a torus
set.seed(30)
n.torus <- 20
rho <- 0.15
sigma <- 1.5
beta <- c(1, 2)
XX <- cbind(rep(1, n.torus^2), log(1:n.torus^2))
mydata <- CAR.simTorus(n.torus, n.torus, rho, 1/sigma)
y <- XX %*% beta + mydata$X
mydata$data.vec <- data.frame(y=y, XX[,-1])

## evaluate the log-likelihood without lambda
loglik.dCAR(pars = c(0.1, 1, 0.9, 2.1), data = mydata)

## evaluate the log-likelihood with lamda
lambda <- eigen(mydata$W, symmetric = TRUE, only.values=TRUE)$values
mydata$lambda <- lambda
loglik.dCAR(pars = c(0.1, 1, 0.9, 2.1), data = mydata)

## evaluate the profile log-likelihood of rho
ploglik.dCAR(rho = 0.1, data = mydata)
```

```
## given rho = 0.1, find the least square estimates for beta and sigma
get.beta.lm(rho = 0.1, data = mydata)
sigmabeta(rho = 0.1, data = mydata)

## find the maximum pseudo-likelihood estimates
mple.dCAR(data = mydata)
```

`mcl.dCAR` *Monte Carlo likelihood calculation for direct CAR models.*

Description

This help page contains functions for calculating the Monte Carlo likelihoods and variances for direct CAR models.

`mcl.prep.dCAR` generates the Monte Carlo samples from the importance sampling distribution to be used in the Monte Carlo likelihood functions.

`mcl.dCAR` calculates the Monte Carlo log-likelihood ratio for a given parameter value to the importance sampler value.

`mcl.profile.dCAR` calculates the Monte Carlo profile log-likelihood ratio of rho.

`vmle.dCAR` calculates the variance at the Monte Carlo MLE.

`Avar.lik.dCAR` calculates the asymptotic variance of the Monte Carlo likelihood for a direct CAR model on a given size of torus and given size of Monte Carlo samples.

Usage

```
mcl.prep.dCAR(psi, n.samples, data)

mcl.dCAR(pars, data, simdata, Evar = FALSE)
mcl.profile.dCAR(rho, data, simdata, Evar = FALSE)

vmle.dCAR(MLE, data, simdata)
Avar.lik.dCAR(pars, psi, data, n.samples, Log = TRUE)
```

Arguments

`psi`	the value of the importance sampler parameters
`n.samples`	the number of Monte Carlo samples
`pars`	the parameter value for the Monte Carlo likelihood function to be evaluated at
`rho`	the value of rho for the Monte Carlo profile likelihood function to be evaluated at
`data`	a `data` object same as described in `loglik.dCAR`
`simdata`	a list object in the same format as the output of `mcl.prep.dCAR`
`Evar`	if TRUE return the estimated variance of the Monte Carlo likelihood
`MLE`	a list object with `par` the estimated Monte Carlo MLE `hessian` the hessian at the Monte Carlo MLE
`Log`	if log = TRUE, then the variance is of the log-likelihood ratio; otherwise it is of the likelihood ratio

Details

The asymptotic and estimated variance is derived for a direct CAR model on a $n \times n$ torus with 'rook' style binary neighbourhood matrix. Details of the derivation is given in Sha (2016) ch. 3.

Value

`mcl.prep.dCAR` returns a list object containing the following following elements:

- `simYa` matrix of the Monte Carlo samples
- `t10` and `t20`statistics pre-calculate for the importance distribution

`mcl.dCAR` and `mcl.proifle.dCAR` return a numeric value of the Monte Carlo likelihood when `Evar = FALSE`; if `TRUE` it returns an array of the Monte Carlo likelihood and the corresponding estimated variance.

`vmle.dCAR` returns the estimated covariance matrix of the Monte Carlo MLE.

`Avar.lik.dCAR` returns a numeric value of the asymptotic variance.

Author(s)

Zhe Sha <sha@stats.ox.ac.uk>

References

Sha, Z. 2016 *Estimating conditional auto-regression models*, DPhil Thesis, Oxford.

See Also

`mcl.glm`, `OptimMCL`, `rsmMCL`

Examples

```
## Simulate some data to work with
set.seed(30)
n.torus <- 10
rho <- 0.15
sigma <- 1.5
beta <- c(1, 2)
XX <- cbind(rep(1, n.torus^2), log(1:n.torus^2))
mydata1 <- CAR.simTorus(n.torus, n.torus, rho, 1/sigma)
y <- XX %*% beta + mydata1$X
mydata1$data.vec <- data.frame(y=y, XX[,-1])

## Choose the importance sampler value to the the MPLE
psi1 <- mple.dCAR(mydata1)

## Prepare the Monte Carlo samples
mcdata1 <- mcl.prep.dCAR(psi = psi1, n.samples = 100, data = mydata1)

## Calculate the Monte Carlo likelihoods
mcl.dCAR(c(rho, sigma, beta), data = mydata1, simdata = mcdata1, Evar =
TRUE)
mcl.profile.dCAR(rho, data = mydata1, simdata = mcdata1, Evar = TRUE)

## Do a direct optimization of the Monte Carlo likelihood function
## to find the MLE but it takes very long time to run and may not converge
```

```
## Not run:
opt <- optim(psi1, fn = mcl.dCAR,
             data = mydata1, simdata = mcdata1,
             hessian = TRUE, control = list(fnscale = -0.5))

## End(Not run)

## Assume the we have obtained the opt with the following values
opt<- list(par = c (0.08013547, 1.70294099, 0.01571957, 2.23203089),
           hessian =  matrix(c(  -190.8791352,  -1.5682773,    0.1863733,   -4.844151,
 -1.5682773,  -16.1605186,    0.4911009,    4.403844,
 0.1863733,   0.4911009,  -41.1496774, -156.686631,
-4.8441507,   4.4038442, -156.6866312, -634.316017), 4, 4))

## Calculate the variance of the MC-MLE
vmle.dCAR(opt, data = mydata1, simdata = mcdata1)

## Calculate the asymptotic variance of the likelihood at MC-MLE
Avar.lik.dCAR(pars = opt$par, psi = psi1, data = mydata1, n.samples =
100)
```

`mcl.glm` *Monte Carlo likelihood calculation for glm with latent CAR variables.*

Description

This help page contains functions for calculating the Monte Carlo likelihoods and variances for generalised linear models (Binomial and Poisson) with latent CAR variables.

`mcl.prep.glm` simulates the Monte Carlo samples from the importance sampling distribution to be used in the Monte Carlo likelihood functions. The samples are from an MCMC Chain generated by the Metropolis-Hastings Algorithms. Details see `postZ`.

`mcl.glm` calculates the Monte Carlo log-likelihood ratio for a given parameter value to the importance sampler value.

`mcl.profile.glm` calculates the Monte Carlo profile log-likelihood ratio of rho and sigma.

`vmle.glm` calculates the variance at the Monte Carlo MLE.

`get.beta.glm` finds the linear coefficients of the glm with given CAR parameters by solving the Monte Carlo gradient with `nleqslv`.

Usage

```
mcl.prep.glm(data, family, psi, Z.start = NULL, mcmc.control=list(),
             pilot.run = TRUE, pilot.plot = FALSE, plot.diag = FALSE)

mcl.glm(pars, mcdata, family, Evar = FALSE)
mcl.profile.glm(pars, beta0, mcdata, family, Evar = FALSE)

vmle.glm(MLE, mcdata, family)

get.beta.glm(beta0, rho.sig, mcdata, family)
```

Arguments

data	a list object given by CAR.simGLM
family	a character equal to either "binom" for Binomial variables or "poisson" for Poisson variables
psi	the value of the importance sampler parameters
Z.start	initial value for simulating the MCMC algorithm
mcmc.control	a list of parameter that control the MCMC algorithm, see more in the Details of postZ.
pilot.run	if TRUE, a pilot run is done for tuning the MCMC parameters
pilot.plot	if TRUE, the MCMC diagnostic plots are plotted for the pilot run
plot.diag	if TRUE, the MCMC diagnostic plots are plotted for the simulated chain
pars	the parameter value of the likelihood function to be evaluated at
mcdata	a list of data generated by mcl.prep.glm
Evar	if TRUE return the estimated variance of the Monte Carlo likelihood
MLE	a list object with
	par the estimated Monte Carlo MLE
	hessian the hessian at the Monte Carlo MLE
beta0	starting point for finding the beta
rho.sig	values of rho and sigma for the CAR covariance matrix

Details

The Monte Carlo samples can be simulated by different MCMC algorithms described in postZ. The log determinant is calculated directly in the evaluation of the Monte Carlo likelihood and the Monte Carlo approximation is only used to approximate the integral for the latent variables.

Given the CAR covariance matrix, the betas are found by solving the non-linear system given by letting the gradient of the Monte Carlo likelihood with respect to beta equal to zero. Due to the Monte Carlo error, the nleqslv use only a maximum of two iteration to find a local solution.

Value

mcl.prep.glm returns a list object that contains all the output of CAR.simGLM plus the following following elements:

- Zy, a matrix of the Monte Carlo samples,
- lHZy.psi, a list of statistics pre-calculate for the importance distribution,
- mcmc.pars, a list of the mcmc tuning parameter used in the simulation.

mcl.glm and mcl.proifle.glm return a numeric value of the Monte Carlo likelihood when Evar = FALSE; if TRUE it returns an array of the Monte Carlo likelihood and the corresponding estimated variance.

vmle.glm returns the estimated covariance matrix of the Monte Carlo MLE.

get.beta.glm returns an object given by nleqslv.

Author(s)

Zhe Sha <sha@stats.ox.ac.uk>

See Also

postZ, mcl.dCAR, OptimMCL, rsmMCL

Examples

```
## Simulate some data to work with
set.seed(33)
n.torus <- 10
nb <- 30
rho <- 0.2
sigma <- 1.5
beta <- c(1, 1)
pars.true <- c(rho, sigma, beta)
X0 <- cbind(rep(1, n.torus^2), sample(log(1:n.torus^2)/5))

mydata2 <- CAR.simGLM(method = "binom", n = c(n.torus, n.torus), pars = pars.true,
                      Xs = as.matrix(X0), n.trial = nb)

## Prepare the Monte Carlo samples
## Find a suitable initial value for the importance sampler parameter, e.g:
library(spdep)
data.glm <- data.frame(y=mydata2$y, mydata2$covX[,-1])
fit.glm <- glm(cbind(y, nb-mydata2$y) ~ .,data = data.glm, family=binomial)
logitp <- log((mydata2$y+0.5)/(mydata2$n.trial - mydata2$y + 0.5))
data.splm <- data.frame(y=logitp, mydata2$covX[,-1])
listW <- mat2listw(mydata2$W)
fit.splm <- spautolm(y~., data = data.splm, listw=listW, family = "CAR")
psi.binom <- c(fit.splm$lambda, fit.splm$fit$s2, coef(fit.glm))

## Set the parameters for the MCMC algorithm
mc.pars <- list(N.YZ =1e3, N.Zy = 1e3, Scale = 1.65/(n.torus^(2/6)), thin = 5,
                  burns = 5e2, method = "mala", scale.fixed = TRUE)

mc.data2 <- mcl.prep.glm(data = mydata2, family = "binom",
                         psi = psi.binom, mcmc.control = mc.pars,
                         pilot.plot = TRUE, plot.diag = TRUE)

## Calculate the Monte Carlo likelihoods
pars.t <- c(rho, sigma, beta)

mcl.glm(pars.t, family = "binom",  mcdata = mc.data2, Evar = TRUE)

## Do a direct optimization of the Monte Carlo likelihood function to find the MLE
## Not run:
library(maxLik)
A <- matrix(0, nrow = 3, ncol = length(pars.t))
A[,1] <- c(1, -1, 0)
A[,2] <- c(0, 0, 1)
B <- c(0.24, 0.24, -0.1)

mle.bin <- maxBFGS(fn = mcl.glm, start=as.numeric(psi.binom), print.level=1,
                  constraints=list(ineqA=A, ineqB=B),
                  mcdata = mc.data2, family = "binom")

## End(Not run)
## Assume we have obtained mle.bin
```

```
mle.bin <- list(estimate = c(0.2271854 , 1.6117040, -0.3870856,  2.6525503),
                hessian = matrix(c( -1981.34398, -71.992190,  -79.910745,  -63.451022,
                                    -71.99219, -13.985701,    2.965628,    2.216893,
                                    -79.91074,   2.965628, -251.320742, -176.993087,
                                   -63.45102,   2.216893, -176.993087, -132.575284), 4,4))

## Calculate the variance of the MC-MLE
v.mle.bin <- vmle.glm(mle.bin, mcdata = mc.data2, family = "binom")


## Find the Monte Carlo MLE of beta, given the value of rho and sigma
get.beta.glm(beta0 = psi.binom[-c(1,2)], rho.sig = psi.binom[1:2],
             mcdata = mc.data2, family = c("binom"))
```

`OptimMCL` *Iterative procedure for maximising the Monte Carlo likelihood*

Description

The function uses an iterative procedure of directly maximising the Monte Carlo likelihood and the updating step size is limited by defining an experimental region using the estimated Monte Carlo variance.

Usage

```
OptimMCL(data, psi0, family, control = list(), mc.control = list())
```

Arguments

`data`	the data set to be estimated; for direct CAR model it is a list objects same used in `loglik.dCAR` and for glm with CAR latent variables, it is the same as the output of `CAR.simGLM`.
`psi0`	the initial value for the importance sampler parameter
`family`	a character take values in "gauss" (for direct CAR models), "binom" and "poisson" (glm Binomial/Poisson with CAR latent variable).
`control`	a list that controls the iterative procedures, see more in Details.
`mc.control`	a list that controls the MCMC algorithm, see more in Details of `postZ`

Details

The iterative procedure starts by using samples from the importance sampler with the parameter values given by `psi0` and directly optimise the Monte Carlo likelihood. Then the importance sampler parameter value is updated by using either the MLE found by current step if it is within the experimental region (to be defined in the following), or a weighted average between the current psi value and the current MLE value, with weight to be specified by `psi.ab` in the list of `control` argument.

The experimental region is defined to be the region where the estimated variance of the Monte Carlo likelihood is smaller than $a/\sqrt{(n.s)}$ where $n.s$ is the number of Monte Carlo samples and a and is some fixed constant that can be specified through `psi.ab` in `control`.

The procedure iterate until convergence which is defined to be that the Monte Carlo likelihood is smaller than min(1, 2*sqrt(mcl.var)) where mcl.var is the variance of the estimated variance of the

Monte Carlo likelihood evaluated at the MC-MLE. At this point, the procedure use a increased number of Monte Carlo samples and do the same iteration until converges again. The MC-MLE found on this second convergence is given as the final result. The Monte Carlo sample size is increased by a multiple of user specified number through the `control` argument.

The `control` argument is a list containing the following elements:

n.iter, maximum number of iterations; default value is 10 for direct CAR models and 20 for glm with latent CAR variables

mc.samples, (for direct CAR model only) an array of two with the first element being the number of Monte Carlo samples and the second being the increasing multiple of the sample size after the first convergence; default is c(500, 2)

s.increase, (for glm with latent CAR variables only) an numeric value for the increasing multiple of the sample size after the first convergence; default value is 2.

rho.range, the valid range for the parameter rho in the direct CAR model; default value is c(-0.249, 0.249)

psi.range, the valid range for the parameters rho and sigma in the glm with latent CAR variables; default is c(-0.249, 0.249, 0.1, 10)

psi.ab, an array with two entries controlling the experimental region and updating rules; the first entry controls the experimental region as a defined above and the second entry controls the updating rule as defined above; default is c(1, 0.5)

mc.var, if true, the estimated covariance matrix for the MC-MLE is returned in every iteration; default is FALSE

trace.all, if true, each iteration is stored and returned in the final output, default is TRUE

verbose, if true, an summary message is printed after each iteration, default is TRUE

Value

When trace.all is TRUE, the function returns a list containing the following objects:

Psi, a list of importance sampler parameter values from all iterations;

MC.datas, a list object generated by `mcl.prep.dCAR` or `mcl.prep.glm` from all iterations;

MC.MLEs, a list of the MC-MLE found by all iterations;

MC.Hess, a list of the Hessian matrix at the MC-MLE from all iterations;

MC.Vars, a list of the estimated covariance matrix of the MC-MLE from all iterations;

data, the data object supplied to the function;

N.iter, the total number of iterations;

total.time, the total time elapsed;

convergence, a logical value indicating whether the procedure converged or not;

mcsamples, an array of two entries for the initial Monte Carlo sample size and the increased Monte Carlo sample size after the first convergence.

When trace.all is FALSE, the function returns the same list with each object in the list containing the result of the final iteration only.

Author(s)

Zhe Sha <sha@stats.ox.ac.uk>

References

Sha, Z. 2016 *Estimating conditional auto-regression models*, DPhil Thesis, Oxford.

See Also

summary.OptimMCL, rsmMCL

Examples

```
## Take long time to run
## Simulate some data to work with
set.seed(33)
n.torus <- 10
nb <- 30
rho <- 0.2
sigma <- 1.5
beta <- c(1, 1)
pars.true <- c(rho, sigma, beta)
X0 <- cbind(rep(1, n.torus^2), sample(log(1:n.torus^2)/5))
mydata2 <- CAR.simGLM(method = "binom", n = c(n.torus, n.torus), pars = pars.true,
                      Xs = as.matrix(X0), n.trial = nb)

## use a glm to find initial values for the importance sampler
data.glm <- data.frame(y=mydata2$y, mydata2$covX[,-1])
fit.glm <- glm(cbind(y, nb-mydata2$y) ~ .,data = data.glm, family=binomial)
library(spdep)
logitp <- log((mydata2$y+0.5)/(mydata2$n.trial - mydata2$y + 0.5))
data.splm <- data.frame(y=logitp, mydata2$covX[,-1])
listW <- mat2listw(mydata2$W)
fit.splm <- spautolm(y~., data = data.splm, listw=listW, family = "CAR")
pars1 <- c(fit.splm$lambda, fit.splm$fit$s2, coef(fit.glm))

## Use the iterative procedure to find the MC-MLE
## !!!NOTE: the example below is only an illustration of usage
## users should increase the number of iterations and MCMC samples
## to get convergence results.
iter.mcmle <- OptimMCL(data = mydata2, psi0 = pars1, family = "binom",
                       control = list(n.iter = 1, mc.var = TRUE),
                     mc.control = list(N.Zy = 1e3, Scale = 1.65/(n.torus^(2/6)), thin = 5,
                    burns = 5e2, method = "mala", scale.fixed = TRUE))
summary(iter.mcmle, family = "binom", mc.covar=TRUE)
```

---

plot.rsmMCL *Plot the fitted response surfaces*

Description

This function plot the fitted response surfaces by using the result from rsmMCL.

Usage

```
## S3 method for class 'rsmMCL'
plot(x, family, trace.all = TRUE, plain=TRUE, ...)
```

Arguments

x	an rsmMCL object retruned by rsmMCL.
family	a character takes value in "guass", "binom" and "poisson".
trace.all	an logic value tells whether the input object given by rsmMCL contains results from all iterations or not.
plain	an logic value indicates whether to show plots in separate devices with a smart display. Use the defalut value TRUE if the function is used in generating documents by using **knitr**.
...	For extra arguments. The Lattice parameter layout can be used to control the layout of the graphs.

Author(s)

Zhe Sha <sha@stats.ox.ac.uk>

See Also

rsmMCL, summary.rsmMCL

Examples

```
## See examples for rsmMCL
```

---

postZ *Sampling the CAR latent variables given the Binomial or Poisson observations in glm models.*

Description

The function uses several different MCMC algorithms to sample the CAR latent variables Z from the distribution P(Zly) given the Binomial or Poisson observations in glm models using the importance sampler parameter values psi.

Usage

```
postZ(data, family, psi, Z.start, mcmc.control = list(), plots = FALSE)
```

Arguments

data	the data set to be estimated and the object has same as the output of CAR.simGLM
family	a character take values in "binom" and "poisson" for Binomial or Poisson glm with CAR latent variable).
psi	Parameters for the importance distribution.
Z.start	Initial value of the CAR variables for the Markov Chain.
mcmc.control	a list that controls the MCMC algorithm, see more in Details.
plots	if TRUE, the diagnostic plots are shown on running the function.

Details

The function uses two MCMC algorithms, the Metropolis Hastings algorithm with random walk proposal (rwmh) and the Metropolis adjusted Langevin algorithm (mala), in sampling the CAR latent variables. The scale parameter for the proposal distribution can be tuned to achieve the optimal acceptance rate by using an adaptive algorithm in a pilot run. The argument `mcmc.control` contains the following objects

- N.Zy, the number of iterations of the Markov Chain
- Scale, the scale for the proposal distribution
- thin, the thinning step
- burns, the number of burn-in samples to be discarded
- method, the name of the MCMC algorithm, either "rwmh" or "mala"
- scale.fixed, if true, the scale parameter is fixed; otherwise an adaptive MCMC algorithm is run for tuning the scale parameter to achieve the optimal acceptance rate by using an stochastic approximation algorithm
- c.ad, an array contains the two values for the tuning parameter for the stochastic approximation algorithm in the adaptive MCMC algorithm; default is set to be c(1, 0.7)

Value

The function return a list of the following objects:

- Z a matrix of samples from from P(Z|y)
- AC.ran array of the acceptance rate updated at each iteration
- Scalesif in the list of mcmc.control scale.fixed = FALSE, an array of the adaptive scales for the MCMC proposal

Author(s)

Zhe Sha <sha@stats.ox.ac.uk>

See Also

`mcl.prep.glm`, `OptimMCL`, `rsmMCL`

Examples

```
set.seed(33)
n.torus <- 10
nb <- 30
rho <- 0.2
sigma <- 1.5
beta <- c(1, 1)
pars.true <- c(rho, sigma, beta)
X0 <- cbind(rep(1, n.torus^2), sample(log(1:n.torus^2)/5))
mydata2 <- CAR.simGLM(method = "binom", n = c(n.torus, n.torus), pars = pars.true,
                      Xs = as.matrix(X0), n.trial = nb)

## use a glm to find initial values for the importance sampler
library(spdep)
data.glm <- data.frame(y=mydata2$y, mydata2$covX[,-1])
fit.glm <- glm(cbind(y, nb-mydata2$y) ~ .,data = data.glm, family=binomial)
```

```
## estimate sigma and rho, transform the binomial to Gaussian by logit
logitp <- log((mydata2$y+0.5)/(mydata2$n.trial - mydata2$y + 0.5))
data.splm <- data.frame(y=logitp, mydata2$covX[,-1])
listW <- mat2listw(mydata2$W)
fit.splm <- spautolm(y~., data = data.splm, listw=listW, family = "CAR")
pars1 <- c(fit.splm$lambda, fit.splm$fit$s2, coef(fit.glm))

## Sample form importance distribution with psi = pars1
mc.control <- list(N.Zy = 1e3, Scale = 1.65/(n.torus^(2/6)), thin = 5,
                   burns = 5e2, method = "mala", scale.fixed = TRUE)
## Binomial
Z.S0 <- CAR.simWmat(pars1[1], 1/pars1[2], mydata2$W)
simZy <- postZ(data = mydata2, Z.start = Z.S0, psi = pars1,
               family = "binom", mcmc.control = mc.control, plots =
TRUE)
```

`rsmMCL` *Response surface method for maximising the Monte Carlo likelihood*

Description

The function maximise the Monte Carlo likelihood by using the response surface methodology.

Usage

```
rsmMCL(data, psi0, family, exact0 = NULL, control = list(), mc.control = list())
```

Arguments

`data, psi0, family, mc.control`	are the same as those in `OptimMCL`
`control`	see more in Details
`exact0`	the exact log likelihood ratio values of a given range or parameters to the importance sampler parameter value. This is used for comparing the fitted response surface of the Monte Carlo likelihood to the exact if the exact evaluation is required.

Details

The response surface methodology finds the maximum of a function by using first and second order polynomial approximations to the target function. The target function is usually unknown or expensive to evaluate. The function is approximated by a first order model in regions that are far away from the maximum and by a second order model around the maximum region. Then the stationary point of the second order model is used as an approximation to the maximum point. By changing the design region, the number and location of the evaluation points, the approximated surface can achieve different accuracy and variance properties. Details of the methodology can be found in Box, G.E.P. and Draper, N.R.(2007) and details of the implementation of the method in this package can be found Sha (2016).

The procedure implemented in this function locates the maximum region of the target function by iteratively updating the design centre and design region. Given a starting point as the design centre, the design region is given based on the finite variance region of the importance sampling

approximation as described in Sha (2016), chap. 4. Then with the design region, design points are generated by the central composite design (Box, G.E.P. and Draper, N.R. (2007)). The Monte Carlo likelihood and the corresponding estimated variance at these design points are evaluated and used to fit the first or second order polynomials.

If the stationary point is not found or found to be outside the current design region then, either a steepest ascent analysis (for a first order approximation) or a ridge analysis (for a second order approximation) is done to find the next design centre. The updating step is also limited by considering the estimated variance of the Monte Carlo likelihood.

The procedure is defined to be converged once the stationary point is found and within the design region or the difference between the design centres in two consecutive iterations is smaller than a given tolerance. The Monte Carlo sample size is then increased by a multiple of m once the procedure converges. With the new Monte Carlo sample size a few more iterations are done until the procedure achieve convergence again. The result of the second procedure is given as the final output.

The `control` argument is a list containing the following elements:

mc.samples, an array of numbers specifying the Monte Carlo sample size(for direct CAR models only), the increasing multiple and the maximum Monte Carlo sample size to use.

n.iter, maximum number of iterations; default value is 20

time.max, maximum total time for the computation

K, an array of two numbers for generating the design region; the first number K[1] decide the initial size of the design region and once the system find stationary point within the design region, K[1] becomes K[1]/K[2] to get a smaller design region for better approximation

mc.var, if true, the estimated covariance matrix for the MC-MLE is returned in every iteration; default is FALSE

psi.lim, a list that contains the range for rho and sigma in the CAR covariance matrix; default values are r.lim = c(-0.2499, 0.2499) for rho and s.lim = c(0.1, 8) for sigma

exacts (for direct CAR only) a list containing the setting for plotting the exact values for comparison with the fitted rsm surface; the default values are eval = TRUE (for plotting the exact values, default for glm is FALSE), rho = c(-0.25, 0.25), sigma = c(0.5, 2) define the plotting region, length = 100 define number of grid point in each coordinate in producing the plotting grid

rsm.fit a list of numbers that controls the fitting of the response surfaces and the elements are

- n01, number of central points in a central design in generating the design points. The central points are used to estimate the variance of the evaluation and the default is 4.
- n02, number of central points in a composite design and the default is 2.
- Rsq, R-squared in the fitted first order model in judging the goodness of fit of the first order model; the default value is 0.9.
- lof, p-value of the lack of fit test for the first order model; the default value is 0.05.
- st.diff, control the range of the approximated function; the approximated surface has less variance for values closes to zero, so this should not be too large and the default value is set to be 50.
- mcl.diff, maximum value of the approximated function evaluated at the MC-MLE; again this value should be close to zero and the default is 10.

plotrsm, if true the approximated response surface with steepest or ridge analysis is plotted; defualt is TRUE

trace.all, if true, each iteration is stored and returned in the final output, default is TRUE

verbose, if true, an summary message is printed after each iteration, default is TRUE

Value

When trace.all is TRUE, the function returns a list containing the following objects:

psi0, a vector of the parameter values used as the initial importance sampler parameter values

Psi, a list of importance sampler parameter values from all iterations

FVbox, a list containing the finite variance regions for all iterations

DAs, a list containing the design region for all iterations

DAbox, a list containing the design region box for all iterations that can be used for plotting

Ttime, a list containing the computational time used in each iteration

Expts, a list of design points in each iteration

Expt.Val1, a list of all values evaluated at the central design points in each iteration

Expt.Val2, a list of all values evaluated at the composite design points in each iteration

RSM.1, a list of all the fitted first-order response model

RSM.2, a list of all the fitted second-order response model

ESAs, a list of all explorations along the steepest ascent path

Sim.data, a list object generated by `mcl.prep.dCAR` or `mcl.prep.glm` from all iterations;

Exacts, a list of exact values in each iteration for the direct CAR models

data, the data object supplied to the function;

N.iter, the total number of iterations;

total.time, the total time elapsed;

convergence, a logical value indicating whether the procedure converged or not;

mcsamples, an array of two entries for the initial Monte Carlo sample size and the increased Monte Carlo sample size after the first convergence.

When trace.all is FALSE, the function returns the same list with each object in the list containing the result of the final iteration only.

Author(s)

Zhe Sha <sha@stats.ox.ac.uk>

References

Box, G. E. P. and Draper, N. R. (2007) *Response Surfaces, Mixtures, and Ridge Analyses*, Wiley, New York

Sha, Z. 2016 *Estimating conditional auto-regression models*, DPhil Thesis, Oxford.

See Also

`summary.rsmMCL`, `plot.rsmMCL`, `OptimMCL`

Examples

```
#### Poisson glm with CAR latent variables
## Simulate some data
set.seed(33)
n.torus <- 10
nb <- 30
rho <- 0.2
sigma <- 1.5
beta <- c(1, 1)
pars.true <- c(rho, sigma, beta)
X0 <- cbind(rep(1, n.torus^2), sample(log(1:n.torus^2)/5))
mydata3 <- CAR.simGLM(method = "poisson", n = c(n.torus, n.torus),
                      pars = pars.true, Xs =as.matrix(X0))

## Fit by rsm
rsm.mcmle2 <- rsmMCL(data = mydata3, psi0 = c(0, 1, 2, 2), family = "poisson",
                     control = list(n.iter = 2, trace.all = TRUE),
                     mc.control = list(N.Zy = 1e3, Scale = 1.65/(n.torus^(2/6)),
                                       thin = 5, burns = 5e2,
                                       method = "mala", scale.fixed = TRUE))
summary(rsm.mcmle2, family = "poisson",  mc.covar=TRUE)
plot(rsm.mcmle2, family = "poisson")
```

`ScotCancer` *Scottish lip cancer dataset from Clayton and Kaldor (1987)*

Description

The Scottish lip cancer dataset.

Format

The data object for the Monte Carlo likelihood estiation and it follows the same structure as described in `CAR.simGLM`. The covX is the model matrix with intercept and the fixed effect Paff

Source

Clayton D, Kaldor J 1987. Empirical Bayes Estimation of Age-standardized Relative Risk for use in Disease Mapping. *Biometrics* 43, 671–681

Ronnegard L, Shen X, and Alam M 2010 hglm: A Package for Fitting Hierarchical Generalized Linear Models. The R Journal, 2(2): 20-28. URL http://journal.r-project.org/archive/2010-2/RJournal_2010-2_Roennegaard~et~al.pdf.

References

Clayton D, Kaldor J 1987. Empirical Bayes Estimation of Age-standardized Relative Risk for use in Disease Mapping. *Biometrics* 43, 671–681

scotplot — *Scottish lip cancer dataset from Clayton and Kaldor (1987) with geo information.*

Description

The Scottish lip cancer dataset with geo information for plotting.

Format

A SpatialPolygonDataFrame contains the geo information and data for plotting

Source

Clayton D, Kaldor J 1987. Empirical Bayes Estimation of Age-standardized Relative Risk for use in Disease Mapping. *Biometrics* 43, 671–681

Lee D 2016. CARBayes: Spatial Generalised Linear Mixed Models for Areal Unit Data. R package version 4.6. https://CRAN.R-project.org/package=CARBayes

References

Clayton D, Kaldor J 1987. Empirical Bayes Estimation of Age-standardized Relative Risk for use in Disease Mapping. *Biometrics* 43, 671–681

summary.OptimMCL — *Summary the output from the iterative procedure of maximising the Monte Carlo likelihood.*

Description

This function summarizes the output of the output from the function OptimMCL.

Usage

```
## S3 method for class 'OptimMCL'
summary(object, family, trace.all = TRUE, mc.covar =
  TRUE, ...)
```

Arguments

object	an OptimMCL object returned by OptimMCL.
family	a character takes value in "guass", "binom" and "poisson".
trace.all	an logic value tells whether the input object given by OptimMCL contains results from all iterations of not
mc.covar	if TRUE, the estimated covariance matrix of the MC-MLE is returned
...	arguments passed to or from other methods.

Value

A list containing the following objects:

MC.mle, the final MC-MLE

N.iter, the total number of iterations

total.time, the total time elapsed

convergence, if TRUE the procedure converges

hessian, the Hessian at the MC-MLE if given; the default is NULL

mc.covar the estimated covariance matrix of the MC-MLE if given; the default is NULL

mc.samples the Monte Carlo samples size used in the initial stage and after the first convergence.

Author(s)

Zhe Sha <sha@stats.ox.ac.uk>

See Also

OptimMCL

Examples

```
## See examples for OptimMCL
```

`summary.rsmMCL` *Summary the output from the response surface method of maximising the Monte Carlo likelihood*

Description

This function summarizes the output of the output from the function rsmMCL.

Usage

```
## S3 method for class 'rsmMCL'
summary(object, family, trace.all = TRUE, mc.covar =
TRUE, ...)
```

Arguments

`object`	an rsmMCL object returned by rsmMCL.
`family`	a character takes value in "gauss", "binom" and "poisson".
`trace.all`	an logic value tells whether the input object given by rsmMCL contains results from all iterations of not
`mc.covar`	if TRUE, the estimated covariance matrix of the MC-MLE is returned
`...`	arguments passed to or from other methods.

Value

A list containing the following objects:

MC.mle, the final MC-MLE

N.iter, the total number of iterations

total.time, the total time elapsed

convergence, if TRUE the procedure converges

hessian, the Hessian at the MC-MLE if given; the default is NULL

mc.covar the estimated covariance matrix of the MC-MLE if given; the default is NULL

mc.samples the Monte Carlo samples size used in the initial stage and after the first convergence.

Author(s)

Zhe Sha <sha@stats.ox.ac.uk>

See Also

rsmMCL

Examples

```
## See examples for rsmMCL
```