

THE INSTITUTE
OF STATISTICS

THE UNIVERSITY OF
NORTH CAROLINA



PARAMETER ESTIMATION AND MODEL SELECTION
IN IMAGE ANALYSIS USING GIBBS-MARKOV RANDOM FIELDS
(DISSERTATION)

by

P.L. Seymour

December 1993

Mimeo Series #23137

DEPARTMENT OF STATISTICS
Chapel Hill, North Carolina

Mimeo P.L. Seymour

Series #2313 PARAMETER ESTIMATION
AND MODEL SELECTION IN
IMAGE ANALYSIS USING
GIBBS-MARKOV RANDOM FIELDS

NAMW

DATE

PARAMETER ESTIMATION AND
MODEL SELECTION IN IMAGE ANALYSIS
USING GIBBS-MARKOV RANDOM FIELDS

Peggy Lynne Seymour

A dissertation submitted to the faculty of The University of North Carolina in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Statistics.

Chapel Hill

10 December 1993

Approved by:

Chambliss Jr Advisor

G. Kallianpur Reader

R. L. Smith Reader

**PEGGY LYNNE SEYMOUR. Parameter Estimation and Model Selection in Image Analysis
using Gibbs-Markov Random Fields (Under the direction of Chuanshu Ji.)**

ABSTRACT

Researchers in the field of statistical image analysis are concerned with different issues, such as image restoration, boundary detection, or even object recognition, which may be used in such different contexts as images returned by satellite or medical images produced by emission tomography. There are, of course, many other issues one might address in using statistics to analyze an image. This particular research focuses on the selection of a model for a digital image.

Although model selection has been studied extensively in many areas of statistics, very little has been done within the context of image analysis. Thus this research is restricted to the most elementary images: those which are of a single texture (*i.e.*, an image which, in its entirety, is nothing but carpet, wood grain, clouds in the sky, or some other single type of “texture”). The models under consideration are parametric Gibbs-Markov random fields.

Parameter estimation is then a critical matter. The maximum likelihood estimator (MLE) is quite intractable for such models. This research focuses on two alternatives to the MLE: a Monte Carlo maximum likelihood estimate (MCMLE), and the maximum pseudo-likelihood estimate (MPLE). Asymptotic rates for the mean square error and for a moderate deviation probability are derived for the MPLE.

The main goal of this research is the development of information criteria for choosing a model, similar to the Bayesian information criteria used in model selection for time series and for exponential families. We establish criteria based on the MLE, the MCMLE and the MPLE. We show that the criteria based on the MLE and MCMLE are both approximations to the true Bayes solution to the model selection problem; and we also show the (weak) consistency of the criterion based on the MPLE.

A simulation study of the useful parameter estimation techniques and model selection criteria is presented, using several simple models. Implementation of the model selection criteria on real textures is also discussed.

Acknowledgement

I would first like to thank my advisor, Chuanshu Ji, for his enthusiastic support and guidance in this research, as well as for his efforts “above and beyond the call of duty” during my second year in helping me to realize this goal.

I would also like to thank my committee members, M. Ross Leadbetter, Gopinath Kallianpur, Gordon Simons, and Richard L. Smith for their participation in my research adventure. I owe a particularly deep debt of gratitude to Richard L. Smith for references and many, many helpful suggestions.

I am especially grateful to Stamatis Cambanis for his efforts, also “above and beyond the call of duty,” during my second year. My thanks also go to Edward Carlstein for several enlightening conversations.

I would like to express my most profound gratitude to my *sensei*, Shinkichi Sasaki. He has taught me so much more than karate: he has purged the notion that “I can’t do it” from my mind, and has instilled in me a self-awareness that has benefited me in all aspects of my life.

I am grateful for the support of my close friends: Courtenay Stark, who has looked into my bare soul without cringing; Charu Krishnamoorthy and Steve Preissler, who have shared with me many deep conversations, hours of Star Trek, and movies at the Yorktowne; and Jim and Wendy Curtis, with whom I have shared Thanksgivings, Durham Bulls games, NASCAR races, and general country rowdiness.

Of course, none of this effort would have been possible without the love and support of my family: my spouse, Robert Lund; my sister, Sandi Seymour Hussey; my parents, Eddie and Peggy Seymour; my aunt, Jane Austin; my grandparents, Howard and Carmye Austin and Mary Jo Lamb; and my late great-grandparents, Walter and Allie Benton and Susie Deason, who died after I was well into my teen years, and well after they had left their marks upon my life. I am a reflection of all of these people.

Finally, I would like to acknowledge Troy State University for the opportunity to attend college, the Department of Mathematics at Auburn University for the opportunity to learn mathematics, and the Department of Statistics at The University of North Carolina for the opportunity to earn an education which is better than any I ever expected.

The more things change, the more they stay the same.

Contents

I. INTRODUCTION TO STATISTICAL IMAGE ANALYSIS AND GIBBS-MARKOV RANDOM FIELDS	1
1.1 Statistics in Image Analysis	1
1.2 Goals of This Work	4
1.3 Introduction to Gibbs-Markov Random Fields	5
II. PARAMETER ESTIMATION FOR GIBBS-MARKOV RANDOM FIELDS.....	11
2.1 Parameter Estimation using the Likelihood.....	11
2.2 Parameter Estimation using the Markov Chain Monte Carlo Likelihood	16
2.3 Parameter Estimation using Besag's Pseudo-Likelihood	19
III. MODEL SELECTION CRITERIA FOR GIBBS-MARKOV RANDOM FIELDS	25
3.1 Perspective on Relevant Model Selection Criteria.....	25
3.2 Framework for Gibbs-Markov Model Selection.....	27
3.3 Model Selection Based on the MLE.....	29
3.4 Model Selection Based on the MCMLE	35
3.5 Model Selection Based on the MPLE	37
IV. SIMULATIONS AND NUMERICAL COMPARISONS	40
4.1 Practical Considerations	40
4.2 The Gibbs Sampler	41
4.3 Models used for the Simulation Studies.....	42
4.4 Simulation Study of the Parameter Estimates.....	44
4.5 Simulation Study of the Model Selection Procedures	51
4.6 Remarks on Application to Real Textures.....	62
APPENDIX ONE. PROOFS OF LEMMAS	63
A.1 Notation Reminder plus Two Supporting Lemmas	63
A.2 Proofs of Lemmas from Chapter II	65
A.3 Proofs of Lemmas from Chapter III	87
APPENDIX TWO. SIMULATED TEXTURES	92
APPENDIX THREE. COMPUTER PROGRAMS	109
RFGEN	109
DRIVER of PRAMEST	114
MODSEL	118
REFERENCES	136

List of Tables

Table 4.4.1:	MPLE for Model 1	45
Table 4.4.2:	MPLE for Model 2	45
Table 4.4.3:	MPLE for Model 3	46
Table 4.4.4:	MPLE vs. MCMLE for Model 1	47
Table 4.4.5:	MPLE vs. MCMLE for Model 2	48
Table 4.4.6:	MPLE vs. MCMLE for Model 3	49
Table 4.4.7:	Variance Reduction Schemes	50
Table 4.5.1:	MCMLE-Based Model Selection with the Uniqueness Condition	52
Table 4.5.2:	MCMLE-Based Model Selection without the Uniqueness Condition	53
Table 4.5.3:	MPLE-Based Model Selection under the Uniqueness Condition when the True Model is Model 1	54
Table 4.5.4:	MPLE-Based Model Selection without the Uniqueness Condition when the True Model is Model 1	54
Table 4.5.5:	Extreme Cases: True Model is Model 1.....	55
Table 4.5.6:	MPLE-Based Model Selection under the Uniqueness Condition when the True Model is Model 2	56
Table 4.5.7:	MPLE-Based Model Selection without the Uniqueness Condition when the True Model is Model 2	57
Table 4.5.8:	Extreme Cases: True Model is Model 2.....	58
Table 4.5.9:	MPLE-Based Model Selection under the Uniqueness Condition when the True Model is Model 3	59
Table 4.5.10:	MPLE-Based Model Selection without the Uniqueness Condition when the True Model is Model 3	60
Table 4.5.11:	Extreme Cases: True Model is Model 3.....	61

List of Figures

Figure 1.3.1: The “Nearest Neighbors” Neighborhood System.....	6
Figure 1.3.2: A More Complex Neighborhood System.....	6
Figure 2.1.1: Tiling.....	11
Figure 3.2.1: Four Nearest Neighbors, $\theta \in \mathbf{R}^1$	27
Figure 3.2.2: Four Nearest Neighbors, $\theta \in \mathbf{R}^2$	27
Figure 3.2.3: Twelve Nearest Neighbors, $\theta \in \mathbf{R}^2$	28
Figure 4.3.1: Neighborhood System and Parameter for Model 1	43
Figure 4.3.2: Neighborhood System and Parameters for Model 2.....	43
Figure 4.3.3: Neighborhood System and Parameters for Model 3.....	43

I. Introduction to Statistical Image Analysis and Gibbs-Markov Random Fields

1.1 Statistics in Image Analysis

Digitized images are very commonplace today with ever-stronger computer technology, and the need for techniques by which these images are analyzed grows stronger as well. A wealth of methods for addressing image analysis has already been developed by computer scientists and engineers; this may be verified simply by browsing through their literature. In fact, the new literature in image analysis has become so large that, in 1992, the Institute of Electrical and Electronics Engineers introduced a new journal: *IEEE Transactions on Image Processing*.

Some of the current methods in image analysis are based on statistical ideas, but most are not. However, statisticians have contributed richly to image analysis via filtering techniques, compression schemes, probability models, and estimation techniques for various image attributes. Since the field of statistical image analysis is still quite young, there are many open problems needing attention. This presentation is primarily concerned with parameter estimation and modelling, though a few other aspects are briefly discussed below.

Hassner and Sklansky (1978, 1980) first proposed Markov random fields as a statistical model for digital images. Both Markov random fields and Gibbs random fields, which are equivalent under certain conditions, were originally developed in statistical mechanics for modeling particle interactions on an integer lattice. The roots of these particular models go back as far as Gibbs (1902) and Ising (1925), but the modern understanding of them was introduced independently by Dobrushin (1968a, 1968b) and Lanford and Ruelle (1969). This approach by Hassner and Sklansky (1980) apparently did not fare well at first because of its computational and mathematical difficulties.

Geman and Geman (1984) gave new life to this approach by making these random field models more practical for image analysis. In their paper, they re-introduced Gibbs-Markov random fields as probability models for images. They introduced Bayesian statistical methodology to image analysis, using the Gibbs-Markov random field as a prior on the image space. (Besag, 1989, gives a good overview of Bayesian image analysis and discusses its potential for

development.) In addition, they developed two stochastic relaxation (*i.e.*, site-by-site updating) algorithms for image analysis: the Gibbs sampler for simulation, and simulated annealing for optimization. Their framework is as follows.

For a digitized image, particles in the original physics models correspond to pixels (picture elements on a computer screen), and interactions between particles correspond to dependence between pixel attributes, such as color, in the image. Using a Bayesian framework, a Gibbs-Markov random field is treated as a prior probability distribution on the image space. This prior incorporates knowledge about the global image as well as local pixel interactions. Inference about the true image is normally made based upon an observed corrupted image, and then the posterior mode(s) or mean can be taken as an estimate of the true image. As an example, consider the medical imaging technique of emission tomography. Here, the true image is the physical distribution of a radioactive isotope in a specific region of the body; the prior probability model will include relevant information about that region of the body. The emission of photons from the isotope is recorded, and a degraded image based on these emissions is observed.

There has been tremendous growth in statistical image analysis since the paper of Geman and Geman (1984). The following discussion gives a sample of some of the activity in statistical image analysis today. Along with a short description of the topic at hand, references are given from which the interested reader may begin further pursuit. The monograph of Geman (1991) as well as the paper by Karr (1991) give a comprehensive presentation of topics in statistical image analysis. Also, the brief overview of image modelling by Rosenfeld (1993) includes a healthy list of references.

Image Restoration. Image restoration involves the recovery of a true image from an observed corrupted image. In reality, images are nearly always degraded by some sort of noise. This degradation might be blur due to faulty optics, as seen in the Hubble Space Telescope's images; it might be Gaussian noise, due to faulty data transmission; or it might even be scratches on old movie film. Whatever the case may be, removal of such distortion is the primary concern.

Geman and Geman (1984) discuss image restoration using Gibbs-Markov random fields and their simulated annealing algorithm for recovering the true image. Besag (1986) uses Markov random field models and proposes another algorithm, called iterated conditional modes, for restoring a degraded image.

Boundary Detection. Boundary detection is the determination of physical discontinuities in an image. Such discontinuities might include actual boundaries between different regions of pixels in an image, or might include changes in elevation found in an original three-

dimensional scene.

Geman, Geman, and Graffigne (1987) propose locating texture and object boundaries by using Gibbs-Markov random field models which incorporate both pixel intensities and labels. (These are the same models which are described below in more detail.) Geman, Geman, Graffigne, and Dong (1990) use a similar model with built-in constraints which identify "forbidden" configurations.

Texture Segmentation and Synthesis. The concept of "texture" has so far eluded a precise mathematical definition. Intuitively, a texture is a region of pixels which exhibits not only global regularity (including periodicities), but also local variability (*e.g.*, wood grain, sand, carpet). Texture segmentation is the decomposition of an image into sets of pixels corresponding to different textures in the image. Here, the true image consists of two arrays: an array of pixel intensities, in which each pixel is assigned a value representing a color, or gray level, in addition to a corresponding array of texture labels, in which each pixel is assigned a number representing the texture to which it belongs. The pixel intensities are observed, and the challenge is to discover the texture labels.

One aspect of segmentation is texture discrimination, in which there are a known number of textures in the image, and the goal is to classify each pixel according to the type of texture to which it belongs. Another aspect of segmentation is texture identification, which is concerned with identifying the textures in the image by comparing them to "training samples," each sample consisting of only one of the textures in question. The distinction between discrimination and identification may not be clear, and the following example may be helpful. An early-warning radar system is performing discrimination when it classifies an object in its sights as "a possible threat" or "not a possible threat." This system would be performing identification if it could identify a flock of birds as such rather than simply classifying it as "not a possible threat."

Geman and Graffigne (1986) use the Gibbs-Markov random field models described above in the context of texture segmentation, and they employ a parameter estimation scheme based on the pseudo-likelihood proposed by Besag (1974). Derin and Elliott (1987) use a hierarchical Gibbs random field model to account for textures as well as additive independent Gaussian noise, and propose a linear least-squares parameter estimation scheme. Hu and Fahmy (1992) propose a hierarchical Markov random field, combining the autobinomial and autologistic models of Besag (1974), to segment an image.

Texture synthesis is the simulation of a real texture based on a given model. Hassner and Sklansky (1980) generate some synthetic textures using isotropic Markov random field models and a sampling scheme much like the Metropolis algorithm (Metropolis, *et.al.*, 1953; the

precursor to the Gibbs sampler). Cross and Jain (1983) use the autobinomial model introduced by Besag (1974) for synthesis. Acuna (1992) uses a modification of the autobinomial model to generate some synthetic textures.

A review of some current aspects of texture analysis in general – both statistical and non-statistical – is given in Tuceryan and Jain (1992).

Object Recognition. Object recognition is currently an extremely difficult and unyielding area of research. Within the statistical community, a great deal of exciting work is being done, particularly by D. Geman, S. Geman, and U. Grenander; but little of their work in object recognition has been published to date.

Clearly, there is significant activity in statistical image analysis. The areas mentioned here are not all-inclusive; however, we hope that the reader has gained the understanding that this is a very broad and potentially useful field. A statistician does not have to look very far to find open problems in this field, and within our own context, we shall point out some such problems as they arise.

1.2 Goals of This Work

This dissertation is primarily concerned with the selection of Gibbs-Markov random field texture models, which is motivated by texture synthesis and analysis. Throughout this work, we will restrict ourselves to an image with no degradation. However, since degradation is part of reality, future work in model selection will have to address such a difficulty. We will also restrict ourselves to an image which consists of a single texture, and then concern ourselves with selecting a model for that one texture. Again due to the nature of reality, future work will need to address model selection in a multiple-texture setting.

First, Chapter I continues with a discussion of the random field framework within which criteria for model selection are developed. This will include notation, definitions, intuition, and pertinent results, as well as discussion of complications which are inevitable when working with Gibbs random fields.

Chapter II discusses the important problem of parameter estimation for Gibbs-Markov random fields. The notion of exponential families for these models is introduced, and maximum likelihood parameter estimation is presented. Specifically, there are some serious difficulties with maximum likelihood estimation in this context which cannot be resolved. Two current alternative methods of parameter estimation are discussed – a Monte Carlo method and a pseudo-likelihood method – and some useful results are proven.

Chapter III discusses the main point of model selection, of which parameter estimation will be a crucial component. A thorough motivation is given, and model selection criteria based on the three parameter estimates discussed in Chapter II are derived. Rigorous justification for each of the selection criteria is also presented.

Chapter IV presents a numerical comparison of the implementable parameter estimation techniques from Chapter II, as well as a discussion of some of the difficulties involved in using these techniques. Chapter IV also presents a numerical comparison of the implementable model selection criteria, and a further study of the superior criterion. Chapter IV concludes with a discussion of the implementation of the superior criterion on real textures.

1.3 Introduction to Gibbs-Markov Random Fields.

This section gives a very simple treatment of Markov random fields and Gibbs random fields, which is enough for our purposes: for a more general treatment, see Ruelle (1978) or Georgii (1988).

Let Λ be a finite lattice in \mathbf{Z}^2 , and let X_i be a random variable associated with the site $i \in \mathbf{Z}^2$. For intuition on real applications, the region Λ can be understood as the computer screen or a region on the computer screen, and X_i can be considered a gray level (color or shade of gray) at pixel i . The following definitions are very basic.

Definition 1.3.1 The random array $X = \{X_i; i \in \mathbf{Z}^2\}$ is called a random field.

Definition 1.3.2 The state space S , a finite set with discrete topology, is the collection of all possible values of X_i for each $i \in \mathbf{Z}^2$.

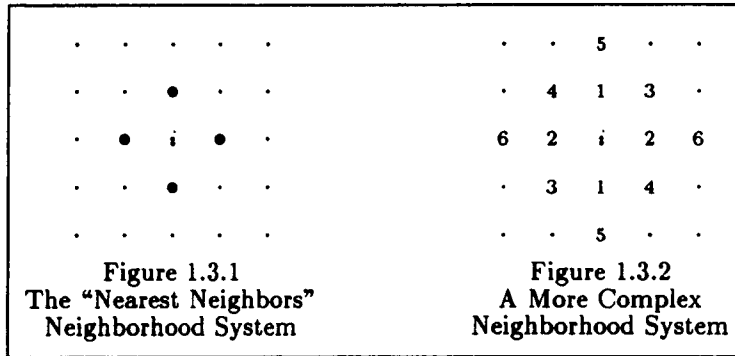
Definition 1.3.3 The configuration space $\Omega = S^{\mathbf{Z}^2}$ is the collection of all possible realizations of the random field X . Correspondingly, the sub-configuration space on the region Λ is given by $\Omega_\Lambda = S^\Lambda$.

For notation, write $X_\Lambda = \{X_i; i \in \Lambda\}$, and ${}_i X = \{X_j; j \neq i\}$. Write $x \in \Omega$ for a realization of X , and $x_i \in S$ for a realization of X_i ; in the same manner, let $x_\Lambda \in \Omega_\Lambda$, and ${}_i x = \{x_j; j \neq i\}$ denote realizations of X_Λ , and ${}_i X$, respectively.

Definition 1.3.4 For a site $i \in \mathbf{Z}^2$, a collection N_i of sites having the properties $i \notin N_i$, and $i \in N_j$ if and only if $j \in N_i$, is called the neighborhood of the site i . The collection

$\mathfrak{N} = \{\mathcal{N}_i; i \in \mathbb{Z}^2\}$ of all neighborhoods is called the neighborhood system.

For example, one neighborhood system might designate the neighbors of a site i to be the closest horizontal and vertical sites to i , thus giving i four neighbors as in Figure 1.3.1. This particular set of neighbors is conventionally referred to as nearest neighbors. A more complex neighborhood system might be the one which is represented in Figure 1.3.2, composed of six types of neighbors – twelve neighbors in all. In general, the neighbors of a site i are the sites on which i is (mathematically) dependent. Different textures may have different neighborhood systems: if the texture is wood grain, then a pixel will have one kind of neighbor “with the grain” of the wood, and another kind of neighbor “against the grain” of the wood; if the texture is sand, then all of the neighbors will be of the same kind.



Definition 1.3.5 The random field X is called a Markov random field (MRF) with respect to the neighborhood system \mathfrak{N} if its probability distribution P on Ω satisfies

$$P(X_i = x_i | X = x) = P(X_i = x_i | X_{\mathcal{N}_i} = x_{\mathcal{N}_i}) \quad (1.3.1)$$

for each $i \in \mathbb{Z}^2$ and $x \in \Omega$. These single-site conditional probabilities are called the local characteristics of the MRF X .

For convenience, we will write the local characteristics as

$$p_i(x) \triangleq P(X_i = x_i | X = x) \quad (1.3.2)$$

for all sites $i \in \mathbb{Z}^2$.

Most of the terminology contained in these next definitions is taken directly from physics, and thus has no intuitive meaning for the statistician even though the terminology is standard. These notions are, however, central to the definition of a Gibbs random field.

Let $\|k\|$ for $k \in \mathbb{Z}^2$ be a norm on \mathbb{Z}^2 . Usually, we consider the norm defined by $\|k\| = |k_1| + |k_2|$ for $k = (k_1, k_2)$, which is called the Manhattan distance or city-block norm (Possolo, 1991). But one may also use some other norm on \mathbb{Z}^2 .

Definition 1.3.6 Let $R > 0$, and define the function $\mathfrak{u}_A: \Omega \rightarrow \mathbb{R}$ by

$$\mathfrak{u}_A(x) = \begin{cases} \mathfrak{u}_i(x_i) & \text{if } A = \{i\} \\ \frac{1}{2} \mathfrak{u}_{ij}(x_i, x_j) & \text{if } A = \{i, j\}, 0 < \|i - j\| \leq R \\ 0 & \text{otherwise} \end{cases} \quad (1.3.3)$$

for some functions $\mathfrak{u}_i: S \rightarrow \mathbb{R}$ and symmetric $\mathfrak{u}_{ij}: S \times S \rightarrow \mathbb{R}$, $i, j \in \mathbb{Z}^2$. Then the collection of such functions $\mathfrak{U} = \{\mathfrak{u}_A: \Omega \rightarrow \mathbb{R}, A \subset \mathbb{Z}^2, |A| \leq 2\}$ is called a pair-potential of range R , where $|A|$ denotes the cardinality of the set A .

The pair-potential simply describes in some deterministic way how the values at sites i and j interact. Note that if $\|i - j\| > R$, then sites i and j have no direct interaction together. Note also that other potentials, not just those involving pairs of sites, may be considered as well. We restrict our attention to pair-potentials for the sake of both precedent and simplicity.

Definition 1.3.7 For every finite $\Lambda \subset \mathbb{Z}^2$ and $x \in \Omega$,

$$H_\Lambda(x) = - \sum_{i \in \Lambda} \mathfrak{u}_i(x_i) - \frac{1}{2} \sum_{\substack{i, j \in \Lambda \\ i < j}} \mathfrak{u}_{ij}(x_i, x_j) - \sum_{\substack{i \in \Lambda \\ j \notin \Lambda}} \mathfrak{u}_{ij}(x_i, x_j) \quad (1.3.4)$$

is called the energy associated with x on Λ .

The energy may be understood as the total amount of interaction taking place in the region Λ . In particular, notice that the middle sum in (1.3.4) is halved because this sum alone is counting all pair-interactions within Λ twice.

Definition 1.3.8 Write $H_\Lambda(x) = H_\Lambda(x_\Lambda; x_{\Lambda^c})$. The random field X is called a Gibbs random field (GRF) induced by the pair-potential \mathfrak{U} if its probability distribution P on Ω satisfies

$$P(X_\Lambda = x_\Lambda \mid X_{\Lambda^c} = x_{\Lambda^c}) = \frac{\exp[-H_\Lambda(x)]}{\mathfrak{Z}(x_{\Lambda^c})} \quad (1.3.5)$$

for each finite $\Lambda \subset \mathbb{Z}^2$ and $x \in \Omega$. The normalizing factor $\mathfrak{Z}(\cdot)$, which is called the partition function, takes the form

$$\mathfrak{Z}(x_{\Lambda^c}) = \sum_{y \in \Omega_\Lambda} \exp[-H_\Lambda(y; x_{\Lambda^c})]. \quad (1.3.6)$$

One problem which will be addressed later in Chapter II is that, in all but the most trivial cases, the partition function is computationally intractable.

It was mentioned in Section 1.1 that, under certain conditions, a MRF is equivalent to a GRF. Several researchers proved the equivalence under very specific conditions, including Spitzer (1971) and Sherman (1973). However, in an unpublished manuscript, Hammersley and Clifford (see Besag, 1974 and Geman, 1991 for further discussion and references), proved that a random array on a finite region is a MRF if and only if it is a GRF induced by finite-range potentials. Thus, such random fields may be described at once locally, using a MRF model described by the collection of its local characteristics, or globally, using a GRF model described by its energy (total interactions) over a region. In light of this result, we shall now use the term Gibbs-Markov random field (GMRF) unless further explanation is needed.

A well-known special case of a GMRF is the two-dimensional Ising model (Ising, 1925), in which $S = \{-1, 1\}$, $R = 1$ and the pair-potential \mathfrak{U} satisfies

$$\begin{aligned} \mathfrak{U}_i(x_i) &= hx_i \\ \mathfrak{U}_{ij}(x) &= \beta x_i x_j, \quad \|i - j\| = 1, \end{aligned} \tag{1.3.7}$$

where $h \in \mathbf{R}$ is called the external field coefficient, and $\beta > 0$ is called the coupling coefficient. Note that $R = 1$ implies that the neighborhood system is nearest neighbors.

Definition 1.3.9 A potential \mathfrak{U} is said to be translation invariant if $\mathfrak{U}_{A+j}(\tau_j x) = \mathfrak{U}_A(x)$ for each $j \in \mathbf{Z}^2$ and $x \in \Omega$, where $A + j = \{i + j: i \in A\}$ and τ_j is the shift operator on Ω defined by $(\tau_j x)_i = x_{i-j}$, $i \in \mathbf{Z}^2$.

Definition 1.3.10 A random field X is said to be stationary under the measure P if for each $i \in \mathbf{Z}^2$, $x \in \Omega$, and finite $\Lambda \subset \mathbf{Z}^2$, we have $P(X_\Lambda = x_\Lambda) = P(X_{\Lambda+i} = x_\Lambda)$.

A translation invariant potential does not necessarily induce a stationary random field, which is a special case of a phenomenon known as symmetry breaking in statistical mechanics (Georgii, 1988). It is assumed throughout this work that single-texture models are induced by a translation invariant pair-potential. Clearly, this is an unreasonable assumption for multi-texture models since, for example, the model for wood grain is different from the model for sand (see the earlier discussion on neighborhood systems).

A difficult issue related to GRFs is that a potential \mathfrak{U} may induce more than one probability distribution on Ω which satisfies the conditional distribution (1.3.5). Such a phenomenon

is called phase transition in statistical mechanics; in statistical terminology, this translates to spatial long-range dependence (for illustration of the presence or absence of spatial long-range dependence, see the sample textures in Appendix Two). To illustrate this notion, let Λ_n be an $n \times n$ region centered at the origin o : when phase transitions occur, $P(X_o = x_o)$ is not necessarily equal to $\lim_{n \rightarrow \infty} P(X_o = x_o | X_{\Lambda_n^c} = x_{\Lambda_n^c})$.

As was noted in the introduction, one of our tasks is to specify a GMRF as the probability model on the configuration space. For our purposes, we need to specify the energy function, which involves the estimation of unknown parameters. Again, let Λ_n be an $n \times n$ square lattice centered at the origin o , and suppose we observe

$$x_{\Lambda_n} \triangleq x(n), \text{ a realization of the random field } X_{\Lambda_n} \triangleq X(n). \quad (1.3.8)$$

We assume that the probability distribution P which generates X is a GMRF induced by the finite-range potential \mathfrak{U} of the form

$$\begin{aligned} \mathfrak{U}_i(x_i) &= hU_1(x_i) \\ \mathfrak{U}_{ij}(x_i, x_j) &= \beta_{j-i}U_2(x_i, x_j) \end{aligned} \quad (1.3.9)$$

for all sites $i, j \in \mathbf{Z}^2$ and for some known functions $U_1: S \rightarrow \mathbf{R}$ and symmetric $U_2: S \times S \rightarrow \mathbf{R}$. For all $j \in \mathbf{Z}^2$, the parameters β_j satisfy the conditions

$$\begin{aligned} \beta_j &= \beta_{-j} \text{ when } 0 < \|j\| \leq R \\ \beta_j &= 0 \text{ otherwise,} \end{aligned} \quad (1.3.10)$$

where R is the range of the GMRF. The energy function for any single site i is then

$$H_{\{i\}}(\mathbf{x}) = -hU_1(x_i) - \sum_{j \in \mathcal{N}_i} \beta_{j-i}U_2(x_i, x_j) \quad (1.3.11)$$

for $\mathbf{x} \in \Omega$. Note that this form allows the incorporation of various kinds of pixel (pair) interactions. The local characteristic at i is given by

$$p_i(\mathbf{x}) = \frac{\exp[-H_{\{i\}}(\mathbf{x})]}{\mathfrak{Z}_{\{i\}}(\mathbf{x})} \quad (1.3.12)$$

for $\mathbf{x} \in \Omega$, with the partition function given by

$$\mathfrak{Z}_{\{i\}}(\mathbf{x}) = \sum_{s \in S} \exp[-H_{\{i\}}(s; \mathbf{x})] \quad (1.3.13)$$

The local characteristics on the finite region $\Lambda \subset \mathbb{Z}^2$ determine all Gibbs measures on Λ (D. Geman, 1991). Note that, since the pair-potentials are translation invariant, so are the local characteristics.

Dobrushin's condition (Dobrushin, 1968c) on the energy function is one which implies the uniqueness of the induced GRF. Recall that non-uniqueness of the GRF corresponds to the presence of phase transitions, or (spatial) long-range dependence. A slightly stronger condition, that of Simon (1979), implies Dobrushin's condition, and it seems easier to verify. The following condition, which will be employed only when explicitly indicated, gives Simon's uniqueness condition for the GRF in our set-up.

Uniqueness Condition 1.3.1 *The pair-potential \mathbf{U} satisfies*

$$\sum_{j \in \mathcal{N}_o} |\beta_j| \bar{U}_2 < 2 \quad (1.3.14)$$

where $\bar{U}_2 = \max_{x_o, x_j} U_2(x_o, x_j) - \min_{x_o, x_j} U_2(x_o, x_j)$.

Note that \bar{U}_2 does not depend on the site j . This inequality, although slightly stronger than that of Dobrushin's condition, is sharp: For every $\varepsilon > 0$ there is a pair-potential \mathbf{U} with $S = \{-1, 1\}$ for which

$$\sum_{j \in \mathcal{N}_o} |\beta_j| \bar{U}_2 < 2 + \varepsilon \quad (1.3.15)$$

holds, such that the GRF is not uniquely induced by \mathbf{U} , and phase transitions occur (Georgii, 1988, p. 144).

Take the parameters β_j together as a column vector β , and write another column vector $\theta = (h, \beta^T)^T \in \Theta = \mathbf{R}^k$ to parametrize the entire energy function. To indicate the dependence of the local characteristics on this parameter θ , write

$$p_i(x; \theta) \triangleq p_i(x). \quad (1.3.16)$$

Definition 1.3.11 *A parameter $\theta \in \Theta$ is called identifiable if $p_i(x; \theta) \neq p_i(x; \theta^*)$ for some $x \in \Omega$ when $\theta \neq \theta^*$, $\theta^* \in \Theta$.*

Now, the task is to estimate the parameter $\theta \in \mathbf{R}^k$ based on the observation $x(n)$.

II. Parameter Estimation for Gibbs-Markov Random Fields

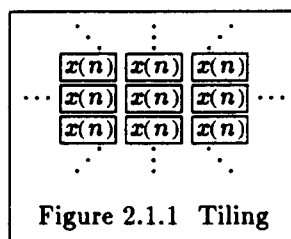
2.1 Parameter Estimation using the Likelihood

Parameter estimation within the context presented in Chapter I bears some interesting features. One is that a single sample, rather than repeated independent samples, of $X(n)$ must be used for estimation. The observations contained in that one sample of $X(n)$ are dependent, and the potential for long-range dependence must be addressed. This sampling also renders impractical the traditional notions of asymptotics, which must now be taken as the dimension n of the sample $x(n)$ goes to infinity.

Another feature is that, because of the dependence, it is extremely difficult to use the usual likelihood function for $X(n)$, which is

$$P_\theta(X(n) = x(n)) = \int_{\Omega_{\Lambda_n^c}} P_\theta(X(n) = x(n) | X_{\Lambda_n^c} = x_{\Lambda_n^c}) dP_\theta(x_{\Lambda_n^c}), \quad (2.1.1)$$

where P_θ is written to emphasize the dependence of the distribution P on the parameter θ . A more convenient conditional likelihood is defined as follows. Extend the observation $x(n)$ to a configuration on \mathbf{Z}^2 by periodization, or tiling, as illustrated in Figure 2.1.1 below.



Denote this periodic extension of $x(n)$ by $\tilde{x} \in \Omega$. Note that this notation suppresses the dependence on the dimension n , which is done just for convenience. Note also that $\tilde{x}_{\Lambda_n} = x(n)$. With this periodic configuration, the conditional likelihood function,

$$\mathcal{L}(x(n), \theta) \triangleq P_\theta(X(n) = \tilde{x}_{\Lambda_n} | X_{\Lambda_n^c} = \tilde{x}_{\Lambda_n^c}) = \frac{\exp[-H_{\Lambda_n}(\tilde{x})]}{\mathcal{Z}(\tilde{x}_{\Lambda_n^c})}, \quad (2.1.2)$$

may be used. Call any value which maximizes $\mathcal{L}(x(n), \cdot)$ a maximum likelihood estimate (MLE) of the parameter θ . Within this context, we will need the following condition for nearly

all of our results.

Identifiability Condition 2.1.1 *The true parameter θ is identifiable in accordance with Definition 1.3.11.*

The MLE is not the only parameter estimate available for random fields. Two other parameter estimation techniques will be discussed in this chapter, but each one bears some similarities to the MLE. Other available estimates include least-squares and method-of-moments estimators. The least-squares alternative, proposed by Derin and Elliott (1987), seems to work best for simple neighborhood systems, and also seems to have some difficulty with long-range dependence. The method-of-moments estimator, proposed by Possolo (1991), is valid for stationary random fields. The MLE and the alternatives presented here have some very desirable properties not yet established for other estimates.

The following definition will be useful in the upcoming discussion.

Definition 2.1.1 *The boundary $\partial\Lambda$ of the region $\Lambda \subset \mathbb{Z}^2$ is given by $\partial\Lambda = \{j \in \mathcal{N}_i; i \in \Lambda\} \setminus \Lambda$.*

In plain English, $\partial\Lambda$ is the set of all sites which are neighbors of sites in Λ but are not contained in Λ .

The conditional likelihood in (2.1.2) may be written as an exponential family:

$$\mathcal{L}(x(n), \theta) = \exp\left\{|\Lambda_n|[\theta^T Y_n - b_n(\theta)]\right\}. \quad (2.1.3)$$

Recall that θ is the column vector $(h, \beta^T)^T$, as defined in Section 1.3. The forms of both the sufficient statistic Y_n and the cumulant generating function $b_n(\theta)$ may now be derived. By explicitly equating (2.1.2) and (2.1.3), we get

$$\frac{\exp\{|\Lambda_n|\theta^T Y_n\}}{\exp\{|\Lambda_n|b_n(\theta)\}} = \frac{\exp[-H_{\Lambda_n}(\tilde{x})]}{\mathfrak{Z}(\tilde{x}_{\Lambda_n^c})}. \quad (2.1.4)$$

If we define

$$|\Lambda_n|b_n(\theta) = \log \mathfrak{Z}(\tilde{x}_{\Lambda_n^c}), \quad (2.1.5)$$

then it follows that

$$|\Lambda_n| \theta^T Y_n = -H_{\Lambda_n}(\tilde{x}). \quad (2.1.6)$$

Clearly, (2.1.6) implies that Y_n is a function of the periodic configuration \tilde{x} based on the observation $x(n)$. Hence, Y_n should be understood to be a function of such a configuration: $Y(\tilde{x}_{\Lambda_n}; \tilde{x}_{\Lambda_n^c})$.

First, we will find the components of Y_n . Toward this goal, we see that the expression for the energy function (1.3.4) plugged into equation (2.1.5) yields

$$|\Lambda_n| \theta^T Y_n = \sum_{i \in \Lambda_n} \mathfrak{u}_i(\tilde{x}_i) + \frac{1}{2} \sum_{i, j \in \Lambda_n} \mathfrak{u}_{ij}(\tilde{x}_i, \tilde{x}_j) + \sum_{\substack{i \in \Lambda_n \\ j \notin \Lambda_n}} \mathfrak{u}_{ij}(\tilde{x}_i, \tilde{x}_j). \quad (2.1.7)$$

Since we are using the potentials given in (1.3.9), we get

$$\theta^T Y_n = \frac{1}{|\Lambda_n|} \left[\sum_{i \in \Lambda_n} h U_1(\tilde{x}_i) + \frac{1}{2} \sum_{i, j \in \Lambda_n} \beta_{j-i} U_2(\tilde{x}_i, \tilde{x}_j) + \sum_{\substack{i \in \Lambda_n \\ j \in \partial \Lambda_n}} \beta_{j-i} U_2(\tilde{x}_i, \tilde{x}_j) \right]. \quad (2.1.8)$$

Note that \mathcal{N}_o is the set of neighbors about the origin o . Because of translation invariance of the potential, we can redefine the index j so that we are summing only on i , and rearrange the terms in (2.1.8) appropriately:

$$\begin{aligned} &= h \frac{1}{|\Lambda_n|} \sum_{i \in \Lambda_n} U_1(\tilde{x}_i) + \\ &\quad \sum_{j \in \mathcal{N}_o} \beta_j \frac{1}{|\Lambda_n|} \left[\sum_{\substack{i \in \Lambda_n \\ i+j \in \Lambda_n}} U_2(\tilde{x}_i, \tilde{x}_{i+j}) + \sum_{\substack{i \in \Lambda_n \\ i+j \in \partial \Lambda_n}} U_2(\tilde{x}_i, \tilde{x}_{i+j}) \right]. \end{aligned} \quad (2.1.9)$$

Hence, the component of Y_n corresponding to h is

$$\frac{1}{|\Lambda_n|} \sum_{i \in \Lambda_n} U_1(\tilde{x}_i) \quad (2.1.10)$$

and the component of Y_n corresponding to β_j is

$$\frac{1}{|\Lambda_n|} \left[\sum_{\substack{i \in \Lambda_n \\ i+j \in \Lambda_n}} U_2(\tilde{x}_i, \tilde{x}_{i+j}) + \sum_{\substack{i \in \Lambda_n \\ i+j \in \partial \Lambda_n}} U_2(\tilde{x}_i, \tilde{x}_{i+j}) \right], \quad (2.1.11)$$

where the first sum accounts for the interactions within the region Λ_n , and the second sum accounts for the interactions involving the boundary of Λ_n .

Next, we derive the cumulant generating function. Using the expression for the parti-

tion function in (1.3.6) and the notation introduced in (2.1.3), the cumulant generating function may be written

$$\begin{aligned}
b_n(\theta) &= \frac{1}{|\Lambda_n|} \log \mathfrak{Z}(\tilde{\mathbf{x}}_{\Lambda_n^c}) \\
&= \frac{1}{|\Lambda_n|} \log \sum_{\mathbf{y} \in \Omega_{\Lambda_n}} \exp[-H_{\Lambda_n}(\mathbf{y}; \tilde{\mathbf{x}}_{\Lambda_n^c})] \\
&= \frac{1}{|\Lambda_n|} \log \sum_{\mathbf{y} \in \Omega_{\Lambda_n}} \exp\{|\Lambda_n| \theta^T Y(\mathbf{y}; \tilde{\mathbf{x}}_{\Lambda_n^c})\} \tag{2.1.12}
\end{aligned}$$

Within this framework, we will require the following condition, which indicates that the covariance matrix of Y_n is positive definite (needed for strict convexity of the likelihood function). This condition is difficult to check, but is not without precedent (Gidas, 1986).

Positivity Condition 2.1.2 For every $\theta \in \mathbf{R}^k$, there exists a positive definite matrix $B(\theta)$ such that

$$B(\theta) = \liminf_{n \rightarrow \infty} |\Lambda_n| E_{\theta}[(Y_n - E_{\theta}Y_n)(Y_n - E_{\theta}Y_n)^T] \tag{2.1.13}$$

where $E_{\theta}(\cdot)$ is the expectation with respect to P_{θ} .

The following discussion gives existence, uniqueness, and consistency properties for the MLE. Because of the Markov property, we may understand that Y_n depends on $\tilde{\mathbf{x}}_{\Lambda_n^c}$ only through $\tilde{\mathbf{x}}_{\partial\Lambda_n}$: $Y(\cdot; \tilde{\mathbf{x}}_{\Lambda_n^c}) = Y(\cdot; \tilde{\mathbf{x}}_{\partial\Lambda_n})$. Therefore, for ease of notation later, we will write $Y(\cdot; \tilde{\mathbf{x}}_{\partial\Lambda_n})$ as simply $Y(\cdot)$, with the understanding that $Y(\cdot)$ is dependent upon the configuration $\tilde{\mathbf{x}}_{\partial\Lambda_n}$.

Note that when we wish to use a quantity as a function of the parameter space, we will use the notation ϑ in place of θ . Then, toward the goals of existence and uniqueness, we will require the gradient of $b_n(\vartheta)$ with respect to ϑ :

$$\begin{aligned}
\nabla b_n(\vartheta) &= \frac{1}{|\Lambda_n|} \cdot \frac{1}{\sum_{z \in \Omega_{\Lambda_n}} \exp\{|\Lambda_n| \vartheta^T Y(z; \tilde{\mathbf{x}}_{\partial\Lambda_n})\}} \cdot \\
&\quad \sum_{y \in \Omega_{\Lambda_n}} |\Lambda_n| Y(y; \tilde{\mathbf{x}}_{\partial\Lambda_n}) \exp\{|\Lambda_n| \vartheta^T Y(y; \tilde{\mathbf{x}}_{\partial\Lambda_n})\}
\end{aligned}$$

$$\begin{aligned}
&= \sum_{y \in \Omega_{\Lambda_n}} Y(y; \tilde{x}_{\partial\Lambda_n}) \cdot \frac{\exp\{|\Lambda_n| \vartheta^T Y(y; \tilde{x}_{\partial\Lambda_n})\}}{\sum_{z \in \Omega_{\Lambda_n}} \exp\{|\Lambda_n| \vartheta^T Y(z; \tilde{x}_{\partial\Lambda_n})\}} \\
&= E_{\vartheta}(Y_n | \tilde{x}_{\partial\Lambda_n}).
\end{aligned} \tag{2.1.14}$$

The likelihood equation may now be written as

$$\begin{aligned}
Y_n &= \nabla b_n(\vartheta) \\
&= E_{\vartheta}(Y_n | \tilde{x}_{\partial\Lambda_n})
\end{aligned} \tag{2.1.15}$$

where the conditional expectation of Y_n is over all possible configurations \tilde{x}_{Λ_n} with $\tilde{x}_{\partial\Lambda_n}$ fixed. Since Y_n is essentially an average, it follows from (2.1.14) and the ergodic theorem given by Georgii (1988, page 306, Theorem (14.A8)) that

$$\lim_{n \rightarrow \infty} [Y_n - \nabla b_n(\theta)] = 0, \quad P_{\theta}\text{-a.s.} \tag{2.1.16}$$

Let $\nabla^2 b_n(\vartheta)$ denote the Hessian matrix of $b_n(\vartheta)$ with respect to ϑ .

Lemma 2.1.1 *Under the Uniqueness Condition 1.3.1 and the Positivity Condition 2.1.2,*

$$c \leq v^T \nabla^2 b_n(\vartheta) v \leq C \tag{2.1.17}$$

for some constants $c, C > 0$; uniformly for all unit vectors $v \in \mathbf{R}^k$; all $\tilde{x}_{\partial\Lambda_n}$; all ϑ in a small neighborhood of θ ; and all large n .

Lemma 2.1.1 implies that there exists a small open neighborhood, say \mathcal{O} , of θ such that $\nabla b_n(\cdot)$ is a continuous bijection with a continuous inverse map (*i.e.*, a homeomorphism). In particular, $\nabla b_n(\mathcal{O})$ is an open region. It follows then from (2.1.16) that the likelihood equation (2.1.15) has a solution $\hat{\theta}_n$. Also by Lemma 2.1.1, the log of the likelihood (2.1.3) is locally strictly concave (*indeed*, this log-likelihood is globally concave because of (A.2.12)); hence $\hat{\theta}_n$ is the unique MLE. Finally, the following lemma, due to Com ets (1992), gives the strong consistency of $\hat{\theta}_n$.

Lemma 2.1.2 *Under the Identifiability Condition 2.1.1, for every $\varepsilon > 0$ there exist $c, C > 0$ such that*

$$P_{\theta}(\|\hat{\theta}_n - \theta\| > \varepsilon) \leq C \exp(-c|\Lambda_n|) \quad (2.1.18)$$

uniformly for all large n , where $\|\cdot\|$ is the Euclidian norm.

Notice that the uniqueness of the MLE is established only under the Uniqueness Condition 1.3.1 on the GRF (no phase transitions), but that the existence and consistency of the MLE do not need this condition.

In spite of these existence, uniqueness, and consistency properties, the MLE is useless for practical purposes because the partition function $\mathfrak{Z}(\cdot)$ of the (conditional) likelihood (2.1.2) is computationally intractable: even in the simple case of a 100×100 random field with each site taking values in $\{-1, 1\}$, the partition function is a sum of $2^{10,000}$ terms. Hence some alternative to the MLE must be pursued. Two alternative approaches are now presented: the first employs a Monte Carlo approximant to the likelihood; and then the second uses a pseudo-likelihood.

2.2 Parameter Estimation using the Markov Chain Monte Carlo Likelihood

Geyer and Thompson (1992) developed a method of Monte Carlo maximum likelihood estimation for an exponential family whose likelihood function cannot be calculated nor well-approximated. In fact, their result may be extended to a likelihood function which is not in the form of an exponential family. What follows is the development of their method within our framework.

Fix a sample $x(n)$ and let P_{ψ} be the Gibbs distribution with *known* parameter $\psi \in \mathbf{R}^k$. Simulate an ergodic Markov chain $X^{(1)}(n), X^{(2)}(n), \dots$ of random fields on Λ_n whose equilibrium distribution is the conditional distribution $P_{\psi}(\cdot | \tilde{x}_{\partial\Lambda_n})$ on Ω_{Λ_n} . Write the likelihood in the form

$$\mathfrak{L}(x(n), \theta) = \frac{\exp\{|\Lambda_n| \theta^T Y(\tilde{x}_{\Lambda_n})\}}{c(\theta)}, \quad (2.2.1)$$

where $c(\theta) = \mathfrak{Z}(\tilde{x}_{\Lambda_n^c})$ is the intractable partition function from (2.1.2). Using the notation introduced in (2.1.3), $c(\theta)$ is given by

$$c(\theta) = \sum_{y \in \Omega_{\Lambda_n}} \exp\{|\Lambda_n| \theta^T Y(y)\}. \quad (2.2.2)$$

Manipulate $c(\theta)$ into an expectation:

$$\begin{aligned}
c(\theta) &= \sum_{\mathbf{y} \in \Omega_{\Lambda_n}} \exp\{|\Lambda_n|(\theta - \psi)^\top Y(\mathbf{y})\} \exp\{|\Lambda_n| \psi^\top Y(\mathbf{y})\} \\
&= c(\psi) \sum_{\mathbf{y} \in \Omega_{\Lambda_n}} \exp\{|\Lambda_n|(\theta - \psi)^\top Y(\mathbf{y})\} \frac{\exp\{|\Lambda_n| \psi^\top Y(\mathbf{y})\}}{c(\psi)} \\
&= c(\psi) \sum_{\mathbf{y} \in \Omega_{\Lambda_n}} \exp\{|\Lambda_n|(\theta - \psi)^\top Y(\mathbf{y})\} P_\psi(\mathbf{y} | \tilde{\mathbf{x}}_{\partial\Lambda_n}) \\
&= c(\psi) E_\psi \left[\exp\{|\Lambda_n|(\theta - \psi)^\top Y_n\} \middle| \tilde{\mathbf{x}}_{\partial\Lambda_n} \right]. \tag{2.2.3}
\end{aligned}$$

We may then define the ratio

$$r(\theta) = \frac{c(\theta)}{c(\psi)} = E_\psi \left[\exp\{|\Lambda_n|(\theta - \psi)^\top Y_n\} \middle| \tilde{\mathbf{x}}_{\partial\Lambda_n} \right], \tag{2.2.4}$$

so that the log-likelihood, to within a multiplicative constant, may now be written as

$$\ell(\mathbf{x}(n), \theta) = \log[c(\psi) \mathcal{L}(\mathbf{x}(n), \theta)] = |\Lambda_n| \theta^\top Y(\tilde{\mathbf{x}}_{\Lambda_n}) - \log r(\theta). \tag{2.2.5}$$

Using the simulated Markov chain, define $r_L(\theta)$ as

$$r_L(\theta) = \frac{1}{L} \sum_{l=1}^L \exp\{|\Lambda_n|(\theta - \psi)^\top Y(X^{(l)}(n))\}. \tag{2.2.6}$$

Let \mathbf{P} denote the probability measure for the entire Markov chain $\{X^{(l)}(n)\}_{l=1}^\infty$. Due to the ergodicity of the Markov chain, we then have $r_L(\theta) \rightarrow r(\theta)$ \mathbf{P} -a.s. as $L \rightarrow \infty$. Thus a Monte Carlo approximation to the log-likelihood (2.2.5) is given by

$$\ell_L(\mathbf{x}(n), \theta) = |\Lambda_n| \theta^\top Y(\tilde{\mathbf{x}}_{\Lambda_n}) - \log r_L(\theta) \tag{2.2.7}$$

We call the value which maximizes $\ell_L(\mathbf{x}(n), \cdot)$ a Monte Carlo MLE (MCMLE), and denote it $\hat{\theta}_{nL}$.

The following lemmas are needed to establish the existence and uniqueness of $\hat{\theta}_{nL}$.

Lemma 2.2.1 *For every fixed $\vartheta \in \Theta$, we have*

$$\ell_L(\mathbf{x}(n), \vartheta) \rightarrow \log \mathcal{L}(\mathbf{x}(n), \vartheta) + |\Lambda_n| b_n(\psi), \tag{2.2.8}$$

$$\nabla \ell_L(\mathbf{x}(n), \vartheta) \rightarrow \nabla \log \mathcal{L}(\mathbf{x}(n), \vartheta), \tag{2.2.9}$$

and

$$\nabla^2 \ell_L(x(n), \vartheta) \rightarrow \nabla^2 \log \mathcal{L}(x(n), \vartheta), \quad (2.2.10)$$

all \mathbf{P} -a.s. as $L \rightarrow \infty$.

Choose η_θ to be a small compact neighborhood of the true parameter θ such that Lemma 2.1.1 holds for each $\vartheta \in \eta_\theta$ (i.e., η_θ is a small compact neighborhood of θ in which the log-likelihood is strictly concave). Let

$$\nu_{jL}(\vartheta) = \frac{\exp\left\{|\Lambda_n| (\vartheta - \psi)^T Y(X^{(j)}(n))\right\}}{\sum_{l=1}^L \exp\left\{|\Lambda_n| (\vartheta - \psi)^T Y(X^{(l)}(n))\right\}} \quad (2.2.11)$$

for $j = 1, \dots, L$. Then $\nu_{jL}(\vartheta)$ are probabilities for each $j = 1, \dots, L$ since they are all positive and $\sum_{j=1}^L \nu_{jL}(\vartheta) = 1$. First, we give a technical lemma.

Lemma 2.2.2 *For every uniformly bounded function f on Ω_{Λ_n} , the family*

$$\mathbf{F} = \left\{ \sum_{j=1}^L f \nu_{jL}(\cdot), L \in \mathbf{N} \right\} \quad (2.2.12)$$

has a subsequence which is uniformly convergent on η_θ .

This lemma may then be used to establish uniform convergence of the Monte Carlo approximations given in Lemma 2.2.1.

Lemma 2.2.3 *Each of the families $\mathbf{F}_1 = \{\ell_L(x(n), \cdot), L \in \mathbf{N}\}$, $\mathbf{F}_2 = \{\nabla \ell_L(x(n), \cdot), L \in \mathbf{N}\}$, and $\mathbf{F}_3 = \{\nabla^2 \ell_L(x(n), \cdot), L \in \mathbf{N}\}$ has a subsequence which is uniformly convergent on η_θ .*

Lemma 2.2.4 *Each of the sequences given in Lemma 2.2.1 are uniformly convergent for each $\vartheta \in \eta_\theta$, \mathbf{P} -a.s. as $L \rightarrow \infty$.*

Now, the MLE $\hat{\theta}_n$ exists uniquely in η_θ , P_θ -a.s. as $n \rightarrow \infty$. Also, $\ell_L(x(n), \vartheta)$ is globally concave since its Hessian matrix, which is given in the derivation of (A.2.22), is positive definite. In addition, $\log \mathcal{L}(x(n), \vartheta)$ is strictly concave on η_θ by Lemma 2.1.1. Hence, by Lemma 2.2.4, $\ell_L(x(n), \vartheta)$ is strictly concave on η_θ , \mathbf{P} -a.s. as $n \rightarrow \infty$. Therefore (in a manner analogous to the argument for the existence and uniqueness of the MLE in Section 2.1) the Monte Carlo likeli-

hood equation

$$\nabla \ell_L(x(n), \vartheta) = 0 \quad (2.2.13)$$

has a solution $\hat{\theta}_{nL} \in \eta_\theta$ which is the unique MCMLE.

There are some important issues to consider when using the MCMLE as an estimate of the true parameter. One is that the magnitude of the dimension n requires that L , the number of Markov chain Monte Carlo samples, must also be large in order to provide a good estimate of θ . Another is that, although ψ is arbitrary in theory, it must be chosen carefully so that the number of Markov chain Monte Carlo samples is not prohibitive. These practical considerations will be discussed in detail in Chapter IV.

2.3 Parameter Estimation using Besag's Pseudo-Likelihood

The alternative to the MLE developed by Besag (1974) is to maximize a "pseudo-likelihood" function. The pseudo-likelihood which he proposed is simply the product of the local characteristics of the sites in Λ_n :

$$\begin{aligned} \mathcal{P}\mathcal{L}(x(n), \theta) &= \prod_{i \in \Lambda_n} P_\theta(X_i = \tilde{x}_i \mid X = \tilde{x}) \\ &= \prod_{i \in \Lambda_n} \frac{\exp[-H_{\{i\}}(\tilde{x})]}{\sum_{s \in S} \exp[-H_{\{i\}}(s; \tilde{x})]} \end{aligned} \quad (2.3.1)$$

where the second expression uses the local characteristics in (1.3.12) and the associated partition function (1.3.13). Any value which maximizes $\mathcal{P}\mathcal{L}(x(n), \cdot)$ is called a maximum pseudo-likelihood estimate (MPLE) of the true parameter θ . We denote the MPLE by $\tilde{\theta}_n$.

There are several motivating factors for using the MPLE as an estimate of θ . A practical one is that these local characteristics, particularly the partition functions for single sites, are quite easily computed. Another is that, intuitively the local geometry of an image is reasonably summarized by the local characteristics; in fact, when the ancestors of the pseudo-likelihood were first proposed, the product in (2.3.1) was over a subset of the data so that the local characteristics were truly conditionally independent (Besag, 1974). Still another compelling motivation for using the MPLE is that Geman and Graffigne (1986) have established the existence, uniqueness, and consistency of the MPLE under very general conditions. In this section, we reiterate these properties of the MPLE for our own edification, and go on to establish

some other asymptotic results needed later in Chapter III.

Write the pseudo-likelihood in “exponential family” form, analogous to the likelihood function in (2.1.3):

$$\mathcal{P}\mathcal{L}(x^{(n)}, \theta) = \exp\left\{|\Lambda_n|[\theta^T V_n - g_n(\theta)]\right\}. \quad (2.3.2)$$

The forms of $V_n \triangleq V(\tilde{x}_{\Lambda_n})$ (this notation is analogous to the similar notation for Y_n given in Section 2.1) and $g_n(\theta)$ are derived below.

Equating (2.3.1) and (2.3.2), we see

$$\frac{\exp\left\{|\Lambda_n| \theta^T V_n\right\}}{\exp\left\{|\Lambda_n| g_n(\theta)\right\}} = \frac{\exp\left\{-\sum_{i \in \Lambda_n} H_{\{i\}}(\tilde{x})\right\}}{\prod_{i \in \Lambda_n} \left[\sum_{s \in S} \exp\{-H_{\{i\}}(s; \tilde{x})\}\right]}. \quad (2.3.3)$$

Define

$$|\Lambda_n| g_n(\theta) = \sum_{i \in \Lambda_n} \log \sum_{s \in S} \exp\{-H_{\{i\}}(s; \tilde{x})\} \quad (2.3.4)$$

so that

$$|\Lambda_n| \theta^T V_n = -\sum_{i \in \Lambda_n} H_{\{i\}}(\tilde{x}). \quad (2.3.5)$$

Note here that summing the single-site energy functions in (2.3.4) counts the interactions in Λ_n twice, so that we see

$$\sum_{i \in \Lambda_n} H_{\{i\}}(x) = H_{\Lambda_n}(x) + \frac{1}{2} \sum_{i, j \in \Lambda_n} \beta_{j-i} U_2(x_i, x_j). \quad (2.3.6)$$

(Refer to equations (2.1.5) and (2.1.8) of the analogous derivation in Section 2.1).

To find the components of V_n , use the energy function (1.3.11) for a single site to get

$$|\Lambda_n| \theta^T V_n = \sum_{i \in \Lambda_n} \left[h U_1(\tilde{x}_i) + \sum_{j \in \mathcal{N}_i} \beta_{j-i} U_2(\tilde{x}_i, \tilde{x}_j) \right], \quad (2.3.7)$$

so that

$$\theta^T V_n = h \frac{1}{|\Lambda_n|} \sum_{i \in \Lambda_n} U_1(\tilde{x}_i) + \frac{1}{|\Lambda_n|} \sum_{i \in \Lambda_n} \sum_{j \in \mathcal{N}_i} \beta_{j-i} U_2(\tilde{x}_i, \tilde{x}_j). \quad (2.3.8)$$

Redefine the index j in the same manner as we did to get (2.1.9) so that the above expression becomes

$$\theta^T V_n = h \frac{1}{|\Lambda_n|} \sum_{i \in \Lambda_n} U_1(\tilde{x}_i) + \sum_{j \in \mathcal{N}_o} \beta_j \frac{1}{|\Lambda_n|} \sum_{i \in \Lambda_n} U_2(\tilde{x}_i, \tilde{x}_{i+j}) \quad (2.3.9)$$

Clearly, then, the components of V_n are

$$\frac{1}{|\Lambda_n|} \sum_{i \in \Lambda_n} U_1(\tilde{x}_i) \quad (2.3.10)$$

corresponding to h in θ , and

$$\frac{1}{|\Lambda_n|} \sum_{i \in \Lambda_n} U_2(\tilde{x}_i, \tilde{x}_{i+j}) \quad (2.3.11)$$

corresponding to β_j in θ .

Now, the form of $g_n(\theta)$ is given by

$$g_n(\theta) = \frac{1}{|\Lambda_n|} \sum_{i \in \Lambda_n} \log \sum_{s \in S} \exp[-H_{\{i\}}(s; \tilde{x})]. \quad (2.3.12)$$

Since the single-site energy function is for a region with $n = 1$, we may rewrite $g_n(\theta)$ as

$$g_n(\theta) = \frac{1}{|\Lambda_n|} \sum_{i \in \Lambda_n} \log \sum_{s \in S} \exp\{\theta^T V_1^i(s)\}, \quad (2.3.13)$$

where the components of $V_1^i(\cdot)$ are the pair-potentials for the single site i , respective to the components of θ . Note that

$$V_n = \frac{1}{|\Lambda_n|} \sum_{i \in \Lambda_n} V_1^i(\tilde{x}_i). \quad (2.3.14)$$

We now try to establish the existence of the MPLE. Toward this end, the gradient of $g_n(\vartheta)$ with respect to ϑ is given by

$$\begin{aligned} \nabla g_n(\vartheta) &= \frac{1}{|\Lambda_n|} \sum_{i \in \Lambda_n} \frac{1}{\sum_{t \in S} \exp\{\vartheta^T V_1^i(t)\}} \sum_{s \in S} V_1^i(s) \exp\{\vartheta^T V_1^i(s)\} \\ &= \frac{1}{|\Lambda_n|} \sum_{i \in \Lambda_n} \sum_{s \in S} V_1^i(s) \frac{\exp\{\vartheta^T V_1^i(s)\}}{\sum_{t \in S} \exp\{\vartheta^T V_1^i(t)\}} \end{aligned}$$

$$= \frac{1}{|\Lambda_n|} \sum_{i \in \Lambda_n} E_{\vartheta}(V_1^i | \tilde{x}_{\mathcal{N}_i}) \quad (2.3.15)$$

so that the corresponding “pseudo-likelihood” equation analogous to the likelihood equation (2.1.15) is

$$V_n = \nabla g_n(\vartheta). \quad (2.3.16)$$

Now, for all $\vartheta \in \mathbf{R}^k$, using the expressions in (2.3.14) and (2.3.15) gives

$$\begin{aligned} E_{\vartheta}[V_n - \nabla g_n(\vartheta)] &= E_{\vartheta}V_n - E_{\vartheta}\left[\frac{1}{|\Lambda_n|} \sum_{i \in \Lambda_n} E_{\vartheta}(V_1^i | \tilde{x}_{\mathcal{N}_i})\right] \\ &= E_{\vartheta}V_n - \frac{1}{|\Lambda_n|} \sum_{i \in \Lambda_n} E_{\vartheta}\left[E_{\vartheta}(V_1^i | \tilde{x}_{\mathcal{N}_i})\right] \\ &= E_{\vartheta}V_n - \frac{1}{|\Lambda_n|} \sum_{i \in \Lambda_n} E_{\vartheta}V_1^i \\ &= E_{\vartheta}V_n - E_{\vartheta}\left[\frac{1}{|\Lambda_n|} \sum_{i \in \Lambda_n} V_1^i\right] \\ &= 0. \end{aligned} \quad (2.3.17)$$

Thus, by the ergodic theorem given by Georgii (1988, page 306, Theorem (14.A8)), we have

$$\lim_{n \rightarrow \infty} [V_n - \nabla g_n(\theta)] = 0, \quad P_{\sigma}\text{-a.s.} \quad (2.3.18)$$

Geman and Graffigne (1986) establish the existence of $\tilde{\theta}_n$. However, the existence may also be shown, using Lemma 2.3.3 below, in a way similar to the one by which the existence of the MLE was shown in Section 2.1.

Now, we must look into uniqueness properties for $\tilde{\theta}_n$. For each $i \in \Lambda_n$, let $\Lambda_{i,R}$ be the $(2R+1) \times (2R+1)$ square lattice centered at i (recall that R is the range of the Gibbs distribution). In particular, let $\Lambda_{o,R} = \Lambda_{2R+1}$. Let $\varsigma \in S$ and $\eta \in \Omega_{\Lambda_{o,R} \setminus \{o\}}$, so that the combined configuration is $\varsigma \oplus \eta \in \Omega_{\Lambda_{2R+1}}$. Let \tilde{X} denote the periodic random field based on the random field X_{Λ_n} . Define

$$\begin{cases} \mathbf{1}_i(\varsigma \oplus \eta) = \mathbf{1}_{\{\tilde{X}_{\Lambda_{i,R}} = \varsigma \oplus \eta\}} \\ \mathbf{1}_i(\eta) = \mathbf{1}_{\{\tilde{X}_{\Lambda_{i,R} \setminus \{i\}} = \eta\}} \end{cases} \quad \text{for } i \in \Lambda_n \quad (2.3.19)$$

and

$$\begin{cases} N_n(\varsigma \oplus \eta) = \sum_{i \in \Lambda_n} \mathbb{1}_i(\varsigma \oplus \eta) \\ N_n(\eta) = \sum_{i \in \Lambda_n} \mathbb{1}_i(\eta) \end{cases} \quad (2.3.20)$$

to aid in denoting empirical probabilities for observing the respective configurations.

Lemma 2.3.1 *There exist positive constants λ , C , and a such that*

$$P_\theta \left(\frac{N_n(\varsigma \oplus \eta)}{|\Lambda_n|} < \lambda \right) \leq C \exp(-an) \quad (2.3.21)$$

uniformly for all large n and all $\varsigma \oplus \eta \in \Omega_{\Lambda_{2R+1}}$.

Define the event

$$\mathfrak{S}_n = \left\{ x(n) \in \Omega_{\Lambda_n} : \frac{N_n(\varsigma \oplus \eta)}{|\Lambda_n|} \geq \lambda \quad \forall \varsigma \oplus \eta \in \Omega_{\Lambda_{2R+1}} \right\}, \quad (2.3.22)$$

on which the empirical probabilities for all $(2R+1) \times (2R+1)$ configurations are bounded away from 0. The proofs of the following lemmas concentrate on \mathfrak{S}_n since its complement is asymptotically negligible according to Lemma 2.3.1.

Lemma 2.3.2 *Under the Identifiability Condition 2.1.1 and on the set \mathfrak{S}_n , there exist $c, C > 0$ such that*

$$c \leq v^\top \nabla^2 g_n(\vartheta) v \leq C \quad (2.3.23)$$

uniformly for all unit vectors $v \in \mathbb{R}^k$, all ϑ in a neighborhood of θ , and all large n .

Thus we have strict convexity of $g_n(\cdot)$ in a neighborhood of θ , implying uniqueness of the MP-LE. The next lemma, due to Com ets (1992), gives an exponential consistency rate for the MP-LE.

Lemma 2.3.3 *Under the Identifiability Condition 2.1.1, for every $\epsilon > 0$ there exist $c, C > 0$ such that*

$$P_\theta \left(\|\hat{\theta}_n - \theta\| > \epsilon \right) \leq C \exp(-c|\Lambda_n|) \quad (2.3.24)$$

uniformly for all large n , where $\|\cdot\|$ is the Euclidian norm.

These next two lemmas provide restricted mean square errors and moderate deviation probabilities for the MPLE.

Lemma 2.3.4 *Under the Identifiability Condition 2.1.1,*

$$E_{\theta} \left\{ \|\tilde{\theta}_n - \theta\|^2 \mathbf{1}_{\mathcal{E}_n} \right\} = O\left(\frac{1}{|\Lambda_n|}\right) \quad (2.3.25)$$

as $n \rightarrow \infty$.

Lemma 2.3.5 *Under the Identifiability Condition 2.1.1, for every $\varepsilon > 0$ there exists $\alpha > 0$ such that*

$$P_{\theta} \left(|\Lambda_n| \|\tilde{\theta}_n - \theta\|^2 > \varepsilon \log n \right) = O\left(\frac{1}{n^{\alpha}}\right) \quad (2.3.26)$$

as $n \rightarrow \infty$.

It seems clear that using the pseudo-likelihood function allows the exploitation of the mathematical advantages of independence. However, the collection of local characteristics does contain a great deal of information about the behavior of the distribution. Thus the product of these local characteristics may be viewed as a sufficient statistic in the sense that it seems to make the best use of the information available in the sample. In addition, the MPLE has some practical advantages over the MCMLE – mainly, the conditions for the practicality of the MPLE are much weaker than those of the MCMLE. These practical advantages, as well as numerical comparisons of the two estimators, will be discussed in Chapter IV.

III. Model Selection Criteria for Gibbs-Markov Random Fields

3.1 Perspective on Relevant Model Selection Criteria

Given a sample from a particular statistical model, one would ideally like to be able to determine the original model. Of course, such a determination is not always possible in this simplistic sense, though selection of the underlying model is possible under certain conditions. Two examples of such a process for selection which motivate our studies are: for time series models, several methods for selecting the order of an autoregression; and for exponential families, procedures for determining the dimension of the parameter space.

Very little has been done to address model selection for random fields. Kashyap and Chellappa (1983) have addressed selection of neighborhoods for random fields in which a gray level at a particular site is a linear combination of neighboring gray levels plus Gaussian noise. More recently, Smith and Miller (1990) have proposed a model selection criterion for MRFs based on the stochastic complexity of Rissanen (1984). Their criterion turns out to be quite similar to the criterion presented in Section 3.5, but they did not discuss any of its properties.

The task at hand is to choose a model for an observed texture from a collection of GMRF models. Since a GMRF is specified by its energy function, the model selection problem is then one of selecting an energy function based on a sample texture. The energy functions which we consider are induced by the finite-range, translation invariant pair-potential \mathbf{U} given in (1.3.9), for which we assume that the deterministic pair-interactions U_1 and U_2 are known and that the parameter $\theta = (h, \beta^T)^T$ is unknown - see Chapter I for details. Thus the model selection problem for GMRFs is one of concurrently selecting the dimension of θ and the neighborhood system \mathfrak{N} .

It is important to point out that the choice of functions which make up the potential is still largely *ad hoc*. Modestino and Zhang (1992) have proposed a method for the design of these functions, but the problem still needs a great deal of attention.

The model selection problem for GMRFs is related to model selection for both time series and exponential families. It is kin to model selection for times series because the chosen model should generate spatially dependent data, like an autoregressive and/or moving average model generates temporally dependent data. It is related to model selection for exponential

families because of that form for the GMRF developed in Chapter II. There are then several model selection criteria in the statistical literature which are relevant to the GMRF model selection problem.

Among the first of the model selection criteria for autoregressive (AR) time series is the final prediction error (FPE) of Akaike (1969). The FPE is an estimate of the one-step prediction mean squared error for a realization of the candidate AR process (not the observation itself). The procedure, which does tend to over-parametrize, is to choose the order p of the AR process which minimizes the FPE. Another criterion for the estimation of the order of an AR model is the criterion AR transfer (CAT) function, due to Parzen (1974). Neither the FPE nor the CAT seem to generalize to our context. Yet another criterion from time series is that of Hannan and Quinn (1979). This criterion provided a strongly consistent procedure to estimate the order of an AR model based on the law of the iterated logarithm for partial autocorrelations. Hannan (1980) gave a similar criterion for choosing an autoregressive moving average model. It is not clear, however, that such criteria can be extended to GMRFs, especially under phase transitions (long-range dependence).

A more popular criterion, which is useful not only for time series but also for exponential families, is the information criterion of Akaike (1973, 1974), which is called AIC. The associated procedure says to choose the model which maximizes a quantity of the general form

$$\text{AIC} = \log(\text{maximum likelihood}) - C \times (\text{number of free parameters}), \quad (3.1.1)$$

where $C > 0$. Intuitively, this criterion first fits a candidate model, and then subtracts a penalty for over-parametrization. The AIC, however, tends to over-parametrize, and it does not give a consistent estimate of the model (Woodroffe, 1982) when the dimension of the true parameter is fixed and finite. Shibata (1980) has shown that the AIC is asymptotically optimal for modelling linear stationary processes of an order which varies along with the sample size.

Bayesian modifications of Akaike's information criterion, called BIC, have proven to be successful in estimating a true model of fixed and finite order. The form of the BIC is similar to that of the AIC: the BIC procedure chooses the model which maximizes a quantity of the general form

$$\begin{aligned} \text{BIC} = \log(\text{maximum likelihood}) & \\ - C \times (\text{number of free parameters}) \times \log(\text{sample size}), & \end{aligned} \quad (3.1.2)$$

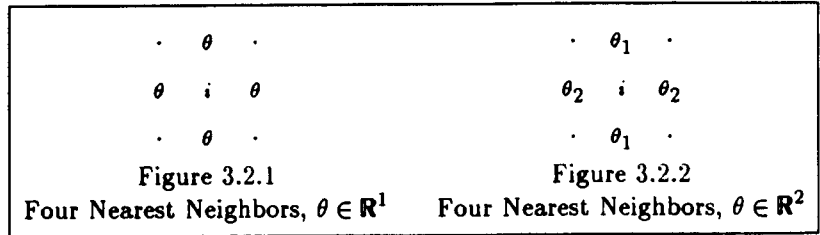
for some constant $C > 0$. Akaike (1978) developed a BIC in which he imposed a non-infor-

mative prior on the parameter space. Concurrently, Schwarz (1978) derived a BIC specifically for selecting the dimension of the parameter in an exponential family, using a very general prior on the parameter space. With a larger penalty term, the BIC is not as prone to over-parametrization as the AIC. In fact, Woodroffe (1982) has shown that, for i.i.d. data, Schwarz' BIC is consistent. Haughton (1988) has also proven consistency for an extension of Schwarz' BIC (with more terms in the criterion itself) for i.i.d. data. In view of these consistency results, we pursue Schwarz' ideas within our own context.

Clearly in this situation, there is an intimate connection between parameter estimation and model selection since the selection criteria are dependent upon a likelihood function evaluated at its maximum. Indeed, as the parameter estimate becomes more accurate, the model selection becomes better because the fit for the true model is more accurate. In addition, consistency of a model selection procedure requires not only the consistency of the employed parameter estimate, but also appropriate consistency rates. Parameter estimation and model selection are also fundamentally different: parameter estimation is an inference problem involving one model, whereas model selection is a multiple decision problem involving several models.

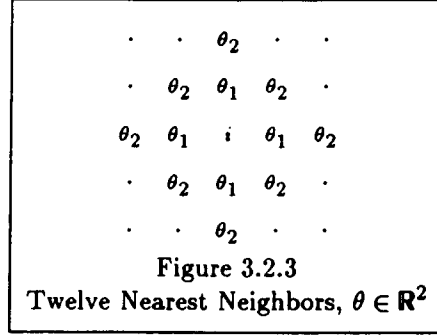
3.2 Framework for Gibbs-Markov Model Selection

As stated in the previous section, the development of a GMRF model selection procedure follows Schwarz' approach for exponential families. Recall that we consider a finite collection of possible energy functions, all with the same pair-interactions. Assume that only one of these energy functions in the collection induces the GMRF P_θ which generates $X(n)$. The goal then is to discover the true energy function.



Specification of an energy function in the current context consists of two inter-connected parts: determination of both the dimension k of the parameter θ as well as the neighborhood system \mathfrak{N} . Denote the set of all candidate models by $\mathcal{M} = \{0, 1, \dots, M\}$, and let \mathfrak{N}_m be the neighborhood system for the model $m \in \mathcal{M}$. Notice that two models $m_1 \neq m_2$ are distinct if their parameter spaces have different dimensions, as shown in Figure 3.2.1 and Figure 3.2.2; or they are associated with different neighborhood systems, as in Figure 3.2.2 and Figure 3.2.3; or per-

haps even both, illustrated by Figure 3.2.1 and Figure 3.2.3.



Let $\Theta = \mathbb{R}^K$ be the parameter space of interest, which is decomposed as a disjoint union of several subspaces:

$$\Theta = \bigcup_{m=0}^M \Theta_m, \quad \Theta_{m_1} \cap \Theta_{m_2} = \emptyset \quad \forall m_1 \neq m_2, \quad (3.2.1)$$

where each Θ_m is the parameter space corresponding to the model $m \in \mathcal{M}$. Assume that the closure $\bar{\Theta}_m$ is a k_m -dimensional linear subspace of \mathbb{R}^K for all $m \in \mathcal{M}$. In particular, Θ_0 corresponds to the fully specified model with no unknown parameter, while Θ_M corresponds to the “largest” model (*i.e.*, the model which may be reduced to any of the others).

To characterize the original belief on the candidate models, define a prior π on Θ as a mixture of mutually singular probability measures:

$$\pi = \sum_{m=0}^M \alpha_m \pi_m, \quad (3.2.2)$$

where $\alpha_m > 0$, $m \in \mathcal{M}$, are constants such that $\sum_{m=0}^M \alpha_m = 1$, which reflect any prior notions about the true model; and π_m is a probability measure supported on the closure $\bar{\Theta}_m$ with a smooth density $\mu_m > 0$, $m \in \mathcal{M}$. This is analogous to the prior which Schwarz (1978) imposed to derive his BIC.

The posterior distribution on Θ given the observation $x(n)$ is then given by

$$\Pi_{x(n)}(A) = \frac{\int_A \mathcal{L}(x(n), \theta) d\pi(\theta)}{\int_{\Theta} \mathcal{L}(x(n), \theta) d\pi(\theta)} \quad (3.2.3)$$

for all measurable $A \subseteq \Theta$, where $\mathcal{L}(x(n), \theta)$ is the likelihood (2.1.2). Let $\varrho: \Omega_{\Lambda_n} \rightarrow \mathcal{M}$ be the decision function such that the selection of a model $m \in \mathcal{M}$ based on the observation $x(n)$ is de-

noted by $\varrho(x(n)) = m$. Impose 0-1 loss so that the posterior Bayes risk for the decision $\varrho(\cdot)$ based on $x(n)$ is

$$\begin{aligned} \mathfrak{R}(x(n), \varrho) &= \int_{\Theta} \mathbf{1}_{\{\theta \notin \Theta_{\varrho(x(n))}\}} d\Pi_{x(n)}(\theta) \\ &= 1 - \Pi_{x(n)}(\Theta_{\varrho(x(n))}). \end{aligned} \quad (3.2.4)$$

Hence the Bayesian solution to the model selection problem - i.e., that solution which minimizes the risk in (3.2.4) - is to choose a model \hat{m} which maximizes the posterior probability (3.2.3) evaluated at the appropriate linear subspace: $\Pi_{x(n)}(\Theta_{\hat{m}}) \geq \Pi_{x(n)}(\Theta_m)$ for all $m \in \mathcal{M}$.

3.3 Model Selection Based on the MLE

Because it is very difficult to work with the posterior distribution, we must find an approximation to the Bayesian solution to the model selection problem. For each $m \in \mathcal{M}$, let $\hat{\theta}_n^m$ denote a MLE restricted to $\bar{\Theta}_m$, and let \mathcal{L}_m denote the minimal form of the exponential family given by

$$\mathcal{L}_m(x(n), \theta) = \exp\left\{ \Lambda_n \left[\theta^T Y_n^m - b_n^m(\theta) \right] \right\}, \quad (3.3.1)$$

where Y_n^m and $b_n^m(\cdot)$ are the sufficient statistic and cumulant generating function, respectively, associated with the model m . Define the information criterion

$$Q_m^{(1)} = \sup_{\vartheta \in \bar{\Theta}_m} \log \mathcal{L}_m(x(n), \vartheta) - \frac{k_m}{2} \log |\Lambda_n|, \quad (3.3.2)$$

which is derived in the proof of Theorem 3.3.1 below.

Procedure 1 Choose the model $\hat{m}_1 \in \mathcal{M}$ which maximizes the criterion $Q_m^{(1)}$.

The decision function associated with Procedure 1 is given by $\varrho_1(x(n)) = \hat{m}_1$. The following result establishes that the criterion (3.3.2) is indeed a BIC.

Theorem 3.3.1 Under the Uniqueness Condition 1.3.1, the Identifiability Condition 2.1.1, the Positivity Condition 2.1.2, and with P_θ -probability one, Procedure 1 chooses the same model that the Bayesian procedure chooses as $n \rightarrow \infty$.

Proof: The primary concern is with showing that the model \hat{m}_1 is, asymptotically, equal to the model which maximizes the posterior probability $\Pi_{x(n)}(\Theta_m)$. Write the collection of candidate models as

$$\mathcal{M} = \mathcal{M}_1(\varpi) \cup \{\varpi\} \cup \mathcal{M}_2(\varpi) \quad (3.3.3)$$

where $\varpi \in \mathcal{M}$ is the true model, $\theta \in \Theta_\varpi$ is the true parameter, $\mathcal{M}_1(\varpi) = \{m \in \mathcal{M} : \theta \notin \bar{\Theta}_m\}$, and $\mathcal{M}_2(\varpi) = \{m \in \mathcal{M} : \bar{\Theta}_\varpi \subset \bar{\Theta}_m\}$. Notice that $\mathcal{M}_1(\varpi)$ corresponds to an under-parametrized choice of model or incorrect specification of neighborhood system (different neighborhoods will correspond to different subspaces which may have the same dimension), while $\mathcal{M}_2(\varpi)$ corresponds to an over-parametrized choice. Note particularly that $\bar{\Theta}_\varpi$ is a *proper* subset of the spaces corresponding to models in $\mathcal{M}_2(\varpi)$.

Also, let M denote the “largest” model (*i.e.*, the model which can be reduced to any of the other candidate models), so that $\bar{\Theta}_m \subseteq \bar{\Theta}_M$ for all $m \in \mathcal{M}$. Note here that $\bar{\Theta}_M = \Theta = \mathbf{R}^K$, and that we may write $\hat{\theta}_n^M = \hat{\theta}_n$ since $\hat{\theta}_n^M$ is a “global” MLE over the set \mathcal{M} .

This proof will proceed in three steps. In Step 1, for a model $m \in \mathcal{M}_2(\varpi)$ where the MLE exists and converges P_θ -a.s. to the true parameter θ , we shall show that the posterior probability $\Pi_{x(n)}(\Theta_m)$ is asymptotically equivalent to $Q_m^{(1)}$ as $n \rightarrow \infty$. In Step 2, we will show that the Bayesian procedure will not choose any model $m \in \mathcal{M}_1(\varpi)$ as $n \rightarrow \infty$. Finally, in Step 3, we will show that Procedure 1 will not choose any model $m \in \mathcal{M}_1(\varpi)$ as $n \rightarrow \infty$.

Step 1: We seek to choose the model $m \in \mathcal{M}_2(\varpi)$ which maximizes $\Pi_{x(n)}(\Theta_m)$; equivalently, we may maximize the logarithm of the posterior probability (3.2.3), given by

$$\log \Pi_{x(n)}(\Theta_m) = \log \int_{\Theta_m} \mathcal{L}(x(n), \vartheta) d\pi(\vartheta) - \log \int_{\Theta} \mathcal{L}(x(n), \vartheta) d\pi(\vartheta). \quad (3.3.4)$$

Obviously, the second term is not a function of m , and therefore does not affect the maximization problem in m .

Concentrating on the first term, write the likelihood as a function of m via reduction of the exponential family to its minimal form: $\mathcal{L}(\cdot, \vartheta) = \mathcal{L}_m(\cdot, \vartheta)$ for $\vartheta \in \Theta_m$. Thus, since $\pi_{m'}(\Theta_m) = 0$ for $m' \neq m$ (because the probability measures π_m , $m \in \mathcal{M}$, are mutually singular), it suffices to show that the asymptotic expansion

$$\log \int_{\Theta_m} \alpha_m \mathcal{L}_m(x(n), \vartheta) d\pi_m(\vartheta) = Q_m^{(1)} + O(1) \quad (3.3.5)$$

holds uniformly on $\{x(n) : \|\hat{\theta}_n^m - \theta\| < \delta\}$ for some $\delta > 0$. This will imply that asymptotically,

Procedure 1 and the Bayesian procedure choose the same model in $\mathcal{M}_2(\varpi)$.

Choose $\delta > 0$ so that Lemma 2.1.1 (which guarantees the uniqueness of the MLE) holds for all ϑ with $\|\vartheta - \theta\| < 3\delta$. Next, on the left-hand side of (3.3.5), factor out the maximum likelihood and split the remaining integral:

$$\begin{aligned} \log \int_{\Theta_m} \alpha_m \mathcal{L}_m(x(n), \vartheta) d\pi_m(\vartheta) &= \log \mathcal{L}_m(x(n), \widehat{\theta}_n^m) + \log \int_{\Theta_m} \alpha_m \frac{\mathcal{L}_m(x(n), \vartheta)}{\mathcal{L}_m(x(n), \widehat{\theta}_n^m)} d\pi_m(\vartheta) \\ &= \log \mathcal{L}_m(x(n), \widehat{\theta}_n^m) \\ &+ \log \left\{ \int_{\|\vartheta - \widehat{\theta}_n^m\| \leq \delta} \alpha_m \frac{\mathcal{L}_m(x(n), \vartheta)}{\mathcal{L}_m(x(n), \widehat{\theta}_n^m)} d\pi_m(\vartheta) + \int_{\|\vartheta - \widehat{\theta}_n^m\| > \delta} \alpha_m \frac{\mathcal{L}_m(x(n), \vartheta)}{\mathcal{L}_m(x(n), \widehat{\theta}_n^m)} d\pi_m(\vartheta) \right\} \end{aligned} \quad (3.3.6)$$

Notice that, using the minimal exponential family version of the likelihood (2.1.3), we may write

$$\frac{\mathcal{L}_m(x(n), \vartheta)}{\mathcal{L}_m(x(n), \widehat{\theta}_n^m)} = \exp \left\{ \left[\Lambda_n \left(\vartheta^T Y_n^m - b_n^m(\vartheta) - \left[(\widehat{\theta}_n^m)^T Y_n^m - b_n^m(\widehat{\theta}_n^m) \right] \right) \right] \right\}. \quad (3.3.7)$$

Now, $\vartheta^T Y_n^m - b_n^m(\vartheta)$ is concave in ϑ for all $\vartheta \in \bar{\Theta}_m$; by Lemma 2.1.1, it is strictly concave in a small neighborhood of θ for $x(n) \in \mathfrak{S}_n$. By Taylor expansion about the MLE, we have

$$\begin{aligned} \vartheta^T Y_n^m - b_n^m(\vartheta) - \left[(\widehat{\theta}_n^m)^T Y_n^m - b_n^m(\widehat{\theta}_n^m) \right] &= \\ &(\vartheta - \widehat{\theta}_n^m)^T \left[Y_n^m - \nabla b_n^m(\widehat{\theta}_n^m) \right] - \frac{1}{2} (\vartheta - \widehat{\theta}_n^m)^T \nabla^2 b_n^m(\vartheta') (\vartheta - \widehat{\theta}_n^m) \end{aligned} \quad (3.3.8)$$

for some $\vartheta' \in \bar{\Theta}_m$ satisfying $\|\vartheta' - \widehat{\theta}_n^m\| \leq \|\vartheta - \widehat{\theta}_n^m\|$. Because $\widehat{\theta}_n^m$ is an MLE, the first term on the right-hand side of (3.3.8) is zero. Then

$$\frac{\vartheta^T Y_n^m - b_n^m(\vartheta) - \left[(\widehat{\theta}_n^m)^T Y_n^m - b_n^m(\widehat{\theta}_n^m) \right]}{\|\vartheta - \widehat{\theta}_n^m\|^2} = \frac{-\frac{1}{2} (\vartheta - \widehat{\theta}_n^m)^T \nabla^2 b_n^m(\vartheta') (\vartheta - \widehat{\theta}_n^m)}{\|\vartheta - \widehat{\theta}_n^m\|^2}, \quad (3.3.9)$$

so that, by Lemma 2.1.1 and for $\|\vartheta - \widehat{\theta}_n^m\| \leq \delta$, we have

$$-\frac{C}{2} \leq \frac{\vartheta^T Y_n^m - b_n^m(\vartheta) - \left[(\widehat{\theta}_n^m)^T Y_n^m - b_n^m(\widehat{\theta}_n^m) \right]}{\|\vartheta - \widehat{\theta}_n^m\|^2} \leq -\frac{c}{2}. \quad (3.3.10)$$

In particular, for $\|\vartheta - \hat{\theta}_n^m\| = \delta$, we have

$$\vartheta^T Y_n^m - b_n^m(\vartheta) - [(\hat{\theta}_n^m)^T Y_n^m - b_n^m(\hat{\theta}_n^m)] \leq -\frac{c}{2} \delta^2 \quad (3.3.11)$$

Then by concavity of $\vartheta^T Y_n^m - b_n^m(\vartheta)$, for $\|\vartheta - \hat{\theta}_n^m\| > \delta$ we have

$$\vartheta^T Y_n^m - b_n^m(\vartheta) - [(\hat{\theta}_n^m)^T Y_n^m - b_n^m(\hat{\theta}_n^m)] \leq -\varepsilon \quad (3.3.12)$$

where $\varepsilon = \frac{c\delta^2}{2} > 0$. Hence, using (3.3.7) along with the fact that π_m is a probability measure, the last integral term in (3.3.6) is

$$\begin{aligned} \int_{\|\vartheta - \hat{\theta}_n^m\| > \delta} \alpha_m \frac{\mathcal{L}_m(x(n), \vartheta)}{\mathcal{L}_m(x(n), \hat{\theta}_n^m)} d\pi_m(\vartheta) &\leq \int_{\|\vartheta - \hat{\theta}_n^m\| > \delta} \alpha_m \exp\{-\varepsilon |\Lambda_n|\} d\pi_m(\vartheta) \\ &\leq \alpha_m \exp\{-\varepsilon |\Lambda_n|\} \pi_m\left(\left\{\vartheta \in \bar{\Theta}_m: \|\vartheta - \hat{\theta}_n^m\| \geq \delta\right\}\right) \\ &\leq \alpha_m \exp\{-\varepsilon |\Lambda_n|\} \end{aligned} \quad (3.3.13)$$

We must now investigate logarithm of the first integral term in (3.3.6). Recall that π_m has a density $\mu_m > 0$. Hence, by (3.3.7) and (3.3.10),

$$\begin{aligned} \log \int_{\|\vartheta - \hat{\theta}_n^m\| \leq \delta} \alpha_m \frac{\mathcal{L}_m(x(n), \vartheta)}{\mathcal{L}_m(x(n), \hat{\theta}_n^m)} d\pi_m(\vartheta) \\ = \log \int_{\|\vartheta - \hat{\theta}_n^m\| \leq \delta} \alpha_m \exp\left\{|\Lambda_n| \left(\vartheta^T Y_n^m - b_n^m(\vartheta) - [(\hat{\theta}_n^m)^T Y_n^m - b_n^m(\hat{\theta}_n^m)]\right)\right\} d\pi_m(\vartheta) \\ \leq \log \int_{\|\vartheta - \hat{\theta}_n^m\| \leq \delta} \alpha_m \exp\left\{-\frac{c}{2} |\Lambda_n| \|\vartheta - \hat{\theta}_n^m\|^2\right\} \mu_m(\vartheta) d\vartheta \end{aligned} \quad (3.3.14)$$

Integrate by substitution using $u = \sqrt{c|\Lambda_n|}(\vartheta - \hat{\theta}_n^m)$, which implies $du = (c|\Lambda_n|)^{\frac{k_m}{2}} d\vartheta$, and continue:

$$= \log \int_{\|u\| \leq \delta \sqrt{c|\Lambda_n|}} \alpha_m \exp\left\{-\frac{1}{2} \|u\|^2\right\} \mu_m\left(\hat{\theta}_n^m + \frac{u}{\sqrt{c|\Lambda_n|}}\right) (c|\Lambda_n|)^{-\frac{k_m}{2}} du$$

$$= -\frac{k_m}{2} \log |\Lambda_n| \quad (3.3.15)$$

$$+ \log \left[c' \int_{\|u\| \leq \delta \sqrt{c|\Lambda_n|}} \alpha_m \exp \left\{ -\frac{1}{2} \|u\|^2 \right\} \mu_m \left(\hat{\theta}_n^m + \frac{u}{\sqrt{c|\Lambda_n|}} \right) du \right]$$

for some $c' > 0$. Now, as $n \rightarrow \infty$,

$$\mu_m \left(\hat{\theta}_n^m + \frac{u}{\sqrt{c|\Lambda_n|}} \right) \rightarrow \mu_m(\theta) > 0. \quad (3.3.16)$$

In addition,

$$\int_{\|u\| \leq \delta \sqrt{c|\Lambda_n|}} \exp \left\{ -\frac{1}{2} \|u\|^2 \right\} du \rightarrow (2\pi)^{\frac{k_m}{2}} \quad (3.3.17)$$

as $n \rightarrow \infty$, which is simply a property of the multivariate standard normal density. Therefore

$$\log \int_{\|\vartheta - \hat{\theta}_n^m\| \leq \delta} \alpha_m \frac{\mathcal{L}_m(x(n), \vartheta)}{\mathcal{L}_m(x(n), \hat{\theta}_n^m)} d\pi_m(\vartheta) \leq -\frac{k_m}{2} \log |\Lambda_n| + O(1). \quad (3.3.18)$$

By the same token,

$$\begin{aligned} \log \int_{\|\vartheta - \hat{\theta}_n^m\| \leq \delta} \alpha_m \frac{\mathcal{L}_m(x(n), \vartheta)}{\mathcal{L}_m(x(n), \hat{\theta}_n^m)} d\pi_m(\vartheta) \\ \geq \int_{\|\vartheta - \hat{\theta}_n^m\| \leq \delta} \alpha_m \exp \left\{ -\frac{C}{2} |\Lambda_n| \|\vartheta - \hat{\theta}_n^m\|^2 \right\} \mu_m(\vartheta) d\vartheta \\ \geq -\frac{k_m}{2} \log |\Lambda_n| + O(1). \end{aligned} \quad (3.3.19)$$

Thus (3.3.5) follows from (3.3.6), (3.3.13), (3.3.18), and (3.3.19).

Step 2: We now show that as $n \rightarrow \infty$, the Bayesian procedure will not choose any model $m \in \mathcal{M}_1(\varpi)$. Since $\theta \notin \bar{\Theta}_m$, we have $\|\vartheta - \theta\| \geq 3\delta$ for all $\vartheta \in \bar{\Theta}_m$ and some $\delta > 0$. Note that with large probability and for large n , $\|\hat{\theta}_n - \theta\| < \delta$ so that $\|\vartheta - \hat{\theta}_n\| > \delta$ for all $\vartheta \in \bar{\Theta}_m$ (recall that $\hat{\theta}_n$ denotes the MLE found using the largest model $\mathcal{L}(x(n), \cdot)$ on Θ).

Now, for $m \in \mathcal{M}_1(\varpi)$,

$$\sup_{\vartheta \in \bar{\Theta}_m} \log \mathcal{L}_m(x(n), \vartheta) = \sup_{\vartheta \in \bar{\Theta}_m} \log \mathcal{L}(x(n), \vartheta)$$

$$= |\Lambda_n| \sup_{\vartheta \in \Theta_m} [\vartheta^T Y_n - b_n(\vartheta)]. \quad (3.3.20)$$

From the same argument made in deriving (3.3.12), it follows that

$$\sup_{\vartheta \in \Theta_m} [\vartheta^T Y_n - b_n(\vartheta)] \leq [\widehat{\vartheta}_n^T Y_n - b_n(\widehat{\vartheta}_n)] - \varepsilon \quad (3.3.21)$$

for some $\varepsilon > 0$. Therefore

$$\begin{aligned} \int_{\Theta_m} \mathcal{L}(x(n), \vartheta) d\pi(\vartheta) &= \int_{\Theta_m} \alpha_m \exp\{|\Lambda_n| [\vartheta^T Y_n - b_n(\vartheta)]\} d\pi_m(\vartheta) \\ &= \int_{\Theta_m} \alpha_m \exp\{|\Lambda_n| [\widehat{\vartheta}_n^T Y_n - b_n(\widehat{\vartheta}_n)]\} d\pi_m(\vartheta) \\ &\leq \int_{\Theta_m} \alpha_m \exp\{|\Lambda_n| [\widehat{\vartheta}_n^T Y_n - b_n(\widehat{\vartheta}_n) - \varepsilon]\} d\pi_m(\vartheta) \\ &\leq \alpha_m \mathcal{L}(x(n), \widehat{\vartheta}_n) \exp\{-\varepsilon |\Lambda_n|\} \end{aligned} \quad (3.3.22)$$

By (3.3.5),

$$\begin{aligned} \log \int_{\Theta_M} \mathcal{L}(x(n), \vartheta) d\pi(\vartheta) &= |\Lambda_n| [\widehat{\vartheta}_n^T Y_n - b_n(\widehat{\vartheta}_n)] - \frac{k_M}{2} \log |\Lambda_n| + O(1) \\ &= \log \mathcal{L}(x(n), \widehat{\vartheta}_n) - \frac{k_M}{2} \log |\Lambda_n| + O(1) \end{aligned} \quad (3.3.23)$$

Hence

$$\begin{aligned} \log \int_{\Theta_m} \mathcal{L}(x(n), \vartheta) d\pi(\vartheta) &\leq \log \int_{\Theta_M} \mathcal{L}(x(n), \vartheta) d\pi(\vartheta) - \left(\varepsilon |\Lambda_n| - \frac{k_M}{2} \log |\Lambda_n| \right) + O(1) \\ &< \log \int_{\Theta_M} \mathcal{L}(x(n), \vartheta) d\pi(\vartheta) \end{aligned} \quad (3.3.24)$$

for large n .

Step 3: Finally, we show that as $n \rightarrow \infty$, Procedure 1 will not choose any model $m \in \mathcal{A}_1(\varpi)$. Using the index (3.3.2) for m and the largest model M , it follows from (3.3.12) that

$$\begin{aligned}
Q_m^{(1)} &= \sup_{\vartheta \in \bar{\Theta}_m} \log \ell_m(x(n), \vartheta) - \frac{k_m}{2} \log |\Lambda_n| \\
&\leq |\Lambda_n| \left[(\hat{\theta}_n)^\top Y_n - b_n(\hat{\theta}_n) - \varepsilon \right] - \frac{k_m}{2} \log |\Lambda_n| \\
&= Q_M^{(1)} - \left[\varepsilon |\Lambda_n| - \left(\frac{k_M}{2} - \frac{k_m}{2} \right) \log |\Lambda_n| \right] \\
&< Q_M^{(1)}
\end{aligned} \tag{3.3.25}$$

for large n . Hence, Procedure 1 will not choose a model $m \in \mathcal{M}_1(\varpi)$ as $n \rightarrow \infty$. \square

Still, Procedure 1 is impractical because the MLE is impossible to calculate. However, the work in this section provides a firm theoretical base from which we may construct similar procedures based on both the MCMLE and the MPLE.

3.4 Model Selection Based on the MCMLE

For each $m \in \mathcal{M}$, let $\hat{\theta}_{nL}^m$ be the MCMLE restricted to $\bar{\Theta}_m$, and based on both $x(n)$ and a sample $X^{(1)}(n), \dots, X^{(L)}(n)$ from an ergodic Markov chain whose equilibrium distribution is $P_\psi(\cdot | \tilde{x}_{\partial\Lambda_n})$ for known $\psi \in \bar{\Theta}_m$. Let $\ell_L^m(\cdot, \cdot)$ denote the Monte Carlo approximant in (2.2.7) and its dependence upon the model $m \in \mathcal{M}$. Define the information criterion

$$Q_m^{(2)} = \sup_{\vartheta \in \bar{\Theta}_m} \ell_L^m(x(n), \vartheta) - \frac{k_m}{2} \log |\Lambda_n|. \tag{3.4.1}$$

Procedure 2 Choose the model $\hat{m}_2 \in \mathcal{M}$ of largest $Q_m^{(2)}$.

The decision function associated with Procedure 2 is $\varrho_2(x(n)) = \hat{m}_2$.

Let \mathfrak{P} denote the joint probability measure of P_θ , the GMRF, and \mathbf{P} , the probability measure for the entire Markov chain $\{X^{(l)}(n)\}_{l=1}^\infty$. Note, however, that \mathfrak{P} is not the product of P_θ and \mathbf{P} . The following result establishes that the criterion (3.4.1) is asymptotically a BIC.

Theorem 3.4.1 Under the Uniqueness Condition 1.3.1, the Identifiability Condition 2.1.1, and the Positivity Condition 2.1.2, there exists a sequence L_n going to infinity as $n \rightarrow \infty$ such that with \mathfrak{P} -probability one, Procedure 2 chooses the same model that the Bayesian procedure does as $n \rightarrow \infty$.

Proof: As in the proof of Theorem 3.3.1, let M denote the largest model, so that $\hat{\theta}_{nL}^M = \hat{\theta}_{nL}$ is a “global” MCMLE over the set \mathcal{M} . Recall that $\hat{\theta}_{nL}$ exists and is unique (see Section 2.2).

Decompose \mathcal{M} as in (3.3.3), and suppose that $m \in \mathcal{M}_1(\varpi)$. In the same spirit used to get (3.3.20), we may write

$$\sup_{\vartheta \in \bar{\Theta}_m} \ell_L^m(x(n), \vartheta) = \sup_{\vartheta \in \bar{\Theta}_m} \ell_L(x(n), \vartheta) \quad (3.4.2)$$

via reparametrization of the exponential family forms. Note that the “global” supremum point $\hat{\theta}_{nL}$ converges to the true parameter θ \mathcal{P} -a.s. as $n \rightarrow \infty$ and as $L \rightarrow \infty$.

Clearly for $m \in \mathcal{M}_1(\varpi)$, $\theta \notin \bar{\Theta}_m$ and thus θ is some positive distance from $\bar{\Theta}_m$. Similarly there is some positive distance between the MLE $\hat{\theta}_n$ and the set $\bar{\Theta}_m$, as well as between $\hat{\theta}_{nL}$ and $\bar{\Theta}_m$ because of their existence and uniqueness properties. Then (3.3.20) and (3.3.21), in addition to Lemma 2.2.4, imply that there exists some $\delta > 0$ such that

$$\sup_{\vartheta \in \bar{\Theta}_m} \ell_L(x(n), \vartheta) \leq \ell_L(x(n), \hat{\theta}_{nL}) - \delta |\Lambda_n| \quad (3.4.5)$$

for large n and for large $L = L_n$, \mathcal{P} -a.s. Therefore,

$$\begin{aligned} Q_m^{(2)} &= \sup_{\vartheta \in \bar{\Theta}_m} \ell_L^m(x(n), \vartheta) - \frac{k_m}{2} \log |\Lambda_n| \\ &\leq \ell_L(x(n), \hat{\theta}_{nL}) - \delta |\Lambda_n| - \frac{k_m}{2} \log |\Lambda_n| \\ &= Q_M^{(2)} + \frac{k_M}{2} \log |\Lambda_n| - \delta |\Lambda_n| - \frac{k_m}{2} \log |\Lambda_n| \\ &= Q_M^{(2)} - \left[\delta |\Lambda_n| - \left(\frac{k_M}{2} - \frac{k_m}{2} \right) \log |\Lambda_n| \right] \\ &< Q_M^{(2)} \end{aligned} \quad (3.4.7)$$

for large n and large $L = L_n$. Hence Procedure 2 will not choose a model $m \in \mathcal{M}_1(\varpi)$ as $n \rightarrow \infty$ and $L_n \rightarrow \infty$.

Now, suppose $m \in \{\varpi\} \cup \mathcal{M}_2(\varpi)$. Then for such m , there exists a unique MCMLE $\hat{\theta}_{nL}^m$ in the set $\eta_\theta \cap \bar{\Theta}_m$. For every large n , \mathcal{P} -a.s., Theorem 3.3.1 holds; i.e., Procedure 1 chooses the same model that the Bayes procedure chooses. Fix such a large n , and let \hat{m} denote the chosen model, so that $Q_{\hat{m}}^{(1)}$ is the largest index.

By Lemma 2.2.1 and Lemma 2.2.4, \mathbf{P} -a.s. as $L = L_n \rightarrow \infty$,

$$\ell_L^m(x(n), \hat{\theta}_{nL}^m) \rightarrow \log \mathcal{L}_m(x(n), \hat{\theta}_n^m) + |\Lambda_n| b_n^m(\psi). \quad (3.4.8)$$

Hence, \mathbf{P} -a.s. for large $L = L_n$, the index $Q_m^{(2)}$ is close to the index $Q_m^{(1)}$ plus some constant for each $m \in \{\varpi\} \cup \mathcal{M}_2(\varpi)$. The index $Q_m^{(2)}$ is thus sifted out as the largest index among the indices $\{Q_m^{(2)}: m \in \{\varpi\} \cup \mathcal{M}_2(\varpi)\}$. Therefore, Procedure 2 chooses the same model as Procedure 1, \mathbf{P} -a.s. as $L = L_n \rightarrow \infty$; and in addition, Procedure 2 chooses the same model as the Bayes procedure \mathcal{P} -a.s. as $n \rightarrow \infty$ and $L_n \rightarrow \infty$. \square

Although Procedure 2 is an approximation to the Bayes solution and is tractable, there are complications involved in its usage. Primarily, these complications arise from using the MC-MLE, whose inherent difficulties are mentioned in Section 2.2 and discussed in Chapter IV. In addition, the applicability of Procedure 2 is restricted by the Uniqueness Condition 1.3.1 and the Positivity Condition 2.1.2.

3.5 Model Selection Based on the MPLE

For each $m \in \mathcal{M}$, let $\bar{\theta}_n^m$ be the MPLE restricted to $\bar{\Theta}_m$. Let $\mathcal{P}\mathcal{L}_m(\cdot, \cdot)$ denote the pseudo-likelihood for model $m \in \mathcal{M}$ in “minimal” form:

$$\mathcal{P}\mathcal{L}_m(x(n), \theta) = \exp\left\{|\Lambda_n| \left[\theta^T V_n^m - g_n^m(\theta)\right]\right\}, \quad (3.5.1)$$

where V_n^m and $g_n^m(\cdot)$ denote functions analogous to the sufficient statistic and cumulant generating function given in (3.3.1), respectively. Define the information criterion

$$Q_m^{(3)} = \sup_{\vartheta \in \bar{\Theta}_m} \log \mathcal{P}\mathcal{L}_m(x(n), \vartheta) - \frac{k_m}{2} \log |\Lambda_n|. \quad (3.5.2)$$

Procedure 3 Choose the model $\hat{m}_3 \in \mathcal{M}$ of largest index $Q_m^{(3)}$.

Although Procedure 3 does not give an approximation to the Bayesian solution, it does have tremendous computational advantages which are discussed and demonstrated in Chapter IV. It is also consistent in the following (weak) sense.

Definition 3.5.1 Assume $x(n)$ is a sample from P_θ with $\theta \in \Theta_\varpi$, $\varpi \in \mathcal{M}$. A selection procedure $\varrho(\cdot)$ is said to be consistent if $\lim_{n \rightarrow \infty} P_\theta(\varrho(x(n)) = \varpi) = 1$.

The decision function associated with Procedure 3 is given by $\varrho_3(x(n)) = \hat{m}_3$. Recall the decom-

position of $\mathcal{M} = \mathcal{M}_1(\varpi) \cup \{\varpi\} \cup \mathcal{M}_2(\varpi)$, as given in (3.3.3).

Lemma 3.5.1 *Under the Identifiability Condition 2.1.1, there exists a constant $c > 0$ such that*

$$P_\theta(\hat{m}_3 \in \mathcal{M}_1(\varpi)) \leq \exp(-n^c) \quad (3.5.3)$$

uniformly for all large n .

In addition, this next result provides an estimate for the error probability associated with $\mathcal{M}_2(\varpi)$.

Lemma 3.5.2 *Under the Identifiability Condition 2.1.1, there exists $\alpha > 0$ such that*

$$P_\theta(\hat{m}_3 \in \mathcal{M}_2(\varpi)) = O\left(\frac{1}{n^\alpha}\right). \quad (3.5.4)$$

These two lemmas are the very heart of the proof of the following theorem.

Theorem 3.5.1 *Under the Identifiability Condition 2.1.1, Procedure 3 is consistent.*

Proof: Let Case 1 correspond to $\hat{m}_3 \in \mathcal{M}_1(\varpi)$; and Case 2 correspond to $\hat{m}_3 \in \mathcal{M}_2(\varpi)$. Then it must be shown that the respective probabilities go to zero at appropriate rates as $n \rightarrow \infty$.

The result for Case 1 follows immediately from Lemma 3.5.1. The result for Case 2 is immediate from Lemma 3.5.2. \square

Recall that the consistency given in Definition 3.5.1 is in the weak sense. A selection procedure $\varrho(\cdot)$ is said to be strongly consistent if for a true model $\varpi \in \mathcal{M}$ and every $\theta \in \Theta_\varpi$, we have $\varrho(x(n)) \rightarrow \varpi$ P_θ -a.s. as $n \rightarrow \infty$. However, Procedure 3 does not seem to be strongly consistent. This can be roughly understood in the following way. The moderate deviation probabilities in Lemma 2.3.5 are not summable in n , and thus the Borel-Cantelli lemma does not apply. This notion is supported by the exact asymptotic order (as opposed to the bounds on the order derived here) for moderate deviation probabilities in the i.i.d. case derived in Rubin and Sethuraman (1965).

The validity of Theorem 3.5.1 without the Uniqueness Condition 1.3.1 and the Positivity Condition 2.1.2 is a major advantage for Procedure 3. Since the dependence involved in texture modelling may vary from short-range to long-range, Procedure 3 should have more exten-

sive applications. Once again, the information summarized in the pseudo-likelihood seems to provide a powerful tool for inference results.

IV. Simulations and Numerical Comparisons

4.1 Practical Considerations

There are several issues which need to be considered upon implementing the parameter estimators and the model selection criteria. One primary issue involves the observed random field. The theory developed to this point is dependent upon the boundary $\partial\Lambda_n$ of the $n \times n$ region Λ_n (see Definition 2.1.1). In practice, this boundary must be taken to be those sites in Λ_n which do not have a complete set of neighbors contained in Λ_n . Thus the theory applies to the $(n-R) \times (n-R)$ region nested in Λ_n , where R is the range of the GMRF under consideration. Of course, when R is small compared to n , this has no effect on the asymptotics developed in Chapter II and Chapter III.

A computational issue for any statistical simulation is the generation of pseudo-random numbers. Since we used the computer programming language FORTRAN for all of our programming needs, we used its random number generator $\text{RAN}(\cdot)$. When this function is initialized with an integer seed, it is supposed to return numbers which behave as if they are independent and uniformly distributed on the interval $(0, 1)$. Indeed, before actually using this function for our own purposes, we generated several samples and performed a χ^2 -goodness-of-fit test on each one: the samples did behave very significantly like i.i.d. uniform random numbers. All of our computer programs, which appear in Appendix Three, are written so that the seed is initialized externally. The user may then duplicate results by using the same initial seed, or study different results by using different initial seeds.

Implementation of the MCMLE presents a few important issues by itself. The choice of the Monte Carlo parameter ψ turns out to be critical: this issue is discussed in detail in Section 4.4. Also of critical importance is the dimension of the square region n versus the number of Monte Carlo samples L . We believe that L should be much larger than n , and we discuss this along with the simulation results for both parameter estimation and model selection in Section 4.4 and Section 4.5. Of less importance is a Monte Carlo sampling scheme for the reduction of variance in the parameter estimates; however, we do address this issue and give our simulation results which show the performance of selected variance-reduction schemes.

For computational purposes, our simulations have been performed on binary (± 1) ran-

dom fields. There are two factors influencing this choice. One is storage: images take up enormous amounts of storage space, and we were able to store the binary random fields rather efficiently. The other is speed: more gray-levels would naturally slow the programs (which are already quite slow in processing large images).

The most difficult issue is the choice of pair-interactions U_1 and U_2 . As was mentioned in Section 3.1, little has been done to address this choice. In our simulations, we used very simple multiplicative pair-interactions. Clearly, however, the model selection problem for GMRFs will remain incomplete until the problem of selecting neighborhood interactions is solved.

4.2 The Gibbs Sampler

Upon contemplating a simulation study of parameter estimation and model selection techniques for GMRFs, the first question that arises is: how does one simulate a random field? Fortunately, there is an easy answer. Geman and Geman (1984) proposed a simulation scheme, called the Gibbs sampler, which generates an ergodic Markov chain whose equilibrium distribution is the GMRF of interest. Its predecessor, the Metropolis algorithm (Metropolis, *et. al.*, 1953), was first proposed to study the equilibrium properties of large systems of molecules or atoms. The Gibbs sampler, a variation of the Metropolis algorithm, seems to be quite versatile and has found great popularity among statisticians who are interested in simulating dependent data.

There is an extensive literature in various Markov chain simulation algorithms, including the Gibbs sampler. Most recently, the papers of Aldous (1993) and Bertsimas and Tsitsiklis (1993) appeared as part of the Report from the Committee on Applied and Theoretical Statistics of the National Research Council on Probability and Algorithms in *Statistical Science*; while the papers of Besag and Green (1993), Smith and Roberts (1993), and Gilks, *et.al.* (1993) were issued from the Royal Statistical Society's Meeting on the Gibbs Sampler and other Markov Chain Monte Carlo Methods. Each of these papers gives an understanding of the current state of the art of Markov chain simulation, as well as insights into future directions. Our attention is restricted to the Gibbs sampler for simulating textures (as it was originally developed), as well as the Markov chain Monte Carlo parameter estimation technique of Geyer and Thompson (1992), described in Chapter II.

Given here is a brief discussion of what the Gibbs Sampler actually does. The Markov chain construction which follows is taken from Geman and Geman (1984). The goal of the construction is to sample from the distribution $P_\theta(\cdot | x_{\partial\Lambda_n})$ whose local characteristics are given by $p_i(\cdot | x_{\mathcal{N}_i})$, $i \in \Lambda_n$, where n is fixed.

Let $X^{(0)}\{n\}$ be an arbitrary random field on Λ_n , and let i_1, i_2, \dots be the schedule by which sites in Λ_n will be visited. Sample x_{i_1} from the distribution $p_{i_1}(\cdot | x_{N_{i_1}})$, and replace the value $X_{i_1}^{(0)}\{n\}$ (i.e., the value of $X^{(0)}\{n\}$ at site i_1) with x_{i_1} ; call this new random field $X^{(1)}\{n\}$. Proceed similarly for each site in the visiting sequence i_1, i_2, \dots , such that the random field $X^{(r)}\{n\}$ differs from the random field $X^{(r-1)}\{n\}$ only at site i_r , $r = 1, 2, \dots$. Thus $\{X^{(r)}\{n\}\}_{r=1}^{\infty}$ is a Markov chain; in fact, Geman and Geman (1984) have shown that, if each site is visited infinitely often, it is an aperiodic and irreducible (though nonstationary) Markov chain with equilibrium distribution $P_{\theta}(\cdot | x_{\partial\Lambda_n})$.

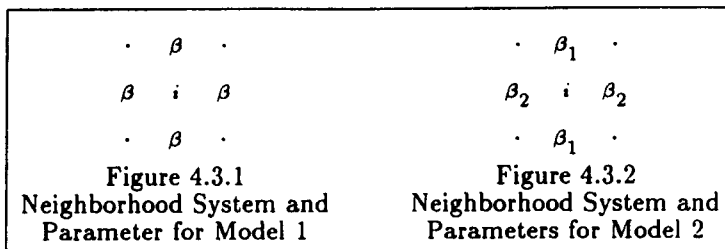
As we have stated, our goal is to simulate an $n \times n$ random field. Towards that goal, we begin with an i.i.d. binary random field (each site is ± 1). This random field is large enough so that it will contain the desired $n \times n$ random field, and so that each site at the edge of the desired random field has a complete set of neighbors. We systematically update each site in the embedded $n \times n$ random field according to the scheme just described, and in our work, we call a complete pass over the entire random field one iteration. (The Gibbs sampler has been used in this way by Besag, York, and Mollié (1991), as well as in Smith and Roberts (1993), and Besag and Green (1993). Such an implementation yields a Markov chain with stationary transition probabilities.) For our simulated random fields, we perform one thousand iterations. All of the 500×500 realizations which we have generated may be seen in Appendix Two. The FORTRAN program which generated them is named RFGEN and is the first program in Appendix Three.

4.3 Models Used for the Simulation Studies

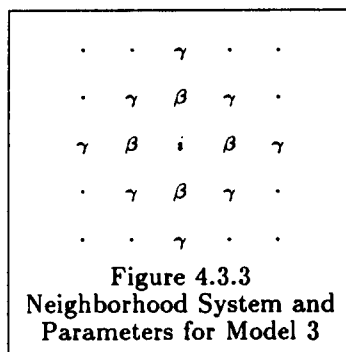
In all of our simulation studies, we have used the three types of models which have been mentioned previously in Chapter III (specifically, Figure 3.2.1, Figure 3.2.2, and Figure 3.2.3) when illustrating the potential for subtle differences among the candidate models. We discuss here our three Candidate Models and their associated version of the Uniqueness Condition 1.3.1, which gives conditions sufficient to guarantee no phase transitions (i.e., the absence of long-range dependence). Note that the Uniqueness Condition 1.3.1 is sufficient for no phase transitions, meaning that not only is there no long-range dependence, but the random field is also far away from exhibiting long-range dependence. See Chapter I for more discussion on the Uniqueness Condition. For convenience, we drop the numerical reference for the Uniqueness Condition from here on.

Model 1 is the simple Ising model, whose neighborhood system is depicted in Figure 4.3.1: four nearest neighbors with parameter space \mathbf{R} . The Ising model is the only one for which there are necessary and sufficient conditions for phase transitions. Let $\beta_c = \frac{1}{2} \log(1 + \sqrt{2})$.

Then there are no phase transitions if and only if $|\beta| < \beta_c$ (Georgii, 1988).



Model 2 is an anisotropic variation of the Ising model, with four nearest neighbors and parameter space \mathbb{R}^2 , as shown in Figure 4.3.2. This model clearly allows prominent or subtle directional features. Since $U_2(x_i, x_j) = x_i x_j$, it follows that $\bar{U}_2 = 2$, and it is easily verified that the Uniqueness Condition to guarantee no long-range dependence is given by $|\beta_1| + |\beta_2| < \frac{1}{2}$.



Model 3, in Figure 4.3.3, is an isotropic extension of the Ising model, with twelve nearest neighbors and parameter space \mathbb{R}^2 . Again, $\bar{U}_2 = 2$, and it follows quite easily the Uniqueness Condition to guarantee no long-range dependence is given by $4|\beta| + 8|\gamma| < 1$.

We have denoted the parameters in these models differently so that there will not be any confusion about the parameters from one model or another. In Model 2, the chosen notation emphasizes both the similarities to Model 1 as well as the directional variations that this model exhibits: β_1 governs the vertical direction, whose sites would have the same first coordinate in the traditional Cartesian system; β_2 governs the horizontal direction, whose sites would have the same second coordinate. In Model 3, again the notation emphasizes the similarity to and differences from Model 1: β is the inner parameter, governing the four nearest neighbors, while γ governs the remaining neighbors.

4.4 Simulation Study of the Parameter Estimates

In this study, we first generated a random field from a particular Candidate Model. Then, we ran a parameter estimation program (FORTRAN program DRIVER, the second program in Appendix Three) on the simulated random field, and by inspection, compared the estimates to the true parameter.

As was mentioned in Chapter II, there are some issues to be addressed when using the MCMLE. First, recall how the MCMLE is implemented: For an arbitrary, fixed parameter ψ , generate a Markov chain of random fields whose equilibrium distribution is $P_\psi(\cdot | x_{\partial\Lambda_n})$. Using this Markov chain and its ergodicity properties, create an approximation to the true likelihood function (to within a multiplicative constant). Any parameter value which maximizes this approximation is a MCMLE, and it converges \mathbf{P} -a.s. to the MLE, where \mathbf{P} is the probability measure for the entire Markov chain.

To begin implementation of the MCMLE, the Monte Carlo parameter ψ must be chosen. Although ψ is arbitrary, the choice does affect the number of Markov chain Monte Carlo (MCMC) samples needed to get a good approximation to the likelihood. (To sample from the Markov chain, we run the Gibbs sampler for 200 iterations and then take the 201st iteration as the first sample.) Our numerical experiments have shown, and it was indicated by Geyer and Thompson (1992), that ψ must be chosen close to the MLE in order to get a good estimate for true parameter θ with a reasonable number of MCMC samples. In fact, we discovered that on a 100×100 realization, a ψ which is off by ten percent from θ is not close enough, since more than 500 MCMC samples was required to get an MCMLE. Geyer and Thompson (1992) proposed an iterative method to get ψ close enough to θ for a good estimate, but in our set-up, we could not seem to get enough MCMC samples (and we used up to 500 samples) to make this method work. We chose to use no more than 500 MCMC samples because of the time involved in the execution of the program, and also because we had no way of determining how many samples it would actually take to produce a satisfactory estimate. So it seems that we need a good parameter estimate just to implement the MCMLE. For its convenience, we have chosen the MPLE as our Monte Carlo parameter ψ . Hence, we shall investigate the performance of the MPLE before we go further with the investigation of the MCMLE.

In Table 4.4.1, we investigate not only the performance of the MPLE for the Ising model (Model 1) with and without the Uniqueness Condition, but also what happens when n , the dimension of the random field, is small. Clearly, when $n = 10$, the MPLE performs very poorly; hence we shall not look at this case for any other model. However, the MPLE does quite well for $n = 100$, and even better for $n = 500$. Note that 100×100 or even 500×500 regions are not unreasonably sized regions, particularly with computer resolutions becoming more and more fine.

dimension	β	$\hat{\beta}$
10 × 10	0.1	0.24789
	1.0	3.75827
100 × 100	0.1	0.09470
	1.0	1.02434
500 × 500	0.1	0.09880
	1.0	1.01763

Table 4.4.1 MPLE for Model 1

We investigate Model 2 in Table 4.4.2. All of the parameter estimates seem perfectly acceptable for both $n = 100$ and $n = 500$. However, in some of the cases, there seems to be a trade-off in improvement as the dimension increases: as the accuracy of one parameter increases, the accuracy of the other decreases. We believe that this trade-off is due to the nature of the function maximization problem. We also believe that the accuracy of the MPLE will improve as the dimension increases, in spite of the apparent trade-off in accuracy.

dimension	β_1	$\hat{\beta}_1$	β_2	$\hat{\beta}_2$
100 × 100	.01	0.00790	0.1	0.09977
	0.1	0.09609	.01	0.01160
	0.1	0.10903	1.0	0.95636
	1.0	0.99652	0.1	0.10542
500 × 500	.01	0.01175	0.1	0.09568
	0.1	0.10343	.01	0.00675
	0.1	0.09975	1.0	1.00741
	1.0	1.00103	0.1	0.10000

Table 4.4.2 MPLE for Model 2

Finally, Table 4.4.3 shows the results for Model 3. Again, the MPLE is doing quite well, and the trade-off in accuracy just discussed does not seem to be as apparent. Notice the case where $n = 100$, $\beta = 1.0$, and $\gamma = 0.1$. Clearly, this estimate is no good, but whatever caused such an anomaly is repaired when n is increased to 500. We believe this is just the nature of statistics.

dimension	β	$\hat{\beta}$	γ	$\hat{\gamma}$
100 × 100	.01	0.00651	0.1	0.09991
	0.1	0.09240	.01	0.00878
	0.1	0.05876	1.0	0.80365
	1.0	1.59903	0.1	-0.14331
500 × 500	.01	0.00949	0.1	0.10080
	0.1	0.09835	.01	0.00944
	0.1	0.07987	1.0	1.03620
	1.0	0.93730	0.1	0.13724

Table 4.4.3 MPLE for Model 3

Thus we have demonstrated that, for our Candidate Models, the MPLE is as good a parameter estimate as the theory says it should be. It was mentioned earlier that we have chosen the MPLE to be our Monte Carlo parameter ψ , to be used in implementation of the MC-MLE. The natural question is: Will the MCMLE procedure improve upon the MPLE? Below, Table 4.4.4, Table 4.4.5, and Table 4.4.6 provide comparisons of the MCMLE with the MPLE.

A computational issue in using the MCMLE is the CPU time required by the program. The vast majority of the CPU time will be taken by the Gibbs Sampler to create the MCMC samples needed to compute the MCMLE. There is a significant difference in the time required to generate a 100×100 random field versus that required for a 500×500 random field – seconds versus minutes, or even minutes versus hours, depending on the number of iterations taken in the Gibbs Sampler. The results presented for this comparison were done with 100×100 random fields, and MCMC sampling was begun only after 200 iterations of the Gibbs Sampler. For the sake of comparison, we computed the MCMLE for 5, 100, and 500 MCMC samples, as indicated in the column MC(\cdot). An entry of NC means that the estimate was not computable (because there were not enough MCMC samples). Approximate CPU times for each parameter estimate are given in the accompanying discussions.

Table 4.4.4 gives a thorough comparison of the parameter estimates for Model 1. For selected values, more than one random field realization was generated; for the two cases of an

extreme value of β , only two of the estimates were attempted (those being the MPLE and the MCMLE with 500 MCMC samples). We also investigated the behavior of the parameter estimates when β is close to $\beta_c = \frac{1}{2} \log(1 + \sqrt{2}) \approx .44$, the critical value for the presence or absence of phase transitions (long-range dependence). For the simulations involving Model 1, the MPLE took approximately 1 second of CPU time to complete; the MCMLE with 5 MCMC samples took about 20 CPU-seconds; the MCMLE with 100 MCMC samples took 30 CPU-seconds; and the MCMLE with 500 samples took 75 CPU-seconds.

In some cases, the MCMLE offers an improvement over the MPLE, but we feel that the improvement is not enough to justify the expenditure in CPU time. In addition, the MCMLE, for small numbers of MCMC samples, has a little bit of difficulty around and past the value β_c , where the MPLE is not affected. In general, we feel that the number of MCMC samples L should be much larger than the dimension n of the random field. Both parameter estimates had difficulty with the extreme value of $\beta = 2$. The problem was corrected for the MPLE when n was increased to 500, implying that such long-range dependence does have an adverse effect on the MPLE, but such an effect may be overcome with a larger n (if available).

β	MPLE	MC(5)	MC(100)	MC(500)
.10	.094700	.090453	.094263	.093712
.10	.108884	.106533	.109537	.109433
.10	.109613	.089351	.105396	.105459
.30	.291658	.265816	.288035	.287200
.40	.404876	.340168	.395253	.398213
.40	.396936	.391755	.394829	.395899
.41	.417348	.400621	.408147	.413911
.42	.420699	NC	.420003	.419800
.43	.427675	NC	.425048	.424298
.44	.433144	NC	.434090	.433187
.45	.442555	NC	.450788	.442547
.50	.500563	NC	NC	.509732
.50	.495683	NC	NC	.502587
1.0	.955562	NC	NC	NC
1.0	1.02434	NC	NC	1.02788
2.0	7.48787			7.47967
2.0	7.23350			7.22983

Table 4.4.4 MPLE vs. MCMLE for Model 1

The investigations for Model 2 were not so thorough, as this model is a slightly more complex version of Model 1. For the comparison using Model 2, we see familiar phenomena: the MCMLE does not offer much improvement over the MPLE. The MCMLE with only 5 MC-MC samples is performing quite poorly even in the first two cases, which satisfy the Uniqueness Condition. More difficulties arise as the Uniqueness Condition is no longer satisfied. For the simulations involving Model 2, the CPU times required are comparable to those required using Model 1.

		MPLE	MC(5)	MC(100)	MC(500)
β_1	.01	.007899	-.338189	.005851	.005171
β_2	0.1	.099767	-.571546	.099162	.099127
β_1	0.1	.096094	-.506890	.096130	.096169
β_2	.01	.011600	-1.10292	.009396	.009783
β_1	1.0	.996524	NC	1.00293	.997085
β_2	0.1	.105420	NC	.091354	.095746
β_1	0.1	.109033	NC	NC	.086748
β_2	1.0	.956369	NC	NC	1.01163

Table 4.4.5 MPLE vs. MCMLE for Model 2

Our investigation of Model 3 is similar to that of Model 2. Here, the MCMLE seems to have quite a struggle improving on the MPLE; of course, the model itself is even more complex than the previous two. Again, the MCMLE has much more difficulty with the last two cases, which do not satisfy the Uniqueness Condition. Also, when the MPLE fails, the MCMLE has no hope of producing a good estimate with a reasonable number of MCMC samples. For the simulations involving Model 3, the CPU times were about five seconds more than the CPU times for the previous models – except the MPLE, whose CPU time remained at approximately one second.

		MPLE	MC(5)	MC(100)	MC(500)
β	.01	.006511	.005347	.004463	.004601
γ	0.1	.099907	.095799	.097638	.097288
β	0.1	.092401	.086311	.089247	.089133
γ	.01	.008777	.009382	.010331	.010222
β	0.1	.058756	NC	-.157837	.031916
γ	1.0	.803648	NC	.895044	.798601
β	1.0	1.59903	NC	NC	NC
γ	0.1	-.143314	NC	NC	NC

Table 4.4.6 MPLE vs. MCMLE for Model 3

Geyer and Thompson (1992) proposed a method for variance-reduction in the MCMLE: rather than sampling the Markov chain sequentially, sample at intervals so that there is not so much dependence between samples. This method appears to use the definition of Geman and Geman (1984), that each realization from the Markov chain differs from the previous one by only one site. However, since we consider one iteration of the Gibbs sampler to be one complete pass over the entire random field, it would seem that we are inherently using a variance-reduction scheme. Nevertheless, we tried three variance-reduction schemes, the results of which are in Table 4.4.7, in the hope that the MCMLE would give a better performance. Scheme 1 took 100 MCMC samples, each being the fifth sample generated; Scheme 2 took 500 samples, each being every second one generated; and Scheme 3 also took 500 samples with each being every third one generated. Note that for Model 1 and Model 3, β_1 in the table corresponds to β , while β_2 corresponds to γ . Even with the variance reduction schemes, the MCMLE experienced the same problems as before.

model	scheme	β_1	$\hat{\beta}_1$	β_2	$\hat{\beta}_2$
1	1	0.1	.09355	--	--
	2		.09363		--
	3		.09393		--
	1	1.0	1.0281	--	--
	2		1.0225		--
	3		1.0208		--
2	1	.01	.00601	0.1	.09813
	2		.00505		.09901
	3		.00502		.09943
	1	0.1	.09618	.01	.00890
	2		.09554		.00935
	3		.09582		.00938
	1	0.1	NC	1.0	NC
	2		.09980		.98798
	3		.09790		.98537
	1	1.0	1.0004	0.1	.09261
	2		.99335		.09586
	3		.99736		.09687
3	1	.01	.00387	0.1	.09713
	2		.00412		.09811
	3		.00398		.09801
	1	0.1	.08986	.01	.00988
	2		.09012		.00993
	3		.09058		.00993
	1	0.1	NC	1.0	NC
	2		NC		NC
	3		NC		NC
	1	1.0	NC	0.1	NC
	2		NC		NC
	3		NC		NC

Table 4.4.7 Variance Reduction Schemes

Thus, numerically and in our current set-up, it would seem that the MPLE is superior to the MCMLE. This is not true in all situations, and Geyer and Thompson (1992) give an example in which the MCMLE is superior to the MPLE. However, it seems clear to us that in situations such as ours in which the configuration space is of high dimensionality, the MPLE will almost always out-perform the MCMLE.

4.5 Simulation Study of the Model Selection Procedures

Here we study the model selection procedures derived in Chapter III. To conduct this study, we first generated a sample from one of the Candidate Models with known parameter values. Then, pretending we did not know the true model, we evaluated the appropriate index $Q_m^{(\cdot)}$ for each of the Candidate Models (depending on which procedure we were studying). The chosen model was then the model with the largest index. The FORTRAN program which conducted the model selection procedures is MODSEL, in Appendix Three.

Of course, Procedure 1, based on the MLE, cannot be implemented due to the intractability of the MLE itself. In spite of the results in Section 4.4, we studied the performance of Procedure 2, based on the MCMLE, specifically with the true model being Model 2 and for two different MCMC sample sizes. We chose Model 2 as the true model for its more complex directional features; we rejected Model 1 because of its simplicity, and we rejected Model 3 because the MCMLE seems to have more difficulty with it (see Table 4.4.4, Table 4.4.5, and Table 4.4.6). We did not employ any variance-reduction scheme since such a scheme requires more time and offers little (if any) improvement in the parameter estimates. Again, all of the MCMLE results were obtained using 100×100 random fields; and for Model 1 and Model 3, β_1 in the table corresponds to β , while β_2 corresponds to γ . The symbol ** indicates the chosen model.

Table 4.5.1 displays the results of using Procedure 2 when the Uniqueness Condition is satisfied by the realization. Obviously, the performance of Procedure 2 is acceptable, and the choice of model is quite clear in the sense that there are no other values close to the maximum index.

$\beta_1 = .01 \quad \beta_2 = .1$				
MCMC sample size	model	$\hat{\beta}_1$	$\hat{\beta}_2$	$Q_m^{(2)}$
100	1	.05197	--	52
	**2	.00483	.09988	90
	3	.05300	.00037	47
500	1	.05181	--	52
	**2	.00493	.09955	90
	3	.05224	-.00041	47
$\beta_1 = .1 \quad \beta_2 = .01$				
MCMC sample size	model	$\hat{\beta}_1$	$\hat{\beta}_2$	$Q_m^{(2)}$
100	1	.05243	--	52
	**2	.09620	.00976	83
	3	.05236	.00187	46
500	1	.05222	--	52
	**2	.09568	.00967	83
	3	.05236	.00082	46

Table 4.5.1 MCMLE-Based Model Selection with the Uniqueness Condition

The results in Table 4.5.2, using Procedure 2 for model selection in which the true model does not satisfy the Uniqueness Condition, are not nearly so pleasant. In all cases, the procedure had difficulty evaluating the MCMLE under Model 3. Even though the correct choice was made in all but one case, the choice was not quite “fair” since each index could not be evaluated. In any case, the difficulties in using Procedure 2 have been clearly illustrated.

$\beta_1 = .1 \quad \beta_2 = 1$				
MCMC sample size	model	$\hat{\beta}_1$	$\hat{\beta}_2$	$Q_m^{(2)}$
100	**1	.49334	--	6342
	2	NC	NC	NC
	3	NC	NC	NC
500	1	.49338	--	6343
	**2	.09718	.98382	7869
	3	NC	NC	NC
$\beta_1 = 1 \quad \beta_2 = .1$				
MCMC sample size	model	$\hat{\beta}_1$	$\hat{\beta}_2$	$Q_m^{(2)}$
100	1	.49715	--	6473
	**2	1.0017	.09577	8237
	3	NC	NC	NC
500	1	.49723	--	6474
	**2	.95175	.06875	8239
	3	NC	NC	NC

Table 4.5.2 MCMLE-Based Model Selection without the Uniqueness Condition

We explore model selection using Procedure 3 much more extensively. The notation in the tables is the same as that used in the investigation of Procedure 2 above. We study Procedure 3 both for 100×100 and 500×500 random fields; there are special cases, which we call “extreme cases,” displayed separately and studied for their strong neighborhood interactions. These are conducted for 500×500 random fields only.

Upon first glance at the results in Table 4.5.3 and Table 4.5.4, it would seem that Procedure 3 tends to over-parametrize. However, upon further inspection, two very interesting things may be observed. The first thing is the parameter estimates. Model 1 may be considered a special case of both Model 2 and Model 3, which is confirmed by the parameter estimates. Indeed, in all cases, the chosen model is very close to the true model. The second thing is the indexes. None of the maximum indexes are decisive in the sense that the maxima are not much greater than the indexes with which they were compared. Again, this is probably due to the fact that all models may be reduced to Model 1. Procedure 3 becomes understandably “confused,” and therefore chooses the most all-inclusive model.

$\beta = .1$				
dimension	model	$\hat{\beta}_1$	$\hat{\beta}_2$	$Q_m^{(3)}$
100 × 100	1	.09470	--	-6488
	2	.09382	.09571	-6492
	**3	.09397	.00160	-6231
500 × 500	1	.09880	--	-167024
	2	.10120	.09651	-167028
	**3	.09921	-.00108	-165694

Table 4.5.3 MPLE-Based Model Selection with the Uniqueness Condition when the True Model is Model 1

$\beta = 1$				
dimension	model	$\hat{\beta}_1$	$\hat{\beta}_2$	$Q_m^{(3)}$
100 × 100	**1	1.0243	--	-128
	2	1.0039	1.0444	-132
	3	1.0663	-.02860	-129
500 × 500	1	1.0176	--	-2523
	2	1.0325	1.0027	-2529
	**3	1.0676	-.02667	-2485

Table 4.5.4 MPLE-Based Model Selection without the Uniqueness Condition when the True Model is Model 1

The extreme cases for Procedure 3, in Table 4.5.5, tell the same story as was told just above. Notice how, for $\beta = 2$, the MPLE has improved for Model 1 as n grew from 100 to 500 (see Table 4.4.4), but it also is doing quite badly when Model 2 is assumed. Also notice the failure of the MPLE for $\beta = 10$, which may be attributed to too much long-range dependence and too small of a sample random field.

$\beta = -1$			
model	$\hat{\beta}_1$	$\hat{\beta}_2$	$Q_m^{(3)}$
1	-1.0242	--	-2702
2	-1.0053	-1.0425	-2708
**3	-.97953	-.02614	-2678
$\beta = 2$			
model	$\hat{\beta}_1$	$\hat{\beta}_2$	$Q_m^{(3)}$
1	1.9805	--	-1206
2	6.3686	1.7551	-1210
**3	1.9371	.04030	-1201
$\beta = 10$			
model	$\hat{\beta}_1$	$\hat{\beta}_2$	$Q_m^{(3)}$
1	2.1456	--	-1151
2	6.5413	1.9286	-1156
**3	2.1077	.03576	-1144

Table 4.5.5 Extreme Cases: True Model is Model 1

In Table 4.5.6, Procedure 3 is still trying to over-parametrize the fitted model, even though now, the true model is Model 2. However, if one looks at the sample random field upon which this selection procedure was conducted, it is nearly indistinguishable from the sample taken from Model 1. This phenomenon will show up again, and once we have displayed all of our results, we shall comment on it.

$\beta_1 = .01 \quad \beta_2 = .1$				
dimension	model	$\hat{\beta}_1$	$\hat{\beta}_2$	$Q_m^{(3)}$
100 × 100	1	.05423	--	-6605
	2	.00790	.09770	-6569
	**3	.05390	-.00123	-6343
500 × 500	1	.05414	--	-170461
	2	.01175	.09568	-169606
	**3	.05403	-.00153	-169099
$\beta_1 = .1 \quad \beta_2 = .01$				
dimension	model	$\hat{\beta}_1$	$\hat{\beta}_2$	$Q_m^{(3)}$
100 × 100	1	.05414	--	-6606
	2	.09609	.01160	-6575
	**3	.05269	.00013	-6345
500 × 500	1	.05557	--	-170388
	2	.10343	.00657	-169255
	**3	.05554	-.00229	-169032

Table 4.5.6 MPLE-Based Model Selection with the Uniqueness Condition when the True Model is Model 2

The results in Table 4.5.7 are very surprising at first glance: Procedure 3 is selecting the correct model quite “confidently” when any long-range dependence may be present. If one looks at the sample on which the model selection was performed, one sees clear directional structures. We believe that the selection procedure is able to discern these structures as well, via the local characteristics, and is thus able to make the correct choice.

$\beta_1 = .1 \quad \beta_2 = 1$				
dimension	model	$\hat{\beta}_1$	$\hat{\beta}_2$	$Q_m^{(3)}$
100 × 100	1	.53636	--	-2692
	**2	.10903	.95636	-1908
	3	.74123	-.12976	-2437
500 × 500	1	.54126	--	-64924
	**2	.09975	1.0074	-44670
	3	.74073	-.12326	-60873
$\beta_1 = 1 \quad \beta_2 = .1$				
dimension	model	$\hat{\beta}_1$	$\hat{\beta}_2$	$Q_m^{(3)}$
100 × 100	1	.53967	--	-2610
	**2	.99652	.10542	-1796
	3	.72550	-.11968	-2388
500 × 500	1	.54077	--	-65320
	**2	1.0010	.10003	-45135
	3	.73140	-.11896	-61489

Table 4.5.7 MPLE-Based Model Selection without the Uniqueness Condition when the True Model is Model 2

The results in Table 4.5.8 seem to confirm our previous comments. The structure exhibited by the associated realizations is again obvious, and the model selection procedure seems to find this information extremely useful in estimating the true model.

$\beta_1 = 1 \quad \beta_2 = -1$			
model	$\hat{\beta}_1$	$\hat{\beta}_2$	$Q_m^{(3)}$
1	-.01349	--	-171907
**2	.97933	-1.0521	-2379
3	-.01865	-.18243	-169900
$\beta_1 = -1 \quad \beta_2 = 1$			
model	$\hat{\beta}_1$	$\hat{\beta}_2$	$Q_m^{(3)}$
1	-.02736	--	-171902
**2	-.98039	1.0469	-2398
3	-.03704	-.19437	-169833

Table 4.5.8 Extreme Cases: True Model is Model 2

We see displayed in Table 4.5.9 several things which we have discussed already. In the first case, notice that there is a much stronger interaction with the outermost neighbors than with the four nearest neighbors. This seems to make the selection more clear since this type of interaction is unique among the candidates. In the second case, the interaction is strongest among the four nearest neighbors, and weak with the outer neighbors, making the true model behave more like Model 1. In this case, the choice was not quite so clear – similar to what happened in Table 4.5.3.

$\beta = .01 \quad \gamma = .1$				
dimension	model	$\hat{\beta}_1$	$\hat{\beta}_2$	$Q_m^{(3)}$
100 × 100	1	.00810	--	-6659
	2	.00726	.00951	-6664
	**3	.00651	.09991	-5893
500 × 500	1	.01641	--	-171703
	2	.01742	.01590	-171709
	**3	.00949	.10080	-156204
$\beta = .1 \quad \gamma = .01$				
dimension	model	$\hat{\beta}_1$	$\hat{\beta}_2$	$Q_m^{(3)}$
100 × 100	1	.09439	--	-6484
	2	.09382	.09513	-6489
	**3	.09240	.00878	-6222
500 × 500	1	.10111	--	-166622
	2	.10312	.09924	-166626
	**3	.09835	.00944	-165187

Table 4.5.9 MPLE-Based Model Selection with the Uniqueness Condition when the True Model is Model 3

We see phenomena in Table 4.5.10 similar to what we saw in Table 4.5.9. In the first case, the outer interaction is much stronger than the inner interaction, and the choice seems to be more clear than in the second case, where the interactions are more like those of Model 1. Notice the choice of Model 1 in the second case, when $n = 100$: this was the particular realization where the MPLE failed to give a good estimate of the true parameters.

$\beta = .1 \quad \gamma = 1$				
dimension	model	$\hat{\beta}_1$	$\hat{\beta}_2$	$Q_m^{(3)}$
100 × 100	1	.72542	--	-257
	2	1.4017	.26369	-218
	**3	.05876	.80365	-58
500 × 500	1	.88311	--	-2741
	2	.68253	1.1027	-2752
	**3	.07987	1.0362	-727
$\beta = 1 \quad \gamma = .1$				
dimension	model	$\hat{\beta}_1$	$\hat{\beta}_2$	$Q_m^{(3)}$
100 × 100	**1	1.3717	--	-37
	2	1.2683	6.6690	-41
	3	1.5990	-.14331	-40
500 × 500	1	1.1482	--	-1656
	2	1.0998	1.2038	-1661
	**3	.93730	.13724	-1606

Table 4.5.10 MPLE-Based Model Selection without the Uniqueness Condition when the True Model is Model 3

The results shown in Table 4.5.11 again seem to indicate that the choice is reasonably clear when the distinguishing structure of Model 3 is emphasized.

$\beta = 1 \quad \gamma = -1$			
model	$\hat{\beta}_1$	$\hat{\beta}_2$	$Q_m^{(3)}$
1	.78420	--	-164212
2	.75423	.81592	-164212
**3	1.0274	-1.0221	-6381
$\beta = -1 \quad \gamma = 1$			
model	$\hat{\beta}_1$	$\hat{\beta}_2$	$Q_m^{(3)}$
1	-1.4425	--	-1922
2	-1.3205	-4.6991	-1924
**3	-.95181	.96529	-1040

Table 4.5.11 Extreme Cases: True Model is Model 3

In conclusion, several things seem clear in the simulation study of the model selection procedures. For Procedure 2, the MCMLE-based model selection procedure, although adequate when there are no phase transitions, does not seem to be able to evaluate the index for all candidate models when there may be phase transitions (partially because there are not enough MCMC samples to provide a good evaluation). This is supported by the theory, which says that Procedure 2 is asymptotically a Bayes solution only under the Uniqueness Condition (*i.e.*, no phase transitions).

Procedure 3 seems to work well in all cases, given allowances for the similarity of the Candidate Models. When there are no phase transitions and an identifying structure cannot be discerned, Procedure 3 tends to over-parametrize. In such cases, the sample from the true model was practically indistinguishable from a sample from Model 1 with no phase transitions. On the other hand, when phase transitions were possible and structures unique to the true model were given strong interactions, Procedure 3 worked extremely well, making a clear and correct choice over all other candidates.

4.6 Remarks on Application to Real Textures

We attempted model selection via Procedure 3 on real textures – taken primarily from the album of Brodatz (1966) – using these Candidate Models. Of course, since the Candidate Models are so simple, the attempts on complex textures such as wood grain were largely unsuccessful. Several things may be done to improve on the performance of this model selection procedure. A larger pool of candidate models is an obvious first step. Next, one may investigate using other, more complex, sets of deterministic interactions than the one we assumed; as we have mentioned a couple of times already, there is still no efficient systematic way of estimating these interactions. Clearly, there is a significant amount of work to be done before Procedure 3 can be easily implemented for modelling real textures.

Appendix One

Proofs of Lemmas

A.1 Notation Reminder plus Two Supporting Lemmas

- $b_n(\theta)$ Cumulant generating function for the likelihood $\mathcal{L}(x(n), \theta)$.
- $B(\vartheta)$ Covariance matrix of Y_n , given in the Positivity Condition 2.1.2.
- $\partial\Lambda$ Boundary of the region $\Lambda \subset \mathbb{Z}^2$.
- E_θ Expectation with respect to P_θ .
- \mathfrak{S}_n Set of configurations on Λ_n on which the empirical probabilities for seeing all configurations $\xi \in \Omega_{\Lambda_{2R+1}}$ is strictly positive.
- $g_n(\theta)$ ‘‘Cumulant generating function’’ for the pseudo-likelihood $\mathcal{P}\mathcal{L}(x(n), \theta)$.
- $\mathbf{1}_n(\xi)$ Indicator function, indicating if a sub-configuration of Λ_n is identical to the configuration $\xi \in \Omega_{\Lambda_{2R+1}}$.
- \mathcal{L} Conditional likelihood function.
- Λ_n $n \times n$ region in \mathbb{Z}^2 .
- \hat{m}_1 Model chosen by Procedure 1.
- \hat{m}_2 Model chosen by Procedure 2.
- \hat{m}_3 Model chosen by Procedure 3.
- ϖ True model.
- \mathcal{M} Set of candidate models.
- $\mathcal{M}_1(\varpi)$ Set of candidate models whose subspaces do not contain Θ_ϖ .
- $\mathcal{M}_2(\varpi)$ Set of candidate models whose subspaces properly contain Θ_ϖ .
- $N_n(\xi)$ Number of sub-configurations of Λ_n identical to the configuration $\xi \in \Omega_{\Lambda_{2R+1}}$.
- \mathcal{N}_i Neighborhood of site $i \in \mathbb{Z}^2$.
- Ω Configuration space for (or, set of all possible realizations on) \mathbb{Z}^2 .

Ω_Λ	Configuration space for the region $\Lambda \subset \mathbb{Z}^2$.
p_i	Local characteristic at the site $i \in \mathbb{Z}^2$.
P_θ	Gibbs-Markov random field with parameter θ .
$\mathfrak{P}\mathcal{L}$	Pseudo-likelihood function.
$\hat{\theta}_n$	MLE of θ based on the observation $x(n)$.
$\hat{\theta}_{nL}$	MCMLE of θ based on the observation $x(n)$ and the simulated Markov chain $\{X^{(l)}(n)\}_{l=1}^L$
$\tilde{\theta}_n$	MPLE of θ based on the observation $x(n)$.
Θ_m	Parameter space for the model m .
V_n	“Sufficient statistic” for the pseudo-likelihood $\mathfrak{P}\mathcal{L}(x(n), \theta)$.
$x(n)$	Observed random field on Λ_n .
\tilde{x}	Periodic configuration constructed from $x(n)$.
\tilde{x}_Λ	That part of the periodic configuration, constructed from $x(n)$, which is in the region $\Lambda \subset \mathbb{Z}^2$
X	Random field on \mathbb{Z}^2 .
X_Λ	Random field on the region $\Lambda \subset \mathbb{Z}^2$.
Y_n	Sufficient statistic for the likelihood $\mathcal{L}(x(n), \theta)$.

Lemma A.1.1 *Let $\mathfrak{B}_1, \dots, \mathfrak{B}_T$ be bounded regions in \mathbb{Z}^2 , $T \in \mathbb{N}$, and $\mathfrak{C} = \mathbb{Z}^2 \setminus (\bigcup_{t=1}^T \mathfrak{B}_t)$ be the corridor between these regions. If the distances between \mathfrak{B}_t and $\mathfrak{B}_{t'}$ are greater than R for all $t \neq t'$, then for any collection of bounded measurable functions $f_t: \Omega_{\mathfrak{B}_t} \rightarrow \mathbb{R}$, $t = 1, \dots, T$, we have*

$$E_\theta \left\{ \prod_{t=1}^T f_t(X_{\mathfrak{B}_t}) \middle| x_{\mathfrak{C}} \right\} = \prod_{t=1}^T E_\theta [f_t(X_{\mathfrak{B}_t}) | x_{\mathfrak{C}}] \quad (\text{A.1.1})$$

uniformly for all corridor configurations $x_{\mathfrak{C}} \in \Omega_{\mathfrak{C}}$, where $E_\theta(\cdot | x_{\mathfrak{C}})$ is the conditional expectation with respect to $P_\theta(\cdot | x_{\mathfrak{C}})$.

Proof: Recall that R is the range of the potentials which induce the GRF P_θ . Because of the Markov property of order R for the Gibbs distribution, the collection of random fields $X_{\mathfrak{B}_t}$, $t = 1, \dots, T$, are conditionally independent given the corridor \mathfrak{C} . The result then follows easily.

□

Lemma A.1.2 Under the Uniqueness Condition 1.3.1,

$$|P_\theta(X_\Lambda = \xi, X_\Delta = \zeta) - P_\theta(X_\Lambda = \xi)P_\theta(X_\Delta = \zeta)| \leq C|\Lambda| \exp[-cd(\Lambda, \Delta)] \quad (\text{A.1.2})$$

for some $c, C > 0$, and uniformly for all configurations $\xi \in \Omega_\Lambda$, $\zeta \in \Omega_\Delta$ and all bounded regions $\Lambda, \Delta \in \mathbb{Z}^2$, where $d(\Lambda, \Delta)$ is the Euclidian distance (or an equivalent metric) between Λ and Δ .

Proof: This lemma is a result of several statements found in Georgii (1988). First, Georgii's Proposition 8.8 says that our Uniqueness Condition 1.3.1 implies Dobrushin's uniqueness condition (we have discussed this equivalence in Chapter I). Georgii's Example 8.28 (1) gives an inequality involving the right-hand side of (A.1.2), which holds under Dobrushin's condition as well as under finite-range potentials. Finally, Georgii's Corollary 8.32, also under Dobrushin's condition, provides an inequality which, in combination with the inequality from his Example 8.28 (1), gives the desired inequality (A.1.2). \square

A.2 Proofs of Lemmas from Chapter II.

Lemma 2.1.1 Under the Uniqueness Condition 1.3.1 and the Positivity Condition 2.1.2,

$$c < v^T \nabla^2 b_n(\vartheta) v \leq C \quad (\text{2.1.17})$$

for some constants $c, C > 0$; uniformly for all unit vectors $v \in \mathbb{R}^k$; all $\tilde{x}_{\partial\Lambda_n}$; all ϑ in a small neighborhood of θ ; and all large n .

Proof: Express Y_n , whose components are given in (2.1.10) and (2.1.11) as:

$$Y_n = \frac{1}{|\Lambda_n|} \sum_{i \in \Lambda_n} Z_i, \quad (\text{A.2.1})$$

where each Z_i is a vector in \mathbb{R}^k – whose components are given by $(Z_{i1}, Z_{i2}, \dots, Z_{ik})^T$ – which is dependent only upon \tilde{x}_i and $\tilde{x}_{\mathcal{N}_i}$. For $i \in \Lambda_n$, write

$$\bar{Z}_i = Z_i - E_\theta(Z_i | \tilde{x}_{\partial\Lambda_n}), \quad (\text{A.2.2})$$

and note that $E_\theta(\bar{Z}_i) = 0$.

In order to complete this proof, we need to show

$$E_\theta\left(\left\|\sum_{i \in \Lambda_n} \bar{Z}_i\right\|^2\right) = O(|\Lambda_n|). \quad (\text{A.2.3})$$

Towards this goal, we may write

$$\begin{aligned} E_\theta\left(\left\|\sum_{i \in \Lambda_n} \bar{Z}_i\right\|^2\right) &= E_\theta\left[\sum_{m=1}^k \left(\sum_{i \in \Lambda_n} \bar{Z}_{im}\right)^2\right] \\ &= \sum_{m=1}^k E_\theta\left(\sum_{i \in \Lambda_n} \bar{Z}_{im}^2 + \sum_{\substack{i, j \in \Lambda_n \\ i \neq j}} \bar{Z}_{im} \bar{Z}_{jm}\right) \\ &= \sum_{m=1}^k \sum_{i \in \Lambda_n} E_\theta(\bar{Z}_{im}^2) + \sum_{m=1}^k \sum_{\substack{i, j \in \Lambda_n \\ i \neq j}} E_\theta(\bar{Z}_{im} \bar{Z}_{jm}), \\ &\leq C_1 |\Lambda_n| + \sum_{m=1}^k \sum_{\substack{i, j \in \Lambda_n \\ i \neq j}} E_\theta(\bar{Z}_{im} \bar{Z}_{jm}) \end{aligned} \quad (\text{A.2.4})$$

for some $C_1 > 0$, because of the translation invariance and the boundedness of the components of \bar{Z}_i . Define $\Delta_i = (\{i\} \cup \mathcal{N}_i)$ for $i \in \mathbb{Z}^2$, and note that $|\Delta_i| < \infty$ because of the MRF property. Also, recall that $E_\theta(\bar{Z}_i) = 0$. Then, for the expectation in (A.2.4), we see

$$\begin{aligned} E_\theta(\bar{Z}_{im} \bar{Z}_{jm}) &= E_\theta(\bar{Z}_{im} \bar{Z}_{jm}) - E_\theta(\bar{Z}_{im}) E_\theta(\bar{Z}_{jm}) \\ &= \sum_{\xi \in \Omega_{\Delta_i}} \sum_{\zeta \in \Omega_{\Delta_j}} \bar{Z}_{im}(\xi) \bar{Z}_{jm}(\zeta) P_\theta(X_{\Delta_i} = \xi, X_{\Delta_j} = \zeta) \\ &\quad - \sum_{\xi \in \Omega_{\Delta_i}} \sum_{\zeta \in \Omega_{\Delta_j}} \bar{Z}_{im}(\xi) \bar{Z}_{jm}(\zeta) P_\theta(X_{\Delta_i} = \xi) P_\theta(X_{\Delta_j} = \zeta) \\ &= \sum_{\xi \in \Omega_{\Delta_i}} \sum_{\zeta \in \Omega_{\Delta_j}} \bar{Z}_{im}(\xi) \bar{Z}_{jm}(\zeta) \cdot \\ &\quad \left[P_\theta(X_{\Delta_i} = \xi, X_{\Delta_j} = \zeta) - P_\theta(X_{\Delta_i} = \xi) P_\theta(X_{\Delta_j} = \zeta) \right] \end{aligned} \quad (\text{A.2.5})$$

so that

$$\begin{aligned} |E_\theta(\bar{Z}_{im} \bar{Z}_{jm})| &\leq \sum_{\xi \in \Omega_{\Delta_i}} \sum_{\zeta \in \Omega_{\Delta_j}} |\bar{Z}_{im}(\xi)| \cdot |\bar{Z}_{jm}(\zeta)| \cdot \\ &\quad \left| P_\theta(X_{\Delta_i} = \xi, X_{\Delta_j} = \zeta) - P_\theta(X_{\Delta_i} = \xi) P_\theta(X_{\Delta_j} = \zeta) \right|. \end{aligned} \quad (\text{A.2.6})$$

Since the components of \bar{Z}_i are bounded, by Lemma A.1.2 and for positive constants C_2 and c_1 , we have

$$\begin{aligned}
&\leq C_2 \sum_{\xi \in \bar{\Omega}_{\Delta_i}} \sum_{\zeta \in \bar{\Omega}_{\Delta_j}} \left| P_\theta(X_{\Delta_i} = \xi, X_{\Delta_j} = \zeta) - P_\theta(X_{\Delta_i} = \xi) P_\theta(X_{\Delta_j} = \zeta) \right| \\
&\leq C_2 |\Omega_{\Delta_i}| |\Omega_{\Delta_j}| |\Delta_i| \exp\{-c_1 d(\Delta_i, \Delta_j)\} \\
&\leq C_3 \exp\{-c_1 d(\Delta_i, \Delta_j)\}
\end{aligned} \tag{A.2.7}$$

for some $C_3 > 0$. Hence, as we return to (A.2.4), we see that (A.2.7) implies

$$\begin{aligned}
E_\theta \left(\left\| \sum_{i \in \Lambda_n} \bar{Z}_i \right\|^2 \right) &\leq C_1 |\Lambda_n| + C_3 \sum_{m=1}^k \sum_{\substack{i, j \in \Lambda_n \\ i \neq j}} \exp\{-c_1 d(\Delta_i, \Delta_j)\} \\
&\leq C_1 |\Lambda_n| + C_4 \sum_{\substack{i, j \in \Lambda_n \\ i \neq j}} \exp\{-c_1 d(\Delta_i, \Delta_j)\}
\end{aligned} \tag{A.2.8}$$

for some $C_4 > 0$. Taking a look at the exponential sum, we find that for distances fixed at 1,

$$\begin{aligned}
\sum_{\substack{i, j \in \Lambda_n \\ d(\Delta_i, \Delta_j) = 1}} \exp\{-c_1\} &\leq \sum_{\substack{i, j \in \Lambda_n \\ \|i-j\| = 1}} \exp\{-c_1\} \\
&= e^{-c_1} 4n(n-1) \\
&= O(|\Lambda_n|).
\end{aligned} \tag{A.2.9}$$

Indeed, it should be clear not only that such a sum over any other fixed distance will have fewer terms than the above sum for fixed distance 1, but also that the additive terms will have exponential decay rates as the distance increases. Hence,

$$E_\theta \left(\left\| \sum_{i \in \Lambda_n} \bar{Z}_i \right\|^2 \right) = O(|\Lambda_n|) \tag{A.2.10}$$

Getting back to the business at hand, we must find $\nabla^2 b_n(\vartheta)$. Begin with the expression for $\nabla b_n(\vartheta)$ given in the first line of (2.1.14):

$$\begin{aligned}
\nabla^2 b_n(\vartheta) &= \nabla(\nabla b_n(\vartheta)) \\
&= \sum_{y \in \Omega_{\Lambda_n}} Y_n(y) \cdot \\
&\quad \left(\frac{\left[\sum_{z \in \Omega_{\Lambda_n}} \exp\{|\Lambda_n| \vartheta^T Y_n(z)\} \right] \cdot |\Lambda_n| Y_n^T(y) \exp\{|\Lambda_n| \vartheta^T Y_n(y)\}}{\left[\sum_{w \in \Omega_{\Lambda_n}} \exp\{|\Lambda_n| \vartheta^T Y_n(w)\} \right]^2} \right. \\
&\quad \left. - \frac{\exp\{|\Lambda_n| \vartheta^T Y_n(y)\} \cdot \left[\sum_{z \in \Omega_{\Lambda_n}} |\Lambda_n| Y_n^T(z) \exp\{|\Lambda_n| \vartheta^T Y_n(z)\} \right]}{\left[\sum_{w \in \Omega_{\Lambda_n}} \exp\{|\Lambda_n| \vartheta^T Y_n(w)\} \right]^2} \right) \\
&= |\Lambda_n| \left(\sum_{y \in \Omega_{\Lambda_n}} Y_n(y) Y_n^T(y) \frac{\exp\{|\Lambda_n| \vartheta^T Y_n(y)\}}{\sum_{w \in \Omega_{\Lambda_n}} \exp\{|\Lambda_n| \vartheta^T Y_n(w)\}} \right. \\
&\quad \left. - \left[\sum_{y \in \Omega_{\Lambda_n}} Y_n(y) \frac{\exp\{|\Lambda_n| \vartheta^T Y_n(y)\}}{\sum_{z \in \Omega_{\Lambda_n}} \exp\{|\Lambda_n| \vartheta^T Y_n(z)\}} \right] \cdot \left[\sum_{z \in \Omega_{\Lambda_n}} Y_n^T(z) \frac{\exp\{|\Lambda_n| \vartheta^T Y_n(z)\}}{\sum_{w \in \Omega_{\Lambda_n}} \exp\{|\Lambda_n| \vartheta^T Y_n(w)\}} \right] \right) \\
&= |\Lambda_n| \left[E_\vartheta(Y_n Y_n^T | \tilde{\mathcal{X}}_{\partial \Lambda_n}) - E_\vartheta(Y_n | \tilde{\mathcal{X}}_{\partial \Lambda_n}) E_\vartheta(Y_n^T | \tilde{\mathcal{X}}_{\partial \Lambda_n}) \right]. \tag{A.2.11}
\end{aligned}$$

Write (A.2.11) in terms of \bar{Z}_i 's:

$$\begin{aligned}
&|\Lambda_n| \left[E_\vartheta(Y_n Y_n^T | \tilde{\mathcal{X}}_{\partial \Lambda_n}) - E_\vartheta(Y_n | \tilde{\mathcal{X}}_{\partial \Lambda_n}) E_\vartheta(Y_n^T | \tilde{\mathcal{X}}_{\partial \Lambda_n}) \right] \\
&= |\Lambda_n| \left(E_\vartheta \left[\left(\frac{1}{|\Lambda_n|} \sum_{i \in \Lambda_n} Z_i \right) \left(\frac{1}{|\Lambda_n|} \sum_{i \in \Lambda_n} Z_i \right)^T \middle| \tilde{\mathcal{X}}_{\partial \Lambda_n} \right] \right. \\
&\quad \left. - E_\vartheta \left[\frac{1}{|\Lambda_n|} \sum_{i \in \Lambda_n} Z_i \middle| \tilde{\mathcal{X}}_{\partial \Lambda_n} \right] E_\vartheta \left[\left(\frac{1}{|\Lambda_n|} \sum_{i \in \Lambda_n} Z_i \right)^T \middle| \tilde{\mathcal{X}}_{\partial \Lambda_n} \right] \right)
\end{aligned}$$

$$\begin{aligned}
&= \frac{1}{|\Lambda_n|} \left(E_\vartheta \left[\left(\sum_{i \in \Lambda_n} \{Z_i - E_\vartheta[Z_i | \tilde{x}_{\partial\Lambda_n}]\} \right) \left(\sum_{i \in \Lambda_n} \{Z_i - E_\vartheta[Z_i | \tilde{x}_{\partial\Lambda_n}]\} \right)^\top \middle| \tilde{x}_{\partial\Lambda_n} \right] \right. \\
&\quad + E_\vartheta \left[\left(\sum_{i \in \Lambda_n} Z_i \right) E_\vartheta \left[\left(\sum_{i \in \Lambda_n} Z_i \right)^\top \middle| \tilde{x}_{\partial\Lambda_n} \right] \middle| \tilde{x}_{\partial\Lambda_n} \right] \\
&\quad + E_\vartheta \left[E_\vartheta \left[\left(\sum_{i \in \Lambda_n} Z_i \right) \middle| \tilde{x}_{\partial\Lambda_n} \right] \left(\sum_{i \in \Lambda_n} Z_i \right)^\top \middle| \tilde{x}_{\partial\Lambda_n} \right] \\
&\quad - E_\vartheta \left(E_\vartheta \left[\left(\sum_{i \in \Lambda_n} Z_i \right) \middle| \tilde{x}_{\partial\Lambda_n} \right] E_\vartheta \left[\left(\sum_{i \in \Lambda_n} Z_i \right)^\top \middle| \tilde{x}_{\partial\Lambda_n} \right] \middle| \tilde{x}_{\partial\Lambda_n} \right) \\
&\quad - E_\vartheta \left[\sum_{i \in \Lambda_n} \{Z_i - E_\vartheta[Z_i | \tilde{x}_{\partial\Lambda_n}]\} \middle| \tilde{x}_{\partial\Lambda_n} \right] E_\vartheta \left[\left(\sum_{i \in \Lambda_n} \{Z_i - E_\vartheta[Z_i | \tilde{x}_{\partial\Lambda_n}]\} \right)^\top \middle| \tilde{x}_{\partial\Lambda_n} \right] \\
&\quad - E_\vartheta \left[\sum_{i \in \Lambda_n} Z_i \middle| \tilde{x}_{\partial\Lambda_n} \right] E_\vartheta \left[\left(\sum_{i \in \Lambda_n} E_\vartheta[Z_i | \tilde{x}_{\partial\Lambda_n}] \right)^\top \middle| \tilde{x}_{\partial\Lambda_n} \right] \\
&\quad - E_\vartheta \left[\sum_{i \in \Lambda_n} E_\vartheta[Z_i | \tilde{x}_{\partial\Lambda_n}] \middle| \tilde{x}_{\partial\Lambda_n} \right] E_\vartheta \left[\left(\sum_{i \in \Lambda_n} Z_i \right)^\top \middle| \tilde{x}_{\partial\Lambda_n} \right] \\
&\quad + E_\vartheta \left[\sum_{i \in \Lambda_n} E_\vartheta[Z_i | \tilde{x}_{\partial\Lambda_n}] \middle| \tilde{x}_{\partial\Lambda_n} \right] E_\vartheta \left[\left(\sum_{i \in \Lambda_n} E_\vartheta[Z_i | \tilde{x}_{\partial\Lambda_n}] \right)^\top \middle| \tilde{x}_{\partial\Lambda_n} \right] \Big) \\
&= \frac{1}{|\Lambda_n|} E_\vartheta \left[\left(\sum_{i \in \Lambda_n} \bar{Z}_i \right) \left(\sum_{i \in \Lambda_n} \bar{Z}_i \right)^\top \middle| \tilde{x}_{\partial\Lambda_n} \right]. \tag{A.2.12}
\end{aligned}$$

Hence, for some unit vector $v \in \mathbf{R}^k$ and using the expressions in (A.2.9) and (A.2.12),

$$\begin{aligned}
v^\top \nabla^2 b(\vartheta) v &= \frac{1}{|\Lambda_n|} E_\vartheta \left[\left\| v \sum_{i \in \Lambda_n} \bar{Z}_i \right\|^2 \middle| \tilde{x}_{\partial\Lambda_n} \right] \\
&= \frac{1}{|\Lambda_n|} O(|\Lambda_n|) \\
&\leq C \tag{A.2.13}
\end{aligned}$$

for some $C > 0$ when n is large.

On the left-hand side, under the Uniqueness Condition 1.3.1, the family of continuous functions (in ϑ) given by $\{\nabla^2 b_n(\vartheta): \tilde{x}_{\partial\Lambda_n} \in \Omega_{\partial\Lambda_n}, n \in \mathbb{Z}^+\}$, whose form is given in (A.2.11) above, converges to $B(\vartheta)$, given in (2.1.13), as $n \rightarrow \infty$ uniformly for ϑ in a neighborhood of θ . Hence the Positivity Condition 2.1.2 implies that $v^T \nabla^2 b_n(\vartheta) v \geq c$ for some $c > 0$. \square

Lemma 2.1.2 *Under the Identifiability Condition 2.1.1, for every $\varepsilon > 0$ there exist some $c, C > 0$ such that*

$$P_\theta(\|\hat{\theta}_n - \theta\| > \varepsilon) \leq C \exp(-c|\Lambda_n|) \quad (2.1.18)$$

uniformly for all large n , where $\|\cdot\|$ is the Euclidian norm.

Proof: See Cométs (1992). \square

Lemma 2.2.1 *For every fixed $\vartheta \in \Theta$, we have*

$$\ell_L(x(n); \vartheta) \rightarrow \log \mathbf{L}(x(n), \vartheta) + |\Lambda_n| b_n(\psi), \quad (2.2.8)$$

$$\nabla \ell_L(x(n); \vartheta) \rightarrow \nabla \log \mathbf{L}(x(n), \vartheta), \quad (2.2.9)$$

and

$$\nabla^2 \ell_L(x(n); \vartheta) \rightarrow \nabla^2 \log \mathbf{L}(x(n), \vartheta), \quad (2.2.10)$$

all \mathbf{P} -a.s. as $L \rightarrow \infty$.

Proof: It has already been noted that for fixed $\vartheta \in \Theta$, $r_L(\vartheta) \rightarrow r(\vartheta)$ \mathbf{P} -a.s. as $L \rightarrow \infty$, because of the ergodicity of the Markov chain. Using this fact on (2.2.7), we get

$$\begin{aligned} \ell_L(x(n), \vartheta) &= |\Lambda_n| \vartheta^T Y(\tilde{x}_{\Lambda_n}) - \log r_L(\vartheta) \\ &\rightarrow |\Lambda_n| \vartheta^T Y(\tilde{x}_{\Lambda_n}) - \log r(\vartheta) \end{aligned} \quad (A.2.14)$$

\mathbf{P} -a.s. as $L \rightarrow \infty$. From (2.2.1) and (2.2.4),

$$= \log \mathbf{L}(x(n), \vartheta) + \log c(\psi), \quad (A.2.15)$$

and by (2.1.4) plus (2.2.1), we have

$$= \log \mathcal{L}(x(n), \vartheta) + |\Lambda_n| b_n(\psi). \quad (\text{A.2.16})$$

Thus the convergence in (2.2.8) is established.

As in Section 2.1, let ∇ denote the gradient operator with respect to ϑ , and ∇^2 denote the Hessian operator with respect to ϑ . Now,

$$\begin{aligned} \nabla \log \mathcal{L}(x(n), \vartheta) &= \nabla \left\{ |\Lambda_n| \vartheta^T Y(\tilde{x}_{\Lambda_n}) - \log c(\vartheta) \right\} \\ &= |\Lambda_n| Y(\tilde{x}_{\Lambda_n}) - \frac{\nabla c(\vartheta)}{c(\vartheta)}, \end{aligned} \quad (\text{A.2.17})$$

where, using the same sort of manipulations as were used to get (2.2.3),

$$\begin{aligned} \nabla c(\vartheta) &= \nabla \left[\sum_{y \in \Omega_{\Lambda_n}} \exp \left\{ |\Lambda_n| \vartheta^T Y(y) \right\} \right] \\ &= \sum_{y \in \Omega_{\Lambda_n}} |\Lambda_n| Y(y) \exp \left\{ |\Lambda_n| \vartheta^T Y(y) \right\} \\ &= c(\psi) E_\psi \left[|\Lambda_n| Y_n \exp \left\{ |\Lambda_n| (\vartheta - \psi)^T Y_n \right\} \middle| \tilde{x}_{\partial \Lambda_n} \right]. \end{aligned} \quad (\text{A.2.18})$$

Using the ergodicity of the Markov chain, we have

$$\begin{aligned} \nabla \ell_L(x(n), \vartheta) &= \nabla \left\{ |\Lambda_n| \vartheta^T Y(\tilde{x}_{\Lambda_n}) - \log r_L(\vartheta) \right\} \\ &= |\Lambda_n| Y(\tilde{x}_{\Lambda_n}) - \frac{\nabla r_L(\vartheta)}{r_L(\vartheta)} \\ &= |\Lambda_n| Y(\tilde{x}_{\Lambda_n}) - \frac{\nabla \left[\frac{1}{L} \sum_{l=1}^L \exp \left\{ |\Lambda_n| (\vartheta - \psi)^T Y(X^{(l)}(n)) \right\} \right]}{r_L(\vartheta)} \\ &= |\Lambda_n| Y(\tilde{x}_{\Lambda_n}) - \frac{\frac{1}{L} \sum_{l=1}^L |\Lambda_n| Y(X^{(l)}(n)) \exp \left\{ |\Lambda_n| (\vartheta - \psi)^T Y(X^{(l)}(n)) \right\}}{r_L(\vartheta)} \\ &\rightarrow |\Lambda_n| Y(\tilde{x}_{\Lambda_n}) - \frac{E_\psi \left[|\Lambda_n| Y_n \exp \left\{ |\Lambda_n| (\vartheta - \psi)^T Y_n \right\} \middle| \tilde{x}_{\partial \Lambda_n} \right]}{r(\vartheta)} \\ &= |\Lambda_n| Y(\tilde{x}_{\Lambda_n}) - \frac{\nabla c(\vartheta)}{c(\vartheta)} \end{aligned}$$

$$= \nabla \log \mathcal{L}(\mathbf{x}(n), \vartheta). \quad (\text{A.2.19})$$

Thus the convergence in (2.2.9) is established.

Finally, we begin verifying the convergence in (2.2.10):

$$\begin{aligned} \nabla^2 \log \mathcal{L}(\mathbf{x}(n), \vartheta) &= \nabla^2 \left\{ |\Lambda_n| \vartheta^T Y(\tilde{\mathbf{x}}_{\Lambda_n}) - \log c(\vartheta) \right\} \\ &= \nabla \left\{ |\Lambda_n| Y(\tilde{\mathbf{x}}_{\Lambda_n}) - \frac{\nabla c(\vartheta)}{c(\vartheta)} \right\} \\ &= \frac{c(\vartheta) \nabla^2 c(\vartheta) - [\nabla c(\vartheta)] [\nabla c(\vartheta)]^T}{[c(\vartheta)]^2}, \end{aligned} \quad (\text{A.2.20})$$

where, using the same sort of manipulations as were used to get (2.2.3),

$$\begin{aligned} \nabla^2 c(\vartheta) &= \nabla^2 \left[\sum_{\mathbf{y} \in \Omega_{\Lambda_n}} \exp \left\{ |\Lambda_n| \vartheta^T Y(\mathbf{y}) \right\} \right] \\ &= \nabla \left[\sum_{\mathbf{y} \in \Omega_{\Lambda_n}} |\Lambda_n| Y(\mathbf{y}) \exp \left\{ |\Lambda_n| \vartheta^T Y(\mathbf{y}) \right\} \right] \\ &= \sum_{\mathbf{y} \in \Omega_{\Lambda_n}} |\Lambda_n|^2 Y(\mathbf{y}) Y^T(\mathbf{y}) \exp \left\{ |\Lambda_n| \vartheta^T Y(\mathbf{y}) \right\} \\ &= c(\psi) E_{\psi} \left[|\Lambda_n|^2 Y_n Y_n^T \exp \left\{ |\Lambda_n| (\vartheta - \psi)^T Y_n \right\} \middle| \tilde{\mathbf{x}}_{\vartheta \Lambda_n} \right]. \end{aligned} \quad (\text{A.2.21})$$

Using the ergodicity of the Markov chain, we have

$$\begin{aligned} \nabla^2 \ell_L(\mathbf{x}(n), \vartheta) &= \nabla^2 \left\{ |\Lambda_n| \vartheta^T Y(\tilde{\mathbf{x}}_{\Lambda_n}) - \log r_L(\vartheta) \right\} \\ &= \nabla \left\{ |\Lambda_n| Y(\tilde{\mathbf{x}}_{\Lambda_n}) - \frac{\nabla r_L(\vartheta)}{r_L(\vartheta)} \right\} \\ &= \frac{r_L(\vartheta) \nabla^2 r_L(\vartheta) - [\nabla r_L(\vartheta)] [\nabla r_L(\vartheta)]^T}{[r_L(\vartheta)]^2} \\ &= \frac{r_L(\vartheta) \nabla^2 \left[\frac{1}{L} \sum_{l=1}^L \exp \left\{ |\Lambda_n| (\vartheta - \psi)^T Y(X^{(l)}(n)) \right\} \right]}{[r_L(\vartheta)]^2} \end{aligned}$$

$$\begin{aligned}
& \frac{\left[\nabla \left(\frac{1}{L} \sum_{l=1}^L \exp \left\{ |\Lambda_n| (\vartheta - \psi)^T Y(X^{(l)}(n)) \right\} \right) \right] \left[\nabla \left(\frac{1}{L} \sum_{l=1}^L \exp \left\{ |\Lambda_n| (\vartheta - \psi)^T Y(X^{(l)}(n)) \right\} \right) \right]^T}{[r_L(\vartheta)]^2} \\
&= \frac{r_L(\vartheta) \nabla \left[\frac{1}{L} \sum_{l=1}^L |\Lambda_n| Y(X^{(l)}(n)) \exp \left\{ |\Lambda_n| (\vartheta - \psi)^T Y(X^{(l)}(n)) \right\} \right]}{[r_L(\vartheta)]^2} \\
&\quad - \frac{\left[\frac{1}{L} \sum_{l=1}^L |\Lambda_n| Y(X^{(l)}(n)) \exp \left\{ |\Lambda_n| (\vartheta - \psi)^T Y(X^{(l)}(n)) \right\} \right]}{[r_L(\vartheta)]^2} \\
&\quad \cdot \frac{\left[\frac{1}{L} \sum_{l=1}^L |\Lambda_n| Y(X^{(l)}(n)) \exp \left\{ |\Lambda_n| (\vartheta - \psi)^T Y(X^{(l)}(n)) \right\} \right]^T}{[r_L(\vartheta)]^2} \\
&= \frac{r_L(\vartheta) \left[\frac{1}{L} \sum_{l=1}^L |\Lambda_n|^2 Y(X^{(l)}(n)) Y^T(X^{(l)}(n)) \exp \left\{ |\Lambda_n| (\vartheta - \psi)^T Y(X^{(l)}(n)) \right\} \right]}{[r_L(\vartheta)]^2} \\
&\quad - \frac{\left[\frac{1}{L} \sum_{l=1}^L |\Lambda_n| Y(X^{(l)}(n)) \exp \left\{ |\Lambda_n| (\vartheta - \psi)^T Y(X^{(l)}(n)) \right\} \right]}{[r_L(\vartheta)]^2} \\
&\quad \cdot \frac{\left[\frac{1}{L} \sum_{l=1}^L |\Lambda_n| Y(X^{(l)}(n)) \exp \left\{ |\Lambda_n| (\vartheta - \psi)^T Y(X^{(l)}(n)) \right\} \right]^T}{[r_L(\vartheta)]^2} \\
&\rightarrow \frac{r(\vartheta) E_\psi \left[|\Lambda_n|^2 Y_n Y_n^T \exp \left\{ |\Lambda_n| (\vartheta - \psi)^T Y_n \right\} \middle| \tilde{x}_{\partial \Lambda_n} \right]}{[r(\vartheta)]^2} \\
&\quad - \frac{\left(E_\psi \left[|\Lambda_n| Y_n \exp \left\{ |\Lambda_n| (\vartheta - \psi)^T Y_n \right\} \middle| \tilde{x}_{\partial \Lambda_n} \right] \right) \left(E_\psi \left[|\Lambda_n| Y_n \exp \left\{ |\Lambda_n| (\vartheta - \psi)^T Y_n \right\} \middle| \tilde{x}_{\partial \Lambda_n} \right] \right)^T}{[r(\vartheta)]^2} \\
&= \frac{c(\vartheta) \nabla^2 c(\vartheta) - [\nabla c(\vartheta)] [\nabla c(\vartheta)]^T}{[c(\vartheta)]^2}
\end{aligned}$$

$$= \nabla^2 \log \mathcal{L}(x(n), \vartheta). \quad (\text{A.2.22})$$

Thus the convergence in (2.2.10) is established. \square

Lemma 2.2.2 *For every uniformly bounded function f on Ω_{Λ_n} , the family*

$$\mathbf{F} = \left\{ \sum_{j=1}^L f \nu_{jL}(\cdot), L \in \mathbf{N} \right\} \quad (\text{2.2.12})$$

has a subsequence which is uniformly convergent on η_θ .

Proof: Note first that \mathbf{F} is a uniformly bounded sequence.

The gradient of ν_{jL} with respect to ϑ is

$$\nabla \nu_{jL}(\vartheta) = |\Lambda_n| \nu_{jL}(\vartheta) \left[Y(X^{(j)}\lambda_n) - \sum_{l=1}^L Y(X^{(l)}\lambda_n) \nu_{lL}(\vartheta) \right], \quad (\text{A.2.23})$$

so that

$$\nabla \left[\sum_{j=1}^L f \nu_{jL}(\vartheta) \right] = \quad (\text{A.2.24})$$

$$|\Lambda_n| \left\{ \sum_{j=1}^L f Y(X^{(j)}\lambda_n) \nu_{jL}(\vartheta) - \left[\sum_{j=1}^L f \nu_{jL}(\vartheta) \right] \left[\sum_{j=1}^L Y(X^{(j)}\lambda_n) \nu_{jL}(\vartheta) \right] \right\}.$$

(A.2.24) is uniformly bounded in norm since $Y(\cdot)$ is uniformly bounded in norm (since, in fact, the pair-potentials which make up $Y(\cdot)$ are so). \mathbf{F} is thus equicontinuous (Royden, 1988, p. 167), and the desired result follows from the Ascoli-Arzelá Theorem (Royden, 1988, p. 169). \square

Lemma 2.2.3 *Each of the families $\mathbf{F}_1 = \{\ell_L(x(n), \cdot), L \in \mathbf{N}\}$, $\mathbf{F}_2 = \{\nabla \ell_L(x(n), \cdot), L \in \mathbf{N}\}$, and $\mathbf{F}_3 = \{\nabla^2 \ell_L(x(n), \cdot), L \in \mathbf{N}\}$ has a subsequence which is uniformly convergent on η_θ .*

Proof: Note first that

$$\nabla \ell_L(x(n), \vartheta) = |\Lambda_n| Y_n - \sum_{j=1}^L |\Lambda_n| Y(X^{(j)}\lambda_n) \nu_{jL}(\vartheta) \quad (\text{A.2.25})$$

and

$$\begin{aligned} \nabla^2 \ell_L(x(n), \vartheta) &= \sum_{j=1}^L |\Lambda_n|^2 Y(X^{(j)\lambda(n)}) Y^\top(X^{(j)\lambda(n)}) \nu_{jL}(\vartheta) \\ &\quad - \left[\sum_{j=1}^L |\Lambda_n| Y(X^{(j)\lambda(n)}) \nu_{jL}(\vartheta) \right] \left[\sum_{j=1}^L |\Lambda_n| Y(X^{(j)\lambda(n)}) \nu_{jL}(\vartheta) \right]^\top. \end{aligned} \quad (\text{A.2.26})$$

Hence, by Lemma 2.2.2, both \mathbf{F}_2 and \mathbf{F}_3 have uniformly convergent subsequences. In addition, \mathbf{F}_1 is uniformly bounded, and it is equicontinuous since \mathbf{F}_2 is uniformly bounded. By the Ascoli-Arzelá Theorem, \mathbf{F}_1 also has a uniformly convergent subsequence (Royden, 1988). \square

Lemma 2.2.4 *Each of the sequences given in Lemma 2.2.1 are uniformly convergent for each $\vartheta \in \eta_\theta$, \mathbf{P} -a.s. as $L \rightarrow \infty$.*

Proof: Since a countable union of \mathbf{P} -null sets is still a \mathbf{P} -null set, each of (2.2.8), (2.2.9), and (2.2.10) from Lemma 2.2.1 hold (pointwise but not uniformly) for ϑ in a countable dense subset of η_θ , \mathbf{P} -a.s. as $L \rightarrow \infty$. Also, since each of the limits in (2.2.8), (2.2.9), and (2.2.10) are continuous functions on η_θ , it follows from Lemma 2.2.3 that these convergences hold uniformly for each $\vartheta \in \eta_\theta$, \mathbf{P} -a.s. as $L \rightarrow \infty$. \square

Lemma 2.3.1 *There exist positive constants λ , C , and c such that*

$$P_\theta \left(\frac{N_n(\varsigma \oplus \eta)}{|\Lambda_n|} < \lambda \right) \leq C \exp(-cn) \quad (2.3.21)$$

uniformly for all large n and all $\varsigma \oplus \eta \in \Omega_{\Lambda_{2R+1}}$.

Proof: Assume without loss of generality that $(3R+1)$ divides n . Partition Λ_n as a union of disjoint tiles: $\Lambda_n = \bigcup_{t=1}^T D_t$, so that each tile D_t is a $(3R+1) \times (3R+1)$ square lattice. Then $T = \left(\frac{n}{3R+1} \right)^2$.

Also, write the decomposition $\Lambda_n = \bigcup_{k=1}^{(3R+1)^2} G_k$, where every G_k contains exactly T pixels with the same relative positions in the disjoint tiles D_t , $t = 1, \dots, T$. For instance, one G_k may consist of the centers of the T tiles, while another G_k may consist of all upper-left corners of the T tiles. In the spirit of (2.3.19) and (2.3.20), we may write

$$N_n(\varsigma \oplus \eta) = \sum_{k=1}^{(3R+1)^2} \sum_{i \in G_k} \mathbb{1}_i(\varsigma \oplus \eta) \quad (\text{A.2.27})$$

so that

$$\begin{aligned} P_\theta \left(\frac{N_n(\varsigma \oplus \eta)}{|\Lambda_n|} < \lambda \right) &= P_\theta \left(\frac{1}{|\Lambda_n|} \sum_{k=1}^{(3R+1)^2} \sum_{i \in G_k} \mathbb{1}_i(\varsigma \oplus \eta) < \lambda \right) \\ &\leq \sum_{k=1}^{(3R+1)^2} P_\theta \left(\frac{1}{|\Lambda_n|} \sum_{i \in G_k} \mathbb{1}_i(\varsigma \oplus \eta) < \lambda \right). \end{aligned} \quad (\text{A.2.28})$$

Note that $|\Lambda_n| = n^2$. Using the Markov inequality, for every $\varsigma \oplus \eta \in \Omega_{\Lambda_{2R}}$ we may further write

$$\begin{aligned} &= \sum_{k=1}^{(3R+1)^2} P_\theta \left(\exp \left\{ -\frac{1}{n} \sum_{i \in G_k} \mathbb{1}_i(\varsigma \oplus \eta) \right\} > \exp(-\lambda n) \right) \\ &\leq \exp(-\lambda n) \sum_{k=1}^{(3R+1)^2} E_\theta \left[\exp \left\{ -\frac{1}{n} \sum_{i \in G_k} \mathbb{1}_i(\varsigma \oplus \eta) \right\} \right] \end{aligned} \quad (\text{A.2.29})$$

Recall that $\Lambda_{i,R}$ is the $(2R+1) \times (2R+1)$ region centered at site i . For a fixed index k , let $\mathbf{c}_k = \mathbf{Z}^2 \setminus \left(\bigcup_{i \in G_k} \Lambda_{i,R} \right)$ be the corridor dividing the regions $\Lambda_{i,R}$, $i \in G_k$. Hence Lemma A.1.1 may be applied to the expectations in (A.2.29):

$$\begin{aligned} E_\theta \left[\exp \left\{ -\frac{1}{n} \sum_{i \in G_k} \mathbb{1}_i(\varsigma \oplus \eta) \right\} \right] &= E_\theta \left(E_\theta \left[\exp \left\{ -\frac{1}{n} \sum_{i \in G_k} \mathbb{1}_i(\varsigma \oplus \eta) \right\} \middle| X_{\mathbf{c}_k} \right] \right) \\ &= E_\theta \left(\prod_{i \in G_k} E_\theta \left[\exp \left\{ -\frac{1}{n} \mathbb{1}_i(\varsigma \oplus \eta) \right\} \middle| X_{\mathbf{c}_k} \right] \right). \end{aligned} \quad (\text{A.2.30})$$

Furthermore, for every $\mathbf{x}_{\mathbf{c}_k}$, we have the Taylor expansion

$$\begin{aligned} E_\theta \left[\exp \left\{ -\frac{1}{n} \mathbb{1}_i(\varsigma \oplus \eta) \right\} \middle| \mathbf{x}_{\mathbf{c}_k} \right] &= 1 - \frac{1}{n} E_\theta \left[\mathbb{1}_i(\varsigma \oplus \eta) \middle| \mathbf{x}_{\mathbf{c}_k} \right] + o\left(\frac{1}{n}\right) \\ &\leq 1 - \frac{c_1}{n} \end{aligned} \quad (\text{A.2.31})$$

for some $c_1 > 0$ and all large n . Therefore, resuming the train of thought in (A.2.29),

$$P_\theta \left(\frac{N_n(\varsigma \oplus \eta)}{|\Lambda_n|} < \lambda \right) \leq \exp(\lambda n) \sum_{k=1}^{(3R+1)^2} \prod_{i \in G_k} \left(1 - \frac{c_1}{n} \right)$$

$$\begin{aligned}
&\leq \exp(\lambda n) \sum_{k=1}^{(3R+1)^2} \left(1 - \frac{c_1}{n}\right)^{\frac{n^2}{(3R+1)^2}} \\
&\leq \exp(\lambda n) (3R+1)^2 \exp\left(-\frac{c_1 n^2}{n(3R+1)^2}\right) \\
&\leq C \exp[-(c_2 - \lambda)n]
\end{aligned} \tag{A.2.32}$$

for some $C > 0$ and $0 < \lambda < c_2$. Hence for some $c > 0$,

$$P_\theta\left(\frac{N_n(\varsigma \oplus \eta)}{|\Lambda_n|} < \lambda\right) \leq C \exp(-cn). \tag{A.2.33}$$

□

Lemma 2.3.2 *Under the Identifiability Condition 2.1.1 and on the set \mathfrak{S}_n , there exist $c, C > 0$ such that*

$$c \leq v^\top \nabla^2 g_n(\vartheta) v \leq C \tag{2.3.23}$$

uniformly for all unit vectors $v \in \mathbb{R}^k$, all ϑ in a neighborhood of θ , and all large n .

Proof: In this proof, we employ some of the notions presented in the lemmas of Geman and Graffigne (1986).

Define $K(\vartheta, n) = \vartheta^\top V_n - g_n(\vartheta)$, which is part of the “exponential family” form of the pseudo-likelihood (2.3.2), written as a function of the parameter space. Intuitively, $K(\vartheta, n)$ may be viewed as an accounting of local characteristics in two different ways. The first is derived from the definition of pseudo-likelihood (2.3.1):

$$\begin{aligned}
K(\vartheta, n) &= \frac{1}{|\Lambda_n|} \log \mathfrak{P} \mathfrak{L}(x(n), \vartheta) \\
&= \frac{1}{|\Lambda_n|} \sum_{i \in \Lambda_n} \log p_i(x_i | x_{\mathcal{N}_i}; \vartheta),
\end{aligned} \tag{A.2.34}$$

where $p_i(x_i | x_{\mathcal{N}_i}; \vartheta)$ is written to denote $P_\vartheta(X_i = x_i | X_{\mathcal{N}_i} = x_{\mathcal{N}_i})$. The second view of $K(\vartheta, n)$ may be given in terms of the actual configuration. For a configuration $\varsigma \oplus \eta \in \Omega_{\Lambda_{2R+1}}$ (where $\varsigma \in S$ is the value, or configuration, at the origin o , and $\eta \in \Omega_{\Lambda_{2R+1} \setminus \{o\}}$ is the configuration on the remainder of Λ_{2R+1}), recall that $N_n(\varsigma \oplus \eta)$ as given in (2.3.20) counts the number of such configurations in the region Λ_n . Let $p_o(\varsigma | \eta; \vartheta)$ be the local characteristic at the origin. Then,

via translation invariance, write

$$\begin{aligned} K(\vartheta, n) &= \frac{1}{|\Lambda_n|} \sum_{\eta} \sum_{\varsigma} N_n(\varsigma \oplus \eta) \log p_o(\varsigma | \eta; \vartheta) \\ &= \sum_{\eta} \frac{N_n(\eta)}{|\Lambda_n|} \sum_{\varsigma} \frac{N_n(\varsigma \oplus \eta)}{N_n(\eta)} \log p_o(\varsigma | \eta; \vartheta) \end{aligned} \quad (\text{A.2.35})$$

for all configurations $\varsigma \oplus \eta \in \Omega_{\Lambda_{2R+1}}$. Write the local characteristic at the origin in exponential family form:

$$p_o(\varsigma | \eta; \vartheta) = \frac{\exp\{\vartheta^T \phi(\varsigma \oplus \eta)\}}{\sum_s \exp\{\vartheta^T \phi(s \oplus \eta)\}}, \quad (\text{A.2.36})$$

where $\vartheta \in \mathbf{R}^k$ and $\phi(\cdot)$ is an appropriate vector-valued function.

Clearly, $\nabla^2 g_n(\vartheta) = -\nabla^2 K(\vartheta, n)$, so we begin the task of finding this expression by first finding $\nabla K(\vartheta, n)$:

$$\begin{aligned} \nabla K(\vartheta, n) &= \nabla \left(\sum_{\eta} \frac{N_n(\eta)}{|\Lambda_n|} \sum_{\varsigma} \frac{N_n(\varsigma \oplus \eta)}{N_n(\eta)} \log \frac{\exp\{\vartheta^T \phi(\varsigma \oplus \eta)\}}{\sum_s \exp\{\vartheta^T \phi(s \oplus \eta)\}} \right) \\ &= \sum_{\eta} \frac{N_n(\eta)}{|\Lambda_n|} \sum_{\varsigma} \frac{N_n(\varsigma \oplus \eta)}{N_n(\eta)} \left(\frac{\exp\{\vartheta^T \phi(\varsigma \oplus \eta)\}}{\sum_s \exp\{\vartheta^T \phi(s \oplus \eta)\}} \right)^{-1} \\ &\quad \left(\frac{\sum_s \exp\{\vartheta^T \phi(s \oplus \eta)\} \phi(\varsigma \oplus \eta) \exp\{\vartheta^T \phi(\varsigma \oplus \eta)\}}{\left(\sum_r \exp\{\vartheta^T \phi(r \oplus \eta)\} \right)^2} \right. \\ &\quad \left. - \frac{\exp\{\vartheta^T \phi(\varsigma \oplus \eta)\} \sum_t \phi(t \oplus \eta) \exp\{\vartheta^T \phi(t \oplus \eta)\}}{\left(\sum_r \exp\{\vartheta^T \phi(r \oplus \eta)\} \right)^2} \right) \\ &= \sum_{\eta} \frac{N_n(\eta)}{|\Lambda_n|} \sum_{\varsigma} \frac{N_n(\varsigma \oplus \eta)}{N_n(\eta)} \left(\phi(\varsigma \oplus \eta) - \sum_t \phi(t \oplus \eta) \frac{\exp\{\vartheta^T \phi(t \oplus \eta)\}}{\sum_r \exp\{\vartheta^T \phi(r \oplus \eta)\}} \right) \\ &= \sum_{\eta} \frac{N_n(\eta)}{|\Lambda_n|} \sum_{\varsigma} \frac{N_n(\varsigma \oplus \eta)}{N_n(\eta)} \left(\phi(\varsigma \oplus \eta) - E_{\vartheta}[\phi(X_o \oplus \eta) | \eta] \right). \end{aligned} \quad (\text{A.2.37})$$

where $E_{\vartheta}(\cdot | \eta)$ is the conditional expectation with respect to $p_o(\cdot | \eta; \vartheta)$. Hence

$$\begin{aligned}
\nabla^2 K(\vartheta, n) &= \sum_{\eta} \frac{N_n(\eta)}{|\Lambda_n|} \sum_{\varsigma} \frac{N_n(\varsigma \oplus \eta)}{N_n(\eta)} \\
&\quad \left\{ - \sum_t \phi(t \oplus \eta) \left(\frac{\sum_s \exp\{\vartheta^T \phi(s \oplus \eta)\} \phi^T(t \oplus \eta) \exp\{\vartheta^T \phi(t \oplus \eta)\}}{\left(\sum_r \exp\{\vartheta^T \phi(r \oplus \eta)\} \right)^2} \right. \right. \\
&\quad \left. \left. - \frac{\exp\{\vartheta^T \phi(t \oplus \eta)\} \sum_u \phi^T(u \oplus \eta) \exp\{\vartheta^T \phi(u \oplus \eta)\}}{\left(\sum_r \exp\{\vartheta^T \phi(r \oplus \eta)\} \right)^2} \right) \right\} \\
&= - \sum_{\eta} \frac{N_n(\eta)}{|\Lambda_n|} \sum_{\varsigma} \frac{N_n(\varsigma \oplus \eta)}{N_n(\eta)} \\
&\quad \left\{ \sum_t \phi(t \oplus \eta) \left(\phi^T(t \oplus \eta) - \sum_u \phi^T(u \oplus \eta) \frac{\exp\{\vartheta^T \phi(u \oplus \eta)\}}{\sum_r \exp\{\vartheta^T \phi(r \oplus \eta)\}} \right) \frac{\exp\{\vartheta^T \phi(t \oplus \eta)\}}{\sum_r \exp\{\vartheta^T \phi(r \oplus \eta)\}} \right\} \\
&= - \sum_{\eta} \frac{N_n(\eta)}{|\Lambda_n|} \sum_{\varsigma} \frac{N_n(\varsigma \oplus \eta)}{N_n(\eta)} \\
&\quad \left\{ \sum_t \phi(t \oplus \eta) \left(\phi^T(t \oplus \eta) - E_{\vartheta}[\phi^T(X_o \oplus \eta) | \eta] \right) \frac{\exp\{\vartheta^T \phi(t \oplus \eta)\}}{\sum_r \exp\{\vartheta^T \phi(r \oplus \eta)\}} \right\} \\
&= - \sum_{\eta} \frac{N_n(\eta)}{|\Lambda_n|} \sum_{\varsigma} \frac{N_n(\varsigma \oplus \eta)}{N_n(\eta)} \\
&\quad \left(E_{\vartheta}[\phi(X_o \oplus \eta) \phi^T(X_o \oplus \eta) | \eta] - E_{\vartheta}[\phi(X_o \oplus \eta) | \eta] E_{\vartheta}[\phi^T(X_o \oplus \eta) | \eta] \right) \\
&= - \sum_{\eta} \frac{N_n(\eta)}{|\Lambda_n|} \sum_{\varsigma} \frac{N_n(\varsigma \oplus \eta)}{N_n(\eta)} \\
&\quad E_{\vartheta} \left[\left(\phi(X_o \oplus \eta) - E_{\vartheta}[\phi(X_o \oplus \eta) | \eta] \right) \left(\phi(X_o \oplus \eta) - E_{\vartheta}[\phi(X_o \oplus \eta) | \eta] \right)^T \middle| \eta \right]
\end{aligned} \tag{A.2.38}$$

finally, since the expectation does not depend on ς , plus the fact that $\sum_{\varsigma} N_n(\varsigma \oplus \eta) = N_n(\eta)$ for

fixed η , we have

$$-\nabla^2 K(\vartheta, n) = \tag{A.2.39}$$

$$\sum_{\eta} \frac{N_n(\eta)}{|\Lambda_n|} E_{\vartheta} \left[\left(\phi(X_o \oplus \eta) - E_{\vartheta}[\phi(X_o \oplus \eta) | \eta] \right) \left(\phi(X_o \oplus \eta) - E_{\vartheta}[\phi(X_o \oplus \eta) | \eta] \right)^{\top} \middle| \eta \right]$$

Now we may investigate the quantity of interest. For $v \in \mathbb{R}^k$, we have

$$\begin{aligned} v^{\top} \nabla^2 g_n(\vartheta) v &= -v^{\top} \nabla^2 K(\vartheta, n) v \\ &= \sum_{\eta} \frac{N_n(\eta)}{|\Lambda_n|} E_{\vartheta} \left\{ \left\| v^{\top} \left(\phi(X_o \oplus \eta) - E_{\vartheta}[\phi(X_o \oplus \eta) | \eta] \right) \right\|^2 \middle| \eta \right\} \end{aligned} \tag{A.2.40}$$

The work leading up to (A.2.10) in Lemma 2.1.1 is just as valid for a single site, specifically the origin o . Therefore the expectation in (A.2.40) is $O(1)$, and

$$\begin{aligned} v^{\top} \nabla^2 g_n(\vartheta) v &= \sum_{\eta} \frac{N_n(\eta)}{|\Lambda_n|} O(1) \\ &\leq O(1) \sum_{\eta} 1 \\ &\leq C \end{aligned} \tag{A.2.41}$$

for some $C > 0$ since there are only finitely many $\eta \in \Omega_{\Lambda_{2R+1} \setminus \{o\}}$.

On the other hand, the Identifiability Condition 2.1.1 guarantees that the “outside” expectation in (A.2.40) is strictly positive for at least one of the η -configurations. Also, for fixed $\varsigma \in S$ and $\eta \in \Omega_{\Lambda_{2R+1} \setminus \{o\}}$, it should be clear that $N_n(\eta) \geq N_n(\varsigma \oplus \eta)$ (because there are more configurations with any value in S at the origin than there are with fixed ς at the origin). Then, on the event \mathfrak{S}_n given by (2.3.22),

$$\frac{N_n(\eta)}{|\Lambda_n|} \geq \frac{N_n(\varsigma \oplus \eta)}{|\Lambda_n|} \geq \lambda > 0 \tag{A.2.42}$$

for all η -configurations. Therefore, $c \leq v^{\top} \nabla^2 g_n(\vartheta) v$ for some $c > 0$. \square

Lemma 2.3.3 *Under the Identifiability Condition 2.1.1, for every $\varepsilon > 0$ there exist $c, C > 0$ such that*

$$P_{\theta} \left(\left\| \tilde{\theta}_n - \theta \right\| > \varepsilon \right) \leq C \exp(-c |\Lambda_n|) \tag{2.3.24}$$

uniformly for all large n , where $\|\cdot\|$ is the Euclidian norm.

Proof: See Cométs (1992). \square

Lemma 2.3.4 *Under the Identifiability Condition 2.1.1,*

$$E_{\theta} \left\{ \|\tilde{\theta}_n - \theta\|^2 \mathbf{1}_{\mathcal{S}_n} \right\} = O\left(\frac{1}{|\Lambda_n|}\right) \quad (2.3.25)$$

as $n \rightarrow \infty$.

Proof: Here, we go through several steps to find a simple condition to guarantee the desired result. We begin by working with the “exponential family” form $K(\vartheta, n) = \vartheta^T V_n - g_n(\vartheta)$.

The Taylor expansion of $K(\vartheta, n)$ about $\tilde{\theta}_n$ is given by

$$\begin{aligned} K(\vartheta, n) - K(\tilde{\theta}_n, n) &= (\vartheta - \tilde{\theta}_n)^T [V_n - \nabla g_n(\tilde{\theta}_n)] - \frac{1}{2}(\vartheta - \tilde{\theta}_n)^T \nabla^2 g_n(\vartheta') (\vartheta - \tilde{\theta}_n) \\ &= -\frac{1}{2}(\vartheta - \tilde{\theta}_n)^T \nabla^2 g_n(\vartheta') (\vartheta - \tilde{\theta}_n) \end{aligned} \quad (A.2.43)$$

for some ϑ' satisfying $\|\vartheta' - \tilde{\theta}_n\| < \|\vartheta - \tilde{\theta}_n\|$. Then, taking the gradient of (A.2.43), we get

$$\nabla K(\vartheta, n) = -\nabla^2 g_n(\vartheta') (\vartheta - \tilde{\theta}_n) \quad (A.2.44)$$

Evaluating (A.2.44) at the true parameter θ , gives

$$\nabla K(\theta, n) = -\nabla^2 g_n(\vartheta') (\theta - \tilde{\theta}_n) \quad (A.2.45)$$

for some ϑ' in a neighborhood of θ , so that

$$\|\nabla K(\theta, n)\|^2 = (\theta - \tilde{\theta}_n)^T [\nabla^2 g_n(\vartheta')]^2 (\theta - \tilde{\theta}_n). \quad (A.2.46)$$

For a symmetric matrix A with generic eigenvalue λ_A , for all unit vectors v of compatible dimension, and some constant $c_1 > 0$, we have

$$v^T A v \geq c_1 \Leftrightarrow \min_{v: \|v\|=1} v^T A v \geq c_1 \Leftrightarrow \min \lambda_A \geq c_1 \Leftrightarrow$$

$$\min \lambda_A^2 \geq c_1^2 \Leftrightarrow \min \lambda_{A^2} \geq c_1^2 \Leftrightarrow \min_{v: \|v\|=1} v^T A^2 v \geq c_1^2 \Leftrightarrow v^T A^2 v \geq c_1^2 \quad (\text{A.2.47})$$

for all v such that $\|v\| = 1$. Therefore, by the result in (A.2.47) and Lemma 2.3.2, we have

$$\|\nabla K(\theta, n)\|^2 \geq c \|\theta - \bar{\theta}_n\|^2 \quad (\text{A.2.48})$$

on \mathfrak{S}_n for some $c > 0$, so that

$$\|\bar{\theta}_n - \theta\|^2 \leq C \|\nabla K(\theta, n)\|^2 \quad (\text{A.2.49})$$

on \mathfrak{S}_n for some $C > 0$. Thus

$$E_\theta \left\{ \|\bar{\theta}_n - \theta\|^2 \mathbf{1}_{\mathfrak{S}_n} \right\} \leq C E_\theta \left\{ \|\nabla K(\theta, n)\|^2 \mathbf{1}_{\mathfrak{S}_n} \right\}, \quad (\text{A.2.50})$$

and it is enough to show

$$E_\theta \left\{ \|\nabla K_m(\theta, n)\|^2 \mathbf{1}_{\mathfrak{S}_n} \right\} = O\left(\frac{1}{|\Lambda_n|}\right). \quad (\text{A.2.51})$$

on \mathfrak{S}_n . In the same spirit used to write (A.2.35), we may rewrite $K(\vartheta, n)$ as

$$K(\theta, n) = \frac{1}{|\Lambda_n|} \sum_{i \in \Lambda_n} \left(\sum_{\zeta \oplus \eta} \mathbf{1}_{i(\zeta \oplus \eta)} \log p_o(\zeta | \eta; \theta) \right). \quad (\text{A.2.52})$$

Then, using the notions in (A.2.36) and (A.2.37), we obtain

$$\nabla K(\theta, n) = \frac{1}{|\Lambda_n|} \sum_{i \in \Lambda_n} W_i \quad (\text{A.2.53})$$

where W_i is the vector

$$W_i = \sum_{\eta} \mathbf{1}_{i(\eta)} \left(\phi(X_i \oplus \eta) - E_\theta[\phi(X_i \oplus \eta) | \eta] \right). \quad (\text{A.2.54})$$

Notice that for each $i \in \Lambda_n$, all of the components of W_i are bounded because the pair-interactions U_1 and U_2 in (1.3.10) which make up ϕ are bounded. Now, since

$$E_\theta \left\{ \|\nabla K(\theta, n)\|^2 \mathbf{1}_{\mathfrak{S}_n} \right\} = \frac{1}{|\Lambda_n|^2} E_\theta \left\{ \left\| \sum_{i \in \Lambda_n} W_i \right\|^2 \mathbf{1}_{\mathfrak{S}_n} \right\} \quad (\text{A.2.55})$$

it is enough to show that

$$E_{\theta} \left\{ \left\| \sum_{i \in \Lambda_n} W_i \right\|^2 1_{\mathfrak{G}_n} \right\} = \mathcal{O}(|\Lambda_n|). \quad (\text{A.2.56})$$

Let w_i denote a particular, arbitrary component of W_i . Then because the squared norm is the sum of a finite number of squared components, it is sufficient to show

$$E_{\theta} \left[\left(\sum_{i \in \Lambda_n} w_i \right)^2 1_{\mathfrak{G}_n} \right] = \mathcal{O}(|\Lambda_n|). \quad (\text{A.2.57})$$

Using the decompositions of Λ_n from the proof of Lemma 2.3.1, we have

$$\begin{aligned} E_{\theta} \left[\left(\sum_{i \in \Lambda_n} w_i \right)^2 1_{\mathfrak{G}_n} \right] &= E_{\theta} \left[\left(\sum_{k=1}^{(3R+1)^2} \sum_{i \in G_k} w_i \right)^2 1_{\mathfrak{G}_n} \right] \\ &= \sum_{k=1}^{(3R+1)^2} \sum_{l=1}^{(3R+1)^2} E_{\theta} \left[\left(\sum_{i \in G_k} w_i \right) \left(\sum_{i \in G_l} w_i \right) 1_{\mathfrak{G}_n} \right] \end{aligned} \quad (\text{A.2.58})$$

and, by the Cauchy-Schwarz inequality,

$$\begin{aligned} &\leq \sum_{k=1}^{(3R+1)^2} \sum_{l=1}^{(3R+1)^2} \sqrt{E_{\theta} \left[\left(\sum_{i \in G_k} w_i \right)^2 1_{\mathfrak{G}_n} \right] E_{\theta} \left[\left(\sum_{i \in G_l} w_i \right)^2 1_{\mathfrak{G}_n} \right]} \\ &\leq \left(\sum_{k=1}^{(3R+1)^2} \sqrt{E_{\theta} \left[\left(\sum_{i \in G_k} w_i \right)^2 1_{\mathfrak{G}_n} \right]} \right)^2. \end{aligned} \quad (\text{A.2.59})$$

Then it is enough to show

$$\left(\sum_{k=1}^{(3R+1)^2} \sqrt{E_{\theta} \left[\left(\sum_{i \in G_k} w_i \right)^2 1_{\mathfrak{G}_n} \right]} \right)^2 = \mathcal{O}(|\Lambda_n|); \quad (\text{A.2.60})$$

and thus it is sufficient to show

$$\sum_{k=1}^{(3R+1)^2} \sqrt{E_{\theta} \left[\left(\sum_{i \in G_k} w_i \right)^2 1_{\mathfrak{G}_n} \right]} = \mathcal{O}(\sqrt{|\Lambda_n|}), \quad (\text{A.2.61})$$

in which case we must establish

$$\sqrt{E_{\theta} \left[\left(\sum_{i \in G_k} w_i \right)^2 1_{\mathfrak{G}_n} \right]} = \mathcal{O}(\sqrt{|\Lambda_n|}) \quad (\text{A.2.62})$$

for each G_k , so that finally, it is enough to show

$$E_\theta \left[\left(\sum_{i \in G_k} w_i \right)^2 1_{\mathfrak{S}_n} \right] = O(|\Lambda_n|). \quad (\text{A.2.63})$$

for each G_k . Let \mathcal{C}_k be a corridor as constructed in the proof of Lemma 2.3.1. Then $E_\theta(W_i | x_{\mathcal{C}_k}) = 0$ for every configuration $x_{\mathcal{C}_k}$. Hence,

$$\begin{aligned} E_\theta \left[\left(\sum_{i \in G_k} w_i \right)^2 1_{\mathfrak{S}_n} \right] &= E_\theta \left\{ E_\theta \left[\left(\sum_{i \in G_k} w_i \right)^2 1_{\mathfrak{S}_n} \mid X_{\mathcal{C}_k} \right] \right\} \\ &= E_\theta \left\{ E_\theta \left[\left(\sum_{i \in G_k} w_i^2 \right) 1_{\mathfrak{S}_n} \mid X_{\mathcal{C}_k} \right] + E_\theta \left[\left(\sum_{i \in G_k} \sum_{\substack{j \in G_k \\ i \neq j}} w_i w_j \right) 1_{\mathfrak{S}_n} \mid X_{\mathcal{C}_k} \right] \right\} \\ &= E_\theta \left\{ \sum_{i \in G_k} E_\theta \left[w_i^2 1_{\mathfrak{S}_n} \mid X_{\mathcal{C}_k} \right] + \sum_{i \in G_k} \sum_{\substack{j \in G_k \\ i \neq j}} E_\theta \left[w_i w_j 1_{\mathfrak{S}_n} \mid X_{\mathcal{C}_k} \right] \right\}, \end{aligned}$$

so that, by Lemma A.1.1, we have

$$= E_\theta \left\{ \sum_{i \in G_k} E_\theta \left[w_i^2 1_{\mathfrak{S}_n} \mid X_{\mathcal{C}_k} \right] + \sum_{i \in G_k} \sum_{\substack{j \in G_k \\ i \neq j}} E_\theta \left[w_i 1_{\mathfrak{S}_n} \mid X_{\mathcal{C}_k} \right] E_\theta \left[w_j 1_{\mathfrak{S}_n} \mid X_{\mathcal{C}_k} \right] \right\}.$$

By the remark after (A.2.63),

$$\begin{aligned} &= E_\theta \left\{ \sum_{i \in G_k} E_\theta \left[w_i^2 1_{\mathfrak{S}_n} \mid X_{\mathcal{C}_k} \right] \right\} \\ &= \sum_{i \in G_k} E_\theta \left[w_i^2 1_{\mathfrak{S}_n} \right] \\ &\leq C |G_k| \end{aligned} \quad (\text{A.2.64})$$

for some $C > 0$ since the elements w_i are bounded. Since $|G_k| = \frac{|\Lambda_n|}{(3R+1)^2}$, (A.2.63) clearly follows. \square

Lemma 2.3.5 *Under the Identifiability Condition 2.1.1, for every $\varepsilon > 0$ there exists $\alpha > 0$ such that*

$$P_\theta(|\Lambda_n| \|\tilde{\theta}_n - \theta\|^2 > \varepsilon \log n) = O\left(\frac{1}{n^\alpha}\right) \quad (2.3.26)$$

as $n \rightarrow \infty$.

Proof: Note first that we restrict ourselves to the set \mathfrak{S}_n , since its complement is negligible (see the remark at the beginning of the proof of Theorem 3.5.1).

We will use the same notions here as we used in Lemma 2.3.4 above. By (A.2.49) on \mathfrak{S}_n we have

$$P_\theta(|\Lambda_n| \|\tilde{\theta}_n - \theta\|^2 > \varepsilon \log n) \leq P_\theta(|\Lambda_n| \|\nabla K(\theta, n)\|^2 > \frac{\varepsilon}{C} \log n) \quad (A.2.65)$$

for every $\varepsilon > 0$ and for some $C > 0$. Let $\varepsilon_1 < \frac{\varepsilon}{C}$. Then by equation (A.2.53), we see that the right-hand side of (A.2.65) is

$$\begin{aligned} &= P_\theta\left(|\Lambda_n| \left\| \frac{1}{|\Lambda_n|} \sum_{i \in \Lambda_n} W_i \right\|^2 > \varepsilon_1 \log n\right) \\ &= P_\theta\left(\left\| \sum_{i \in \Lambda_n} W_i \right\|^2 > \varepsilon_1 |\Lambda_n| \log n\right). \end{aligned} \quad (A.2.66)$$

As in the proof of Lemma 2.3.4 above, let w_i be a generic component of the vector W_i . Hence

$$\leq C_1 P_\theta\left(\left| \sum_{i \in \Lambda_n} w_i \right|^2 > \varepsilon_2 |\Lambda_n| \log n\right) \quad (A.2.67)$$

where C_1 is some positive constant greater than the dimension of the vector W_i , and $\varepsilon_2 < \frac{\varepsilon_1}{C_1}$. Thus,

$$\leq C_1 P_\theta\left(\left| \sum_{i \in \Lambda_n} w_i \right| > \varepsilon_3 \sqrt{|\Lambda_n| \log n}\right), \quad (A.2.68)$$

where $\varepsilon_3 < \sqrt{\varepsilon_2}$. Again, decompose Λ_n as in the proof of Lemma 2.3.1. Then from the triangle inequality, we have

$$\begin{aligned} &\leq C_1 P_\theta\left(\sum_{k=1}^{(3R+1)^2} \left| \sum_{i \in G_k} w_i \right| > \varepsilon_3 \sqrt{|\Lambda_n| \log n}\right) \\ &\leq C_2 P_\theta\left(\left| \sum_{i \in G_k} w_i \right| > \varepsilon \sqrt{|\Lambda_n| \log n}\right) \end{aligned} \quad (A.2.69)$$

for $C_2 > (3R+1)^2 C_1$ and $\varepsilon < \varepsilon_2 / (3R+1)^2$. Therefore it is enough to show that for every $\varepsilon > 0$,

$$P_\theta \left(\left| \sum_{i \in G_k} w_i \right| > \varepsilon \tau_n \right) = O\left(\frac{1}{n^\alpha}\right), \quad (\text{A.2.70})$$

where $\tau_n = \sqrt{|\Lambda_n| \log n}$.

Consider the two cases for the absolute value, studying first the positive case. Upon rearranging and using the Markov inequality, we have for $\rho > 0$

$$\begin{aligned} P_\theta \left(\sum_{i \in G_k} w_i > \varepsilon \tau_n \right) &= P_\theta \left(\frac{1}{\sqrt{\tau_n}} \sum_{i \in G_k} w_i > \varepsilon \sqrt{\tau_n} \right) \\ &\leq \exp(-\rho \varepsilon \sqrt{\tau_n}) E_\theta \left[\exp \left(\sum_{i \in G_k} \frac{\rho w_i}{\sqrt{\tau_n}} \right) \right]. \end{aligned} \quad (\text{A.2.71})$$

Now, using Lemma A.1.1 and the and the construction of the corridor \mathcal{C}_k in the proof of Lemma 2.3.1,

$$\begin{aligned} &= \exp(-\rho \varepsilon \sqrt{\tau_n}) E_\theta \left\{ E_\theta \left[\prod_{i \in G_k} \exp \left(\frac{\rho w_i}{\sqrt{\tau_n}} \right) \middle| X_{\mathcal{C}_k} \right] \right\} \\ &= \exp(-\rho \varepsilon \sqrt{\tau_n}) E_\theta \left\{ \prod_{i \in G_k} E_\theta \left[\exp \left(\frac{\rho w_i}{\sqrt{\tau_n}} \right) \middle| X_{\mathcal{C}_k} \right] \right\}. \end{aligned} \quad (\text{A.2.72})$$

Recall that $E_\theta(W_i | x_{\mathcal{C}_k}) = 0$ for every corridor configuration $x_{\mathcal{C}_k}$. By Taylor expansion of the rightmost exponential (the one within the expectations), we then get

$$\begin{aligned} &= \exp(-\rho \varepsilon \sqrt{\tau_n}) \prod_{i \in G_k} \left(1 + \frac{\rho^2}{\tau_n} O(1) \right) \\ &\leq \exp(-\rho \varepsilon \sqrt{\tau_n}) \left(1 + \frac{\rho^2}{\tau_n} O(1) \right)^{|G_k|} \\ &\leq \exp(-\rho \varepsilon \sqrt{\tau_n}) \exp \left(\frac{\alpha' \rho^2 |G_k|}{\tau_n} \right). \end{aligned}$$

for some $\alpha' > 0$. Let $\alpha'' = \frac{\alpha'}{(3R+1)^2}$. Then since $|G_k| = \frac{|\Lambda_n|}{(3R+1)^2}$ and $\tau_n = \sqrt{|\Lambda_n| \log n}$,

$$\begin{aligned}
&\leq \exp(-\rho\varepsilon\sqrt{\tau_n}) \exp\left(\frac{\alpha''\rho^2|\Lambda_n|}{\tau_n}\right) \\
&= \exp(-\rho\varepsilon\sqrt{\tau_n}) \exp\left(\frac{\alpha''\rho^2\sqrt{|\Lambda_n|}}{\sqrt{\log n}}\right).
\end{aligned}$$

Set $\rho = \varepsilon n^{-1/2}(\log n)^{3/4}(2\alpha'')^{-1}$ for $\nu > 0$, and note that $|\Lambda_n| = n^2$. Then

$$\begin{aligned}
&= \exp\left(\frac{-\varepsilon^2 \log n}{2\alpha''}\right) \exp\left(\frac{\alpha''\varepsilon^2 \log n}{(2\alpha'')^2}\right) \\
&= \exp\left(\frac{-\varepsilon^2 \log n}{4\alpha''}\right) \\
&= \frac{1}{n^\alpha} \tag{A.2.73}
\end{aligned}$$

with $\alpha = \frac{\varepsilon^2}{4\alpha''}$.

For the negative case,

$$P_\theta\left(-\sum_{i \in G_k} w_i > \varepsilon\tau_n\right) = O\left(\frac{1}{n^\alpha}\right) \tag{A.2.74}$$

can be derived in the same way. Hence the result follows. \square

A.3 Proofs of Lemmas from Chapter III

Lemma 3.5.1 *Under the Identifiability Condition 2.1.1, there exists a constant $c > 0$ such that*

$$P_\theta(\hat{m}_3 \in \mathcal{M}_1(\varpi)) \leq \exp(-n^c) \tag{3.5.3}$$

uniformly for all large n .

Proof: Recall that $\mathcal{M}_1(\varpi) = \{m \in \mathcal{M} : \theta \notin \bar{\Theta}_m\}$. Then there exists $\varepsilon_1 > 0$ such that $\|\theta - \vartheta\| \geq 3\varepsilon_1$ for all $\vartheta \in \bar{\Theta}_m$ and all $m \in \mathcal{M}_1(\varpi)$.

As in the proofs of Theorem 3.3.1 and Theorem 3.4.1, let $M \in \mathcal{M}$ be the “largest” model; i.e., $\bar{\Theta}_M = \Theta$. Also, since $\tilde{\theta}_n^M$ is the “global” MPLE, write $\tilde{\theta}_n^M = \tilde{\theta}_n$. In addition, let

$$\mathfrak{D}_n = \left\{x(n) \in \Omega_{\Lambda_n} : \|\tilde{\theta}_n - \theta\| \leq \varepsilon_1\right\}, \tag{A.3.1}$$

where ε_1 is given above. Then,

$$\begin{aligned} P_\theta(\{\mathfrak{E}_n \cap \mathfrak{T}_n\}^c) &= P_\theta(\mathfrak{E}_n^c \cup \mathfrak{T}_n^c) \\ &\leq P_\theta(\mathfrak{E}_n^c) + P_\theta(\mathfrak{T}_n^c) \end{aligned}$$

so that, applying Lemma 2.3.1 to $P_\theta(\mathfrak{E}_n^c)$ and Lemma 2.3.3 to $P_\theta(\mathfrak{T}_n^c)$, we have

$$\leq C_1 \exp(-c_1 n) + C_2 \exp(-c_2 |\Lambda_n|)$$

for some $c_1, c_2, C_1, C_2 > 0$ and for all large n . Indeed,

$$\leq \exp(-n^{a_1}) \tag{A.3.2}$$

for some $a_1 > 0$ and for all large n . Hence we now restrict our attention to $\mathfrak{E}_n \cap \mathfrak{T}_n$.

Let $m \in \mathcal{M}_1(\varpi)$. Recall that $K(\vartheta, n) = \vartheta^\top V_n - g_n(\vartheta) = \vartheta^\top V_n^M - g_n^M(\vartheta)$. By Lemma 2.3.2, $K(\vartheta, n)$ is globally concave, and locally strictly concave so that $\tilde{\theta}_n$ is the unique MPLE. Then, since the true parameter θ is away from $\bar{\Theta}_m$ (analogous to (3.3.20) in the proof of Theorem 3.3.1), there exists $\delta > 0$ such that

$$\sup_{\vartheta \in \bar{\Theta}_m} K(\vartheta, n) \leq K(\tilde{\theta}_n, n) - \delta \tag{A.3.3}$$

for all large n , P_θ -a.s. Then

$$\begin{aligned} Q_M^{(3)} - Q_m^{(3)} &= \left[\log \mathfrak{P}\mathcal{L}(x(n), \tilde{\theta}_n) - \frac{k_M}{2} \log |\Lambda_n| \right] - \left[\sup_{\vartheta \in \bar{\Theta}_m} \log \mathfrak{P}\mathcal{L}_m(x(n), \vartheta) - \frac{k_m}{2} \log |\Lambda_n| \right] \\ &= |\Lambda_n| K(\tilde{\theta}_n, n) - |\Lambda_n| \sup_{\vartheta \in \bar{\Theta}_m} K(\vartheta, n) - \left(\frac{k_M}{2} - \frac{k_m}{2} \right) \log |\Lambda_n| \\ &\geq \delta |\Lambda_n| - \left(\frac{k_M}{2} - \frac{k_m}{2} \right) \log |\Lambda_n| \\ &\geq a_2 |\Lambda_n| \end{aligned} \tag{A.3.4}$$

for some $a_2 > 0$ and for all $m \in \mathcal{M}_1(\varpi)$. Hence, by the Markov inequality,

$$P_\theta(\hat{m}_3 \in \mathcal{M}_1(\varpi)) \leq P_\theta(Q_{\hat{m}_3}^{(3)} - Q_M^{(3)} > 0)$$

$$\begin{aligned}
&= P_\theta\left(\left\{Q_{\hat{m}_3}^{(3)} - Q_M^{(3)} > 0\right\} \cap \{\mathfrak{S}_n \cap \mathfrak{D}_n\}\right) + \\
&\quad P_\theta\left(\left\{Q_{\hat{m}_3}^{(3)} - Q_M^{(3)} > 0\right\} \cap \{\mathfrak{S}_n \cap \mathfrak{D}_n\}^c\right)
\end{aligned}$$

so that, P_θ -a.s. as $n \rightarrow \infty$,

$$\begin{aligned}
&= P_\theta\left(\left\{\exp\left\{Q_{\hat{m}_3}^{(3)} - Q_M^{(3)}\right\} > 1\right\} \cap \{\mathfrak{S}_n \cap \mathfrak{D}_n\}\right) \\
&\leq E_\theta\left[\exp\left\{Q_{\hat{m}_3}^{(3)} - Q_M^{(3)}\right\} \mathbf{1}_{\{\mathfrak{S}_n \cap \mathfrak{D}_n\}}\right] \\
&\leq \exp\{-a_3|\Lambda_n|\} \\
&\leq \exp(-n^c)
\end{aligned} \tag{A.3.5}$$

for some $c > 0$. \square

Lemma 3.5.2 *Under the Identifiability Condition 2.1.1, there exists $\alpha > 0$ such that*

$$P_\theta(\hat{m}_3 \in \mathcal{M}_2(\varpi)) = O\left(\frac{1}{n^\alpha}\right). \tag{3.5.4}$$

Proof: Again, we restrict our attention to the set \mathfrak{S}_n , whose complement is negligible.

Recall that $\mathcal{M}_2(\varpi) = \{m \in \mathcal{M} : \bar{\Theta}_\varpi \subset \bar{\Theta}_m\}$, and let

$$\mathfrak{T}_n^c = \left\{|\Lambda_n| \|\tilde{\theta}_n^m - \theta\|^2 > \varepsilon \log n\right\}, \tag{A.3.6}$$

so that by Lemma 2.3.5, $P_\theta(\mathfrak{T}_n^c) = O\left(\frac{1}{n^\alpha}\right)$ for some $\alpha > 0$.

On \mathfrak{S}_n , for a chosen model $\hat{m} \in \mathcal{M}_2(\varpi)$, we have

$$\begin{aligned}
Q_\varpi^{(3)} - Q_{\hat{m}}^{(3)} &= \left[\log \mathfrak{P}_\varpi \mathcal{L}_\varpi(x(n), \tilde{\theta}_n^\varpi) - \frac{k_\varpi}{2} \log |\Lambda_n|\right] - \left[\log \mathfrak{P}_{\hat{m}} \mathcal{L}_{\hat{m}}(x(n), \tilde{\theta}_n^{\hat{m}}) - \frac{k_{\hat{m}}}{2} \log |\Lambda_n|\right] \\
&= |\Lambda_n| K_\varpi(\tilde{\theta}_n^\varpi, n) - |\Lambda_n| K_{\hat{m}}(\tilde{\theta}_n^{\hat{m}}, n) - k_\varpi \log n + k_{\hat{m}} \log n \\
&= |\Lambda_n| \left[K_\varpi(\tilde{\theta}_n^\varpi, n) - K_{\hat{m}}(\tilde{\theta}_n^{\hat{m}}, n) \right] - (k_\varpi - k_{\hat{m}}) \log n.
\end{aligned} \tag{A.3.7}$$

Using the “minimal” form of the pseudo-likelihood (3.5.1), the expression exactly analogous to

the Taylor expansion (3.3.7) for the MPLE, and Lemma 2.3.2 (guaranteeing the uniqueness of the MPLE), we now have

$$-\frac{C}{2} \leq \frac{\vartheta^\top V_n^{\hat{m}} - g_n^{\hat{m}}(\vartheta) - [(\tilde{\theta}_n^{\hat{m}})^\top V_n^{\hat{m}} - g_n^{\hat{m}}(\tilde{\theta}_n^{\hat{m}})]}{\|\vartheta - \tilde{\theta}_n^{\hat{m}}\|^2} \leq -\frac{c}{2} \quad (\text{A.3.8})$$

for some $c, C > 0$ and for $\vartheta \in \bar{\Theta}_{\hat{m}}$ in a neighborhood of $\tilde{\theta}_n^{\hat{m}}$. Now, because $\bar{\Theta}_\varpi \subset \bar{\Theta}_{\hat{m}}$ (from the definition of $\mathcal{M}_2(\varpi)$), we know $\tilde{\theta}_n^\varpi \in \bar{\Theta}_{\hat{m}}$ and we may thus evaluate (A.3.8) at $\vartheta = \tilde{\theta}_n^\varpi$:

$$-\frac{C}{2} \leq \frac{(\tilde{\theta}_n^\varpi)^\top V_n^{\hat{m}} - g_n^{\hat{m}}(\tilde{\theta}_n^\varpi) - [(\tilde{\theta}_n^{\hat{m}})^\top V_n^{\hat{m}} - g_n^{\hat{m}}(\tilde{\theta}_n^{\hat{m}})]}{\|\tilde{\theta}_n^\varpi - \tilde{\theta}_n^{\hat{m}}\|^2} \leq -\frac{c}{2}. \quad (\text{A.3.9})$$

Via the minimal form of the exponential family, we then get

$$-\frac{C}{2} \leq \frac{(\tilde{\theta}_n^\varpi)^\top V_n^\varpi - g_n^\varpi(\tilde{\theta}_n^\varpi) - [(\tilde{\theta}_n^{\hat{m}})^\top V_n^{\hat{m}} - g_n^{\hat{m}}(\tilde{\theta}_n^{\hat{m}})]}{\|\tilde{\theta}_n^\varpi - \tilde{\theta}_n^{\hat{m}}\|^2} \leq -\frac{c}{2}. \quad (\text{A.3.10})$$

By (A.3.10) plus the triangle inequality,

$$Q_\varpi^{(3)} - Q_{\hat{m}}^{(3)} \geq |\Lambda_n| \left[-\frac{C}{2} \left(\|\tilde{\theta}_n^{\hat{m}} - \theta\|^2 + \|\tilde{\theta}_n^\varpi - \theta\|^2 \right) \right] - (k_\varpi - k_{\hat{m}}) \log n$$

for some $C > 0$, so that, on on the set \mathfrak{F}_n ,

$$\begin{aligned} &\geq -2C\varepsilon \log n - (k_\varpi - k_{\hat{m}}) \log n \\ &\geq a_1 \log n \end{aligned} \quad (\text{A.3.11})$$

where $a_1 = k_{\hat{m}} - k_\varpi - 2C\varepsilon > 0$, provided ε is sufficiently small. Hence, by the Markov inequality,

$$P_\theta(\hat{m}_3 \in \mathcal{M}_1(\varpi)) \leq P_\theta(Q_{\hat{m}}^{(3)} - Q_\varpi^{(3)} > 0)$$

$$\begin{aligned}
&= P_\theta\left(\left\{\exp\left\{Q_{\hat{m}}^{(3)} - Q_{\bar{w}}^{(3)}\right\} > 1\right\} \cap \{\mathfrak{E}_n \cap \mathfrak{F}_n\}\right) + \\
&\quad P_\theta\left(\left\{\exp\left\{Q_{\hat{m}}^{(3)} - Q_{\bar{w}}^{(3)}\right\} > 1\right\} \cap \{\mathfrak{E}_n \cap \mathfrak{F}_n\}^c\right) \\
&\leq E_\theta\left[\exp\left\{Q_{\hat{m}}^{(3)} - Q_{\bar{w}}^{(3)}\right\} 1_{\{\mathfrak{E}_n \cap \mathfrak{F}_n\}}\right] \\
&\leq \exp\{-a_1 \log n\} \\
&= n^{-a_1} \\
&= O\left(\frac{1}{n^\alpha}\right) \tag{A.3.12}
\end{aligned}$$

for some $\alpha > 0$. \square

Appendix Two Simulated Textures

For all of the following models, $U_1(x_i) = 0$ and $U_2(x_i, x_j) = x_i x_j$ for all sites i and all $j \in \mathcal{N}_i$ (see Section 1.3 for more details). The format for display is as follows.

First, the number identifying the Candidate Model is given (see Section 4.3 for a detailed description). Then, the reader is given a reminder of the neighborhood system as well as of the Uniqueness Condition 1.3.1 (see Chapter I). Next, we give a parameter value, and then we indicate whether the Uniqueness Condition 1.3.1 is satisfied by indicating if there is definitely no long-range dependence (recall that if the Uniqueness Condition 1.3.1 is satisfied, then not only is there no long-range dependence, but the random field will look very much like an i.i.d. field). Finally, a realization from the described model is shown.

For reasons of practicality, the presentation begins on the next page.

Candidate Model: 1 (Ising Model)

Neighborhood: $\begin{matrix} \cdot & \beta & \cdot \\ \beta & i & \beta \\ \cdot & \beta & \cdot \end{matrix}$

Uniqueness Condition 1.3.1:

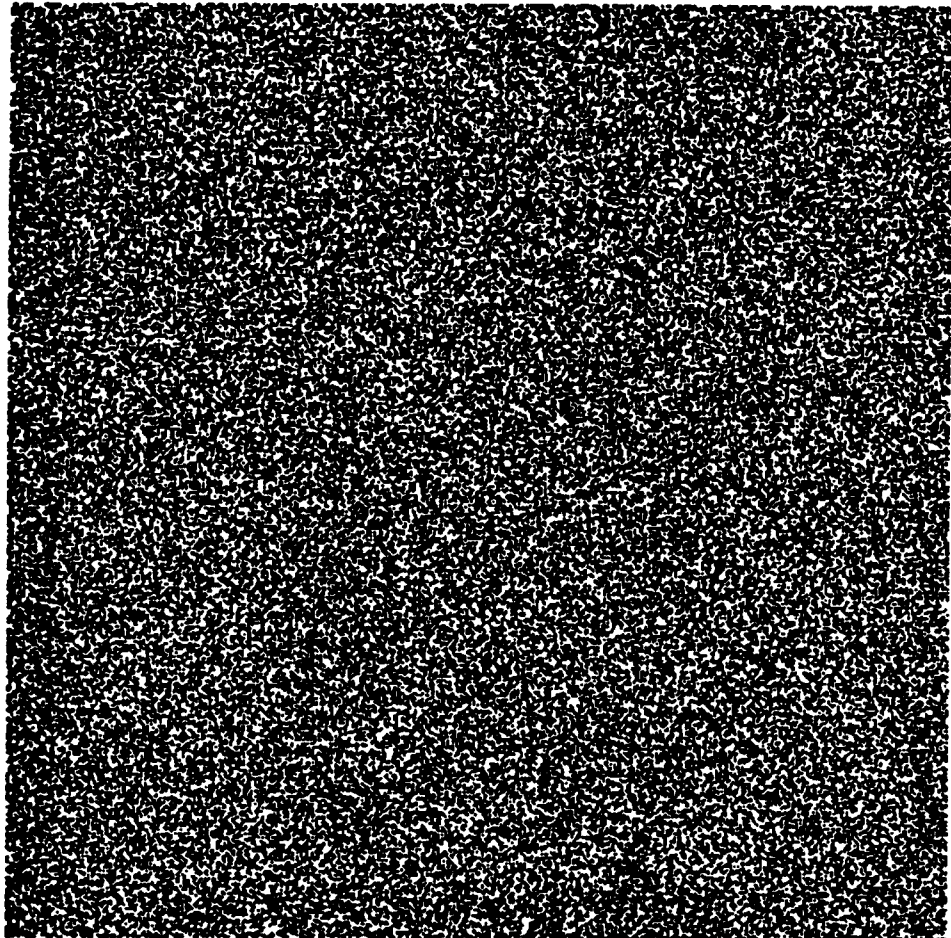
Necessary and sufficient condition for no phase transitions (see Section 4.3):

$$|\beta| < \beta_c = \frac{1}{2} \log(1 + \sqrt{2})$$

Parameter value: $\beta = .1$

Long-range dependence: No.

Realization:



Candidate Model: 1 (Ising Model)

Neighborhood: $\begin{matrix} \cdot & \beta & \cdot \\ \beta & i & \beta \\ \cdot & \beta & \cdot \end{matrix}$

Uniqueness Condition 1.3.1:

Necessary and sufficient condition for no phase transitions (see Section 4.3):

$$|\beta| < \beta_c = \frac{1}{2} \log(1 + \sqrt{2})$$

Parameter value: $\beta = 1$

Long-range dependence: Yes.

Realization:



Candidate Model: 1 (Ising Model)

Neighborhood: $\begin{matrix} \cdot & \beta & \cdot \\ \beta & i & \beta \\ \cdot & \beta & \cdot \end{matrix}$

Uniqueness Condition 1.3.1:

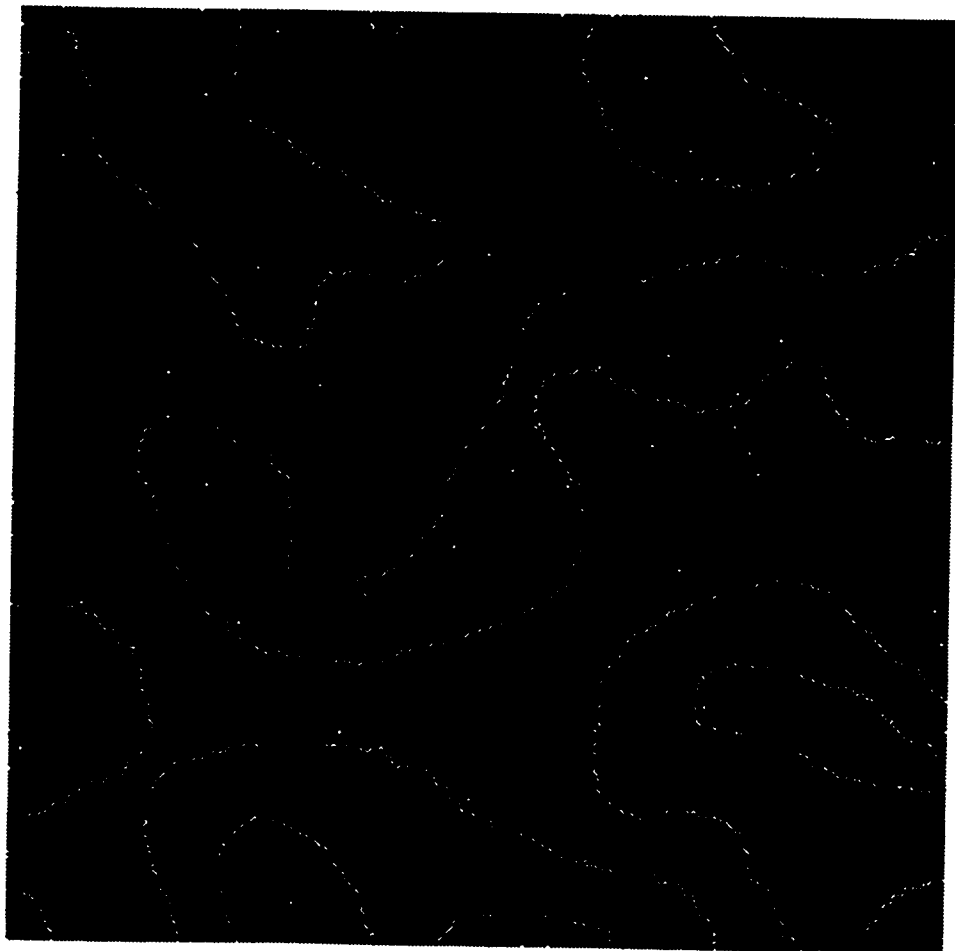
Necessary and sufficient condition for no phase transitions (see Section 4.3):

$$|\beta| < \beta_c = \frac{1}{2} \log(1 + \sqrt{2})$$

Parameter value: $\beta = -1$

Long-range dependence: Yes.)

Realization:



Candidate Model: 1 (Ising Model)

Neighborhood: $\begin{matrix} \cdot & \beta & \cdot \\ \beta & i & \beta \\ \cdot & \beta & \cdot \end{matrix}$

Uniqueness Condition 1.3.1:

Necessary and sufficient condition for no phase transitions (see Section 4.3):

$$|\beta| < \beta_c = \frac{1}{2} \log(1 + \sqrt{2})$$

Parameter value: $\beta = 2$

Long-range dependence: Yes.

Realization:



Candidate Model: 2

Neighborhood: $\begin{matrix} \cdot & \beta_1 & \cdot \\ \beta_2 & i & \beta_2 \\ \cdot & \beta_1 & \cdot \end{matrix}$

Uniqueness Condition 1.3.1:

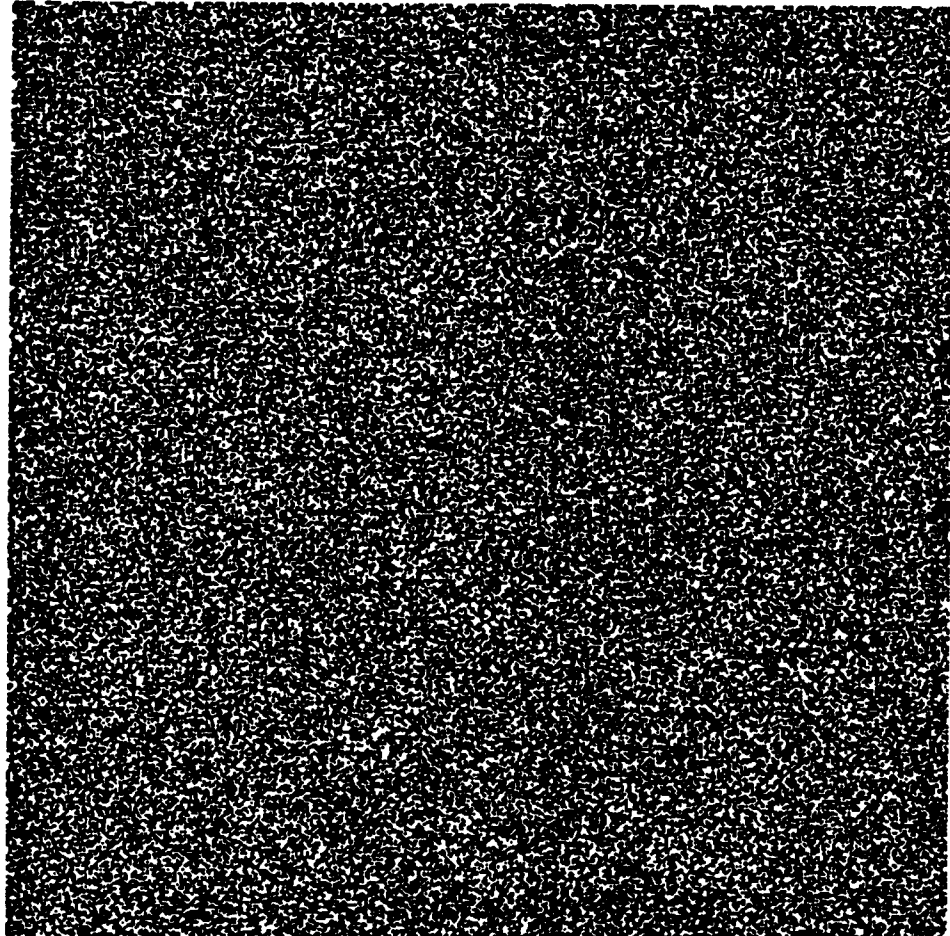
Sufficient condition for no phase transitions (see Section 4.3):

$$|\beta_1| + |\beta_2| < \frac{1}{2}$$

Parameter value: $\begin{pmatrix} \beta_1 \\ \beta_2 \end{pmatrix} = \begin{pmatrix} .01 \\ .1 \end{pmatrix}$

Long-range dependence: No.

Realization:



Candidate Model: 2

Neighborhood: $\begin{matrix} \cdot & \beta_1 & \cdot \\ \beta_2 & i & \beta_2 \\ \cdot & \beta_1 & \cdot \end{matrix}$

Uniqueness Condition 1.3.1:

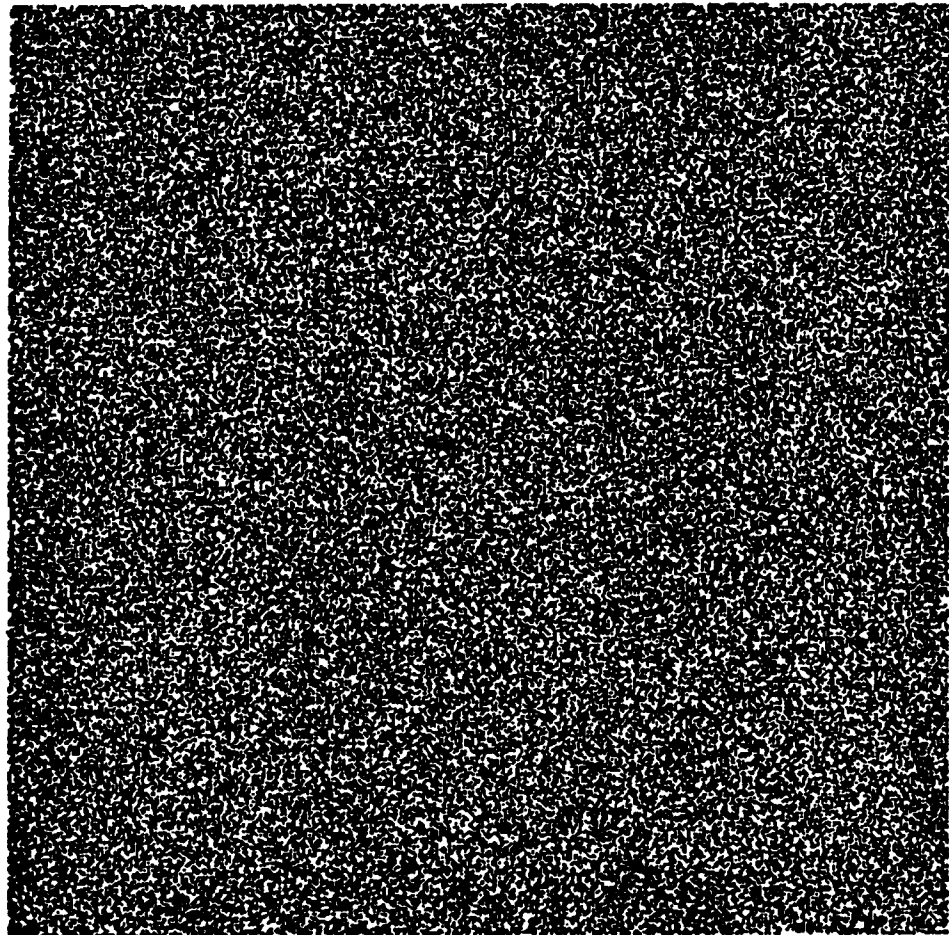
Sufficient condition for no phase transitions (see Section 4.3):

$$|\beta_1| + |\beta_2| < \frac{1}{2}$$

Parameter value: $\begin{pmatrix} \beta_1 \\ \beta_2 \end{pmatrix} = \begin{pmatrix} .1 \\ .01 \end{pmatrix}$

Long-range dependence: No.

Realization:



Candidate Model: 2

Neighborhood: $\begin{matrix} \cdot & \beta_1 & \cdot \\ \beta_2 & i & \beta_2 \\ \cdot & \beta_1 & \cdot \end{matrix}$

Uniqueness Condition 1.3.1:

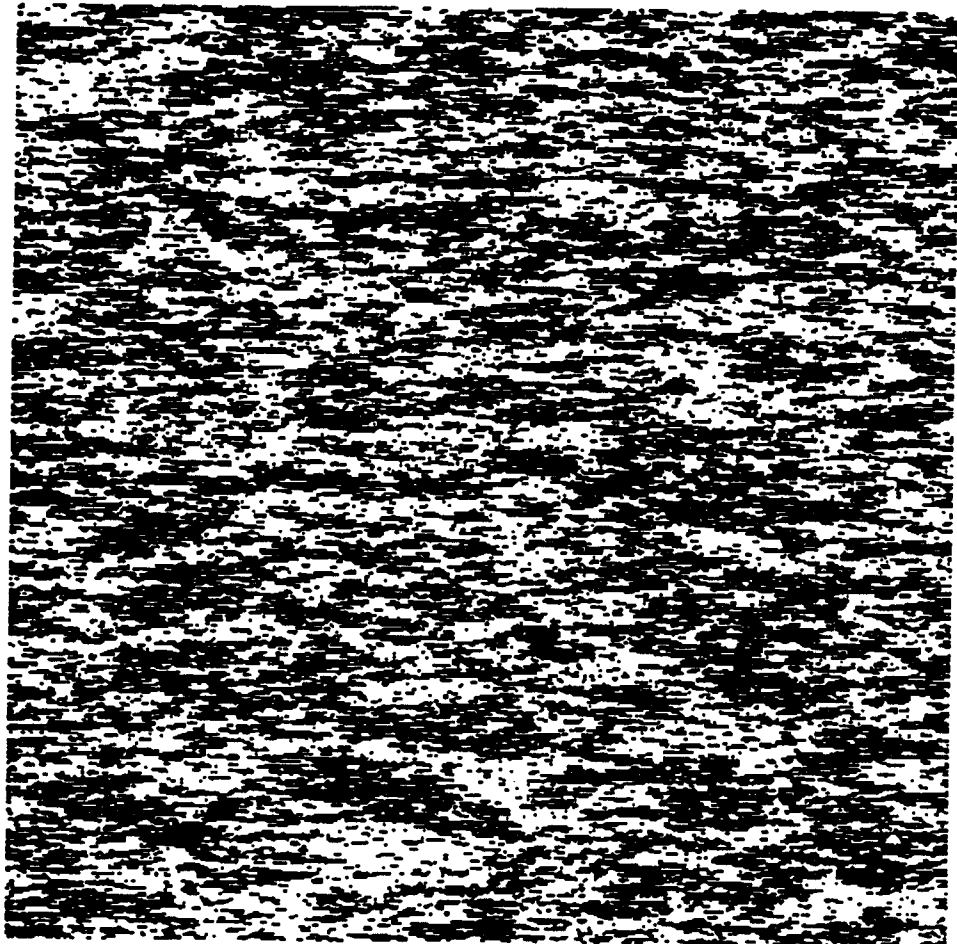
Sufficient condition for no phase transitions (see Section 4.3):

$$|\beta_1| + |\beta_2| < \frac{1}{2}$$

Parameter value: $\begin{pmatrix} \beta_1 \\ \beta_2 \end{pmatrix} = \begin{pmatrix} .1 \\ 1 \end{pmatrix}$

Long-range dependence: ?

Realization:



Candidate Model: 2

Neighborhood: $\begin{matrix} \cdot & \beta_1 & \cdot \\ \beta_2 & i & \beta_2 \\ \cdot & \beta_1 & \cdot \end{matrix}$

Uniqueness Condition 1.3.1:

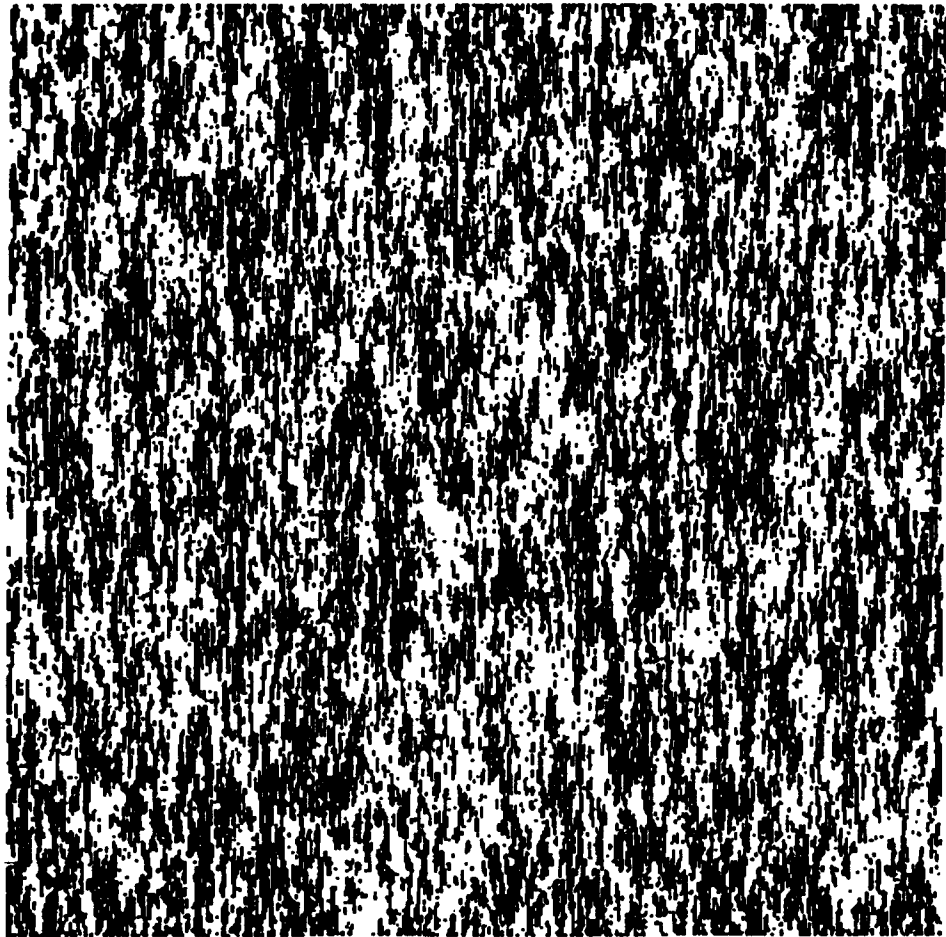
Sufficient condition for no phase transitions (see Section 4.3):

$$|\beta_1| + |\beta_2| < \frac{1}{2}$$

Parameter value: $\begin{pmatrix} \beta_1 \\ \beta_2 \end{pmatrix} = \begin{pmatrix} 1 \\ .1 \end{pmatrix}$

Long-range dependence: ?

Realization:



Candidate Model: 2

Neighborhood: $\begin{matrix} \cdot & \beta_1 & \cdot \\ \beta_2 & i & \beta_2 \\ \cdot & \beta_1 & \cdot \end{matrix}$

Uniqueness Condition 1.3.1:

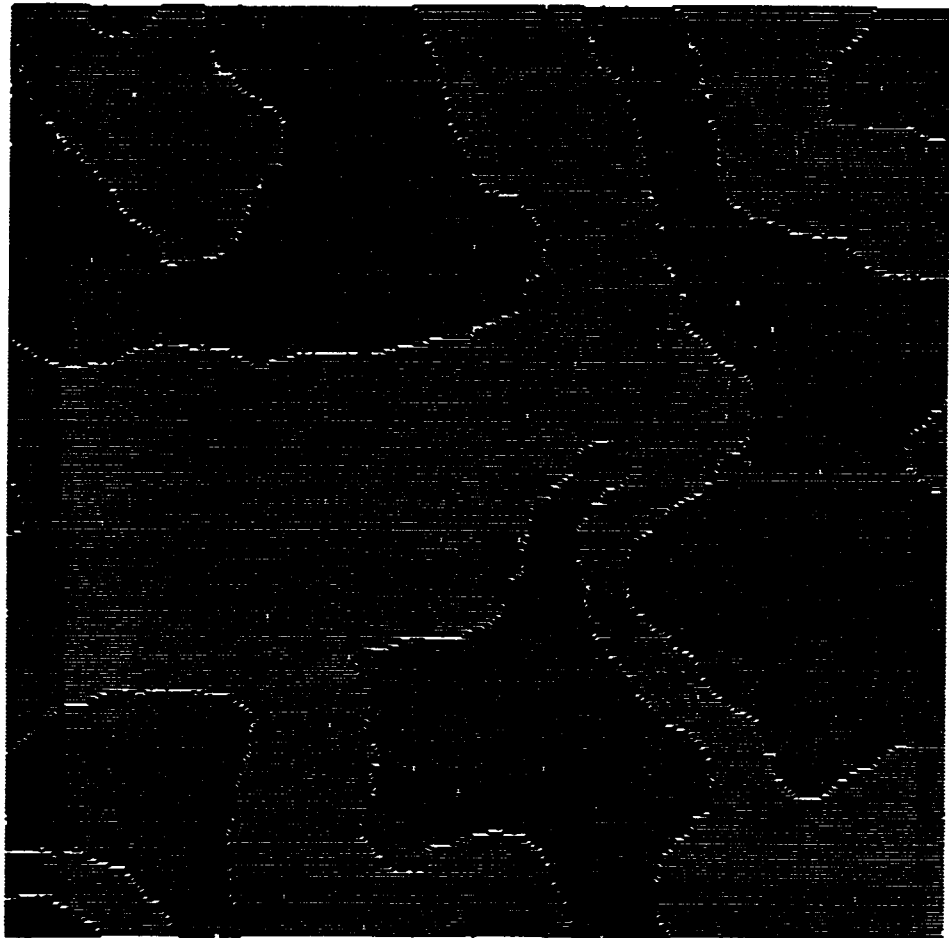
Necessary and sufficient condition for no phase transitions (see Section 4.3):

$$|\beta_1| + |\beta_2| < \frac{1}{2}$$

Parameter value: $\begin{pmatrix} \beta_1 \\ \beta_2 \end{pmatrix} = \begin{pmatrix} -1 \\ 1 \end{pmatrix}$

Long-range dependence: ?

Realization:



Candidate Model: 2

Neighborhood: $\begin{matrix} \cdot & \beta_1 & \cdot \\ \beta_2 & i & \beta_2 \\ \cdot & \beta_1 & \cdot \end{matrix}$

Uniqueness Condition 1.3.1:

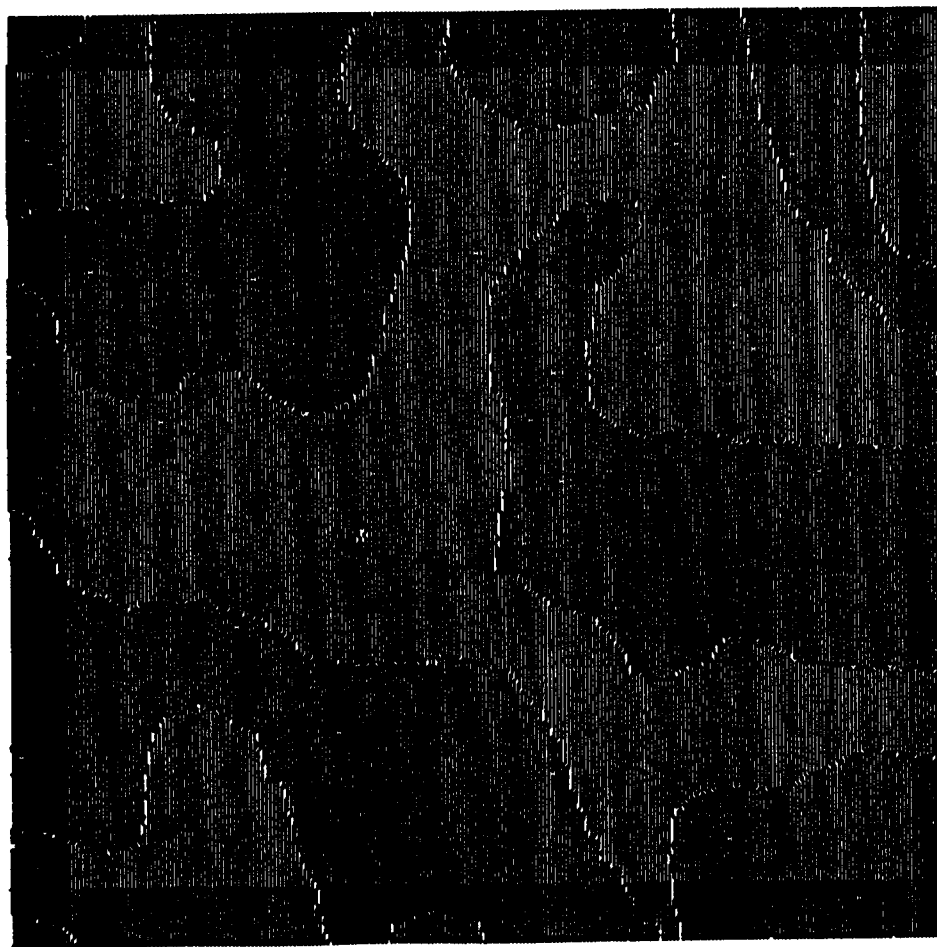
Sufficient condition for no phase transitions (see Section 4.3):

$$|\beta_1| + |\beta_2| < \frac{1}{2}$$

Parameter value: $\begin{pmatrix} \beta_1 \\ \beta_2 \end{pmatrix} = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$

Long-range dependence: ?

Realization:



Candidate Model: 3

· · γ · ·
· γ β γ ·
Neighborhood: γ β i β γ
· γ β γ ·
· · γ · ·

Uniqueness Condition 1.3.1:

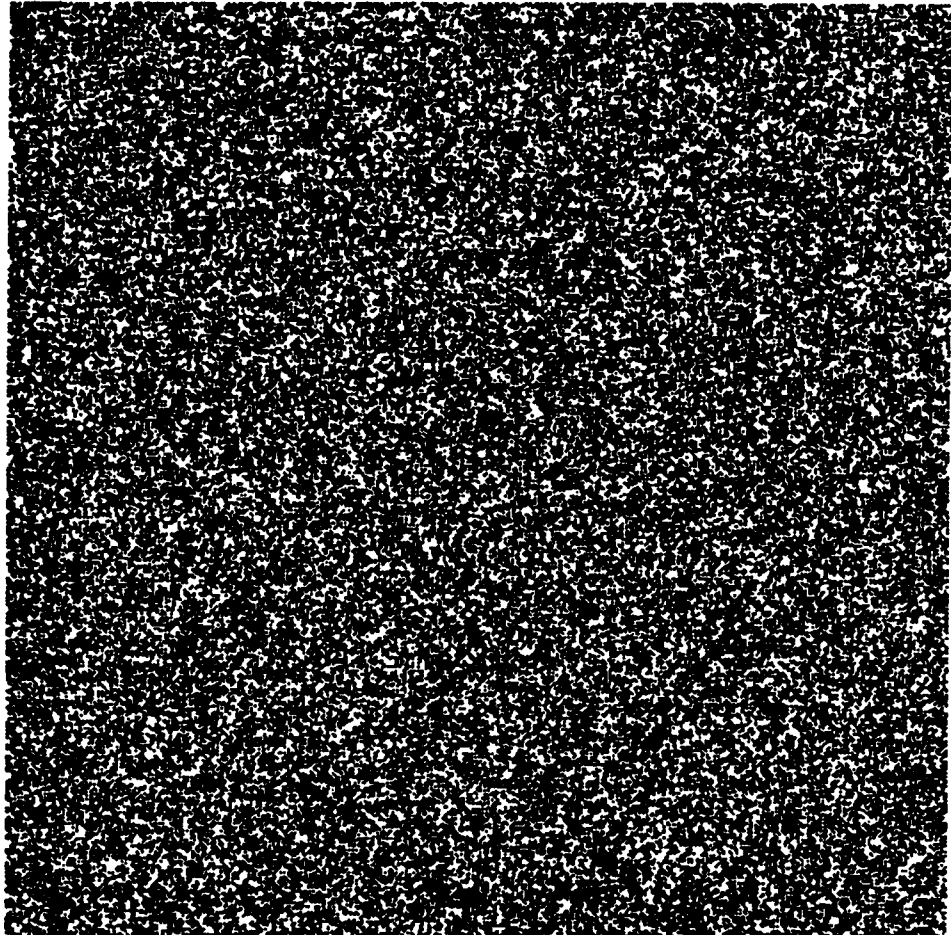
Sufficient condition for no phase transitions (see Section 4.3):

$$4|\beta| + 8|\gamma| < 1$$

Parameter value: $\begin{pmatrix} \beta \\ \gamma \end{pmatrix} = \begin{pmatrix} .01 \\ .1 \end{pmatrix}$

Long-range dependence: No.

Realization:



Candidate Model: 3

· · γ · ·
· γ β γ ·
Neighborhood: γ β i β γ
· γ β γ ·
· · γ · ·

Uniqueness Condition 1.3.1:

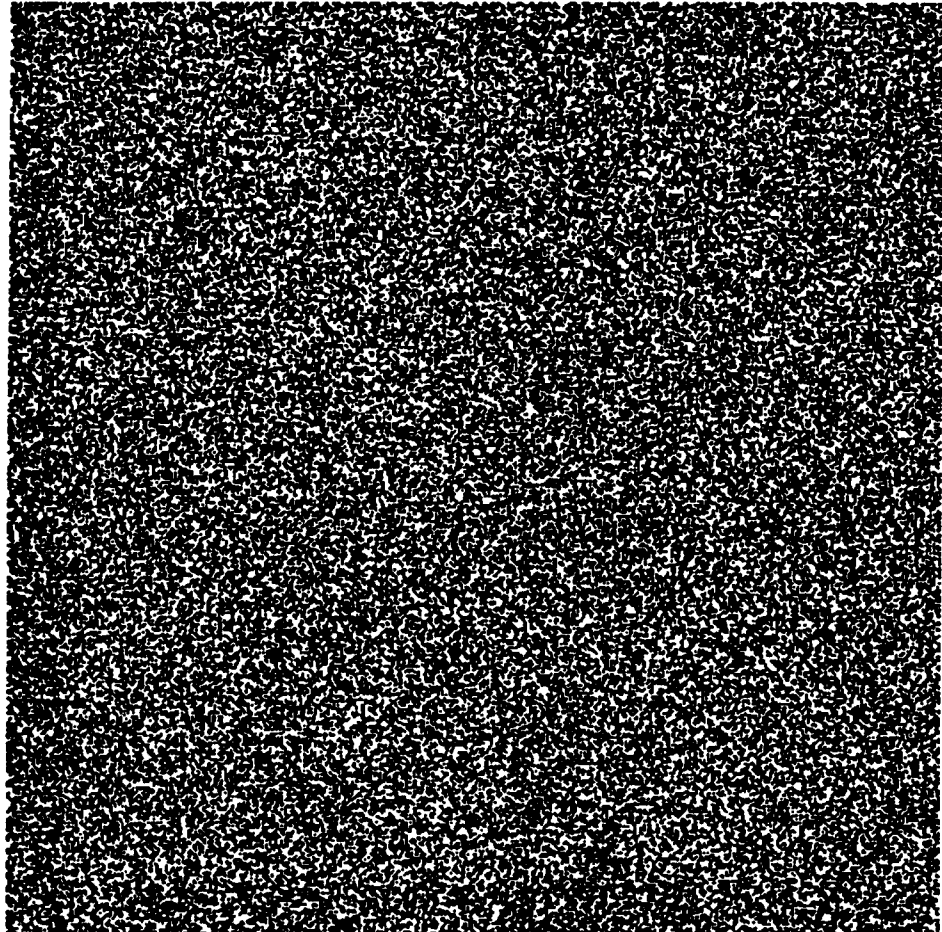
Sufficient condition for no phase transitions (see Section 4.3):

$$4|\beta| + 8|\gamma| < 1$$

Parameter value: $\begin{pmatrix} \beta \\ \gamma \end{pmatrix} = \begin{pmatrix} .1 \\ .01 \end{pmatrix}$

Long-range dependence: No.

Realization:



Candidate Model: 3

· · γ · ·
· γ β γ ·
Neighborhood: γ β β β γ
· γ β γ ·
· · γ · ·

Uniqueness Condition 1.3.1:

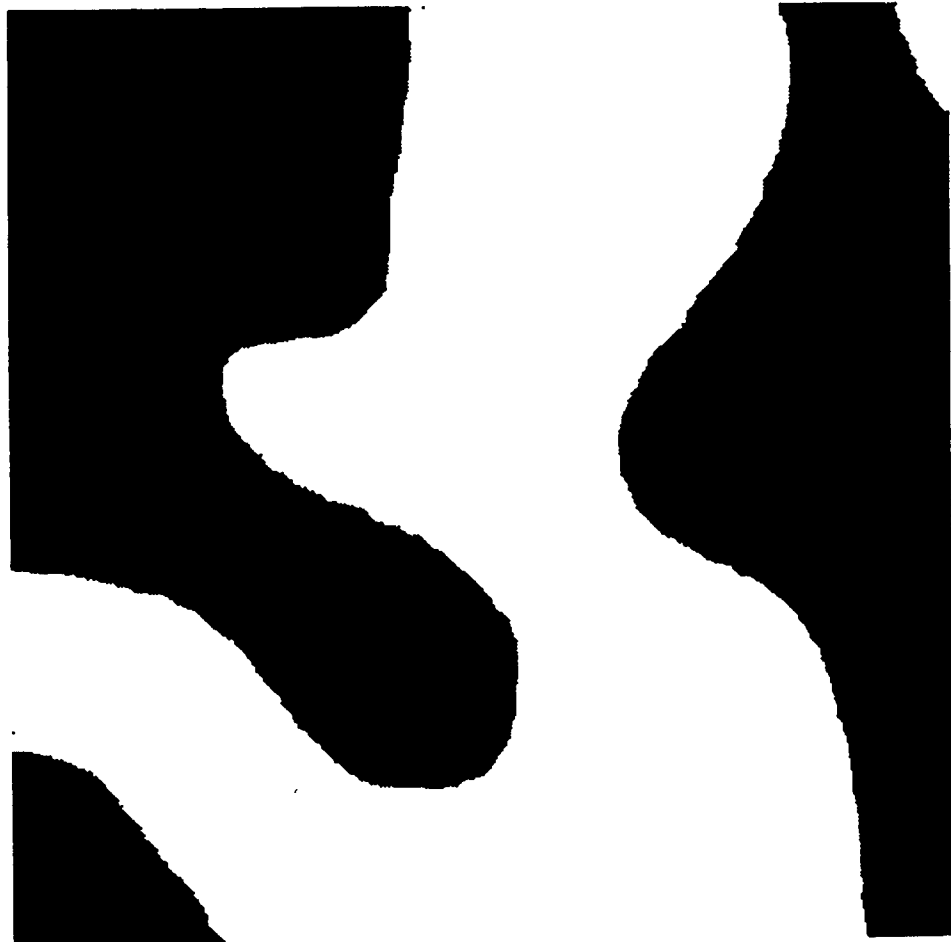
Sufficient condition for no phase transitions (see Section 4.3):

$$4|\beta| + 8|\gamma| < 1$$

Parameter value: $\begin{pmatrix} \beta \\ \gamma \end{pmatrix} = \begin{pmatrix} .1 \\ 1 \end{pmatrix}$

Long-range dependence: ?

Realization:



Candidate Model: 3

· · γ · ·
· γ β γ ·
Neighborhood: γ β i β γ
· γ β γ ·
· · γ · ·

Uniqueness Condition 1.3.1:

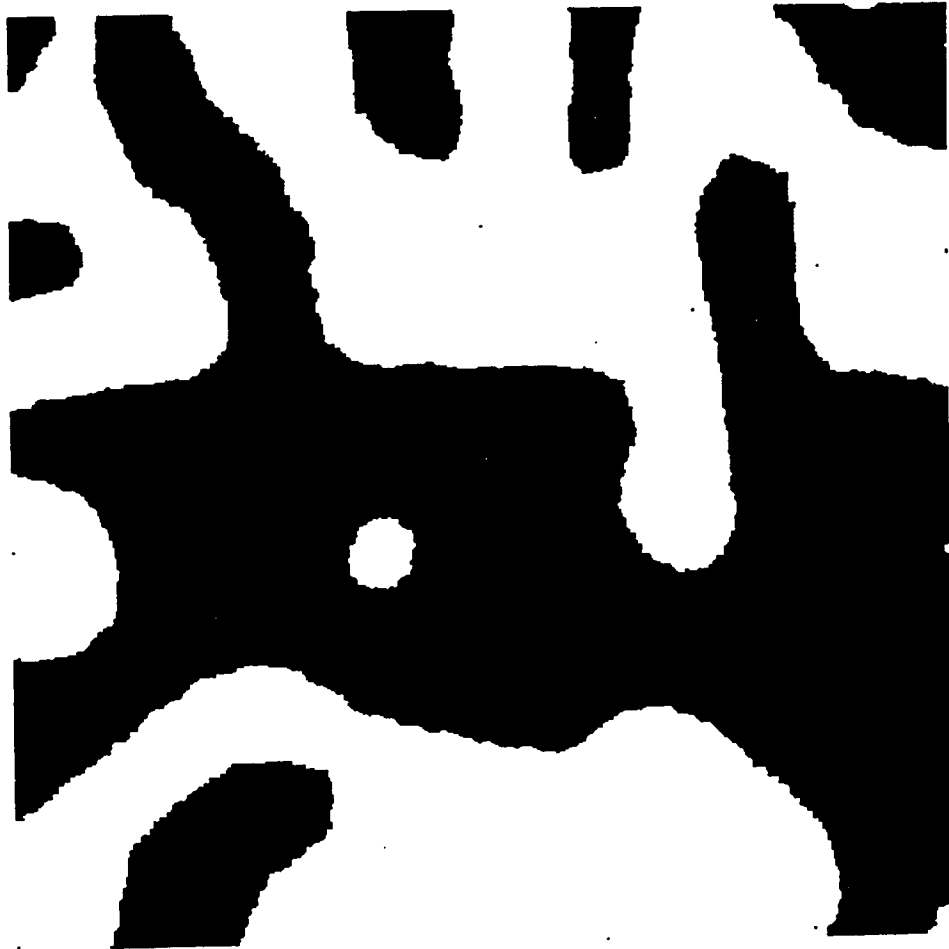
Sufficient condition for no phase transitions (see Section 4.3):

$$4|\beta| + 8|\gamma| < 1$$

Parameter value: $\begin{pmatrix} \beta \\ \gamma \end{pmatrix} = \begin{pmatrix} 1 \\ .1 \end{pmatrix}$

Long-range dependence: ?

Realization:



Candidate Model: 3

· · γ · ·
· γ β γ ·
Neighborhood: γ β i β γ
· γ β γ ·
· · γ · ·

Uniqueness Condition 1.3.1:

Sufficient condition for no phase transitions (see Section 4.3):

$$4|\beta| + 8|\gamma| < 1$$

Parameter value: $\begin{pmatrix} \beta \\ \gamma \end{pmatrix} = \begin{pmatrix} -1 \\ 1 \end{pmatrix}$

Long-range dependence: ?

Realization:



Candidate Model: 3

Neighborhood: $\begin{matrix} \cdot & \cdot & \gamma & \cdot & \cdot \\ \cdot & \gamma & \beta & \gamma & \cdot \\ \gamma & \beta & i & \beta & \gamma \\ \cdot & \gamma & \beta & \gamma & \cdot \\ \cdot & \cdot & \gamma & \cdot & \cdot \end{matrix}$

Uniqueness Condition 1.3.1:

Sufficient condition for no phase transitions (see Section 4.3):

$$4|\beta| + 8|\gamma| < 1$$

Parameter value: $\begin{pmatrix} \beta \\ \gamma \end{pmatrix} = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$

Long-range dependence: ?

Realization:



Appendix Three Computer Programs

program rfggen

```
c This program employs the Gibbs sampler to generate a random field
c with certain local characteristics, as given in the subroutine
c LOCHAR.
c
c LOCHAR should be the only model-specific subroutine.
c
c For convenience in changing program parameters, the required input
c is read from the file RFGEN.IN in the subroutine INITIAL.
c
c Mnemonic: Random Field GENERator
c
c Subroutines called: INITIAL, BERGEN, GIBSAM
```

```
integer*1 rf
character*80 outfile
common/dims/ nx,ny,layer
common/rfld/ rf(900,900)
```

```
c Initialize the program by reading the necessary input from the
c file RFGEN.IN. If the dimensions or choice of model is
c inappropriate, as indicated by IFLAG, the the program terminates.
```

```
call initial(outfile,iflag)
if (iflag.eq.1) stop
```

```
c The Gibbs Sampler says that we can start with any random field we
c like, so just begin with an iid Bernoulli-like random field.
```

```
call bergen
```

```
c With a place to start, we now go through the Gibbs Sampler.
```

```
call gibsam
```

```
c Now write out the random field, and end. Note: The random field is
c treated as if it is sitting on an inverted Cartesian plane, with the
c origin at the upper left-hand corner (just like a graphics screen!).
```

```
open(unit=10,file=outfile,status='new')
```

```

do 10 iy=1+layer,ny+layer
  write(10,*) (rf(ix,iy),ix=1+layer,nx+layer)
10 continue

close(unit=10)

stop
end

```

C*****

```

subroutine initial(outfile,iflag)

```

```

c This subroutine reads the file RFGEN.IN, which contains all of the
c parameters required to initialize this program.

```

```

c

```

```

c The variable GARBAGE is for reading in meaningful titles from the
c input file (so that the user knows what is what in that file).

```

```

c

```

```

c Mnemonic: INITIALize the program

```

```

character*80 outfile,garbage
logical outrnge
common/misc/ nit,p,iseed
common/dims/ nx,ny,layer
common/modl/ beta,gamma,model

```

```

c Fix the probability for the Bernoulli random field.

```

```

data p/.5/

```

```

c Set the flag for bad input to 0, which means that all is OK.

```

```

iflag=0

```

```

c Open the input file.

```

```

open(unit=11,file='rfggen.in',status='old')

```

```

c Read in a seed for the random number generator.

```

```

read(11,'(A)') garbage
read(11,*) iseed

```

```

c Read in the dimensions of the random field.

```

```

read(11,'(A)') garbage
read(11,*) nx,ny
outrnge=nx.le.0.or.nx.gt.900.or.ny.le.0.or.ny.gt.900
if(outrnge) then
  write(*,*) "Dimensions are less than 0 or greater than 900"
  iflag=1
  goto 10
end if

```

c Read in the choice of model.

```
    read(11,'(A)') garbage
    read(11,*) model
```

c If MODEL=1, then there is only one LAYER of neighbors.
c If MODEL=2, then there is also only one LAYER of neighbors.
c If MODEL=3, then there are two LAYERS of neighbors.

```
    if(model.eq.1) then
      layer=1
    else if(model.eq.2) then
      layer=1
    else if(model.eq.3) then
      layer=2
    else
      write(*,*) "Bad choice of model."
      iflag=1
      goto 10
    end if
```

c Read in the appropriate parameters. If MODEL=1, then GAMMA should
c be 0. If MODEL=2, then BETA covers the vertical direction and
c GAMMA covers the horizontal direction.

```
    read(11,'(A)') garbage
    read(11,*) beta
    read(11,'(A)') garbage
    read(11,*) gamma
```

c Read in the maximum number of iterations for the Gibbs Sampler.

```
    read(11,'(A)') garbage
    read(11,*) nit
```

c Read in the name of the output file.

```
    read(11,'(A)') garbage
    read(11,'(A)') outfile
```

```
    close(unit=11)
```

```
10  return
    end
```

C*****

```
    subroutine bergen
```

c This subroutine generates an iid random field that is "sort of"
c Bernoulli in that each site takes on 1 with probability P and
c -1 with probability 1-P.

c

c Mnemonic: BERnoulli GENerator

```

integer*1 rf
common/misc/ nit,p,iseed
common/dims/ nx,ny,layer
common/rfld/ rf(900,900)

c We put an extra LAYER on top, bottom, and on each side so that the
c random field we want is sitting smack in the middle. The extra
c around the edges is so that we can evaluate the local
c characteristics at the edges of the random field (the one in the
c middle).

do 20 iy=1,ny+2*layer
  do 10 ix=1,nx+2*layer
    unif=ran(iseed)
    if(unif.le.p) then
      rf(ix,iy)=1
    else
      rf(ix,iy)=-1
    end if
10   continue
20   continue

return
end

*****

subroutine gibsam

c This subroutine performs the Gibbs Sampler on each site in RF except
c for the edges (where there are not complete neighborhoods).
c Note: The subroutine LOCHAR may be written however you need to
c evaluate the local characteristics at the site (IX,IY).
c
c Mnemonic: GIBbs SAMpler
c
c Subroutines called: LOCHAR

common/misc/ nit,p,iseed
common/dims/ nx,ny,layer

c One iteration consists of one pass over the entire random field.

do 30 kit=1,nit
  do 20 iy=1+layer,ny+layer
    do 10 ix=1+layer,nx+layer
      call lochar(ix,iy)
10   continue
20   continue
30   continue

return
end

```

C*****

```
subroutine lochar(ix,iy)
```

```
c This subroutine evaluates the local characteristics at the specific  
c site IX,IY. P1 is the probability that the site is 1.
```

```
c
```

```
c Mnemonic: LOCal CHARacteristics
```

```
integer*1 rf  
common/misc/ nit,p,iseed  
common/mod1/ beta,gamma,model  
common/rfld/ rf(900,900)
```

```
c Depending on the model, calculate the appropriate neighborhood  
c sum and exponent.
```

```
if(model.eq.1) then  
  isum=rf(ix-1,iy)+rf(ix+1,iy)+rf(ix,iy-1)+rf(ix,iy+1)  
  expnt=beta*isum  
else if(model.eq.2) then  
  isum1=rf(ix,iy-1)+rf(ix,iy+1)  
  isum2=rf(ix-1,iy)+rf(ix+1,iy)  
  expnt=beta*isum1+gamma*isum2  
else  
  isum1=rf(ix,iy-1)+rf(ix,iy+1)+rf(ix-1,iy)+rf(ix+1,iy)  
  isum2=rf(ix,iy-2)+rf(ix-1,iy-1)+rf(ix-2,iy)+rf(ix-1,iy+1)+  
&      rf(ix,iy+2)+rf(ix+1,iy+1)+rf(ix+2,iy)+rf(ix+1,iy-1)  
  expnt=beta*isum1+gamma*isum2  
end if
```

```
c P1 is the probability that the site is 1 given the neighbors.
```

```
p1=exp(expnt)/(exp(expnt)+exp(-expnt))  
unif=ran(iseed)  
if(unif.le.p1) then  
  rf(ix,iy)=1  
else  
  rf(ix,iy)=-1  
end if
```

```
return  
end
```

program driver

```
c This program simply executes the subroutine PRAMEST for either
c Markov chain Monte Carlo or pseudo-likelihood parameter
c estimation.
c
c The subroutine PRAMEST and all of the subroutines it requires may be
c found in the program MODSEL.
c
c Mnemonic: DRIVER for the subroutine PRAMEST
c
c Subroutines called: INITIAL, PRAMEST
```

```
implicit double precision (a-h,o-z)
character*80 outfile
dimension theta(5)
common/dims/ nx,ny,npram
common/mcar/ mcopt,mcsam,ithone
common/modl/ model,layer
common/pram/ psi(5)
```

```
c Initialize the program. This includes reading in the observed
c random field.
```

```
call initial(outfile)
```

```
c Do the parameter estimation.
```

```
call pramest(theta,fncmin,ifail)
```

```
c Write it all out.
```

```
if(ifail.eq.0) then
```

```
c If everything is OK, then print out the relevant parameters and the
c answers.
```

```
open(unit=12,file=outfile,status='new')
write(12,*) "dims = ",nx,ny
write(12,*) "model = ",model
write(12,*) "theta = ",(theta(n),n=1,npram)
if(mcopt.eq.1) then
  write(12,*) "psi = ",(psi(n),n=1,npram)
  write(12,*) "Monte Carlo samples = ",mcsam
  write(12,*) "Skip factor = ",ithone
endif
close(unit=12)
else
```

```
c Otherwise, give an error message. File 13 is always the error file.
```

```
open(unit=13,file='error.out',status='unknown')
write(13,*) "output file",outfile
if(ifail.eq.1) then
  write(13,*) "Number of parameters is out of range:"
```

```

        write(13,*) npram
    else if(ifail.eq.2) then
        write(13,*) "Too many function evaluations."
        write(13,*) "theta=",(theta(i),i=1,npram)
        write(13,*) "fncmin=",fncmin
    else if(ifail.eq.3) then
        write(13,*) "Initial point too big."
        write(13,*) fncmin
    endif
    close(unit=13)
endif

stop
end

```

C*****

```

subroutine initial(outfile)

```

```

c This subroutine reads the file PRAMEST.IN, which contains all of the
c input required to initialize the driver program for the subroutine
c PRAMEST.
c

```

```

c The variable GARBAGE is for reading in meaningful titles from the
c input file (so that the user knows what is what in that file).
c

```

```

c Note: file unit 13 is always the error file.
c

```

```

c Mnemonic: INITIALize the program
c

```

```

c Called from: MAIN

```

```

character*80 garbage,infile,outfile
integer rf
logical outrnge
common/dims/ nx,ny,npram
common/mcar/ mcopt,mcsam,ithone
common/misc/ nit,iseed
common/modl/ model,layer
common/rfld/ rf(900,900)

```

```

c Set the warning flag. If IFLAG is 0, then everything is OK.

```

```

    iflag=0

```

```

c Open the input file.

```

```

    open(unit=10,file='pramest.in',status='old')

```

```

c Read in the seed for the random number generator.

```

```

    read(10,'(A)') garbage
    read(10,*) iseed

```

```

c Read in the dimensions of the random field.

```

```

read(10,'(A)') garbage
read(10,*) nx,ny
outrnge=nx.le.0.or.nx.gt.900.or.ny.le.0.or.ny.gt.900
if(outrnge) then
  iflag=1
  goto 20
endif

```

c Read in the model specifics. LAYER is the number of layers in the
c neighborhood system for the particular MODEL. NPRAM is the number
c of parameters for the particular MODEL.

```

read(10,'(A)') garbage
read(10,*) model,layer,npram
outrnge=model.le.0.or.model.gt.3.or.layer.le.0.or.layer.gt.2
outrnge=outrnge.or.npram.le.0.or.npram.gt.5
if(outrnge) then
  iflag=2
  goto 20
endif

```

c Read in the Monte Carlo option. If MCOPT=0, then we're doing MPLE.
c If MCOPT=1, we're doing MCMLE.

```

read(10,'(A)') garbage
read(10,*) mcopt
if(mcopt.lt.0.or.mcopt.gt.1) then
  iflag=3
  goto 20
endif

```

c Read in the Monte Carlo specifics. If you're not doing the Monte
c Carlo thing, then these quantities will be ignored in the program.
c The variable ITHONE should be 1 if you are not using a variance-
c reducing scheme. Otherwise, the program will skip every ITHONE-1
c realizations in the Monte Carlo Markov chain and sample the ITHONE
c (get it? . . . ith one!!).

```

read(10,'(A)') garbage
read(10,*) mcsam,ithone
outrnge=mcsam*ithone.lt.0.or.mcsam.gt.500
if(outrnge) then
  iflag=4
  goto 20
endif

```

c Read in the name of the input file for the observed rf.

```

read(10,'(A)') garbage
read(10,'(A)') infile

```

c Read in the name of the output file.

```

read(10,'(A)') garbage

```

```

        read(10,'(A)') outfile

        close(unit=10)

c Now read in that observed rf. Initialize it first so it won't choke
c on the input!!

        open(unit=11,file=infile,status='old')
        do 15 iy=1,ny
            do 10 ix=1,nx
                rf(ix,iy)=0
10         continue
            read(11,*) (rf(ix,iy),ix=1,nx)
15        continue
        close(unit=11)

        return

c Deal with the errors, if there are any.

20    open(unit=13,file='error.out',status='new')
        if(iflag.eq.1) then
            write(13,*) "Something is wrong with the dimensions:"
            write(13,*) nx,ny
        else if(iflag.eq.2) then
            write(13,*) "Something is wrong with the model specifics:"
            write(13,*) model,layer,npram
        else if(iflag.eq.3) then
            write(13,*) "MCOPT is wrong:",mcopt
        else if(iflag.eq.4) then
            write(13,*) "Monte Carlo specifics are wrong:"
            write(13,*) mcsam,ithone
        else
            write(13,*) "IFLAG is wrong:",iflag
        endif
        close(unit=13)
        stop

    end

```

program modsel

c This program performs model selection on the input random field from
c a given pool of candidate models. The selection criterion is based
c on either the pseudo-likelihood (Besag, 1974, 1986) or the Monte
c Carlo approximant to the likelihood (Geyer and Thompson, 1992).

c
c Subroutines called:

c Main Routines: INITIAL, PRAMEST
c Pseudo-Likelihood Routines: MPLE, PLKHD
c Monte Carlo Routines: MCMLE, SUFSTAT, MCSTAT, MCLKHD
c Gibbs Sampler Routines: BERGEN, GIBSAM, LOCHAR
c Minimizer Routines: QNMIN, LKHD, GRAD

c
c The routines PLKHD, SUFSTAT, MCLKHD, and LOCHAR are model-dependent.
c The rest are, hopefully, very general.

```
implicit double precision(a-h,o-z)
character*80 outfile
dimension theta(20),thetamax(20)
common/dims/ nx,ny,npram
common/mcar/ mcopt,mcsam,ithone
common/modl/ model,layer
```

c Initialize the program.

```
call initial(outfile)
```

c Open the output file.

```
open(unit=12,file=outfile,status='new')
```

c Write out some informative headers.

```
write(12,*) "dims = ",nx,ny
if(mcopt.eq.0) then
  write(12,*) "Using pseudo-likelihood criterion"
else
  write(12,*) "Using Monte Carlo criterion"
  write(12,*) "Sampling",mcsam,"at intervals of",ithone
end if
```

c Initialize the selection process.

```
npram=1
model=1
layer=1
do 15 i=1,npram
  theta(i)=.5d0
15 continue
```

c Start with the first model, and store its results in the MAX
c holding places.

```
call pramest(theta,fncmin,ifail)
```

```

        if(ifail.ne.0) goto 50
        modmax=model
        npramax=npram
        do 20 i=1,npram
            thetamax(i)=theta(i)
20    continue
        bicmax=(-fncmin)-.5d0*npram*dlog(1.d0*nx*ny)
        write(12,*) model,(theta(i),i=1,npram),bicmax

```

c Begin looping in search of the model which maximizes the BIC

```

30    model=model+1
        if(model.eq.2) npram=2
        if(model.eq.3) layer=2
        do 35 i=1,npram
            theta(i)=.5d0
35    continue
        call pramest(theta,fncmin,ifail)
        if(ifail.ne.0) goto 50
        bic=(-fncmin)-.5*npram*log(1.*nx*ny)
        if(bic.gt.bicmax) then
            modmax=model
            npramax=npram
            do 40 i=1,npram
                thetamax(i)=theta(i)
40    continue
            bicmax=bic
        end if
        write(12,*) model,(theta(i),i=1,npram),bic

        if(model.eq.3) goto 60
        goto 30

```

```

50    write(12,*) model," ifail=",ifail

```

c This file won't have a name, but unit number 13 is always my error
c file.

```

        write(13,*) "model=",model
        if(ifail.eq.1) then
            write(13,*) "Number of parameters is out of range:"
            write(13,*) npram
        else if(ifail.eq.2) then
            write(13,*) "Too many function evaluations."
            write(13,*) "theta=",(theta(i),i=1,npram)
            write(13,*) "fncmin=",fncmin
        else if(ifail.eq.3) then
            write(13,*) "Initial point too big."
            write(13,*) fncmin
        endif
        if(model.ne.3) goto 30
        close(unit=13)

60    write(12,*)
        write(12,*) "Model chosen:"

```

```

write(12,*) modmax,(thetamax(i),i=1,npramax),bicmax
close(unit=12)

stop
end

```

```

C*****
c                               MAIN ROUTINES
C*****

```

```

subroutine pramest(theta,fncmin,ifail)

```

```

c This subroutine performs parameter estimation on the input random
c field according to the indicated Gibbs-Markov model. There is a
c choice of parameter estimates: one, based on the pseudo-likelihood
c of Besag, and the other, a Monte Carlo estimate of Geyer and
c Thompson.

```

```

c
c Mnemonic: PaRAMeter ESTimator
c
c Called from: MAIN
c
c Subroutines called: MPLE, MCMLE

```

```

implicit double precision (a-h,o-z)
dimension theta(5)
common/mcar/ mcopt,mcsam,ithone

```

```

if(mcopt.eq.0) then
call mple(theta,fncmin,ifail)
else
call mcmle(theta,fncmin,ifail)
endif

```

```

return
end

```

```

C*****

```

```

subroutine initial(outfile)

```

```

c This subroutine reads the file MODSEL.IN, which contains all of the
c input required to initialize this program.

```

```

c
c The variable GARBAGE is for reading in meaningful titles from the
c input file (so that the user knows what is what in that file).

```

```

c
c Note: file unit 13 is always the error file.

```

```

c
c Mnemonic: INITIALize the program

```

```

c
c Called from: MAIN

```

```

character*80 garbage,infile,outfile
integer rf

```

```
logical outrnge
common/dims/ nx,ny,npram
common/mcar/ mcopt,mcsam,ithone
common/misc/ nit,iseed
common/modl/ model,layer
common/rfld/ rf(900,900)
```

c Set the warning flag. If IFLAG is 0, then everything is OK.

```
iflag=0
```

c Open the input file.

```
open(unit=10,file='modsel.in',status='old')
```

c Read in the seed for the random number generator.

```
read(10,'(A)') garbage
read(10,*) iseed
```

c Read in the dimensions of the random field.

```
read(10,'(A)') garbage
read(10,*) nx,ny
outrnge=nx.le.0.or.nx.gt.900.or.ny.le.0.or.ny.gt.900
if(outrnge) then
  iflag=1
  goto 30
endif
```

c Read in the Monte Carlo option.

```
Read(10,'(A)') garbage
read(10,*) mcopt
if(mcopt.lt.0.or.mcopt.gt.1) then
  iflag=2
  goto 30
endif
```

c Read in the Monte Carlo specifics. If you're not doing the Monte Carlo thing, then these quantities will be ignored.

```
read(10,'(A)') garbage
read(10,*) mcsam,ithone
outrnge=mcsam*ithone.lt.0.or.mcsam.gt.1000
if(outrnge) then
  iflag=3
  goto 30
endif
```

c Read in the name of the input file for the observed rf.

```
read(10,'(A)') garbage
read(10,'(A)') infile
```

```

c Read in the name of the output file.

      read(10,'(A)') garbage
      read(10,'(A)') outfile

      close(unit=10)

c Now read in that observed rf. Initialize the rf to keep the program
c from choking on the input.

      open(unit=11,file=infile,status='old')
      do 20 iy=1,ny
        do 10 ix=1,nx
          rf(ix,iy)=0
10      continue
        read(11,*) (rf(ix,iy),ix=1,nx)
20      continue
      close(unit=11)

      return

c Deal with the errors.

30      open(unit=13,file='error.out',status='new')
      if(iflag.eq.1) then
        write(13,*) "Something is wrong with the dimensions:"
        write(13,*) nx,ny
      else if(iflag.eq.2) then
        write(13,*) "MCOPT is wrong:",mcopt
      else if(iflag.eq.3) then
        write(13,*) "Monte Carlo specifics are wrong:"
        write(13,*) mcsam,ithone
      else
        write(13,*) "IFLAG is wrong:",iflag
      endif
      close(unit=13)
      stop

      end

C*****
c                               PSEUDO-LIKELIHOOD ROUTINES
C*****

      subroutine mple(theta,fncmin,ifail)

c This subroutine calculates the maximum pseudo-likelihood estimate of
c the parameter using the assumed model.
c
c Mnemonic: Maximum Pseudo-Likelihood Estimator
c
c Called from: PRAMEST, MCMLE
c
c Subroutines called: QNMIN

```

```

        implicit double precision (a-h,o-z)
        dimension theta(5)
        common/dims/ nx,ny,npram

c Initialize the parameter THETA.  Gotta start somewhere!

        do 10 i=1,npram
            theta(i)=.5d0
10     continue

c Find the maximum pseudo-likelihood estimate of THETA by minimizing
c -LN(PLKHD).

        call qnmin(npram,theta,fncmin,200,ifail)

c If anything is wrong (IFAIL is not 0), the error message is given
c upon the return from PRAMEST.

        return
    end

C*****

        subroutine plkhd(n,b,p)

c This subroutine calculates the negative log of the pseudo-likelihood
c function for a given parameter B.
c
c Note: This routine is very model-specific.
c
c Mnemonic: (negative log) Pseudo-LiKeliHooD function
c
c Called from: LKHD

        implicit double precision (a-h,o-z)
        dimension b(5)
        double precision lnnum,ln-den
        integer rf
        common/dims/ nx,ny,npram
        common/modl/ model,layer
        common/rfld/ rf(900,900)

        sum=0.d0

c Recall that we start one LAYER into the observed random field so
c that each site taken has a complete set of neighbors.

        do 20 iy=1+layer,ny-layer
            do 10 ix=1+layer,nx-layer
                if(model.eq.1) then
                    nbrsum=rf(ix-1,iy)+rf(ix,iy+1)+rf(ix+1,iy)+rf(ix,iy-1)
                    factor=b(n)*nbrsum
                else if(model.eq.2) then
                    nbrsum1=rf(ix,iy-1)+rf(ix,iy+1)
                    nbrsum2=rf(ix-1,iy)+rf(ix+1,iy)

```

```

        factor=b(1)*nbrsum1+b(2)*nbrsum2
    else
        nbrsum1=rf(ix-1,iy)+rf(ix,iy+1)+rf(ix+1,iy)+rf(ix,iy-1)
        nbrsum2=rf(ix-2,iy)+rf(ix-1,iy-1)+rf(ix,iy-2)+
&         rf(ix+1,iy-1)+rf(ix+2,iy)+rf(ix+1,iy+1)+
&         rf(ix,iy+2)+rf(ix-1,iy+1)
        factor=b(1)*nbrsum1+b(2)*nbrsum2
    end if
    lnnum=rf(ix,iy)*factor
    lnden=dlog(dexp(factor)+dexp(-factor))
    sum=sum-lnnum+lnden
10    continue
20    continue
    p=sum

    return
end

```

```

C*****
c                                     MONTE CARLO ROUTINES
C*****

```

```

    subroutine mcml(theta,fncmin,ifail)

```

```

c This subroutine performs the Monte Carlo maximum likelihood
c estimation as outlined in the paper of Geyer and Thompson (1991).

```

```

c Mnemonic: Monte Carlo Maximum Likelihood Esimator

```

```

c Called from: PRAMEST

```

```

c Subroutines called: SUFSTAT, MPLE, BERGEN, MCSTAT, QNMIN

```

```

    dimension theta(5)
    double precision psi,theta,fncmin
    integer rf
    common/dims/ nx,ny,npram
    common/mcar/ mcopt,mcsam,ithone
    common/sfst/ obst(5),samt(5,1000)
    common/pram/ psi(5)
    common/rfld/ rf(900,900)

```

```

c Calculate the statistic on the observed random field. Load the
c result into OBST. Set the option so that SUFSTAT knows where the
c random field is, and also so that LKHD knows to call PLKHD to
c calculate the MPLE.

```

```

    mcopt=0
    call sufstat(obst,rf)

```

```

c Calculate the PSI (i.e., the Monte Carlo parameter) we use, which is
c the MPLE for the observed random field.

```

```

    call mple(psi,fncmin,ifail)

```

```
c If something is wrong, then return. Otherwise, reset the option so
c that the program will know to do Monte Carlo stuff now!
```

```
    if(ifail.ne.0) return
    mcopt=1
```

```
c Since the Gibbs Sampler can start anywhere, just start with a
c Bernoulli-like random field.
```

```
c
c Note: we use the same space for the observed random field as we do
c for the Monte Carlo samples, so this will wipe out the observed
c random field!
```

```
    call bergen
```

```
c Get the Monte Carlo stats for the PSI we use, which is the MPLE.
```

```
    call mcstat
```

```
c Initialize the parameter THETA. Gotta start somewhere!
```

```
    do 10 n=1,npram
        theta(n)=.5d0
10    continue
```

```
c Find the Monte Carlo maximum likelihood estimate of THETA by
c minimizing -LN(MCLKHD).
```

```
    call qnmin(npram,theta,fnclmin,200,ifail)
```

```
c If everything is not OK (IFAIL is not 0), the error message will be
c given upon return from PRAMEST.
```

```
    return
end
```

```
C*****
```

```
    subroutine mclkh(d(npram,theta, val)
```

```
c This subroutine calculates the negative log of the Monte Carlo
c approximant to the likelihood function for a given parameter THETA.
```

```
c
c Mnemonic: (negative log) Monte Carlo LiKeliHooD function
```

```
c
c Called from: LKHD
```

```
    dimension theta(5),expnt(1000)
    double precision psi,theta,val
    double precision sum.expnt,exmax,exmin,esum,exfac
    common/mcar/ mcopt,mcsam,ithone
    common/misc/ nit,iseed
    common/pram/ psi(5)
    common/sfst/ obst(5),samt(5,1000)
```

```

sum=0.d0
exmax=0.d0

do 20 i=1,mcsam
  esum=0.d0
  do 10 n=1,npram
    esum=esum+(theta(n)-psi(n))*samt(n,i)
10  continue
    expnt(i)=esum
    if(expnt(i).gt.exmax) then
      exmax=expnt(i)
    endif
20  continue

do 30 i=1,mcsam
  sum=sum+dexp(expnt(i)-exmax)
30  continue
  esum=0.d0
  do 40 n=1,npram
    esum=esum+theta(n)*obst(n)
40  continue
  val=-esum+exmax+dlog(sum/(1.d0*mcsam))
  return
end

C*****

subroutine sufstat(stat,rf)

c This function calculates the "sufficient statistic" for the finite
c random field RF.
c
c Note: This subroutine is very model-specific.
c
c Mnemonic: SUFFICIENT STATistic
c
c Called from: MCMLE, MCSTAT

dimension isum(5),stat(5),rf(900,900)
integer rf
common/dims/ nx,ny,npram
common/mcar/ mcopt,mcsam,ithone
common/modl/ model,layer

do 10 n=1,npram
  isum(n)=0
10  continue

lshift=0
if(mcopt.eq.1) lshift=layer

c For this calculation, we don't need to worry about the LAYER of
c neighbors - we're just using all available pair-cliques.

c Sum over vertical 1-step pairs.

```

```

do 20 i=1+lshift,nx+lshift
  do 15 j=1+lshift,ny-1+lshift
    isum(1)=isum(1)+rf(i,j)*rf(i,j+1)
15    continue
20    continue

    if(model.eq.1 .or. model.eq.3) then

c Sum over horizontal 1-step pairs.

    do 30 i=1+lshift,nx-1+lshift
      do 25 j=1+lshift,ny+lshift
        isum(1)=isum(1)+rf(i,j)*rf(i+1,j)
25      continue
30      continue
        if(model.eq.3) then

c Sum over horizontal 2-step pairs.

          do 40 i=1+lshift,nx-2+lshift
            do 35 j=1+lshift,ny+lshift
              isum(2)=isum(2)+rf(i,j)*rf(i,j+2)
35            continue
40            continue

c Sum over vertical 2-step pairs.

            do 50 i=1+lshift,nx+lshift
              do 45 j=1+lshift,ny-2+lshift
                isum(2)=isum(2)+rf(i,j)*rf(i+2,j)
45              continue
50              continue

c Sum over right-diagonal pairs.

              do 60 i=1+lshift,nx-1+lshift
                do 55 j=1+lshift,ny-1+lshift
                  isum(2)=isum(2)+rf(i,j)*rf(i+1,j+1)
55                continue
60                continue

c Sum over left-diagonal pairs.

                do 70 i=1+lshift,nx+lshift
                  do 65 j=1+lshift,ny-1+lshift
                    isum(2)=isum(2)+rf(i,j)*rf(i-1,j+1)
65                  continue
70                  continue
                  endif
                else if(model.eq.2) then

c Sum over horizontal 1-step pairs.

                  do 80 i=1+lshift,nx-1+lshift

```

```

        do 75 j=1+lshift,ny+lshift
            isum(2)=isum(2)+rf(i,j)*rf(i+1,j)
75      continue
80      continue
endif

do 85 n=1,npram
    stat(n)=float(isum(n))
85      continue

return
end

```

C*****

```

subroutine mcstat

```

```

c This subroutine calculates the "suffecient statistics" for each of
c the Monte Carlo samples it takes and dumps them into the array SAMT.
c
c Mnemonic: Monte Carlo STATistics
c
c Called from: MCMLE
c
c Subroutines called: GIBSAM, SUFSTAT

```

```

    dimension stat(5)
    integer rf
    common/dims/ nx,ny,npram
    common/mcar/ mcopt,mcsam,ithone
    common/misc/ nit,iseed
    common/mcrf/ rf(900,900)
    common/sfst/ obst(5),samt(5,1000)

```

```

c A Bernoulli field has already been loaded into RF, so just start:
c run into the Markov chain for a little ways . . .

```

```

    nit=200
    call gibsam

```

```

c Calculate the first sample statistic.

```

```

    call sufstat(stat,rf)
    do 10 n=1,npram
        samt(n,1)=stat(n)
10      continue

```

```

c Now do the same thing. sampling MCSAM times. taking every ITHONE.

```

```

    nit=ithone
    do 30 m=2,mcsam
        call gibsam
        call sufstat(stat,rf)
        do 20 n=1,npram
            samt(n,m)=stat(n)

```

```
20     continue
30     continue
```

```
     return
end
```

```
C*****
C                                     GIBBS SAMPLER ROUTINES
C*****
```

```
     subroutine bergen
```

```
c This subroutine generates an iid random field that is "sort of"
c Bernoulli in that each site takes on 1 with probability P and -1
c with probability 1-P.
```

```
c
c Mnemonic: BERnoulli GENerator
c
c Called by: MCMLE
```

```
     integer rf
     common/dims/ nx,ny,npram
     common/misc/ nit,iseed
     common/modl/ model,layer
     common/mcrf/ rf(900,900)
     data p/.5/
```

```
c We put an extra LAYER on top, bottom, and on each side so that the
c random field we want is sitting smack in the middle. The extra
c around the edges is so that we can evaluate the local
c characteristics at the edges of the random field (the one in the
c middle).
```

```
     do 20 iy=1,ny+2*layer
       do 10 ix=1,nx+2*layer
         unif=ran(iseed)
         if(unif.le.p) then
           rf(ix,iy)=1
         else
           rf(ix,iy)=-1
         end if
       10     continue
     20     continue
```

```
     return
end
```

```
C*****
C                                     gibsam
C*****
```

```
c This subroutine performs the Gibbs Sampler on each site in RF except
c for the edges (where there are not complete neighborhoods).
```

```
c
c Note: The subroutine LOCHAR may be written however you need to
```

```

c evaluate the local characteristics at the site (IX,IY).
c
c Mnemonic: GIBbs SAMpler
c
c Called from: MCSTAT
c
c Subroutines called: LOCHAR

      common/dims/ nx,ny,npram
      common/misc/ nit,iseed
      common/modl/ model,layer

c One iteration consists of one pass over the entire random field.

      do 30 kit=1,nit
        do 20 iy=1+layer,ny+layer
          do 10 ix=1+layer,nx+layer
            call lochar(ix,iy)
10          continue
20        continue
30      continue

      return
      end

C*****

      subroutine lochar(ix,iy)

c This subroutine evaluates the local characteristics at the specific
c site IX,IY. P1 is the probability that the site is 1.
c
c Note: This routine is very model-specific.
c
c Mnemonic: LOCal CHARacteristics
c
c Called from: GIBSAM

      double precision psi
      integer rf
      common/dims/ nx,ny,npram
      common/misc/ nit,iseed
      common/modl/ model,layer
      common/pram/ psi(5)
      common/mcrf/ rf(900,900)

c Depending on the model, calculate the appropriate neighborhood
c sum and exponent.

      if(model.eq.1) then
        isum=rf(ix-1,iy)+rf(ix+1,iy)+rf(ix,iy-1)+rf(ix,iy+1)
        expnt=psi(1)*isum
      else if(model.eq.2) then
        isum1=rf(ix,iy-1)+rf(ix,iy+1)
        isum2=rf(ix-1,iy)+rf(ix+1,iy)

```

```

    expnt=psi(1)*isum1+psi(2)*isum2
  else
    isum1=rf(ix,iy-1)+rf(ix,iy+1)+rf(ix-1,iy)+rf(ix+1,iy)
    isum2=rf(ix,iy-2)+rf(ix-1,iy-1)+rf(ix-2,iy)+rf(ix-1,iy+1)+
&      rf(ix,iy+2)+rf(ix+1,iy+1)+rf(ix+2,iy)+rf(ix+1,iy-1)
    expnt=psi(1)*isum1+psi(2)*isum2
  endif

```

c P1 is the probability that the site is 1 given the neighbors.

```

  p1=exp(expnt)/(exp(expnt)+exp(-expnt))
  unif=ran(iseed)
  if(unif.le.p1) then
    rf(ix,iy)=1
  else
    rf(ix,iy)=-1
  end if

  return
end

```

```

C*****
c                                     MINIMIZER ROUTINES
C*****

```

```

  subroutine lkhd(n,b,p)

```

c This subroutine accesses the required negative log-likelihood
c functions according to the option MCOPT. All we want from outside
c of this program is MCOPT. In order to leave the "parent" subroutine
c QNMIN reasonably intact and generalized, this routine is constructed
c differently from the others.

```

c
c Mnemonic: (negative log) LikeliHooD function
c
c Called from: QNMIN, GRAD
c
c Subroutines called: PLKHD, MCLKHD

```

```

  implicit double precision (a-h,o-z)
  dimension b(5)
  common/mcar/ mcopt,mcsam,ithone

```

c If MCOPT=0, then we are doing maximum pseudo-likelihood estimation.
c Otherwise, MCOPT should be 1, and we are doing the Monte Carlo
c thing.

```

  if(mcopt.eq.0) then
    call plkhd(n,b,p)
  else
    call mclkhd(n,b,p)
  endif

  return
end

```

C*****

subroutine grad(n,b,g,f)

c This subroutine approximates a gradient vector for the function
c computed in the subroutine LKHD.

c

c Mnemonic: GRADient

c

c Called from: QNMIN

c

c Subroutines called: LKHD

implicit double precision (a-h,o-z)

dimension b(5),x(5),g(5)

data h/.001d0/

do 10 i=1,n
x(i)=b(i)
10 continue

do 20 i=1,n
x(i)=x(i)+h
call lkhd(n,x,fh)
g(i)=(fh-f)/h
x(i)=b(i)
20 continue

return

end

C*****

subroutine qnmin(n,b,p0,nevals,ifail)

c This subroutine is a quasi-newton minimizer from Nash, "Compact
c Numerical Algorithms for Computers" (1979), alg 21.

c

c Modifications:

c B.D. Ripley, 5/1980

c R.L. Smith, 4/1988

c P.L. Seymour 10/1991

c

c N Dimension of parameter B.

c B Parameter to be returned which minimizes the function in LKHD.

c P0 Value of pseudo-likelihood at B.

c NEVALS Maximum number of function evaluations.

c IFAIL Failure indicator (0 = OK)

c

c Mnemonic: Quasi-Newton MINimizer.

c

c Called from: MPLE, MCMLE

c

c Subroutines called: LKHD, GRAD

```
implicit double precision(a-h,o-z)
dimension b(5),h(5,5)
dimension x(5),c(5),g(5),t(5)
double precision k
integer count
data w/.2d0/,tol/1.d-4/,eps/1.d-6/
```

c If N is out of range, exit.

```
if(n.lt.0.or.n.gt.5) goto 160
```

c Is this an infeasible point? If so, exit.

```
call lkhd(n,b,p0)
if(p0.gt.1.d9) goto 180
```

c Otherwise, calculate the gradient and increment IFN, the number of function evaluations, and IG, the number of gradient calculations.

```
call grad(n,b,g,p0)
ifn=n+1
ig=1
```

c Reset Hessian.

```
10  do 30 i=1,n
      do 20 j=1,n
        h(i,j)=0.d0
20  continue
      h(i,i)=1.d0
30  continue
    ilast=ig
```

c Top of iteration.

c Store parameter and gradient.

```
40  do 50 i=1,n
      x(i)=b(i)
      c(i)=g(i)
50  continue
```

c Find search direction t

```
d1=0.d0
sn=0.d0
do 70 i=1,n
  s=0.d0
  do 60 j=1,n
    s=s-h(i,j)*g(j)
60  continue
  t(i)=s
  sn=sn+s*s
  d1=d1-s*g(i)
```

```

70    continue

c Check if downhill.

      if(d1.le.0.d0.and.ilast.eq.ig) goto 150
      if(d1.le.0.d0) goto 10

c Search along T.

      sn=.5d0/dsqrt(sn)
      k=dmin1(1.d0,sn)
80    count=0
      do 90 i=1,n
         b(i)=x(i)+k*t(i)
         if(dabs(b(i)-x(i)).lt.eps) count=count+1
90    continue

c Check if converged.  If they are all "close", exit.

      if(count.eq.n) goto 150

c Otherwise . . .

      call lkhd(n,b,p)
      ifn=ifn+1
      if(ifn.ge.nevals) goto 170
      if(p.lt.p0-d1*k*tol) goto 100
      k=w*k
      goto 80

c New lowest value.

100   p0=p
      ig=ig+1
      call grad(n,b,g,p)
      ifn=ifn+n

c update hessian

      d1=0.d0
      do 110 i=1,n
         t(i)=k*t(i)
         c(i)=g(i)-c(i)
         d1=d1+t(i)*c(i)
110   continue

c check if +ve def addition

      if(d1.le.0.d0) goto 10
      d2=0.d0
      do 130 i=1,n
         s=0.d0
         do 120 j=1,n
            s=s+h(i,j)*c(j)
120   continue

```

```

        x(i)=s
        d2=d2+s*c(i)
130   continue
        d2=1+d2/d1
        do 140 i=1,n
            do 140 j=1,n
                h(i,j)=h(i,j)-(t(i)*x(j)+t(j)*x(i)-d2*t(i)*t(j))/d1
140   continue

c Top of iteration.

        goto 40

c Successful conclusion.

150   ifail=0
        return

c N out of range.

160   ifail=1
        return

c Too many function evaluations.

170   ifail=2
        return

c Initial point infeasible.

180   ifail=3
        return
end

```

References

- C. O. Acuna (1992). Texture modeling using Gibbs distributions. *Computer Vision, Graphics, and Image Processing: Graphical Models and Image Processing* **54** 210-222.
- H. Akaike (1969). Fitting autoregressive models for prediction. *Annals of the Institute of Statistical Mathematics* **21** 243-247.
- H. Akaike (1973). Information theory and an extension of the maximum likelihood principle. *Proceedings of the 2nd International Symposium on Information Theory* (B. N. Petrov and F. Csaki, eds.) 267-281. Akademiai Kiado, Budapest.
- H. Akaike (1974). A new look at statistical model identification. *IEEE Transactions on Automatic Control* **19** 716-723.
- H. Akaike (1978). A Bayesian analysis of the minimum AIC procedure. *The Annals of the Institute of Statistical Mathematics, Part A* **30** 9-14.
- D. Aldous (1993). Approximate counting via Markov chains. *Statistical Science* **8** 16-19.
- O. Barndorff-Nielsen (1978). *Information and Exponential Families in Statistical Theory*. John Wiley & Sons, New York.
- D. Bertsimas and J. Tsitsiklis (1993). Simulated Annealing. *Statistical Science* **8** 10-15.
- J. Besag (1974). Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society, Series B* **6** 192-236.
- J. Besag (1986). On the statistical analysis of dirty pictures (with discussion). *Journal of the Royal Statistical Society, Series B* **48** 259-302.
- J. Besag (1989). Towards Bayesian image analysis. *Journal of Applied Statistics* **16** 395-407.
- J. Besag and P. J. Green (1993). Spatial statistics and Bayesian computation. *Journal of the Royal Statistical Society, Series B* **55** 24-38.
- J. Besag, J. York, and A. Mollié (1991). Bayesian image restoration, with two applications in spatial statistics (with discussion). *Annals of the Institute of Statistical Mathematics* **43** 1-59.
- P. J. Bickel and K. A. Doksum (1977). *Mathematical Statistics: Basic Ideas and Selected Topics*. Holden-Day, Oakland, California.
- P. Billingsley (1986). *Probability and Measure* (second edition). John Wiley and Sons, New York.
- P. J. Brockwell and R. A. Davis (1987). *Time Series: Theory and Methods*. Springer-Verlag, New York.
- P. Brodatz (1966). *Textures: A Photographic Album for Artists and Designers*. Dover, New York.
- L. D. Brown (1986). *Fundamentals of Statistical Exponential Families with Applications in Statistical Decision Theory*. IMS Lecture Notes - Monograph Series **9**.

- F. Cométs (1992). On consistency of a class of estimators for exponential families of Markov random fields on the lattice. *Annals of Statistics* **20** 455-468.
- F. Cométs and B. Gidas (1992). Parameter estimation for Gibbs distributions from partially observed data. *Annals of Applied Probability* **2** 142-170.
- G. Cross and A. Jain (1983). Markov random field texture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **5** 25-39.
- H. Derin and H. Elliott (1987). Modeling and segmentation of noisy and textured images. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **9** 39-55.
- R. L. Dobrushin (1968a). The description of a random field by means of conditional probability and conditions of its regularity. *Theory of Probability and its Applications* **13** 197-224.
- R. L. Dobrushin (1968b). Gibbsian random fields for lattice systems with pairwise interactions. *Functional Analysis and its Applications* **2** 292-301.
- R. L. Dobrushin (1968c). The problem of uniqueness of a Gibbs random field and the problem of phase transition. *Functional Analysis and its Applications* **2** 302-312.
- R. S. Ellis (1985). *Entropy, Large Deviations, and Statistical Mechanics*. Springer-Verlag, New York.
- A. Erdélyi (1956). *Asymptotic Expansions*. Dover Publications, New York.
- D. Geman (1991). Random fields and inverse problems in imaging. *Lecture Notes in Mathematics* **1427** 113-193. Springer-Verlag, New York.
- D. Geman, S. Geman, C. Graffigne, and P. Dong (1990). Boundary detection by constrained optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **12** 609-628.
- S. Geman and D. Geman (1984). Stochastic relaxation, Gibbs distributions, and Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **6** 721-741.
- S. Geman, D. Geman, and C. Graffigne (1987). Locating texture and object boundaries. *Pattern Recognition Theory and Applications* (P. A. Devijver and J. Kittler, editors). Springer-Verlag, New York.
- S. Geman and C. Graffigne (1986). Markov random field image models and their applications to computer vision. *Proceedings of the International Congress of Mathematicians* 1496-1517. Berkeley, California.
- H. O. Georgii (1988). *Gibbs Measures and Phase Transitions*. Walter de Gruyter, Berlin-New York.
- C. J. Geyer and E. A. Thompson (1992). Constrained Monte Carlo maximum likelihood for dependent data. *Journal of the Royal Statistical Society, Series B* **54** 657-699.
- J. W. Gibbs (1902). *Elementary Principles of Statistical Mechanics*. Yale University Press.
- B. Gidas (1986). Consistency of maximum likelihood and pseudo-likelihood estimators for Gibbs distributions. *Proceedings of the Workshop on Stochastic Differential Systems with*

Applications in Electrical/Computer Engineering, Control Theory, and Operations Research. Institute of Mathematics and its Applications, University of Minnesota.

- B. Gidas (1993). Parameter Estimation for Gibbs Distributions from Fully Observed Data. *Markov Random Fields: Theory and Applications* (R. Chellappa and A. Jain, editors) 471-498. Academic Press, New York.
- W. R. Gilks, D. G. Clayton, D. J. Spiegelhalter, N. G. Best, A. J. McNeil, L. D. Sharples, and A. J. Kirby (1993). Modelling complexity: applications of Gibbs sampling in medicine. *Journal of the Royal Statistical Society, Series B* 55 39-52.
- C. Graffigne (1987). Experiments in texture analysis and segmentation. Ph.D. dissertation, Division of Applied Mathematics, Brown University.
- U. Grenander (1983). Tutorial in Pattern Theory. Division of Applied Mathematics, Brown University.
- U. Grenander (1989). Advanced pattern theory. *Annals of Statistics* 17 1-30.
- E. J. Hannan (1980). The estimation of the order of an ARMA process. *Annals of Statistics* 8 1071-1081.
- E. J. Hannan and B. G. Quinn (1979). The determination of the order of an autoregression. *Journal of the Royal Statistical Society, Series B* 41 190-195.
- M. Hassner and J. Sklansky (1978). Markov random fields of digitized image texture. *Proceedings of the IEEE Computer Society Conference on Pattern Recognition and Image Processing* 346-351.
- M. Hassner and J. Sklansky (1980). The use of Markov random fields as models of texture. *Computer Graphics and Image Processing* 12 357-370.
- D. M. A. Haughton (1988). On the choice of a model to fit data from an exponential family. *Annals of Statistics* 16 342-355.
- R. Hu and M. M. Fahmy (1992). Texture segmentation based on a hierarchical Markov random field model. *IEEE Transactions on Signal Processing* 26 285-305.
- E. Ising (1925). Beitrag zur Theorie des Ferromagnetismus. *Zeitschrift Physik* 31 253-258.
- C. Ji (1990). Sieve estimators for pair-interaction potentials and local characteristics in Gibbs random fields. Institute of Statistics Mimeo Series #2037, Department of Statistics, University of North Carolina at Chapel Hill.
- C. Ji and L. Seymour (1991). On the selection of Markov random field texture models. Institute of Statistics Mimeo Series #2062, Department of Statistics, University of North Carolina at Chapel Hill.
- S. Karlin and H. M. Taylor (1975). *A First Course in Stochastic Processes*. Academic Press, New York.
- A. Karr (1991). Statistical models and methods in image analysis: A survey. *Inference for Stochastic Processes* (I. V. Basawa and N. U. Prabhu, editors). Marcel Dekker.
- R. Kashyap and R. Chellappa (1983). Estimation and choice of neighbors in spatial-interaction

- models of images. *IEEE Transactions on Information Theory* **29** 60-72.
- O. E. Lanford and D. Ruelle (1969). Observables at infinity and states with short range correlations in statistical mechanics. *Communications in Mathematical Physics* **13** 194-215.
- N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller (1953). Equations of state calculations by fast computing machines. *Journal of Chemical Physics* **21** 1087-1092.
- J. W. Modestino and J. Zhang (1992). A Markov random field model-based approach to image interpretation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **14** 606-615.
- J. L. Nash (1979). *Compact Numerical Algorithms for Computers*. Adam Hilger.
- E. Parzen (1974). Some recent advances in time series modelling. *IEEE Transactions on Automatic Control* **19** 723-730.
- A. Possolo (1991). Subsampling a random field. *Spatial Statistics and Imaging* (A. Possolo, editor). IMS Lecture Notes - Monograph Series **20** 286-294.
- B. D. Ripley (1988). *Statistical Inference for Spatial Processes*. Cambridge University Press, Cambridge-New York.
- J. Rissanen (1984). Stochastic complexity. *Journal of the Royal Statistical Society, Series B* **49** 223-239.
- A. Rosenfeld (1993). Image Modeling during the 1980's: A Brief Overview. *Markov Random Fields: Theory and Applications* (R. Chellappa and A. Jain, editors) 1-10. Academic Press, New York.
- H. L. Royden (1988). *Real Analysis* (third edition). Macmillan Publishing Company, New York.
- H. Rubin and J. Sethuraman (1965). Probabilities of moderate deviations. *Sankhyā, Series A* **27** 325-346.
- D. Ruelle (1978). *Thermodynamic Formalism*. Addison-Wesley, Reading, Massachusetts.
- G. Schwarz (1978). Estimating the dimension of a model. *The Annals of Statistics* **6** 461-464.
- S. Sherman (1973). Markov random fields and Gibbs random fields. *Israel Journal of Mathematics* **14** 92-103.
- R. Shibata (1980). Asymptotically efficient selection of the order of the model for estimating parameters of a linear process. *Annals of Statistics* **8** 147-164.
- B. Simon (1979). A remark on Dobrushin's uniqueness theorem. *Communications in Mathematical Physics* **68** 183-185.
- A. F. M. Smith and G. O. Roberts (1993). Bayesian Computation via the Gibbs sampler and related Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society, Series B* **55** 3-23.
- K. Smith and M. Miller (1990). A Bayesian approach incorporating Rissanen complexity for

learning Markov random field texture models. *Proceedings of the 15th International Conference on Acoustics, Speech, and Signal Processing* **4** 2317-2320.

F. Spitzer (1971). Markov random fields and Gibbs ensembles. *American Mathematical Monthly* **78** 142-154.

M. Tuceryan and A. K. Jain (1992). Texture Analysis. To appear in *The Handbook of Pattern Recognition and Computer Vision* (C. H. Chen, L. F. Pau, and P. S. P. Wang, editors). World Scientific Publishing Company.

M. Woodroffe (1982). On model selection and the arcsine laws. *Annals of Statistics* **10** 1182-1194.