

Preface

This thesis is based upon a study conducted during March–August 2010 with the supervision of Håkon Tjelmeland at Department of Mathematical Science, NTNU, Trondheim, Norway.

I would like to express my sincere gratitude to my supervisor Håkon Tjelmeland, without his advice and special support this thesis would never had become a reality.

I would like to thank Henning Omre and Ayele Taye for coordinating MASMO project and giving me this wonderful chance to pursue my Msc program in one of a prestigious University and work this thesis.

Finally, I wish to express my greatest thanks to my family, friend and colleagues, who have supported me, especially my mom Emau Reta, her life and strength is my day to day Energy and Motive also very much thank you my fiancée Selamawit Fikru.

In memory of my Father

Abstract

The purpose of this study was to develop a recursive algorithm for computing a maximum a posteriori (MAP) estimate of a binary Markov random field (MRF) by using the MAP-MRF framework. We also discuss how to include an approximation in the recursive scheme, so that the algorithm becomes computationally feasible also for larger problems. In particular, we discuss how our algorithm can be used in an image analysis setting. We consider a situation where an unobserved latent field is assumed to follow a Markov random field prior model, a Gaussian noise-corrupted version of the latent field is observed, and we estimate the unobserved field by the MAP estimator.

Contents

1	Introduction	1
2	Markov random field (MRF)	2
2.1	Binary Markov random fields	7
2.2	Computation of interaction parameters	10
2.3	Graphical representation of Markov random field	11
3	Image analysis problem	12
3.1	Markov random field prior model	13
3.2	The likelihood model	14
3.3	The posterior model	14
4	Recursive computation of maximum a posteriori probability (MAP) for a binary MRF	15
4.1	Beta ($\beta(\Lambda)$) computation	21
5	Approximate recursive computation	23
6	Concluding remarks	24

1 Introduction

Markov random field (MRF) theory is often used in conjunction with statistical decision and estimation theories so as to formulate an objective function in terms of established optimality principles. One of the main problems underlying many applications of MRFs amounts to inferring the hidden state x based on a Gaussian noise-corrupted observation y . This is often done through a maximum a posteriori (MAP) approach which finds \hat{x}_{MAP} by maximizing the posterior probability $P(x|y)$. i.e.

$$\hat{x}_{MAP} = \arg \max_x P(x|y). \quad (1)$$

The Maximum a posteriori (MAP) algorithm applied to Markov random field determines the most likely value of each state of the Markov random field with respect to its Markovian neighbourhood system. This is achieved by computing an a-posteriori probability (APP) for each state sequentially.

Estimation of binary Markov random field's state by MAP is formulated using the Bayesian setting despite the difficulty of the existing intractable normalizing constant in the binary MRF prior model. See Tjelmeland and Austad (2010) for several ways of dealing with the intractable normalizing constant. In this thesis, we focus on developing a recursive algorithm for MAP estimate of a hidden binary Markov random field by assuming the prior model, $P(x)$, to be a Markov random field and a Gaussian likelihood $P(x|y)$ with constant variance σ^2 . In the course of computing the recursive MAP estimation algorithm we adopt a canonical representation of the Markov random field.

Note that one of the famous models related to image analysis is the Ising model which is first proposed by Dr. Earnest Ising, which is a model defined for a binary Markov random field with a 2D $n \times n$ lattice. Most theories of the binary MRF including the neighbourhood system examples used in this thesis are defined related to the Ising model.

Basically, we have developed the estimation algorithm for an n -dimensional binary Markov random field using MAP-MRF framework (Geman and Geman 1984). An approximation is done by approximating every interaction parameter to zero if they got a value less than a given threshold value ϵ and

also the approximation to zero applies for higher order interaction parameters if all the corresponding lower order interactions are zero (approximated to zero). Note that the approximation is done in parallel with the recursive computation. As a scope of the algorithm we can apply it for estimating a unobserved latent state of a binary MRF from a noise-corrupted observation with a binary MRF prior model. In this thesis, image restoration from a Gaussian noise-corrupted observation with constant variance is the main application area of the MAP estimation method by considering prior distribution of the true image to be MRF.

This thesis is organized as follows. Section 2 contains MRF theory and properties including the MAP-MRF framework specifically for a binary MRF. Section 3 discusses the most important area of application for MRFs, which is Bayesian image analysis. In Section 4 we explain in detail the recursive computation of MAP for a binary MRF. Section 5 discusses approximations necessary to make the recursive computations feasible for larger lattices. The last section provides the closing remarks.

2 Markov random field (MRF)

Markov random field theory is a branch of probability theory. It provides a foundation for the characterization of contextual constraints and the derivation of the probability distribution of interacting features. MRF theory is often used in conjunction with statistical decision and estimation theories so as to formulate objective functions in terms of established optimality principles. Maximum a posteriori (MAP) estimation is a popular choice in MRF modeling. MRFs have been used widely to solve problems like image restoration and segmentation (Bouman, 1995), edge detection (Tardón, et al, 2006), object matching and recognition (Li, 1994) etc, including restoration of an unobserved image from noisy data which is going to be discussed in detail in this thesis.

MRFs and the MAP criterion together give rise to the MAP-MRF framework which is recommended in Geman and Geman (1984). In the MAP-MRF framework, the objective is to compute the joint posterior distribution of the MRF sites. From the prior distribution and the conditional probability distribution of the observed data, the form and the parameters of the

2 Markov random field (MRF)

posterior distribution are determined. Two major parts of the MAP-MRF modeling framework are to derive the form of the posterior distribution and to determine the parameters in it so as to completely define the a posterior probability (APP). Another important use of the MAP-MRF framework is to design an optimization algorithm for finding the maximum of the posterior distribution. In this thesis we consider how the MAP-MRF framework algorithm is developed to restore an unobserved image from noisy data.

The concept of MRFs is a generalization of a Markov process, which are widely used in sequence analysis. A Markov process is a sequence (chain) of random variables (x_1, \dots, x_n) defined on the time indices $1, \dots, n$. A Markov process is defined on a time domain, whereas an MRF is defined in space. We start by defining a random field.

Definition 1. *Let $x = (x_1, \dots, x_n)$ be a family of random variables defined on the set $S = \{1, \dots, n\}$ in which each random variable takes a value x_i in a domain D . Then the family x is called a Random Field.*

An MRF is also defined on a set $S = \{1, \dots, n\}$. Moreover, it is defined with respect to a so called neighbourhood system. When setting up MRF in practice we shall require quite specific information on the relative positions of sites, in order to assess the likely interdependency between the associated random variables. The sites in S are related to one another via the neighbourhood system.

Definition 2. *A neighbourhood system for S is defined as: $\mathcal{N} = \{\mathcal{N}_i \mid \forall i \in S\}$ where \mathcal{N}_i is the set of sites neighbouring i . The neighbouring relationship is required to have the following properties:*

1. *A site is not neighbouring itself: $i \notin \mathcal{N}_i$.*
2. *The neighbouring relationship is mutual for all distinct pairs of sites $i, i' \in S$: $i \in \mathcal{N}_{i'} \Leftrightarrow i' \in \mathcal{N}_i$.*

As the simplest example, suppose that $\{x_1, \dots, x_n\}$ is a Markov chain. Then it can be shown that site i ($2 \leq i \leq n - 1$) has neighbours $i - 1$ and $i + 1$ whilst the sites 1 and n have the single neighbours 2 and $n - 1$, respectively.

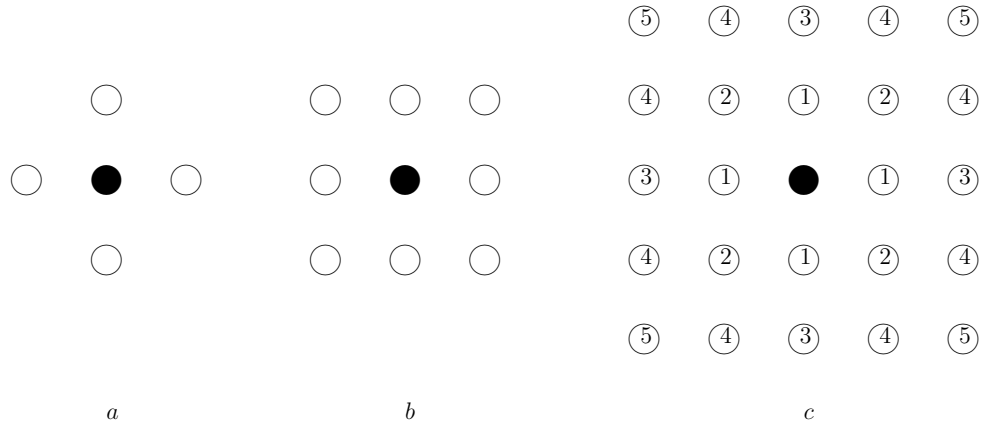


Figure 1: The 1st, 2nd and n^{th} order ($n = 1, 2, 3, 4, 5$) neighbourhood structure for imaging.

Commonly used neighbourhood structures in imaging, i.e. when the sites form a finite rectangular lattice $S = \{(u, v), 1 \leq u, v \leq n\}$ corresponding to the pixels of an $n \times n$ image in the 2D plane, are

- First order neighbourhood system (4-neighbourhood) when every interior site have four neighbours, see Figure 1 *a*.
- Second order neighbourhood system (8-neighbourhood) when every interior site have eight neighbours, see Figure 1 *b*.

Generally, for $n = 1, \dots, 5$ order neighbourhood system see Figure 1 *c*. The numbers $n = 1, \dots, 5$ in Figure 1 *c* show the outermost neighbouring sites to the black site in the n^{th} order neighbourhood system. For example, in case of $n = 3$ which is the 3rd order neighbourhood to the black site, all sites assigned with 3, 2 and 1 are elements of the 3rd order neighbourhood to the black site.

For a regular lattice (a lattice with sites distributed regularly) S , we can generally define the neighbouring site to i as follows.

2 Markov random field (MRF)

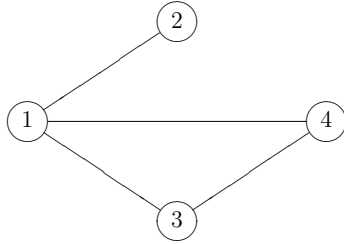


Figure 2: An undirected graph $V = (S, E)$ visualizing the neighbourhood system for a toy example of a Markov random field with $S = \{1, 2, 3, 4\}$ and $\mathcal{N} = \{\mathcal{N}_1, \dots, \mathcal{N}_4\} = \{\{2, 3, 4\}, \{1\}, \{1, 4\}, \{1, 3\}\}$.

Definition 3. For a regular lattice S , the set of neighbours of i is defined as the set of sites within a radius of \sqrt{r} from i , i.e.

$$\mathcal{N}_i = \{i' \in S \mid [\text{dist}(\text{site } i', \text{site } i)]^2 \leq r, \quad i' \neq i\}, \quad (2)$$

where $\text{dist}(a, b)$ denotes the Euclidean distance between a and b while r takes a positive value.

According to Hurn et al (2003), we can have a graphical display of a neighbourhood system \mathcal{N} by an undirected graph $V = (S, E)$ which has one vertex corresponding to each site in S and an edge between any pairs of sites that are neighbours, i.e. $E = \{(k, l) \mid k \in \mathcal{N}_l\}$. For a site without any neighbours we just put it as a vertex with no edge to other sites. For example, let us consider a toy example with $S = \{1, 2, 3, 4\}$ and $\mathcal{N} = \{\mathcal{N}_1, \dots, \mathcal{N}_4\} = \{\{2, 3, 4\}, \{1\}, \{1, 4\}, \{1, 3\}\}$, then see the undirected graph $V = (S, E)$ in Figure 2 for the toy example.

According to the above explanation of the n^{th} order neighbourhood, sites at the boundary, at the corner and those at the interior of the lattice may have different number of neighbours despite of having same order of neighbourhoods. For example in a first order neighbourhood of a rectangular

lattice the interior sites have four neighbours whereas sites at the boundary have three neighbouring sites and those at the corner have two. According to Hurn et al (2003), the value at the boundary may have a significant effect on the marginal distribution of interior sites. Thus we can use the upcoming Torus boundary condition to solve this unequal number of neighbours.

The *torus boundary condition* for a rectangular lattice allows any site in the lattice to have equal number of neighbours regardless of its position during a given order of neighbourhood. For example, in the first order neighbourhood of an $n \times n$ lattice the four neighbours of sites (u, n) , ($u \neq 1$ and n) are $(u - 1, n)$, $(u + 1, n)$, $(u, n - 1)$ and $(u, 1)$. Similarly, the site at $(1, n)$ has four neighbours i.e. $(2, n)$, (n, n) , $(1, n - 1)$ and $(1, 1)$.

In addition to the neighbourhood system, we have another set of sites property which is called a clique, see the upcoming definition of clique.

Definition 4. *Any set of sites which either consists of a single site or in which every site is a neighbour of every other site in the set is called a clique.*

According to Definition 4, a clique consists of either a single site, $c_1 = \{i\}$, a pair of neighbouring sites, $c_2 = \{\{i, j\}; j \in \mathcal{N}_i\}$, a triple of neighbouring sites, $c_3 = \{i, j, k; i, j, k \text{ are neighbours to one another}\}$ and so on depending on the size of S and existence of the clique combinations. The collection of all cliques for (S, \mathcal{N}) is called the clique system and is denoted as \mathcal{C} . The cliques of the toy example given above are $\mathcal{C} = \{\emptyset, \{1\}, \{2\}, \{3\}, \{4\}, \{1, 2\}, \{1, 3\}, \{1, 4\}, \{3, 4\}, \{1, 3, 4\}\}$.

Using the definition of a neighbourhood system given above, we define a Markov random field as follows.

Definition 5. *A random field $x = (x_1, \dots, x_n)$ is said to be a Markov random field (MRF) on $S = \{1, \dots, n\}$ with respect to a neighbourhood system \mathcal{N} if and only if the following holds true*

- $P(x) > 0, \quad \forall x \in D^n, \quad (\text{positivity condition})$
- $P(x_i | x_{S \setminus \{i\}}) = P(x_i | x_{\mathcal{N}_i}), \quad (\text{Markovian condition})$

where $x_{S \setminus \{i\}} = (x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n)$ and $x_{\mathcal{N}_i} = (x_j, j \in \mathcal{N}_i)$.

2.1 Binary Markov random fields

The Markovian condition is the most essential and needs to be checked or validated in MRF before further use. Besag (1974) discusses the positivity condition as follows. If values x_1, \dots, x_n individually occur at the sites $1, \dots, n$, respectively then, the configuration (x_1, \dots, x_n) also must have a positive probability for occurring. Formally, if $P(x_i) > 0$ for each i , then $P(x_1, \dots, x_n) > 0$. In other words, when the positivity condition is satisfied, the joint probability of any random field in the domain is uniquely determined by its local conditional probability (Besag 1974).

The *Hammersley-Clifford* theorem gives then the most general definition form of an MRF.

Theorem 1. (*Hammersley-Clifford*): *A distribution satisfying $P(x) > 0$ for all configurations $x \in D^n$ is a Markov random field if, and only if, it has a joint distribution of the form*

$$P(x) = \text{Const} \exp \left(- \sum_{c \in \mathcal{C}} Z_c(x_c) \right). \quad (3)$$

for some set of functions $Z_c(x_c)$, $c \in \mathcal{C}$, where \mathcal{C} is the set of all cliques and *Const* is the normalizing constant.

To prove this theorem, one must show two things. The first and easiest one is to let the joint distribution of x be given by (3) then, compute the full conditional and show that the full conditional fulfills the Markov property. The second way, which is the tough one, is to let the full conditional be the expression of an MRF and then, compute the joint distribution and see that it must be look like (3). For this we refer to Besag (1974) and Hurn et al (2003).

2.1 Binary Markov random fields

Above we have introduced MRFs in general. In this section we will consider in detail binary MRFs. Let, as above, $S = \{1, \dots, n\}$ be a discrete set of n sites. To each site $i \in S$ we associate a binary variable $x_i \in D = \{0, 1\}$, which is the reason to say an MRF is binary and let $x = (x_1, \dots, x_n) \in D^n = \{0, 1\}^n$. We also use the notation $x_\Lambda = (x_i, i \in \Lambda)$ and $x_{-\Lambda} = x_{S \setminus \Lambda}$ for $\Lambda \subset S$, and $x_{-i} = x_{S \setminus \{i\}}$ for $i \in S$. Note that x_Λ and $x_{-\Lambda}$ are disjoint and that

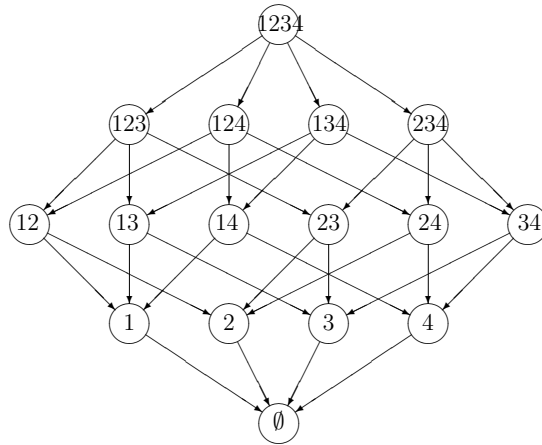


Figure 3: The graph $\mathcal{G}(\mathcal{P}(S))$ for $S = \{1, 2, 3, 4\}$.

$$x = (x_\Lambda, x_{-\Lambda}).$$

Given a set S , the power set of S , $\mathcal{P}(S)$, is the set of all subsets of S including the empty set and the set S itself i.e. $\mathcal{P}(S) = \{\Lambda \mid \Lambda \subseteq S\}$. For example let us again consider our toy example of $S = \{1, 2, 3, 4\}$. Then $\mathcal{P}(S) = \{\emptyset, \{1\}, \{2\}, \{3\}, \{4\}, \{1, 2\}, \{1, 3\}, \{1, 4\}, \{2, 3\}, \{2, 4\}, \{3, 4\}, \{1, 2, 3\}, \{1, 2, 4\}, \{1, 3, 4\}, \{2, 3, 4\}, \{1, 2, 3, 4\}\}$. See Figure 3 for a graphical representation of the power set of the toy example.

An important aspect of MRFs in image analysis is to simplify the challenge we face when modeling the legitimate joint distribution of the sites of an image. The MRF modeling approach this issue through the full conditionals, the conditional distribution of a site given the rest, and breaks down the problem into a more manageable task. Using an MRF we focus on the behavior of the i^{th} site if we know the configuration of its neighbours. For example, the probability of x_i to be zero may increase with the increase number of neighbouring sites to site i that are zero. One possibility would be to

2.1 Binary Markov random fields

express this by

$$P(x_i = 0 | k \text{ zero labeled neighbours}) \propto \exp(\beta k), \quad (4)$$

where β is a positive parameter.

Although we might therefore approach the MRF modeling by specifying a neighbourhood and the full conditionals for each site like (4), the uniqueness and legitimacy of the joint distribution is questionable. By the *Hammersely-Clifford theorem* one is ensured a legitimate joint distribution by specifying the model via the set of functions $Z_c(x_c)$, $c \in \mathcal{C}$.

Let us assume we are given a probability distribution for x denoted by $P(x)$, $x \in D^n = \{0, 1\}^n$. We may express $P(x)$ as

$$P(x) = Const \exp(-U(x)), \quad (5)$$

where the sum to one property of $P(x)$ with respect to x is maintained by $Const$ which is called a normalizing constant.

The positivity condition, $P(x) > 0$, is maintained by using the exponential, $\exp(\cdot)$, of the expression of the energy function $U(x)$. A canonical representation of the energy function in terms of a set of interaction parameters, $\beta = \{\beta(\Lambda), \Lambda \subseteq S\}$, is defined as

$$U(x) = \sum_{\Lambda \subseteq S} \beta(\Lambda) \prod_{i \in \Lambda} x_i. \quad (6)$$

Theorem 2. *Let x be a Markov random field with respect to a neighbourhood system \mathcal{N} , let \mathcal{C} be the corresponding set of all cliques, and let $\{\beta(\Lambda), \Lambda \subseteq S\}$ be the interaction parameter between or among the subset of S . Then $\beta(\Lambda) = 0$ for all $\Lambda \notin \mathcal{C}$.*

A proof is given in Tjelmeland and Austad (2010).

In (6) we define the energy function $U(x)$ in terms of a set of interaction parameters, $\beta = \{\beta(\Lambda), \Lambda \subseteq S\}$. According to Theorem 2, if x is known to be Markov with respect to a neighbourhood system \mathcal{N} , there will be set of Λ , $\Lambda \subseteq S$ such that $\beta(\Lambda) = 0$. In short $\beta(\Lambda) = 0$ for all $\Lambda \notin \mathcal{C}$. If $\beta(\Lambda) = 0$ for a given Λ , $\Lambda \subseteq S$, including those zero valued interaction parameter in

(6) have no use. Deducting those zero valued interaction parameters from (6) we get

$$U(x) = \sum_{\Lambda \subseteq B} \beta(\Lambda) \prod_{i \in \Lambda} x_i, \quad (7)$$

where

$$B = \bigcup_{\Lambda \in \mathcal{P}(S): \beta(\Lambda) \neq 0} \Lambda. \quad (8)$$

2.2 Computation of interaction parameters

We have seen before that the joint distribution of the MRF can be expressed like (5) using the energy function (7). To compute the MAP estimate we first have to represent the given energy function $U(x)$ on the form (7) and then using this $U(x)$ we have to compute the interaction parameters $\beta(\Lambda)$ for all $\Lambda \in \mathcal{C}$.

As before, let $x = (x_1, \dots, x_n)$ where $x_i \in D = \{0, 1\}$ for $i \in S = \{1, \dots, n\}$ and let $x(\Lambda) = (I(1 \in \Lambda), \dots, I(n \in \Lambda))$ where $I(\cdot)$ is an indicator function defined as

$$I(a) = \begin{cases} 1, & \text{if condition of "a" is true,} \\ 0, & \text{if condition of "a" is false.} \end{cases} \quad (9)$$

Note that, according to the definition of $I(\cdot)$, if $\Lambda = S$ we have $x(\Lambda) = (1, \dots, 1)$ and similarly if $\Lambda = \emptyset$ we have $x(\Lambda) = (0, \dots, 0)$.

To compute the interaction parameter $\beta(\cdot)$ from (6), insert $x(\Lambda)$ in place of x in (6) to get

$$U(x(\Lambda)) = \sum_{A \subseteq \Lambda} \beta(A) \prod_{i \in A} I(i \in A). \quad (10)$$

Note that in (10) we get $I(i \in A) = 1$ because $A \subseteq \Lambda$. Thereby we can rewrite the expression to get

$$U(x(\Lambda)) = \sum_{A \subseteq \Lambda} \beta(A) + \beta(\Lambda). \quad (11)$$

Finally, the interaction parameter of a given Λ , $\Lambda \subseteq S$ becomes

$$\beta(\Lambda) = U(x(\Lambda)) - \sum_{A \subseteq \Lambda} \beta(A). \quad (12)$$

2.3 Graphical representation of Markov random field

When we compute the interaction parameters using (12) we have to compute them recursively depending on the element size in Λ and go in increasing order. We first compute $\beta(\emptyset)$, next $\beta(\Lambda)$ for $|\Lambda| = 1$ and then next for $|\Lambda| = 2$ and goes till $|\Lambda| = n$. On the course of computation of the interaction parameters by the above method we can note that it is not computationally efficient because the same sum will be computed repeatedly. We recommend to have a look in Tjelmeland and Aустad (2010) for a computationally more efficient method for computing the interaction parameters from the given energy function.

2.3 Graphical representation of Markov random field

Using all Λ , $\Lambda \in \mathcal{P}(S)$, we can have a graphical representation of $\mathcal{P}(S)$ denoted as $\mathcal{G}(\mathcal{P}(S))$ with every Λ represented as a vertex in the graph, $\mathcal{G}(\mathcal{P}(S))$. See Figure 3 for the toy example of $S = \{1, 2, 3, 4\}$. For every $\Lambda \in \mathcal{P}(S)$ there is a corresponding interaction parameter value $\beta(\Lambda)$ which is computed from the energy function $U(x)$ using (12). By computing, storing and assigning the corresponding interaction parameter value of each vertex of the graph $\mathcal{G}(\mathcal{P}(S))$ we form a new named graph called a vertex-weighted graph $\mathcal{G}(\mathcal{P}(S), \beta)$. Principally and according to Tjelmeland and Aустad (2010) the joint distribution $P(x)$ can be represented by the vertex-weighted graph, $\mathcal{G}(\mathcal{P}(S), \beta)$.

According to the definition that x is Markov random field and the use of Theorem 2, $\beta(\Lambda) = 0$ if $\Lambda \notin \mathcal{C}$ where \mathcal{C} is the clique system and $\Lambda \in \mathcal{P}(S)$ which reduces the vertex size of the vertex-weighted graph $\mathcal{G}(\mathcal{P}(S), \beta)$ and also substantiate the usefulness of defining B as we did in (8). Using the vertexes, $\Lambda \in B$ we can re-define a new vertex-weighted graph, $\mathcal{G}(B, \beta)$, which is used to represent our joint distribution of the Markov random field x .

Figure 4 shows the unweighted graph $\mathcal{G}(B)$ for the toy example $S = \{1, 2, 3, 4\}$ when $\beta(\Lambda) \neq 0$ for $\Lambda \in \{\{1, 2\}, \{1, 3\}, \{1, 4\}, \{3, 4\}\}$ and $\beta(\Lambda) = 0$ for $\Lambda \in \{\{2, 3\}, \{2, 4\}, \{1, 2, 3\}, \{1, 2, 4\}, \{1, 3, 4\}, \{2, 3, 4\}\}$. Note that in case of $\mathcal{G}(B)$, which is the reduced version of $\mathcal{G}(S)$ by assuming Markovianity, we should always contain the vertexes having \emptyset and $\{i\}$, $i \in S$ in the graph $\mathcal{G}(B)$.

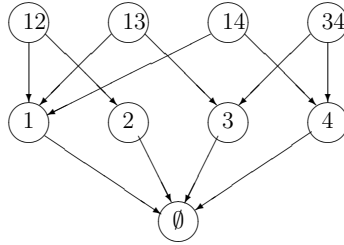


Figure 4: The unweighted graph $\mathcal{G}(B)$ for the toy example.

3 Image analysis problem

One unifying theme for the different types of imaging problems, according to Hurn et al (2003), is that they may generally be regarded as being of the forms

$$Signal = Image \otimes Noise \tag{13}$$

or

$$Signal = f(Image) \otimes Noise \tag{14}$$

where \otimes represents a suitable combination operator and the function $f(\cdot)$ indicates that the signal is not of the same format as the original image.

In this thesis we are assuming that the signals have the same format as the original image with additive effect of the noise on the image i.e.

$$Signal = Image + Noise. \tag{15}$$

We also assume that the images are degraded by additive Gaussian noise and that the noises are uncorrelated to each other with zero mean and constant variance σ^2 . To make the understanding and computational representation easy we denote the signal which is the data by $y = (y_1, \dots, y_n)$

3.1 Markov random field prior model

and the underlying unobserved image by $x = (x_1, \dots, x_n)$, and the noise by $\epsilon = (\epsilon_1, \dots, \epsilon_n)$, where

$$\epsilon_i \stackrel{iid}{\sim} N(0, \sigma^2). \quad (16)$$

Thus the observed data, $y = x + \epsilon$, have a Gaussian property i.e

$$y_i \sim N(x_i, \sigma^2), \quad (17)$$

for $i \in S$. Note that $\forall \{i, j\} \in S$ and $i \neq j$ then y_i and y_j are uncorrelated.

Remembering our main objective of the thesis, we are looking to apply the MAP estimation method on the unobserved latent state of an image x . We have assumed the noise (error) to be Gaussian with zero mean and constant variance σ^2 and the data is given or is observed from the latent hidden state of the image. MAP estimation method differs from the maximum likelihood very basically because of the prior information that we have in case of MAP estimation method. Choosing the prior model is one of the most critical aspects of Bayesian analysis. The Bayesian approach for estimating the unobserved state of the image has a benefit in image analysis and interpretation because it permits the use of prior knowledge concerning the situation under study and incorporates that prior knowledge into data analysis. According to Hanson (1993), MAP estimation has been successfully employed in spatial settings and his opinion also support that the MAP formulation (Bayes approach in general) is well-suited to the restoration of images.

3.1 Markov random field prior model

The prior model expresses the degree of certainty concerning the situation before the data are taken. The model we select for the prior has a very direct influence on the outcome we expect. If the prior is too specific and too strong, the restoration of the unobserved image can only closely resemble what is expected, see in Hanson (1993). Furthermore, departures from what is anticipated may be missed. The strength of the prior relative to the likelihood is a critical parameter in any Bayesian analysis. How the strength of the prior distribution is being determined, is a basic question in Bayesian analysis. According to Hanson (1993), as the strength of the prior is increased, the MAP solution is biased. However, using a very weak prior to solve the biasedness the effect of the prior becomes negligible. Gull (1989) has offered a mean to determine the prior's strength based on the posterior,

viewed as a function of the strength of the entropic prior.

In this particular thesis, we deal with restoration of an unobserved image x which is assumed to be an MRF from the specified prior. Thus we assume that the prior is an MRF. Therefore, we are assuming our prior model to be an MRF expressed as (5). i.e.

$$P(x) = Const \exp \left\{ - \sum_{\Lambda \subseteq S} \beta(\Lambda) \prod_{i \in \Lambda} x_i \right\}. \quad (18)$$

3.2 The likelihood model

The likelihood model encompasses the information contained in the new data. As we have assumed before, the data is Gaussian with zero mean, constant variance σ^2 and uncorrelated additive noise terms,

$$y = x + \epsilon. \quad (19)$$

Applying the Gaussian distribution and the conditioning on x it can be shown that the observed data y will have a Gaussian distribution with mean x and variance σ^2 i.e. $y \sim N(x, \sigma^2)$. Therefore our likelihood model can be expressed as

$$P(y|x) = Const \exp \left\{ \frac{-1}{2\sigma^2} \sum_{i=1}^n (y_i - x_i)^2 \right\}. \quad (20)$$

3.3 The posterior model

When both the prior and likelihood models are known, the best result is achieved by maximizing a Bayes criterion. MAP is the most popular estimation method if both the likelihood and prior models are defined. However, If the knowledge about the data is available but not the prior model, the maximum likelihood estimation can be used.

The Bayes law states that the posterior probability distribution is proportional to the product of the likelihood and the prior distributions. From (18) and (20) we get the posterior distribution model $P(x|y)$ to be

4 Recursive computation of maximum a posteriori probability (MAP) for a binary MRF

$$\begin{aligned}
 P(x|y) &= \text{const } P(y|x)P(x) \\
 &= \text{const } \exp \left\{ \sum_{i=1}^n \left(-\beta(i) + \frac{2y_i - 1}{2\sigma^2} \right) x_i - \sum_{\substack{\Lambda \subseteq S \\ |\Lambda| \neq 1}} \beta(\Lambda) \prod_{k \in \Lambda} x_k \right\}.
 \end{aligned} \tag{21}$$

The expression in the exponent can be rewritten in the canonical or standard form discussed above. This representation will help us to understand the expression better. Thus, we denote the posterior interaction parameter by $\beta^*(\Lambda)$ and get

$$\beta^*(\Lambda) = \begin{cases} \beta(\Lambda) - \frac{2y_i - 1}{2\sigma^2} & \text{if } \Lambda = \{i\}, \quad i \in S, \\ \beta(\Lambda) & \text{if } |\Lambda| \neq 1. \end{cases} \tag{22}$$

Note here that $\beta(\Lambda)$ denotes the interaction parameter given in the prior distribution $P(x)$, that is before the observation effect have been observed, whereas $\beta^*(\Lambda)$ is going to be the new interaction parameter, that is the interaction parameter after the effect of the observation. Note that the interaction parameter of the prior model $\beta(\Lambda)$ only differ from the interaction parameter of the posterior $\beta^*(\Lambda)$ when the number of element in Λ equals one.

Applying the new interaction parameter $\beta^*(\Lambda)$ in (21), we can rewrite the posterior distribution as

$$P(x|y) = \text{const } \exp \left\{ - \sum_{\Lambda \subseteq S} \beta^*(\Lambda) \prod_{k \in \Lambda} x_k \right\}. \tag{23}$$

We see that $P(x|y)$ is of the same form as the prior $P(x)$.

4 Recursive computation of maximum a posteriori probability (MAP) for a binary MRF

Maximum a posteriori probability (MAP) is a common image estimation method in image analysis. In Bayesian analysis, maximum a posteriori probability (MAP) estimate is the mode of the posterior distribution. In this thesis we are in need of estimating an image x whose posterior distribution is given by (23) using MAP estimation method. Note that we are dealing

with MRFs thus we can use (7) to redefine the energy function of (23) and using this we compute MAP as follows.

The MAP estimator of $P(x|y)$ is given as

$$\begin{aligned}\hat{x}_{MAP} &= \arg \max_x P(x|y) \\ &= \arg \max_x \exp \left\{ - \left[\sum_{\Lambda \in B} \beta^*(\Lambda) \prod_{i \in \Lambda} x_i \right] \right\}.\end{aligned}\quad (24)$$

We can redefine the maximum a posterior probability estimation method of the image x for the specific expression given above as

$$\hat{x}_{MAP} = \arg \min_x \sum_{\Lambda \in B} \beta^*(\Lambda) \prod_{i \in \Lambda} x_i. \quad (25)$$

Though we define the MAP estimation of a binary MRF by (24) or (25), due to the Markovian neighbourhood character of each $x_i : i \in S$ the computation needs an ordered recursive step. To compute \hat{x}_{MAP} we recursively have to maximize $P(x|y)$ with respect to each state (x_1, \dots, x_n) to get $(\hat{x}_1, \dots, \hat{x}_n)$ and these as a whole gives \hat{x}_{MAP} . However, each x_i at every i^{th} step is influenced by its neighbouring sites which are not yet been estimated since we start the estimation from $i = 1$, till $i = n$ in increasing order.

To compute an estimate of x_i , we first have to find and store the resulting estimate of x_i for all possible combination of values for the neighbouring sites. Let $\psi_i(x_{\mathcal{N}_i})$ denote this estimate of x_i . This can also be organized in a table where the first $|\mathcal{N}_i|$ columns gives the values of $x_{\mathcal{N}_i}$ and the last column gives $\psi_i(x_i)$. See Table 1 for the toy examples distribution maximized with respect to x_1 .

Let us consider the toy example again. We know from the toy example that $\mathcal{N}_1 = \{2, 3, 4\}$. We start by computing expression (25) with respect to x_1 and denote the result as $\psi_1(x_{\mathcal{N}_1})$ i.e.

4 Recursive computation of maximum a posteriori probability (MAP) for a binary MRF

x_2	x_3	x_4	$\psi_1(x_{\mathcal{N}_1})$
0	0	0	$\psi_1(0, 0, 0)$
0	0	1	$\psi_1(0, 0, 1)$
0	1	0	$\psi_1(0, 1, 0)$
1	0	0	$\psi_1(1, 0, 0)$
0	1	1	$\psi_1(0, 1, 1)$
1	0	1	$\psi_1(1, 0, 1)$
1	1	0	$\psi_1(1, 1, 0)$
1	1	1	$\psi_1(1, 1, 1)$

Table 1: Combination of $x_{\mathcal{N}_1} = \{0, 1\}^{|\mathcal{N}_1|}$ with estimate of x_1 , $\hat{x}_1 = \psi_1(x_{\mathcal{N}_1})$

$$\begin{aligned}
 \psi_1(x_{\mathcal{N}_1}) &= \arg \min_{x_1} \left\{ \sum_{\Lambda \in B_{-1}} \beta^*(\Lambda) \prod_{i \in \Lambda} x_i + \sum_{\Lambda \in B_1} \beta^*(\Lambda) \prod_{i \in \Lambda} x_i \right\} \\
 &= \arg \min_{x_1} \left\{ \sum_{\Lambda \in B_1} \beta^*(\Lambda) \prod_{i \in \Lambda} x_i \right\}. \tag{26}
 \end{aligned}$$

where $B_1 = \{\Lambda \in B \mid 1 \in \Lambda\}$ and $B_{-1} = \{\Lambda \in B \mid 1 \notin \Lambda\}$. We know from the toy example that x_1 is a neighbour of x_2, x_3 and x_4 which gives that $\psi_1(x_{\mathcal{N}_1})$ of (26) is a function of x_2, x_3 and x_4 .

To store the value of $\psi_1(x_{\mathcal{N}_1})$ we define a table to contain all the possible combinations of $x_{\mathcal{N}_1}$ and concatenate one last column for storing $\psi_1(x_{\mathcal{N}_1})$ i.e. for each combination of (x_2, x_3, x_4) given in Table 1 we compute and store the MAP estimate of x_1 , $\psi_1(x_{\mathcal{N}_1})$. To see how $\psi_1(x_{\mathcal{N}_1})$ is computed we define a function $f(\cdot)$ representing the kernel of (26) i.e.

$$f(x_{\mathcal{N}_1}, x_1) = \sum_{\Lambda \in B_1} \beta^*(\Lambda) \prod_{i \in \Lambda} x_i, \tag{27}$$

where $x_{\mathcal{N}_1} = (x_2, x_3, x_4)$. Taking all possible values of $x_{\mathcal{N}_1}$ from Table 1 we compute and compare $f(x_{\mathcal{N}_1}, x_1 = 0)$ and $f(x_{\mathcal{N}_1}, x_1 = 1)$ for each row. We assign $\psi_1(x_{\mathcal{N}_1}) = 1$ if $f(x_{\mathcal{N}_1}, x_1 = 1) < f(x_{\mathcal{N}_1}, x_1 = 0)$, otherwise $\psi_1(x_{\mathcal{N}_1}) = 0$. For example, comparing the results obtained for $(x_2 = 1, x_3 = 0, x_4 = 0)$ and $x_1 = 0$ versus $(x_2 = 1, x_3 = 0, x_4 = 0)$ and $x_1 = 1$ at (27) we

come up with the result of x_1 with the smaller $f(\cdot)$ to be stored at $\psi_1(1, 0, 0)$. In other words, $\psi_1(1, 0, 0)$ at Table 1 is the maximum probable value of x_1 when $(x_2 = 1, x_3 = 0, x_4 = 0)$. Note that there may be a possibility of $f(x_{\mathcal{N}_1}, x_1 = 0)$ and $f(x_{\mathcal{N}_1}, x_1 = 1)$ to be equal. In such a case we recommend to prefer one of them freely for $\psi_1(x_{\mathcal{N}_1})$.

Once we have computed all $\psi_1(x_{\mathcal{N}_1})$ as we did in Table 1 for the toy example, we are done finding the estimate value of x_1 which maximize $P(x|y)$ with respect to the others. According to the MAP recursive property, once we are done with x_1 we should proceed to the next. Once we have found $\psi_1(x_{\mathcal{N}_1})$, the value of x_1 which maximize $P(x|y)$ with respect to the neighbours, we insert that estimated value of x_1 in $P(x|y)$ in place of x_1 and update $P(x|y)$ for the next step of estimation or maximization. Thus, we get

$$\max_{x_1} P(x|y) = P(x_1 = \psi_1(x_{\mathcal{N}_1}), x_2, \dots, x_n|y). \quad (28)$$

We proceed by computing $P(x_1 = \psi_1(x_{\mathcal{N}_1}), x_2, \dots, x_n|y)$. Note that a more detailed derivation of the following expressions are given in Section 4.1. We get that (28) can be reexpressed as

$$\max_{x_1} P(x|y) = \text{const} \exp \left\{ - \left[\sum_{\Lambda \in B_{-1}} \beta_1^*(\Lambda) \prod_{i \in \Lambda} x_i \right] - A \right\}, \quad (29)$$

where

$$A = \left\{ \sum_{\Lambda \in B_1} \beta_1^*(\Lambda) \prod_{\substack{j \in \Lambda \\ x_1=1}} x_j \right\} \left[\psi_1(0, \dots, 0) \prod_{j \in \mathcal{N}_1} (1 - x_j) + \dots + \psi_1(1, \dots, 1) \prod_{j \in \mathcal{N}_1} x_j \right] \quad (30)$$

and $\beta_1^*(\Lambda) = \beta^*(\Lambda)$, $\Lambda \subseteq S$. For the toy example, A will be

4 Recursive computation of maximum a posteriori probability (MAP) for a binary MRF

$$\begin{aligned}
A = & \left\{ \sum_{\Lambda \in B_1} \beta^*(\Lambda) \prod_{\substack{i \in \Lambda \\ x_1=1}} x_i \right\} \times \left\{ \psi_1(0,0,0)(1-x_2)(1-x_3)(1-x_4) \right. \\
& + \psi_1(1,0,0)x_2(1-x_3)(1-x_4) + \psi_1(0,1,0)x_3(1-x_2)(1-x_4) \\
& + \psi_1(0,0,1)x_4(1-x_2)(1-x_3) + \psi_1(1,1,0)x_2x_3(1-x_4) \\
& + \psi_1(1,0,1)x_2x_4(1-x_3) + \psi_1(0,1,1)x_3x_4(1-x_2) \\
& \left. + \psi_1(1,1,1)x_2x_3x_4 \right\}.
\end{aligned} \tag{31}$$

Expression (29) can again be written in the canonical form as

$$\max_{x_1} P(x|y) = C \exp \left\{ - \sum_{\Lambda \in B_{-1}} \beta_2^*(\Lambda) \prod_{i \in \Lambda} x_i \right\}, \tag{32}$$

where C is a constant and $\beta_2^*(\cdot)$ is the new interaction parameter value resulted after we maximize $P(x|y)$ with respect to x_1 . Alternatively, one may consider $\beta_2^*(\cdot)$ as an interaction parameter of $\Lambda \subseteq S \setminus \{1\}$ resulted from MAP estimation of $P(x|y)$ with respect to x_1 . See Figure 5 for the unweighted graphical representation of (32) for our toy example resulted after maximization with respect to x_1 . Note here that, after we maximize $P(x|y)$ with respect to x_1 we come up with (32). However, note that C is no longer be a normalizing constant just it is any constant.

We should note that some of $\beta_2^*(\cdot)$ may get a non-zero value for the first time which mean, their value was zero before we maximize $P(x|y)$ with respect to x_1 , in another word during the vertex-weighted graph of $P(x|y)$ those $\beta_2^*(\cdot)$ were not existing. This is true for $\beta_2^*(\{2,3\})$, $\beta_2^*(\{2,4\})$ and $\beta_2^*(\{2,3,4\})$ for our toy example. Thus after we maximize $P(x|y)$ with respect to x_1 , we come up with a new set of interaction parameters $\beta_2^*(\cdot)$ and with a new neighbourhood system. According to expression (32) we can possibly define a new neighbourhood notation i.e.

$$\mathcal{N}^2 = \{\mathcal{N}_j^2\}, \quad j \in S \setminus \{1\} = (2, \dots, n). \tag{33}$$

Note that $\mathcal{N}^1 = \mathcal{N} = \{\mathcal{N}_i\}$, $i \in S$, was the neighbourhood system before we maximize $P(x|y)$ with respect to x_1 . Implementing expression (32) in our toy example, it results in a new neighbourhood as defined in (33) i.e.

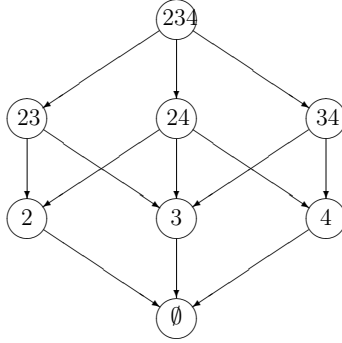


Figure 5: The unweighted graph $\mathcal{G}(S \setminus \{1\})$ of the toy example with $S = \{1, \dots, 4\}$.

$\mathcal{N}^2 = (\mathcal{N}_2^2, \mathcal{N}_3^2, \mathcal{N}_4^2)$ where $\mathcal{N}_2^2 = \{3, 4\}$, $\mathcal{N}_3^2 = \{2, 4\}$ and $\mathcal{N}_4^2 = \{2, 3\}$. In a similar way we continue for x_2 and compute

$$\psi_2(x_{\mathcal{N}_2^2}) = \arg \max_{x_2} \left[\max_{x_1} P(x|y) \right] \quad (34)$$

$$= \arg \max_{x_2} \left[P(x_1 = \psi_1(x_{N_1}), x_2, \dots, x_n | y) \right]. \quad (35)$$

After we compute $\psi_2(x_{\mathcal{N}_2^2})$ for each combination of x_2 's neighbouring value arranged in table format as we did for x_1 we get a new function i.e.

$$\max_{x_2} \left(\max_{x_1} P(x|y) \right) = P(x_1 = \psi_1(x_{N_1}), x_2 = \psi_2(x_{N_2^2}), x_3, \dots, x_n | y). \quad (36)$$

After we have maximized over x_1 and x_2 we continue to maximize over each of x_i , $i = 3, \dots, n$ in turn. Thus we get $\psi_i(x_{\mathcal{N}_i^i})$ for $i = 1, 2, \dots, n$. Note that ψ_n is not a function of any variable, it is a constant.

When the forward recursion (computation of $\psi_i(x_{\mathcal{N}_i^i})$, $i = 1, \dots, n$) is finished we can find the MAP estimate via a backward recursion (computation

4.1 Beta ($\beta(\Lambda)$) computation

of $\hat{x}_i, i = 1, \dots, n$). First we get $\hat{x}_n = \psi_n$. Next we insert \hat{x}_n for x_n in $\psi_{n-1}(x_{\mathcal{N}_{n-1}^{n-1}})$ and get $\hat{x}_{n-1} = \psi_{n-1}(x_{\mathcal{N}_{n-1}^{n-1}})$. Then we insert \hat{x}_n for x_n and \hat{x}_{n-1} for x_{n-1} in $\psi_{n-2}(x_{\mathcal{N}_{n-2}^{n-2}})$ to get \hat{x}_{n-2} . We continue in this way until we have all $\hat{x}_1, \dots, \hat{x}_n$. Putting these together we get

$$\hat{x}_{MAP} = (\hat{x}_1, \dots, \hat{x}_n). \quad (37)$$

Note that during the backward recursion for computing \hat{x}_j , we only insert \hat{x}_i in $\psi_j(x_{\mathcal{N}_j^j})$ for $j < i$ where $i \in \mathcal{N}_j^j$ since the non-neighbouring sites to j have no influence during the computation of \hat{x}_j .

4.1 Beta ($\beta(\Lambda)$) computation

Due to the Markovian property, we can write (23) as follows

$$P(x|y) = \text{const} \exp \left\{ - \left[\sum_{\Lambda \in B_{-1}} \beta_1^*(\Lambda) \prod_{j \in \Lambda} x_j + \sum_{\Lambda \in B_1} \beta_1^*(\Lambda) \prod_{j \in \Lambda} x_j \right] \right\}, \quad (38)$$

where $\beta_1^*(\Lambda) = \beta^*(\Lambda)$. Using the $\psi_1(x_{\mathcal{N}_1})$ resulted from (26) we could write (38) replacing x_1 by $\psi_1(x_{\mathcal{N}_1})$ as

$$\begin{aligned} \max_{x_1} P(x|y) &= P(x_1 = \psi_1(x_{\mathcal{N}_1}), x_2, \dots, x_n | y) \\ &= \text{const} \exp \left\{ \underbrace{- \sum_{\Lambda \in B_{-1}} \beta_1^*(\Lambda) \prod_{j \in \Lambda} x_j}_{\text{part I}} \right. \\ &\quad \left. + \underbrace{\left[- \sum_{\Lambda \in B_1} \beta_1^*(\Lambda) \prod_{j \in \Lambda \setminus \{1\}} x_j \psi_1(x_{\mathcal{N}_1}) \right]}_{\text{part II}} \right\}. \end{aligned} \quad (39)$$

We first focus on part II of this expression. As we have discussed before, $\psi_1(x_{\mathcal{N}_1})$ is a function of $x_{\mathcal{N}_1}$. Note that it can be written in an expanded form as follows

$$\psi_1(x_{\mathcal{N}_1}) = \sum_{\Lambda \in \mathcal{P}(\mathcal{N}_1)} \left\{ \psi_1(x_\Lambda = 0, x_{\mathcal{N}_1 \setminus \Lambda} = 1) \prod_{j \in \Lambda} (1 - x_j) \prod_{j \in \mathcal{N}_1 \setminus \Lambda} x_j \right\}. \quad (40)$$

After a bit manipulation we can rewrite (40) as

$$\psi_1(x_{\mathcal{N}_1}) = \sum_{\Lambda \in \mathcal{P}(\mathcal{N}_1)} \underbrace{\left[\sum_{z \in \{0,1\}^{|\Lambda|}} \left((-1)^{(\sum z) + |\Lambda|} \right) \psi_1(x_{\mathcal{N}_1 \setminus \Lambda} = 0, x_\Lambda = z) \right]}_{\text{part III}} \prod_{j \in \Lambda} x_j. \quad (41)$$

Note that in the above expression z is dependent only on $|\Lambda|$, not about what element Λ contains. For example if $|\Lambda| = 1$, always $z \in \{0,1\}$. Also if $|\Lambda| = 2$, $Z \in \{\{0,0\}, \{0,1\}, \{1,0\}, \{1,1\}\}$. Let us reconsider the toy example to elaborate these. From the toy example we know that $x_{\mathcal{N}_1} = (x_2, x_3, x_4)$. Thus one possibility of Λ is $\Lambda = \{2\}$. In this case $z \in \{0,1\}$ and also $\psi_1(x_{\mathcal{N}_1})$ is equal to $\psi_1(x_2 = 0, x_3 = 0, x_4 = 0)$ or $\psi_1(x_2 = 1, x_3 = 0, x_4 = 0)$ when $z = \{0\}$ and $z = \{1\}$ respectively. When $\Lambda = \{2,4\}$, $z_\Lambda \in \{\{0,0\}, \{1,0\}, \{0,1\}, \{1,1\}\}$ and then $\psi_1(x_{\mathcal{N}_1}) \in \{\{x_3 = 0, x_2 = 0, x_4 = 0\}, \{x_3 = 0, x_2 = 1, x_4 = 0\}, \{x_3 = 0, x_2 = 0, x_4 = 1\}, \{x_3 = 0, x_2 = 1, x_4 = 1\}\}$. Inserting (41) in part II of (39) and after some mathematical rearrangements of this we get

$$\text{part II} = \sum_{\Lambda \in \mathcal{P}(\mathcal{N}_1)} \left(\text{sum}(\Lambda) \sum_{\Lambda^* \in B_1} \left(\beta_1^*(\Lambda^*) \prod_{j \in (\Lambda \cup \Lambda^*) \setminus \{1\}} x_j \right) \right), \quad (42)$$

where

$$\text{sum}(\Lambda) = \sum_{z \in \{0,1\}^{|\Lambda|}} \left((-1)^{(\sum z) + |\Lambda|} \right) \psi_1(x_{\mathcal{N}_1 \setminus \Lambda} = 0, x_\Lambda = z). \quad (43)$$

During maximization of $P(x|y)$ with respect to x_1 , the interaction parameters of the neighbours of x_1 will be updated, but not the interaction parameters of the sites which were not neighbour to x_1 . To manage this mathematically let us split part I as with and without neighbour to x_1 i.e.

$$\text{part I} = \sum_{\Lambda \in B_{-1} \setminus \mathcal{P}(\mathcal{N}_1)} \beta_1^*(\Lambda) \prod_{j \in \Lambda} x_j + \sum_{\Lambda \in \mathcal{P}(\mathcal{N}_1)} \beta_1^*(\Lambda) \prod_{j \in \Lambda} x_j. \quad (44)$$

5 Approximate recursive computation

Remember that the combination of expression (42), which is part II, and expression (44), which is part I, gives expression (39). Therefore, we can collect similar variable coefficients from expressions (42) and (44) to get the result of $\max P(x|y)$ with respect to x_1 with the new energy function defined below.

$$\begin{aligned}
 U(x) = & \sum_{\Lambda \in B_{-1} \setminus \mathcal{P}(\mathcal{N}_1)} \beta_1^*(\Lambda) \prod_{j \in \Lambda} x_j + \sum_{\Lambda \in \mathcal{P}(\mathcal{N}_1)} \left[\beta_1^*(\Lambda) \right. \\
 & \left. + \sum_{\Lambda^* \in \mathcal{P}(\Lambda)} \left[\text{sum}(\Lambda^*) \sum_{A \in \mathcal{P}(\Lambda^*)} \beta_1^*(\Lambda \cup \{1\} \setminus A) \right] \right] \prod_{j \in \Lambda} x_k. \quad (45)
 \end{aligned}$$

We can again write (45) in a canonical form as

$$U(x) = \sum_{\Lambda \in B_{-1}} \beta_2^*(\Lambda) \prod_{k \in \Lambda} x_k, \quad (46)$$

where

$$\beta_2^*(\Lambda) = \begin{cases} \beta_1^*(\Lambda), & \text{if } \Lambda \in B_{-1} \setminus \mathcal{P}(\mathcal{N}_1) \\ \beta_1^*(\Lambda) + \sum_{\Lambda^* \in \mathcal{P}(\Lambda)} \text{sum}(\Lambda^*) \sum_{A \in \mathcal{P}(\Lambda^*)} \beta_1^*(\Lambda \cup \{1\} \setminus A) & \text{if } \Lambda \in \mathcal{P}(\mathcal{N}_1) \end{cases} \quad (47)$$

Thus, once we have computed $\max P(x|y)$ with respect to x_1 then next we will compute (36) with the neighbourhood definition given at (33) where $x = \{x_j\}$, $j \in S = \{2, \dots, n\}$.

In a similar way we continue computing the rest (with respect to x_3, \dots, x_n) recursively.

5 Approximate recursive computation

As we have seen in the previous sections, we must work a computation for every row of a $2^{|\mathcal{N}_i|} \times |\mathcal{N}_i|$, $i \in S$ table to get the exact recursive computational estimate of $P(x)$, $x = (x_i, \dots, x_n)$ with respect to x_i . We can easily see as the size of n and (or) \mathcal{N}_i increase, it requires too much computation time and computer memory to perform the computation for x . Due to this, the demand to have an approximate version of the MAP estimation method is essential. For this a strategy similar to the one adopted in Tjelmeland and

Austad (2010) can be used.

The approximation method is implemented in two steps by giving attention to (12). According to (12), we first compute $\beta(\Lambda)$ for $\Lambda \subseteq P(S)$ with $|\Lambda| = 1$ from the given energy function $U(x)$, then next for $\beta(\Lambda)$ with $|\Lambda| = 2$ and so on in increasing number of element in Λ . Again using $\beta(\Lambda)$ computed from (12) we get $\beta^*(\Lambda)$ using (22). The approximation is not going to be implemented at this stage yet i.e. not for $\beta(\Lambda)$ or not for $\beta^*(\Lambda)$ ($= \beta_1^*(\Lambda)$). However the approximation is done for $\beta_i^*(\Lambda)$, $2 \leq i \leq n$. The first step of approximation is to set all $\beta_2^*(\Lambda)$ to zero for those having value less than a given threshold value ε i.e. $\beta_2^*(\Lambda) = 0$ if $|\beta_2^*(\Lambda)| < \varepsilon$. The second step of approximation is to set $\beta_2^*(\Lambda)$ to zero if $\beta_2^*(A)$ was approximated to zero (was zero) $\forall A \subset \Lambda$ and $|A| = |\Lambda| - 1$. Once we done approximating $\beta_2^*(\Lambda)$, we get a set of interaction parameters which have interaction value great than Epselon ε . Using this new set of interaction parameters we apply all the previous section computations to get $\beta_3^*(\Lambda)$ and then, next the approximation continues for this $\beta_3^*(\cdot)$ interaction parameter groups. This computation goes till end $\beta_n^*(\cdot)$.

Note that, the first step of approximation reduces the computer storage problem whereas, the second step of approximation is the frequently used assumption in statistics that is higher order effects can be present only if corresponding lower order effects are present. Therefore, by implementing the combination of the two approximation steps we come up with an efficient (both in computer memory requirement and run time) recursive algorithm for MAP estimation method.

6 Concluding remarks

In this thesis, we have carried out a detailed recursive computation of MAP estimate of a MRF using the MAP-MRF framework. In addition, we are also able to write every recursively resulted MRFs in a canonical form. Thus the ability to write the MRFs in a canonical form make it easy to understand and compute for any size n which becomes just repeating the same technique over and over again. By applying MAP estimation method at a noise-corrupted image, we are capable of estimating the true image without much worry about the intractability of the normalizing constant exist in the

6 Concluding remarks

assumed MRF prior distribution. Due to large size of MRFs and (or) the quality (in run time and computer memory) of the algorithm, the approximation of the interaction parameters during the MAP estimation method plays a significant role.

Today it is common to consider images with more colours. Thus, it is desirable to generalize the procedure we discuss in this thesis to such a situation. Note that the skeleton of the algorithm will be unchanged, but the computational complexity of the algorithm grows rapidly with the number of colours in question. However, also in the k-colour case the computations repeat in a similar fashion for each site i .

Note that, the threshold value ϵ is not yet suggested since it need a practical situation which need to be fixed accordingly by considering the cost limit both in computer memory and run time of the algorithm. However, we still need a prior knowledge or a pilot study for finding an averagely acceptable range of ϵ for a similar image family and work the approximation by considering this Epsilon ϵ .

References

- [1] Allahverdyan, A. and Galstyan, A. (2009), "On maximum a posteriori estimation of hidden Markov processes", USC Information Sciences Institute Marina del Rey, CA 90292, USA.
- [2] Besag, J., (1974), "Spatial interaction and the statistical analysis of lattice systems", Journal of the Royal Statistical Society, series B, **36**, pp. 192–236.
- [3] Bouman, C.A. (1995), " Markov random fields and stochastic image models", IEEE International Conference on Image Processing. Washington, D.C.
- [4] Geman, S. and Geman, D. (1984), " Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images", IEEE Transaction on Image Ptocessing **6**, pp. 721–741.
- [5] Gull, S.F. (1989), " Developments in maximum-entropy data analysis", In J. Skilling, editor, Maximum Entropy and Bayesian Method, pp 53–71.
- [6] Hanson, K.M. (1993), "Introduction to Bayesian image analysis", In Medical Imaging: Image Processing, M.H.Loew, ed. Proc. SPIE 1898, pp. 716–731.
- [7] Hurn, M.A., Husby, O.K. and Rue, H., (2003), "A tutorial on image analysis", In Møller, J. Spatial Statistics and Computational Methods, Springer, pp 87–141.
- [8] Li, S.Z (1994), "A Markov random field model for object matching under contextual constraints", In Proc. CVPR, pp. 866–869, Seattle, Washington.
- [9] Li, S.Z (2009), "Markov Random Field Modeling in Image Analysis", 3rd edition, Springer-Verlag London Limited.
- [10] Tardón, L.J., Barbancho, I. and Márquez.F (2006)"A Markov Random Field Approach to Edge Detection", IEEE MELECON, Benalmádena (Málaga), Spain.

REFERENCES

- [11] Tjelmeland, H. and Austad, H.M (2010), "Exact and approximate recursive calculation for binary Markov random fields defined on graphs", technical report, Department of Mathematical Science, NTNU, Trondheim, Norway.