Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Mathematical Sciences

Marte Fjelltveit

Image Analysis

Bayesian Inversion in Hidden Markov Models

Master's thesis in Natural Science with Teacher Education
Supervisor: Karl Henning Omre

January 2021

NTNU
Norwegian University of
Science and Technology

Marte Fjelltveit

Image Analysis

Bayesian Inversion in Hidden Markov Models

Master's thesis in Natural Science with Teacher Education
Supervisor: Karl Henning Omre
January 2021

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Mathematical Sciences

NTNU
Norwegian University of
Science and Technology

Preface

This report is my MSc Thesis (MA3950, 30 stp) in my study program in natural science with teacher education. The study has been carried out during the year 2020, mainly the fall, at the Department of Mathematical Sciences, NTNU.

I would like to thank my supervisor Professor Henning Omre for his guidance and advises throughout the study. Thank you for motivating conversations and useful feedback, I am grateful to have you as my supervisor. I would also like to thank my co-students whose conversations and meetings have been much appreciated during a challenging year affected by a pandemic. Thank you to my parents and my brother for spell checking the report. Finally, I would like to thank my fiancé Benjamin for great support and our son Felix for always brightening my days.

Abstract

Image analysis is performed by Bayesian inversion in hidden Markov models. The recursive reverse algorithm which allows computing the posterior model of a Markov random profile directly is introduced and demonstrated by examples of Markov random profiles. The recursive reverse algorithm is used in an iterative profile block Gibbs algorithm that focuses on one random row or column in a Markov random field iteratively. The algorithm efficiency is compared to the regular single-site Gibbs algorithm, focusing on one random grid node in a Markov random field iteratively, using examples of Markov random fields and a brain MRI image.

The study reveals that for smaller images, there is little to no difference in the algorithm efficiency of the two algorithms. For larger images, there is a major difference in the algorithm efficiency in favor of the profile block Gibbs algorithm. The results from the study encourage more research on the subject.

Sammendrag

I denne studien blir Bayesiansk inversjon i skjulte Markov modeller brukt til å analysere bilder. Den rekursive bakvendte algoritmen, som regner ut sannsynlighetene i posterior-modellen til en Markov random profil direkte, er introdusert og bruken av algoritmen er demonstrert ved eksempler av Markov random profiler. Den bakvendte algoritmen er brukt i en profil block Gibbs algoritme som fokuserer på en tilfeldig rad eller kolonne i et Markov random felt iterativt. Effektiviteten til algoritmen er sammenlignet med effektiviteten til den mer vanlige single-site Gibbs algoritmen som fokuserer på en enkelt node i et Markov random felt, ved å se på eksempler fra Markov random felt og et medisinsk bilde av en hjerne.

Studien avslører at for mindre bilder er det liten eller ingen forskjell i algoritmeeffektiviteten til de ulike algoritmene. For større bilder er det derimot en stor forskjell i algoritmeeffektiviteten i favør til profil block Gibbs algoritmen. Resultatene oppmuntrer til videre forskning på emnet.

Contents

1 Introduction **1**

2 Markov models and notations **3**

2.1 Definitions and notations . . . 3
2.2 Bayesian inversion in a hidden Markov model . . . 4
2.3 Likelihood model . . . 5
2.4 Prior model . . . 5
2.5 Posterior model . . . 6
2.6 Parameter estimation . . . 8

3 Markov random profile **9**

3.1 Likelihood model . . . 9
3.2 Prior model . . . 9
3.3 Posterior model . . . 12
3.4 Calculating the posterior model . . . 14
3.5 Simulating realizations and predictions . . . 16
3.5.1 Simulating from the posterior model . . . 17
3.5.2 Maximum posterior predictor . . . 17
3.5.3 Marginal maximum posterior predictor . . . 18
3.6 Estimating the parameter . . . 19
3.7 Examples of Markov random profiles . . . 20
3.7.1 Short Markov random profile . . . 20
3.7.2 Long Markov random profile . . . 29

4 Markov random field **31**

4.1 Likelihood model . . . 31
4.2 Prior model . . . 31
4.3 Posterior model . . . 34
4.4 Simulating realizations and predictions . . . 36
4.5 Estimating the parameters . . . 38
4.6 Examples of Markov random field . . . 38
4.6.1 Example 1 . . . 39
4.6.2 Example 2 . . . 45

5 Applications to brain MRI **53**

6 Conclusion **61**

References **63**

1 Introduction

My study focus is on becoming a teacher at junior high school or high school, and the theme in this study is not directly relevant to what I will teach my future students. However, as a teacher I find it crucial to be passionate about the subject I will be teaching, and I believe having an MSc Thesis on mathematics will help me inspire the students. When they ask me "what can I do with higher level mathematics?", I will be able to refer to my thesis and have a real example from my own experience. I missed a good answer to the stated question from my teachers when I went to high school myself. Also, this thesis has helped me improve my programming skills and I have been working more with algorithms, which is very relevant for the current math curriculum in school today. The new teaching plan, "Kunnskapsløftet 2020", has programming and algorithm as core elements in mathematics [Udir, 2020], but there is not much computing implemented in the education plan for the future teachers. I therefore find it very useful to have more experience within both programming and algorithmic thinking. It is important for me to help my students to achieve their full potential in the subject. I want to be able to inspire and help my students to expand their mathematical horizon, no matter which academic start-point they have.

In this study, the focus is on discretized hidden Markov models (HMM) as Bayesian inversion, which combines the likelihood model and the prior model to achieve the posterior model. The product of the likelihood model and the prior model is proportional to the posterior model which is the ultimate solution in Bayesian inversion. The posterior model can in one dimension be assessed by the reverse algorithm, which is a recursive algorithm passing backwards through the HMM once. Marginal maximum a posteriori (MMAP), maximum a posteriori (MAP), probability maps and parameter estimation are generated by algorithms presented in this study. The reverse algorithm can further improve simulation of HMMs in two dimensions using the block Gibbs algorithm of a profile from a two-dimensional image. One example from a real case study can be found in [Fjeldstad et al., 2020].

The history concerning Markov models are provided by [Basharin et al., 2004] and [Kouemou, 2011]. A Markov model in its simplest form is a stochastic temporal model where the next step in the process of modeling is only influenced by the state of the current step. The Markov chain originates from the Russian mathematician Andrey Markov, and the first paper on the subject is written in 1906 [Markov, 1906]. Here Markov introduced the term "chain" where each node could occur in one out of two states, namely $\{0, 1\}$. The motivation of Markov appears to be the extension of the law of large numbers for independent observations, but there are also other topics from an earlier time, such as Brownian motion, that can be considered a Markov processes. The term "Markov chain" was used for the first time by Bernstein [Bernstein, 1927], and Kolmogorov presented a generalization with countable finite state spaces [Kolmogorov, 1931].

The Markov chain is useful in a wide variety of applications, which provides motivation to achieve more complicated models with similar properties [Kindermann and Snell, 1980]. A series of papers focusing on Markov models are published in the 1960s and 1970s, and HMMs, introduced by Baum [Baum and Eagon, 1967], have been frequently extended since. Different algorithms related to HMMs have been developed over the years. The expectation-maximization (EM) algorithm, the Baum-Welsh algorithm, and the Viterbi algorithm are examples of this [Viterbi, 1967, Baum et al.,

1970, Dempster et al., 1977]. The EM algorithm is a general iterative algorithm mostly used to find the maximum likelihood estimates of parameters in the Markov model. The general algorithm is described in [Dempster et al., 1977], but as they also point out in the paper, the algorithm has been proposed multiple times before them. The Baum-Welsh algorithm is an iterative algorithm that adjusts the model parameters in an HMM to the observations by using EM algorithm [Scott, 2002]. The algorithm is based on forward and backward recursions and computes the marginal probabilities in the HMM. The MMAP contains marginal predictions of the image, but also probability maps for the image. The Viterbi algorithm computes the most likely path of the HMM, the "Viterbi-path", and the probability of this path [Viterbi, 1967]. While many previous studies focus on the maximum marginal probability as posterior prediction, the main focus in this study is on the full posterior probability model, which provides all the information on the HMM necessary for further analysis.

The foundations for the theory of spatial Markov random fields (MRF) are introduced as a result of generalizing the Ising model [Kindermann and Snell, 1980]. An MRF is a random field that satisfies the Markov properties. For any MRF, the probability concerning the state of a node given the rest of the field depends singularly on its specified neighbors [Besag, 1974]. Using the Hammersley-Clifford theorem one can construct a valid MRF with defined clique and neighborhood systems and a fulfilled positivity condition [Besag, 1974].

As posterior models usually are very computationally demanding, iterative algorithms are used to simulate from the posterior model. A common approach to assess the posterior model is using the single-site Gibbs algorithm. In this study, a profile block Gibbs algorithm is suggested as a way to assess the posterior model. The aim is to find out if the profile block Gibbs algorithm is a more efficient way to simulate from the posterior model compared to the regular single-site Gibbs algorithm. The results from this comparison can be useful in several scientific fields, and my motivation has been medical images. In robotic surgery, the accuracy requirements in the images are high, as any errors can cause dramatic harm. It is also helpful in medical diagnosis with accurate images to avoid misdiagnosing patients. A more efficient procedure of reducing the noise in the images benefits all industries where image analysis is crucial.

We introduce some basic notations for this study. Bold low letter symbols like $\mathbf{l}$ and $\mathbf{d}$ are vectors. The vector $\mathbf{l}_{-\mathbf{x}}$ denotes the entire vector $\mathbf{l}$ apart from $l_{\mathbf{x}}$. The functions $p(\cdot)$ and $p(\cdot|\cdot)$ are probability functions. In this study we do not notationally differentiate probability density functions (pdfs) from probability mass functions (pmfs). The function $I(A)$ is the indicator function such that $I(A) = 1$ if statement A is true, and $I(A) = 0$ if statement A is false. More notation will be specified in the next section.

In Section 2 we consider general Markov models, both in one and two dimensions, and notations to make further reading easier. The structure of the models is defined and we look at the connection between the models in one and two dimensions. A Markov field is defined, and we show that the general models presented are Markov models by the Hammersley-Clifford theorem. In the next sections, we look into specific one- and two-dimensional cases where we consider closest neighbor cliques and three states. The reverse algorithm is presented in Section 3. We consider a few examples in both dimensions, and the reverse algorithm is used to simulating from a one-dimensional HMM directly and a two-dimensional HMM by the iterative profile block Gibbs algorithm. In two dimensions, we compare it to the regular single-site Gibbs algorithm. A training image is used to estimate the

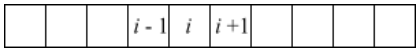
parameters in the prior distributions, and in one dimension we also estimate the parameters using marginal likelihood. Finally, we use the method on a real image of a brain MRI, again with three states, in Section 5. Section 6 summarize the study and the results.

2 Markov models and notations

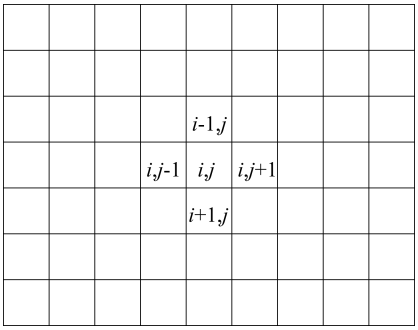
This section contains some notations related to the study. An introduction to the models, and some of the relevant properties of the models are provided.

2.1 Definitions and notations

We consider the categorical image with variables $l_{\mathbf{x}} \in \mathbb{L}$, where $\mathbb{L} = \{1, \ldots, L\}$ is a non-ordered set of states, $\mathbf{x}$ being a spatial reference in $\mathcal{D}$ and $\mathcal{D}$ being a subset of $\mathbb{R}^m$. The dimensions are $m = 1$ and $m = 2$. Assume the reference domain $\mathcal{D}$ is discretized into a regular grid $\mathcal{L}_{\mathcal{D}}$, and the variable is $\mathbf{l} = \{l_{\mathbf{x}}; \mathbf{x} \in \mathcal{L}_{\mathcal{D}} \subset \mathcal{D}\}$. Figure 1 illustrates the discretized profile in one dimension and the discretized image in two dimensions. We specify the model for each dimension. For one dimension, we consider $\mathbf{l} = \{l_i; i \in \mathcal{L}_{\mathcal{D}} \subset \mathbb{R}^1\}$ and for two dimensions, $\mathbf{l} = \{l_{i,j}; i, j \in \mathcal{L}_{\mathcal{D}} \subset \mathbb{R}^2\}$. In this study we assume the HMM id discretized into finite dimensions, hence in one dimension, $i \in \{1, \ldots, n\}$ and in two dimensions $i \in \{1, \ldots, n_1\}$ and $j \in \{1, \ldots, n_2\}$ with $n = n_1 \times n_2$.



(a) A discretized profile $\mathcal{L}_{\mathcal{D}} \subset \mathcal{D} \subset \mathbb{R}^1$.



(b) A discretized grid $\mathcal{L}_{\mathcal{D}} \subset \mathcal{D} \subset \mathbb{R}^2$.

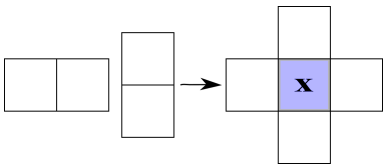
Figure 1: The discretized image in one and two dimensions.

A clique system is a system consisting of sets of nodes in $\mathcal{L}_{\mathcal{D}}$. We define the clique system $\boldsymbol{\mathcal{C}} = \{\mathbf{c}_1, \ldots, \mathbf{c}_{n_c}\}$ where $\mathbf{c} \subset \mathcal{L}_{\mathcal{D}}$. Figure 2 illustrates some examples of cliques. To ease further notation, define the set of cliques $\boldsymbol{\mathcal{C}}_{\mathbf{x}} = \{\mathbf{c} \in \boldsymbol{\mathcal{C}}; \mathbf{x} \in \mathbf{c}\}$ to be all the cliques that contains the node $\mathbf{x}$.

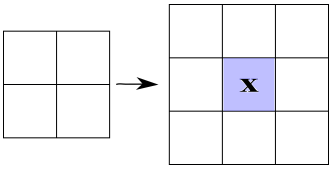
Let the neighborhood system be $\boldsymbol{\mathcal{N}} = \{\mathbf{n}_{\mathbf{x}}; \mathbf{x} \in \mathcal{L}_{\mathcal{D}}\}$, where each neighborhood $\mathbf{n}_{\mathbf{x}} = \{\mathbf{x}; \mathbf{x} \in \mathbf{c}, \mathbf{c} \in \boldsymbol{\mathcal{C}}_{\mathbf{x}}\} \backslash \mathbf{x}$ consists of all the nodes that are in a clique with $\mathbf{x}$, but not the node $\mathbf{x}$ itself. The

neighborhood system consists of sets of nodes, one for each $\mathbf{x} \in \mathcal{L}_{\mathcal{D}}$. Any model on $\mathcal{L}_{\mathcal{D}}$ with specified neighbors is a Markov field [Besag, 1974, Hurn et al., 2003].

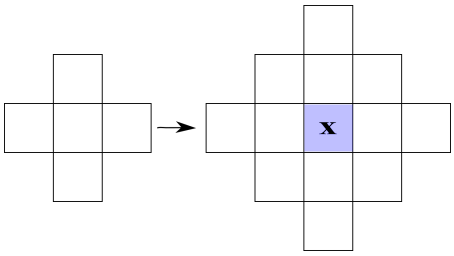
As the neighborhood system is defined by the clique system, Figure 2 displays some examples of clique and neighborhood relations. The neighborhood of the colored node $\mathbf{x}$ does not include $\mathbf{x}$ itself, but the remaining nodes in the figures. In this study we consider the clique system in Figure 2 (a) denoted the closest pairwise cliques.



(a) Four closest neighbors



(b) Eight closest neighbors



(c) Twelve closest neighbors

Figure 2: The relations between the cliques and the neighborhood defined by the cliques of an inner node $\mathbf{x}$.

The MRF is hidden, so we are not able to observe the variable $\mathbf{l}$ directly. However, we assume to have some observations, $\mathbf{d} : \{d_{\mathbf{x}}; \mathbf{x} \in \mathcal{L}_{\mathcal{D}}\}$, related to the variable $\mathbf{l} : \{l_{\mathbf{x}}; \mathbf{x} \in \mathcal{L}_{\mathcal{D}}\}$. The observations can be either real-valued or categorical. The aim is to assess $\mathbf{l}$ given $\mathbf{d}$, hence $[\mathbf{l}|\mathbf{d}]$. To do this we use Bayesian inversion.

2.2 Bayesian inversion in a hidden Markov model

Consider the prior model, a user specific base to the problem, with probability function $p(\mathbf{l})$. Further, assume the observation procedure is known, with likelihood model $p(\mathbf{d}|\mathbf{l})$. The ultimate solution in Bayesian inversion is the posterior model with probability function $p(\mathbf{l}|\mathbf{d})$ defined by Bayes rule,

$$p(\mathbf{l}|\mathbf{d}) = \text{const} \times p(\mathbf{d}|\mathbf{l})p(\mathbf{l}) \tag{1}$$

where

$$\text{const} = \Big[\sum_{\mathbf{l}' \in \mathbb{L}^n} p(\mathbf{d}|\mathbf{l}')p(\mathbf{l}') \Big]^{-1}. \tag{2}$$

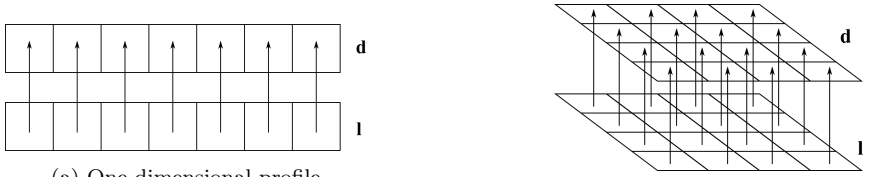
The constant is defined by summing over all possible configurations of $\mathbf{l} \in \mathbb{L}^n$, which is usually very computationally demanding depending on the size of the grid $\mathcal{L}_\mathcal{D}$. We aim at being able to assess the posterior model in an efficient way. We further consider the likelihood, prior and posterior models in general terms.

2.3 Likelihood model

Assume we observe some data $\mathbf{d} = \{d_\mathbf{x}; \mathbf{x} \in \mathcal{L}_\mathcal{D}\}$ related to the variable $\mathbf{l}$. The observation procedure is defined by the likelihood model, $p(\mathbf{d}|\mathbf{l})$, and this is the connection between the hidden random variables and the observations. In the expression $p(\mathbf{d}|\mathbf{l})$ the vector $\mathbf{d}$ is known and conditioned on, while the unknown variable of interest is $\mathbf{l}$. Hence the function $p(\mathbf{d}|\mathbf{l})$ is not a probability density function with respect to $\mathbf{l}$ and need not be normalized. Assume the likelihood model is conditionally independent with single-site response,

$$p(\mathbf{d}|\mathbf{l}) = \prod_{\mathbf{x} \in \mathcal{L}_\mathcal{D}} p(d_\mathbf{x}|l_\mathbf{x}). \tag{3}$$

The response can be states such that $\mathbf{d} \in \mathbb{L}^n$ with some known probabilities of misclassifying any $l' \in \mathbb{L}$ given some l. Alternatively the response can be $\mathbf{d} \in \mathbb{R}^n$, with $[d_i|l_i] = \mu_i(l_i) + \epsilon_{d_i}$, where $\mu_i(\cdot) \in \mathbb{R}$ and $\epsilon_{d_i} \in \mathbb{R}$ being independent errors for each d_i.



(a) One-dimensional profile

(b) Two-dimensional field

Figure 3: Each observation depend on the corresponding node in a hidden model.

Figure 3 illustrates that each observation $d_\mathbf{x}$ given $\mathbf{l}$ is only influenced by its corresponding node in the hidden variable.

2.4 Prior model

Consider the prior model related to the HMM, $p(\mathbf{l}; \boldsymbol{\theta})$, where $\boldsymbol{\theta}$ are unknown parameters suppressed in the notation. This prior model expresses the initial assumptions about the HMM. We use the Hammersley-Clifford theorem to ensure we have a valid HMM [Besag, 1974] and define the prior model on Gibbs form,

$$p(\mathbf{l}) = \text{const} \times \exp\Big\{ \sum_{\mathbf{x} \in \mathcal{L}_\mathcal{D}} \nu_\mathbf{x}(l_\mathbf{x}) + \sum_{\mathbf{c} \in \boldsymbol{\mathcal{C}}} \nu_\mathbf{c}(l_\mathbf{x}; \mathbf{x} \in \mathbf{c}) \Big\}, \tag{4}$$

where the functions $\nu_{\mathbf{x}}(\cdot, \boldsymbol{\theta})$ and $\nu_{\mathbf{c}}(\cdot, \cdot; \boldsymbol{\theta})$ are real functions, containing the parameters $\boldsymbol{\theta}$, such that the positivity condition is fulfilled. The clique system $\boldsymbol{\mathcal{C}}$ is the user specified clique system for the HMM under study. For an HMM the set of conditional marginal probabilities can be written in Markov form in terms of the neighborhood system $\boldsymbol{\mathcal{N}}$ defined by the cliques system $\boldsymbol{\mathcal{C}}$,

$$p(l_{\mathbf{x}}|\mathbf{l}_{-\mathbf{x}}) = p(l_{\mathbf{x}}|l_{\mathbf{y}}; \mathbf{y} \in \mathbf{n}_{\mathbf{x}}); \forall \mathbf{x} \in \mathcal{L}_{\mathcal{D}}. \tag{5}$$

The prior model on Gibbs form consists of one n-dimensional probability model, while the prior on Markov form consists of n one-dimensional probability models. To ensure a valid HMM by the Hammersley-Clifford theorem we define Markov formulation $p(l_{\mathbf{x}}|\mathbf{l}_{-\mathbf{x}})$ for the general prior model using the Gibbs form. Consider one particular grid node $\mathbf{x} \in \mathcal{L}_{\mathcal{D}}$,

$$\begin{aligned}
p(l_{\mathbf{x}}|\mathbf{l}_{-\mathbf{x}}) &= \frac{p(\mathbf{l})}{p(\mathbf{l}_{-\mathbf{x}})} = \frac{p(\mathbf{l})}{\sum_{l'_{\mathbf{x}} \in \mathcal{L}} p(l'_{\mathbf{x}}, \mathbf{l}_{-\mathbf{x}})} \\
&= \frac{\text{const} \times \exp\left\{ \sum_{\mathbf{y} \in \mathcal{L}_{\mathcal{D}} \setminus \mathbf{x}} \nu_{\mathbf{y}}(l_{\mathbf{y}}) + \sum_{\mathbf{c} \in \boldsymbol{\mathcal{C}} \setminus \boldsymbol{\mathcal{C}}_{\mathbf{x}}} \nu_{\mathbf{c}}(l_{\mathbf{y}}; \mathbf{y} \in \mathbf{c}) \right\}}{\text{const} \times \exp\left\{ \sum_{\mathbf{y} \in \mathcal{L}_{\mathcal{D}} \setminus \mathbf{x}} \nu_{\mathbf{y}}(l_{\mathbf{y}}) + \sum_{\mathbf{c} \in \boldsymbol{\mathcal{C}} \setminus \boldsymbol{\mathcal{C}}_{\mathbf{x}}} \nu_{\mathbf{c}}(l_{\mathbf{y}}; \mathbf{y} \in \mathbf{c}) \right\}} \\
&\times \frac{\exp\left\{ \nu_{\mathbf{x}}(l_{\mathbf{x}}) + \sum_{\mathbf{c} \in \boldsymbol{\mathcal{C}}_{\mathbf{x}}} \nu_{\mathbf{c}}(l_{\mathbf{x}}, l_{\mathbf{y}}; \mathbf{y} \in \mathbf{c} \setminus \mathbf{x}) \right\}}{\sum_{l'_{\mathbf{x}} \in \mathbb{L}} \exp\left\{ \nu_{\mathbf{x}}(l'_{\mathbf{x}}) + \sum_{\mathbf{c} \in \boldsymbol{\mathcal{C}}_{\mathbf{x}}} \nu_{\mathbf{c}}(l'_{\mathbf{x}}, l_{\mathbf{y}}; \mathbf{y} \in \mathbf{c} \setminus \mathbf{x}) \right\}} \\
&= \frac{\exp\left\{ \nu_{\mathbf{x}}(l_{\mathbf{x}}) + \sum_{\mathbf{c} \in \boldsymbol{\mathcal{C}}_{\mathbf{x}}} \nu_{\mathbf{c}}(l_{\mathbf{x}}, l_{\mathbf{y}}; \mathbf{y} \in \mathbf{c} \setminus \mathbf{x}) \right\}}{\sum_{l'_{\mathbf{x}} \in \mathbb{L}} \exp\left\{ \nu_{\mathbf{x}}(l'_{\mathbf{x}}) + \sum_{\mathbf{c} \in \boldsymbol{\mathcal{C}}_{\mathbf{x}}} \nu_{\mathbf{c}}(l'_{\mathbf{x}}, l_{\mathbf{y}}; \mathbf{y} \in \mathbf{c} \setminus \mathbf{x}) \right\}} \\
&= \text{const} \times \exp\left\{ \nu_{\mathbf{x}}(l_{\mathbf{x}}) + \sum_{\mathbf{c} \in \boldsymbol{\mathcal{C}}_{\mathbf{x}}} \nu_{\mathbf{c}}(l_{\mathbf{x}}, l_{\mathbf{y}}; \mathbf{y} \in \mathbf{c} \setminus \mathbf{x}) \right\} \\
&= p(l_{\mathbf{x}}|l_{\mathbf{y}}; \mathbf{y} \in \mathbf{n}_{\mathbf{x}}),
\end{aligned} \tag{6}$$

with normalizing constant,

$$\text{const} = \left[\sum_{l'_{\mathbf{x}} \in \mathbb{L}} \exp\left\{ \nu_{\mathbf{x}}(l'_{\mathbf{x}}) + \sum_{\mathbf{c} \in \boldsymbol{\mathcal{C}}_{\mathbf{x}}} \nu_{\mathbf{c}}(l'_{\mathbf{x}}, l_{\mathbf{y}}; \mathbf{y} \in \mathbf{c} \setminus \mathbf{x}) \right\} \right]^{-1}.$$

The normalizing constant is feasible to compute since the sum only includes L terms. The prior model on Markov form,

$$p(l_{\mathbf{x}}|\mathbf{l}_{-\mathbf{x}}) = \text{const} \times \exp\left\{ \nu_{\mathbf{x}}(l_{\mathbf{x}}) + \sum_{\mathbf{c} \in \boldsymbol{\mathcal{C}}_{\mathbf{x}}} \nu_{\mathbf{c}}(l_{\mathbf{x}}, l_{\mathbf{y}}; \mathbf{y} \in \mathbf{c} \setminus \mathbf{x}) \right\}; \forall \mathbf{x} \in \mathcal{L}_{\mathcal{D}},$$

is thereby feasible to compute.

2.5 Posterior model

As previously stated, the ultimate solution in Bayesian inversion is the posterior model in Expression 1 for $[\mathbf{l}|\mathbf{d}]$. The likelihood model is, as previously stated, a conditional independent one-to-one model.

Each node of the hidden variable depends on the nodes in its neighborhood, as illustrated in Figure 4, and this coupling must be taken into account assessing the posterior variable $[\mathbf{l}|\mathbf{d}]$.

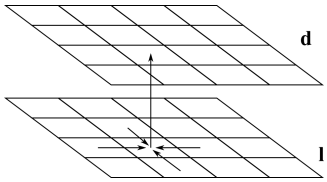
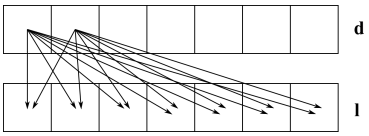
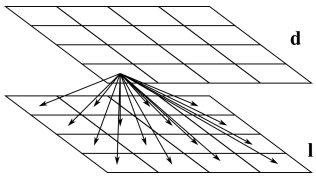


Figure 4: One node in the observation depend on its corresponding node in the hidden model which again depend on its neighborhood.

The posterior model, reversing the arrows, is then influenced by the total set of observations as illustrated in Figure 5. Only the influence of a few nodes are illustrated, but it is correspondingly for the rest of the grid nodes in the observations.



(a) One-dimensional profile



(b) Two-dimensional field

Figure 5: Each observation influences the entire posterior model. (a) two observations influences the entire posterior model in one dimension. (b) one observations influences the entire posterior model in two dimensions.

The posterior model is uniquely defined by the product of the likelihood and prior models, as in Expression 1,

$$\begin{aligned} p(\mathbf{l}|\mathbf{d}) &= \text{const} \times p(\mathbf{d}|\mathbf{l})p(\mathbf{l}) \\ &= \text{const} \times \prod_{\mathbf{x}\in\mathcal{L}_\mathcal{D}} p(d_\mathbf{x}|l_\mathbf{x}) \times \exp\Big\{ \sum_{\mathbf{x}\in\mathcal{L}_\mathcal{D}} \nu_\mathbf{x}(l_\mathbf{x}) + \sum_{\mathbf{c}\in\mathcal{C}} \nu_\mathbf{c}(l_\mathbf{x}; \mathbf{x}\in\mathbf{c})\Big\}. \end{aligned} \tag{7}$$

The posterior model is also an MRF on Gibbs form as the likelihood model can be included in the exponential by taking the logarithm,

$$p(\mathbf{l}|\mathbf{d}) = \text{const} \times \times \exp\Big\{ \sum_{\mathbf{x}\in\mathcal{L}_\mathcal{D}} \Big[\log\big(p(d_\mathbf{x}|l_\mathbf{x})\big) + \nu_\mathbf{x}(l_\mathbf{x})\Big] + \sum_{\mathbf{c}\in\mathcal{C}} \nu_\mathbf{c}(l_\mathbf{x}; \mathbf{x}\in\mathbf{c})\Big\}. \tag{8}$$

Similarly to the prior model, we define the Markov formulation of the posterior model $p(l_\mathbf{x}|\mathbf{l}_{-\mathbf{x}}, \mathbf{d})$.

Consider one particular grid node $\mathbf{x} \in \mathcal{L}_{\mathcal{D}}$,

$$
\begin{aligned}
p(l_{\mathbf{x}}|\mathbf{l}_{-\mathbf{x}}, \mathbf{d}) &= \frac{p(\mathbf{l}, \mathbf{d})}{p(\mathbf{l}_{-\mathbf{x}}, \mathbf{d})} = \frac{p(\mathbf{d}|\mathbf{l})p(\mathbf{l})}{\sum_{l'_{\mathbf{x}} \in \mathbb{L}} p(\mathbf{d}|l'_{\mathbf{x}}, \mathbf{l}_{-\mathbf{x}})p(l'_{\mathbf{x}}, \mathbf{l}_{-\mathbf{x}})} \\
&= \frac{\text{const} \times \prod_{\mathbf{y} \in \mathcal{L}_{\mathcal{D}} \setminus \mathbf{x}} p(d_{\mathbf{y}}|l_{\mathbf{y}}) \exp\Big\{ \sum_{\mathbf{y} \in \mathcal{L}_{\mathcal{D}} \setminus \mathbf{x}} \nu_{\mathbf{y}}(l_{\mathbf{y}}) + \sum_{\mathbf{c} \in \mathcal{C} \setminus \mathcal{C}_{\mathbf{x}}} \nu_{\mathbf{c}}(l_{\mathbf{y}}; l_{\mathbf{y}} \in \mathbf{c}) \Big\}}{\text{const} \times \prod_{\mathbf{y} \in \mathcal{L}_{\mathcal{D}} \setminus \mathbf{x}} p(d_{\mathbf{y}}|l_{\mathbf{y}}) \exp\Big\{ \sum_{\mathbf{y} \in \mathcal{L}_{\mathcal{D}} \setminus \mathbf{x}} \nu_{\mathbf{y}}(l_{\mathbf{y}}) + \sum_{\mathbf{c} \in \mathcal{C} \setminus \mathcal{C}_{\mathbf{x}}} \nu_{\mathbf{c}}(l_{\mathbf{y}}; l_{\mathbf{y}} \in \mathbf{c}) \Big\}} \\
&\times \frac{p(d_{\mathbf{x}}|l_{\mathbf{x}}) \exp\Big\{ \nu_{\mathbf{x}}(l_{\mathbf{x}}) + \sum_{\mathbf{c} \in \mathcal{C}_{\mathbf{x}}} \nu_{\mathbf{c}}(l_{\mathbf{x}}, l_{\mathbf{y}}; \mathbf{y} \in \mathbf{c} \setminus \mathbf{x}) \Big\}}{\sum_{l'_{\mathbf{x}} \in \mathbb{L}} p(d_{\mathbf{x}}|l'_{\mathbf{x}}) \exp\Big\{ \nu_{\mathbf{x}}(l'_{\mathbf{x}}) + \sum_{\mathbf{c} \in \mathcal{C}_{\mathbf{x}}} \nu_{\mathbf{c}}(l'_{\mathbf{x}}, l_{\mathbf{y}}; \mathbf{y} \in \mathbf{c} \setminus \mathbf{x}) \Big\}} \\
&= \text{const} \times p(d_{\mathbf{x}}|l_{\mathbf{x}}) \exp\Big\{ \nu_{\mathbf{x}}(l_{\mathbf{x}}) + \sum_{\mathbf{c} \in \mathcal{C}_{\mathbf{x}}} \nu_{\mathbf{c}}(l_{\mathbf{x}}, l_{\mathbf{y}}; \mathbf{y} \in \mathbf{c} \setminus \mathbf{x}) \Big\} \\
&= p(l_{\mathbf{x}}|l_{\mathbf{y}}; \mathbf{y} \in \mathbf{n}_{\mathbf{x}}, d_{\mathbf{x}}),
\end{aligned}
\tag{9}
$$

with normalizing constant,

$$
\text{const} = \left[\sum_{l'_{\mathbf{x}} \in \mathbb{L}} p(d_{\mathbf{x}}|l'_{\mathbf{x}}) \exp\Big\{ \nu_{\mathbf{x}}(l'_{\mathbf{x}}) + \sum_{\mathbf{c} \in \mathcal{C}_{\mathbf{x}}} \nu_{\mathbf{c}}(l'_{\mathbf{x}}, l_{\mathbf{y}}; \mathbf{y} \in \mathbf{c} \setminus \mathbf{x}) \Big\} \right]^{-1}.
$$

The constant is feasible to compute since the sum only includes L terms, where L is the number of states. The posterior model on Markov form,

$$
\begin{aligned}
p(l_{\mathbf{x}}|\mathbf{l}_{-\mathbf{x}}, \mathbf{d}) = &\left[\sum_{l'_{\mathbf{x}} \in \mathbb{L}} p(d_{\mathbf{x}}|l'_{\mathbf{x}}) \exp\Big\{ \nu_{\mathbf{x}}(l'_{\mathbf{x}}) + \sum_{\mathbf{c} \in \mathcal{C}_{\mathbf{x}}} \nu_{\mathbf{c}}(l'_{\mathbf{x}}, l_{\mathbf{y}}; \mathbf{y} \in \mathbf{c} \setminus \mathbf{x}) \Big\} \right]^{-1} \\
&\times \exp\Big\{ \nu_{\mathbf{x}}(l_{\mathbf{x}}) + \sum_{\mathbf{c} \in \mathcal{C}_{\mathbf{x}}} \nu_{\mathbf{c}}(l_{\mathbf{x}}, l_{\mathbf{y}}; \mathbf{y} \in \mathbf{c} \setminus \mathbf{x}) \Big\}; \forall \mathbf{x} \in \mathcal{L}_{\mathcal{D}},
\end{aligned}
$$

consists of n one-dimensional probability models, and is feasible to compute compared to the one n-dimensional probability model on Gibbs form.

2.6 Parameter estimation

We assume that the prior model is parameterized by unknown parameters, $\boldsymbol{\theta}$. To estimate the parameters, we consider the marginal likelihood $p(\mathbf{d}; \boldsymbol{\theta})$,

$$
p(\mathbf{d}; \boldsymbol{\theta}) = \sum_{\mathbf{l}' \in \mathbb{L}^n} p(\mathbf{d}|\mathbf{l}')p(\mathbf{l}'; \boldsymbol{\theta}). \tag{10}
$$

An estimate of the unknown parameters $\boldsymbol{\theta}$ is found by maximizing the marginal likelihood for $\mathbf{d}$,

$$
\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} \Big\{ p(\mathbf{d}; \boldsymbol{\theta}) \Big\}.
$$

Calculating $p(\mathbf{d};\boldsymbol{\theta})$ is usually very computationally demanding as it sums over all possible configurations of the hidden variable. In mathematical terms, this would mean summing over L^n terms where L is the number of states and n is the number of nodes in the field.

Another method of estimating the unknown parameters is by using a training image $\mathbf{l}^0 \in \mathbb{L}^m$. Assume the training image is defined on the same grid spacing $\mathcal{L}_\mathcal{D}$. An estimate of the parameters is then found by maximizing the prior model $p(\mathbf{l}^0;\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$,

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} \Big\{ p(\mathbf{l}^0;\boldsymbol{\theta}) \Big\}.$$

3 Markov random profile

Consider a stationary spatial profile like in Figure 1 (a) with $l_i; i = 1,\ldots,n$. Each grid node $l_i \in \mathbb{L}$ where $\mathbb{L} = \{\text{black, grey, white}\}$.

3.1 Likelihood model

The likelihood model is specified in Section 2 as a conditionally independent single-site model illustrated in Figure 3 (a). The likelihood model for the random profile is defined as follows,

$$p(\mathbf{d}|\mathbf{l}) = \prod_{i=1}^{n} p(d_i|l_i). \tag{11}$$

3.2 Prior model

The prior model of the random profile is defined on Gibbs form,

$$p(\mathbf{l}) = \text{const} \times \exp\Big\{ \sum_{i=1}^{n} \nu(l_i) + \sum_{i=2}^{n} \nu_\mathcal{C}(l_{i-1}, l_i) \Big\}. \tag{12}$$

Hence the clique system consists of the two closest neighbors as shown in Figure 6. In Expression 12, $\nu(\cdot;\boldsymbol{\theta})$ is a function of a single node, and $\nu_\mathcal{C}(\cdot,\cdot;\boldsymbol{\theta})$ is a function of one clique consisting of two closest neighbors. They are both real functions, $\nu(\cdot;\boldsymbol{\theta}), \nu_\mathcal{C}(\cdot,\cdot;\boldsymbol{\theta}) \in \mathbb{R}$.

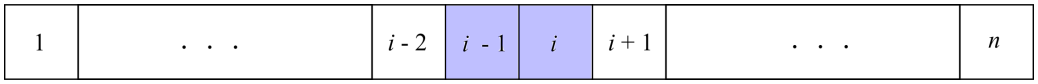


Figure 6: The stationary random profile. Colored nodes indicates one clique.

By the Markov form of the general prior model defined in Expression 6, the Markov profile form of this specified prior follows,

$$\begin{aligned} p(l_i|\mathbf{l}_{-i}) &= \text{const} \times \exp\left\{\nu(l_i) + \nu_{\mathcal{C}}(l_{i-1}, l_i) + \nu_{\mathcal{C}}(l_i, l_{i+1})\right\} \\ &= p(l_i|l_{i-1}, l_{i+1}), \quad i \in \{2 \ldots, n-1\} \\ p(l_1|\mathbf{l}_{-1}) &= \text{const} \times \exp\left\{\nu(l_1) + \nu_{\mathcal{C}}(l_1, l_2)\right\} \\ &= p(l_1|l_2) \\ p(l_n|\mathbf{l}_{-n}) &= \text{const} \times \exp\left\{\nu(l_n) + \nu_{\mathcal{C}}(l_{n-1}, l_n)\right\} \\ &= p(l_n|l_{n-1}). \end{aligned} \tag{13}$$

For any pdf,

$$p(\mathbf{l}) = p(l_1) \times \prod_{i=2}^{n} p(l_i|\mathbf{l}_{1:(i-1)}),$$

but note that $p(l_i|\mathbf{l}_{1:(i-1)})$ is not necessarily feasible to compute. Consider the Markov chain form for l_i, $i \in (2, \ldots, n)$,

$$
\begin{aligned}
p(l_i|\mathbf{l}_{1:(i-1)}) &= \frac{p(\mathbf{l}_{1:i})}{p(\mathbf{l}_{1:(i-1)})} = \frac{\sum_{l'_n\in\mathbb{L}}\cdots\sum_{l'_{i+1}\in\mathbb{L}} p(\mathbf{l}_{1:i}, l'_{i+1},\cdots,l'_n)}{\sum_{l'_n\in\mathbb{L}}\cdots\sum_{l'_i\in\mathbb{L}} p(\mathbf{l}_{1:(i-1)}, l'_i,\cdots,l'_n)} \\
&= \frac{\text{const}\times\exp\Big\{\sum_{u=1}^{i-1}\nu(l_u)+\sum_{u=2}^{i-1}\nu_{\mathcal{C}}(l_{u-1},l_u)\Big\}}{\text{const}\times\exp\Big\{\sum_{u=1}^{i-1}\nu(l_u)+\sum_{u=2}^{i-1}\nu_{\mathcal{C}}(l_{u-1},l_u)\Big\}} \\
&\times \frac{\exp\Big\{\nu(l_i)+\nu_{\mathcal{C}}(l_{i-1},l_i)\Big\}\sum_{l'_n\in\mathbb{L}}\cdots\sum_{l'_{i+1}\in\mathbb{L}}\big[\exp\Big\{\nu_{\mathcal{C}}(l_i,l'_{i+1})\Big\}}{\sum_{l'_i\in\mathbb{L}}\Big[\exp\Big\{\nu(l'_i)+\nu_{\mathcal{C}}(l_{i-1},l'_i)\Big\}\sum_{l'_n\in\mathbb{L}}\cdots\sum_{l'_{i+1}\in\mathbb{L}}\big[\exp\Big\{\nu_{\mathcal{C}}(l'_i,l'_{i+1})\Big\}} \\
&\times \frac{\exp\Big\{\sum_{u=i+1}^{n}\nu(l'_u)+\sum_{u=i+2}^{n}\nu_{\mathcal{C}}(l'_{u-1},l'_u)\Big\}\big]}{\exp\Big\{\sum_{u=i+1}^{n}\nu(l'_u)+\sum_{u=i+2}^{n}\nu_{\mathcal{C}}(l'_{u-1},l'_u)\Big\}\big]\Big]} \\
&= \frac{\exp\Big\{\nu(l_i)+\nu_{\mathcal{C}}(l_{i-1},l_i)\Big\}\sum_{l'_n\in\mathbb{L}}\cdots\sum_{l'_{i+1}\in\mathbb{L}}\big[\exp\Big\{\nu_{\mathcal{C}}(l_i,l'_{i+1})\Big\}}{\sum_{l'_i\in\mathbb{L}}\Big[\exp\Big\{\nu(l'_i)+\nu_{\mathcal{C}}(l_{i-1},l'_i)\Big\}\sum_{l'_n\in\mathbb{L}}\cdots\sum_{l'_{i+1}\in\mathbb{L}}\big[\exp\Big\{\nu_{\mathcal{C}}(l'_i,l'_{i+1})\Big\}} \\
&\times \frac{\exp\Big\{\sum_{u=i+1}^{n}\nu(l'_u)+\sum_{u=i+2}^{n}\nu_{\mathcal{C}}(l'_{u-1},l'_u)\Big\}\big]}{\exp\Big\{\sum_{u=i+1}^{n}\nu(l'_u)+\sum_{u=i+2}^{n}\nu_{\mathcal{C}}(l'_{u-1},l'_u)\Big\}\big]\Big]} \\
&= \frac{\exp\Big\{\nu(l_i)+\nu_{\mathcal{C}}(l_{i-1},l_i)\Big\}\times h(l_i)}{\sum_{l'_i\in\mathbb{L}}\exp\Big\{\nu(l'_i)+\nu_{\mathcal{C}}(l_{i-1},l'_i)\Big\}\times h(l'_i)} \\
&= \text{const}\times\exp\Big\{\nu(l_i)+\nu_{\mathcal{C}}(l_{i-1},l_i)\Big\}\times h(l_i) \\
&= p(l_i|l_{i-1}),
\end{aligned}
\tag{14}
$$

with normalizing constant,

$$
\text{const} = \left[\sum_{l'_i\in\mathbb{L}}\exp\Big\{\nu(l'_i)+\nu_{\mathcal{C}}(l_{i-1},l'_i)\Big\}\times h(l'_i)\right]^{-1}.
$$

The term $h(l_i)$ can be written backward recursively,

$$\begin{aligned} h(l_i) &= \sum_{l'_{i+1}\in\mathbb{L}} \cdots \sum_{l'_n\in\mathbb{L}} \exp\Big\{\nu(l'_{i+1}) + \nu_{\mathcal{C}}(l_i, l'_{i+1})\Big\} \exp\Big\{ \sum_{u=i+2}^{n} \nu(l'_u) + \nu_{\mathcal{C}}(l'_{u-1}, l'_u)\Big\} \\ &= \sum_{l'_{i+1}\in\mathbb{L}} \exp\Big\{\nu(l'_{i+1}) + \nu_{\mathcal{C}}(l_i, l'_{i+1})\Big\} \sum_{l'_{i+2}\in\mathbb{L}} \exp\Big\{\nu(l'_{i+2}) + \nu_{\mathcal{C}}(l_{i+1}, l'_{i+2})\Big\} \cdots \\ &\cdots \sum_{l'_{n-1}\in\mathbb{L}} \exp\Big\{\nu(l'_{n-1}) + \nu_{\mathcal{C}}(l'_{n-2}, l'_{n-1})\Big\} \sum_{l'_n\in\mathbb{L}} \exp\Big\{\nu(l'_n) + \nu_{\mathcal{C}}(l'_{n-1}, l'_n)\Big\} \\ &= \sum_{l'_{i+1}\in\mathbb{L}} \exp\Big\{\nu(l'_{i+1}) + \nu_{\mathcal{C}}(l_i, l'_{i+1})\Big\} \times h(l'_{i+1}). \end{aligned} \tag{15}$$

Notice that Expression 14 is valid for $i \in 2, \ldots, n$, so we define $p(l_1)$,

$$p(l_1) = \frac{h(l_1)}{\sum_{l'_1\in\mathbb{L}} h(l'_1)} = \text{const} \times h(l_1). \tag{16}$$

This means that $p(\mathbf{l}) = p(l_1) \times \prod_{i=2}^{n} p(l_i|\mathbf{l}_{1:(i-1)})$ can be factorized such that

$$p(\mathbf{l}) = p(l_1) \times \prod_{i=2}^{n} p(l_i|l_{i-1}).$$

The prior model can be considered a Markov random chain (MRC), where the next step in the process only depends on the current step [Norris, 1998].

3.3 Posterior model

The posterior model is uniquely defined by the prior and likelihood models, so consider Expression 11 and 12 when we define the posterior model as in Expression 7,

$$\begin{aligned} p(\mathbf{l}|\mathbf{d}) &= \text{const}^* \times p(\mathbf{d}|\mathbf{l})p(\mathbf{l}) \\ &= \text{const} \times \prod_{i=1}^{n} p(d_i|l_i) \times \exp\Big\{ \sum_{i=1}^{n} \nu(l_i) + \sum_{i=2}^{n} \nu_{\mathcal{C}}(l_{i-1}, l_i)\Big\}. \end{aligned} \tag{17}$$

By the Markov form defined for the general posterior model in Expression 9, the Markov profile form for the specified posterior model is expressed,

$$\begin{aligned}
p(l_i|\mathbf{l}_{-i},\mathbf{d}) &= \text{const} \times p(d_i|l_i)\exp\Big\{\nu(l_i)+\nu_{\mathcal{C}}(l_{i-1},l_i)+\nu_{\mathcal{C}}(l_i,l_{i+1})\Big\} \\
&= p(l_i|l_{i-1},l_{i+1},d_i), \quad i\in\{2\ldots,n-1\} \\
p(l_1|\mathbf{l}_{-1},\mathbf{d}) &= \text{const} \times p(d_1|l_1)\exp\Big\{\nu(l_1)+\nu_{\mathcal{C}}(l_1,l_2)\Big\} \\
&= p(l_1|l_2,d_1) \\
p(l_n|\mathbf{l}_{-n},\mathbf{d}) &= \text{const} \times p(d_n|l_n)\exp\Big\{\nu(l_n)+\nu_{\mathcal{C}}(l_{n-1},l_n)\Big\} \\
&= p(l_n|l_{n-1},d_n).
\end{aligned} \tag{18}$$

We know that $p(\mathbf{l}|\mathbf{d}) = p(l_1|\mathbf{d}) \times \prod_{i=2}^{n} p(l_i|\mathbf{l}_{1:(i-1)},\mathbf{d})$, but again the term $p(l_i|\mathbf{l}_{1:(i-1)},\mathbf{d})$ is not necessarily feasible to compute. Consider the Markov chain form for $p(l_i|\mathbf{l}_{1:(i-1)},\mathbf{d})$ for node $i \in (2,\ldots,n)$,

$$\begin{aligned}
p(l_i|\mathbf{l}_{1:(i-1)},\mathbf{d}) &= \frac{p(\mathbf{l}_{1:i},\mathbf{d})}{p(\mathbf{l}_{1:(i-1)},\mathbf{d})} = \frac{\sum_{l'_n\in\mathbb{L}}\cdots\sum_{l'_{i+1}\in\mathbb{L}} p(\mathbf{l}_{1:i},l'_{i+1},\cdots,l'_n,\mathbf{d})}{\sum_{l'_n\in\mathbb{L}}\cdots\sum_{l'_{i+1}\in\mathbb{L}}\sum_{l'_i\in\mathbb{L}} p(\mathbf{l}_{1:(i-1)},l'_i,l'_{i+1},\cdots,l'_n,\mathbf{d})} \\
&= \frac{\text{const}\times\prod_{u=1}^{i-1} p(d_u|l_u)\exp\Big\{\sum_{u=1}^{i-1}\nu(l_u)+\sum_{u=2}^{i-1}\nu_{\mathcal{C}}(l_{u-1},l_u)\Big\}}{\text{const}\times\prod_{u=1}^{i-1} p(d_u|l_u)\exp\Big\{\sum_{u=1}^{i-1}\nu(l_u)+\sum_{u=2}^{i-1}\nu_{\mathcal{C}}(l_{u-1},l_u)\Big\}} \\
&\times \frac{p(d_i|l_i)\exp\Big\{\nu(l_i)+\nu_{\mathcal{C}}(l_{i-1},l_i)\Big\}\times\sum_{l'_n\in\mathbb{L}}\cdots\sum_{l'_{i+1}\in\mathbb{L}}\big[\prod_{u=i+1}^{n} p(d_u|l'_u)}{\sum_{l'_i\in\mathbb{L}}\Bigg[p(d_i|l'_i)\exp\Big\{\nu(l'_i)+\nu_{\mathcal{C}}(l_{i-1},l'_i)\Big\}\sum_{l'_n\in\mathbb{L}}\cdots\sum_{l'_{i+1}\in\mathbb{L}}\big[\prod_{u=i+1}^{n} p(d_u|l'_u)} \\
&\times \frac{\exp\Big\{\nu(l'_{i+1})+\nu_{\mathcal{C}}(l_i,l'_{i+1})+\sum_{u=i+2}^{n}\nu(l'_u)+\nu_{\mathcal{C}}(l'_{u-1},l'_u)\Big\}\big]}{\exp\Big\{\nu(l'_{i+1})+\nu_{\mathcal{C}}(l'_i,l'_{i+1})+\sum_{u=i+2}^{n}\nu(l'_u)+\nu_{\mathcal{C}}(l'_{u-1},l'_u)\Big\}\big]\Bigg]} \\
&= \frac{p(d_i|l_i)\exp\Big\{\nu(l_i)+\nu_{\mathcal{C}}(l_{i-1},l_i)\Big\}\times g(l_i,\mathbf{d}_{(i+1):n})}{\sum_{l'_i\in\mathbb{L}} p(d_i|l'_i)\exp\Big\{\nu(l'_i)+\nu_{\mathcal{C}}(l_{i-1},l'_i)\Big\}\times g(l'_i,\mathbf{d}_{(i+1):n})} \\
&= \text{const}\times p(d_i|l_i)\exp\Big\{\nu(l_i)+\nu_{\mathcal{C}}(l_{i-1},l_i)\Big\}\times g(l_i,\mathbf{d}_{(i+1):n}) \\
&= p(l_i|l_{i-1},\mathbf{d}_{i:n}),
\end{aligned} \tag{19}$$

with normalizing constant,

$$\text{const} = \left[\sum_{l'_i \in \mathbb{L}} p(d_i|l'_i) \exp\left\{ \nu(l'_i) + \nu_{\mathcal{C}}(l_{i-1}, l'_i) \right\} \times g(l'_i, \mathbf{d}_{(i+1):n}) \right]^{-1}.$$

The term $g(l_i, \mathbf{d}_{(i+1):n})$ can be written backward recursively,

$$\begin{aligned}
g(l_i, \mathbf{d}_{(i+1):n}) = & \sum_{l'_{i+1} \in \mathbb{L}} \cdots \sum_{l'_n \in \mathbb{L}} \Big[\exp\left\{ \nu(l'_{i+1}) + \nu_{\mathcal{C}}(l_i, l'_{i+1}) \right\} \\
& \times \prod_{u=i+1}^{n} p(d_u|l'_u) \exp\Big\{ \sum_{u=i+2}^{n} \nu(l'_u) + \nu_{\mathcal{C}}(l'_{u-1}, l'_u) \Big\} \Big] \\
= & \sum_{l'_{i+1} \in \mathbb{L}} p(d_{i+1}|l'_{i+1}) \exp\left\{ \nu(l'_{i+1}) + \nu_{\mathcal{C}}(l_i, l'_{i+1}) \right\} \sum_{l'_{i+2} \in \mathbb{L}} \cdots \sum_{l'_{n-1} \in \mathbb{L}} p(d_{n-1}|l'_{n-1}) \qquad (20) \\
& \times \exp\left\{ \nu(l'_{n-1}) + \nu_{\mathcal{C}}(l'_{n-2}, l'_{n-1}) \right\} \sum_{l'_n \in \mathbb{L}} p(d_n|l'_n) \exp\left\{ \nu(l'_n) \nu_{\mathcal{C}}(l'_{n-1}, l'_n) \right\} \\
= & \sum_{l'_{i+1} \in \mathbb{L}} p(d_{i+1}|l'_{i+1}) \exp\left\{ \nu(l'_{i+1}) + \nu_{\mathcal{C}}(l_i, l'_{i+1}) \right\} \times g(l'_{i+1}, \mathbf{d}_{(i+2):n}).
\end{aligned}$$

Consider $p(l_1|\mathbf{d})$ separately,

$$p(l_1|\mathbf{d}) = \frac{p(d_1|l_1) \times g(l_1, \mathbf{d}_{2:n})}{\sum_{l'_1 \in \mathbb{L}} p(d_1|l'_1) g(l'_1, \mathbf{d}_{2:n})} = \text{const} \times p(d_1|l_1) \times g(l_1, \mathbf{d}_{2:n}). \qquad (21)$$

Combining this with the fact that $p(\mathbf{l}|\mathbf{d}) = p(l_1|\mathbf{d}) \times \prod_{i=2}^{n} p(l_t|\mathbf{l}_{1:(i-1)}, \mathbf{d})$,

$$p(\mathbf{l}|\mathbf{d}) = p(l_1|\mathbf{d}) \times \prod_{i=2}^{n} p(l_i|l_{i-1}, \mathbf{d}_{i:n}). \qquad (22)$$

This means that also the posterior can be considered an MRC [Norris, 1998], and by calculating the different transition probabilities we can simply simulate directly from the posterior model using the calculated probabilities.

3.4 Calculating the posterior model

The calculations have provided a first ordered MRC in Expression 22. The different transitions probabilities $p(l_i|l_{i-1}, \mathbf{d}_{i:n})$ can be assessed recursively, starting at the end point $p(l_n|l_{n-1}, d_n)$ [Moja et al., 2019],

$$
\begin{aligned}
p(l_i|l_{i-1}, l_{i+1}, d_i) &= \text{const} \times p(l_{i+1}|l_i, \mathbf{d}_{(i+1):n}) p(l_i|l_{i-1}, \mathbf{d}_{i:n}) \\
&\Updownarrow \\
p(l_i|l_{i-1}, \mathbf{d}_{i:n}) &= \text{const} \times \frac{p(l_i|l_{i-1}, l_{i+1}, d_i)}{p(l_{i+1}|l_i, \mathbf{d}_{(i+1):n})} \\
&\Updownarrow \\
p(l_i|l_{i-1}, \mathbf{d}_{i:n}) &= \text{const} \times \frac{p(l_i|l_{i-1}, l_{i+1}) p(d_i|l_i)}{p(l_{i+1}|l_i, \mathbf{d}_{(i+1):n})}.
\end{aligned} \tag{23}
$$

The normalizing constant is feasible to compute,

$$
\text{const} = \left[\sum_{l'_i \in \mathbb{L}} \frac{p(l'_i|l_{i-1}, l_{i+1}) p(d_i|l'_i)}{p(l_{i+1}|l'_i, \mathbf{d}_{(i+1):n})} \right]^{-1}.
$$

Further consider the end points l_1 and l_n,

$$
\begin{aligned}
p(l_1|l_2, d_1) &= \text{const} \times p(l_2|l_1, \mathbf{d}_{2:n}) p(l_1|\mathbf{d}) \\
&\Updownarrow \\
p(l_1|\mathbf{d}) &= \text{const} \times \frac{p(l_1|l_2, d_1)}{p(l_2|l_1, \mathbf{d}_{2:n})} \\
&\Updownarrow \\
p(l_1|\mathbf{d}) &= \text{const} \times \frac{p(l_1|l_2) p(d_1|l_1)}{p(l_2|l_1, \mathbf{d}_{2:n})},
\end{aligned} \tag{24}
$$

$$
p(l_n|l_{n-1}, d_n) = \text{const} \times p(l_n|l_{n-1}) p(d_n|l_n). \tag{25}
$$

The posterior model can then be computed recursively starting at l_n, using Algorithm 1.

Algorithm 1: Reverse Algorithm - Calculating the posterior model

for *all* $l_n, l_{n-1} \in \mathbb{L}$ *and* d_n **do**
  $p(l_n|l_{n-1}, d_n) = \text{const} \times p(d_n|l_n)p(l_n|l_{n-1})$
end
$\text{const} = \left[\sum_{l'_n \in \mathbb{L}} p(d_n|l'_n)p(l'_n|l_{n-1})\right]^{-1}$
for *all* $i = n-1, \ldots, 2$ **do**
  for *all* $l_i, l_{i-1} \in \mathbb{L}$ *and arbitrary* $l_{i+1} \in \mathbb{L}$ *and* $\mathbf{d}_{i:n}$ **do**
    $p(l_i|l_{i-1}, \mathbf{d}_{i:n}) = \text{const} \times \frac{p(d_i|l_i)p(l_i|l_{i-1}, l_{i+1})}{p(l_{i+1}|l_i, \mathbf{d}_{(i+1):n})}$
  end
  $\text{const} = \left[\sum_{l'_i \in \mathbb{L}} \frac{p(d_i|l_i)p(l'_i|l_{i-1}, l_{i+1})}{p(l_{i+1}|l'_i, \mathbf{d}_{(i+1):n})}\right]^{-1}$
end
for *all* $l_1 \in \mathbb{L}$ *and arbitrary* $l_2 \in \mathbb{L}$ *and* $\mathbf{d}$ **do**
  $p(l_1|\mathbf{d}) = \text{const} \times \frac{p(d_1|l_1)p(l_1|l_2)}{p(l_2|l_1, \mathbf{d}_{2:n})}$
end
$\text{const} = \left[\sum_{l'_1 \in \mathbb{L}} \frac{p(d_1|l_1)p(l'_1|l_2)}{p(l_2|l'_1, \mathbf{d}_{2:n})}\right]^{-1}$

Algorithm 1 calculates all the values of the posterior model,

$$p(\mathbf{l}|\mathbf{d}) = p(l_1|\mathbf{d}) \prod_{i=2}^{n} p(l_i|l_{i-1}, \mathbf{d}_{i:n}),$$

for any configuration of $\mathbf{l}$.

3.5 Simulating realizations and predictions

The aim is to assess the posterior model to simulate realizations and predictions. The MAP is calculated by the Viterbi algorithm and we also consider the marginal maximum a posteriori, both as prediction and probability profile.

3.5.1 Simulating from the posterior model

With the transition matrices generated by Algorithm 1, realizations of the posterior model are generated directly.

Algorithm 2: Simulating from the posterior model

$l_1^s \leftarrow p(l_1|\mathbf{d})$
for *all* $i = 2, \ldots, n$ **do**
  $l_i^s \leftarrow p(l_i|l_{i-1}^s, \mathbf{d}_{i:n})$
end

Algorithm 2 generates one realization $\mathbf{l}^s$ from $[\mathbf{l}|\mathbf{d}]$.

3.5.2 Maximum posterior predictor

To find the maximum a posteriori (MAP) predictor, we use the Viterbi algorithm [Viterbi, 1967], finding the most likely path, $\hat{\mathbf{l}}$,

$$\hat{\mathbf{l}} = \arg\max_{\mathbf{l}} \left\{ p(\mathbf{l}|\mathbf{d}) \right\}. \tag{26}$$

Algorithm 3: The Viterbi algorithm - MAP predictor

for *all* $l_2 \in \mathbb{L}$ **do**
  $\max_{l_1} p(\mathbf{l}_{1:2}|\mathbf{d}) = \max_{l_1} \left[p(l_1|\mathbf{d}) \times p(l_2|l_1, \mathbf{d}_{2:n}) \right]$
end
for *all* $i = 2, \ldots, n-1$ **do**
  for *all* $l_{i+1} \in \mathbb{L}$ **do**
    $\max_{\mathbf{l}_{1:i}} p(\mathbf{l}_{1:(i+1)}|\mathbf{d}) = \max_{l_i} \left[\max_{\mathbf{l}_{1:(i-1)}} p(\mathbf{l}_{1:i}|\mathbf{d}) \times p(l_{i+1}|l_i, \mathbf{d}_{(i+1):n}) \right]$
  end
end
$\hat{l}_n = \arg\max_{l_n} \left[\max_{\mathbf{l}_{1:(n-1)}} p(\mathbf{l}|\mathbf{d}) \right]$
for *all* $i = n-1, \ldots, 1$ **do**
  $\hat{l}_i = \arg\max_{l_i} \left[\max_{\mathbf{l}_{1:(i-1)}} p(\mathbf{l}_{1:i}|\mathbf{d}) \times p(\hat{l}_{i+1}|l_i, \mathbf{d}_{(i+1):n}) \right]$
end

Algorithm 3 generates a MAP predictor, $\hat{\mathbf{l}}$, based on the Viterbi algorithm.

3.5.3 Marginal maximum posterior predictor

To find the marginal maximum a posteriori predictor (MMAP) we calculate the most likely result for each l_t marginally,

$$\tilde{\mathbf{l}} = \left\{ \tilde{l}_i = \arg\max_{l_i} \left\{ p(l_i|\mathbf{d}) \right\}, \forall i \in (1, \ldots, n) \right\}, \tag{27}$$

with

$$p(l_i|\mathbf{d}) = \sum_{l'_{i-1} \in \mathbb{L}} p(l_i|l'_{i-1}, \mathbf{d}) p(l'_{i-1}|\mathbf{d}). \tag{28}$$

Algorithm 4: The MMAP predictor

$\tilde{l}_1 = \arg\max_{l_1} \left[p(l_1|\mathbf{d}) \right]$
for *all* $i = 2, \ldots, n$ **do**
  for *all* $l_i \in \mathbb{L}$ **do**
    $p(l_i|\mathbf{d}) = \sum_{l'_{i-1} \in \mathbb{L}} p(l_i|l'_{i-1}, \mathbf{d}) p(l'_{i-1}|\mathbf{d})$
  end
  $\tilde{l}_i = \arg\max_{l_i} p(l_i|\mathbf{d})$
end

Algorithm 4 generates a MMAP predictor, $\tilde{\mathbf{l}}$.

Probability profiles is assessed by the MMAP predictor in Algorithm 5 as it calculates the probability of all $l_i \in \mathbb{L}$ for every grid node i.

Algorithm 5: Computing probability profiles

$\tilde{l}_1 = \arg\max_{l_1} \left[p(l_1|\mathbf{d}) \right]$
for *all* $i = 2, \ldots, n$ **do**
  for *all* $l_i \in \mathbb{L}$ **do**
    $p(l_i|\mathbf{d}) = \sum_{l'_{i-1} \in \mathbb{L}} p(l_i|l'_{i-1}, \mathbf{d}) p(l'_{i-1}|\mathbf{d})$
  end
end

Algorithm 5 generates L profiles displaying the marginal probabilities of each state $l_i \in \mathbb{L}$ occurring in every grid node i.

3.6 Estimating the parameter

The marginal likelihood model $p(\mathbf{d};\boldsymbol{\theta})$ can be used to estimate the parameters,

$$\begin{aligned} p(\mathbf{d};\boldsymbol{\theta}) &= \sum_{\mathbf{l}'\in\mathbb{L}^n} p(\mathbf{d}|\mathbf{l}')p(\mathbf{l}';\boldsymbol{\theta}) \\ &= \sum_{l'_1\in\mathbb{L}} \cdots \sum_{l'_n\in\mathbb{L}} p(d_1|l'_1)p(l'_1;\boldsymbol{\theta}) \prod_{i=2}^{n} p(d_i|l'_i)p(l'_i|l'_{i-1};\boldsymbol{\theta}) \\ &= \sum_{l'_1\in\mathbb{L}} p(d_1|l'_1)p(l'_1;\boldsymbol{\theta}) \sum_{l'_2\in\mathbb{L}} p(d_2|l'_2)p(l'_2|l'_1;\boldsymbol{\theta}) \sum_{l'_3\in\mathbb{L}} \cdots \\ &\quad \sum_{l'_{n-1}\in\mathbb{L}} p(d_{n-1}|l'_{n-1})p(l'_{n-1}|l'_{n-2};\boldsymbol{\theta}) \sum_{l'_n\in\mathbb{L}} p(d_n|l'_n)p(l'_n|l'_{n-1};\boldsymbol{\theta}). \end{aligned} \tag{29}$$

Since $\mathbf{d}$ is a set of conditionally independent data, note the following for $i = 3,\ldots,n$,

$$p(d_{i-1}|l_{i-1})p(l_{i-1}|l_{i-2};\boldsymbol{\theta})p(\mathbf{d}_{i:n}|l_{i-1};\boldsymbol{\theta}) = p(\mathbf{d}_{(i-1):n}|l_{i-1})p(l_{i-1}|l_{i-2};\boldsymbol{\theta}). \tag{30}$$

We can estimate the unknown parameters $\boldsymbol{\theta}$ by maximizing the function with respect to the parameters, $p(\mathbf{d};\boldsymbol{\theta})$,

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} \Big\{ p(\mathbf{d};\boldsymbol{\theta}) \Big\}.$$

Algorithm 6: Calculating $p(\mathbf{d};\boldsymbol{\theta})$

for *all* $l_{n-1} \in \mathbb{L}$ **do**
  $g(d_n, l_{n-1};\boldsymbol{\theta}) = \sum_{l'_n\in\mathbb{L}} p(d_n|l'_n)p(l'_n|l_{n-1};\boldsymbol{\theta})$
end
for *all* $i = n-1,\ldots,2$ **do**
  for *all* $l_{i-1} \in \mathbb{L}$ **do**
    $g(\mathbf{d}_{i:n}, l_{i-1};\boldsymbol{\theta}) = \sum_{l'_i\in\mathbb{L}} p(d_i|l'_i)p(l'_i|l_{i-1};\boldsymbol{\theta}) \times g(\mathbf{d}_{(i+1):n}, l'_i;\boldsymbol{\theta})$
  end
end
$p(\mathbf{d};\boldsymbol{\theta}) = \sum_{l'_1\in\mathbb{L}} p(d_1|l'_1)p(l'_1;\boldsymbol{\theta}) \times g(\mathbf{d}_{2:n}, l'_1;\boldsymbol{\theta})$

Algorithm 6 returns $p(\mathbf{d};\boldsymbol{\theta})$ and will further on be used to estimate $\boldsymbol{\theta}$. Recall the function $g(l_t|\mathbf{d}_{(t+1):T})$ from Expression 20. The parameters is estimated by maximizing $p(\mathbf{d};\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$. Alternatively, one may estimate the parameters by using a training image $\mathbf{l}^0$. Similarly as to using the marginal likelihood model, an estimate of the parameters is found by maximizing the prior model $p(\mathbf{l}^0;\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$,

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} \Big\{ p(\mathbf{l}^0;\boldsymbol{\theta}) \Big\}.$$

The prior model is assessed by the recursive reverse algorithm in Algorithm 7.

Algorithm 7: Reverse Algorithm - Calculating the prior model

for *all* $l_n, l_{n-1} \in \mathbb{L}$ **do**
  $p(l_n|l_{n-1}) = \text{const} \times p(l_n|l_{n-1})$
  $\text{const} = \Big[\sum_{l'_n \in \mathbb{L}} p(l'_n|l_{n-1})\Big]^{-1}$
end
for *all* $i = n-1, \ldots, 2$ **do**
  for *all* $l_i, l_{i-1} \in \mathbb{L}$ *and arbitrary* $l_{i+1} \in \mathbb{L}$ **do**
    $p(l_i|l_{i-1}) = \text{const} \times \frac{p(l_i|l_{i-1},l_{i+1})}{p(l_{i+1}|l_i)}$
  end
  $\text{const} = \Big[\sum_{l'_i \in \mathbb{L}} \frac{p(l'_i|l_{i-1},l_{i+1})}{p(l_{i+1}|l'_i)}\Big]^{-1}$
end
for *all* $l_1 \in \mathbb{L}$ *and arbitrary* $l_2 \in \mathbb{L}$ **do**
  $p(l_1) = \text{const} \times \frac{p(l_1|l_2)}{p(l_2|l_1)}$
end
$\text{const} = \Big[\sum_{l'_1 \in \mathbb{L}} \frac{p(l'_1|l_2)}{p(l_2|l'_1)}\Big]^{-1}$
$p(\mathbf{l}^0) = p(l^0_1) \prod_{i=2}^{n} p(l^0_i|l^0_{i-1})$

Algorithm 7 returns $p(\mathbf{l}^0; \boldsymbol{\theta})$ with $\mathbf{l}^0$ as input.

3.7 Examples of Markov random profiles

The examples are based on the Markov profile with Gibbs form,

$$\begin{aligned} p(\mathbf{l}) = \text{const} \times \exp\Big\{ &\sum_{i=1}^{n} \beta_1 \mathrm{I}(l_i \in \{B, W\}) + \sum_{i=2}^{n} \beta_2 \mathrm{I}(l_{i-1} = l_i) \\ &- 100\big(\mathrm{I}(l_{i-1,j} = B, l_{i,j} = W) + \mathrm{I}(l_{i-1,j} = W, l_{i,j} = B)\big)\Big\}, \end{aligned} \tag{31}$$

with $\beta_1, \beta_2 > 0$. There is a higher probability for {black, white} nodes than {grey}, and the neighbors tend to be of the same state. The prior also indicate that it is highly unlikely that a black node appears next to a white node.

3.7.1 Short Markov random profile

The profile consists of 100 grid nodes, so $n = 100$. Setting the parameters in Expression 31 to $\beta_1 = 1.00$ and $\beta_2 = 0.50$, the profile of interest $\mathbf{l}^T$ is simulated and displayed in Figure 7. No black

node occurs next to a white node, and there are more {black, white} nodes then {grey} nodes as expected.

Figure 7: The true image $\mathbf{l}^T$.

A training image is used to estimate the parameters, and in this case the training image is the true image $\mathbf{l}^T$. Figure 8 displays the values of $p(\mathbf{l}^T; \beta_1, \beta_2)$ calculated using Algorithm 7 for the different combinations of β_1, β_2 as a density plot. The variances and covariance is assessed approximately from the density plot,

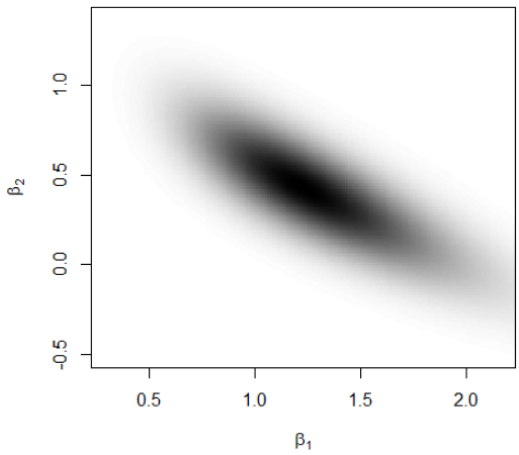


Figure 8: Density plot model of the prior with respect to the parameters.

$$\begin{aligned}
\hat{\beta}_1 &= 1.23 \\
\hat{\beta}_2 &= 0.43 \\
\hat{\mathrm{Var}}(\hat{\beta}_1) &= 0.15 \\
\hat{\mathrm{Var}}(\hat{\beta}_2) &= 0.08 \\
\hat{\mathrm{Cov}}(\hat{\beta}_1, \hat{\beta}_2) &= -0.09
\end{aligned}$$

The variance of $\hat{\beta}_1$ is slightly larger than the variance of $\hat{\beta}_2$, and the covariance is a negative, low-valued number as expected from the shape of the density in Figure 8.

Further we consider three different cases with the truth in Figure 7. The observations procedures have different likelihood models being variations of a Gaussian and misclassification likelihoods.

Gaussian likelihood, case 1

Consider the truth $\mathbf{l}^T$ in Figure 7 and a Gaussian likelihood model. As the likelihood model is a conditionally independent single-site model, we define the likelihood model for any observation $[d_i|l_i]$, $i = 1, \ldots, n$,

$$d_i|l_i \sim \text{Gauss}(\mu_{l_i}, 0.1^2)$$

$$\mu_{l_i} = \begin{cases} -1 & \text{if } l_i = \text{black} \\ 0 & \text{if } l_i = \text{grey} \\ 1 & \text{if } l_i = \text{white.} \end{cases}$$

Figure 9: Observations from a Gaussian likelihood model with $\sigma^2 = 0.1^2$.

The observations are displayed in Figure 9. The observations are very similar to the truth due to the relatively small variance in the likelihood model. The observations in this case contain a small amount of noise.

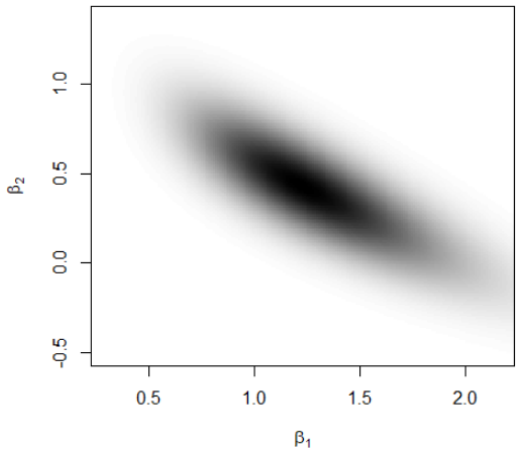


Figure 10: Density plot of the marginal likelihood model with respect to the parameters.

The marginal likelihood calculated using Algorithm 6 is used to estimate the parameters. Due to the small variance in the likelihood model, it is not surprising that the density plot for the marginal likelihood with respect to the parameters in Figure 10 is similar to the density plot in Figure 8.

Again the estimates, the variances and the covariance of the parameters is approximated,

$$\begin{aligned}\hat{\beta}_1 &= 1.23\\ \hat{\beta}_2 &= 0.43\\ \hat{\mathrm{Var}}(\hat{\beta}_1) &= 0.15\\ \hat{\mathrm{Var}}(\hat{\beta}_2) &= 0.08\\ \hat{\mathrm{Cov}}(\hat{\beta}_1, \hat{\beta}_2) &= -0.09.\end{aligned}$$

Realizations and predictions are generated using the parameters estimated from the marginal likelihood model, $\hat{\beta}_1 = 1.23$ and $\hat{\beta}_2 = 0.43$.

(a) $\mathbf{l}^{s1}$

(b) $\mathbf{l}^{s2}$

(c) $\mathbf{l}^{s3}$

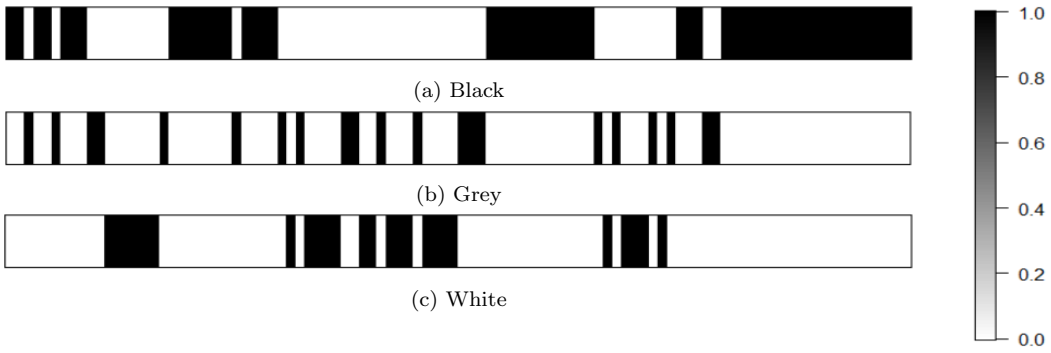
Figure 11: Three realizations generated by the posterior model.

Using Algorithm 1 and 2, realizations $\mathbf{l}^s$ are generated from the posterior model, and Figure 11 displays three realizations. The realizations are in this case exactly the same as the truth $\mathbf{l}^T$ in Figure 7, which is plausible due to the relatively small variance in the likelihood model. The MAP and MMAP are calculated using Algorithms 3 and 4 respectively and displayed in Figure 12 and 13. Also the predictions are identical to the truth, which is expected when the likelihood model contains such a small amount of noise.

Figure 12: MAP predictor by the Viterbi algorithm (Algorithm 3).

Figure 13: MMAP predictor by Algorithm 4.

The probability profiles are generated by Algorithm 5 and displayed in Figure 15. For all the different states, the probabilities in each grid node is approximately 0 or 1, which causes the realizations and the predictions to be the same, and further identical to the truth.



(a) Black

(b) Grey

(c) White

Figure 15: Probability profiles for the possible states {black, grey, white}.

When the observations contain a very small amount of noise, the precision and accuracy in the predictions and realizations are high for obvious reasons.

Gaussian likelihood, case 2

Again consider the truth $\mathbf{l}^T$ in Figure 7 and a Gaussian likelihood model. We define the likelihood model for any observation $[d_i|l_i]$, $i = 1, \ldots, n$,

$$d_i|l_i \sim \text{Gauss}(\mu_{l_i}, 0.5^2)$$

$$\mu_{l_i} = \begin{cases} -1 & \text{if } l_i = \text{black} \\ 0 & \text{if } l_i = \text{grey} \\ 1 & \text{if } l_i = \text{white.} \end{cases}$$

The only difference in this likelihood model compared to the previous one, is that the variance is larger $\sigma^2 = 0.5^2$. There are more noise in the observations, which is clear comparing the observations $\mathbf{d}$ in Figure 16 to the truth in Figure 7.

Figure 16: Observations from a Gaussian likelihood model with $\sigma^2 = 0.5^2$.

The parameters are estimated by the marginal likelihood model generated by Algorithm 6, and the density plot with respect to the parameters is displayed in Figure 17.

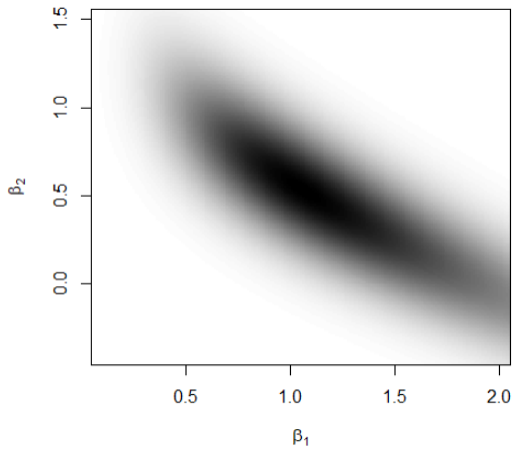
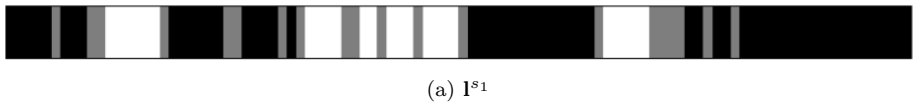


Figure 17: Density plot of the marginal likelihood model with respect to the parameters.

The parameters are estimated, and the related variances and the covariance are approximated. As the likelihood model in this example contains more noise, the density plot in Figure 17 indicates that the estimations are more uncertain, which further is confirmed by the variances and covariance,

$$\begin{aligned}\hat{\beta}_1 &= 1.06\\ \hat{\beta}_2 &= 0.53\\ \hat{\mathrm{Var}}(\hat{\beta}_1) &= 0.57\\ \hat{\mathrm{Var}}(\hat{\beta}_2) &= 0.27\\ \hat{\mathrm{Cov}}(\hat{\beta}_1, \hat{\beta}_2) &= -0.35.\end{aligned}$$

More noise in the likelihood model leads to higher uncertainty in the parameter estimates, and the variances and covariance increase. With these parameter estimates, realizations are generated from the posterior model by Algorithm 1 and 2. Three realizations are displayed in Figure 18, and although they are similar, some differences occur.



(a) $\mathbf{l}^{s_1}$

(b) $\mathbf{l}^{s_2}$

(c) $\mathbf{l}^{s_3}$

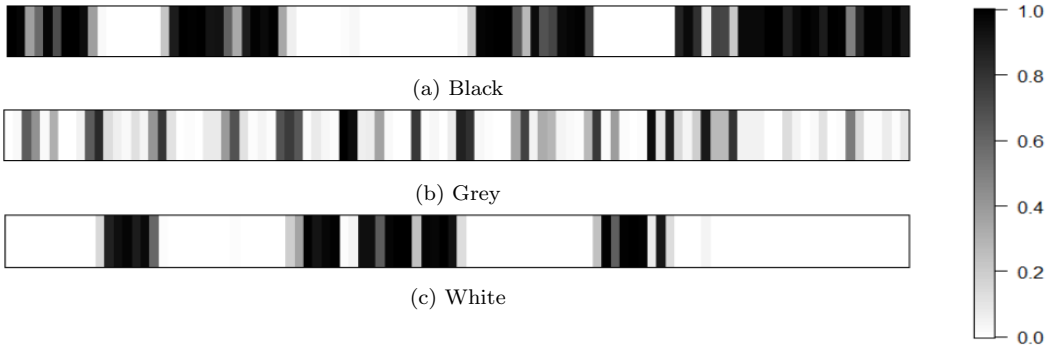
Figure 18: Three realizations of the posterior model.

The MAP and MMAP are generated by Algorithm 3 and 4 respectively and displayed in Figure 19 and 20. They are very similar, but there are a few grid nodes that differs in the two prediction plots, which is plausible as the likelihood model contains a higher uncertainty than in the previous example.

Figure 19: MAP predictor by the Viterbi algorithm.

Figure 20: MMAP predictor by Algorithm 4.

The probability profiles of the posterior model generated by Algorithm 5 is displayed in Figure 22. In comparison to the previous example, where the probability profiles consisted of approximately only 0 and 1 probabilities, we observe that there are higher uncertainty in several grid nodes.



(a) Black

(b) Grey

(c) White

Figure 22: Probability profiles for the possible states {black, grey, white}.

The predictions of the posterior model are similar to the truth, but the images are not exactly the same. Also the probability profiles generated by the MMAP indicates that there are uncertainties related to specific positions in the grid.

Misclassification likelihood

Still considering the truth $\mathbf{l}^T$ in Figure 7, the observation procedure is defined with a misclassification likelihood. The observation probabilities for any observation $[d_i|l_i]$, $i = 1, \ldots, n$ are provided in matrix P,

$$P = \begin{array}{c} \\ d_i = B \\ d_i = G \\ d_i = W \end{array} \begin{array}{c} \begin{array}{ccc} l_i = B & l_i = G & l_i = W \end{array} \\ \begin{bmatrix} 0.85 & 0.10 & 0.05 \\ 0.10 & 0.80 & 0.10 \\ 0.05 & 0.10 & 0.85 \end{bmatrix} \end{array}$$

The first column contains the probabilities when the true color of the node is black, the middle column contains the probabilities when the true color of the node is grey and the last column contains the probabilities when the true color of the node is white.

Figure 23: Observation from a misclassification based likelihood model.

The observations $\mathbf{d}$ displayed in Figure 23 consist of an "impossible" configuration, as there are black nodes next to white nodes in several occasions. This configuration is highly unlikely considering the prior in Expression 31.

As previously, the parameters are estimated by the marginal likelihood model calculated using Algorithm 6. Figure 24 displays the density plot with respect to the parameters.

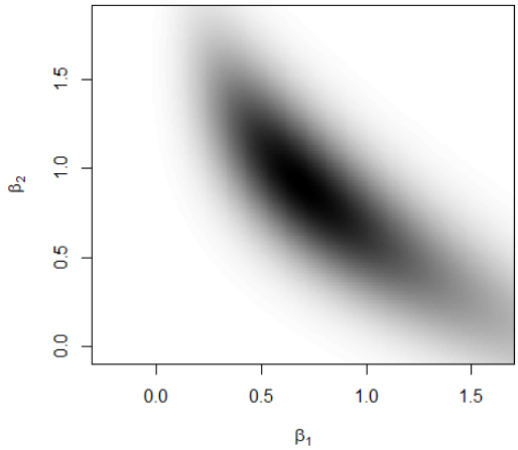


Figure 24: Density plot of the marginal likelihood model with respect to the parameters

The parameters are estimated and the related variances and covariance are computed approximately. The estimates values are different from the previous estimates generated from the truth $\mathbf{l}^T$ in Figure

7, which is plausible as the variances of the estimated parameters are relatively high,

$$\begin{aligned}\hat{\beta}_1 &= 0.70\\ \hat{\beta}_2 &= 0.91\\ \hat{\mathrm{Var}}(\hat{\beta}_1) &= 0.45\\ \hat{\mathrm{Var}}(\hat{\beta}_2) &= 0.34\\ \hat{\mathrm{Cov}}(\hat{\beta}_1, \hat{\beta}_2) &= -0.34.\end{aligned}$$

When the observation probabilities consist of values that makes the probability of misclassification relatively high, the parameter estimates will contain a correspondingly high level of uncertainty. Calculating the posterior model, we use the parameters estimated by the marginal likelihood model, that is $\hat{\beta}_1 = 0.70$ and $\hat{\beta}_1 = 0.91$.

(a) $\mathbf{l}^{s_1}$

(b) $\mathbf{l}^{s_2}$

(c) $\mathbf{l}^{s_3}$

Figure 25: Three realizations of the posterior model.

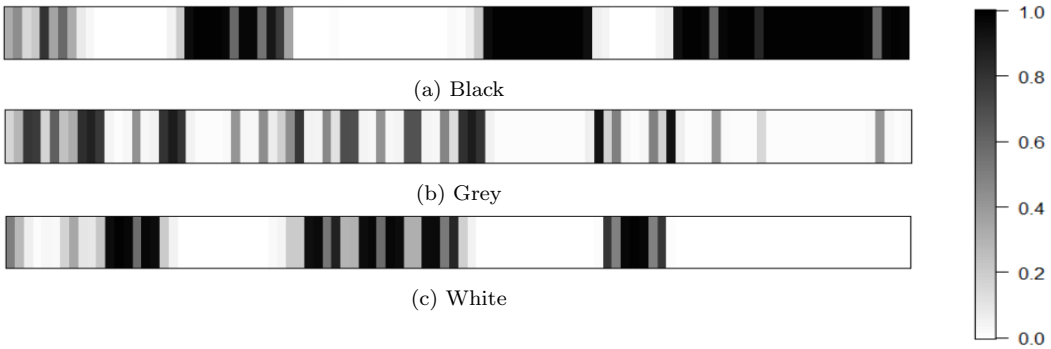
Figure 25 displays three realizations of the posterior model. Even though they look similar, there are also visible differences. Particularly in the first realization $\mathbf{l}^{s_1}$, the first left-most nodes differs considerably from the other two realizations as it does not contain any black nodes.

Figure 26 and 27 display the MAP and MMAP respectively. Since the MAP is the most likely path all over, there will be no "impossible" cliques, so in Figure 26 there are no black nodes next to white nodes. As the MMAP calculates the marginal probabilities for every grid node, there is no guarantee that we avoid "impossible" cliques. Hence in Figure 27, there are black nodes next to white nodes in two locations. As previously stated, this configuration is highly unlikely using the prior model in Expression 31.

Figure 26: MAP prediction generated by the Viterbi algorithm (Algorithm 3).

Figure 27: MMAP prediction generated by Algorithm 4.

The probability profiles generated by Algorithm 5 are displayed in Figure 29, and as the MMAP prediction indicates with "impossible" cliques, there are some relatively high uncertainties.



(a) Black

(b) Grey

(c) White

Figure 29: Probability profiles for the possible states {black, grey, white}

With a misclassification likelihood, the probability of misclassification will determine the certainty in the posterior model, including the estimated parameters. With the observation probabilities provided in matrix P, the chance of misclassification is relatively high. This provides uncertainty both considering estimating parameters and generating predictions.

3.7.2 Long Markov random profile

To examine the influence the size of the MRP has on the parameter estimates, let the profile consist of 1000 grid nodes and consider the same prior model from Expression 31. Choosing parameters $\beta_1 = 1.00$ and $\beta_2 = 0.50$, the truth $\mathbf{l}^T$ is simulated and displayed in Figure 30.

Figure 30: The true image.

The parameters are estimated using the truth $\mathbf{l}^T$ as training image and displayed in a density plot with respect to the parameters in Figure 31. Compared to previous parameter estimates for a profile

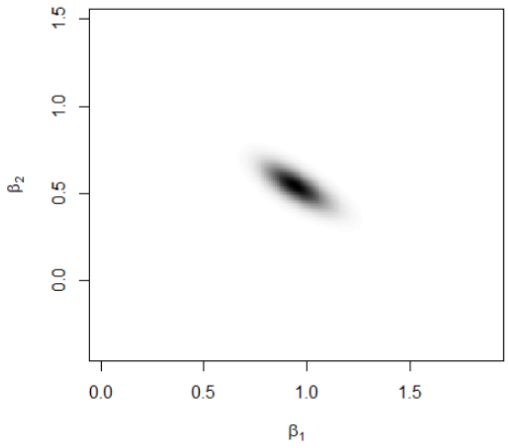


Figure 31: Density plot of the prior with respect to the parameters.

consisting of fewer grid nodes, the density plot is more focused, indicating more reliable estimates of the parameters. The variances and covariance related to the estimates are approximately calculated,

$$\begin{aligned}\hat{\beta}_1 &= 0.95\\ \hat{\beta}_2 &= 0.54\\ \hat{\mathrm{Var}}(\hat{\beta}_1) &= 0.010\\ \hat{\mathrm{Var}}(\hat{\beta}_2) &= 0.007\\ \hat{\mathrm{Cov}}(\hat{\beta}_1, \hat{\beta}_2) &= -0.006.\end{aligned}$$

The variances and covariance related to the estimates are in this case relatively small compared to the previous examples using the truth $\mathbf{l}^T$ displayed in Figure 7. The estimates $\hat{\beta}_1 = 0.95$ and $\hat{\beta}_2 = 0.54$ are close to the parameter values used in simulating the truth $\mathbf{l}^T$ in Figure 30. The low variances and covariance are not surprising considering the focused density plot in Figure 31.

The recursive reverse algorithm (Algorithm 1) provides a way to feasibly compute the posterior of an MRP. As any realization of an HMM, in that case, is generated directly from the posterior model, it will always be more efficient than an iterative algorithm. The accuracy in the results from the realizations and predictions depends on the accuracy in the observation procedure. As the predictions are calculated and not approximated, they are as reliable as the probability profiles indicate. The results from the examples are expected as a likelihood model containing a low level of noise provides accurate results both concerning the truth and the certainty in the results. A likelihood model containing a higher level of noise provides results that differ from the truth at a larger scale, and the certainty decrease correspondingly. The example of a longer MRP implies that the estimated parameters from the prior are more accurate than for a shorter MRP.

4 Markov random field

Consider a stationary spatial field like in Figure 1 (b), with $l_{i,j}; i = 1, \ldots, n_1, j = 1, \ldots, n_2, n = n_1 \times n_2$. Each grid node $l_{i,j} \in \mathbb{L}$ where $\mathbb{L} = \{\text{black, grey, white}\}$.

4.1 Likelihood model

The likelihood model is specified in Section 2 as a conditionally independent single-site model illustrated in Figure 3 (b). The likelihood model for the MRF is defined in the following expression,

$$p(\mathbf{d}|\mathbf{l}) = \prod_{i=1}^{n_1} \prod_{j=1}^{n_2} p(d_{i,j}|l_{i,j}). \tag{32}$$

4.2 Prior model

The prior of the MRF is defined on Gibbs form,

$$p(\mathbf{l}) = \text{const} \times \exp\Big\{ \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \nu(l_{i,j}) + \sum_{i=1}^{n_1} \sum_{j=2}^{n_2} \nu_h(l_{i,j-1}, l_{i,j}) + \sum_{i=2}^{n_1} \sum_{j=1}^{n_2} \nu_v(l_{i-1,j}, l_{i,j}) \Big\}. \tag{33}$$

Hence the clique system is as in Figure 2 (a), consisting of pairwise closest neighbors. Figure 32 displays one vertical and one horizontal clique in a discretized two dimensional field. There are $(n_1 - 1) \times n_2$ vertical cliques and $n_1 \times (n_2 - 1)$ horizontal cliques.

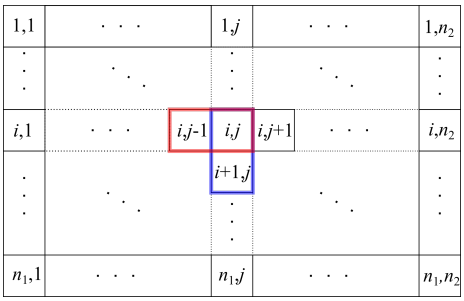


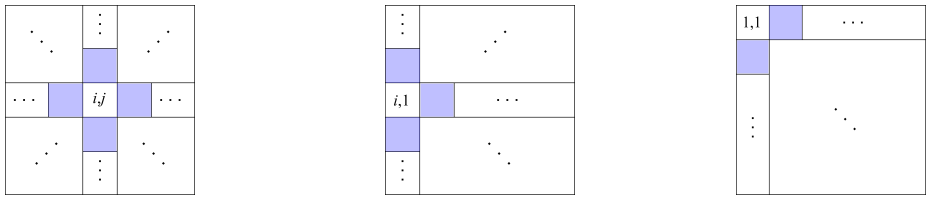
Figure 32: The stationary random field. The red indicates one horizontal clique, while the blue indicates one vertical clique.

In the prior model, $\nu(\cdot; \boldsymbol{\theta})$ is a function of a single node, $\nu_h(\cdot, \cdot; \boldsymbol{\theta})$ is a function of the horizontal clique consisting of the two closest horizontal neighbors and $\nu_v(\cdot, \cdot; \boldsymbol{\theta})$ is a function of the vertical clique consisting of the two closest vertical neighbors.

All of the grid nodes in the inner field (that excludes all nodes at the border) are in cliques with four other grid nodes. The borders, excluded the corners, are in cliques with three other grid nodes. And finally, the corners are in cliques with two other grid nodes. The neighborhoods of the different positioned grid nodes are illustrated in Figure 33.

Using definition of the Markov formulation of the general posterior in Expression 9, the Markov formulation $p(l_{i,j}|\mathbf{l}_{-i,j})$ for node i,j, $i \in \{1,\ldots,n_1\}$, $j \in \{1,\ldots,n_2\}$ is expressed,

$$p(l_{i,j}|\mathbf{l}_{-i,j}) = p(l_{i,j}|l_{\mathbf{x}}; \mathbf{x} \in \mathbf{n}_{i,j}). \tag{34}$$



(a) (b) (c)

Figure 33: The neighborhoods for different positioned grid nodes. (a) is valid for all grid nodes i,j, $i \in \{2,\ldots,n_1-1\}$, $j \in \{2,\ldots,n_2-1\}$ (b) is valid all grid nodes $i,1$, with $i \in \{2,\ldots,n_1\}$, and the neighborhood is similar in the rest of the borders except from the corners. (c) is valid for grid node $1,1$, and the neighborhood is similar in the three remaining corners.

The inner grid nodes like the one in Figure 33 (a), node i,j, $i \in \{2,\ldots,n_1-1\}$, $j \in \{2,\ldots,n_2-1\}$, are written on Markov formulation using the specified prior model in Expression 33,

$$\begin{aligned} p(l_{i,j}|\mathbf{l}_{-i,j}) = \text{const} \times \exp\Big\{ & \nu(l_{i,j}) + \nu_h(l_{i,j-1}, l_{i,j}) \\ & + \nu_v(l_{i-1,j}, l_{i,j}) + \nu_h(l_{i,j}, l_{i,j+1}) + \nu_v(l_{i,j}, l_{i+1,j}) \Big\}. \end{aligned} \tag{35}$$

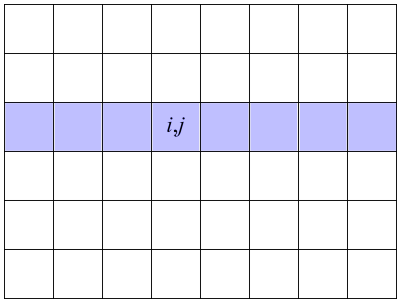
For the remaining nodes the Markov formulation is similar, but we exclude the clique functions that does not exist due to border issues. For instance, the node in Figure 33 (b) has Markov formulation,

$$p(l_{i,1}|\mathbf{l}_{-i,1}) = \text{const} \times \exp\big\{\nu(l_{i,1}) + \nu_v(l_{i-1,1}, l_{i,1}) + \nu_h(l_{i,1}, l_{i,2}) + \nu_v(l_{i,1}, l_{i+1,1})\big\},$$

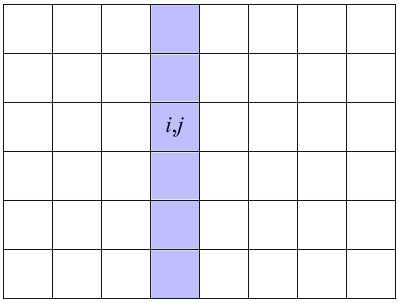
while the node in Figure 33 (c) has Markov formulation,

$$p(l_{1,1}|\mathbf{l}_{-1,1}) = \text{const} \times \exp\big\{\nu(l_{1,1}) + \nu_h(l_{1,1}, l_{1,2}) + \nu_v(l_{1,1}, l_{2,1})\big\}.$$

Assessing the prior model, and further the posterior model, is more challenging in two dimensions in comparison to one dimension, where it could be calculated using the recursive reverse algorithm (Algorithm 1 and 7). Profile block Gibbs algorithm is a suggested approach to assess the model by considering the profiles as blocks, and the efficiency of the algorithm is compared to the efficiency of the regular single-site Gibbs algorithm where we consider one random grid node at a time.



(a) The row $\mathbf{l}_{i,\cdot}$ is colored



(b) The column $\mathbf{l}_{\cdot,j}$ is colored

Figure 34: A two dimensional grid. (a) illustrates all the nodes included in $\mathbf{l}_{i,\cdot}$, (b) illustrates all the nodes included in $\mathbf{l}_{\cdot,j}$.

Consider one row or column at the time, and set the rest of the random field fixed. Figure 34 illustrates the notations for considering a row or a column in the field separately. The row $\mathbf{l}_{i,\cdot}$ is colored in Figure 34 (a), and the column $\mathbf{l}_{\cdot,j}$ is colored in Figure 34 (b). The node i,j is in both $\mathbf{l}_{i,\cdot}$ and $\mathbf{l}_{\cdot,j}$. For notational convenience, $\mathbf{l}_{-i,\cdot}$ consists of all the nodes not colored in Figure 34 (a), while $\mathbf{l}_{-\cdot,j}$ is all the nodes not colored in Figure 34 (b). Consider the probability of these profiles given the remaining grid nodes in the field, $p(\mathbf{l}_{i,\cdot}|\mathbf{l}_{-i,\cdot})$ and $p(\mathbf{l}_{\cdot,j}|\mathbf{l}_{-\cdot,j})$,

$$\begin{aligned} p(\mathbf{l}_{i,\cdot}|\mathbf{l}_{-i,\cdot}) &= p(l_{i,1}|\mathbf{l}_{-i,\cdot}) \times \prod_{j=2}^{n_2} p(l_{i,j}|\mathbf{l}_{i,1:(j-1)}, \mathbf{l}_{-i,\cdot}) \\ p(\mathbf{l}_{\cdot,j}|\mathbf{l}_{-\cdot,j}) &= p(l_{1,j}|\mathbf{l}_{-\cdot,j}) \times \prod_{i=2}^{n_1} p(l_{i,j}|\mathbf{l}_{1:(i-1),j}, \mathbf{l}_{-\cdot,j}). \end{aligned} \tag{36}$$

Focusing on the row $i \in \{2 \ldots, n_1 - 1\}$, the Markov chain form can be expressed,

$$\begin{aligned} p(l_{i,j}|\mathbf{l}_{i,1:(j-1)}, \mathbf{l}_{-i,\cdot}) &= \text{const} \times \exp\Big\{\nu(l_{i,j})\nu_h(l_{i,j-1}, l_{i,j})\Big\} \\ &\quad \times \exp\Big\{\nu_v(l_{i-1,j}, l_{i,j}) + \nu_v(l_{i,j}, l_{i+1,j})\Big\} \times h_v(l_{i,j}) \\ &= p(l_{i,j}|l_{i,j-1}, l_{i+1,j}, l_{i-1,j}), \quad j \in \{2, \ldots, n_2\} \\ p(l_{i,1}|\mathbf{l}_{-i,\cdot}) &= \text{const} \times \exp\Big\{\nu_v(l_{i-1,1}, l_{i,1}) + \nu_v(l_{i,1}, l_{i+1,1})\Big\} \times h_v(l_{i,1}) \\ &= p(l_{i,1}|l_{i-1,1}, l_{i+1,1}), \end{aligned} \tag{37}$$

where

$$
\begin{aligned}
h_v(l_{i,j},\mathbf{l}_{i-1,\cdot},\mathbf{l}_{i+1,\cdot}) = &\sum_{l'_{i,j+1}\in\mathbb{L}} \cdots \sum_{l'_{i,n_2}\in\mathbb{L}} \Bigg[\exp\Big\{\nu(l'_{i,j+1}) + \nu_h(l_{i,j}, l'_{i,j+1})\Big\} \\
&\times \exp\Big\{ \sum_{w=j+2}^{n_2} \big(\nu(l'_{i,w}) + \nu_h(l'_{i,w-1}, l'_{i,w})\big) + \sum_{w=j+1}^{n_2} \big(\nu_v(l_{i-1,w}, l'_{i,w}) + \nu_v(l'_{i,w}, l_{i+1,w})\big)\Big\}\Bigg] \\
&= \sum_{l'_{i,j+1}\in\mathbb{L}} \exp\Big\{\nu(l'_{i,j+1}) + \nu_h(l_{i,j}, l'_{i,j+1}) + \nu_v(l_{i-1,j+1}, l'_{i,j+1}) + \nu_v(l'_{i,j+1}, l_{i+1,j+1})\Big\} \\
&\times \sum_{l'_{i,j+2}\in\mathbb{L}} \exp\Big\{\nu(l'_{i,j+1}) + \nu_h(l'_{i,j+1}, l'_{i,j+2}) + \nu_v(l_{i-1,j+2}, l'_{i,j+2}) + \nu_v(l'_{i,j+2}, l_{i+1,j+2})\Big\} \\
&\times \cdots \sum_{l'_{i,n_2}\in\mathbb{L}} \exp\Big\{\nu(l'_{i,n_2}) + \nu_h(l'_{i,n_2-1}, l'_{i,n_2}) + \nu_v(l_{i-1,n_2}, l'_{i,n_2}) + \nu_v(l'_{i,n_2}, l_{i+1,n_2})\Big\} \\
&= \sum_{l'_{i,j+1}\in\mathbb{L}} \exp\Big\{\nu(l'_{i,j+1}) + \nu_h(l_{i,j}, l'_{i,j+1}) + \nu_v(l_{i-1,j+1}, l'_{i,j+1}) \\
&+ \nu_v(l'_{i,j+1}, l_{i+1,j+1})\Big\} \times h_v(l'_{i,j+1}, \mathbf{l}_{i-1,\cdot}, \mathbf{l}_{i+1,\cdot}).
\end{aligned}
\tag{38}
$$

For $i = 1$ or $i = n_1$ it is similar, only we do not include $\nu_v(l_{i-1,j}, l_{i,j})$ and $\nu_v(l_{i-1,w}, l'_{i,w})$ or $\nu_v(l_{i,j}, l_{i+1,j})$ and $\nu_v(l'_{i,w}, l_{i+1,w})$ respectively. In general, for $i \in \{1, \ldots, n_1\}$,

$$
p(l_{i,j}|\mathbf{l}_{i:1:(j-1)}, \mathbf{l}_{-i,\cdot}) = p(l_{i,j}|l_{\mathbf{x}}; \mathbf{x} \in \mathbf{n}_{i,j} \backslash l_{\mathbf{x}} \in \mathbf{l}_{i,j:n_2}).
$$

For the columns, the procedure is done correspondingly as for the rows.

4.3 Posterior model

The posterior model is expressed on Gibbs form as in Expression 7,

$$
\begin{aligned}
p(\mathbf{l}|\mathbf{d}) = \text{const}^* \times p(\mathbf{d}|\mathbf{l})p(\mathbf{l}) = \text{const} \times &\prod_{i=1}^{n_1}\prod_{j=1}^{n_2} p(d_{i,j}|l_{i,j}) \\
&\times \exp\Big\{ \sum_{i=1}^{n_1}\sum_{j=1}^{n_2} \nu(l_{i,j}) + \sum_{i=1}^{n_1}\sum_{j=2}^{n_2} \nu_h(l_{i,j-1}, l_{i,j}) + \sum_{i=2}^{n_1}\sum_{j=1}^{n_2} \nu_v(l_{i-1,j}, l_{i,j})\Big\}.
\end{aligned}
\tag{39}
$$

Using Expression 9 we consider the Markov form of the posterior model,

$$
p(l_{i,j}|\ \mathbf{l}_{-i,j}, \mathbf{d}) = p(l_{i,j}|l_{\mathbf{x}}, d_{i,j}; \mathbf{x} \in \mathbf{n}_{i,j}), \quad i \in \{1 \ldots, n_1\}, j \in \{1 \ldots, n_2\}. \tag{40}
$$

As we did for the prior model, focus in the rows and the columns separately and consider the Markov chain form for the profiles,

$$\begin{aligned} p(\mathbf{l}_{i,\cdot}|\mathbf{l}_{-i,\cdot},\mathbf{d}) &= p(l_{i,1}|\mathbf{l}_{-i,\cdot},\mathbf{d}) \times \prod_{j=2}^{n_2} p(l_{i,j}|\mathbf{l}_{i,1:(j-1)},\mathbf{l}_{-i,\cdot},\mathbf{d}), \\ p(\mathbf{l}_{\cdot,j}|\mathbf{l}_{-\cdot,j},\mathbf{d}) &= p(l_{1,j}|\mathbf{l}_{-\cdot,j},\mathbf{d}) \times \prod_{i=2}^{n_1} p(l_{i,j}|\mathbf{l}_{1:(i-1),j},\mathbf{l}_{-\cdot,j},\mathbf{d}). \end{aligned} \tag{41}$$

Again we focus on the rows as the procedure is done correspondingly for the columns. Recall Expression 19 and let $i \in \{2,\ldots,n_1-1\}$,

$$\begin{aligned} p(l_{i,j}|\mathbf{l}_{-i,\cdot},\mathbf{l}_{i,1:(j-1)},\mathbf{d}) &= \text{const} \times \exp\Big\{\nu(l_{i,j}) + \nu_h(l_{i,j-1},l_{i,j})\Big\} \\ &\quad \times \exp\Big\{\nu_v(l_{i-1,j},l_{i,j}) + \nu_v(l_{i,j},l_{i+1,j})\Big\} p(d_{i,j}|l_{i,j}) \times g_v(l_{i,j},\mathbf{d}_{i,(j+1):n_2}) \\ &= p(l_{i,j}|l_{i,j-1},l_{i+1,j},l_{i-1,j},\mathbf{d}_{i,j:n_2}), \quad j \in \{2,\ldots,n_2\}, \\ p(l_{i,1}|\mathbf{l}_{-i,\cdot},\mathbf{d}) &= \text{const} \times \exp\Big\{\nu(l_{i,1}) + \nu_v(l_{i-1,1},l_{i,1}) + \nu_v(l_{i,1},l_{i+1,1})\Big\} \\ &\quad \times p(d_{i,1}|l_{i,1}) \times g_h(l_{i,1},\mathbf{d}_{i,2:n_2}) \\ &= p(l_{i,1}|l_{i-1,1},l_{i+1,1},\mathbf{d}_{i,1:n_2}), \end{aligned} \tag{42}$$

where

$$
\begin{aligned}
g_h(l_{i,j}, \mathbf{d}_{i,(j+1):n_2}, \mathbf{l}_{i-1,\cdot}, \mathbf{l}_{i+1,\cdot}) = \sum_{l'_{i,j+1}\in\mathbb{L}} \cdots \sum_{l'_{i,n_2}\in\mathbb{L}} \Bigg[\exp\Big\{ \nu(l'_{i,j+1}) + \nu_h(l_{i,j}, l'_{i,j+1}) \\
+ \sum_{w=j+2}^{n_2} \nu(l'_{i,w}) + \nu_h(l'_{i,w-1}, l'_{i,w}) \Big\} \exp\Big\{ \sum_{w=j+1}^{n_2} \nu_v(l_{i-1,w}, l'_{i,w}) \\
+ \nu_v(l'_{i,w}, l_{i+1,w}) \Big\} \prod_{w=j+1}^{n_2} p(d_{i,w}|l'_{i,w}) \Bigg] = \sum_{l'_{i,j+1}\in\mathbb{L}} \exp\Big\{ \nu(l'_{i,j+1}) \\
+ \nu_h(l_{i,j}, l'_{i,j+1}) + \nu_v(l_{i-1,j+1}, l'_{i,j+1}) + \nu_v(l'_{i,j+1}, l_{i+1,j+1}) \Big\} p(d_{i,j+1}|l'_{i,j+1}) \\
\times \sum_{l'_{i,j+2}\in\mathbb{L}} \cdots \sum_{l'_{i,n_2-1}\in\mathbb{L}} \exp\Big\{ \nu(l'_{i,n_2-1}) + \nu_h(l'_{i,n_2-2}, l'_{i,n_2-1}) + \nu_v(l_{i-1,n_2-1}, l'_{i,n_2-1}) \\
+ \nu_v(l'_{i,n_2-1}, l_{i+1,n_2-1}) \Big\} p(d_{i,n_2-1}|l'_{i,n_2-1}) \sum_{l'_{i,n_2}\in\mathbb{L}} \exp\Big\{ \nu(l'_{i,n_2}) \\
+ \nu_h(l'_{i,n_2-1}, l'_{i,n_2}) + \nu_v(l_{i-1,n_2}, l'_{i,n_2}) + \nu_v(l'_{i,n_2}, l_{i+1,n_2}) \Big\} \times p(d_{i,n_2}|l'_{i,n_2}) \\
= \sum_{l'_{i,j+1}\in\mathbb{L}} \exp\Big\{ \nu(l'_{i,j+1}) + \nu_h(l_{i,j}, l'_{i,j+1}) + \nu_v(l_{i-1,j+1}, l'_{i,j+1}) \\
+ \nu_v(l'_{i,j+1}, l_{i+1,j+1}) \Big\} p(d_{i,j+1}|l'_{i,j+1}) \times g_h(l'_{i,j+1}, \mathbf{d}_{i,(j+2):n_2}, \mathbf{l}_{i-1,\cdot}, \mathbf{l}_{i+1,\cdot}).
\end{aligned} \tag{43}
$$

Similarly as for the prior model; for $i = 1$ or $i = n_1$ the probability function is similar, only we do not include $\nu_v(l_{i-1,j}, l_{i,j})$ and $\nu_v(l_{i-1,w}, l'_{i,w})$ or $\nu_v(l_{i,j}, l_{i+1,j})$ and $\nu_v(l'_{i,w}, l_{i+1,w})$ respectively. Having this established we start looking at simulating realizations and predictions.

4.4 Simulating realizations and predictions

The regular single-site Gibbs algorithm is provided in Algorithm 8 where,

$$p(l_{i,j}|\mathbf{l}^s_{-i,j}, \mathbf{d}) = p(l_{i,j}|l^s_{\mathbf{x}}, d_{i,j}; \mathbf{x} \in \mathbf{n}_{i,j}),$$

by Expression 40. The algorithm generates one realization of the posterior model.

Algorithm 8: Single-site Gibbs Algorithm

Initialise $\mathbf{l}^s$ such that $p(\mathbf{l}^s) > 0$
repeat
  $i \sim \text{unif}[1, \ldots, n_1]$
  $j \sim \text{unif}[1, \ldots, n_2]$
  $l^s_{i,j} \leftarrow p(l_{i,j}|\mathbf{l}^s_{-i,j}, \mathbf{d})$
until *convergence*;

We can not simply calculate the posterior model in the same way as we could for the MRPs. Computing the probabilities focusing on one profile, row or column, is feasible. By changing the $p(d_i|l_i)$ from Algorithm 1 to some $p^*(d_{i,j}|l_{i-1,j}, l_{i+1,j})$ that includes the influence from the grid nodes in the vertical cliques, we can use Algorithm 1 to calculate $p(\mathbf{l}_{i,\cdot}|\mathbf{l}_{-i,\cdot})$. Similarly, Algorithm 1 computes the probability of a column given the remaining field.

The profile block Gibbs algorithm (Algorithm 9) generates one realization, $\mathbf{l}^s$, of the posterior model, iteratively calculating the probability of a random profile given the remaining field. The probabilities in Algorithm 9 are calculated by combining Expression 42 and Algorithm 1.

Algorithm 9: Profile block Gibbs Algorithm

Initialise $\mathbf{l}^s$ such that $p(\mathbf{l}^s) > 0$
repeat
  $t \sim \text{unif}(0, 1)$
  if $t \leq \frac{n_1}{n_1+n_2}$ **then**
    $i \sim \text{unif}[1, \ldots, n_1]$
    $l^s_{i,1} \leftarrow p(l_{i,1}|\mathbf{l}^s_{-i,\cdot}, \mathbf{d})$
    for *all* $j = 2, \ldots, n_2$ **do**
      $l^s_{i,j} \leftarrow p(l_{i,j}|\mathbf{l}^s_{-i,\cdot}, \mathbf{l}^s_{i,1:(j-1)}, \mathbf{d})$
    end
  else
    $j \sim \text{unif}[1, \ldots, n_2]$
    $l^s_{1,j} \leftarrow p(l_{1,j}|\mathbf{l}^s_{-\cdot,j}, \mathbf{d})$
    for *all* $i = 2, \ldots, n_1$ **do**
      $l^s_{i,j} \leftarrow p(l_{i,j}|\mathbf{l}^s_{-\cdot,j}, \mathbf{l}^s_{1:(i-1),j}, \mathbf{d})$
    end
  end
until *convergence*;

The efficiencies of Algorithm 8 and 9 are compared in Section 4.6 and 5. Further, we consider the MMAP, and since we do not have the probabilities for the entire field, the MMAP is approximated by creating many realizations $\mathbf{l}^s$ and computing the average. Probability maps are also approximated by calculating proportion of the states in each individual grid node using many realizations.

4.5 Estimating the parameters

Assume the clique functions $\nu(\cdot)$, $\nu_h(\cdot,\cdot)$ and $\nu_v(\cdot,\cdot)$ consists of one or several parameters $\boldsymbol{\theta}$. A training image, $\mathbf{l}^0$ is used in estimating the parameters by maximizing,

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} p(\mathbf{l}^0; \boldsymbol{\theta}).$$

The prior model from Expression 33 is computationally demanding to calculate, so the probability $p(\mathbf{l}^0; \boldsymbol{\theta})$ is assessed approximately using the Markov formulation,

$$\hat{p}(\mathbf{l}^0; \boldsymbol{\theta}) = \prod_{\mathbf{x} \in \mathcal{L}_{\mathcal{D}}} p(l^0_{\mathbf{x}} | \mathbf{l}^0_{\mathbf{y}}; \mathbf{y} \in \mathbf{n}_{\mathbf{x}}; \boldsymbol{\theta}),$$

and we can assess a pseudo-estimate of the parameters. The probability $p(\mathbf{l}^0; \boldsymbol{\theta})$ is calculated directly for the MRP, but an approximation is performed for the MRF.

4.6 Examples of Markov random field

The examples and their corresponding training images are drawn to have specific geometries. We present two examples where the HMMs have different properties. We define the prior model of the MRF on Gibbs form as in Expression 33,

$$\begin{aligned} p(\mathbf{l}) = \text{const} \times \exp\Big\{ & \sum_{i=1}^{n_1}\sum_{j=1}^{n_2} \beta_1 \mathrm{I}(l_{i,j} \in \{B, W\}) \\ & + \sum_{i=1}^{n_1}\sum_{j=2}^{n_2} \beta_{h1} \mathrm{I}(l_{i,j-1} = l_{i,j}) + \beta_{h2}\big(\mathrm{I}(l_{i,j-1} = B, l_{i,j} = W) + \mathrm{I}(l_{i,j-1} = W, l_{i,j} = B)\big) \\ & + \sum_{i=2}^{n_1}\sum_{j=1}^{n_2} \beta_{v1} \mathrm{I}(l_{i-1,j} = l_{i,j}) + \beta_{v2}\big(\mathrm{I}(l_{i-1,j} = B, l_{i,j} = W) + \mathrm{I}(l_{i-1,j} = W, l_{i,j} = B)\big)\Big\}. \end{aligned} \tag{44}$$

The prior model is general, and contains several parameters. Some parameters can be fixed due to the nature of the training image, the remaining parameters are estimated. The parameters are estimated using the training image, which should be similar to the field of interest. In the examples, the truth and the training image are constructed to look similar.

The parameter β_1 says something about the proportions of {black, white} compared to {grey}, so we expect that the field contains more black and white nodes than grey if $\beta_1 > 0$. The parameters β_{h1} and β_{h2} contain information regarding the horizontal cliques. If $\beta_{h1} > 0$, horizontal neighbors tend to be the same color. If $\beta_{h_2} > 0$, a black node is likely to occur after a white node. Similarly, β_{v1} and β_{v2} contain the same information concerning the vertical cliques. This general prior is used both in the examples of MRF, and later in the applications to brain MRI image in Section 5.

4.6.1 Example 1

The MRF consists of 20×13 grid nodes, so $n_1 = 20$, $n_2 = 13$ and $n = 20 \times 13 = 260$. The truth $\mathbf{l}^T$ is constructed and displayed in Figure 35. The training image $\mathbf{l}^0$ is constructed to be similar to the truth and is displayed in Figure 36. There are no black node next to a white node in horizontal direction in any location in the training image, so we assume that it is "impossible" and set the corresponding parameter fixed, $\beta_{h2} = -100$. In the vertical direction, this is not the case, so the remaining parameters need to be estimated.

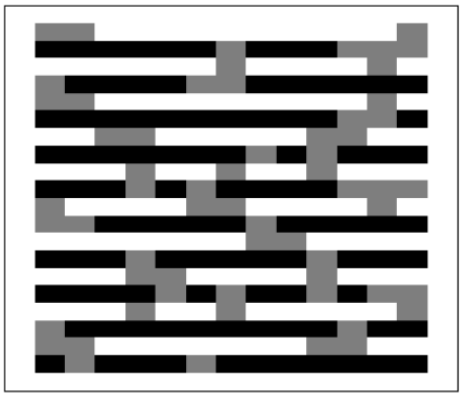


Figure 35: The true image $\mathbf{l}^T$.

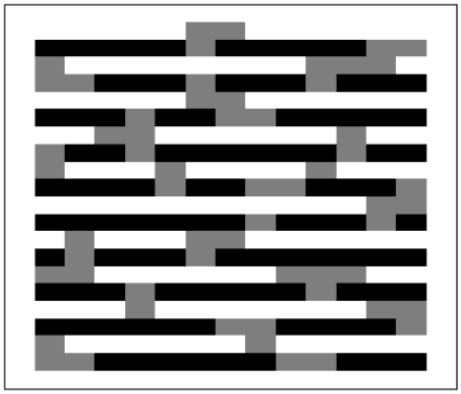


Figure 36: The training image $\mathbf{l}^0$.

The remaining parameters in the prior in Expression 44 are estimated,

$$\begin{aligned}\hat{\beta}_1 &= -0.6\\ \hat{\beta}_{h1} &= 0.6\\ \hat{\beta}_{v1} &= 0.2\\ \hat{\beta}_{v2} &= 1.\end{aligned}$$

The estimates are within reason considering the training image, as they indicate that it is likely that a black node occurs next to a white node in the vertical direction. The estimate $\hat{\beta}_1$ is not immediately expected to be negative by the look of the training image, but this is balanced by the estimate $\hat{\beta}_{v2}$, which is relatively high.

Consider a Gaussian likelihood model. The likelihood model is defined for any observation $[d_{i,j}|l_{i,j}]$, $i = 1, \ldots, n_1$, $j = 1, \ldots, n_2$,

$$d_{i,j}|l_{i,j} \sim \text{Gauss}(\mu_{l_{i,j}}, 0.5^2)$$

$$\mu_{l_{i,j}} = \begin{cases} -1 & \text{if } l_{i,j} = \text{black} \\ 0 & \text{if } l_{i,j} = \text{grey} \\ 1 & \text{if } l_{i,j} = \text{white.} \end{cases}$$

The observations $\mathbf{d}$ are displayed in Figure 37, and it seems to be a medium level of noise as the original image is hard to recognize.

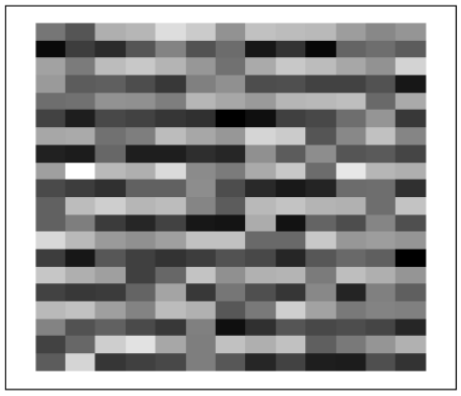
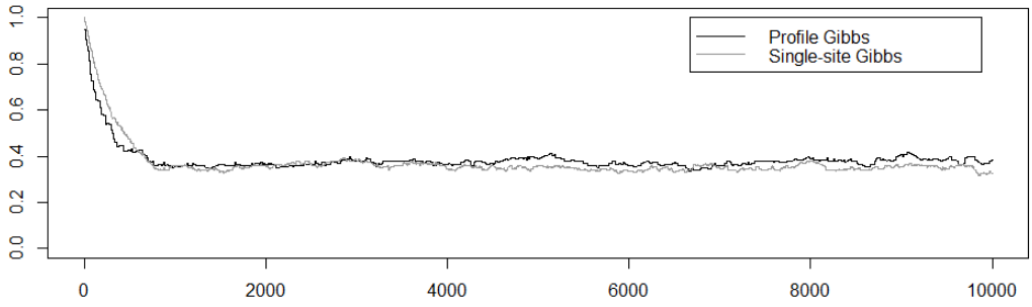


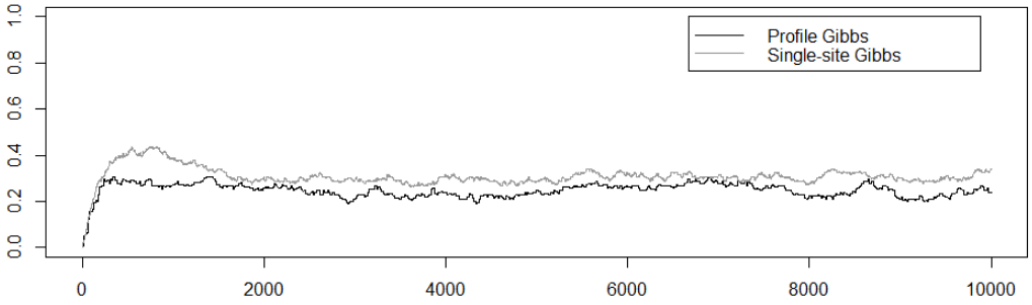
Figure 37: Observations from a Gaussian likelihood model with $\sigma^2 = 0.5^2$.

Using the estimated parameters, convergence plots achieved by the profile block Gibbs algorithm (Algorithm 9) are compared to convergence plots achieved by the single-site Gibbs algorithm (Algorithm 8). The convergence plots illustrate the proportions of the different colors in relation to the number of iterations. In the single-site Gibbs algorithm, one random node is chosen. Each time a node is picked, the number of iterations increases with one. For the profile block Gibbs algorithm, a random row or column is picked, and the number of iterations increases with the number of grid nodes in the picked row or column. The convergence plots are displayed in Figures 38 - 40, for different initial images.

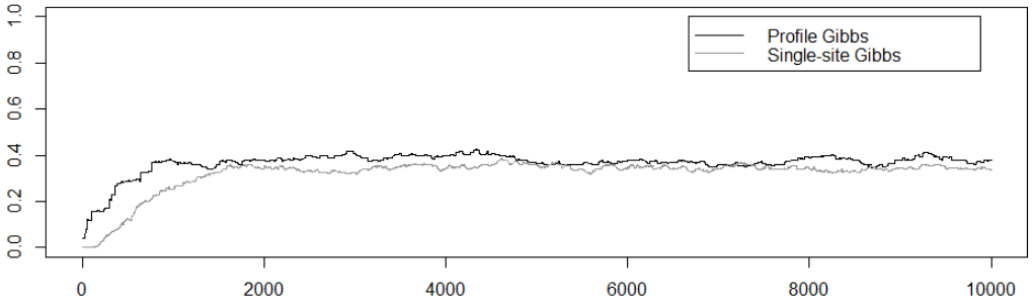
In Figure 38, the initial image is all black, which is clear as the proportions of black starts at one in Figure 38 (a). Although the profile block Gibbs algorithm seems to converge slightly before the single-site Gibbs algorithm, there is not a major difference. Similarly, in Figure 39 and 40, the initial images are all grey and all white respectively. The single-site Gibbs algorithm does not in any case converge visibly faster than the profile block Gibbs algorithm, but in these cases, neither one of the algorithms can be assumed to be more efficient than the other. After 2000 iterations, the algorithm seems to have converged in both algorithms, so the burn-in is set to be 2000. We can not conclude that the profile block Gibbs algorithm is more efficient than the single-site Gibbs algorithm.



(a) Proportion of image being black.

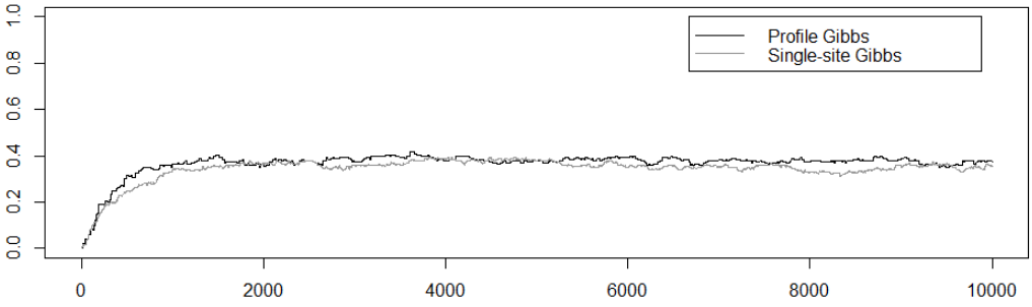


(b) Proportion of image being grey.

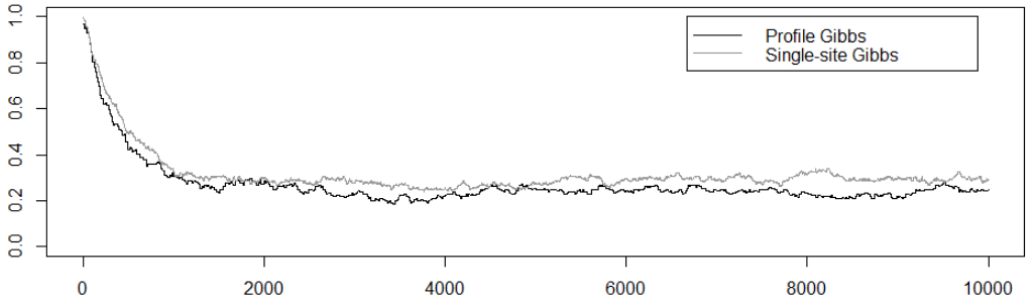


(c) Proportion of image being white.

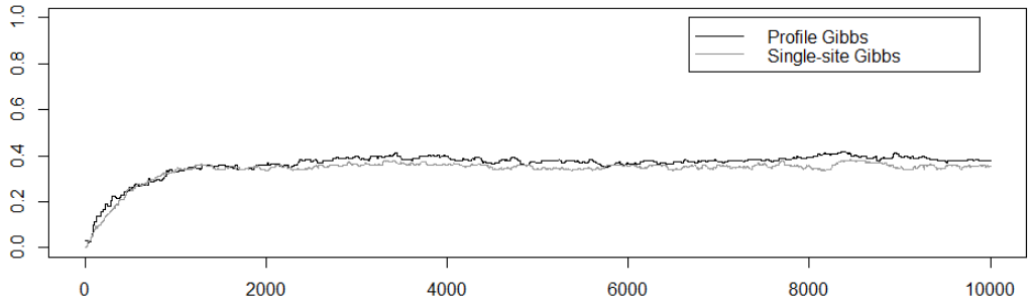
Figure 38: Initial image $\mathbf{l}^{s_0}$ is all black.



(a) Proportion of image being black.

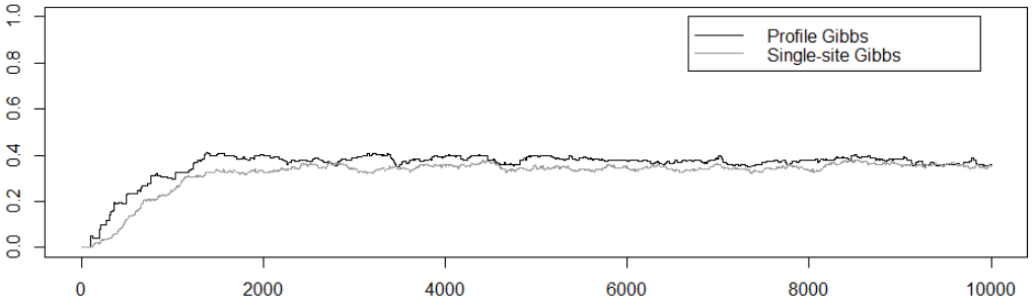


(b) Proportion of image being grey.

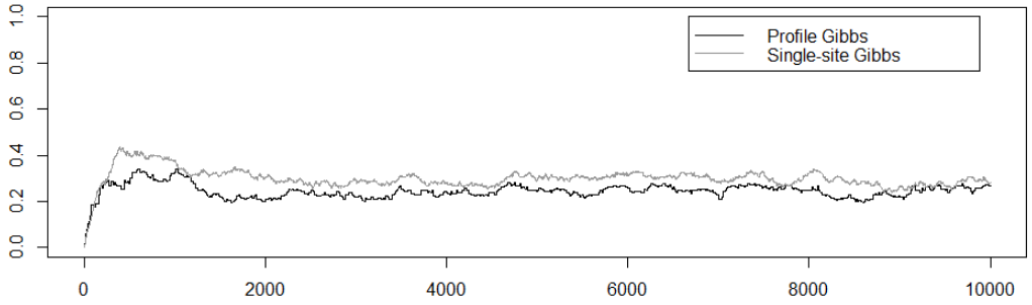


(c) Proportion of image being white.

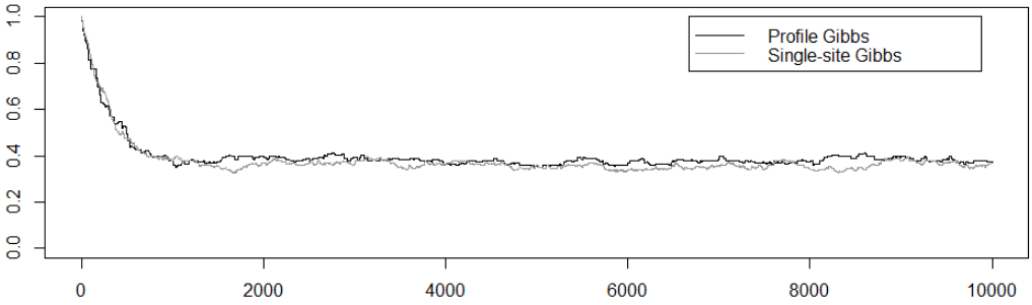
Figure 39: Initial image $\mathbf{I}^{s_0}$ is all grey.



(a) Proportion of image being black.



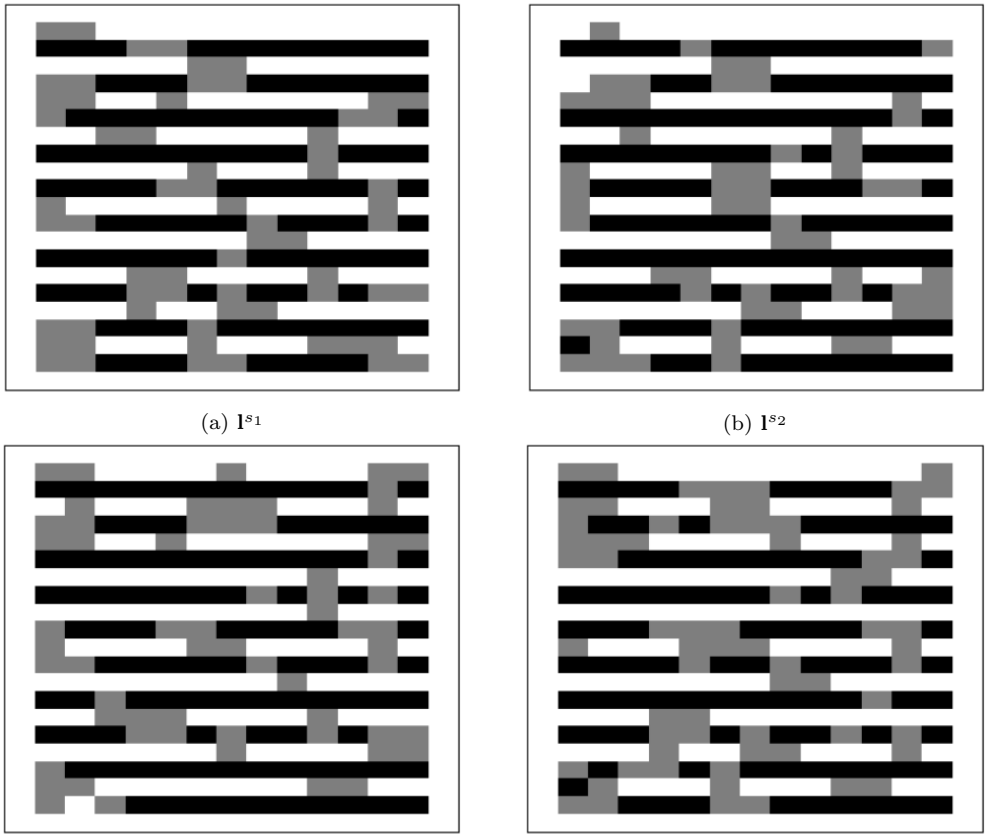
(b) Proportion of image being grey.



(c) Proportion of image being white.

Figure 40: Initial image $\mathbf{l}^{s_0}$ is all white.

Four realizations are simulated from the posterior model and displayed in Figure 41 using Algorithm 9. As the observations contain a medium level of noise, it is not surprising that the realizations contain several differences.



(a) $\mathbf{l}^{s_1}$ (b) $\mathbf{l}^{s_2}$

(c) $\mathbf{l}^{s_3}$ (d) $\mathbf{l}^{s_4}$

Figure 41: Four realizations generated from the posterior model.

Predictions for two-dimensional images are computationally demanding, so the MMAP is approximated by looking at the average of 1000 realizations. After the burn-in time on 2000 iterations, each realization is picked after 500 new iterations to enable a full sweep and collected until there are 1000 realizations. A full sweep would mean to change/attempt to change in expectation every grid node in the field. The MMAP is displayed in Figure 42. Although the prediction is not identical to the true image, they are very similar.

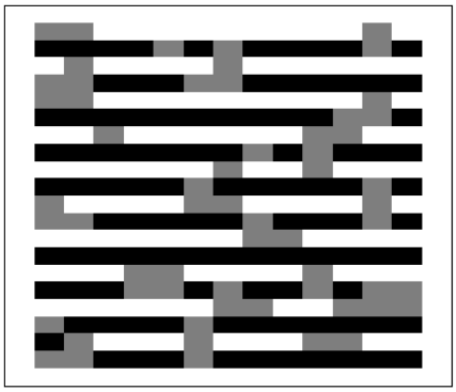
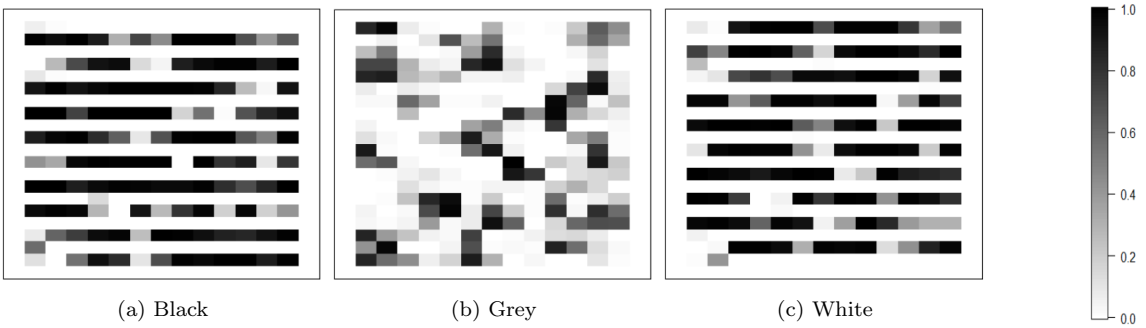


Figure 42: Approximated MMAP by taking the average of 1000 simulations.

Figure 43 displays the probability maps for the different possible states. The probability maps are not calculated but approximated looking at the state proportion in each grid node after 1000 realizations. It is clear that there are many uncertainties in several grid nodes, but in many grid nodes, we have a rather high probability of classifying the state of the grid node correctly. The areas with high probability for a specific state seems to be mostly correct in comparison to the truth $\mathbf{l}^T$ in Figure 35.



(a) Black (b) Grey (c) White

Figure 43: Approximated probability map considering 1000 simulations.

4.6.2 Example 2

The MRF consists of 20×13 grid nodes, so $n_1 = 20$, $n_2 = 13$ and $n = 20 \times 13$. The truth $\mathbf{l}^T$ is generated and displayed in Figure 44. The truth does not indicate that there are obvious differences in the horizontal clique functions compared to the vertical clique functions. It seems that the probability for {black,white} nodes is higher than the probability for {grey} nodes. The training image $\mathbf{l}^0$ is constructed to be similar and is displayed in Figure 45. There are no black node

next to a white node in any direction, so we assume that it is "impossible" and the parameters β_{h2}, β_{v2} are set fixed to -100.

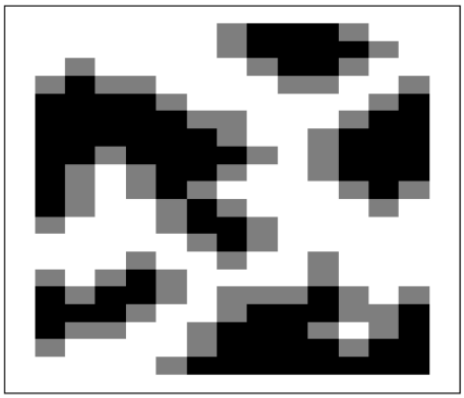


Figure 44: The true image.

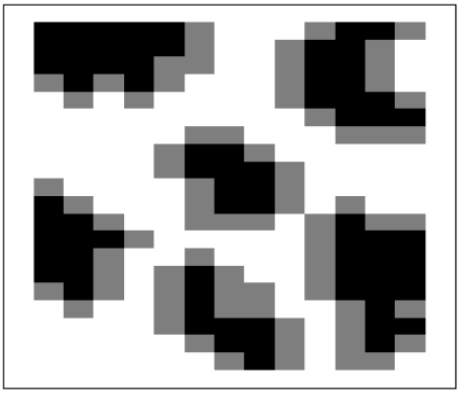


Figure 45: The training image.

The remaining parameters in the prior in Expression 44 are estimated,

$$\hat{\beta}_1 = 2.7$$
$$\hat{\beta}_{h1} = 0.2$$
$$\hat{\beta}_{v1} = 0.9,$$

where the estimates are within reason as they agree with what we expected looking at the training image. A high valued $\hat{\beta}_1$ indicates that there are more {black, white} nodes than {grey} nodes in the field. The estimates for β_{h1} and β_{v1} differ, but both indicate that neighbors tend to be of the same state in any direction.

Consider the same Gaussian likelihood model as in the previous example. The likelihood model is defined for any observation $[d_{i,j}|l_{i,j}]$, $i = 1, \ldots, n_1$, $j = 1, \ldots, n_2$,

$$d_{i,j}|l_{i,j} \sim \text{Gauss}(\mu_{l_{i,j}}, 0.5^2)$$

$$\mu_{l_{i,j}} = \begin{cases} -1 & \text{if } l_{i,j} = \text{black} \\ 0 & \text{if } l_{i,j} = \text{grey} \\ 1 & \text{if } l_{i,j} = \text{white.} \end{cases}$$

The observations $\mathbf{d}$ displayed in Figure 46 seems to contain a medium level of noise as expected from the likelihood model. The truth $\mathbf{l}^T$ in Figure 44 can be recognized in the observations.

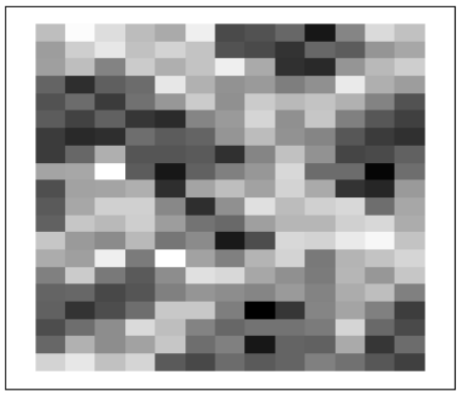
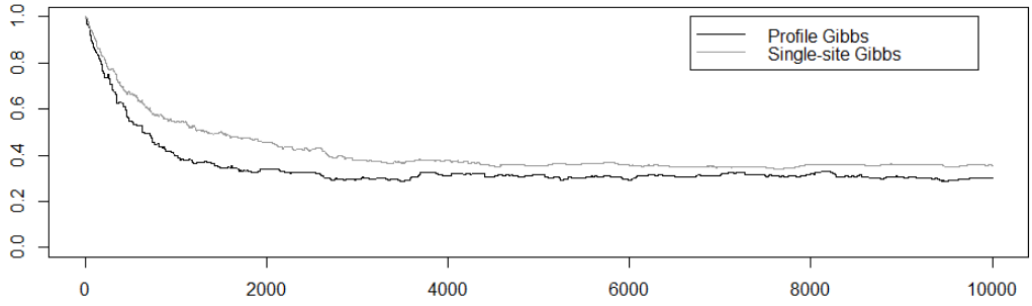


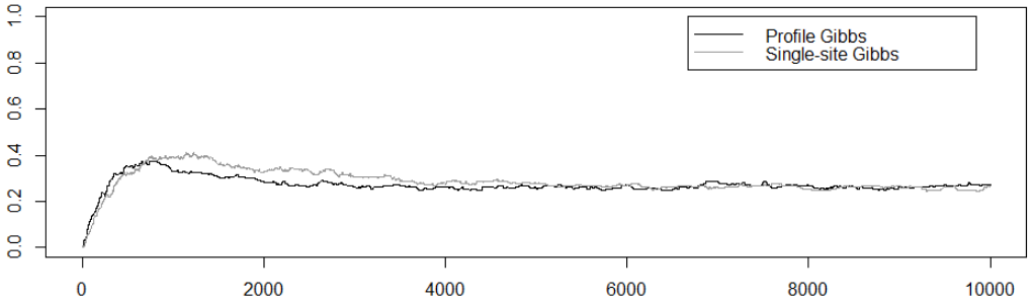
Figure 46: Observations from a Gaussian likelihood model with $\sigma^2 = 0.5^2$.

Using the estimated parameters, convergence plots achieved using Algorithm 9 are compared to convergence plots achieved using Algorithm 8. The convergence plots are displayed in Figure 47 - 49, for different initial images. The convergence plots are constructed in the same way as in the previous example, plotting the state proportions with the number of iterations.

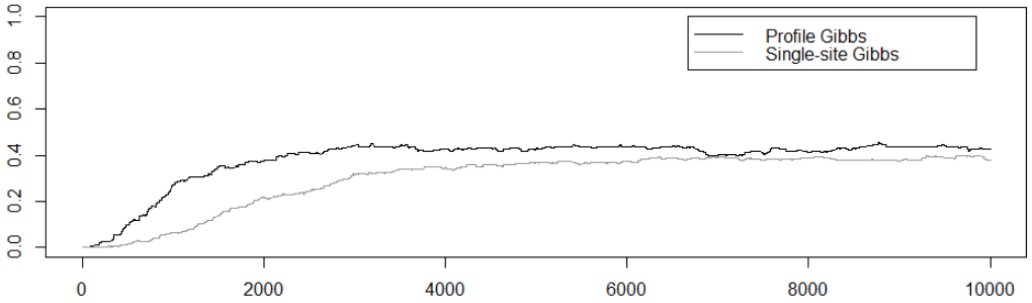
In Figure 47, the initial image is all black, and the profile block Gibbs algorithm converges faster than the single-site Gibbs algorithm. The difference is very visible in Figure 47 (a) and (c). When the initial image is all grey in Figure 48, the efficiency in the different algorithms seems to be the same or at least similar. Figure 49 displays the convergence plots when the initial image is all white, and the profile block Gibbs converges faster than the single-site Gibbs algorithm. In comparison with the previous example, there are more differences considering the algorithm efficiency in this example. For the profile block Gibbs algorithm we set the burn-in time at 3000 iterations.



(a) Proportions of image being black.

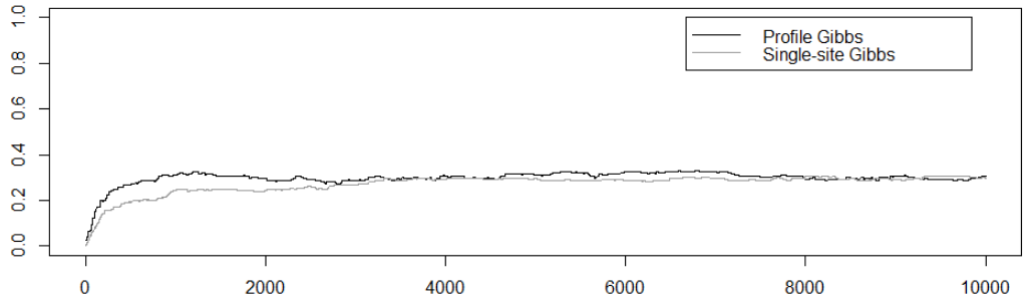


(b) Proportions of image being grey.

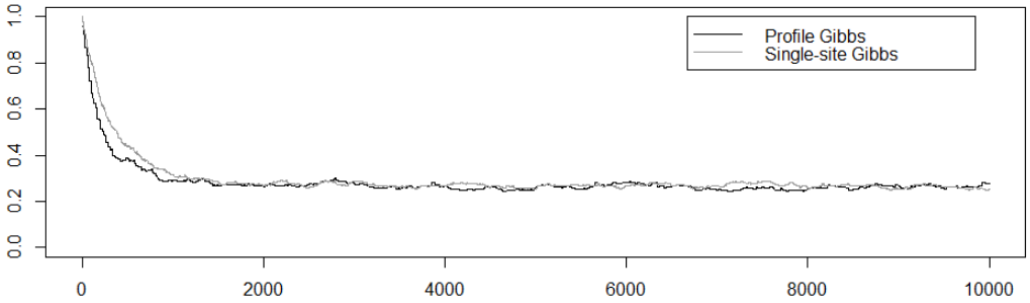


(c) Proportions of image being white.

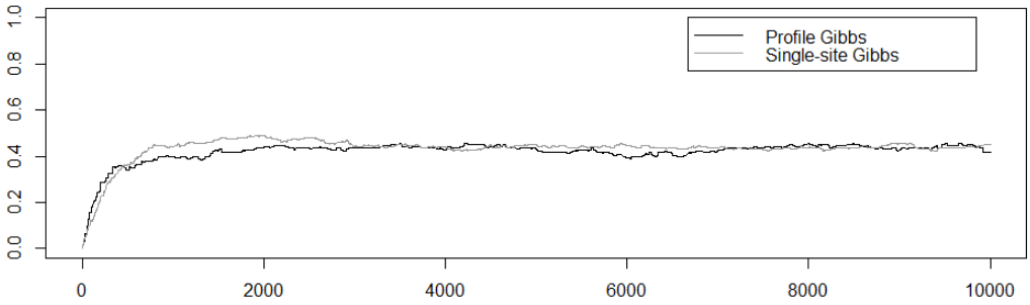
Figure 47: Initial image $\mathbf{l}^{s_0}$ is all black.



(a) Proportions of image being black.

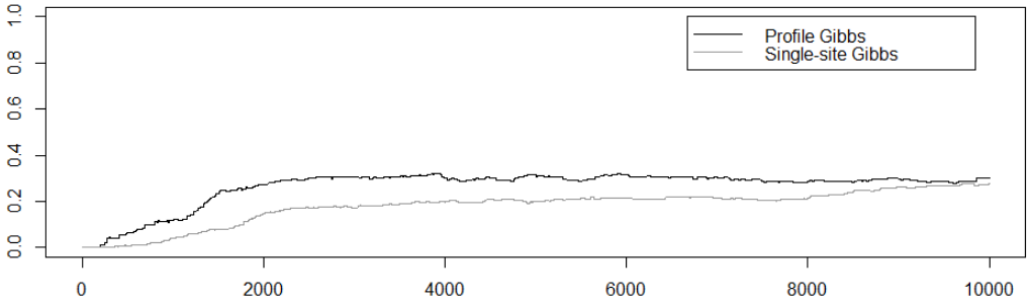


(b) Proportions of image being grey.

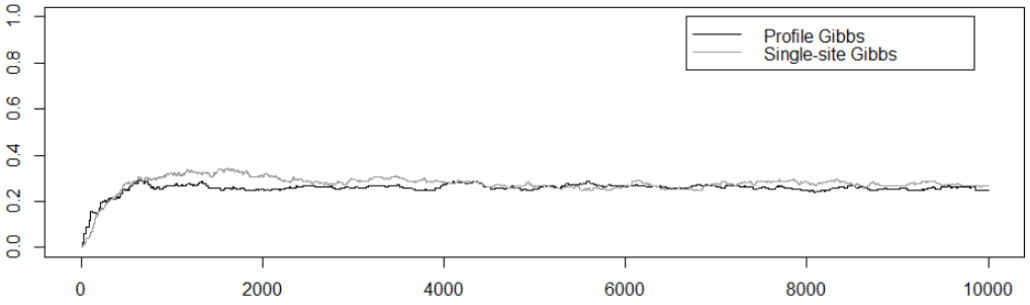


(c) Proportions of image being white.

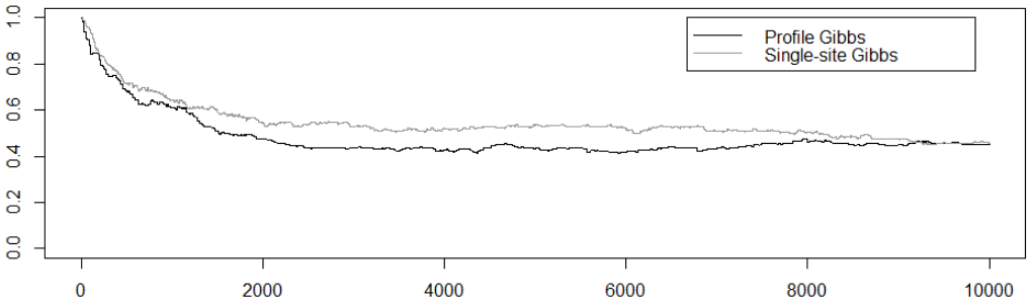
Figure 48: Initial image $\mathbf{I}^{s_0}$ is all grey.



(a) Proportions of image being black.



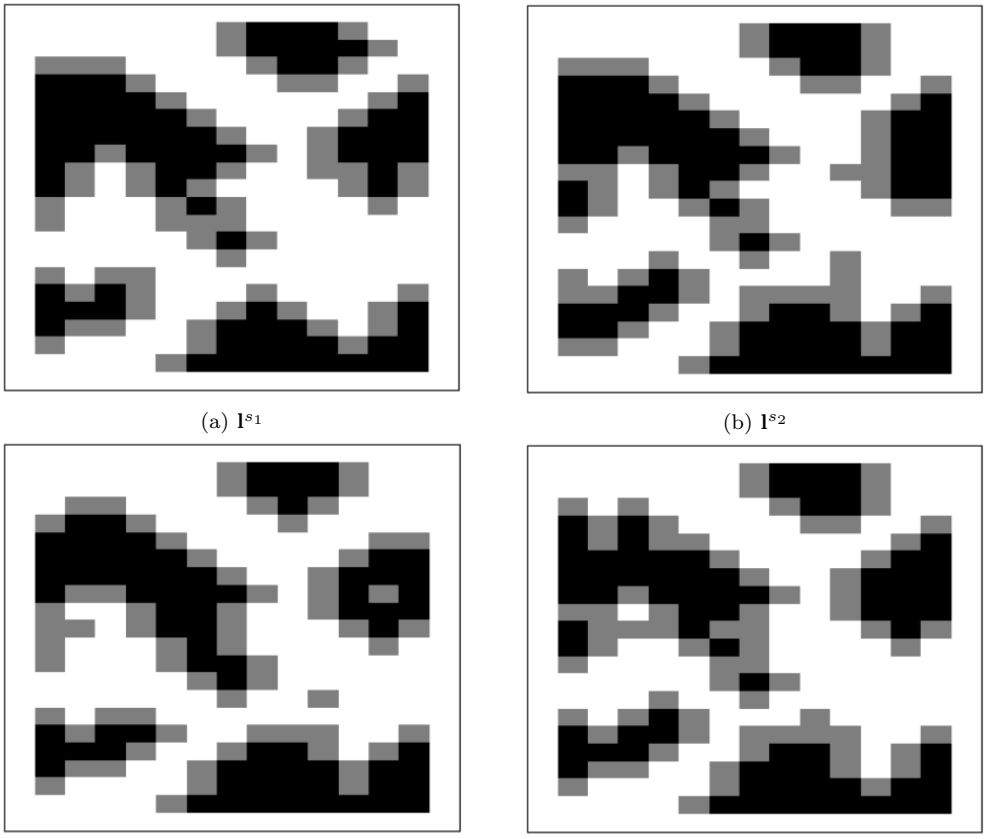
(b) Proportions of image being grey.



(c) Proportions of image being white.

Figure 49: Initial image $\mathbf{l}^{s_0}$ is all white.

Figure 50 displays four realizations of the posterior model. Although they are similar, they contain several differences as expected from the level of noise in the likelihood model.



(a) $\mathbf{l}^{s_1}$ (b) $\mathbf{l}^{s_2}$

(c) $\mathbf{l}^{s_3}$ (d) $\mathbf{l}^{s_4}$

Figure 50: Four realizations generated from the posterior model.

The MMAP is approximated by looking at the average of 1000 realizations generated by Algorithm 9, collected after the burn-in time at 3000 iterations, and then after every 500 iteration to enable a full sweep of the image. The MMAP $\tilde{\mathbf{l}}$ is generated and displayed in Figure 51. Again the prediction is not identical to the truth, but it is similar.

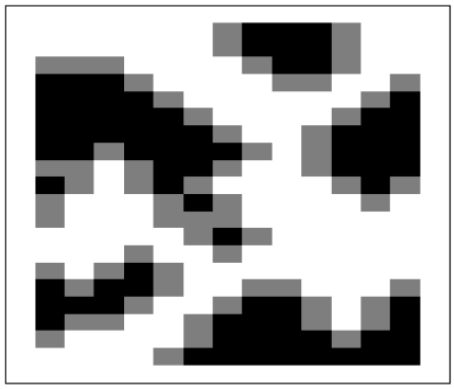
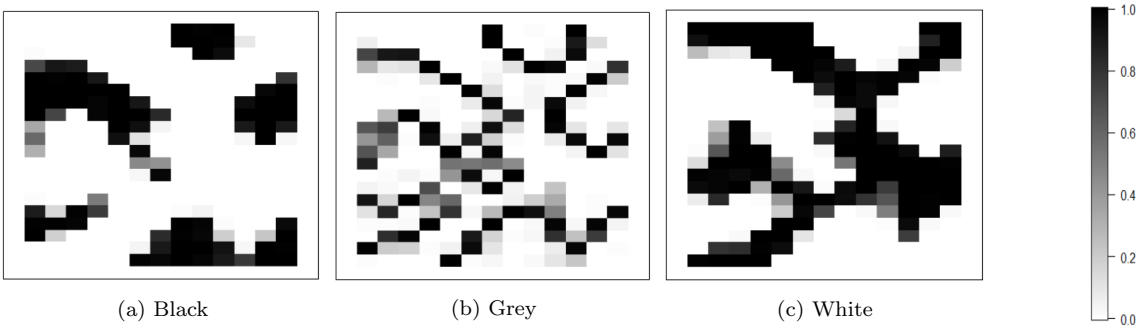


Figure 51: Approximated MMAP $\tilde{\mathbf{l}}$ by taking the average of 1000 simulations.

The probability maps for the different possible states are generated in the same way as in the previous example and are displayed in Figure 52. The probability maps indicate that the probability of identifying the grid node is rather high in most grid nodes, but with uncertainties especially in areas where the states from one grid node to the next are not the same. The grid nodes that contain a high probability of one specific state are mostly correct according to the truth $\mathbf{l}^T$ in Figure 44.



(a) Black (b) Grey (c) White

Figure 52: Approximated probability map considering 1000 simulations.

Based on the examples of MRFs, we can not conclude that the profile block Gibbs algorithm (Algorithm 9) is more efficient than the single-site Gibbs algorithm (Algorithm 8). The single-site Gibbs algorithm does never converge faster than the profile block Gibbs algorithm, which provides motivation to explore the algorithms further, as there are some visible differences in the convergence plots. As the images in the examples of MRFs are relatively small, a suggestion is to compare the algorithm efficiency using larger images, as they need more iterations to converge for obvious reasons.

5 Applications to brain MRI

As the motivation behind this study is to improve the procedure to denoise medical images, we consider a brain MRI image with noise. The brain MRI image $\mathbf{l}^B$ displayed in Figure 53 is collected from [BrainWeb, 2020], consisting of grid nodes having values between zero and 151. This brain MRI image $\mathbf{l}^B$ is used in generating observations and a training image.

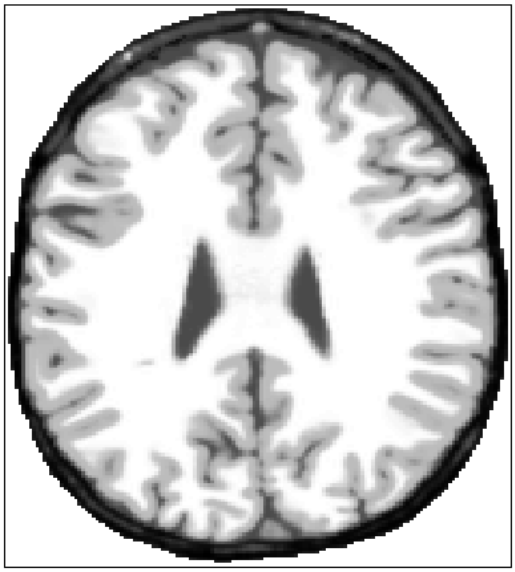


Figure 53: Simulated brain MRI image, $\mathbf{l}^B$, obtained from [BrainWeb, 2020] with 0% noise.

In this study the likelihood model is assumed to be known, so the observations are collected by generating an image of three states using the brain MRI image $\mathbf{l}^B$ from Figure 53, and then add noise. First a "truth" $\mathbf{l}^T$ is generated using the matrix P,

$$P = \begin{array}{r} \\ l^B_{i,j} < 50 \\ l^B_{i,j} \in [50, 80] \\ l^B_{i,j} \in [81, 90] \\ l^B_{i,j} \in [91, 110] \\ l^B_{i,j} \in [111, 120] \\ l^B_{i,j} \in [121, 130] \\ l^B_{i,j} > 130 \end{array} \begin{array}{c} \begin{array}{ccc} B & G & W \end{array} \\ \begin{bmatrix} 1.00 & 0.00 & 0.00 \\ 0.90 & 0.10 & 0.00 \\ 0.10 & 0.90 & 0.00 \\ 0.00 & 1.00 & 0.00 \\ 0.00 & 0.90 & 0.10 \\ 0.00 & 0.10 & 0.90 \\ 0.00 & 0.00 & 1.00 \end{bmatrix} \end{array}.$$

The values in the matrix P are probabilities of generating the different states {black, grey, white} depending on the values in the brain MRI image $\mathbf{l}^B$ in Figure 53.

Some noise is added to $\mathbf{l}^T$ by a Gaussian likelihood model to generate the observations $\mathbf{d}$,

$$d_{i,j}|l_{i,j} \sim \text{Gauss}(\mu_{l^T_{i,j}}, 0.5^2)$$

$$\mu_{l_{i,j}} = \begin{cases} -1 & \text{if } l^T_{i,j} = \text{black} \\ 0 & \text{if } l^T_{i,j} = \text{grey} \\ 1 & \text{if } l^T_{i,j} = \text{white.} \end{cases}$$

The aim is to denoise the observations $\mathbf{d}$ displayed Figure 54 into three states {black, grey, white}, where the black state is cerebrospinal fluid (CSF), the grey state is grey matter and the white state is white matter.

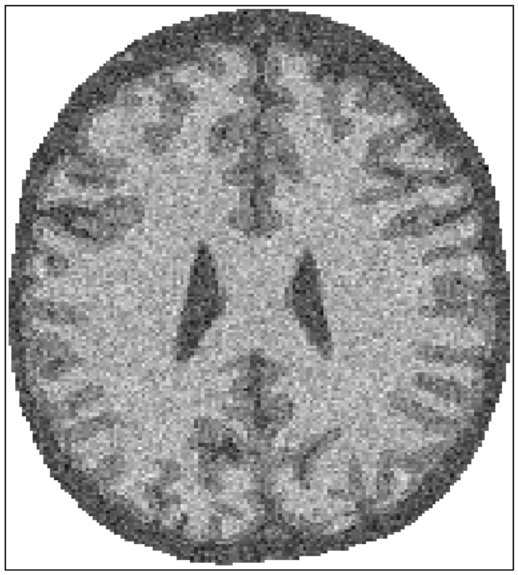


Figure 54: Simulated brain MRI image obtained from [BrainWeb, 2020] with added noise.

The rectangular grid size is $n_1 = 184$ and $n_2 = 145$, but this is a circular object with $n = 21085 < n_1 \times n_2$. Hence the grid $\mathcal{L}_\mathcal{D}$ is circular consisting of 21085 grid nodes. The prior is assumed to be similar to the examples from MRFs as in Expression 44. To estimate the parameters we use the training image $\mathbf{l}^0$ provided in Figure 55. The training image is generated from the brain MRI image $\mathbf{l}^B$ in Figure 53 by simple thresholding,

$$l^0_{i,j} = \begin{cases} -1 & \text{if } l^B_{i,j} < 80 \\ 0 & \text{if } l^B_{i,j} \in [80, 120] \\ 1 & \text{if } l^B_{i,j} > 120. \end{cases}$$

Since the training image $\mathbf{l}^0$ is generated by a slightly different procedure than the "truth" $\mathbf{l}^T$, they are not necessarily identical.

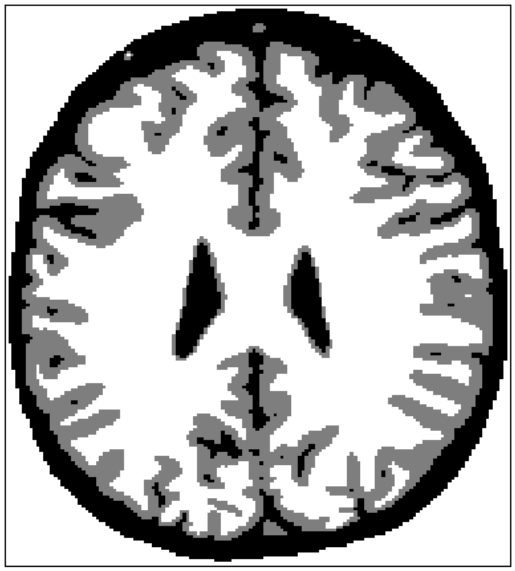


Figure 55: The training image.

Studying the training image, there is in no case a black node next to a white node in the vertical direction. The parameter β_{v2} is set fixed to -100 as we consider that "impossible". In the horizontal direction, a black node occurs next to a white node in 15 locations, so we can not make the same assumption regarding β_{h2}.

With one parameter from the prior in Expression 44 fixed, the four remaining parameters are estimated using the training image $\mathbf{l}^0$ in Figure 55,

$$\begin{aligned}\hat{\beta}_1 &= 0.1\\ \hat{\beta}_{h1} &= 1.8\\ \hat{\beta}_{h2} &= -2.8\\ \hat{\beta}_{v1} &= 1.7.\end{aligned}$$

The estimates are expected considering the training image. Neighbors tend to be of the same state in any direction, so the estimates for β_{h1} and β_{v2} are relatively high. Analyzing the training image revealed that there is a black node next to a white node in the horizontal direction in 15 locations, so even though it may occur, it is not the tendency. The low valued estimate of β_{h2} is therefore expected. The estimate regarding β_1 indicates that the overall proportion of {black, white} nodes compared to {grey} is not crucial in the prior, it is the neighbors that have the biggest impact on an individual grid node.

The observations $\mathbf{d}$ in Figure 54, based on the brain MRI image $\mathbf{l}^B$ in Figure 53, have a Gaussian

likelihood model defined for each observation $d_{i,j}$, $i, j \in \mathcal{L}_{\mathcal{D}}$,

$$d_{i,j}|l_{i,j} \sim \text{Gauss}(\mu_{l_{i,j}}, 0.5^2)$$

$$\mu_{l_{i,j}} = \begin{cases} -1 & \text{if } l_{i,j} = \text{black} \\ 0 & \text{if } l_{i,j} = \text{grey} \\ 1 & \text{if } l_{i,j} = \text{white.} \end{cases}$$

The posterior model is assessed iteratively by the two algorithms single-site Gibbs algorithm (Algorithm 8) and profile block Gibbs algorithm (Algorithm 9), using the estimated parameters. The convergence plots of the algorithms are displayed in Figure 56 - 58. In comparison to the examples from MRF, there are obvious differences. The convergence happens at a slower rate than any of the examples, so Figure 56 - 58 are displayed over 10^6 iterations in comparison to 10^4 in the examples. This is not surprising as there are more grid nodes in the brain MRI image, also the parameter estimates are more "extreme" for the brain MRI. After convergence in the simulations, the images appear as stationary.

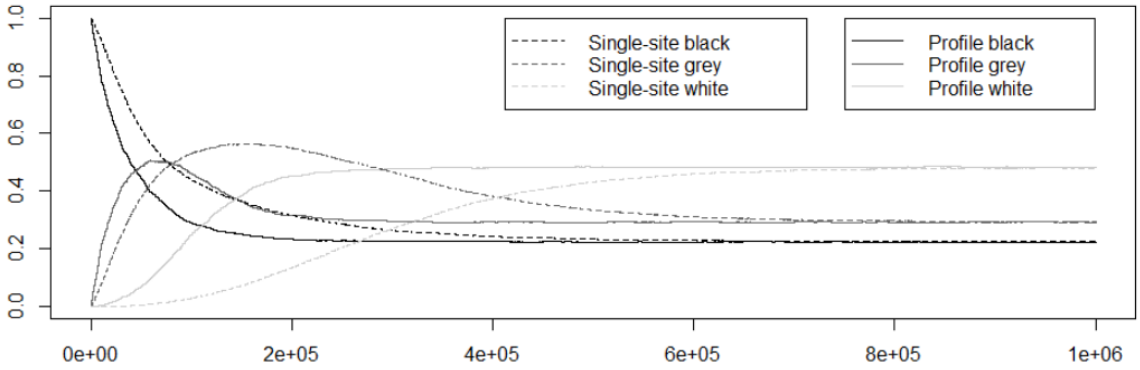


Figure 56: $\mathbf{l}^{s_0}$ is all black.

In Figure 56 the initial image is all black. The proportion of grey nodes increases rapidly before it stabilize due to the fact that it is unlikely for a black node to occur next to a white node.

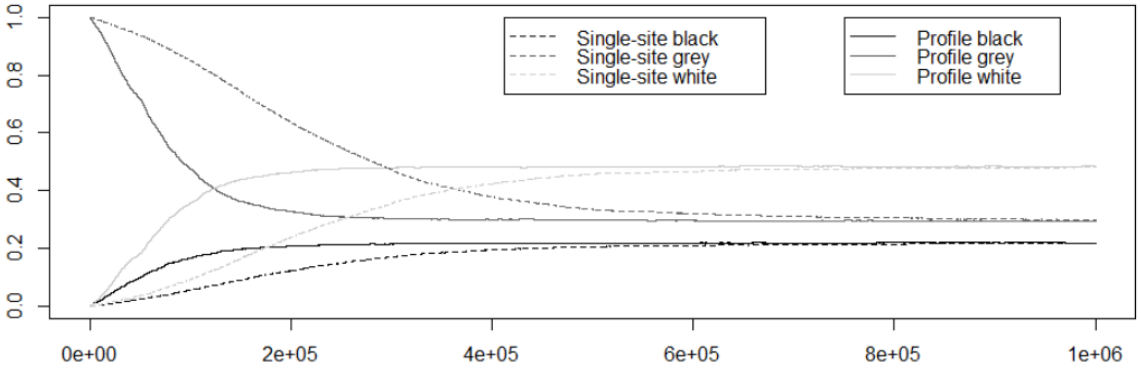


Figure 57: $\mathbf{l}^{s_0}$ is all grey.

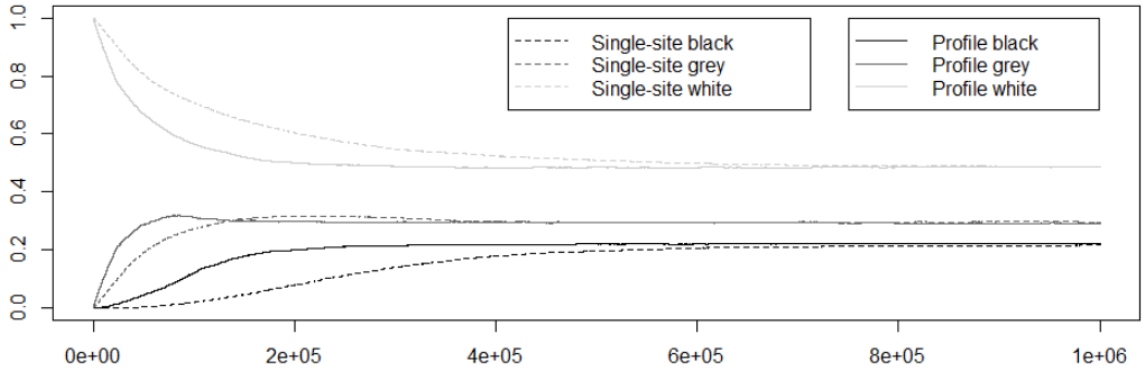


Figure 58: $\mathbf{l}^{s_0}$ is all white.

Figure 57 and 58 display the convergence plots when the initial image is respectively all grey and all white. The profile block Gibbs algorithm is visibly more efficient than the single-site Gibbs algorithm for all tested initial images.

In all convergence plots, a burn-in at 3×10^5 iterations seems reasonable for the profile block Gibbs algorithm. For the single-site Gibbs algorithm, the burn-in is set at 7×10^5. The algorithms are in this study not implemented by a trained programmer, but as they both are implemented by the same person it is reasonable to assume that they are similarly optimized. Using an all grey initial image, the times of the burn-in period are compared for 10 simulations of both profile block Gibbs algorithm and single-site Gibbs algorithm. While benchmarking the algorithms no background programs were running on the computer, so any interruptions were reduced to a minimum. Figure 59 displays the results from the benchmarking where the numbers are seconds the algorithm used to simulate.

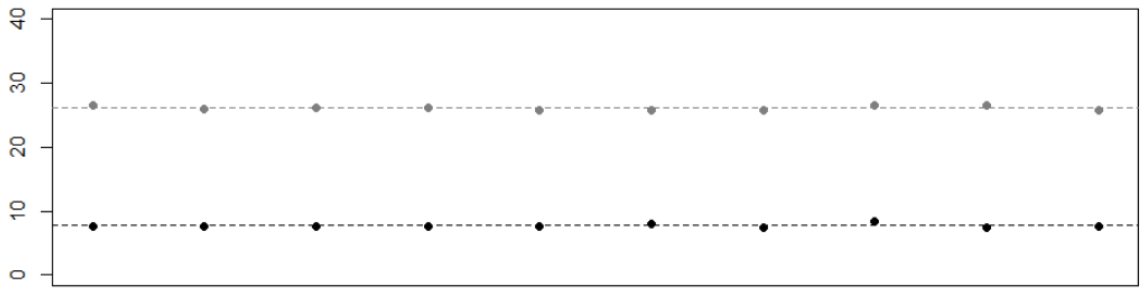
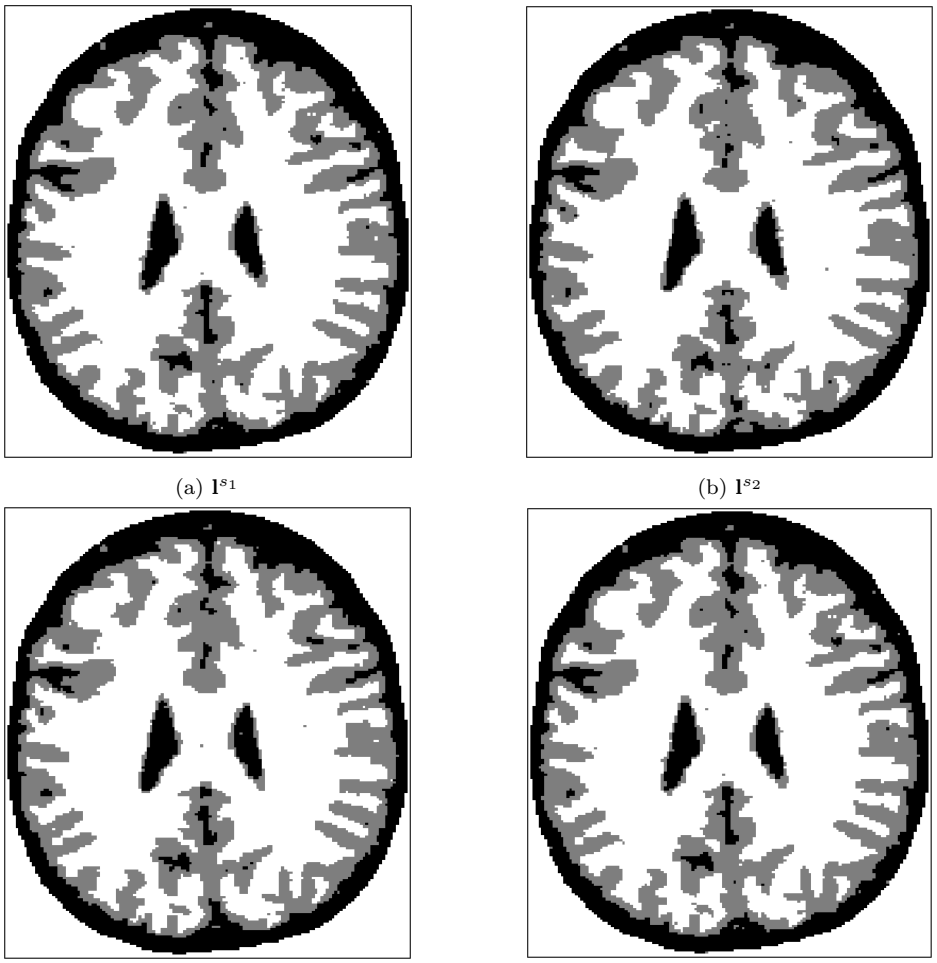


Figure 59: Time for the burn-in period in seconds for ten simulations of the single-site Gibbs algorithm (grey) and ten simulations of the profile block Gibbs algorithm (black). The straight lines are the averages.

The grey dots in Figure 59 represent the single-site Gibbs algorithm running times, and the black dots represent the profile block Gibbs algorithm running times. The average running time for the burn-in period at 7×10^5 iterations for the single-site Gibbs algorithm is 26.14 seconds (grey, dotted line), and the average running time for the burn-in period at 3×10^5 iterations for the profile block Gibbs algorithm is 7.71 seconds. The results from the benchmarking indicate that in addition to the fact that the single-site Gibbs algorithm requires a longer burn-in than the profile block Gibbs algorithm, each iteration also requires more time for the iteration definition in this study. Simulating from the profile block Gibbs algorithm is approximately three times faster than simulating from the single-site Gibbs algorithm.

Figure 60 displays four realizations generated by the posterior model using the profile block Gibbs algorithm (Algorithm 9). They are not identical but very similar. In this case, there is no true image, but in comparison to the training image $\mathbf{l}^0$ in Figure 55, they all seem to contain more grey nodes.



(a) $\mathbf{l}^{s_1}$ (b) $\mathbf{l}^{s_2}$

(c) $\mathbf{l}^{s_3}$ (d) $\mathbf{l}^{s_4}$

Figure 60: Four realizations generated from the posterior model.

To simulate predictions, a high number of realizations are required to calculate averages. After a burn-in period of 3×10^5 iteration in the profile block Gibbs algorithm, realizations were collected with 3×10^4 iteration spacing to enable a full sweep of the image. The MMAP is approximated using 1000 realizations and displayed in Figure 61.

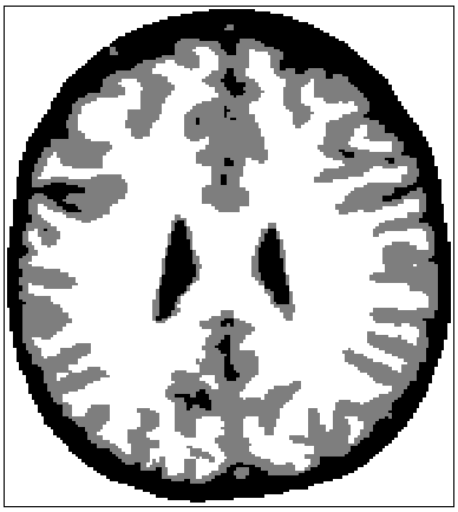
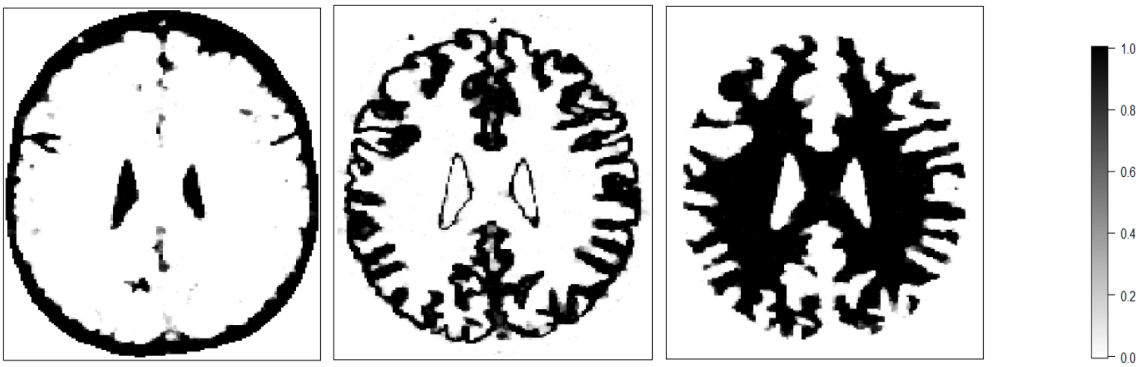


Figure 61: Approximated MMAP by taking the average of 1000 simulations.

The probability maps for the different possible states is displayed in Figure 62, and they are approximated using 1000 realizations. The probability maps indicate that the probability of identifying the true state of a grid node is high in most grid nodes supposedly located in an area where every neighbor is of the same state.



(a) Black (b) Grey (c) White

Figure 62: Approximated probability map considering 1000 simulations.

In comparison to the examples of MRFs, the brain MRI images are larger and the parameter estimates are more "extreme" values which have an impact on the algorithm efficiencies for both algorithms tested. For both algorithms, the convergence rate is rather slow. The brain MRI image is more circular than the images in the examples from MRFs, both the shape of the image and the pattern. The prior model may not be an ideal fit for the brain MRI as it contains a horizontal and

vertical closest neighbors based clique system. For instance, using cliques defined by angles can be more representative for the brain MRI image, and improve the efficiency of both algorithms. This creates a slightly more complicated model, so it is not tested in this study.

Comparing the two different algorithms, the results using the brain MRI image revealed that the profile block Gibbs algorithm appears as approximately three times more efficient than the single-site Gibbs algorithm. The results indicate that larger images implicate greater difference in efficiency in favor of the profile block Gibbs algorithm. For complicated and large images, the profile block Gibbs algorithm provides equally precisely analyzed images at a faster rate compared to the single-site Gibbs algorithm, which improves optimization in several scientific fields.

6 Conclusion

The recursive reverse algorithm is introduced and described in detail in Algorithm 1. This algorithm calculates the posterior model of an MRP and depending on the number of possible states, L, this is usually not a computationally demanding operation. Similarly, the marginal likelihood model for an MRP can be calculated by another recursive algorithm presented in Algorithm 6, and the parameters in the prior model are estimated by maximizing the marginal likelihood model with respect to the parameters. Parameters estimates are also generated using training images. The algorithms are demonstrated by examples of MRPs. The posterior model of an MRF is assessed iteratively, and the profile block Gibbs algorithm (Algorithm 9) simulates from the posterior model by considering the probability of one profile of the field given the remaining field. This algorithm is compared to the regular single-site Gibbs algorithm (Algorithm 8) considering the probability of one grid node given the remaining field, using examples of MRFs and a brain MRI image.

The examples of MRPs compare parameter estimates obtained by the marginal likelihood model with the estimates obtained by the truth as a training image. When the observations contain a small amount of noise, the parameter estimates obtained by the marginal likelihood are similar to those obtained by the training image. When the observations contain a higher level of noise, the parameters estimated by the marginal likelihood model contain higher variance and might differ more from the estimates generated by the truth as a training image. The examples of MRPs also revealed that when a profile consists of a higher amount of grid nodes, the algorithms provide more accurate parameter estimates, meaning the variances are smaller compared to the estimates from a profile consisting of a smaller amount of grid nodes.

Both the regular single-site Gibbs algorithm and the profile block Gibbs algorithm are iterative algorithms that converge towards the posterior model, and the main purpose of this study is to find out if the profile block Gibbs algorithm is more efficient. The examples of MRFs in Section 4.6 reveals that the difference in algorithm efficiency of the profile block Gibbs algorithm compared to the single-site Gibbs algorithm is little to no difference for some images. The results from analyzing the brain MRI image reveal that the profile block Gibbs algorithm is approximately three times more efficient than the single-site Gibbs algorithm.

The results encourage more research on the topic, as they indicate that the profile block Gibbs

algorithm is more efficient. The brain MRI image used in this study is of different geometry and size than the images created in the examples, it is preferable to use a more representative prior model for the analyzed image to improve the results as well as the algorithm efficiency. This study indicates that using the profile block Gibbs algorithm in image analysis may have a great impact on the efficiency, depending on the image size. To make further conclusions regarding which algorithm is preferable, more images with different properties and geometries must be analyzed.

References

[Basharin et al., 2004] Basharin, G. P., Langville, A. N., and Naumov, V. A. (2004). The life and work of a.a. markov. *Linear Algebra and its Applications*, 386:3 – 26. Special Issue on the Conference on the Numerical Solution of Markov Chains 2003.

[Baum and Eagon, 1967] Baum, L. E. and Eagon, J. A. (1967). An inequality with applications to statistical estimation for probabilistic functions of markov processes and to a model for ecology. *Bulletin of the American Mathematical Society*, 73:360–363.

[Baum et al., 1970] Baum, L. E., Petrie, T., Soules, G., and Weiss, N. (1970). A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains. *Ann. Math. Statist.*, 41(1):164–171.

[Bernstein, 1927] Bernstein, S. (1927). Sur l'extension du théorème limite du calcul des probabilités aux sommes de quantités dépendantes. *Mathematische Annalen*, 97(1):1–59.

[Besag, 1974] Besag, J. (1974). Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society: Series B (Methodological)*, 36(2):192–236.

[BrainWeb, 2020] BrainWeb (2020). Simulated brain database. `http://www.bic.mni.mcgill.ca/brainweb/`. Accessed: 2020.

[Dempster et al., 1977] Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22.

[Fjeldstad et al., 2020] Fjeldstad, T., Avseth, P., and Omre, H. (2020). A one-step bayesian inversion framework for three-dimensional reservoir characterization based on a gaussian mixture model – a norwegian sea demonstration. `https://doi.org/10.1190/geo2020-0094.1`. GEOPHYSICS 0: 1-68.

[Hurn et al., 2003] Hurn, M. A., Husby, O. K., and Rue, H. (2003). *A Tutorial on Image Analysis*, pages 87–141. Springer New York, New York, NY.

[Kindermann and Snell, 1980] Kindermann, R. and Snell, J. L. (1980). *Markov Random Fields and Their Applications*. American Mathematical Society, Providence, Rhode Island.

[Kolmogorov, 1931] Kolmogorov, A. (1931). Über die analytischen methoden in der wahrscheinlichkeitsrechnung. *Mathematische Annalen*, 104(1):415–458.

[Kouemou, 2011] Kouemou, G. L. (2011). History and theoretical basics of hidden markov models. In Dymarski, P., editor, *Hidden Markov Models*, chapter 1. IntechOpen, Rijeka.

[Markov, 1906] Markov, A. A. (1906). Rasprostranenie zakona bol'shih chisel na velichiny, zavisyaschie drug ot druga. *Izvestiya Fiziko-matematicheskogo obschestva pri Kazanskom universitete*, 2-ya seriya, tom 15, 9 4:135–156.

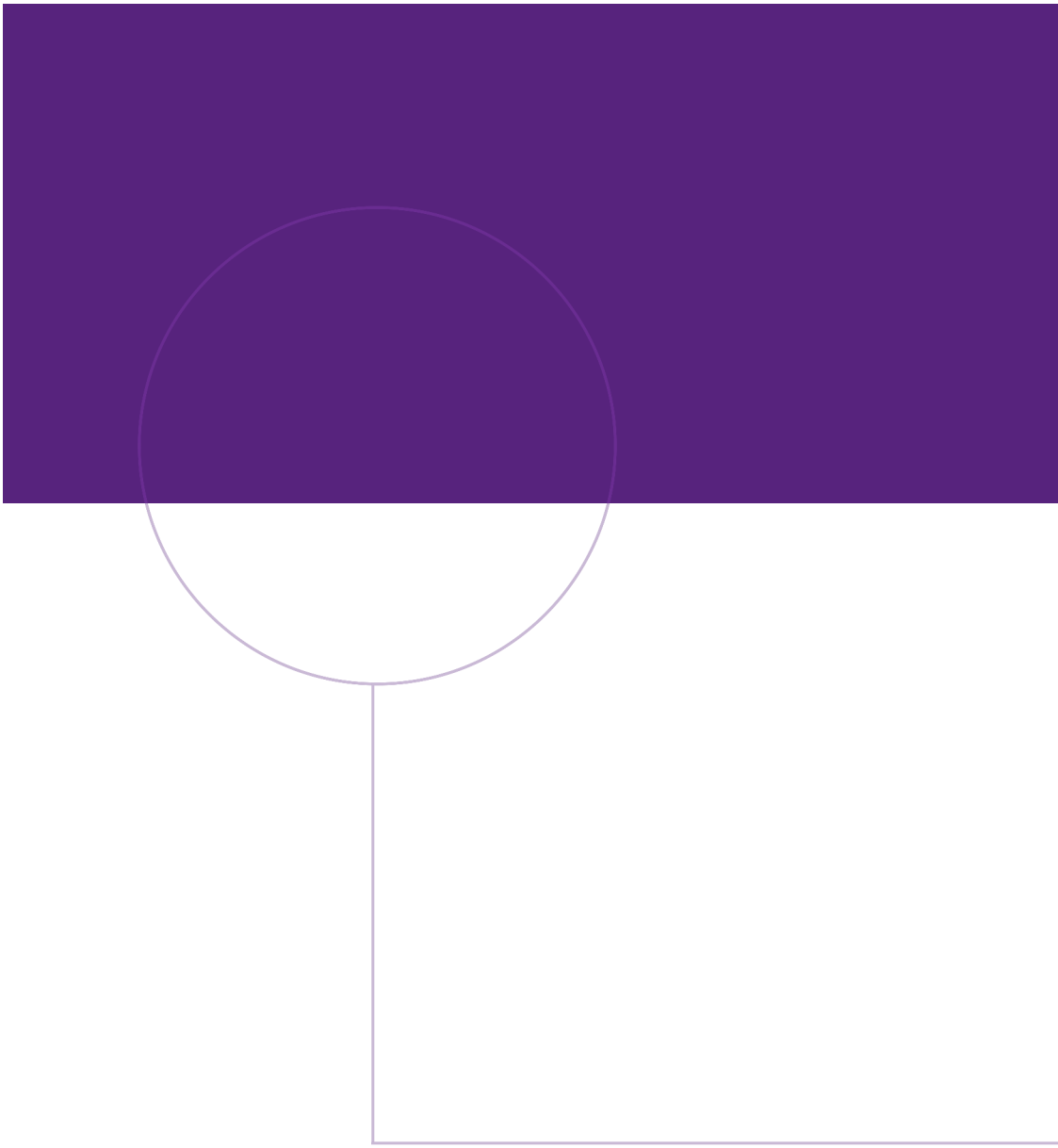
[Moja et al., 2019] Moja, S. S., Asfaw, Z. G., and Omre, H. (2019). Bayesian inversion in hidden markov models with varying marginal proportions. *Mathematical Geosciences*, 51:463–484.

[Norris, 1998] Norris, J. R. (1998). *Markov chains*, volume 2. Cambridge university press.

[Scott, 2002] Scott, S. L. (2002). Bayesian methods for hidden markov models: Recursive computing in the 21st century. *Journal of the American statistical Association*, 97(457):337–351.

[Udir, 2020] Udir (2020). Nye læreplaner - grunnskolen og gjennomgående fag vgo. `https://www.udir.no/laring-og-trivsel/lareplanverket/Nye-lareplaner-i-grunnskolen-og-gjennomgaende-fag-vgo/`. Utdanningsdirektoratet, Accessed: 25.11.2020.

[Viterbi, 1967] Viterbi, A. J. (1967). Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, 13(2):260–269.



NTNU

Norwegian University of
Science and Technology