Cutset Based Processing and Compression of Markov Random Fields

by

Matthew G. Reyes

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Electrical Engineering: Systems)
in The University of Michigan
2011

Doctoral Committee:

Professor David L. Neuhoff, Chair
Professor Andreas R. Blass
Professor Alfred O. Hero III
Associate Professor Thrasos N. Pappas
Assistant Professor Sandeep P. Sadanandarao

To my grandparents, Mike Reyes and Frances Falcon; and Phillip Morales and Edna Moody

ACKNOWLEDGEMENTS

There are many people whose kindness, love, and support helped get me through the past five and a half years.

First off, I want to thank my family for always being there for me, to help when I needed it, and in general just being wonderful, interesting people and providing great examples.

There are two people that stand out the most as having contributed to my successful tenure at Michigan. The first of these is Jeannie Hernandez, my best friend during my time at Michigan. She showed a great deal of love and support to me, was a truly great friend. She also inspired me with her vast intelligence, work ethic, and overall passion for life.

The second person is my friend and advisor Dave Neuhoff. I took an image processing course from Dave my second semester at Michigan and was greatly impressed by his attention to detail, his willingness to help students during office hours, his general professionalism, and even his sense of humor. Based on my experience in the course I asked if I could do a research project with him, at the time having very little knowledge about what exactly he worked on. He proposed a number of possible topics, but the one that really jumped out at me involved applying these things called Markov random fields to image compression. I had learned about Markov chains in my stochastic processes course my first semester at Michigan and found them to be very interesting. It was a truly great experience to learn and discover new things about Markov fields with Dave the past four and a half years.

I want to thank the rest of my committee for their support, guidance and encouragement along the way. Thrasos Pappas was an inspiration for his great enthusiasm for research and his multidisciplinary approaches to image processing and compression. He is also a friend, a great host, and fun company. Interacting with him the past four years was definitely a pleasure. Al Hero recommended to me a Master's thesis on Markov random fields that was hugely influential in shaping my understanding of this important area. He was always eager to help with any questions I had and had stores of suggestions and references to help augment my understanding. I took a graph theory course from Andreas Blass my third semester at Michigan and found the class, and the professor, very enjoyable. I liked going to his office hours to discuss extra problems I had decided to work through and found that he was excited just to talk about possible proofs, even for problems not assigned for the course. Later, as I was beginning my work on Local Conditioning (at the time not knowing it had already been developed), I visited Prof. Blass and discussed this algorithm with him. He provided insight which proved pivotal in forming the necessary ideas. Sandeep Pradhan was always helpful and willing to answer questions that I might have. Also, it was in his network information theory course that I was able to do a project on distributed inference using methods of information geometry that helped contribute to my grasp of this area and shape my perspective on future work.

I would like to thank Rudy Urbanke for spending time discussing MRFs, the Ising model, and Belief Propagation with me at ISIT 2008, though I must admit at the time I was under the distinct impression that his name was Martin Wainwright. It was only later that I learned, upon visiting the website for "Modern Coding Theory", that the helpful individual in Toronto was in fact Prof. Urbanke. I would also like to thank the real Martin Wainwright for meeting with me at NIPS 2008 and spending time discussing MRFs and Belief Propagation. His insight was very helpful in cementing my thoughts on the topic. He has also been helpful in email correspondences since

then. I would like to thank Tsachy Weissman for a very helpful and informative discussion in the shuttle ride to the Salt Lake City airport from the Snowbird Lodge following DCC 2010. This fortuitous encounter has lead to continued discussion about possible collaboration as well as support and guidance that I am very thankful for.

There were many great friendships from which I benefited during my time in Ann Arbor. Awlok, your fearless zest for life was an inspiration. Arnau and Yongsheng, my running buddies, were great to talk to on our voyages around Ann Arbor and were always so helpful. Patrick, I enjoyed talking with you and getting your perspective on things, and looked up to you for reasons beyond your formidable height. Vicky, thanks so much for all your help along the way, especially during the home stretch when I had so many questions about the thesis and Latex. Ali, I really enjoyed getting to know you and thanks so much for all your help and encouragement during the past summer. Mathieu, it was great being your mentor and it has been all the more enjoyable becoming good friends. Debbie Mitchell, thanks for being a good mentor and role model, as well as for showing me such kindness. Yisa, it was great talking to you and becoming your friend. Ami, I thoroughly enjoyed talking with you about graphical models and getting your perspective on things. David, thanks so much for always helping out with the cats, I will miss grabbing lunch with you. And Ray, it was great talking with you at the pool, gaining insight into various strokes, of swimming and life in general. Sid, playing racquetball was great fun, hope you keep at it. Vanessa and Jane, thanks as well for always helping with the cats, being fun to talk to, and best of luck in the Chi. To anyone I may not mentioned, my deepest apologies. I put this section off till the end and I seem to have run out of time!

I owe many thanks to my great (in all senses of the word) uncle, Andrew Moody, for years of friendship, support, encouragement, and serving as a role model. Some of my fondest memories are of playing pool in Andy's basement and eating Di'Giorno's pizza.

Finally, I wish that my grandmother, Edna Moody Morales, was still here to see this day.

TABLE OF CONTENTS

LIST OF FIGURES

Figure

ABSTRACT

Cutset Based Processing and Compression of Markov Random Fields

by

Matthew G. Reyes

Chair: David L. Neuhoff

Markov random fields (MRFs) are an extremely and increasingly important class of probability distributions. MRFs are essentially probability distributions on the ways to color an undirected graph and are particularly suited for problems where the data in question is spatially distributed or arises through the interaction of a large number of individual units. The have been used to model natural phenomena, including the ferromagnetic interactions of iron (and other) atoms, dynamic processes on social networks, spatially distributed data such as images, and more recently, gene regulatory networks. In addition, they are being applied to man-made scenarios such as sensor and control networks in which a large number of interacting units combine to provide some global effect. They have been studied extensively from the points of view of statistical inference, modelling, parameter estimation and more. An important area in which they had not been thoroughly studied is data compression, the process of finding efficient representations of data to be stored on a computer, camera, cell phone or any other digital device. With the relevance of MRFs as great as it is, and seeming to grow by the year, and our world's ever-increasing reliance on digital media as a means of obtaining and communicating information, compression

of MRFs is an significant topic. This thesis develops a framework for the compression of Markov random fields. In particular, it presents a method for compressing MRFs based on a special type of subset within the graph associated with a given MRF. However, the analytical and algorithmic tools established have application to other potential means of compressing MRFs and indeed to the wide range of traditional research areas for MRFs. Though the compression methods and analysis we present in this thesis are applicable to general MRFs, we focus on images modelled as MRFs as image compression is our motivating concern.

In this thesis we present a number of results related to the problem of compressing a Markov random field (MRF) $\mathbf{X}$ defined on a graph $G = (V, E)$ by first losslessly compressing a cutset of sites U and then either losslessly compressing or estimating the remaining sites conditioned on the cutset values.

As an instance of the latter problem, we consider a square grid cutset consisting of evenly spaced rows and columns of the image for a homogeneous Ising model. We present analytical solutions to the MAP estimate of a block conditioned on the commonly occurring boundaries with two or fewer runs of black, for both 4 pt. and 8 pt. grid graphs. Using these results we empirically demonstrate that Max-Product Loopy Belief Propagation converges to the correct results. We present a simple adaptive Arithmetic Encoding (AC) based method for losslessly compressing a square grid cutset consisting of evenly spaced rows and columns of a binary image and, applying the Ising reconstruction results, show that the resulting lossy bilevel image coder is competitive compared to other such methods. Our results are significant in the context of the Ising model as it is the first MRF to be studied and the principal example of an MRF used in most contexts. The application of this algorithm to the lossy coding of bilevel images could represent a breakthrough in bilevel image compression as smooth contours are reconstructed from edge-preserving information from the original image. The grid cutset that is losslessly encoded preserves edges from the original

image and preserving this key structural information allows faithful reconstructions from a simple algorithm.

We present a rigorous development of Local Conditioning for MRFs, an algorithm for exact inference in cyclic graphs. Again, our theoretical contributions with regard to LC are significant. Because many interesting and difficult problems are modelled as MRFs (generally, on cyclic graphs), our developments here could potentially have wide effect. More specifically, for problems that must be solved in a truly distributed manner, for instance sensor networks, the more popular method of clustering nodes into supernodes will not work, so LC will be required to optimize performance of an inference or related algorithm.

We prove that the entropy of a family of MRFs is monotone increasing in the associated exponential parameters and that the exponential parameters for the moment-matching reduced MRF induced by U for a given subset of nodes are component-wise greater than the original exponential parameters within U. We also show that the divergence between an MRF induced by exponential parameter θ and another induced by θ' is monotone increasing in θ'. Furthermore, we prove that the divergence between the marginal distribution for $\mathbf{X}$ and a reduced MRF follows a Pythagorean decomposition, providing a reduced MRF analogue to a well-known result in information geometry.

We also present efficient algorithms for optimal AC based lossless compression of acyclic and EASY cyclic MRFs, and use these as the basis for a suboptimal lossless compression for HARD cyclic MRFs, called *Reduced Cutset Coding*. We show experimental results of RCC on homogeneous Ising models and verify that the performance is nearly optimal. Moreover, the empirical performance of RCC provides estimates of upper and lower bounds to the entropy. One of the very intriguing side benefits of our work on RCC is that it can be used to obtain estimates of important quantities of an MRF model. For example, the entroy upper and lower bounds mentioned earlier.

CHAPTER I

Introduction

1.1 Motivation

Digital media play an ever increasing role in our modern world. With the internet well into its second decade of popular use, one may be forgiven for taking for granted the speed and accessibility with which information - in the form of photos, documents, forms, etc., - pinballs between computers scattered over nearly the entire globe. Of course, it is the mathematical tools of data storage and transmission, studied and developed under the name *Information Theory*, that make this efficient process possible.

First, the methods of *Source Coding*, or *Data Compression*, allow for information to be stored efficiently, in computers, cell phones, cameras, and more. The storage is necessarily in binary digits, or *bits*, as this is the currency of modern computers. Efficiency is obtained when redundancy in the data is sufficiently removed so that, on average, the minimum number of bits is required to store the information. Once the data are compressed into a sequence of bits, referred to as a *bitstring*, these bits of information can be transmitted over a wireless network, uploaded or downloaded. Reliability of this transmission is within the purview of *Channel Coding*, which adds redundancy to the bitstring in order detect errors introduced through the possibly corrupted medium. The focus of this thesis is on data compression, in particular the

compression of sources of information that can be described as Markov random fields, as well as the algorithmic and analytical tools related to this problem that also find application in other problems. The process of representing the original data in binary form is called *encoding*, while the reverse process of transforming the binary digits back to the original, or an approximation thereof, is called *decoding*.

It can be argued that the perpetual gains being made in computer memory and processing power obviate the need for further sophistication of data compression and transmission systems. However, with oil's last drop around the corner, we should fully grasp the horizon in front of us: a world in which mass transportation, of all forms, is greatly diminished, at a time when global commerce, research, and general information-sharing is likely to be even more intertwined than at present. This means that our reliance on digital data and the accompanying concerns of storage and transmission are, if anything, to increase significantly in the future. And perhaps more importantly, the underlying mathematical questions will still be there, and as a civilized society how can we allow them to lie untouched?

Data compression, pursued in an information-theoretic sense, is necessarily probabilistic. If we look at a particular type of data, for instance images to be recorded by a personal camera, we clearly expect some images to occur more frequently than others. For example, we would expect images of faces, houses, and trees to be more likely than pictures of the moon or of an anthill. Therefore, if we want an efficient data compression system we need a probability model that roughly describes the relative likelihoods of different pieces of data. This is important because otherwise we might assign long bitstrings to images that are very likely to occur and short bitstrings to images that are not likely, which is of course the reverse of what we want. An important class of probability models that has gained increasing amounts of attention in varied fields over the last few decades is Markov random fields.

The goal of this thesis is to develop an outline or structure for how to approach

the compression of Markov random fields. While Markov random fields have been the focus of intense research and application, very little has been undertaken previously regarding their compression. Along the way we will encounter problems that have application in other areas and in fact are quite significant in their own right.

1.2 Problem Statement

The basic problem we are interested in is the following. A collection of N random variables $\mathbf{X} = X_V = X_1, \ldots, X_N$, indexed by a set V, are generated according to a Markov random field (MRF) distribution p. We observe a *realization* (also *configuration* or *image*) $\mathbf{x} = (x_1, \ldots, x_N)$, encode it into a bitstring of length $l(\mathbf{x})$, then decode the bitstring back to the original $\mathbf{x}$ or to an approximation $\hat{\mathbf{x}}$. If the bitstring is decoded to the original, the compression is *lossless*, whereas if we decode to an approximation, the compression is *lossy*.

For lossless compression, the performance criterion is the length of the average bitstring over all possible configurations or images. One would therefore like to minimize this average over all possible encoding/decoding schemes, and it is well-known that for any encoding/decoding scheme the average bitstring length is at least as big as the *entropy* of the source. If the average length is the entropy, this is referred to as *optimal compression*. Without significant loss of optimality, we can assume that *Arithmetic Encoding* (AC) is used, which involves ordering the random variables $X_1, X_2, \ldots, X_N$ into a one-dimensional sequence, referred to as a *scan*. To encode the sequence $X_1, \ldots, X_N$ one computes a *coding distribution* for each X_i in the sequence, and in so doing effectively computes a coding distribution for the entire sequence. The closeness of the computed coding distribution to the true distribution for $X_1, \ldots, X_N$ is called the *divergence* between the two distributions, and gives the difference between the average bitstring length and the source's entropy. Therefore, optimal compression requires *exact inference*. Often we speak of the *rate*, which is the average bitstring

length normalized by the number of variables.

For lossy compression, there are two performance criteria. Since we are not requiring a perfect reconstruction of the original, there is great flexibility in the average number of bits that can be encoded. However, for a given average number of encoded bits we want to measure the *distortion* or error between the original and the output. Thus in lossy compression we keep track of both the average number of bits per pixel stored and the distortion between the original and reconstructed images. In this thesis we narrow this a bit by losslessly encoding a subset of the variables X_U and then estimating the unencoded variables $X_{V\setminus U}$ conditioned on the values of those that were encoded. In this sense we are approaching lossy compression in two stages, the first being a lossless compression problem, the second a sampling and reconstruction problem. From an operational point of view the additional goal in our approach to lossy compression is to find a subset together with an efficient estimation algorithm such that the resulting reconstructions have small distortion, however we choose to measure that. If the image were bandlimited, for instance, then Shannon's sampling theory tells us that what subgrids provide for optimal reconstruction [47]. In other words, the subset should depend on the type of signal being observed.

The Markov random field distribution for $\mathbf{X}$ means that we can view p as a distribution on the ways to color the nodes of an associated undirected graph $G = (V, E)$, where V is the set of nodes and E the set of edges connecting nodes in V. The structure of G has implications for our ability to perform exact inference for $\mathbf{X}$, and hence to optimally compress $\mathbf{X}$. We will see that for a general MRF, lossless compression will also need to be performed in two stages, the first where a subset X_U is suboptimally encoded, the second where the remaining variables $X_{V\setminus U}$ are optimally encoded given the first. Similarly, the structure or topology of G has much to do with our ability to perform efficient and accurate estimation of unsampled variables conditioned on the sampled variables. More specifically, the topology of G dictates

what subsets of initially encoded, i.e. sampled, variables will permit efficient and accurate estimation of the unsampled variables. As we have in mind image coding, the node set V is a rectangular array of sites and the edges in E connect either horizontally and vertically adjacent nodes, or these plus diagonally adjacent nodes. In the former case the graph G is called a 4 pt. grid graph and in the latter G is an 8 pt. grid graph.

In addition to the graphical structure, an MRF is specified by attaching *weights* to the edges and nodes within the graph, where larger weights indicate stronger correlations. In both the lossless and lossy compression problems we will perform an initial encoding of a subset of the variables. This compression will be suboptimal since exact inference will be intractable. Therefore, an approximate probability distribution is computed for the initial subset, and we would like to quantify the divergence between the approximate and true probability distributions for X_U. This will allow us to optimize the approximate distribution over some restricted class of approximate distributions by analyzing the divergence between the true distribution and members of this class. Furthermore, since we will be modeling X_U using another distribution, we would like to compare the entropy of X_U and the entropy of the approximate distribution. Because the graphical structure of G is what determines whether exact inference is possible, the type of approximations pursued will be those based on *thinning* the original graph G by removing edges. Using an approximate distribution obtained by thinning or removing edges is common for MRFs and other probability distributions on graphs [33, 55, 31].

Once a tractable, thinned model is chosen for the approximate distribution of X_U, optimal inference can be performed on this model. Furthermore, once the subset X_U is encoded, exact inference is required on $X_{V\setminus U}$, in the lossless compression case, and efficient estimation of $X_{V\setminus U}$ is needed in the lossy case. One can therefore view this thesis as consisting of a few main problems. First, how to optimally compress an

acyclic MRF and then extend this to optimally compress a tractable cyclic MRF. Second, how to compare the distribution for a subset of random variables X_U with an approximate distribution chosen to be a tractable MRF. And third, how to estimate a subset of sites conditioned on X_U.

1.3 Why Care About Markov?

Before getting to the contributions of this thesis, it is natural to first ask of the value in looking at the compression of Markov random fields. A quick survey of the literature [21, 22, 8, 15, 2, 53, 56, 11, 54, 31] will show that many groups throughout the country and indeed the world are looking into practical ways to model various information sources as MRFs.

If p is *positive*, in that $p(x) > 0$ for all x, then we can parameterize p so that it can be expressed as a product of functions defined over subsets of the random variables. This defines a graph $G = (V, E)$ with respect to which p is said to be *Markov* in that the values of two nodes not connected by an edge are independent of each other conditioned on the values of all other nodes. Therefore, at least to the extent that one believes in the positivity assumption, one must also believe in the Markov assumption. There are of course nonpositive Markov distributions. Alternatively, one can start with a graph and then define a distribution as a product of functions on the graph. This is the approach adopted in this thesis.

A motivating reason for considering Markov models for real world data is the observation or assumption that for information that is spatially distributed, if we need to know (estimate) what happened in a particular region of the image, it is not necessary to know what occurred in the entire surrounding region. Instead, it is sufficient, i.e., just as good, to know the information in some smaller surrounding region. Implicit here is the notion of a *boundary* for a set of pixels, conditioned upon which the set of pixels is completely independent of all other data, which is

an alternative formulation of the definition of Markov. In Chapter III we apply this belief in a lossy cutset based coding method for a class of Ising models. The type of Ising model considered there is in fact the oldest known MRF and has been suggested many times as an important model for information theory.

At this point it is important to draw a distinction between studying the compression of sources that follow a Markov random field distribution versus studying Markov random fields or MRF models as a means of compressing, say, images. The former is a purely mathematical investigation, as an MRF is a mathematical object, while the latter is a practical pursuit, seeking to exploit certain properties of MRFs or algorithms developed for such for the efficient compression of images, for example. It is a subtle point, but what is important to emphasize is that in order to seriously undertake the second task, that of compressing images using MRF models, there must already be in place a solid foundation for how to compress Markov random fields, as the mathematical objects that they are. Then, one can experiment with different MRF models, in each case applying the general and therefore applicable rules on how to compress such a source. As such, the principal concern of this thesis is the development of tools, analysis, and direct methods for compressing MRFs.

1.4 Thesis Overview and Contributions

In a sense, the primary contribution of this thesis is presenting a coherent framework for the compression of Markov random fields, and along the way, many algorithmic and analytic tools that promise to be useful in other potential MRF compression schemes as well as myriad other MRF research areas. Moreover, the theory in this thesis is demonstrated on the Ising model, the first and still best known example of an MRF, which places the ideas presented in this thesis on a firm footing. In the following we detail the specific contributions of each chapter.

Chapter II: Background Information. In this chapter we introduce the required graph-theoretic notation; discuss MRFs as probability distributions with respect to graphs, the representation of MRFs as members of exponential families, and Belief Propagation (BP) as a tool for performing inference within MRFs; and introduce the necessary ideas from information theory and source coding.

Chapter III: Lossy Cutset Coding of Ising Model. In this chapter we introduce the idea of lossy Cutset Coding in the context of a homogeneous Ising model. This approach entails first losslessly encoding a square grid cutset consisting of evenly spaced rows and columns of the image and then estimating the unencoded interiors conditioned on the cutset. Though the Ising model has been the object of intense research [7, 40, 32, 19, 48] since being introduced [29], including bounds on its lossy compression [36], this is the first (practical) method presented for its lossy compression. In fact, this is the first practical lossy coding method presented for any MRF, though there have been existential results for certain classes of algorithms [60, 34]. MAP decoding is one estimation criterion which reduces to MAP estimating a block conditioned on its boundary. In Section 3.2 we give analytical solutions for MAP reconstructions of a block when the boundary has four or fewer black-white transitions, which are the most frequently occurring cases. These are done for both the 4 pt. and 8 pt. grid graphs and are applicable for a general sampling and reconstruction problem. Using these analytical results as a point of comparison, we find empirically in 3.3 that the Max-Product version of Loopy Belief Propagation converges to the correct values in this sampling and reconstruction setting. This is significant as the performance of this algorithm is generally not well-understood [37]. In Section 3.4 we give reconstruction results for both simulated and real-world images and in Section 3.5 we introduce a modification to the MAP decoding algorithm that enhances the perceptual quality of the reconstructions. We compare this modified lossy compression algorithm to competing algorithms and find that it outperforms

them. In the final section of the chapter we present a practical method to do the lossless compression component of the overall lossy algorithm, and we make an important connection between Cutset Coding and the recently introduced concept of erasure entropy [51, 52].

Chapter IV: Local Conditioning for Markov Random Fields. Local Conditioning (LC) is an algorithm for exact inference in graphs with cycles that greatly reduces the complexity of performing exact inference in graphs with cycles. In this chapter we develop the Local Conditioning algorithm for Markov random fields on undirected graphs. LC has been developed for Bayesian networks [38, 39, 49, 6, 46, 13, 14], which are essentially MRFs on directed graphs. The undirectedness of the edges in MRFs makes analysis more accessible, and therefore allows us to put LC for MRFs on a firm theoretical foundation, which could potentially be extended into the Bayesian network case. In Section 4.3 we give message recursion and belief computation formulas for LC that are analogous to those for traditional BP. In 4.4 we look at examples of different unwrappings based on a given loop cutset for a 4 pt. grid graph and analyze the complexity of LC.

Chapter V: Monotonicity and Reduced MRFs. The lossy and lossless compression methods of Chapter III and Section 6.2, respectively, begin by encoding a subset of pixels. To simplify the encoding we simplify the graph by removing edges connecting pixels in the subset to pixels outside the subset. Cutting an edge, of course, is equivalent to setting the associated edge weight to zero. We are therefore interested in the effect that this cutting has on the entropy of the subset and in the divergence between the marginal distribution of the subset and the new distribution after removing edges, which we call a *reduced MRF* distribution. We prove that for any subset of nodes, the marginal entropy of that subset is upper bounded by the entropy of the MRF defined on the subgraph induced by the subset. As a step in this proof, we prove that moment-matching for the MRF on the induced subgraph

corresponds to a monotone increase in the exponential coordinates associated with nodes and edges within the subset. We also show that the divergence between the marginal distribution of the subset and the reduced MRF follows a Pythagorean decomposition analogous to a well-known result comparing an original MRF with a thinned model on the original set of nodes [3]. This decomposition makes it possible to optimize the coding distribution over all possible reduced MRFs on the induced subgraph.

It is intuitive that, for MRFs where the random variables are coupled in a way to favor similar values, the entropy should decrease if the weight on an edge is increased. Despite its intuitiveness, this is proved for the first time in this chapter and examples are provided illustrating this concept. Besides being significant in its own right, this result is used in proving the above mentioned inequality regarding the marginal entropy of a subset and the reduced MRF distribution obtained by dropping edges. This is because it is well-known that the moment-matching reduced MRF has higher entropy than the marginal distribution [9] and from our earlier result that moment-matching requires a monotone increase in the edge weights. Moreover, we prove a similar monotonicity result for the divergence between MRFs, as this is relevant for the common scenarios where a thinned model is used as an approximation to an intractable original. We also state a conjecture about the asymmetry of divergence in MRFs and give a simple example supporting it.

Chapter VI: Lossless Compression of MRFs. In this chapter we develop Arithmetic Encoding based lossless coding techniques for Markov random fields. We give the formulas for the optimal coding distributions for an acyclic MRF. We also give an algorithm for optimal lossless compression of an acyclic MRF, which arises through interpreting the optimal coding distribution formula. We present two algorithms for optimal compression of a cyclic MRF defined on a tractable graph, one based on a well-known clustering algorithm [53] for exact inference in cyclic MRFs, the other

based on the Local Conditioning algorithm presented in Chapter IV. The clustering based optimal compression algorithm uses the same coding distribution formulas as the acyclic MRF case, whereas a new optimal coding distribution formula is presented for the Local Conditioning based algorithm. For cyclic MRFs that are not defined on tractable graphs, we present Reduced Cutset Coding (RCC), a suboptimal method based on first encoding a cutset of sites. We demonstrate that this algorithm is nearly optimal for the homogeneous Ising model on the 4 pt. and 8 pt. grid graphs.

CHAPTER II

Background

2.1 Markov Random Fields

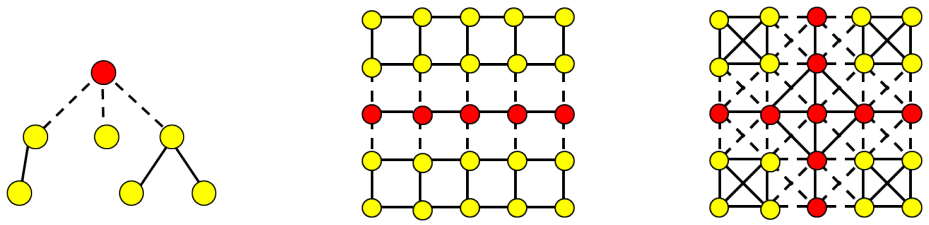
In this section we introduce Markov random fields (MRFs), which again, can broadly be described as probability distributions on the colorings of a graph.

2.1.1 Graphs

A graph G is defined by a pair of sets V and E and is expressed as $G = (V, E)$, where V is a set of *nodes* (also called *pixels* or *sites*), and E is a set of unordered pairs of nodes, referred to as *edges*. We assume throughout that the set V is finite. A graph is depicted by considering the elements of V to be points where two points are connected by an undirected line if the points are an edge in E. If $\{i, j\}$ is an edge in E, nodes i and j are said to be *adjacent*. A *path* in a graph is a sequence of nodes, each successive pair joined by an edge in E. Alternatively, a path can be described as a sequence of edges. A graph is said to be *connected* if every pair of nodes $i, j \in V$ can be joined by some path, and *disconnected* otherwise. A *component* of a disconnected graph G is a maximal connected subset $C \subset V$. A *cycle* in a graph is a path where the first and last nodes coincide. A graph without any cycles is called *acyclic*. A connected acyclic graph is called a *tree* while one that is disconnected is referred to as a *forest*. In working with trees one often designates a particular node

chosen as the *root* of the tree. Non root nodes that have only one neighbor are called *leaves* of the tree. This defines what are called *parent-child* relations throughout the tree as follows. For node i and neighbor j, if j lies on the path connecting i to the root, then j is said to be the *parent* of i and is denoted by $\pi(i)$. If j lies on a path connecting i to a leaf node that does not go through the root, then j is said to be a *child* of i and belongs to the set $\sigma(i)$.

For any $U \subset V$, its *boundary* ∂U is the set of nodes not in U connected by an edge to a member of U. As a shorthand, ∂i denotes $\partial\{i\}$, $i \in V$. The *surface* of a set U consists of those nodes in U that are connected by an edge to some element of ∂U and will be denoted by $\gamma(U)$. For a subset U, we let $E_U \subset E$ be the subset of edges both of whose endpoints are contained in U. Then, the graph $G_U = (U, E_U)$ is the *subgraph induced by* U. For subset U, the *complement* of U is the set $V \setminus U$. The graph $G \setminus U$ is obtained by removing U and all edges incident to it from G. In other words, $G \setminus U$ is the subgraph $G_{V \setminus U}$ induced by the complement of U. If G is connected, then a subset $U \subset V$ is called a *cutset* if $G \setminus U$ is disconnected. For a general G, U is a cutset if $G \setminus U$ contains more components than G. In this case $G \setminus U$, like any disconnected graph, is simply the collection of the (disjoint) subgraphs $\{G_{C_i}\}$ induced by the respective components. For connected G, if $U \subset V$ is a cutset and subsets $C_1, C_2 \subset V$ are contained in distinct components of $G \setminus U$, then U is said to *separate* C_1 and C_2.



(a) (b) (c)

Figure 2.1: (a) Acyclic graph; (b) 4pt. grid graph; (c) 8pt. grid graph. Sample cutsets are indicated in red.

The graphs of particular significance in this thesis are *grid graphs*, graphs in which the node set can be described as a rectangular array of sites, for instance the pixel locations of an image. Of these, we will focus on the *4-pt.* grid graph, whose edge set consists of horizontal and vertical nearest neighbors, and the *8-pt.* grid graph, the edges in which are horizontal and vertical, as well as diagonal nearest neighbors. Figure 2.1 shows (a) an acyclic graph, (b) a 4-pt. grid graph, and (c) an 8-pt. grid graph.

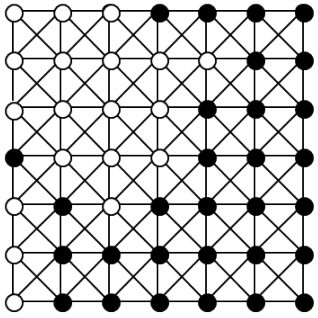
A *clique* of a graph G is a subset of nodes K such that if i and j are in K, then i and j are connected by an edge, and is also called a *complete subgraph*. In the 4-pt. grid graph the cliques are individual nodes and edges, whereas in the 8-pt. grid graph, the cliques are nodes and edges as well as triples and quadruples of neighboring nodes.

2.1.2 Images on Graphs

It is common practice to model images as a random process $\mathbf{X}$ assuming values on the sites or nodes of an undirected graph. Figure 2.2 (a) shows a smooth binary image and (b) illustrates a binary image on a graph. In this thesis, white and black nodes will denote binary values assigned to the associated pixels.

For each node $i \in V$, an associated random variable X_i can assume values in alphabet $\mathcal{X}_i$, a particular value being denoted by x_i. In this thesis we only consider MRFs where the alphabets at each node are finite in size. For an entire image assuming values on V, the alphabet is denoted $\mathcal{X}$, an image $\mathbf{x}$ and the random process on the entire site set $\mathbf{X}$. For a subset $U \subset V$ of sites, the random subfield X_U assumes values in alphabet $\mathcal{X}_U$, where a configuration on U is denoted as x_U. We will assume throughout that each node i has a common alphabet $\mathcal{X}_i = \mathcal{X}$. For an *image* or *configuration* $\mathbf{x}$ the probability of $\mathbf{x}$ is denoted generically as $p(\mathbf{x})$. When we wish to emphasize the underlying graph G, the probability of configuration $\mathbf{x}$ on G is

(a)



(b)

Figure 2.2: (a) Smooth binary image; (b) Representation of a binary image on 8pt. grid graph.

denoted $p_G(\mathbf{x})$, the probability of a configuration on a subset U of pixels $p_G^U(x_U)$, and the probability of a configuration on pixel i as $p_G^i(x_i)$.

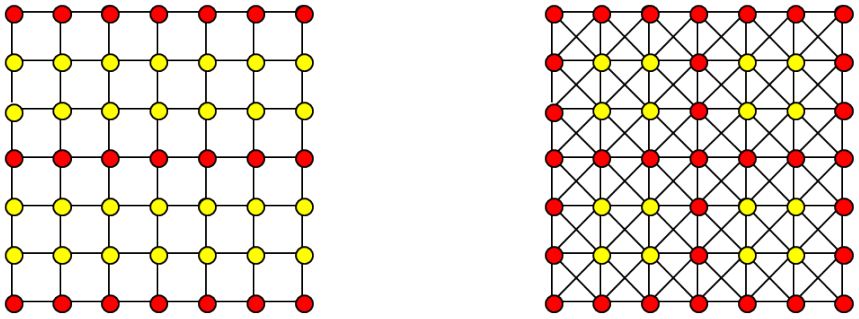
2.1.3 Conditional Independence

A random field $\mathbf{X}$ is said to be *Markov* with respect to G if for any two subsets $C_1, C_2 \subset V$ of nodes separated by a third subset U, the random subfields X_{C_1} and X_{C_2} are conditionally independent of each other given the values of X_U [35]. In other words, conditioning on the values on a cutset U leaves the components of $G \setminus U$ conditionally independent. In this thesis we will look primarily at cutsets of grid graphs consisting either of evenly spaced rows of the graph, or of evenly spaced rows and columns of the graph. Figure 2.3 illustrates these respective cases.

Depending on whether the underlying graph for an MRF is cyclic or acyclic, we say that we have a *cyclic* or *acyclic* MRF, respectively.

2.1.4 Product Representation

The probability distribution of an MRF $\mathbf{X}$ can be expressed as a product of functions defined on cliques of the graph [25]. In this thesis we focus on *pairwise* MRFs in which functions are defined over nodes and edges of the graph. This is



(a) (b)

Figure 2.3: (a) Cutset of lines in 4pt. grid graph; (b) Grid cutset in 8pt. grid graph.

useful both for analytical reasons and also for the Belief Propagation algorithms introduced in Section 2.4.1 For each edge $\{i,j\} \in E$ there is a nonnegative function $\Psi_{i,j} : \mathcal{X} \times \mathcal{X} \longrightarrow \mathbb{R}_+$ called an *edge potential*; and for each node $i \in V$, there is a nonnegative function $\Phi_i : \mathcal{X} \longrightarrow \mathbb{R}_+$ called a *self-potential* [1]. Then for any $\mathbf{x} = \{x_i : i \in V\}$, the probability that $\mathbf{X}$ equals $\mathbf{x}$ is given by the *Gibbs distribution*

$$p_G(\mathbf{x}) \;=\; \frac{1}{Q} \prod_{\{i,j\} \in E} \Psi_{i,j}(x_i, x_j) \prod_{i \in V} \Phi_i(x_i), \tag{2.1}$$

where Q is a normalizing constant called the *partition function*. For subset $A \subset V$ and an assignment of values x_A to nodes in A,

$$Z_A(x_A) \triangleq \sum_{x'_{V \setminus A}} \prod_{\{i,j\} \in E} \Psi_{i,j}(x_i, x_j) \prod_{i \in V} \Phi_i(x_i) \tag{2.2}$$

is the partial sum over all variables not in the set A, with nodes in A fixed to configuration x_A. The partition function can then be expressed as $Q = \sum_{x_A} Z_A(x_A)$. Because the probability that X_A equals x_A is proportional to $Z_A(x_A)$, the vector $Z_A = (Z_A(x_A) : x_A \in \mathcal{X}_A)$ is called the *belief* for A. As shorthand, we denote $Z_{\{i\}}$ by Z_i. One can verify the Markov property using the product representation in (2.1). In doing so, one can see that for any subset $C \subset V$ of nodes, the conditional probability

[1] Actually, it is more common to call $-\log \Psi$ and $-\log \Phi$ *potentials*.

$p(X_C|X_{\partial C})$ given the configuration on the boundary is itself an MRF distribution on the induced subgraph G_C, the only modifications to the original potentials in C being updates to the self-potentials on those nodes on the surface of C, which take into account the boundary values $x_{\partial C}$.

2.1.5 Exponential Family Representation

It is often convenient to make use of the so-called exponential representation of a family of MRFs. For instance, consider scaling each of the potentials in (2.1) by an exponent that accounts for the relative importance of Ψ_{ij} or Φ_i, as in

$$p_G(\mathbf{x};\theta) = \frac{1}{Q(\theta)} \prod_{\{i,j\}\in E} \Psi_{ij}^{\theta_{ij}} \prod_{i\in V} \Phi_i^{\theta_i}.$$

We can the rewrite the probability as

$$\begin{aligned} p_G(\mathbf{x};\theta) &= \exp\{ \sum_{\{i,j\}\in E} \theta_{ij} \ln \Psi_{ij} + \sum_{i\in V} \theta_i \ln \Phi_i - \ln Q(\theta)\} \\ &= \exp\{ \sum_{\{i,j\}\in E} \theta_{ij} t_{ij} + \sum_{i\in V} \theta_i t_i - \Phi(\theta)\} \end{aligned}$$

where we have introduced the statistics $\{t_i = \ln \Phi_i\}$ and $\{t_{ij} = \ln \Psi_{ij}\}$ and where $\Phi(\theta) = \ln Q(\theta)$ is the *log-partition function*. As before, the subscript G on p indicates the graph on which the MRF is defined. We let t stand for the collection of both the $\{t_i\}$ and $\{t_{ij}\}$.

The entire family of MRFs on G *based on* t is generated through the exponential parameter $\theta = (\theta_i, i \in V; \theta_{ij}, \{i,j\} \in E)$ where for each node i, and neighbor $j \in \partial i$, θ_i and θ_{ij} scale the sensitivity of the distribution $p_G(\mathbf{x}) = p_G(\mathbf{x};\theta)$ to the functions t_i and t_{ij}, respectively. We will suppress dependence of the MRF on t. The probability distribution $p_G(\theta)$ of an MRF X based on t with exponential parameter θ can be

expressed compactly as

$$p_G(\mathbf{x};\theta) \;=\; \frac{1}{Q(\theta)}\exp\{\langle\theta, t(\mathbf{x})\rangle\}, \tag{2.3}$$

where $\langle\ ,\ \rangle$ denotes inner product, $Q(\theta)$ is the partition function. The set $\Theta(G) = \{\theta \in \mathbb{R}_+^{|V|+|E|}\}$ is the set of *admissible exponential parameters* for MRFs on G, while $\mathcal{F}_t(G) = \{p_G(\theta) \mid \theta \in \Theta(G)\}$ is the family of all MRFs on G based on t. The set $\Theta = \Theta(G)$ is a coordinate system for MRFs in $\mathcal{F} = \mathcal{F}_t(G)$. In this sense, $\mathcal{F}$ is statistical manifold, where probability distributions in $\mathcal{F}$ are indexed by elements θ of the coordinate system Θ [3].

For a subset of nodes $U \subset V$ we let t_U and θ_U be the components of θ and t, respectively, corresponding to nodes and edges in U. We will sometimes partition an exponential parameter θ in the form $\theta = (\theta_U, \backslash\theta_U)$, where $\backslash\theta_U$ is the complement of θ_U in θ.

2.1.6 Gibbs Sampling

Given an MRF model for images on a particular graph, it is often desired to *simulate* an instance from that probability distribution, one on which empirical averages are very close to statistical averages for the MRF model. Such an image is referred to as a *typical* image, and the standard algorithm for generating typical images from an MRF or Gibbs distribution is the *Gibbs Sampler* [21].

The most basic version of the Gibbs Sampler operates by generating a sequence of images $\{\mathbf{x}(t)\}$ by changing or updating the value at a single site per iteration. That is, images $\mathbf{x}(t)$ and $\mathbf{x}(t-1)$ differ at a single site. The initial image $\mathbf{x}(1)$ is usually chosen arbitrarily. The site $i(t)$ to be updated at time t is usually chosen at random. Then one looks at the configuration $x_{\partial i}(t-1)$ on the neighborhood of $i(t)$, and draws a random sample x' from the conditional distribution $p_G^i(\cdot|x_{\partial i}(t-1))$. The

image $\mathbf{x}(t)$ is generated by replacing $x_{i(t-1)}$ with x' and leaving all other values the same. Research into the convergence properties of Gibbs Sampling continues, but it is generally understood that a few hundred "visitations" per site is sufficient to generate a good sample [61] for typical (i.e., 256×256 or 512×512) grid graphs.

2.1.7 Ising Model

The first Markov random field (MRF) to be studied, and the principal example throughout this thesis, is the Ising model. The Ising model was first studied in statistical physics to model magnetism, the value at a given node is historically referred to as a *spin* and can take values in $\{-1, 1\}$. Give our application of interest, we will typically refer to the values as colors, black and white being the alternatives, with black connoting -1 and white $+1$. The probability $p_G(\mathbf{x}; \theta)$ of an image under an Ising model is

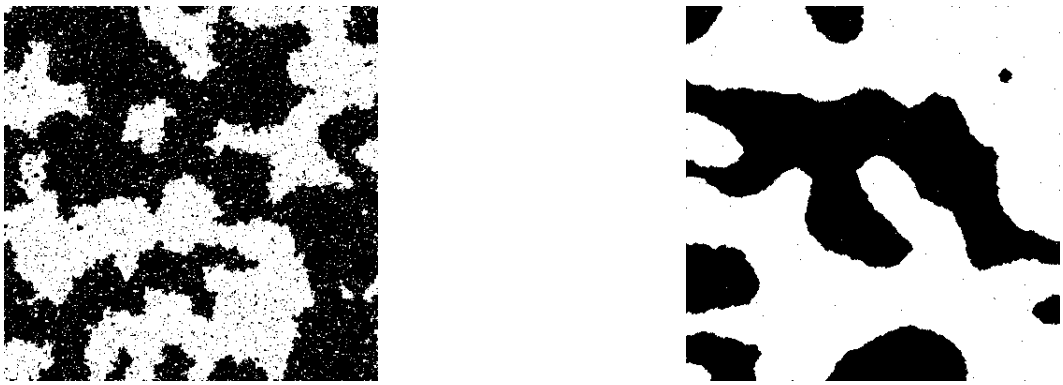
$$p_G(\mathbf{x}; \theta) = \frac{1}{Q(\theta)} \exp\{ \sum_{\{i,j\} \in E} \theta_{ij} x_i x_j + \sum_{i \in V} \theta_i x_i \} \tag{2.4}$$

Historically, the Ising model was originally associated with the 4 pt. grid graph, but has come to mean any binary MRF with the form given in (2.4). The coordinates $\{\theta_{ij}\}$ are generally thought of as coupling parameters whereas the coordinates $\{\theta_i\}$ are interpreted as part of an external field biasing the values of the individual nodes. In this thesis we consider the case that $\theta_{ij} = \theta$ for all edges $\{i, j\} \in E$ and some θ. Also, we assume that $\theta_i = 0$ for all $i \in V$. This is referred to as the case of no external field and uniform coupling, in which case (2.4) reduces to

$$p_G(\mathbf{x}; \theta) = \frac{1}{Q'(\theta)} \exp\{-2\theta O(\mathbf{x})\}, \tag{2.5}$$

where $O(\mathbf{x})$ is the number of edges in the image which have different colors on the endpoints. Such edges are referred to as *black-white transitions*, or more commonly,

as *odd bonds*. Note that $Q'()$ in (2.5) is in general not equal to $Q()$ in (2.4). Figures (2.4) (a) and (b) show sample images of Ising models generated with Gibbs Sampling with $\theta = 0.5$ on, respectively, a 400×400 4pt. and 8pt. grid graph.



(a) (b)

Figure 2.4: (a) Sample of uniform, no external field Ising model on 400×400 4pt. grid graph; (b) sample of Ising model on 400×400 8pt. grid graph.

The Ising model has been studied for quite some time [29, 7, 40, 32] and is the model that motivated the concept of a Markov random field and its subsequent development [17, 18, 25, 19, 48, 41]. In 1925, Ernst Ising introduced the model that would come to bear his name, as a model for the ferromagnetic interactions of iron atoms [29]. It was known that for temperatures beneath a certain so-called critical temperature, the magnetic spins of the iron atoms would spontaneously align, indicating long-range dependency between the spin values. This phenomenon is referred to as *spontaneous magnetization* or more generally as a *phase transition*. In particular, Ising was curious whether the parameter θ of the homogeneous model could predict the temperature-dependent phase transition known to occur in reality. He was disappointed to find that in one dimension no such phase transition existed, and erroneously concluded that the same was true in two dimensions as well. Peierls later showed that on the 4-pt. grid graph such a phase transition does occur [40].

In this thesis we consider homogeneous Ising models as models for binary images because typical images from this distribution (for certain ranges of values of the

parameter θ) are "blobby" with well-defined contours separating regions of white and black.

2.1.8 Dual Parameterizations

For a given exponential coordinate vector θ, we let $\mu = \mu(\theta)$ denote the vector expected value of the statistic t under the MRF induced by θ, and we refer to μ as the *moment* parameter of the MRF. The set of moment parameters arising from exponential parameters in $\Theta(G)$ is $\mathcal{M}_t(G)$, which is referred to as the set of *achievable moment parameters* for MRFs on G based on t. If the components of t are affinely independent, t is called *minimal*, and the mapping between Θ and $\mathcal{M} = \mathcal{M}_t(G)$ is one-to-one, and therefore, $\mathcal{M}$ provides a second coordinate system for MRFs in $\mathcal{F}$ [3]. We can then index an MRF p by either the associated exponential parameter θ or the corresponding moment parameter μ, where the moment parameter can similarly be expressed in partitioned form as $\mu = (\mu_U, \backslash\mu_U)$. Figure 2.5 illustrates the correspondence between an MRF p and the associated exponential parameter θ and moment parameter μ. Moreover, we can index an MRF p in mixed notation, for example, as $p \sim (\mu_U; \backslash\theta_U)$, in the sense that this too is a one-to-one correspondence [3].

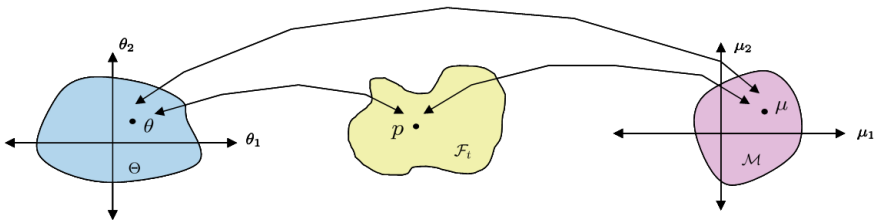


Figure 2.5: Mappings between manifold $\mathcal{F}_t(G)$ and sets of exponential parameters $\Theta(G)$ and moment parameters $\mathcal{M}_t(G)$.

2.1.9 Reduced Markov Random Fields

For a subset of nodes $U \subset V$, the marginal distribution of X_U for the original MRF on G under exponential parameter $\theta \in \Theta(G)$ is

$$p_G^U(x_U;\theta) = \sum_{x_{V\setminus U}} p_G(x_U, x_{V\setminus U};\theta),$$

where the superscript U denotes the subset of nodes whose marginal distribution is being considered. We can also consider MRF distributions for X_U. We say that an MRF on the subgraph G_U based on t_U is a *reduced MRF*, where as before t_U denotes the restriction of t to coordinates corresponding to nodes and edges in U. Specified in exponential coordinates $\theta_U \in \Theta(G_U)$, the reduced MRF distribution is denoted $p_{G_U}(X_U;\theta_U)$ and has the form

$$p_{G_U}(x_U;\theta_U) = \sum_{x_{V\setminus U}} p_G(x_U, x_{V\setminus U};\theta) \exp\{\langle \theta_U, t_U\rangle - \Phi_U(\theta_U)\} \tag{2.6}$$

If t is minimal for the family of MRFs on G, the subvector t_U is minimal for the family of MRFs $\mathcal{F}(G_U)$ on the induced subgraph G_U, so $\Theta(G_U)$ and $\mathcal{M}(G_U)$ are dual coordinate systems for MRFs on G_U. For $\mu_U \in \mathcal{M}(G_U)$, the reduced MRF on G_U is denoted by $p_{G_U}(\mu_U)$. The marginal and reduced MRF distributions will be expressed in shorthand as $p_G^U(\theta)$ and $p_{G_U}(\theta_U)$, respectively.

For a subset C of nodes, the conditional distribution on X_C given a configuration $x_{\partial C}$ on its boundary is a reduced MRF on G_C, where the self potentials on the surface γC of the set C are modified to account for $x_{\partial C}$.

2.2 Basics of Information Theory

In this section we introduce the basic information theoretic quantities considered in this thesis. These are entropy, KL-divergence, and mutual information.

2.2.1 Entropy

The notion of entropy was introduced in the context of statistical physics by Boltzmann . A slightly generalized version was presented in 1948 by Shannon in the context of a communication problem. This latter version is the one we consider here. The *entropy* of a random field $\mathbf{X}$ is defined to be

$$H(\mathbf{X}) \quad = \quad -\sum_{\mathbf{x}\in\mathcal{X}} p(\mathbf{x}) \log p(\mathbf{x}).$$

The *conditional entropy* of random variable X_2 given another random variable X_1 is

$$H(X_2|X_1) = -\sum_{x_1} p(x_1) \sum_{x_2} p(x_2|x_1) \log p(x_2|x_1).$$

The entropy is a measure of the randomness of a stochastic process. For an MRF, the entropy has the form

$$H_G(\mathbf{X};\theta) \quad = \quad \log Q(\theta) - \mu^T\theta.$$

In words, the entropy is the log-partition function minus the inner product between the moment and exponential parameter vectors. The entropy of an MRF $p_G(\theta)$ will be denoted by $H_G(X;\theta)$, or in shorthand by $H_G(\theta)$, and the marginal entropy of a subfield X_U, $U \subset V$, by $H_G^U(X;\theta)$ or $H_G^U(\theta)$. The conditional entropy of a subset of nodes W given another subset U is denoted $H_G^{W|U}(\theta)$.

2.2.2 Divergence

The *Kullback-Leibler* (KL) divergence between probability distributions p and q is

$$D(\ p\ ||\ q\ ) = \sum_{x\in\mathcal{X}} p(\mathbf{x}) \log \frac{p(\mathbf{x})}{q(\mathbf{x})}. \tag{2.7}$$

If $p, q \in \mathcal{F}$ are MRFs on G induced by exponential parameters θ_1 and θ_2, respectively, the KL-divergence between p and q has the form

$$D(\ p(\theta_1)\ ||\ q(\theta_2)\ ) \quad = \quad \log \frac{Q_G(\theta_2)}{Q_G(\theta_1)} - \mu_1^T(\theta_2 - \theta_1). \tag{2.8}$$

The divergence is generally taken to be a measure of "distance" between two probability distributions. However, it is asymmetric, as $D(p||q)$ and $D(q||p)$ in general are not equal.

2.2.3 Mutual Information

The *mutual information* between random variables X_1 and X_2 is

$$\begin{aligned} I(X_1; X_2) &= H(X_1) - H(X_1|X_2) = H(X_2) - H(X_2|X_1) && (2.9) \\ &= H(X_1) + H(X_2) - H(X_1, X_2). && (2.10) \end{aligned}$$

and it is known that $I(X_1; X_2) = D(p(X_1, X_2)||p(X_1)p(X_2))$.

2.3 Basics of Source Coding

In this section we discuss the basic aspects of source coding. First we look at lossless compression, then Arithmetic Encoding (AC), then lossy compression. We will see that the quantities introduced in the previous section have operational significance for the compression of an information source.

In a source coding or compression scheme there is an encoder which transforms the input data $\mathbf{x}$ into a bitstring to be stored on a computer and a decoder that takes the binary data and transforms it into the output data $\hat{\mathbf{x}}$. The goals are storage of the data on computers in efficient form and faithful reconstructions, so the performance measures are the average number of bits stored over all possible input data, and the

average distortion. These are discussed below in more detail. In lossless compression the fidelity criterion is perfect reconstruction and therefore we want to minimize the average number of bits stored. In lossy compression, the fidelity criterion may vary according to the application, but for a fixed level of permitted distortion we want to minimize the average number of bits. Alternatively, the application may place a constraint on the average number of bits stored or transmitted, and for a fixed limit on this quantity, we want to minimize the distortion. Figure 2.6 illustrates this these basic concepts.

2.3.1 Lossless Compression

In lossless compression the output data $\hat{\mathbf{x}}$ in Figure 2.6 is identical to the input data $\mathbf{x}$. Here, we observe an observation $\mathbf{x}$ from a finite alphabet $\mathcal{X}$ and the encoder assigns to it a bit string of length $l(\mathbf{x})$. It is this bit string that is stored on a computer or transmitted over a channel and the encoding must be done in such a way that $\mathbf{x}$ is faithfully reproduced as the decoder output. To optimize the efficiency of such a coding scheme, we wish to minimize the expected bit string length $\mathbb{E}\left[l(\mathbf{X})\right]$ over all possible encoding schemes.

The following is well-known [9]:

Proposition II.1. For any random field $\mathbf{X}$,

$$\mathbb{E}\left[l(\mathbf{X})\right] \;\geq\; H(\mathbf{X}),.$$

It always possible to come within one bit of entropy, though for a finite alphabet $\mathbf{X}$, it may not be possible to attain it.

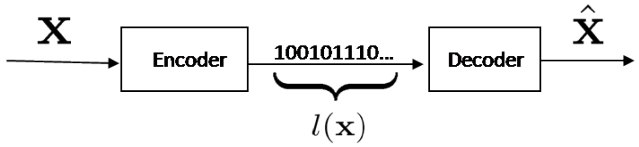


Figure 2.6: Block diagram illustrating basic idea of source coding.

2.3.2 Arithmetic Encoding

Arithmetic Encoding (AC) is a well-known lossless coding algorithm [44]. To apply Arithmetic Encoding to an MRF, we first order or *scan* the nodes, i.e., arrange them into a one-dimensional sequence, $\mathbf{x} = (x_1, x_2, \ldots, x_N)$. Then for $i = 1, \ldots, N$, the ith node value x_i is fed to the arithmetic *encoder* along with a *coding distribution* f_i, which is a function $f_i : \mathcal{X} \to [0, 1]$, $\sum_{x \in \mathcal{X}} f_i(x) = 1$. Ordinarily, f_i will also depend on some or all of the previous pixel values $x_1^{i-1} = (x_1, \ldots, x_{i-1})$, but this is not reflected in the notation. The encoder outputs a sequence of bits from which the arithmetic *decoder* recreates $\mathbf{x}$. This sequence of bits is a *codeword* for the image $\mathbf{x}$, and we denote its length as $l(\mathbf{x})$. The decoder uses a prefix of the codeword to decode x_1^{i-1}, and uses subsequent bits and f_i (which it can compute since $x_1, \ldots, x_{i-1}$ are known) in the decoding of x_i. For the purposes of this thesis, it suffices to state the well known fact [45] that the length of the codeword produced by AC when encoding $\mathbf{x}$ with the sequence of coding distributions $f_1, \ldots, f_N$ is

$$l(\mathbf{x}) \;\approx\; -\log\left(\prod_{i=1}^{N} f_i(x_i)\right) \;=\; -\sum_{i=1}^{N} \log f_i(x_i) \;,$$

where all logs have base 2. In a slight abuse of notation we refer to the distribution $f(\mathbf{x}) \stackrel{\Delta}{=} \prod_{i=1}^{N} f_i(x_i)$ as *the coding distribution*. Figure 2.7 illustrates the AC encoding and decoding algorithm.

We let $p_{i|*}(x_i|x_1^{i-1})$ denote the conditional probability that the ith scanned node assumes value x_i when the preceding $i-1$ symbols have the value x_1^{i-1}. It is straight-

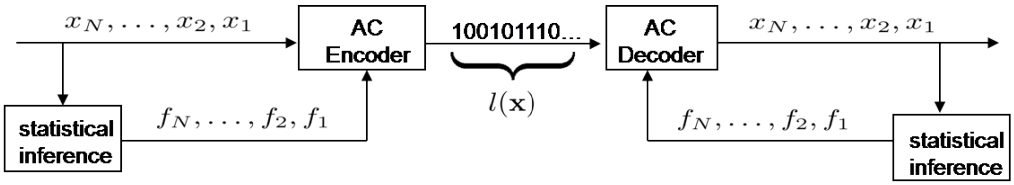


Figure 2.7: Block diagram illustrating Arithmetic Encoding.

forward to show that if $f_i = p_{i|*}$ for all i, then the average codeword length satisfies

$$\mathbb{E}\, l(\mathbf{X}) \;\approx\; H(\mathbf{X}) \;,$$

where $\mathbb{E}$ denotes expected value. In this case we say that the coding distribution f is *optimal for* p. If the source $\mathbf{X}$ is distributed according to distribution p and the coding distribution is optimal for another distribution q, it is well-known [9] that the average codeword length is

$$\mathbb{E}\, l(\mathbf{X}) \;\approx\; H(\mathbf{X}) + D(p||q) \;,$$

where $D(p||q)$ is the KL-divergence between distributions p and q. Since the term $D(p||q)$ is amount by which the rate exceeds optimal coding, it is referred to as the *redundancy* in rate.

2.3.3 Lossy Compression

Sometimes it is unnecessary to require perfect reconstruction at the decoder. In the case of lossy compression, the output $\hat{\mathbf{x}}$ in Figure 2.6 is in general not identical to the input $\mathbf{x}$. We introduce a *distortion measure* $d : \mathcal{X} \times \hat{\mathcal{X}} \mapsto \mathbb{R}^+$ such that if $\mathbf{x}$ is the original image and $\hat{\mathbf{x}}$ is the reconstruction, then $d(\mathbf{x}, \hat{\mathbf{x}})$ is the distortion between

the original and reconstructed images. We then define

$$D \triangleq \mathbb{E}\left[d(\mathbf{X}, \hat{\mathbf{X}})\right]$$

to be the average distortion over all images. The goal is few bits and small average distortion.

The transition probability $p^T(\hat{\mathbf{x}}|\mathbf{x})$ distribution between input images $\mathbf{x}$ and reconstructions $\hat{\mathbf{x}}$ allows us to compute the average distortion

$$\begin{aligned} \bar{D}(\mathbf{X}, \hat{\mathbf{X}}) &= \mathbb{E}_{p^T}\left[d(\mathbf{X}, \hat{\mathbf{X}})\right] \\ &= \sum_{\mathbf{x}} p(\mathbf{x}) \sum_{\hat{\mathbf{x}}} p^T(\hat{\mathbf{x}}|\mathbf{x}) d(\mathbf{x}, \hat{\mathbf{x}}). \end{aligned}$$

2.4 Inference and Estimation

Inference in the context of this thesis includes computing a marginal probability at a given node i or a block of nodes B; or computing the the maximum probabilities for blocks with different values at node i. The former involves computing Z_i, the belief for node i, and since marginalization involves summing over a subset of nodes and because the distribution of an MRF can be expressed as a product, marginalization[2] is referred to as the *Sum-Product* problem. The beliefs Z_i, then, are really *Sum-Product* beliefs but since we use these more throughout the thesis, they will simply be called beliefs. Analogously, the latter problem, of finding maximum probabilities, is called the *Max-Product* problem. For instance, if observations are made at a subset of nodes ∂B, we can compute, for each node i and each value x_i, the *Max-Product belief*

$$Z_i^{\mathrm{MP}}(x_i) \triangleq \max_{x_{B\setminus i}} \prod_{k \in B} \Phi_k(x_k) \prod_{\{k,j\} \in E} \Psi_{k,j}(x_k, x_j), \tag{2.11}$$

[2]Also, finding a conditional probability, which is a marginal with certain nodes fixed.

where B is a block of sites and E is the set of edges in $B \cup \partial B$. As with Sum-Product beliefs, the Max-Product beliefs are considered equivalent up to a scale factor, though in the Max-Product case the components will in general not sum to 1. If for each node $i \in B$ we choose an estimate $\hat{x}_i \in \operatorname{argmax}_{x'_i} Z_i^{\mathrm{MP}}(x'_i)$, this is used as a proxy for finding a *maximum aposteriori* (MAP) estimate of the nodes in B conditioned on $x_{\partial B}$, giving

$$\hat{x}_B \in \operatorname*{argmax}_{x'_B} p_G^{B|\partial B}(x'_B | x_{\partial B}). \tag{2.12}$$

When there are ties for $\hat{x}_i \in \operatorname{argmax}_{x'_i} Z_i^{\mathrm{MP}}(x'_i)$, then breaking the ties at individual nodes rather than jointly may result in something that is not truly MAP. Computing either the beliefs or the Max-Product beliefs for a node i is nontrivial since it involves, respectively, a number of summations or comparisons that is exponential in $|V|$ or $|B|$. However,when the underlying graph G is acyclic, there is an efficient algorithm, called *Belief Propagation* (BP) [39], for computing these beliefs. The algorithm involves a sequence of recursive message passing operations between neighboring nodes, where a message is a table of numbers each corresponding to a value in the alphabet of the receiving node.

2.4.1 Belief Propagation

We now introduce the Belief Propagation algorithm and do so in the context of the beliefs Z_i. However, the same algorithm applies for computing Max-Product beliefs Z_i^{MP}, replacing "$\sum$" with "max" in what follows. An edge $\{i, j\}$ of a connected graph G such that removing the edge connecting i and j disconnects the graph is called a *cut edge*. For a cut edge $\{i, j\}$, we let $G_{i \setminus j}$ be the component of $G \setminus j$ that contains i, and similarly for $G_{j \setminus i}$. Analogously, we let $Z_{i \setminus j}(x_i)$ be the belief at node i in $G_{i \setminus j}$ evaluated at the value x_i. This notation allows us to derive the familiar BP equations.

Definition II.2. *For cut edge* $\{i, j\}$*, the* message *from node* j *to node* i *is defined to be the set of all values of the function*

$$m_{j\to i}(x_i) = \sum_{x_j} \Psi_{ij}(x_i, x_j) Z_{j\setminus i}(x_j). \tag{2.13}$$

Alternatively, $m_{j\to i} = \{m_{j\to i}(x_i) : x_i \in \mathcal{X}\}$ *can be viewed as a message from the subgraph* $G_{j\setminus i}$ *to the subgraph* $G_{i\setminus j}$*.*

For an acyclic graph every edge is a cut edge, and the above definition can be used to establish a recursive relationship between outgoing messages of adjacent nodes in the graph. Below is a well-known proposition for the beliefs of an acyclic MRF computed with BP. The proofs of Propositions II.3 and II.4 are given in Section 2.5.

Proposition II.3 (Belief Decomposition). Let $G = (V, E)$ be acyclic and connected. Then, for each node $i \in V$ and all $x_i \in \chi_i$, we have

$$Z_i(x_i) \;=\; \Phi_i(x_i) \prod_{j\in\partial i} m_{j\to i}(x_i), \tag{2.14}$$

where $m_{j\to i}$ is the message from node j to node i.

The messages in the above proposition are easily computed in a recursive manner using the proposition below.

Proposition II.4 (Message Recursion). For an acyclic graph, the Belief Propagation messages follow the recursion

$$m_{j\to i}(x_i) \;=\; \sum_{x_j} \Psi_{ji}(x_j, x_i)\Phi_j(x_j) \prod_{k\in\partial j\setminus i} m_{k\to j}(x_j),$$

where $\partial k \setminus i$ is the set of neighbors of node k except i.

Propositions II.4 and II.3 motivate a recursive algorithm for computing messages and beliefs. The recursion begins with each leaf node j sending to its unique neighbor

δj the message component

$$m_{j\to\delta j}(x_{\delta j}) \;=\; \sum_{x_j} \Phi_j(x_j)\Psi_{j,\delta j}(x_j, x_{\delta j}) \tag{2.15}$$

for each $x_{\delta j} \in \mathcal{X}$. The algorithm proceeds with each non-leaf node j passing to neighbor $i \in \delta j$ the message $m_{j\to i}$ with components determined by

$$m_{j\to i}(x_i) = \sum_{x_j} \Psi_{j,i}(x_j, x_i)\Phi_j(x_j) \prod_{k\in\partial j\setminus i} m_{k\to j}(x_j) \;, \tag{2.16}$$

with the rule that node j does not form the message to send to i until receiving incoming messages from all other neighbors. After all messages are computed the beliefs for any node i can be computed using (2.14). Moreover, the messages $\{m_{j\to i}, m_{i\to j} : \{i,j\} \in E\}$ can be used to compute, for example, the belief for a connected subset of nodes $A \subseteq V$[30]. The complexity of BP is on the order of $|V||\mathcal{X}|^2$. That is, it is linear in the number of nodes of G for an acyclic graph and the square of the alphabet size.

We can also express the belief propagation equations in matrix notation, which will be useful in comparing analogous formulas for Local Conditioning algorithm presented later. For edge $\{i,j\}$, we define $A_{i,j} = [\Psi_{i,j}(x_i, x_j)]$ be the matrix of potentials between different values of x_i, x_j, where the rows correspond to values of x_j and columns values of x_i. If we let $m_{j\to i}$ denote the vector $[m_{j\to i}(0), \ldots, m_{j\to i}(|\mathcal{X}|-1)]^T$, Φ_j the vector $[\Phi_j(0), \ldots, \Phi_j(|\mathcal{X}|)]^T$, $\odot$ component-wise multiplication, and $\cdot$ ordinary matrix multiplication, the message recursion is

$$m_{j\to i} = A_{j,i} \cdot \left[\Phi_j \odot \prod_{k\in\partial j\setminus i}^{\odot} m_{k\to j} \right], \tag{2.17}$$

while the belief at node i is

$$Z_i = \Phi_i \odot \prod_{k \in \partial i}^{\odot} m_{k \to i}. \tag{2.18}$$

2.4.2 BP with Fixed Nodes

Suppose that a subset of the nodes $L \subset V$ of an acyclic graph T has fixed values $\bar{x}_l$, $l \in L$. The BP algorithm presented in Section 2.4.1 can still be applied to an MRF on T by modifying the self-potential of the fixed leaf node as

$$\hat{\Phi}_l(x_l) = \Phi_l(x_l)\delta(\bar{x}_l, x_l), \tag{2.19}$$

With this modification to the self-potentials of fixed nodes, the usual message and belief formulas apply, the delta function picking out the fixed value for each node.

2.4.3 BP on a Forest

Suppose that T is disconnected. For simplicity's sake, assume that T consists of two components, $T_1 = (V_1, E_1)$ and $T_2 = (V_2, E_2)$. The belief at node $i \in T_1$ is then

$$\begin{aligned}
Z_i(x_i) &= \Phi_i(x_i) \sum_{x_{V \setminus i}} \prod_{k \in V \setminus i} \Phi_k(x_k) \prod_{\{j,k\}} \Psi_{j,k}(x_j, x_k) \\
&= \Phi_i(x_i) \sum_{x_{V_1 \setminus i}} \prod_{k \in V_1 \setminus i} \Phi_k(x_k) \prod_{\{j,k\} \in E_1} \Psi_{j,k}(x_j, x_k) \\
&\quad \cdot \sum_{x_{V_2}} \prod_{k \in V_2 \setminus i} \Phi_k(x_k) \prod_{\{j,k\} \in E_2} \Psi_{j,k}(x_j, x_k) \qquad (2.20) \\
&= Z_i^{T_1}(x_i) Q^{T_2} \\
&= \Phi_i(x_i) \prod_{k \in \partial i} m_{k \to i}(x_i) Q^{T_2}, \qquad (2.21)
\end{aligned}$$

where $Z_i^{T_1}$ is the belief at node i in component T_1 and Q^{T_2} is the partition function for component T_2. Equality (2.20) above follows from the fact that there are no edges connecting T_1 and T_2, and therefore the two sums separate into factors. Since Q^{T_2}

is a constant for all values of x_1 this shows that each component can be viewed as a separate MRF, and hence the MRFs on T_1 and T_2 are independent of each other. This naturally generalizes to graphs of more than two components.

Equation (2.21) indicates that it is not necessary to run BP on the separate components. For instance, one can simply begin the algorithm at leaves as usual and follow the BP operations. However, for node i the product of its self-potential and the incoming messages is not the belief as defined in (2.2), but rather the belief with respect to the MRF *on its component*. To get the full belief as in (2.2) we must multiply the component belief by the partition functions of the remaining components. However, since the values of these partition functions are constant for all values of x_i, the probability distribution at each node i can be computed using only the self-potential and the incoming messages that are computed on its component. Later, in Section 4.2.1.2, we will encounter a situation where we will be running BP on a disconnected graph where not only are some of the nodes fixed, but fixed nodes on different components are constrained to assume the same value. In this case, as we will discuss then, the partition functions from different components cannot be ignored.

2.4.4 Clustering Algorithm for Cyclic Graphs

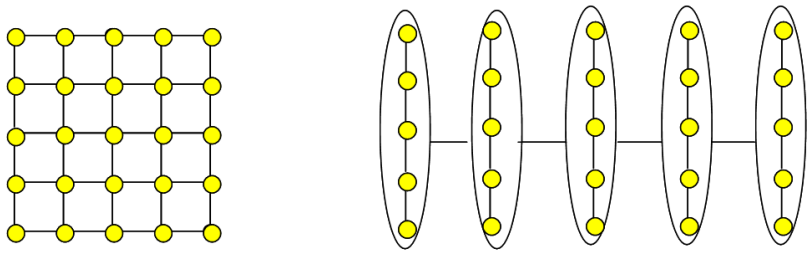
So far BP applies to acyclic graphs. When the graph has cycles, one possibility is to partition the nodes into supernodes and form an acyclic graph of supernodes. It is always possible to partition the nodes into supernodes. A *cluster graph*, is then formed by connecting two supernodes with an edge if each of the supernodes contains an endpoint of an edge of the original cyclic graph. If supernodes can be chosen so that the resulting cluster graph is a *cluster tree*, then one can apply BP and use it to compute beliefs for the nodes in the original graph. Figure 2.8 (a) shows an original grid graph and (b) a cluster graph formed by lumping columns into supernodes.

Potentials Φ^{C} and Ψ^{C} are assigned to the cluster tree in the following way. For a supernode c_i, the self potential Φ^{C} is defined as

$$\Phi^{\mathrm{C}}_{c_i}(x_{c_i}) = \prod_{j \in c_i} \Phi_j(x_j) \prod_{i,j \in c_i} \Psi_{i,j}(x_i, x_j). \tag{2.22}$$

For two supernodes c_i and c_k connected by an edge in $\hat{G}$, the edge potential is defined as

$$\Psi^{\mathrm{C}}_{c_i,c_j}(x_{c_i}, x_{c_j}) = \prod_{m \in c_i, n \in c_j} \Psi_{m,n}(x_m, x_n). \tag{2.23}$$



(a) (b)

Figure 2.8: (a) Original 4-pt. grid graph; (b) Cluster graph.

This defines an MRF on $\hat{G}$ for which beliefs Z_{c_i} are naturally defined. Since the junction tree is acyclic, BP can be run on the cluster tree, computing beliefs for supernodes c_i. Beliefs for individual nodes i of G can be computed, for example, by computing the beliefs for a supernode containing i and summing out the nodes other than i. The complexity of the junction tree algorithm is on the order of the number of supernodes times $|\mathcal{X}|^{\max_i |c_i|}$. Thus, it is exponential in the size of the largest supernode. As can be seen from Figure 2.8, the complexity of performing exact inference on an $M \times N$ grid graph using the cluster method is exponential in the smaller of M and N.

2.4.5 Loopy Belief Propagation

For cyclic graphs, the complexity of the Cluster Tree method might be too great, in which case an approximation algorithm must be used. The most common such algorithm is called *Loopy Belief Propagation* (LBP) [39], which is an iterative algorithm that operates by treating the cyclic graph as if it were locally a tree. In this thesis we will consider only the Max-Product variant of LBP, in Section 3.3, so we give the LBP update formulas in terms of it:

$$m_{j\to i}^{(n)}(x_i) = \max_{x_j} \Psi_{j,i}(x_j, x_i)\Phi_j(x_j) \prod_{k\in\partial j\setminus i} m_{k\to j}^{(n-1)}(x_j) \ , \tag{2.24}$$

which give, at each iteration n and each node i, an updated LBP Max-Product belief

$$\widehat{Z}_i^{(n),\mathrm{MP}}(x_i) = \Phi_i(x_i) \prod_{j\in\partial i} m_{j\to i}^{(n-1)}(x_i), \tag{2.25}$$

where the iterates $\{\widehat{Z}_i^{(n)}\}$ are estimates of the Max-Product belief vector Z_i^{MP}. The algorithm is typically initialized with uniform values, as in

$$m_{j\to i}^{(0)} = \mathbf{1}.$$

Example II.5. *This example is relevant to the lossy coding problem discussed in Chapter III. Consider a binary MRF on a graph G, and let B denote a block of sites where a configuration $x_{\partial B}$ is observed on its boundary ∂B. Now suppose that we run Max-Product LBP in B and that the beliefs $Z_i^{(n)}$ converge after N iterations. Furthermore, suppose that the computed beliefs are correct, i.e., $Z_i^{(N)} = Z_i^{MP}$ for all $i \in B$. We now wish to use the (correct) computed beliefs to do MAP estimation of X_B conditioned on $X_{\partial B} = x_{\partial B}$.*

If for $i \in B$, $Z_i^{(N)}(-1) > Z_i^{(N)}(1)$, then in all MAP configurations $\hat{x}_B$ on B given

$x_{\partial B}$, $\hat{x}_i = -1$, so we can safely estimate X_i as -1. Similarly, if $Z_i^{(N)}(1) > Z_i^{(N)}(-1)$, X_i can be estimated as a 1. However, if for site i, $Z_i^{(N)}(-1) = Z_i^{(N)}(1)$, then there is a MAP configuration $\hat{x}_B^1$ in which $\hat{x}_i^1 = -1$ and a MAP configuration $\hat{x}_B^2$ in which $\hat{x}_i^2 = 1$. If there is only one site i that can assume different values in different MAP configurations on B, then we can choose arbitrarily between estimating X_i as -1 or 1, since all other sites in B assume only one possible value under all MAP configurations. However, if there are two sites i and j for which $Z_i^{(N)}(-1) = Z_i^{(N)}(1)$ and $Z_j^{(N)}(-1) = Z_j^{(N)}(1)$, then we cannot independently estimate X_i and X_j. This is because $\hat{x}_i$ and $\hat{x}_j$ may equal each other in all MAP configurations, so that estimating them independently could result in a block estimate $\hat{x}$ where $\hat{x}_i \neq \hat{x}_j$, which would not be MAP.

Loopy Belief Propagation has been the subject of intense research over the last decade plus [57, 37, 58, 50, 62, 55], in part because its distributed nature is appealing and part because it has been demonstrated to work well in practice [37]. Despite the empirical success, little is known about the exact answers produced by LBP [37]. However, it is now known that LBP can be understood as a limiting case of a family of approximate reparameterizations based on MRFs on subtrees of the original cyclic graph [55], and this analysis has afforded insight into the closeness of the approximation. Additionally, it can also be viewed as an extreme case of a family of approximate clustering algorithms [62], where greater accuracy can be obtained by clustering into larger groups of nodes.

2.5 Proofs

2.5.1 A Useful Lemma

The following lemma is used in proving Propositions II.3 and II.4.

Lemma II.6. *If node i lies on cut edge $\{i, j\}$, then*

$$Z_i(x_i) = Z_{i\setminus j}(x_i) m_{j\to i}(x_i), \tag{2.26}$$

where we say that $m_{j\to i}(x_i) = \sum_{x'_j} \Psi_{ij}(x_i, x'_j) Z_{j\setminus i}(x'_j)$ is the message *from node j to node i.*

Proof. First, for a subset $A \subset V$ of nodes, we let

$$\Phi'_A(x_A) = \prod_{i\in A} \Phi_i(x_i) \prod_{\{i,j\}\subset A} \Psi_{i,j}(x_i, x_j).$$

In other words, we define the self-potential Φ' on a subset of nodes to be the product of the potentials Φ and Ψ on nodes and edges contained within the subset Then, for fixed but arbitrary $\bar{x}_i$, we see that

$$\begin{aligned}
Z_i(\bar{x}_i) &= \Phi_i(\bar{x}_i) \sum_{x_{\setminus i}} \Psi_{ij}(\bar{x}_i, x_j) \Phi'_{V_{i\setminus j}}(x_{V_{i\setminus j}}) \Phi'_{V_{j\setminus i}}(x_{V_{j\setminus i}}) \\
&= \Phi_i(\bar{x}_i) \left(\sum_{x_{V_{i\setminus j}\setminus i}} \Phi'_{V_{i\setminus j}}(x_{V_{i\setminus j}}) \right) \cdot \left(\sum_{x_{V_{j\setminus i}}} \Psi_{ij}(\bar{x}_i, x_j) \Phi'_{V_{j\setminus i}}(x_{V_{j\setminus i}}) \right) \\
&= Z_{i\setminus j}(\bar{x}_i) \left(\sum_{x_j} \Psi_{ij}(\bar{x}_i, x_j) \sum_{x_{V_{j\setminus i}\setminus j}} \Phi'_{V_{j\setminus i}}(x_{V_{j\setminus i}}) \right) \\
&= Z_{i\setminus j}(\bar{x}_i) \left(\sum_{x_j} \Psi_{ij}(\bar{x}_i, x_j) Z_{j\setminus i}(x_j) \right) \\
&= Z_{i\setminus j}(\bar{x}_i) m_{j\to i}(\bar{x}_i),
\end{aligned}$$

which proves the lemma. □

2.5.2 Proof of Proposition II.3

Proof. We prove by induction on the number $n = \mid \partial i \mid$ of neighbors of i. For $n = 1$ this follows immediately Lemma II.6, as

$$\begin{aligned} Z_i(x_i) &= Z_{i\backslash j}(x_i) m_{j\to i}(x_i) \\ &= \Phi_i(x_i) m_{j\to i}(x_i). \end{aligned}$$

We now make the induction hypothesis that the proposition holds for some $\mid \partial i \mid = k \geq 1$. For the case $\mid \partial i \mid = k+1$, we enumerate the members of ∂i as $1, \ldots, k+1$. Then,

$$\begin{aligned} Z_i(x_i) &= Z_{i\backslash k+1}(x_i) m_{k+1\to i}(x_i) && (2.27) \\ &= \Phi_i(x_i) \prod_{t=1}^{k} m_{t\to i}(x_i) m_{k+1\to i}(x_i) && (2.28) \\ &= \Phi_i(x_i) \prod_{t\in\partial i} m_{t\to i}(x_i), \end{aligned}$$

where (2.27) follows from Proposition II.6, and (2.28) follows from the inductive hypothesis. □

2.5.3 Proof of Proposition II.4

Proof.

$$\begin{aligned} m_{i\to j}(x_j) &= \sum_{x_i'} \Psi_{ij}(x_i', x_j) Z_{i\backslash j}(x_i') \\ &= \sum_{x_i'} \Psi_{ij}(x_i', x_j) \Phi_i(x_i') \prod_{k\in\partial i\backslash j} m_{k\to i}(x_i'), \end{aligned}$$

where the last equality comes from applying Proposition (II.3) to the subgraph $G_{i\setminus j}$.

□

CHAPTER III

Lossy Cutset Coding of Ising Model

In this chapter we introduce the concept of lossy Cutset Coding and apply it to a homogeneous Ising model. An image $\mathbf{x}$ from an Ising distribution is observed at the encoder. The encoder losslessly compresses the values x_U on a *square grid* cutset U of the graph and transmits the resulting bit string to the decoder. The decoder perfectly reconstructs the cutset values x_U and estimates the missing values $x_{V\setminus U}$ conditioned on x_U. The cutset U partitions the remaining sites $V \setminus U$ into disjoint blocks that are conditionally independent of each other from the Markov property, so that each block can be estimated independently on the basis of its immediate boundary. There are many estimation criteria for $X_{V\setminus U}$, including MAP estimation of each pixel given the cutset or of each block conditioned on the cutset. For reasons discussed in Section 3.1 we focus on block MAP estimation. The main emphasis of this chapter, then, is deriving analytical solutions to the block MAP estimation of a block from its immediate boundary. In Section 3.2 we give exact solutions to the set of block MAP configurations conditioned on boundaries with four or fewer black-to-white transitions, which are the predominantly occurring boundary configurations.

We are able to find these solutions because block MAP estimation in a homogeneous Ising model is the combinatorial problem of finding the interior configuration that results in the fewest odd bonds together with the boundary. One can also use

Loopy Belief Propagation to do the estimation, at least approximately. In Section 3.3 we use the exact analytical solutions to the cases of four or fewer boundary odd bonds to examine the performance of Max-Product Loopy Belief Propagation as a decoding, i.e., interpolation, tool. Specifically, the iterations of LBP produce for each site in a block a sequence of estimates for the difference between the minimum numbers of odd bonds when the site is white or black. Our analytical block MAP solutions permit us to state what these differences actually should be in the most common cases mentioned above, and we can run Max-Product LBP and compare the answers given by it to the correct answers. We see empirically that LBP converges and converges to the correct values.

Peierls showed that for the family of homogeneous Ising models on the 4 pt. grid graph a phase transition did occur [40] and he did so by considering a site at the center of a large block of sites. He demonstrated that for ranges of the coupling parameter θ, the configuration on the boundary of the block biased the conditional distribution of the center site, even as the block size tended to infinity. And it was the existence of this bias that indicated the long-range dependence associated with a phase transition.

The block MAP reconstruction results in this chapter, therefore, are in a sense complementary to Peierls' results. Because we give block MAP solutions for boundaries of four or fewer boundary odd bonds, these results apply to small or moderate sized blocks as opposed to the very large blocks of Peierls' analysis. Moreover, whereas Peierls' work showed the existence of an effect of the boundary configuration on the conditional distribution of a center pixel, our work describes the precise effect of a boundary configuration on an extremum of the conditional distribution for the entire block interior.

In Section 3.1 we formally introduce the new encoding/decoding framework and in Section 3.2 we discuss analytical solutions for MAP reconstructions of a block, in

both the 4 pt. and 8 pt. homogeneous Ising models, for boundary configurations with four or fewer black-white transitions. In Section 3.3 we use the analytical results from Section 3.2 to analyze the performance of Max-Product Loopy Belief Propagation as an alternative decoding/interpolation algorithm. In Section 3.4 we give reconstruction results for different images and in Section 3.5 we introduce a modification to the MAP decoding algorithm to enhance perceptual quality of the reconstructions. Finally, in Section 3.6 we discuss a simple adaptive method for losslessly encoding the grid cutset, show a rate-distortion plot of performance and discuss how erasure entropy can be used to analyze this rate. Proofs are given either inline or in the final section of this chapter.

3.1 Lossy Cutset Coding

The information source is a (homogeneous) Ising model defined on either a 4-pt. or an 8-pt. grid graph G. A cutset $U \subset V$ of sites consisting of every Nth row and every Nth column of G, later referred to as a *grid cutset*, is chosen and the image x_U on the cutset is losslessly encoded at rate

$$R_U = \frac{1}{|U|}\mathbb{E}\left[l(x_U)\right] \text{ bits/sample}$$

for example using Arithmetic Encoding. Because the remaining pixels $x_{V\setminus U}$ are not encoded, the rate $R_{V\setminus U}$ is zero, so that the overall rate is

$$R = \frac{|U|}{|V|}R_U.$$

The grid spacing parameter N determines the rate R_U. Moreover, as the cutset partitions the remaining sites $V \setminus U$ into disjoint $N-1 \times N-1$ blocks, by the Markov property these blocks can be optimally estimated independently of one an-

other. Therefore, the grid spacing N also determines the size of the blocks that must be estimated from their respective boundaries. As an estimation criterion, we can find the per pixel MAP reconstruction, that is, to find the value at each pixel that maximizes its conditional probability distribution given the cutset values. This estimation criterion is equivalent to minimizing the *Hamming* distortion, which is given by

$$D(\mathbf{x}, \hat{\mathbf{x}}) = \frac{1}{|V|} \sum_{i \in V} \mathbb{I}_{(x_i \neq \hat{x}_i)},$$

where $\mathbb{I}_{(\cdot)}$ is the indicator function assuming value 1 when the argument is true and 0 otherwise. Alternatively, we can find the block MAP reconstruction, that is, find the configuration within each block that maximizes its conditional probability given the cutset values. This is equivalent to minimizing the number of odd bonds within the block estimate and between it and its boundary. Both of these estimation schemes can in theory be carried out with a variant of BP, the former as a Sum-Product problem and the latter as a Max-Product. Because of the many cycles, however, LBP would have to be used and the performance of this algorithm is unknown in this setting [37].

Since the estimation can be carried out independently block-by-block and since MAP estimation reduces to minimizing the number of odd bonds in the block configuration, we can develop some simple combinatorial arguments showing the complete set of MAP estimates. We do this analytically in Section 3.2 for the most commonly occurring boundaries consisting of a single run of black pixels and two runs of black pixels, and in Section 3.3 we use these results to analyze LBP as a decoding algorithm. Moreover, minimizing the number of odd bonds within each block yields a "smoothest" reconstruction consistent with the preserved cutset information. This is important since we do have in mind the application of this coding method to the encoding of real-world bilevel images.

In Section 3.4 we consider the application of the MAP decoding algorithm to real-world bilevel images, such as shown in Figure 2.2 (a) as well as to typical images from

different homogeneous Ising models. In Section 3.5 we introduce a modification to MAP decoding, called *decision-bit* decoding, on the basis of the following observation. For blocks whose boundaries contain two runs of black and two runs of white, there are two principal ways to reconstruct the block interior, one with a white swath through the interior and the other with a black swath. For blocks with two runs of white and black, then, the encoder encodes an additional bit informing the decoder which of these two basic patterns to pursue. The reason for this modification, of course, is to enhance the perceptual quality of the reconstructed images.

The lossless coding of the grid can be done in a variety of ways. In Section 3.6 we consider a simple adaptive method using Arithmetic Encoding. In Chapter V we explore relationships between the marginal distribution of a given subset X_U of an MRF and an MRF defined on the subgraph induced by U. Furthermore, the concept of lossy Cutset Coding is related to *erasure entropy* [51, 52] and in Section 3.6 we also begin to explore erasure entropy as a means of analyzing the rate of encoding the grid cutset.

3.2 Analytical Block MAP Solutions

In this section we derive analytic solutions for the set of (block) MAP estimates of the interior of a block B given the values on its boundary ∂B. In the case of uniform edge potentials ($\theta_{ij} = \theta$) and no external field ($\alpha_{ij} = 0$), the probability of a block x_B conditioned on the boundary $x_{\partial B}$ is a monotone function of the number of odd bonds in the joint realization $(x_B, x_{\partial B})$. Therefore, if x_B^* is a MAP estimate given $x_{\partial B}$, then for any other configuration x_B, $(x_B, x_{\partial B})$ has at least as many odd bonds

as $(x_B^*, x_{\partial B})$. In other words, a MAP estimate satisfies

$$\begin{aligned} \hat{x}_B &\in \operatorname*{argmax}_{x'_B \in \mathcal{X}_B} \Pr\left(x'_B | x_{\partial B}\right) \\ &= \operatorname*{argmin}_{x'_B \in \mathcal{X}_B} O(x'_B, x_{\partial B}). \end{aligned}$$

If there were no boundary constraints, an optimal block would be either the all black or all white configurations. It is intuitive, then, that with the boundary constraints an optimal interior x_B^* would be such that the sites decoded as 1 (black) should be connected to the sites on the boundary whose value is 1 and the sites decoded as -1 (white) should be connected to the boundary sites whose value is -1. Where it gets interesting are the contours separating regions of black and white. We first prove a basic result that allows us to limit attention to block reconstructions defined by "smooth" contours connecting the endpoints of these runs of 1s. We then argue constructively what the form of these MAP reconstructions should be in the case that there is a single run of 1s on the boundary. We then use the reconstruction results for the single-run case to construct MAP solutions for the case of two runs on the boundary.

3.2.1 Preliminaries

We consider square grid graphs with either the 4pt. or 8pt. topology. We will often refer to the 4 pt. or 8 pt. *models*, which include both the particular topology and the homogenous coupling parameter and the absence of an external field. A block or block *interior* is denoted by B and we define $\overline{B} = B \cup \partial B$ to be the *closure* of B. We will index sites in $\bar{B}$ with coordinates (n, m) in the usual way for matrices. That is, the top left corner has coordinates $(1, 1)$ and the bottom right corner has coordinates $(N + 1, N + 1)$. Recall that a path is a sequence of sites such that any two successive sites are connected by an edge and that a subset of sites $A \subset V$ is

connected if for all distinct sites $i, j \in A$, there exists a path from i to j. An HV-path is a path p where successive sites are joined by a vertical or horizontal edge in E. All paths in the 4-pt. topology are HV-paths. The subset A is said to be *HV*-connected if for distinct sites $i, j \in A$, there exists an HV-path from i to j. If A is connected but not HV-connected, then we say that A is 8-*connected*. Clearly, in the 4pt. topology, a set is either HV-connected or disconnected whereas in the 8pt. topology a set can be HV-connected, 8-connected, or disconnected. A *loop* is a path where the first and last sites coincide. As with paths, all loops in the 4-pt. topology consist of horizontal and vertical edges. A site i is in the *interior* $I(l)$ of a loop l if every HV-path from i to the boundary intersects l.

We now consider subsets of B induced by a particular configuration in the interior. Under configuration x_B, the set $\bar{B}^b = \bar{B}^b(x_B)$ is the set of sites in $\overline{B}$ whose value is 1 (black) in x_B. Similarly for $\bar{B}^w$. In the following, the dependence of $\bar{B}^b$ or $\bar{B}^w$ on $x_{\bar{B}}$ should be clear from context and will be omitted. Under a given configuration, a *monotone* path is a path of sites whose values (colors) are all the same. Likewise for a monotone loop, whose interior is defined as above.

An *island* is a subset of sites that form the interior of a monotone loop. The sites in an island are not connected through a monotone HV path to the boundary. Alternatively, if each site in the block interior is connected to the boundary through a monotone HV path, then there are no islands. We have the following lemma about islands in MAP reconstructions, which holds for both 4 pt. and 8 pt. models.

Lemma III.1 (Cycle filling)**.** *Let x_B^* be a MAP configuration on B conditioned on $x_{\partial B}$. If l is a monotone loop of sites, then the interior is monotone of the same color. Equivalently, in a MAP reconstruction, there is a monotone HV path connecting each interior site to the boundary.*

We define a *widget* to be 2×2 subblock of $x_{\bar{B}}$ such that the upper right and lower left corners are black (or white) and the upper left and lower right corners are white

(or black).

3.2.2 No Odd Bounds in Boundary

If there are no odd bonds in the boundary, then the boundary is either all black or all white. By Lemma III.1 the interior must be the same color.

3.2.3 Two Odd Bonds in Boundary

In this section we will derive the MAP solutions for block boundaries with two odd bonds. One can look at the boundary as consisting of two *runs*, one of black sites and the other of white sites, and because of this we will refer to such boundaries as *one-run* boundaries. We denote the black run as $r^b = (r_1^b, r_2^b)$ and the white run as $r^w = (r_1^w, r_2^w)$, where (r_i^b) and (r_i^w) are the coordinates of the *run endpoints*. Lemma III.1 tells us that for each site $i \in B$, there is a monotone HV-path to the boundary. For a boundary with two odd bonds, the black boundary sites are HV-connected. Therefore, in a MAP reconstruction the set $\bar{B}_b$ is HV-connected. Similarly with the white boundary sites and a MAP reconstruction set $\bar{B}_w$. This means that a MAP reconstruction $x^*_{\bar{B}}$ can be described by a (potentially backtracking) monotone *reconstruction path* connecting the endpoints of either the black or white runs that exclude boundary sites of the opposite color. Figure 3.16 (a) shows a block boundary, (b) an arbitrary coloring of the inside, and (c) and (d) two different applications of the Cycle-Filling lemma. Note that because the black regions are HV-connected, the outer contours can be described by a path. By indicating a black reconstruction path from r_1^b to r_2^b, one simultaneously indicates an adjacent white reconstruction path from r_1^w to r_2^w. It is these adjacent paths that determine a MAP reconstruction in the following way: the black reconstruction path from r_1^b to r_2^b combined with the black run r^b make a monotone loop the interior of which must be filled in black by Lemma III.1. Likewise for the monotone loop from r_1^w to r_2^w and the monotone path r^w. For

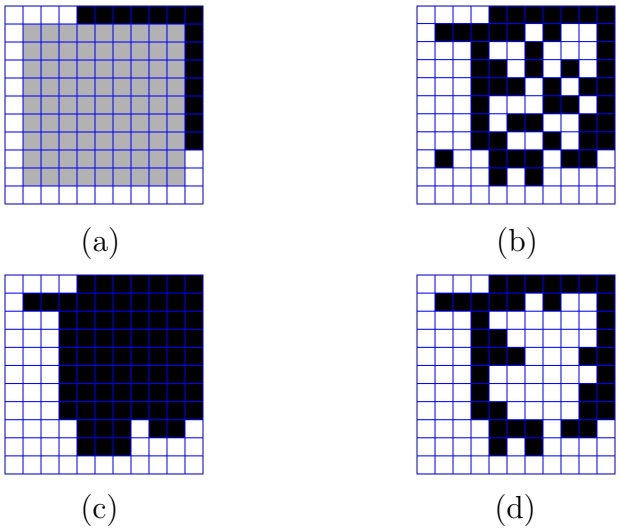


Figure 3.1: (a) Block boundary; (b) Arbitrary filling of interior; c) Interior obtained by applying Cycle filling lemma; (d) Another block obtained through Cycle filling lemma.

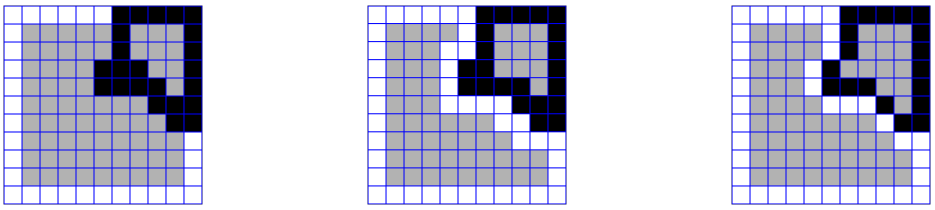
concreteness we will focus on black reconstruction paths connecting the black run endpoints, though of course all statements apply equally to the white reconstruction paths with appropriate substitutions.

We note that two different reconstruction paths can yield the same block reconstruction, and this is illustrated in Figure 3.2. However, we adopt the convention that shortest path describing a given reconstruction is the one considered, e.g., as many diagonal edges are used as possible, further noting that the shortest such path will in general differ according to whether we are considering the 4 pt. or 8 pt. topologies. For example, in the 8 pt. topology, a horizontal edge followed by a vertical edge is sometimes equivalent to a diagonal edge, whereas in the 4 pt. topology the horizontal, then vertical, edge combination is minimal.

The coordinates of the black run endpoints are $r_1^b = (k_1^b, l_1^b)$ and $r_2^b = (k_2^b, l_2^b)$ and from this we define

$$\Delta_{\mathrm{V}} = |k_2^b - k_1^b|, \ \Delta_{\mathrm{H}} = |l_2^b - l_1^b|$$

to be the vertical and horizontal differences, respectively. The *major* and *minor*



(a) (b) (c)

Figure 3.2: (a) Black reconstruction path; (b) Equivalent black path with adjacent white path in 4 pt. topology; (c) Equivalent black path with adjacent white path in 8 pt. topology.

differences are then defined as

$$\begin{aligned} \Delta_{\max} &= \max\{\Delta_{\mathrm{V}}, \Delta_{\mathrm{H}}\} \\ \Delta_{\min} &= \min\{\Delta_{\mathrm{V}}, \Delta_{\mathrm{H}}\}. \end{aligned}$$

Note that the quantities $\Delta_{\max}$ and $\Delta_{\min}$ will differ according to whether the white or black run is being considered. A *simple* path between endpoints r_1^b and r_2^b is a shortest path connecting the two points, one in which each step or edge of the path decreases the distance to the endpoint.

3.2.3.1 4-pt. Topology

In the 4 pt. topology, simple paths consists of horizontal and vertical edges and as such each simple path has $\Delta_{\max} + \Delta_{\min}$ edges. There will be $\Delta_{\max}$ edges in the direction of major difference and $\Delta_{\min}$ edges in the direction of minor difference and one see that there are $\binom{\Delta_{\max}+\Delta_{\min}}{\Delta_{\min}}$ simple paths connecting the endpoints. The theorem below characterizes the MAP reconstruction paths for one-run boundaries in the 4 pt. model.

Theorem III.2 (4 pt. one-run-optimal paths)**.** *Let r^b be the black run on the boundary and let Δ_{max} and Δ_{min} be the major and minor differences of this run. Also, let c denote the number of corners of δB contained in r^b. The following characterize the*

set of MAP reconstructions determined by MAP paths connecting r_1^b and r_2^b.

(a) If $c = 0$, the one and only MAP path consists of the run itself.

(b) If $c = 1$, the $\binom{\Delta_{\max}+\Delta_{\min}}{\Delta_{\min}}$ simple paths connecting r_1^b and r_2^b generate all $\binom{\Delta_{\max}+\Delta_{\min}}{\Delta_{\min}}$ distinct MAP reconstructions.

(c) If $c = 2$, the $\binom{\Delta_{\max}+\Delta_{\min}-1}{\Delta_{\min}}$ simple paths connecting r_1^b and r_2^b that exclude r^w generate all $\binom{\Delta_{\max}+\Delta_{\min}-1}{\Delta_{\min}}$ distinct MAP reconstructions.

(d) If $c = 3$, the $\binom{\Delta_{\max}+\Delta_{\min}-2}{\Delta_{\min}-1}$ simple paths connecting r_1^b and r_2^b that do not intersect r^w generate all $\binom{\Delta_{\max}+\Delta_{\min}-2}{\Delta_{\min}-1}$ distinct MAP reconstructions.

(e) If $c = 4$, the one and only MAP path is not simple: if the direction of maximum distance is vertical, then the first and final steps are horizontal with $\Delta_{\max}$ vertical steps in between; if the direction of maximum distance is horizontal, the the first and last steps are vertical with $\Delta_{\max}$ horizontal steps in between.

(f) The number of odd bonds in a MAP reconstruction is

$$O_1^*(r^b) = \begin{cases} \Delta_{\max} + \Delta_{\min} - c + 3, & 0 \le c \le 3 \\ \Delta_{\min} + 1, & c = 4 \end{cases}$$

The reconstruction path that together with the run encloses the most sites is called the *outer* path, while the run that encloses the fewest sites, together with the run, is the *inner* path. In Figure 3.3 we can see some MAP reconstructions for different block boundaries. In (c) the black run contains no corners and the MAP path is the run itself. In (a) and (d) we see different reconstructions for the same boundary. Note the range of equiprobable MAP reconstructions. In (b) and (e) we see MAP reconstructions for a given boundary containing two corners. In both of these, the first step is constrained to be vertical so as not to intersect the white run. In (f) we see a MAP reconstruction for a boundary containing three corners. For this boundary, the first step is constrained to be vertical and the last step is constrained

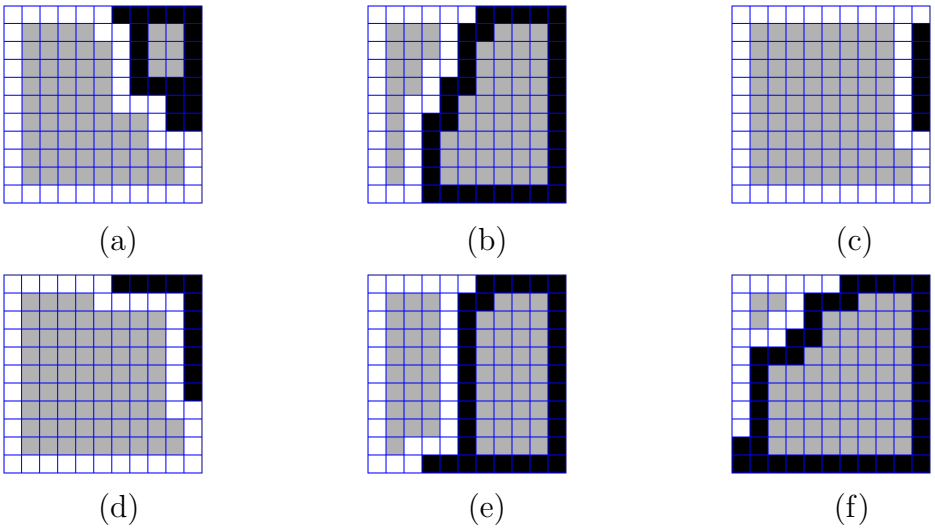


Figure 3.3: Different one-run boundaries and MAP reconstruction paths in the 4-pt. topology.

to be horizontal.

3.2.3.2 8-pt. Topology

In the 8 pt. model, simple paths between r_1^b and r_2^b consist of $\Delta_{\max}$ edges where $\Delta_{\max} - \Delta_{\min}$ of the edges are either vertical or horizontal, in the direction of major difference, and $\Delta_{\min}$ of the edges are diagonal. There are $\binom{\Delta_{\max}}{\Delta_{\min}}$ such simple paths connecting the endpoints r_1^b and r_2^b. Note that simple paths in the 8 pt. model consist of fewer steps to connect the same points. This is because a diagonal edges is used in place of successive horizontal-vertical or vertical-horizontal combinations.

Theorem III.3 (8pt. one-run-optimal paths)**.** *Let r^b be the black run on the boundary and let Δ_{max} and Δ_{min} be the major and minor differences of this run. Also, let c denote the number of corners of δB contained in r^b. The following characterize the set of MAP reconstructions determined by MAP paths connecting r_1^b and r_2^b.*

(a) Every MAP path for r^b has Δ_{max} edges.

(b) If $0 \leq c \leq 2$, the $\binom{\Delta_{max}}{\Delta_{min}}$ simple paths connecting r_1^b and r_2^b generate all $\binom{\Delta_{max}}{\Delta_{min}}$ distinct MAP reconstructions.

(c) If $c = 3$, the $\binom{\Delta_{max}-1}{\Delta_{min}-1}$ simple paths connecting r_1^b and r_2^b that do not intersect r^w generate all $\binom{\Delta_{max}-1}{\Delta_{min}-1}$ distinct MAP reconstructions.

(d) If $c = 4$, the one and only MAP path is not simple: it begins with a diagonal edge, ends with the other type of diagonal edge, and in between it has $\Delta_{max} - 2$ edges that are all vertical or all horizontal.

(e) The number of odd bonds in a MAP reconstruction is

$$O_1^*(r^b) = \begin{cases} 3\Delta_{max} + \Delta_{min} - 2c + 5, & 0 \leq c \leq 3 \\ 3\Delta_{min} - 1, & c = 4 \end{cases}$$

A MAP path will be termed *one-run-optimal* for r^b. We reiterate that the above theorems hold for the white run r^w as well, with a different $\Delta_{\max}$ and $\Delta_{\min}$, as well as a different number of corners. For instance, if r^b contains three corners, the above theorem could be applied to r^w using the different $\Delta_{\max}$ and $\Delta_{\min}$ and letting $c = 1$. In this way, it is always possible to describe a MAP reconstruction as being determined by a MAP path connecting the endpoints of a run containing two or fewer corners. However, Theorems III.2 and III.3 as stated will be useful in the following sections.

In Figure 3.4 we see some MAP reconstructions for different one-run boundaries.

3.2.4 Four Odd Bonds in Boundary

This is the case of four odd bonds in the boundary, or two runs of black and two runs of white. Let $\underline{r} = (r^{b,1}, r^{b,2}, r^{w,1}, r^{w,2})$ be the four runs on the boundary. Unless otherwise specified we will use a clockwise ordering of the endpoints of the runs so that, for example, the endpoints of run $r^{b,1}$ will be denoted as $r_1^{b,1}$ and $r_2^{b,1}$. We let $O_2^*(\underline{r})$ be the number of odd bonds in a MAP reconstruction for the boundary with runs $\underline{r}$. We will analyze the case of a two-run boundary by attempting to decompose the problem into one-run problems. For example, for run $r^{b,1}$, we will consider the

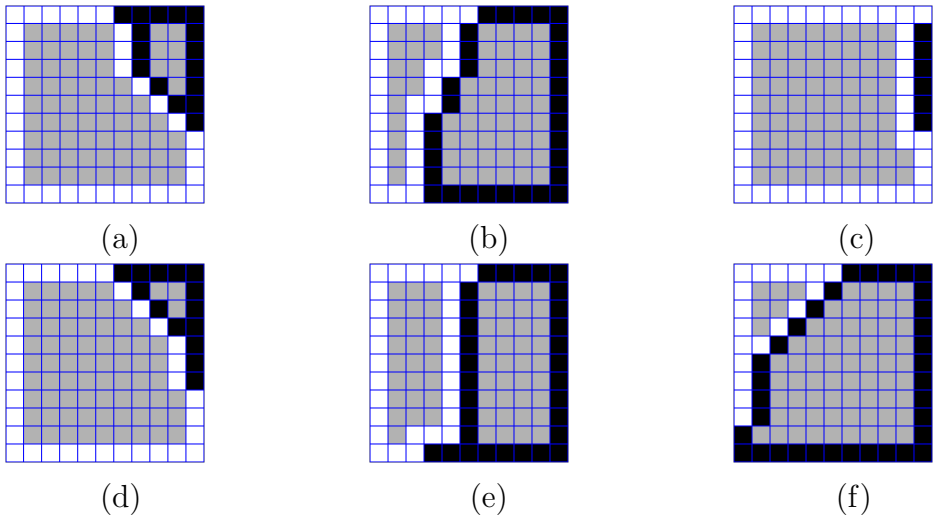


Figure 3.4: Different one-run boundaries and MAP reconstruction paths in the 8-pt. topology.

boundary with black run $r^{b,1}$ and the rest of the boundary white, and look at one-run optimal reconstructions for such a boundary. Recall from the last section that $O_1^*(r^{b,1})$ is the number of odd bonds in a MAP reconstruction for that boundary, determined by a one-run optimal path.

Looking at the two black runs $r^{b,1}$ and $r^{b,2}$, suppose that we connect the endpoints of $r^{b,1}$ with a one-run-optimal path and do the same for $r^{b,2}$. We now define the quantity

$$O_2(r^{b,1}, r^{b,2}) \triangleq O_1^*(r^{b,1}) + O_1^*(r^{b,2}),$$

which is the number of odd bonds in a reconstruction determined by these two one-run-optimal paths *provided they do not touch*. If these two one-run-optimal paths touch, then the number of odd bonds in the reconstruction would be lower than $O_2(r^{b,1}, r^{b,2})$. Let $B_{b,1}$ be the set of sites enclosed (inclusively) by the outer path for $r^{b,1}$ and likewise for $B_{b,2}, B_{w,1}$, and $B_{w,2}$. If $B_{b,1}$ and $B_{b,2}$ are not adjacent (i.e., there are no edges directly connecting them), then any pair of one-run-optimal paths for $r^{b,1}$ and $r^{b,2}$ give a white-HV reconstruction with $O_2(r^{b,1}, r^{b,2})$ odd bonds that is the best white-HV reconstruction.

Let $O_{b-HV}(\underline{r})$ be the minimum number of odd bonds of all reconstructions in which the black sites $\bar{B}^b$ are HV connected, and similarly for $O_{w-HV}(\underline{r})$ as the minimum number of odd bonds of all reconstructions in which the white sites $\bar{B}^w$ are HV connected. Let $O_{wid}(\underline{r})$ be the minimum number of odd bonds from all reconstructions with a single widget and no islands.

3.2.4.1 4-pt. Topology

We now discuss the set of block MAP reconstructions for boundaries in the 4 pt. model with two runs of black and two runs of white.

Lemma III.4. *In the 4 pt. model, $\bar{B}_b$ is connected if and only if $\bar{B}_w$ is disconnected.*

Proof. Assume that $\bar{B}_b$ is connected and assume that $r^{w,1}$ comes after $r^{b,2}$ and before $r^{b,1}$ in a clockwise ordering of the runs. Let $B'_{w,1}$ denote the set of white pixels connected to $r^{w,1}$. Since $\bar{B}_b$ is connected there is a path from the first endpoint $r_1^{b,1}$ of the first black run to the second endpoint $r_2^{b,2}$ of the second black path. If we hypothetically flip the white run $r^{w,1}$ from white to black, then we have a black loop, adjoining the original path from $r_1^{b,1}$ to $r_2^{b,2}$ with the new black path through $r^{w,1}$. Therefore, the sites in $B'_{w,1}$ are contained in a black loop, thus are not connected to $r^{w,2}$. □

For the 4 pt. model, there are four possible scenarios:

(i) black connected: the set $\bar{B}_b$ is connected and the set $\bar{B}_w$ is disconnected,

(ii) white connected: the set $\bar{B}_w$ is connected and the set $\bar{B}_b$ is disconnected,

(iii) disconnected: both sets $\bar{B}_b$ and $\bar{B}_w$ are disconnected.

Lemma III.5. *In the 4 pt. model, there can be at most one interior widget in a MAP reconstruction.*

We now characterize the sets of MAP reconstructions for two-run boundaries in the 4 pt. model.

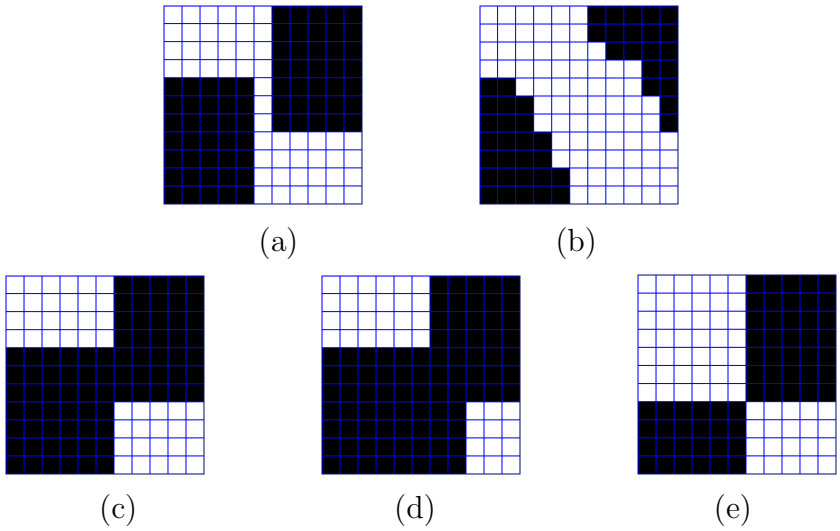


Figure 3.5: Reconstructions for different two run boundaries in 4pt. model.

Theorem III.6 (4 pt. Two Run Optimal)**.** *Let $x_{\partial B}$ be a boundary configuration with runs $\underline{r} = (r^{b,1}, r^{b,2}, r^{w,1}, r^{w,2})$. Then,*

(a) $O^*(\underline{r}) = \min\{O_2(r^{b,1}, r^{b,2}), O_2(r^{w,1}, r^{w,2})\}$

(b) If $O^*(\underline{r}) = O_2(r^{b,1}, r^{b,2})$, then merging any one-run-optimal reconstruction for $r^{b,1}$ with any one-run-optimal reconstruction for $r^{b,2}$ yields a MAP reconstruction given $x_{\partial B}$.

(c) If $O^*(\underline{r}) = O_2(r^{w,1}, r^{w,2})$, then merging any one-run-optimal reconstruction for $r^{w,1}$ with any one-run-optimal reconstruction for $r^{w,2}$ yields a MAP reconstruction given $x_{\partial B}$.

Again from Theorem III.6 we can note the wide range of equiprobable MAP reconstructions for a given boundary in the 4 pt. model. Even for boundaries where neither pair of $B_{b,1}$ and $B_{b,2}$ or $B_{w,1}$ and $B_{w,2}$ intersect or are adjacent, reconstructions involving two large black rectangles in the interior and the reconstruction with an all white interior are equally MAP. A wide range of MAP reconstructions is especially true for block boundaries where the sets $B_{b,1}$, $B_{b,2}$, $B_{w,1}$, and $B_{w,2}$ all meet at a widget. In this case the number of odd bonds from white one-run-optimal paths is the same as that from black one-run-optimal paths. This means that white HV

connected reconstructions determined by black one-run-optimal paths and black HV reconstructions determined by white one-run-optimal paths have the same number of odd bonds and therefore are equally likely. In particular, the reconstructions with all white interior and all black interior are both MAP and equally likely. These are likewise as equally likely as the reconstruction with the widget.

3.2.4.2 8-pt. Topology

In the 8 pt. model, there are diagonal edges so the presence of a widget would reduce the number of odd bonds due to the diagonal edges being even. Whereas in the 4 pt. model, the sets $B_{b,1}$ and $B_{b,2}$ can be HV connected or disconnected, in the 8 pt. model the possible cases are:

(i) the set $\bar{B}_b$ is HV-connected and the set $\bar{B}_w$ is disconnected,

(ii) the set $\bar{B}_w$ is HV-connected and the set $\bar{B}_b$ is disconnected,

(iii) the sets $\bar{B}_b$ and $\bar{B}_w$ are each 8-connected; in this case the block reconstruction is said to be *bi-connected*.

Note that in a bi-connected reconstruction there must be a widget. However, we can narrow things down a bit in terms of where such a widget may be located in a MAP reconstruction in the 8 pt. model.

Lemma III.7. *In the 8-pt. model, if there is a widget in a MAP reconstruction, then the widget is on the boundary.*

With this information and the notation for the endpoints of the boundary runs, if there is a widget on the boundary, it must contain one of the following edges:

$$\{r_2^{w,1}, r_1^{b,1}\},\ \{r_2^{b,1}, r_1^{w,2}\},\ \{r_2^{w,2}, r_1^{b,2}\},\ \{r_2^{b,2}, r_1^{w,1}\} \tag{3.1}$$

Theorem III.8. *If x_B^* is a bi-connected MAP reconstruction, then x_B^* has exactly one widget on its boundary.*

The above lemma and theorem make it possible to consider only four possible locations for a boundary widget to occur.

Theorem III.9 (8 pt. Two Run Optimal)**.** *Let $x_{\partial B}$ be a boundary configuration with runs $\underline{r} = (r^{b,1}, r^{b,2}, r^{w,1}, r^{w,2})$. Then,*

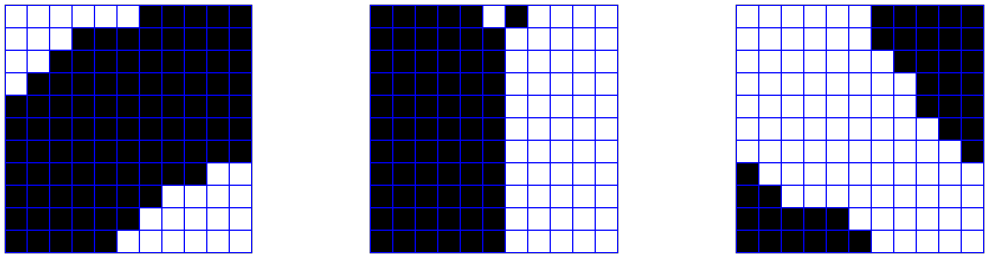
(a) $O^*(\underline{r}) = \min\{O_2(r^{b,1}, r^{b,2}), O_2(r^{w,1}, r^{w,2}), O_{wid}(\underline{r})\}$

(b) If the sets $B_{b,1}$, $B_{b,2}$, $B_{w,1}$, and $B_{w,2}$ meet at a boundary widget, then the reconstruction determined by these sets is MAP given $x_{\partial B}$.

(c) If the sets $B_{b,1}$, $B_{b,2}$, $B_{w,1}$, and $B_{w,2}$ do not meet at a widget, then

(i) If $O^*(\underline{r}) = O_2(r^{b,1}, r^{b,2})$, then merging a one-run-optimal reconstruction for $r^{b,1}$ with a one-run-optimal reconstruction for $r^{b,2}$ yields a MAP reconstruction of the interior given $x_{\partial B}$.

(ii) If $O^*(\underline{r}) = O_2(r^{w,1}, r^{w,2})$, then the merging of a one-run-optimal reconstruction for $r^{w,1}$ with a one-run-optimal reconstruction for $r^{w,2}$ yields a MAP reconstruction of the interior given $x_{\partial B}$.



(a) (b) (c)

Figure 3.6: Some block reconstructions for two run boundaries in 8 pt. model.

3.2.5 Procedure for MAP Decoding

We now give an outline for performing MAP Decoding.

(i) If block has no runs, decode interior as same color as boundary

(ii) If block has one run, follow Theorem III.2 or Theorem III.3.

(iii) If block has two runs, follow Theorem III.6 or Theorem III.9.

(iv) If block has more than two runs, consider the two longest runs and apply (iii). This is a suboptimal but reasonable approach. Note that this is asymmetric in the black and white runs.

3.3 BP Block MAP Solutions

In this section we consider Max-Product LBP as an algorithm for estimating X_B conditioned on boundary configuration $x_{\partial B}$, and use the analytical Block MAP solutions of the previous section to evaluate its performance. The probability of a block configuration x_B conditioned on a boundary configuration $x_{\partial B}$ is a function of the number of odd bonds within the joint configuration $(x_B, x_{\partial B})$. The components of the Max-Product belief at a given site in B are therefore going to be functions of the number of odd bonds under some subset of configurations. It will be convenient in this chapter to indicate sites in B using bivariate coordinates (i, j) where i is the row of B the site is in and j is the column. For site $(i, j) \in B$, we let

$$\begin{aligned} o^*_{ij}(-1) &= \min_{\substack{x_B: \\ (x_B)_{ij}=0}} o(x_B, x_{\partial B}) \\ o^*_{ij}(1) &= \min_{\substack{x_B: \\ (x_B)_{ij}=1}} o(x_B, x_{\partial B}) \end{aligned}$$

be the minimum number of odd bonds under configuration on B, conditioned on $x_{\partial B}$, where site (i, j) has value -1 or 1, respectively. Therefore, using (2.5), the Max-

Product belief vector Q_{ij}^{MP} for site (i,j) is proportional to

$$Z_{ij}^{\mathrm{MP}} = \begin{bmatrix} \exp\{-2\theta o_{ij}^*(-1)\} \\ \exp\{-2\theta o_{ij}^*(1)\} \end{bmatrix} \tag{3.2}$$

$$\propto \begin{bmatrix} 1 \\ \exp\{-2\theta(o_{ij}^*(1) - o_{ij}^*(-1))\} \end{bmatrix} \tag{3.3}$$

$$= \begin{bmatrix} 1 \\ \exp\{-2\theta o_{ij}^*\} \end{bmatrix}, \tag{3.4}$$

where we let

$$o_{ij}^* = o_{ij}^*(1) - o_{ij}^*(-1)$$

be the difference between these two minimum numbers of odd bonds.

We can think of Loopy Belief Propagation, then, as computing for each $(i,j) \in B$ a sequence of estimates $\{o_{ij}^{(n)}\}$ of the quantity o_{ij}^*. Figure 3.7 shows the sequence of estimates $\{o_{ij}^{(n)}\}$ for all sites within a block of the 4 pt. grid graph, for different values of n. The red squares indicate negative values. We can use the results of the previous section, in particular Theorem III.2, to state what the true o_{ij}^* values should be and use these to compare to the empirical values shown in Figure 3.7.

Theorem III.10. *Let B be a block of sites in the 4 pt. model. Let boundary configuration $x_{\partial B}$ be given with runs $\underline{r} = (r^b, r^w)$. Furthermore, let B_b be the set of sites enclosed, inclusively, by the run r^b and the outer path for it, and similarly for B_w. For site $(i,j) \in B$ the quantity o_{ij}^* is determined as follows*

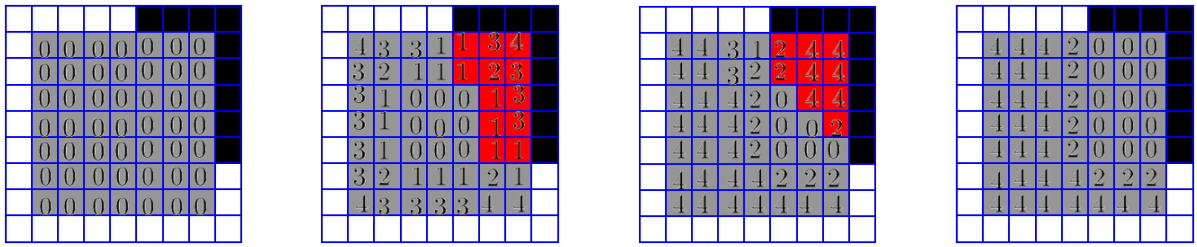
if $(i,j) \in B_w \cap B_b$, then $o_{ij}^* = 0$;

if $(i,j) \in B_w \setminus B_b$ and $(i,j) \in \partial B_b$, then $o_{ij}^* = 2$;

if $(i,j) \in B_b \setminus B_w$ and $(i,j) \in \partial B_w$, then $o_{ij}^* = -2$;

if $(i,j) \in B_w \setminus (B_b \cup \partial B_b)$, then $o_{ij}^* = 4$;

if $(i,j) \in B_b \setminus (B_w \cup \partial B_w)$, *then* $o^*_{ij} = -4$.



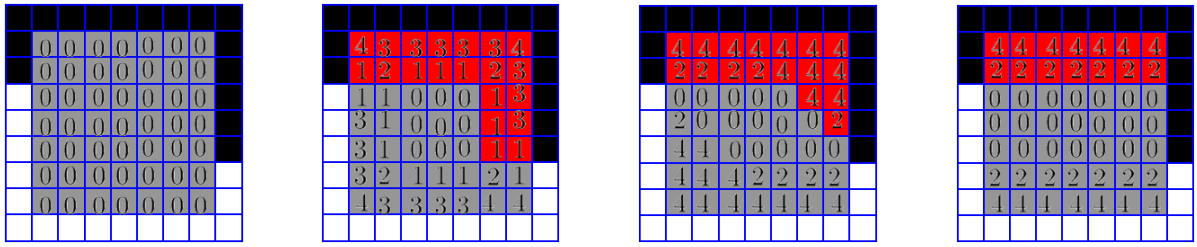
(a) (b) (c) (d)

Figure 3.7: Block of $\{o^{(n)}_{ij}\}$ values after (a) zero; (b) two; (c) six; (d) and ten iterations.

Referring back to Figure 3.7, we can compare against Theorem III.2 and see that after ten iterations Max-Product LBP converges to the correct collection of values $\{o^*_{ij}\}$. In particular, we see that for sites (i,j) that are enclosed by the black outer path, that $o^*_{ij} = -1$. This is because these sites are also enclosed by the white outer path, which means that for each of these sites (i,j) there is a MAP reconstruction in which each this site has value -1 and another MAP reconstruction in which this site has value 1. Therefore $o^*_{ij}(-1)$ and $o^*_{ij}(1)$ will be equal for such sites. In Figure 3.7 we see that in the early iterations of the algorithm, the sites nearest the black run have negative values, which means that at these early stages of the LBP run, the estimates of their Max-Product beliefs $Q^{(n)}$ are biased towards 1 (black). This is because after only a few iterations each site has only received "information" from the boundary sites closest to them. Also, we can see that those sites (i,j) that are not enclosed by the black outer path have positive o^*_{ij} values. This is because in every MAP reconstruction these sites have value 0.

In Figure 3.8 we show the block updates for a different one run boundary and again see that Max-Product LBP converges after nine iterations. In this case, neither the black outer path nor white outer path encloses all sites in the block interior, so that some o^*_{ij} values are positive and some are negative. We can see in (b) and (c) that before convergence of the algorithm, sites closest to the black run have estimates $o^{(n)}$

that are negative in value, and those that are closest to the white run have estimates $o^{(n)}$ that are positive. We again see in (d) that those sites that are enclosed by both the white and black outer paths have o^*_{ij} values of 0.



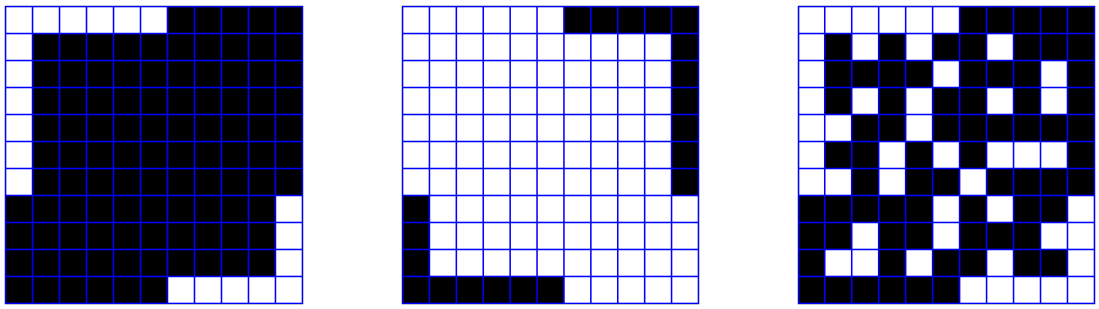
(e) (f) (g) (h)

Figure 3.8: Block of $\{o^{(n)}_{ij}\}$ values after (a) zero; (b) two; (c) six; (d) and nine iterations.

We tried a number of cases of block boundaries of one and two runs in both the 4 pt. and 8 pt. models and found that after a reasonable number of iterations (no more than twice N) the algorithm would converge to the correct values. We know that the answers are correct for boundaries with one and two run because of Theorem III.10, which followed from the results of the last section. We make the following conjecture.

Conjecture III.11. *In the 4 pt. and 8 pt. homogeneous Ising models, Max-Product BP converges to the correct values* $\{o^*_{ij}\}$ *in the boundary interpolation problem.*

If this conjecture is correct, it would seem that Max-Product LBP would be an ideal choice for an interpolation algorithm for homogeneous Ising models, in that it would be both efficient and correct. It is also natural, given the distributed nature of the algorithm, for it to be employed in estimation problems that are truly distributed, in the sense that decisions are made at individual nodes independently of the decisions at other nodes. That is, in estimation problems where there is no centralized observer that can jointly decide the values of multiple sites. For instance, if $o^*_{ij} > 0$, then the site (i, j) can decode itself as a 1 (black), and if $o^*_{ij} < 0$, as a -1 (white). If $o^*_{ij} = 0$, however, there are MAP configurations with (i, j) as -1 and as 1. If the



(a) (b) (c)

Figure 3.9: Reconstructions for different two run boundaries in 4pt. model.

estimation decisions are made independently at each node (i, j) for which $o^*_{ij} = 0$, then the resulting block estimate might not be MAP, since the collection of values $\{x_{(i,j)} : o^*_{ij} = 0\}$ may not appear in any MAP configuration. In Figure 3.9 (a) and (b) we show two MAP configurations for a given boundary in the 4 pt. model. Because there is a MAP configuration in which each site can be 0 or a 1, $o^*_{ij} = 0$ for all (i, j). Thus independent estimation at each site could produce a non-MAP configuration as in (c). We note that there will always be sites within a block for which the o^* value is 0. We can see this by considering a one run boundary and the MAP configuration determined by the black outer path. This reconstruction is also determined by the white inner path. Since the white outer path encloses more sites than the white inner path, it must enclose sites that are also enclosed by the black outer path, and as proved by Theorem III.10 and seen in Figures 3.7 and 3.8, these sites have o^* values of 0.

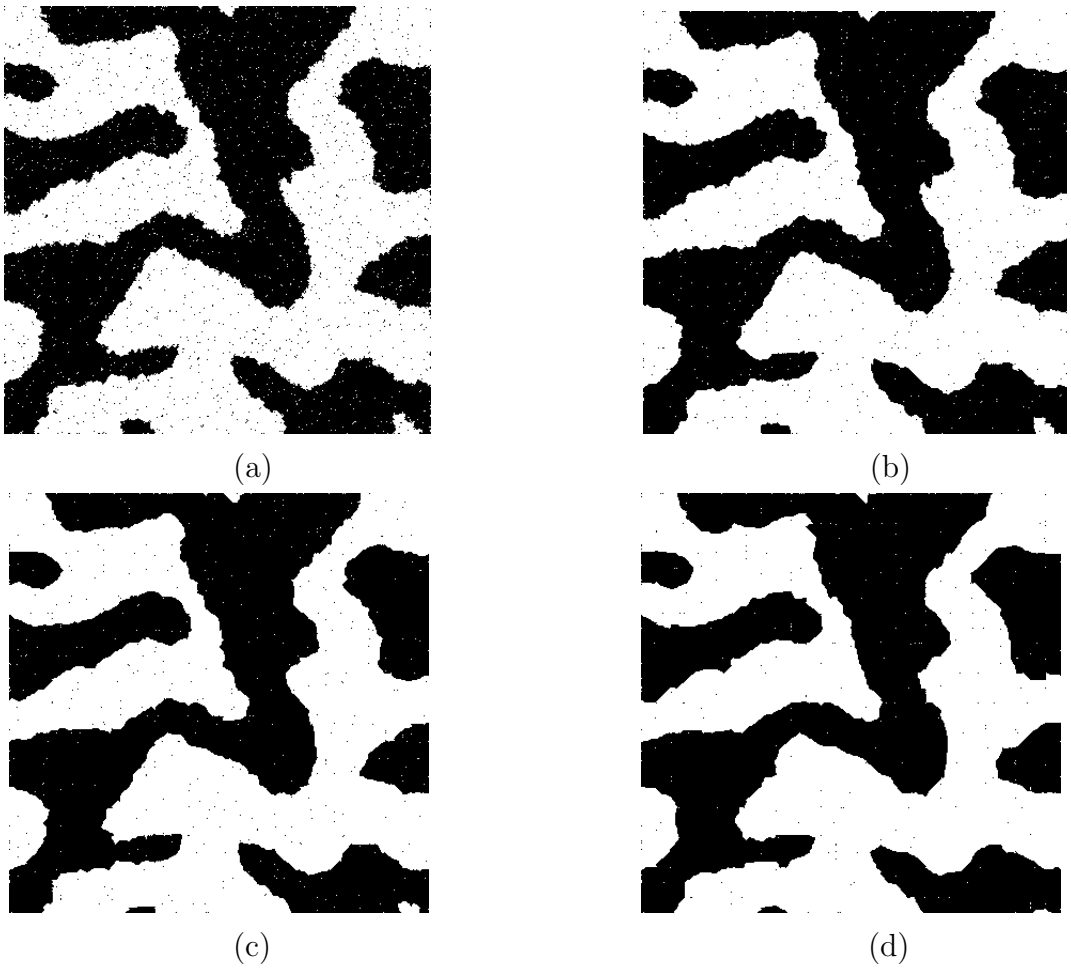


Figure 3.10: (a) Typical image; (b) - (d) MAP reconstructions.

3.4 Experimental Testing of MAP Reconstructions on Whole Images

In this section we look at MAP reconstructions for a few different images using the analytical solutions of Section 3.2. Before looking at specific image reconstructions, we give a few words on how one might expect the reconstruction quality of the lossy Cutset Coding algorithm to depend on different input images. Exact solutions were given in Section 3.2 for block boundaries with zero, one or two runs of black (alternatively, white) pixels.

A block boundary will be more likely to have fewer than three runs if the original

image is smooth, in the sense that there are large regions of white and large regions of black that meet at given contours within the image. Since for blocks of three or more runs we resort to a sub-optimal ad-hoc approach, we expect the algorithm to give reconstructions that are closer to optimal for smooth images, as there will be more block boundaries for which the exact solutions can be applied. Another issue affecting reconstruction quality is the size of the blocks being interpolated. If the original block interior contains an island, then this will clearly be lost in a MAP reconstruction of the interior, by Lemma III.1. For small- to moderate-sized blocks, there is less chance of the original interior containing an island, so block interpolations will "miss" potentially important image information such as islands less frequently for smaller blocks. This difference between large and small blocks is especially true for less smooth images as it is more likely that these will have "activity" within a block.

For simulated, i.e., typical, images from a homogeneous Ising distribution, we can see the relationship between image smoothness and reconstruction quality by varying the parameter θ. In Figure 3.10 (a) we see an original typical image from a homogeneous Ising distribution on the 8 pt. grid with parameter $\theta = .2875$. We are interested in this value of θ because using a Gibbs Sampling based method of parameter estimation [23, 15], this was the estimate obtained for the real-world image in Figures 2.2 (a) and 3.13 (a) modeling it as a homogeneous Ising model on the 8 pt. grid graph.

We see that the image in (a) is fairly smooth with a small amount of speckle. In Figure 3.10 (b), (c) and (d) we see the reconstruction using interior block sizes $N-1 = 4$, 7, and 11. We see that as the block size increases, the reconstructions has less speckle. This is because the block MAP reconstructions will not contain islands, so the only speckle that will appear in a reconstruction is that which occurs on the perfectly preserved grid cutset. For larger block sizes, the grid contains fewer pixels

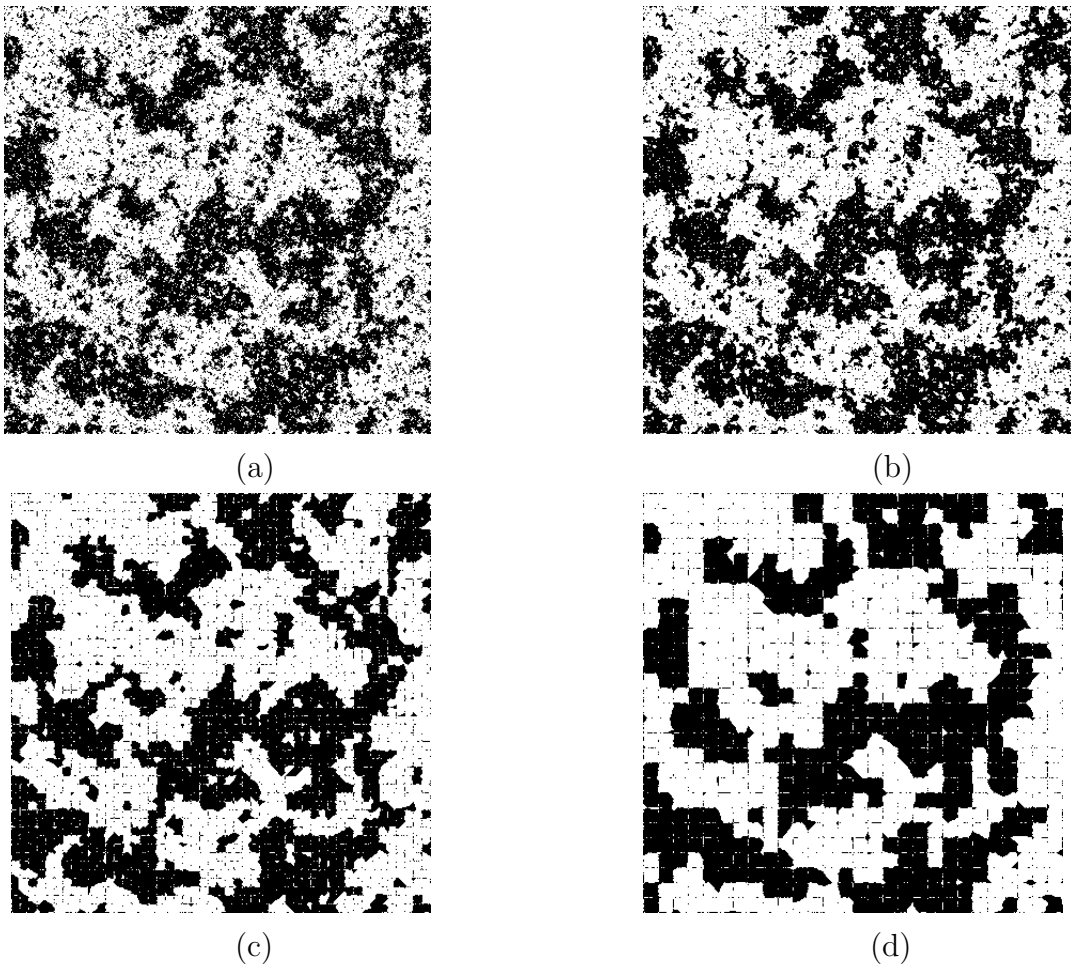


Figure 3.11: (a) Typical image; (b) - (d) MAP reconstructions.

and the grid pixels that are preserved farther apart.

In Figure 3.11 (a) we show typical image from a homogeneous Ising model on the 8 pt. grid graph with exponential parameter $\theta = .1875$. This is clearly a less smooth image so we would expect the reconstructions from lossy cutset coding to be less faithful. In (b), (c), and (d) we see reconstructions with interior block sizes $N-1 = 2$, 7, and 13. We see that the reconstructions are less accurate in reproducing the image in (a). Because there is much more speckle or islands in the original image, much more is preserved on the grid cutset, which is apparent in the reconstructions (b), (c), and (d) in the form of dotted horizontal and vertical lines throughout the reconstruction. Moreover, the contours that cut through the original blocks are less

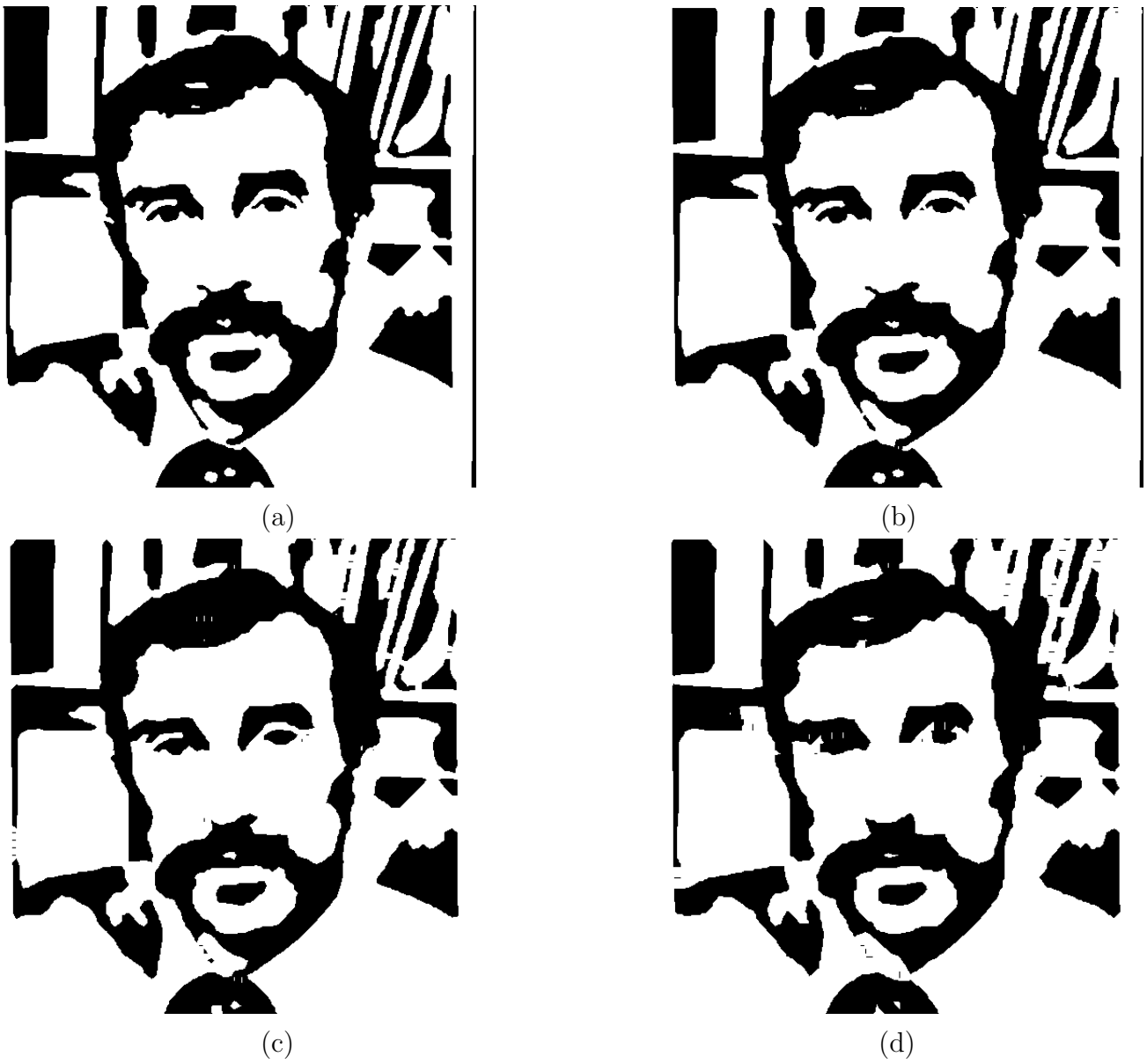


Figure 3.12: (a) Original image; (b) - (d) MAP reconstructions.

smooth than those produced by MAP decoding, so the reconstructed blocks appear much smoother than the original blocks. This is especially true for the reconstruction in (d).

In Figure 3.12 (a) we see a smooth binary real-world image. In (b), (c), and (d) we see reconstructions using the lossy cutset coding method with interior block sizes $N-1 = 4$, 7, and 11. In Figure 3.13 (a) we see a less smooth binary real-world image.

Figure 3.13: (a) Original image; (b) - (d) MAP reconstructions.

In (b), (c), and (d) we see reconstructions using the lossy cutset coding method with interior block sizes $N - 1 = 4$, 7, and 11. Note that in block MAP decoding, the parameter value θ does not matter, so it is independent of a parameter estimate.

(a) (b)

Figure 3.14: (a) MAP reconstruction; (b) reconstruction with decision-bits.

3.5 Decision-bit Encoding/Decoding

We can see in the reconstructions in Figures 3.12 (c) and 3.13 (c) that regions of the hair, the collar, and lines in the upper right of the image are broken up by the MAP decoder. The reason for this is simple: the blocks in these regions have two or more runs on their boundaries. With a block boundary that has two runs of black and two runs of white, there are two main patterns that can be reconstructed, one with a white swath through the block, the other with a black swath. In Figure 3.13 (c), for example, in the lines at the upper right of the image, it is possible that for a few of the blocks in this region, while the original block had a black swath through the interior, the MAP estimate shown has a white swath. Since perceptual quality is the most important criterion for a decoding algorithm, we need to correct this error.

Fixing this problem is rather simple: for a block with two runs of black and two runs of white we need to transmit to the decoder an extra bit indicating which of the two general reconstruction patterns to pursue. During the decoding process, when the decoder observes that the current block has two runs of black and two runs of

white, it will access the string of *decision bits* for the bit corresponding to the current block, and then interpolate correctly. There are various ways to assign the decision bits. The decision bit rule that is currently being implemented is that when the encoder encounters a block with two runs of black and two runs of white, it finds the block reconstruction determined by the inner paths connecting the endpoints of the black runs and it also find the reconstruction determined by the inner white paths. It computes the Hamming distortion between each of these reconstructions and the original block and transmits a bit corresponding to the one having the minimum distance to the original block. For blocks of three or more runs, we follow the following ad-hoc procedure: for each color, we consider the two longest runs and change all other runs of this color to the opposite color, reconstruct according to the inner paths for this color; and then compare each of these reconstructions to the original. Figure 3.14 (b) shows the reconstruction with interior block size $N - 1 = 7$ using the decision bit reconstruction, while (a) shows the MAP reconstruction with the same block size. We see that the decision-bit decoder produces a much nicer-looking reconstruction. The lines are no longer broken, and in general, the reconstructed image could pass for an original, though there it is obviously not identical to the original image.

3.6 Encoding the Grid Cutset and Rate Analysis

We described in the last section that decision bits are transmitted from the encoder to the decoder for blocks of two or more runs to determine the best reconstruction pattern. The majority of the bits sent from the encoder to the decoder are those for lossless compression of the grid cutset pixels X_U. Figure 3.15 shows a plot of Hamming distortion versus rate of the MAP decoder and decision-bit decoder, along with two competing methods, for the figure in Figure 3.13 (a). The method used to encode the grid is a simple adaptive method based on Arithmetic Encoding. We first consider the subgraph G_U induced by the cutset. Then, for each grid node that is in both a column and a row, but not the top row, we remove the edge connecting it to its left neighbor, if it has one. We then form a tree rooted at the upper left node of the graph. This determines parent-child relations down to the leaves of the tree. We then scan the nodes of this tree left to right in each row, top to bottom from the top row. This is a *lineal* scan in that the parent is always encoded before its children. The coding distributions computed are *first-order* in that the coding distribution for the ith node is a conditional distribution conditioned on the value of the parent node $\pi(i)$. This will of course be sub-optimal since the grid cutset subfield X_U is not really Markov with respect to a tree. The coding distribution for the first node is $f_1(x_1) = 1/2$ and for the ith node, $i = 2, \ldots, |U|$, $f_i(x_i|x_{\pi(i)})$ is computed as

$$f_i(x_i|x_{\pi(i)}) = \frac{\sum_{k=2}^{i=1} \mathbb{I}_{(x_k=x_i, x_{\pi(k)}=x_{\pi(i)})}}{\sum_{k=2}^{i=1} \mathbb{I}_{(x_{\pi(k)}=x_{\pi(i)})}}. \tag{3.5}$$

In other words, the coding distribution used for the ith node of the scan is a conditional distribution given the value of its parent. The conditional distribution used is not computed from the model, but rather from the empirical statistics of the previously scanned grid pixels.

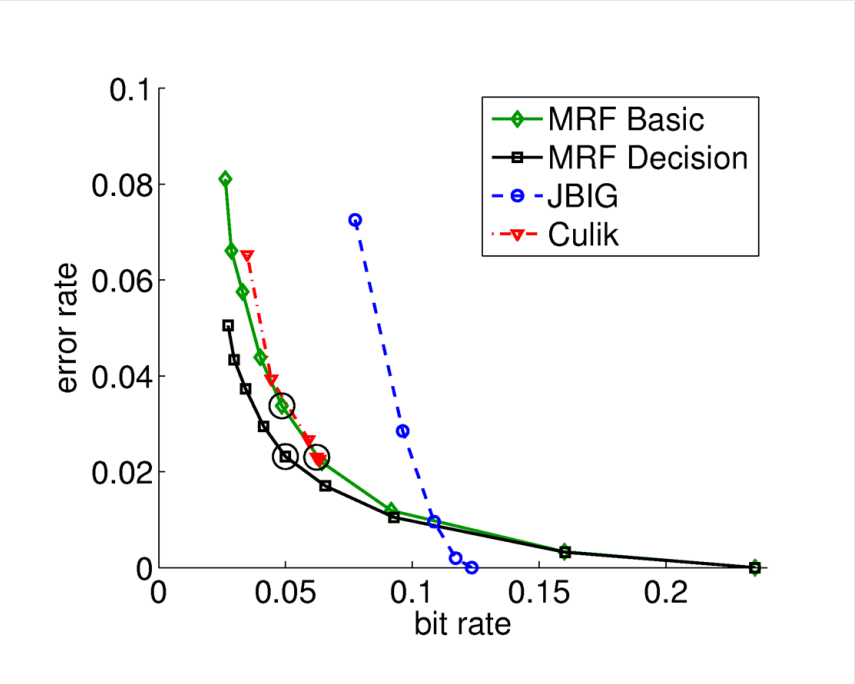


Figure 3.15: Rate vs. Distortion performance of MAP and Decision-bit decoding on image in Figure 3.13 (a).

We see in Figure 3.15 that the decision-bit coder gives better performance than the MAP decoder in that for comparable bit rates, the Hamming distortion is less for the decision-bit decoder. We found that the typical rate R_U for the grid pixels was around .3, and this was fairly constant over different block sizes. Also in Figure 3.15 is the Hamming distortion vs. rate performance of two competing algorithms. The labeled "Culik" in the plot is a finite automata based method [28] and the one labeled "JBIG" is a simulated lossy coder created by concatenating a morphological filter followed by lossless JBIG [27].

The rate of lossy cutset coding can be analyzed further through the recently introduced concept of *erasure entropy* [51, 52]. The erasure entropy $H^-(\mathbf{X})$ is defined as

$$H^-(\mathbf{X}) = \frac{1}{|V|} \sum_{i=1}^{|V|} H(X_i | X_{V \setminus i}),$$

which is the normalized sum of the conditional entropies of each random variable given

the remaining random variables. The interpretation of the i-th term is the average number of bits needed to describe the random variable X_i conditioned on all other variables, or in the context of Markov random fields, conditioned on its neighbors $X_{\partial i}$. For MRFs on a rectangular grid of sites V we can extend this concept to *block erasure entropy* for $N-1 \times N-1$ blocks of sites as

$$H^-_{N-1}(\mathbf{X}) = \frac{1}{|\mathcal{B}|} \sum_{B \in \mathcal{B}} H(X_B | X_{\partial B}), \tag{3.6}$$

where the B_i are $N-1 \times N-1$ blocks and the set $\mathcal{B}$ is the set of all $N-1 \times N-1$ blocks within the grid, containing the upper left $N-1 \times N-1$ block of sites and such that if $B_i \in \mathcal{B}$, then the block obtained by shifting B_i one row down or one column to the right is also in $\mathcal{B}$ so long as it is contained within the rectangular grid of sites. The quantity $H^-_{N-1}(\mathbf{X})$ would of course be the average number of bits required to losslessly encode an $N-1 \times N-1$ block of missing sites conditioned on its boundary.

Now suppose the MRF is a homogeneous Ising model on a grid graph G. In this case, each term of (3.6) is the same except for those blocks B_i that do not have a full boundary in G, i.e., those that intersect either the first or last row or first or last column of G. If G is large enough, the $N-1 \times N-1$ block erasure entropy $H^-_{N-1}(\mathbf{X})$ will be dominated by blocks having a full boundary within G and will roughly equal the conditional entropy $H(X_{B_i} | X_{\partial B_i})$ of such a block given its boundary. If U is a grid cutset with row and column spacing N, each component of $G \setminus U$ is an $N-1 \times N-1$ block of sites with a full boundary in G, thus the conditional entropy of each component of $G \setminus U$ given its boundary is approximately $H^-_{N-1}(\mathbf{X})$. Assuming there are M such blocks in $G \setminus U$, the entropy of the homogeneous Ising model can be expressed as

$$H(\mathbf{X}) = H(X_U) + M H^-_{N-1}(\mathbf{X}). \tag{3.7}$$

which says that the rate saved by not encoding the blocks is at least $MH_N^-(\mathbf{X})$. In Section 6.2 we discuss an algorithm for lossless compression for MRFs and in Section 6.3 apply this to homogeneous Ising models on 4 pt. and 8 pt. grid graphs. As explained in this latter section, the encoding rates on a typical image can be used to obtain a close estimate $\hat{H}(\mathbf{X})$ of $H(\mathbf{X})$. By a very similar method as discussed in Section 6.3, one can obtain a close estimate $\hat{H}_{N-1}^-(\mathbf{X})$ of $H_{N-1}^-(\mathbf{X})$. Therefore, by computing these estimates and plugging them into (3.7), one can get an estimate $\hat{H}(X_U)$ of the entropy $H(X_U)$ of the grid cutset for various block sizes $N-1$. Since the rate of lossy cutset coding is dominated by the rate of lossless coding of the grid cutset, one can then use these estimates and compare versus the rates obtained, for example, in Figure 3.15.

3.7 Proofs

3.7.1 Proof of Lemma III.1

3.7.1.1 8-pt. topology

To prove Lemma III.1 for the 8-pt. model we need the following definition and lemma. Given disjoint sets $U, V \subset \bar{B}$, let $e(U, V)$ denote the number of edges from a pixel in U to a pixel in V.

Lemma III.12. *In the 8-pt. topology, if l is any loop in $\bar{B}$, then*

$$e(I(l), l) \geq e(I(l), E(l))$$

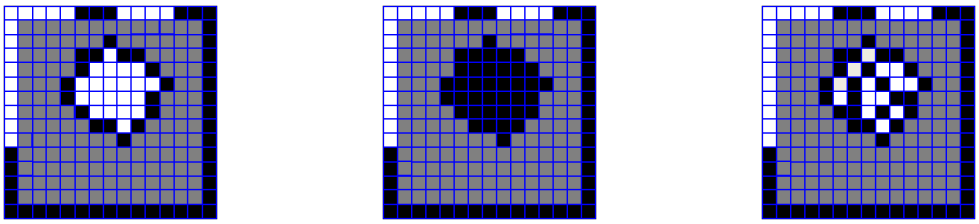
, with equality if and only if l consists of the horizontal and vertical neighbors of a single pixel.

Proof. Let l be an arbitrary loop in $\bar{B}$. First note that each edge from the interior to the exterior of l, counted in $e(I(l), E(l))$, is diagonal. We first observe that if $s \in I(l)$ and t is a neighbor of s in the exterior $E(l)$, then the two neighbors that s and t share are in l. This is because if u were in $\partial s \cap \partial t$ but not in l, then $s - u - t$ would be an HV-path from s to t not intersection l, which when appended to an HV-path from t to ∂B would be an HV-path from s to the boundary not intersecting l. This would contradict the fact that s is in the interior of l. Therefore, u must be in l.

Clearly,

$$e(I(l), E(l)) = \sum_{i \in I(l)} \mid \partial i \cap E(l) \mid .$$

For each i, $\mid \partial i \cap E(l) \mid = 0$, 1, 2, 3 or 4 since the only edges from the interior to the exterior are diagonal. Using the earlier observation, one may directly check that in each of these five cases $\mid \partial i \cap E(l) \mid \leq \mid \partial i \cap l \mid$, with equality only in the cases that $\mid \partial i \cap E(l) \mid = 0$ or $\mid \partial i \cap E(l) \mid = 4$. The former does not contribute to the summation



(a) (b) (c)

Figure 3.16: Black monotone loop with (a) all white interior; (b) all black interior; (c) arbitrary interior.

above, while the latter happens when and only when l consists of the four horizontal and vertical pixels surrounding a single pixel. It follows then that

$$e(I(l), E(l)) \leq \sum_{i \in I(l)} \mid \partial i \cap l \mid = e(I(l), l).$$

Equality holds when and only when for each i, $\mid \partial i \cap E(l) \mid = \mid \partial i \cap l \mid$, which can happen when and only when l consists of the horizontal and vertical neighbors of a single pixel. □

Proof. We now prove Lemma III.1 for the 8 pt. model. Assume that $\hat{x}_B$ is a MAP reconstruction for a boundary $x_{\partial B}$ that contains a monotone loop l, which we can assume without loss of generality to be black. We must show that the interior $x_{I(l)}$ is entirely black. The key fact, shown below, is that an entirely black loop interior is strictly better than an entirely white loop interior, no matter the color of the pixels surrounding the loop. This fact is used to prove the lemma as follows. If there were any white pixels in the interior of the loop l, then there must be a connected set of white pixels in the interior of l surrounded by a monotone black loop l' of pixels within $l \cup I(l)$, whose interior $I(l')$ is the connected set of white pixels. By the key fact, replacing $x_{I(l')}$ with black pixels would strictly decrease the number of odd bonds in $\hat{x}_{\bar{B}}$. Since $\hat{x}_{\bar{B}}$ is MAP, this is not possible. Therefore, there must not be any white pixels in the interior of l, i.e. all pixels in $I(l)$ must be black.

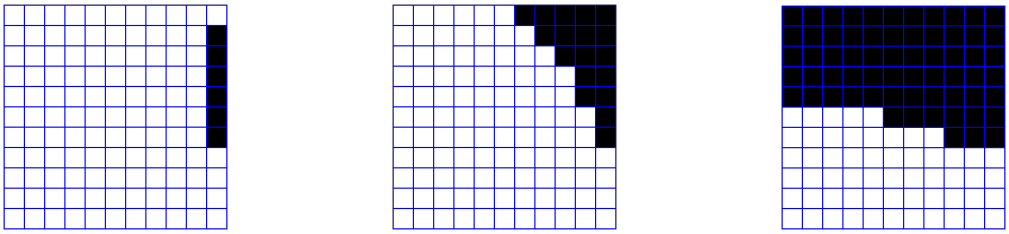
We now prove the key fact that an entirely black interior of a loop l is strictly better than an entirely white interior, no matter the color of pixels in the exterior of l. First note that the number of odd bonds between pixels in the exterior of l is the same regardless of whether the interior is all white or all black. The same is true for the number of odd bonds between the loop and the exterior of the loop. Moreover, for both and all white or an all black interior, the number of odd bonds between sites in the interior is zero. Therefore, the only odd bonds that depend on the color of $I(l)$ are those between the interior and the exterior of the loop, and those between the interior of the loop and the loop itself. Since in Lemma 3.7.1.1 we showed that the number of edges between the interior of the loop to the loop is at least as great as the number of edges between the interior and exterior of the loop, it follows that a smaller number of odd bonds results by making the interior of the loop the same color as the loop itself. Thus, if l does not consist of the vertical and horizontal neighbors of a single pixel, then an entirely black interior is strictly better than an entirely white interior.

On the other hand, if l is the horizontal and vertical neighbors of a single pixel, then the single pixel must be black. For, if one of the diagonal neighbors of this single pixel is black, then it will have more black neighbors than white, and it will therefore be strictly better for it to be black as well. If, on the other hand, all of the diagonal neighbors are white, this creates a widget, which by Lemma III.7 is not possible in a MAP reconstruction. This completes the proof that the interior of a black loop is entirely black in a MAP reconstruction.

Figure 3.16 illustrates the above arguments. □

3.7.1.2 4-pt. topology

Proof. Let l be a monotone black loop. If i is in the interior of l then all of i's neighbors are either also in the interior or are in the loop l. For, if k is some node



(a) (b) (c)

Figure 3.17: Black boundary run containing (a) no corners; (b) one corner; (c) two corners.

in the exterior that is also a neighbor of i, then $i - k$ is an HV-path which, when concatenated with the HV-path from k to the boundary, gives an HV-path from i to the boundary, which would be a contradiction. Since there are no diagonal edges, there are no edges from a node in the loop interior node to a node in the loop exterior. Therefore, an black interior is strictly better than an all white interior. By similar arguments to those in the previous proof, an all black interior is better to an arbitrary interior. □

3.7.2 Proof of Theorem III.2

Suppose $x_{\partial B}$ contains one run of black pixels and one run of white pixels. Without loss of generality we restrict attention to reconstructions generated by the black run and black reconstruction paths. We will prove the theorem by cases, according to the number of corners contained by the black run. Considering a counterclockwise ordering of the endpoints of the black run, we will let r_1^b denote the first endpoint and r_2^b the second endpoint. We will prove the theorem for the different cases by first considering an arbitrary run containing no corners and then successively and arbitrarily extending r_2^b counterclockwise around the block boundary so that each of the corner cases is considered in succession. The statements for each of the corner cases will be proved by beginning with and modifying the conclusions of the previous case. One may wish to consult Figure 3.17 in reading the arguments below.

no corners: If the black run contains no corners, then it is entirely contained within one side of the block, which without loss of generality we assume to be the right side. The simple path consisting of the black run itself results in $\Delta_{\max}+1$ horizontal odd bonds and $\Delta_{\min}+2$ vertical odd bonds. Thus there are $\Delta_{\max}+\Delta_{\min}+3$ odd bonds, as stated in (f). Moreover, there is $\binom{\Delta_{\max}+\Delta_{\min}}{\Delta_{\min}}=1$ such path as stated in (a).

one corner: Beginning with a black run contained within the right side of the block, we note that all odd bonds are vertical or horizontal and to the left of the run. Arbitrarily placing r_2^b on the top of the block so that the black run contains one (the upper right) corner, we prove the theorem for the one case in the following way. There are $\binom{\Delta_{\max}+\Delta_{\min}}{\Delta_{\min}}$ simple paths connecting r_1^b and r_2^b as stated in (a). Furthermore, t is straightforward to see that each of these results in the same number of odd bonds per row and column as in the no corner case, with the following exception. The vertical odd bond between r_2^b and the row above is no longer included. Thus there are $\Delta_{\max}+\Delta_{\min}+3-c$ odd bonds, since $c=1$.

two corners: We now place r_2^b arbitrarily on the left side of the block, so that the resulting run now contains two corners. There are now $\binom{\Delta_{\max}-1}{\Delta_{\min}}$ simple paths connecting r_1^b and r_2^b. We can see this, for example if r_2^b is closer to the top row than r_1^b, because the first step from r_2^b to r_1^b must be horizontal otherwise it would run into a white boundary pixel. This verifies (a) for the case that the run contains two corners. Additionally, the same number of odd bonds per row and column are counted as in the one corner case except that now the vertical odd bond between r_2^b and the row below is not included. Thus, since $c=2$, there are $\Delta_{\max}+\Delta_{\min}+3-c$ odd bonds.

three corners: If the black run contains three corners, then the white run contains one corner. We let $\Delta'_{\max}$, $\Delta'_{\min}$ and c' refer to the white run and note that $\Delta_{\max}=\Delta'_{\max}+1$ and $\Delta_{\min}=\Delta'_{\min}+1$ and $c=c'+2$. We then compute that the number of odd bonds is

$$\Delta'_{\max} + \Delta'_{\min} + 3 - c' = (\Delta_{\max} - 1) + (\Delta_{\min} - 1) + 3 - (c - 2) \quad (3.8)$$

$$= \Delta_{\max} + \Delta_{\min} + 3 - c \quad (3.9)$$

There are now $\binom{\Delta_{\max}-2}{\Delta_{\min}-1}$ because the first step from r_2^b to r_1^b must be horizontal while the last step must be vertical.

four corners: If the black run contains four corners then the white run contains no corners and the claims can be established by considering the no corner case. Note that the resulting MAP path is no simple.

3.7.3 Proof of Theorem III.3

To prove the theorem we will use the following fact, which is easy to verify.

Fact: The number of diagonal odd bonds between two successive rows of pixels is (a) 0, if the two rows are monotone of the same color,

(b) 1, if one row is monotone and all pixels of the other have the same color except one at its end,

(c) 2, if one row is monotone and the other has the same color except for the pixels on each end of the row,

(d) 2, if one row has white pixels to the left, black pixels to the right and the second row is identical to the first, or has the same form except that the transition from white to black occurs one pixel to the left or to the right of the transition in the first row,

(e) at least 3 in all other cases

One can verify the above fact using Figure 3.18.

Now, suppose we have a run of black pixels with major and minor differences $\Delta_{\max}$ and $\Delta_{\min}$, and furthermore suppose that the vertical distance between the endpoints

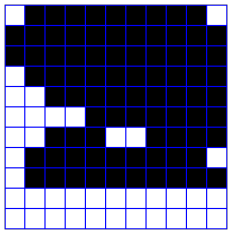


Figure 3.18: Rows of a block illustrating the Fact.

is greater than the horizontal difference. Now consider a reconstruction formed by a simple path connecting the endpoints of the run with those pixels falling between the simple path and the boundary run being filled in as black, and the rest as white. Note that there are $\Delta_{\max}$ (consecutive) pairs of adjacent rows, one of which is non-monotone, and that each such pair satisfies either case (c) or (d) above, and therefore has 2 diagonal odd bonds between the two rows. Moreover, if a reconstruction is formed by a path that is not simple, then there exists a pair of adjacent rows, one of which is non-monotone, such that case (e) applies and that therefore there are more than 2 odd bonds between them. Were the major difference in the horizontal direction we could apply the same argument counting diagonal odd bonds between adjacent columns of reconstructions determined by simple paths. We conclude that in the 8 pt. model reconstructions determined by simple paths attain the minimum number of diagonal odd bonds. Furthermore, by noting that simple paths between endpoints r_1^b and r_2^b in the 8 pt. model are a subset of those between these endpoints in the 4 pt. model, and recalling that simple paths between given endpoints in the 4 pt. model attain the minimum number of horizontal and vertical odd bonds, we additionally conclude that reconstructions in the 8 pt. model resulting from simple paths attain the minimum total number of odd bonds. Thus it remains to count the numbers of valid simple reconstruction paths for each case, as well as the numbers of odd bonds for each.

Suppose $x_{\partial B}$ contains one run of black pixels and one run of white pixels. Without loss of generality we restrict attention to reconstructions generated by the black run

and black reconstruction paths. Again, one may wish to consult Figure 3.17 in reading the arguments below.

no corners: If the black run contains no corners, then it is entirely contained within one side of the block, which without loss of generality we assume to be the right side. By Theorem III.2 there are $\Delta_{\max} + \Delta_{\min} + 3 - c$ horizontal and vertical odd bonds in a reconstruction determined by the single simple path connecting r_1^b and r_2^b. By the above Fact, there are $2\Delta_{\max}$ diagonal odd bonds between the rows intervening r_1^b and r_2^b. In addition there is a diagonal odd bond between r_2^b and its neighbor to the northwest and an additional diagonal odd bond between r_1^b and its neighbor to the southwest. Thus, there are $3\Delta_{\max} + \Delta_{\min} + 5$ odd bonds.

one corner: Arbitrarily placing r_2^b on the top of the block so that the resulting run contains one corner, we note that the number of horizontal, vertical and diagonal odd bonds is a $3\Delta_{\max} + \Delta_{\min} + 5 - 4$ since r_2^b no longer has neighbors to the north and northwest. One can see that there are again $\binom{\Delta_{\max}}{\Delta_{\min}}$ simple paths connecting r_1^b and r_2^b, as stated in (a) and (b).

two corners: Arbitrarily placing r_2^b on the left side of the block so that the resulting run contains two corners, we note that the number of horizontal, vertical and diagonal odd bonds is a $3\Delta_{\max} + \Delta_{\min} + 5 - 2$ since r_2^b no longer has neighbors to the west and southwest. One can see that there are $\binom{\Delta_{\max}}{\Delta_{\min}}$ simple paths connecting r_1^b and r_2^b, as stated in (a) and (b).

three corners: If the black run contains three corners, then the white run contains one corner. We let $\Delta'_{\max}$, $\Delta'_{\min}$ and c' refer to the white run and note that $\Delta_{\max} = \Delta'_{\max} + 1$ and $\Delta_{\min} = \Delta'_{\min} + 1$ and $c = c' + 2$. We then compute that the number of odd bonds is

$$3\Delta'_{\max} + \Delta'_{\min} + 5 - 2c' = 3(\Delta_{\max} - 1) + (\Delta_{\min} - 1) + 5 - 2(c-2) \quad (3.10)$$

$$= 3\Delta_{\max} + \Delta_{\min} + 5 - 2c \quad (3.11)$$

We see that there are again $\binom{\Delta_{\max}}{\Delta_{\min}}$ valid simple paths connecting r_1^b and r_2^b.

four corners: If the black run contains four corners then the white run contains no corners and the claims can be established by considering the no corner case.

3.7.4 Proof of Lemma III.5

Let $C = (w_1, b_2, w_2, b_2)$ be an interior widget in a MAP reconstruction, where w_1 is in the upper left corner, b_1 in the upper right, w_2 in the lower right, and b_2 in the lower left. By Lemma III.1, b_1 is connected by a monotone path to a black run, say, $r^{b,1}$, and b_2 is monotone connected to the other black run, $r^{b,2}$. This is because if b_1 and b_2 were monotone connected to the same run, then concatenating the monotone path from b_1 to $r^{b,1}$ with the monotone path from $r^{b,1}$ to b_2 would result in a monotone loop the interior of which would contain either w_1 or w_2. This would contradict the MAP assumption. Now assume there exists a second widget $C' = (w'_1, b'_2, w'_2, b'_2)$. By the same argument as above, b'_1 and b'_2 are monotone connected to different runs. Assume without loss of generality that b_1 and b'_1 are monotone connected to the same black run and b_w and b'_w are monotone connected to the other black run. Now, concatenating the monotone paths from b_1 to $r^{b,1}$, $r^{b,1}$ to b'_1, b'_1 to b'_2, b'_2 to $r^{b,2}$, $r^{b,2}$ to b_2, and that from b_2 to b_1 results in a monotone loop the interior of which contains either w_2 or w_1. This again contradicts MAP and therefore there is no other widget. This completes the proof.

3.7.5 Proof of Theorem III.6

(a) We must show that $O^*(\underline{r}) = \min\{O_2(r^{b,1}, r^{b,2}), O_2(r^{w,1}, r^{w,2})\}$. First, $O^*(\underline{r}) \leq O_2(r^{b,1}, r^{b,2})$, because as mentioned at the beginning of Section 3.2.4, merging the black pixels in one-run-optimal black reconstructions for $r^{b,1}$ and $r^{b,2}$ yields a reconstruction with at most $O_2(r^{b,1}, r^{b,2})$ odd bonds. Similarly, $O^*(\underline{r}) \leq O_2(r^{b,1}, r^{b,2})$. It follows that $O^*(\underline{r}) \leq \min\{O_2(r^{b,1}, r^{b,2}), O_2(r^{w,1}, r^{w,2})\}$. To show the reverse inequality, we note that $O^*(\underline{r}) = \min\{O_{w-HV}(\underline{r}), O_{b-HV}(\underline{r})\}$, where $O_{w-HV}(\underline{r})$ denotes the minimum number of odd bonds of all reconstructions where the white pixels are connected, and $O_{b-HV}(\underline{r})$ is defined similarly. This is due to Lemma III.5, in that if there is a single widget in a MAP reconstruction, because there are no diagonal edges in the 4 pt. model, we can flip one of the black widget pixels to give a white-HV reconstruction, so that $O_{wid} \geq O_{w-HV}$, and likewise $O_{wid} \geq O_{b-HV}$. Since $O^*(\underline{r}) = \min\{O_{w-HV}(\underline{r}), O_{b-HV}(\underline{r}), O_{wid}(\underline{r})\}$, this verifies the claim that $O^*(\underline{r}) = \min\{O_{w-HV}(\underline{r}),\ O_{b-HV}(\underline{r})\}$. We now show that $O_{w-HV}(\underline{r}) \geq O_2(r^{b,1}, r^{b,2})$. Let $B_w, B'_{b,1}, B'_{b,2}$ prescribe a white HV-connected reconstruction with fewest odd bonds. Since, $B'_{b,1}$ and $B'_{b,2}$ do not touch, the number of odd bonds in this reconstruction is at least $O^*(r^{b,1}) + O^*(r^{b,2}) \stackrel{\Delta}{=} O_2(r^{b,1}, r^{b,2})$. Therefore, $O_{w-HV}(\underline{r}) \geq O_2(r^{b,1}, r^{b,2})$. Similarly, $O_{b-HV}(\underline{r}) \geq O_2(r^{b,1}, r^{b,2})$. It now follows that $O^*(\underline{r}) = \min\{O_{w-HV}(\underline{r}), O_{b-HV}(\underline{r})\} \geq \min\{O_2(r^{b,1}, r^{b,2}), O_2(r^{w,1}, r^{w,2})\}$, and this completes the proof of (a).

(b,c) Merging the black pixels in any one-run-optimal black reconstructions for $r^{b,1}$ and $r^{b,2}$ yields a reconstruction with at most $O_2(r^{b,1}, r^{b,2})$ odd bonds. In fact, the number of odd bonds in such a reconstruction equals $O_2(r^{b,1}, r^{b,2})$ if and only if the reconstructions for $r^{b,1}$ and $r^{b,2}$ do not touch. Thus, if there exists one-run-optimal reconstructions for $r^{b,1}$ and $r^{b,2}$ that do touch, then there exists a reconstruction with fewer odd bonds than $O_2(r^{b,1}, r^{b,2})$, which would mean $O^*(\underline{r}) < O_2(r^{b,1}, r^{b,2})$. The contrapositive says that if $O^*(\underline{r}) = O_2(r^{b,1}, r^{b,2})$, then no pair of one-run-optimal re-

constructions for $r^{b,1}$ and $r^{b,2}$ can touch. Furthermore, since the number of odd bonds in the reconstruction generated by merging is $O_2(r^{b,1}, r^{b,2})$, which by hypothesis equals $O^*(\underline{r})$, we see that every such pair yields a white connected MAP reconstruction. This proves (b) and (c) is proved similarly.

3.7.6 Proof of Lemma III.7

Proof. This is proved by contraposition. Namely, we prove that if a configuration $\hat{x}_B$ contains a widget, then one or two of the pixels in the widget can be changed to reduce the number of odd bonds, indicating that $\hat{x}_B$ is not MAP. Let C denote a 2×2 subblock of B in which $\hat{x}_B$ is a widget. We let i and j denote the black widget sites and let k and l denote the white widget sites. Site i has five neighbors that are not in C, and these are referred to as *non-C neighbors*. By the Pigeon Hole Principal, at least three are black or at least three are white. The same holds for j. There are two basic cases to consider.

(A) Assume that at least three of i's non-C neighbors are white. Since i has two neighbors in C that are white, the number of odd bonds in $\hat{x}_B$ can be decreased by changing i to white. The same holds if j has at least three non-C white neighbors. Therefore, if i or j has at least three white non-C neighbors, then $\hat{x}_B$ is not MAP.

(B) The remaining case is where both i and j have at least three black non-C neighbors. Pixel i shares two non-C neighbors with site k and two more with site l. Let $b_i(k)$ and $b_i(l)$ denote the numbers of black non-C neighbors that pixel i shares with k and l, respectively. Then $0 \leq b_i(k), b_i(l) \leq 2$, and because i has at least three black non-C neighbors, we have $b_i(k) + b_i(l) \geq 2$. Similar definitions are made for $b_j(k)$ and $b_j(l)$, and the same bounds apply. We consider three specific cases based on the values of the b_i's and b_j's.

(B1) Suppose $b_i(k) = 0$, so that $b_i(l) = 2$. There are two subcases. (B1a) If $b_j(l) > 0$, then one can see that l has more black neighbors than white, which means

that changing l to black decreases the number of odd bonds, and therefore $\hat{x}_B$ is not MAP. (B1b) If $b_j(l) = 0$, then $b_j(k) = 2$, and k and l each have at least four black neighbors, i.e. at leat four odd bonds. By changing both to black means that each has at least five black neighbors, because each becomes a black neighbor of the other. Thus, each now has at most three odd bonds, a reduction, which again shows that $\hat{x}_B$ is not MAP.

(B2) Suppose that $b_i(k) = 1$, so that $b_i(l) \geq 1$. There are three subcases. (B2a) If $b_j(k) = 0$, then $b_j(l) = 2$, and pixel l has at least five black neighbors, so that changing it to black reduces the number of odd bonds, and hence $\hat{x}_B$ is not MAP. (B2b) If $b_j(k) = 1$, then $b_j(l) \geq 1$, and both k and l have at least four black neighbors. Changing both to black as in (B1b) reduces the number of odd bonds, hence $\hat{x}_B$ is not MAP. (B2c) If $b_j(k) = 2$, then k has at least five black neighbors. Changing it to black reduces the number of odd bonds. Hence $\hat{x}_B$ is not MAP.

(B3) Suppose $b_i(k) = 2$. There are two subcases. (B3a) If $b_j(k) > 0$, then k has at least five black neighbors, and changing it to white reduces the number of odd bonds. Hence $\hat{x}_B$ is not MAP. (B3b) If $b_j(k) = 0$, then both k and l have at least four black neighbors, so that changing them both to black reduces the number of odd bonds. Hence, $\hat{x}_B$ is not MAP.

This completes the proof of Lemma III.7 □

3.7.7 Proof of Theorem III.8

Proof. Assume there exists a widget w on the boundary of $(x_B^*, x_{\partial B})$. Assume without loss of generality that w includes the edge $\{r_2^{w,1}, r_1^{b,1}\}$ consisting of the last endpoint of the first white run and the first endpoint of the second black run. First, assume that neither $r_2^{w,2}$ nor $r_1^{b,1}$ is a corner pixel. The other black pixel in the widget is w_b and the other white widget site is w_w. Assuming clockwise indexing, the black runs are $r^{b,1} = (r_1^{b,1}, r_2^{b,1})$ and $r^{b,2} = (r_1^{b,2}, r_2^{b,2})$, and similarly for the white runs $r^{w,1}$ and $r^{w,2}$.

The edge $\{w_b, r_1^{b,1}\}$ is diagonal. If w_b is connected to $r^{b,1}$ through a path intersecting $r^{b,1}$ at a site r' distinct from $r_1^{b,1}$, then the white widget pixel w_w is enclosed in the loop $w_b - r' - r_1^{b,1} - w_b$, which contradicts the assumption that x_B^* is MAP. Thus the black site w_b is not HV-connected to the run $r^{b,1}$. By Lemma III.1, therefore, w_b is monotone (black) HV-connected to $r^{b,2}$ which means that the sets B_1^w and B_2^w are connected only through the diagonal edge $\{w_w, r^{w,1}2\}$.

There is a path from w_b to $r_2^{b,2}$ and a path from w_b to $r_1^{b,2}$[1]. Hence, there is a monotone black cycle $w_b - r_1^{b,2} - r_2^{b,2} - w_b$. Now assume there is a widget at $\{r_2^{b,2}, r_1^{w,1}\}$, and again assume that neither $r_2^{b,2}$ nor $r_1^{w,1}$ is a corner pixel. Letting w_b' and w_w' denote the black and white interior pixels of this second widget, we note that the edge $\{r_2^{b,2}, r''\}$ is diagonal and the white pixel w_w' is contained in the above mentioned black monotone cycle. Again, this contradicts the assumption that x_B^* is MAP. Of course the same arguments hold if we assume a second widget at $\{r_1^{b,2}, r_2^{w,2}\}$, and naturally, for any other choice of two or more widgets.

Now assume that $r_1^{b,1}$ is a corner pixel. Since by assumption $r_1^{b,1}$ is part of a widget, the black run $r^{b,1}$ consists of a single pixel, hence $r_1^{b,1} = r_2^{b,1}$. Obviously, then, w_b is not monotone HV connected to $r^{b,1}$ and is therefore monotone HV connected to $r^{b,2}$. Now assume without loss of generality that there is a second widget at $\{r_2^{b,2}, r_1^{w,1}\}$ where neither endpoint of this edge is a corner pixel. By arguments similar to those above, this implies that the white interior pixel w_w' of the second widget is contained within a monotone black loop, which contradicts the MAP assumption.

Now assume that there are two widgets that each contain a corner. It is straightforward to show that corner pixel of both widgets must be the same color. Without loss of generality, assume that both corner widget pixels are black. Thus the boundary consists of two isolated black pixels and the rest white. It is clear that the MAP reconstruction has an all white interior.

[1]Each of these paths is a one-run optimal path.

This proves that if x_B^* is a bi-connected MAP reconstruction, then x_B^* has exactly one widget on the boundary. □

3.7.8 Proof of Theorem III.9

Proof. (a) We must show that $O^*(\underline{r}) = \min\{O_2(r_1^b, r_2^b), O_2(r_1^w, r_2^w), O_{wid}(\underline{r})\}$. First, $O^*(\underline{r}) \leq O_2(r_1^b, r_2^b)$, because as mentioned just before the theorem, merging the black pixels in one-run-optimal black reconstructions for r_1^b and r_2^b yields a reconstruction with at most $O_2(r_1^b, r_2^b)$ odd bonds. Similarly, $O^*(\underline{r}) \leq O_2(r_1^b, r_2^b)$. By definition $O^*(\underline{r}) \leq O_{wid}(\underline{r})$. It follows that $O^*(\underline{r}) \leq \min\{O_2(r_1^b, r_2^b), O_2(r_1^w, r_2^w), O_{wid}(\underline{r})\}$. To show the reverse inequality, we note that $O^*(\underline{r}) = \min\{O_{w-HV}(\underline{r}), O_{b-HV}(\underline{r}), O_{wid}(\underline{r})\}$. We now show that $O_{w-HV}(\underline{r}) \geq O_2(r_1^b, r_2^b)$. Let $B_w, B'_{b,1}, B'_{b,2}$ prescribe a white HV-connected reconstruction with fewest odd bonds. Since, $B'_{b,1}$ and $B'_{b,2}$ do not touch, the number of odd bonds in this reconstruction is at least $O^*(r_1^b) + O^*(r_2^b) \stackrel{\Delta}{=} O_2(r_1^b, r_2^b)$. Therefore, $O_{w-HV}(\underline{r}) \geq O_2(r_1^b, r_2^b)$. Similarly, $O_{b-HV}(\underline{r}) \geq O_2(r_1^b, r_2^b)$. It now follows that $O^*(\underline{r}) = \min\{O_{w-HV}(\underline{r}), O_{b-HV}(\underline{r}), O_{wid}(\underline{r})\} \geq \min\{O_2(r_1^b, r_2^b), O_2(r_1^w, r_2^w), O_{wid}(\underline{r})\}$, and this completes the proof of (a).

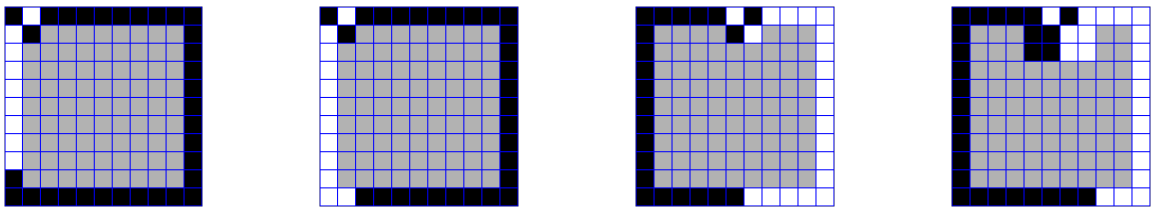
(b) We first argue that $O_{wid}(\underline{r}) \geq \max\{O_2(r_1^b, r_2^b), O_2(r_1^w, r_2^w)\} - 2$. Consider a bi-connected reconstruction with one widget on the boundary and no islands that has $O_{wid}(\underline{r})$ odd bonds. It is determined by two black paths and two white paths. Consider the black path that does not include the boundary widget pixel, and call it p_1. It is monotone HV connected to one of the black runs, say, r_1^b. If we flip the black boundary widget pixel to white, then the run p_1 has at least $O_1^*(r_1^b)$ odd bonds. Now consider the black path p_2 that is monotone HV connected to r_2^b and thus includes the boundary widget pixel. If we flip the black interior widget pixel to white, then p_2 has at least $O_1^*(r_2^b)$ odd bonds. Since the widget reduces the number of odd bonds from these two paths by a total of 2, $O_{wid}(\underline{r}) \geq \max\{O_2(r_1^b, r_2^b), O_2(r_1^w, r_2^w)\} - 2$. If the sets $B_{b,1}, B_{b,2}, B_{w,1}$, and $B_{w,2}$ meet at a widget then $O_2(r_1^b, r_2^b) = O_2(r_1^w, r_2^w)$ and

the resulting reconstruction therefore has $\min\{O_2(r_1^b, r_2^b), O_2(r_1^w, r_2^w)\} - 2$ odd bonds, which means $O_{wid}(\underline{r}) = O_2(r_1^b, r_2^b) - 2 = O_2(r_1^w, r_2^w) - 2$, hence $O^*(\underline{r}) = O_{wid}(\underline{r})$. The reconstruction determined by $B_{b,1}, B_{b,2}, B_{w,1}$, and $B_{w,2}$ obtain this minimum.

(c) (i, ii) Below we prove that if a bi-connected reconstruction has fewer odd bonds than any white or black HV connected reconstruction, then the sets $B_{b,1}, B_{b,2}, B_{w,1}$, and $B_{w,2}$ meet at a widget. The contrapositive of this says that if $B_{b,1}$ and $B_{b,2}$ do not touch at the boundary or if $B_{w,1}$ and $B_{w,2}$ do not touch at the boundary, then there is an HV connected reconstruction with at most $O_{wid}(\underline{r})$ odd bonds. In particular, if $B_{w,1}$ and $B_{w,2}$ touch at the boundary, the number of odd bonds determined by these sets is $O_2(r_1^w, r_2^w) - 2$, whereas a white HV connected reconstruction would have at least $O_2(r_1^w, r_2^w)$ odd bonds, so that $O_{w-HV} > O_{wid}(\underline{r})$. This means that if $B_{b,1}$ and $B_{b,2}$ do not touch at the boundary, then any pair of one-run-optimal paths for r_1^b and r_2^b will have $O_2(r_1^b, r_2^b)$ odd bonds, which by the contrapositive, and the fact that $O_{w-HV} > O_{wid}(\underline{r})$, will be no more than $O_{wid}(\underline{r}) \leq O_2(r_1^w, r_2^w) - 2$. Thus, $O^*(\underline{r}) = O_2(r_1^b, r_2^b)$ and merging the reconstructions for any pair of one-run-optimal paths for r_1^b and r_2^b will determine a MAP reconstruction. If neither $B_{b,1}$ and $B_{b,2}$ nor $B_{w,1}$ and $B_{w,2}$ meet at the boundary, then $O^*(\underline{r}) = \min\{O_2(r_1^b, r_2^b), O_2(r_1^w, r_2^w)\}$ by the contrapositive and any pair of one-run-optimal paths for the runs corresponding to this minimum will determine a MAP reconstruction.

Now we prove that if a bi-connected reconstruction has fewer odd bonds than any white or black HV connected reconstruction, then the sets $B_{b,1}, B_{b,2}, B_{w,1}$, and $B_{w,2}$ meet at a widget. There are two cases to consider: Neither widget pixel being in a corner, and one of the widget pixels being in the corner. We first consider the former.

Assume that a bi-connected reconstruction $\bar{x}_B$ is uniquely MAP. Consider the 3×4 subrectangle where the middle two sites of the top row are the two boundary widget sites. This is illustrated in Figure 3.19 (c) and (d). Without loss of generality assume that the upper right widget pixel is black and the upper left is white. We



(a) (b) (c) (d)

Figure 3.19: Blocks illustrating the last part of the proof of Theorem III.9

denote the white boundary widget pixel as w_1^w and the black boundary widget pixel as w_1^b. The middle two pixels in the second row of the rectangle are widget pixels and we refer to the white and black interior widget pixels as w_2^w and w_2^b, respectively. We will enumerate the remaining sites of the rectangle $1, 2, \ldots, 8$ beginning at the top left and moving counterclockwise to the top right site of the rectangle. Since we assume that $\bar{x}_B$ is uniquely MAP, then the interior black widget pixel w_2^b must have at least four black neighbors in addition to w_1^b. Since it has two white neighbors in w_1^w and w_2^w, w_2^b can have four or five black neighbors in positions 1 through 5. By the symmetric argument, the interior white widget pixel w_2^w must have four or five white neighbors in positions 4 through 8. Since there are only 8 available positions, it is clear that positions 1 through 3 must be black and positions 6 through 8 must be white. If a white pixel is in position 4 and a black pixel is in position 5, then this would form an interior widget with w_2^w and w_2^b, so positions 1 through 4 must be black and positions 5 through 8 must be white. This completes the configuration inside the 3×4 rectangle.

To show that this configuration corresponds to outer black and outer white paths, first note that the isolated boundary widget pixels are themselves (trivial) runs. Next, the black pixel in position 1 is one endpoint of the other black run. Let us denote the other endpoint of this run by r_b. Note that if we remove the top row of the block, by Lemma III.7 the remainder of the block contains no widgets and the contour connecting w_2^b and r_b must be one-run-optimal. If r_b is directly under or to the right

of w_2^w, then there is a white column of some white between w_1^b and a black pixel, call it b, directly beneath w_2^w. Moreover, since the contour between w_2^b and r_b is one-run-optimal, the entire under w_1^w is black. It can be shown that filling in the white column between w_2^w and b with black yields a black HV connected reconstruction that is no worse (i.e., does not increase the number of odd bonds), which contradicts the assumption that $\bar{x}_B$ is uniquely MAP. Therefore, the other black run endpoint r_b must be directly under or to the left of w_2^b. By the symmetric argument, the run endpoint r_w (connected to the white pixel in position 8) must be directly under or to the right of w_w^2. This implies that r_b is directly under w_2^b and r_w is directly under w_2^w. Thus the path from the black pixel in position 1 to r_b is the outer path, and the path from the white pixel in position 8 to r_w is the outer path. This completes the proof for the case that neither boundary widget pixel is in the corner.

For the case where one of the boundary widget pixels is in the corner, assume without loss of generality that the upper left corner of $\bar{x}_B$ is white. Then, the pixels to the right and below this pixel are black. For the analysis we consider the 3×3 subblock in the upper corner of $\bar{B}$, illustrated in Figure 3.19 (a) and (b). We label the widget pixels in the same manner as before, as w_1^w, w_1^b, w_2^b and w_2^w, and the remaining sites of the 3×3 subblock as $1, 2, \ldots, 5$ beginning in the lower left corner and proceeding counterclockwise. First note that w_1^b and w_2^b belong to distinct monotone HV components. At least four of pixels 1 through 5 must be white, or else pixel w_2^w could be flipped to black to yield a black HV-connected reconstruction with fewer odd bonds, which by assumption is not possible. If all five of $1, \ldots, 5$ are white, then we are done, as w_1^b and w_2^b would be the (trivial) outer paths of the runs consisting of themselves, w_1^w would similarly be the outer path for the run consisting of itself, and $1 - w_2^w - 5$ would be the outer path for the other white run. If there are four white pixels, then pixel 1 or 5 must be black, because if 2, 3, or 4 is black they would not be HV connected to w_1^b or w_2^b which would be the two black runs. Without loss

of generality assume that site 1 is black and 2 through 5 are white. Now let b be the endpoint of the black run containing w_2^b and let w be the endpoint of the white run containing pixel 5. Clearly, b and w are adjacent.

The endpoint b must be on the left border of ∂B. Otherwise we can find a black HV reconstruction that is at least as good. For example, if b is under or to the right of w_1^b, then there will be a black pixel, call it b', directly under w_1^b so that we can fill in the entire column between w_1^b and b' black and create a black HV reconstruction that is at least as good. Since b is on the left border of the block, the only simple path from w_2^b to b is the black run itself, and the resulting white path from 5 to w is the outer path for this white run. This complete the proof for the case where the widget includes a corner of the block, and it completes the proof that if a bi-connected reconstruction has fewer odd bonds than all HV reconstructions, then the outer paths from r_1^b to r_2^b touch and those from r_1^w to r_2^w all touch. □

3.7.9 Proof of Theorem III.10

Proof. We will prove this case by case.

First assume that $(i,j) \in B_w \cap B_b$. Since $(i,j) \in B_w$ there is a MAP configuration $\hat{x}^1$ with $\hat{x}^1_{(i,j)} = 0$ and since $(i,j) \in B_b$ there is a MAP configuration $\hat{x}^2$ with $\hat{x}^2_{(i,j)} = 1$. Since $\hat{x}^1$ and $\hat{x}^2$ are both MAP, they have the same number of odd bonds, therefore $o^*_{ij}(0) = o^*_{ij}(1)$ and hence $o^*_{ij} = 0$.

Now assume that $(i,j) \in B_w \setminus B_b$ and $(i,j) \in \partial B_b$. Since $(i,j) \notin B_b$, in all MAP configurations $\hat{x}$ on B, $\hat{x}_{(i,j)} = 0$. Now let $\hat{x}$ be the MAP configuration determined by B_b. We claim that since $(i,j) \in \partial B_b$, it is connected to only one site in B_b. To see this, suppose that (i,j) has two neighbors (m_1,n_1) and (m_2,n_2) in B_b. Suppose without loss of generality that (m_1,n_1) is a horizontal neighbor of (i,j) and (m_2,n_2) is a vertical neighbor of (i,j). Flipping site (i,j) to black reduces the number of odd bonds between (i,j) and its neighbors by 2, since it would now agree with both

(m_1, n_1) and (m_2, n_2). Since (i, j) only has four neighbors, the net result of flipping (i, j) to black is either a reduction in the number of odd bonds or keeping the number the same. In the former case, this contradicts the assumption that $\hat{x}$ is MAP; in the latter case this contradicts the assumption that $\hat{x}$ is determined by the outer path for r^b, which by definition, contains the most black pixels of all MAP reconstructions. Therefore, we conclude that (i, j) has only a single neighbor in B_b.

The configuration $\hat{x}$ determined by B_b obtains the minimum number of odd bonds of all configurations with site (i, j) as 0. We now want to flip (i, j) to 1 and count the increase in the number of odd bonds. We claim that the resulting number of odd bonds is the minimum number of odd bonds out of all configurations with (i, j) have the value 1. Since (i, j) has only one neighbor in B_b, assume without loss of generality that it is a horizontal neighbor $(m, n) = (i, j + 1)$. It is straightforward to verify that $B_b \cap B$ is a rectangle of sites. It follows that columns $1, \ldots, j$ are white, i.e. have value 0, in $\hat{x}$. If we flip (i, j) to black, then, we remove the odd bond between (i, j) and $(i, j + 1)$ and add an odd bond between (i, j) and $(i, j - 1)$, so the number of horizontal odd bonds is the same. However, there will be an addition of at least two odd bonds in column j by letting pixel (i, j) be 1. Now suppose there is a MAP configuration in which site $(i, j+1)$ is white. Now, all neighbors of site (i, j) are white so flipping it to black will result in four additional odd bonds. Therefore $o^*_{ij} = 2$ as claimed in the theorem.

Now suppose that $(i, j) \in B_w \setminus (B_b \cup \partial B_b)$. Let $\hat{x}$ again be the MAP configuration determined by B_b. In $\hat{x}$ all neighbors of site (i, j) are white so flipping (i, j) to black increases the number of odd bonds by four. Suppose without loss of generality that site $(i, j-k) \in B_b$ is the nearest site in B_b to (i, j). Flipping additional sites of $B \setminus B_b$ to black in order to connect (i, j) to B_b would incur at least $2(k-1)$ additional vertical odd bonds, where $k \geq 2$, while removing only 2 horizontal odd bonds. Therefore, $o^*_{ij} = 4$.

Similar arguments show that if $(i,j) \in B_b \setminus B_w$ and $(i,j) \in \partial B_w$, then $o^*_{ij} = -2$ and that if $(i,j) \in B_b \setminus (B_w \cup \partial B_w)$, then $o^*_{ij} = -4$. $\square$

CHAPTER IV

Local Conditioning for Markov Random Fields

In this chapter we discuss Local Conditioning, a method for performing exact statistical inference in a cyclic MRF. As mentioned in Section 2.4, we focus on inference for a single node but this can of course be generalized to larger subsets as we discuss briefly at the end of Section 4.1. By *exact* we mean that it can be shown that said probabilities are correct for the particular model, as opposed to *approximate* methods for which no guarantee can be given. In this chapter we present Local Conditioning in the context of the Sum-Product problem, but everything holds just as well for Max-Product, along with any other problem that can fit in the framework of the Generalized Distributive Law [1]. To compute the probability $p_G(x_i)$ that node i assumes value x_i, we need to compute the belief $Z_i(x_i)$, from which the probability can be computed straightforwardly as

$$\begin{aligned} p_G(x_i) &= \frac{Z_i(x_i)}{Z} \\ &= \frac{Z_i(x_i)}{\sum\limits_{x_i' \in \mathcal{X}_i} Z_i(x_i')}. \end{aligned}$$

If G has no cycles, the beliefs can be computed using the respective message recursion and belief computation formulas (II.4) and (II.3).

In order to compute exact beliefs for nodes of a cyclic graph, one can use, for ex-

ample, the clustering method described in Section 2.4.4, or some other such method involving grouping together nodes of the original graph. However, approximate methods such as LBP and its Tree-reweighted generalization are appealing because they are truly distributed, meaning that messages are exchanged between individual nodes of the graph rather than aggregate supernodes. This is important for scenarios such as sensor networks, where there is no natural analogue to "lumping nodes together". The Local Conditioning algorithm discussed in this chapter has the best of both of these worlds, in that it is an exact inference algorithm for cyclic graphs in which messages are passed between individual nodes rather than supernodes.

Local Conditioning is an adaptation to a general algorithm known as Conditioning, which centers around the concept of a *loop cutset* of nodes, a subset $L \subset V$ of sites such that removing those sites and all edges connected to them leaves an acyclic graph. Fixing the value at a node i is equivalent to removing i from the graph and augmenting the self-potential $\Phi_j(x_j)$ of each neighbor $j \in \partial i$ by the factor $\Psi_{ij}(\overline{x}_i, x_j)$, where $\overline{x}_i$ is the fixed value at node i, giving the new self-potential $\hat{\Phi}_j(x_j) = \Psi_{i,j}(\overline{x}_i, x_j)\Phi_j(x_j)$. Therefore, by fixing the values on a loop cutset L, we can remove L from the graph, modify the self-potentials on neighbors of L, and run BP on the acyclic remainder. Messages and beliefs computed in this manner would of course be conditioned on the particular configuration $\overline{x}_L$ assigned to L, but by iterating over all possible assignments $\overline{x}_L$ to nodes in L, the overall beliefs for nodes in $V \setminus L$ can be computed. Because of this it is helpful to think of the loop cutset nodes as representing or corresponding to *deterministic variables*, values that can change, but once fixed, remain constant while the values at other nodes are allowed to vary. This is the basis of Conditioning algorithms.

In order to compute beliefs for nodes in L, however, it is convenient to, instead of removing L, *splitting* the nodes in L into multiple *copies*. This will create a new graph $\overline{G}$, called an *edge cover*, in which there is a one-one correspondence between edges

in G and edges in $\overline{G}$. Note that the copies of nodes in L can be viewed as simply representations of the induced self-potentials described above. Retaining copies of nodes in L not only makes it easier to compute beliefs for nodes in L, since there will be nodes to "receive" information for them, but also for performing Conditioning in a parallel rather than serial fashion. Here, instead of iterating over different loop cutset configurations $\overline{x}_L$, at each iteration passing the usual vector messages, we can perform a single round of BP with matrix-valued messages. In this case, a column of the message matrix would correspond to the vector-valued message computed for that edge conditioned on some value $\overline{x}_L$ of the loop cutset nodes.

Conditioning was introduced in Pearl's early work on BP [38, 39] in the context of probabilistic models on *directed* graphs. Much research followed in regard to finding small loop cutsets [49, 6, 13] and various algorithm adaptations [46, 14]. Diez introduced the idea of Local Conditioning as a means of reducing complexity [16] by recognizing that the size of the matrix message over a given edge could be reduced by looking at the topology of the graph. Then Fay and Jaffray [20] gave an analytical justification for LC, by focusing on a (canonical) example of a directed graph. The reason the analysis in [20] was restricted to an example is that the directionality of the edges in such models limited what could be stated in terms of global update formulas and complexity formulas.

As mentioned above, Local Conditioning has a distinct advantage over clustering algorithms in that the latter require lumping nodes together into super-nodes, while the former is truly distributed with messages passed between individual units. This is important not only for sensor network type applications, but also hardware implementation on digital circuits. Furthermore, an appealing aspect of LC is that it is analogous to the "reasoning by cases" that humans do [38].

In Section 4.1 we discuss edge covers and conditioning. In Section 4.2 we review the basics of the Method of Conditioning for MRFs. In Section 4.3 we introduce Local

Conditioning for MRFs. Finally, in Section 4.4 we look at an example loop cutset for a 4 pt. grid graph and analyze the complexity of Local Conditioning under three different splitting schemes.

4.1 Edge Covers and Conditioning

In this section we discuss the idea of splitting nodes of a graph G to produce an edge cover $\overline{G}$ and how to properly define potentials on $\overline{G}$ to yield an *equivalent MRF*, by which we mean an MRF, defined on $\overline{G}$, on which computed beliefs correspond to the true beliefs on the original MRF defined on G.

An *edge cover* for G is a graph $\overline{G} = (\bar{V}, \bar{E})$ for which there exists an onto mapping $\rho : \bar{V} \to V$ satisfying the following conditions: if $\{\bar{i}, \bar{j}\} \in \bar{E}$, then $\{\rho(\bar{i}), \rho(\bar{j})\} \in E$; and if $\{i, j\} \in E$, there exists a unique edge $\{\bar{i}, \bar{j}\} \in \bar{E}$ such that $\rho(\bar{i}) = i$ and $\rho(\bar{j}) = j$. In other words, there is a one-to-one correspondence between edges of a graph G and those of an edge cover $\overline{G}$ for it. The inverse mapping ρ^{-1} from V to $\bar{V}$ is not a function from V to $\overline{V}$, as some nodes in V will be mapped to multiple nodes in $\bar{V}$, but instead can be thought of as a *splitting* of G, meaning, if for node $i \in V$, $\rho^{-1}(i)$ consists of more than one node in $\bar{V}$, the mapping ρ^{-1} can be thought of as splitting i into $|\rho^{-1}(i)|$ *copies*. For instance, if i has k neighbors in G, then we can split i into 2, 3, up to k, copies. We let $L \subset V$ denote the set of nodes that are split and let $\bar{L} \subset V$ denote the copies of these split nodes in $\overline{G}$. From here on out, we restrict attention to edge covers such that if $i \in L$ and $j \in \partial i$, then $j \notin L$. For $l \in L$, let $k(l)$ denote the number of copies of l in $\overline{G}$. Then, for $i = 1, 2, \ldots, k(l)$, we let $l^{(1)}, l^{(2)}, \ldots, l^{(k(l))}$ denote the respective copies of l in $\bar{V}$. The map ρ induces a partition of ∂l according to the copies of l to which it is connected. That is, for $i = 1, 2, \ldots, k(l)$, ρ maps the neighborhood $\partial l^{(i)}$ to a subset A_i^l of ∂l, i.e., $\rho(\partial l^{(i)}) = A_i^l$. Therefore there is a correspondence between a split node l and the set of copies $l^{(1)}, l^{(2)}, \ldots, l^{(k(l))}$ of l in $\overline{G}$. For each non-split node $i \in V \setminus L$ there is a unique non-copy node $\bar{i} \in \bar{V} \setminus \bar{L}$ that

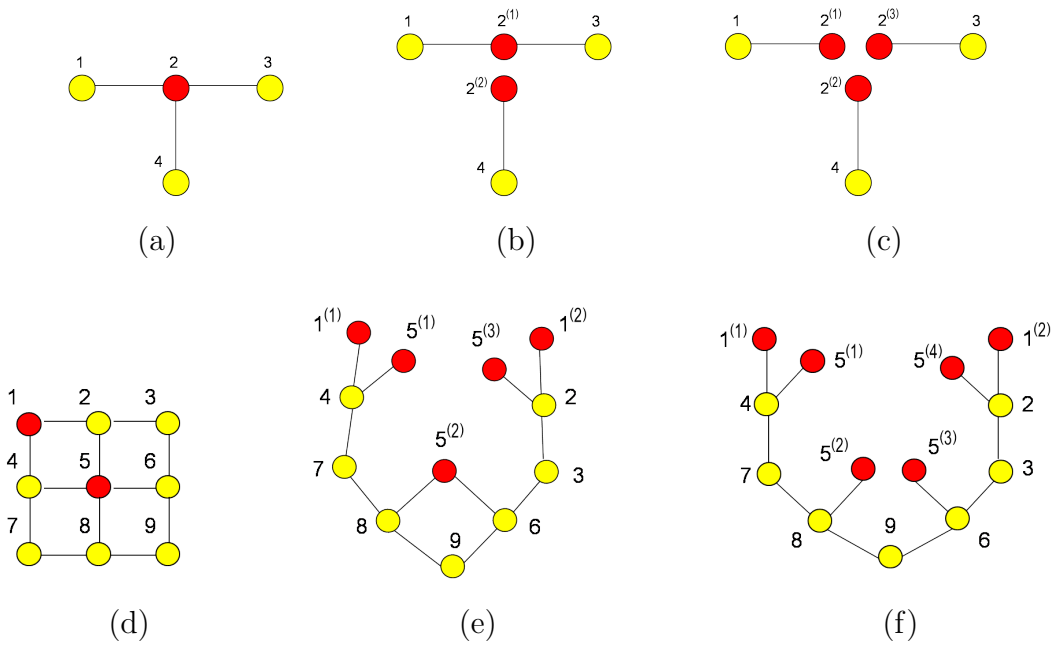


Figure 4.1: (a) An acyclic graph with split node indicated in red; (b) partial unwrapping, (c) complete unwrapping. (d) Cyclic grid graph with split node indicated in red; (e) partial unwrapping, (f) complete unwrapping.

corresponds to it. As such we will often interchange i and $\bar{i}$ when these correspond. If $\overline{G}$ is an edge cover for G and is acyclic, then it is called an acyclic edge cover. If $\overline{G}$ is acyclic and disconnected, then $\overline{G}$ is called a *forest*. Note that an edge cover $\overline{G}$ is a function both of the set of split nodes L and the splitting ρ^{-1}. We will at times refer to an edge cover both as a splitting and an *unwrapping* of the original graph G. If the splitting map ρ^{-1} is such that for each split node $l \in L$ there are as many copies of l in $\overline{G}$ as there are neighbors of l in G, then we refer to this as a *complete* unwrapping of G based on L, and denote it by $\overline{G}^* = \overline{G}^*_L$. This means that in $\overline{G}$, each copy of l is a leaf node with a unique neighbor. Figure 4.1 illustrates splitting nodes for two different graphs. In (a) we see an original acyclic graph and in (b) and (c) we see two different unwrappings. In (d) we see a cyclic grid graph and in (e) and (f) two different unwrappings for it. Subsequently, we refer to L as split nodes and $\bar{L}$ as copy nodes.

An assignment $\overline{x}_{\bar{L}}$ to the copy nodes in $\overline{G}$ is determined or induced by a config-

uration x_L on the split nodes of G in the following way. Given a value x_l for split node l, each copy $l^{(i)}$ of l is assigned the value $\overline{x}_{l^{(i)}} = x_l$. That is, all copies of a given split node are constrained to have the same value and that value corresponds to the same value on the original split node. Given the value x_j on a non-split node j, the corresponding non-copy node $\overline{j}$ is assigned the value $\overline{x}_{\overline{j}} = x_j$. In this way, we have established a one-to-one correspondence between configurations $\mathbf{x}$ on G and configurations $\bar{\mathbf{x}}$ on $\overline{G}$. The correspondence can be thought of in a two-step manner. First, a configuration x_L on the split nodes induces or selects a configuration $\overline{x}_{\bar{L}}$ on the deterministic copy nodes of $\overline{G}$. Then, an assignment to nodes in $V \setminus L$ corresponds to an identical assignment to nodes in $\bar{V} \setminus \bar{L}$. A configuration on the copy nodes $\bar{L}$ such that all copies of the same split node have the same value is referred to as *induced* or *valid* configuration and corresponds to a configuration on the split nodes. There is a one to one correspondence between configurations on the split nodes in L and valid configurations on $\bar{L}$. In the context of the upcoming Conditioning algorithms, we will perform inference over all configurations of $\mathbf{X}$ on G by fixing a configuration on $\bar{L}$, then performing inference over all configurations on $\bar{V} \setminus \bar{L}$, then repeating for all subsequent valid configurations on $\bar{L}$, and then summing them in the appropriate way.

The purpose of using an unwrapped graph $\overline{G}$ and configurations on it is to use it to compute beliefs for nodes in G. For a non-split node $j \in V \setminus L$, we want to compute the belief $Z_j(x_j)$ for different values of x_j as

$$Z_j(x_j) = \sum_{x_L} Z_{j \cup L}(x_j, x_L) \tag{4.1}$$

and for a split node $j \in L$, as

$$Z_j(x_j) = \sum_{x_{L \setminus j}} Z_L(x_j, x_{L \setminus j}). \tag{4.2}$$

With the goal of using beliefs on $\overline{G}$ to compute these beliefs on G, we now consider potentials on the nodes and edges of $\overline{G}$, namely $\overline{\Phi}$ and $\overline{\Psi}$. This will specify an MRF. However, the configurations of this MRF do not all contain valid configurations for the copy nodes. Therefore, we are not interested in this MRF, but we are interested in the distribution of the configurations of this MRF conditioned on valid configurations of the copy nodes. We find such conditional distributions by finding *conditional beliefs*. In particular, for a fixed configuration x_L and induced configuration $\overline{x}_{\bar{L}}$, the conditional belief for a node $\bar{j}$ on $\overline{G}$ and conditioned on x_L is denoted $\overline{Z}_{\bar{j}}^{(x_L)}$, and is well defined in the usual[1] way for beliefs:

for non-copy node $\bar{j} \notin \bar{L}$

$$\overline{Z}_{\bar{j}}^{(x_L)}(\overline{x}_{\bar{j}}) \;=\; \sum_{\overline{x}_{\bar{V}\setminus(\bar{L}\cup\bar{j})}} \prod_{\bar{i}\in\bar{V}} \overline{\Phi}_{\bar{i}}(\overline{x}_{\bar{i}}) \prod_{\{\bar{i},\bar{k}\}} \overline{\Psi}_{\bar{i},\bar{k}}(\overline{x}_{\bar{i}}, \overline{x}_{\bar{k}}) \tag{4.3}$$

for $\bar{j} \in \bar{L}$

$$\overline{Z}_{\bar{j}}^{(x_L)}(\overline{x}_{\bar{j}}) \;=\; \begin{cases} \displaystyle\sum_{\overline{x}_{\bar{V}\setminus\bar{L}}} \prod_{\bar{i}\in\bar{V}} \overline{\Phi}_{\bar{i}}(\overline{x}_{\bar{i}}) \prod_{\{\bar{i},\bar{k}\}} \overline{\Psi}_{\bar{i},\bar{k}}(\overline{x}_{\bar{i}}, \overline{x}_{\bar{k}}), & \overline{x}_{\bar{j}} = (x_L)_j \\ 0, & \text{else} \end{cases} \tag{4.4}$$

If we were not restricting attention to configurations with valid MRFs, beliefs would be equivalent up to a scale factor. However, in this case it will matter for computing not only beliefs for copy nodes but non-copy nodes as well, as we will see shortly. Notice that the belief vector for each copy node $\bar{j}$ in $\bar{L}$ has one nonzero component and it is the same for all copy nodes, namely, $\overline{Z}_{\bar{j}}^{(x_L)}((x_L)_{\bar{j}})$ is the same for all $\bar{j}$.

In order for beliefs computed on $\overline{G}$ to be used to compute beliefs on G, the

[1]Except for the fact that we are not summing over the values of copy nodes, since they are assumed to be fixed.

potentials $\overline{\Phi}_{\bar{i}}$, $\overline{\Psi}_{\bar{i},\bar{j}}$ should be defined to preserve the local characteristics of the original MRF on G. Namely, for edge $\{\bar{i},\bar{j}\} \in \bar{E}$, $\bar{i},\bar{j} \notin \bar{L}$, we assign the edge potential $\overline{\Psi}_{\bar{i},\bar{j}}(\overline{x}_{\bar{i}},\overline{x}_{\bar{j}}) \stackrel{\Delta}{=} \Psi_{i,j}(x_i,x_j)$, where $i=\rho(\bar{i})$ and $j=\rho(\bar{j})$. For edge $\{l^{(k)},\bar{j}\} \in \bar{E}$, where $\bar{j}$ corresponds to a neighbor of l in G, we assign the edge potential $\overline{\Psi}_{l^{(k)},\bar{j}}(\overline{x}_{l^{(k)}},\overline{x}_{\bar{j}}) \stackrel{\Delta}{=} \Psi_{l,j}(x_l,x_j)$, where $\overline{x}_{l^{(k)}} = x_l$ and $\overline{x}_{\bar{j}} = x_j$. That is, since each edge of the edge cover corresponds to an edge in the original graph, the edge potentials from the MRF on the original graph are used on the corresponding edges for the MRF on the edge cover. By assumption there are no edges from a copy of one node to another copy of that node or to a copy of another node. Likewise, for $\bar{i} \notin \bar{L}$ the self potential $\overline{\Phi}_{\bar{i}}(\overline{x}_{\bar{i}}) \stackrel{\Delta}{=} \Phi_i(x_i)$ is used, where the arguments must agree. However, for a split node $l \in L$, there are now multiple copies in $\overline{G}$ corresponding to l in the original MRF, so some modification to the self-potential of the original node must be made in order for beliefs computed on $\overline{G}$ to be used for beliefs on G. Self-potentials on split nodes are assigned as $\overline{\Phi}_{l^{(k)}}(x_l) \stackrel{\Delta}{=} \Phi_l(x_l)^{e(k)}\delta(\overline{x}_{\bar{l}})$, where the $e(k)$ can be chosen to be any real numbers that sum to one.[2] Implicit is the dependence of $\overline{\Phi}$ and $\overline{\Psi}$ on the configuration x_L.

Theorem IV.1. *Given an MRF* $\mathbf{X}$ *on* G*, an edge cover* $\overline{G}$*, with split nodes* L*, a given* x_L *on* L*, and* $\overline{\Phi}$ *and* $\overline{\Psi}$ *chosen as above, we have for each* $j \notin L$*,*

$$Z_{j\cup L}(x_j,x_L) = \overline{Z}_{\bar{j}}^{(x_L)}(\overline{x}_{\bar{j}}),$$

where $\overline{x}_{\bar{j}} = x_j$*, and for each* $j \in L$*,*

$$Z_L(x_L) = \overline{Z}_{\bar{j}}^{(x_L)}(\overline{x}_{\bar{j}}),$$

for each $\bar{j}$ *that is a copy of* j*, where* $\overline{x}_{\bar{j}} = (x_L)_j$*.*

[2]What matters is that the product of self-potentials of copy nodes equals the self-potential of the original split node.

Note that the above holds not only for any copy $\bar{j}$ of j, but also for any split node $j \in L$. This is straightforward to see, as under a given configuration x_L, the copy nodes $\overline{L} = \rho^{-1}(L)$ are fixed to the configuration $\overline{x}_{\overline{L}}$ induced by the configuration on the split nodes. Therefore, for a fixed configuration x_L on the split nodes, the belief $\overline{Z}_{\bar{j}}^{(x_L)}(\overline{x}_{\bar{j}})$ is constant for all $\bar{j} \in \overline{L}$ and all $\overline{x}_{\bar{j}}$, as long as $\overline{x}_{\bar{j}}$ is induced by x_L. It follows that in order to use beliefs on the unwrapped graph to compute beliefs (conditioned on a configuration of the split nodes) for nodes in the original graph, one computes the belief for the corresponding node of each non split nodes, and the belief of *one of* the copies of a split node.

If we substitute the results of this theorem into (4.1) and (4.2), we see that beliefs for nodes in the original graph G can be computed straightforwardly:

For $j \notin L$,

$$\begin{aligned} Z_j(x_j) &= \sum_{x'_L} Z_{j \cup L}(x_j, x'_L) \\ &= \sum_{x'_L} \overline{Z}_{\bar{j}}^{(x'_L)}(\overline{x}_{\bar{j}}), \end{aligned} \tag{4.5}$$

and for node $j \in L$,

$$\begin{aligned} Z_j(x_j) &= \sum_{x'_{L \setminus j}} Z_L(x_j, x'_{L \setminus j}) \\ &= \sum_{x'_{L \setminus j}} \overline{Z}_{\bar{j}}^{(x'_L)}(\overline{x}_{\bar{j}}) \end{aligned} \tag{4.6}$$

for any copy $\bar{j}$ of j. This is how we use the unwrapped graph to compute beliefs for G and is the basis of the method of Conditioning discussed in the following section.

The generalization of (4.1) and (4.2) to a subset $A \subset V$ of arbitrary size is

$$Z_A(x_A) = \sum_{x_{L \setminus A}} Z_{L \cup A}(x_A, x_{L \setminus A}). \tag{4.7}$$

As with single node beliefs, we can compute the belief Z_A by computing the belief for a corresponding subset of nodes $\overline{A} \subset \overline{L}$ in an unwrapped graph $\overline{G}$ based on the loop cutset L. Before, when computing the belief for a split node $j \in L$, we computed the belief for one copy $\bar{j} \in \overline{L}$ of j. Similarly, the subset $\overline{A}$ is chosen such that for each node $j \in A \cap L$, exactly one copy of j appears in $\overline{A}$. It is straightforward to show, analogously to Theorem IV.1, that

$$Z_{L\cup A}(x_A, x_{L\setminus A}) = \overline{Z}_{\overline{A}}^{(x_L)}(\overline{x}_{\overline{A}}), \tag{4.8}$$

where the beliefs $\overline{Z}_{\overline{A}}^{(x_L)}(\overline{x}_{\overline{A}})$ are defined similarly to (4.4), the constraint in the top line of (4.4) being now applied to nodes $\bar{j} \in \overline{A} \cap \overline{V}$. Therefore, as before, the belief for subset $A \subset V$ is computed as

$$\begin{aligned} Z_A(x_A) &= \sum_{x_{L\setminus A}} Z_{L\cup A}(x_A, x_{L\setminus A}) \\ &= \sum_{x_{L\setminus A}} \overline{Z}_{\overline{A}}^{(x_L)}(\overline{x}_{\overline{A}}). \end{aligned} \tag{4.9}$$

In the remainder of this chapter we will restrict attention to the case where A consists of a single node. However, in Section 6.1.2.2 we will compute beliefs $\overline{Z}_{\overline{A}}^{(x_L)}$ for subsets in an unwrapped graph as a means of computing optimal coding distributions for nodes in an original cyclic MRF.

4.2 The Method of Conditioning

We saw in the last section that beliefs computed for valid configurations of an MRF on $\overline{G}$ can be used to compute beliefs in the original MRF $\mathbf{X}$ on G. In this section we discuss an algorithm for doing this when the original graph G is cyclic. We choose L to be a *loop cutset*, which is a subset of nodes whose removal leaves an acyclic graph. In this case, the complete splitting $\overline{G}^*$ is acyclic. Therefore, for any

valid configuration, one can exactly compute conditional beliefs for $\overline{G}$ using the BP algorithm for acyclic graphs with fixed nodes as described in Section 2.4.2. Indeed, as we will review, if one runs BP for all valid configurations one can combine the results to get the beliefs for nodes in the original graph. This is called the *method of conditioning*. For the remainder of this chapter we assume the original G is cyclic and the split nodes will be loop cutset nodes of G and we will be referred to as such. The corresponding nodes in the edge cover will still be called copy nodes.

Conditioning is used to compute beliefs for both non loop cutset nodes and loop cutset nodes. Therefore, messages will be passed to and from all nodes, including copies of loop cutset nodes. All of the BP operations will be the same, the only difference being that, for a given configuration x_L on the loop cutset nodes, each copy node will have only one value that it can assume. In the following subsections we will discuss running BP on the edge cover for a fixed valid configuration on the loop cutset L and doing so in a serial manner over all configurations on the loop cutset. Then, we will discuss performing the BP runs on the conditional MRFs in a parallel manner.

4.2.1 Serial Conditioning

In this section we discuss running BP for valid configurations of an MRF on an acyclic edge cover one valid configuration at a time, and combining the results to get beliefs for the original MRF. We choose a loop cutset L, form an acyclic unwrapping $\overline{G}$ based on L, and assign potentials to nodes and edges of $\overline{G}$ to satisfy the conditions used by Theorem IV.1. We then fix a configuration x_L on the loop cutset nodes of the original graph G which induces the fixed value $\overline{x}_{\bar{L}}$ to the copy nodes of the edge cover. Belief Propagation is run on $\overline{G}$ for each configuration x_L of loop cutset nodes. To indicate that the message are being passed on the edge cover, they will be denoted, for instance, as $\overline{m}_{\bar{k}\to\bar{j}}$. We will separately consider the cases where $\overline{G}$ is connected or

not.

4.2.1.1 $\overline{G}$ Connected

If the unwrapped graph $\overline{G}$ is connected, to compute beliefs we run BP as usual, beginning with the leaves and passing the messages

$$\overline{m}_{\bar{l}\to\bar{k}}^{(\overline{x}_{\bar{L}})}(\overline{x}'_{\bar{k}}) = \sum_{\overline{x}'_{\bar{l}}} \overline{\Phi}_{\bar{l}}(\overline{x}'_{\bar{l}})\overline{\Psi}_{\bar{l},\bar{k}}(\overline{x}'_{\bar{l}}, \overline{x}'_{\bar{k}}). \tag{4.10}$$

Subsequently, internal nodes pass the messages

$$\overline{m}_{\bar{k}\to\bar{j}}^{(\overline{x}_{\bar{L}})}(\overline{x}'_{\bar{j}}) = \sum_{\overline{x}'_{\bar{k}}} \overline{\Phi}_{\bar{k}}(\overline{x}'_{\bar{k}})\overline{\Psi}_{\bar{k},\bar{j}}(\overline{x}'_{\bar{k}}, \overline{x}'_{\bar{j}}) \prod_{\bar{i}\in\partial\bar{k}\setminus\bar{j}} \overline{m}_{\bar{i}\to\bar{j}}(\overline{x}'_{\bar{j}}) \tag{4.11}$$

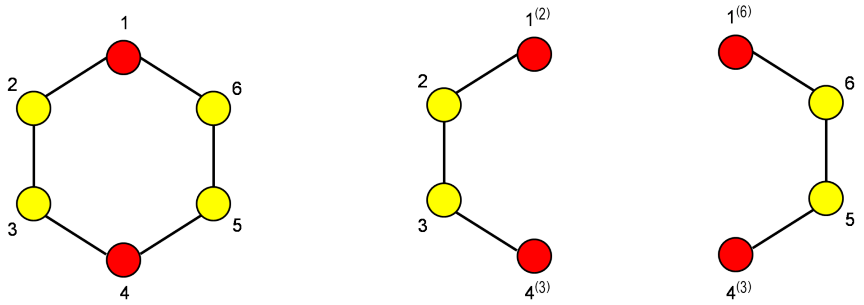
once they have received messages from all other neighbors. Note that in the above summations, if $\bar{l}$ or $\bar{k}$, are copy nodes, the sum consists of the single term corresponding to the sole value assumed by the respective copy node. The beliefs are then computed as

$$\overline{Z}_{\bar{j}}^{(\overline{x}_{\bar{L}})}(\overline{x}'_{\bar{j}}) = \overline{\Phi}_{\bar{j}}(\overline{x}'_{\bar{j}}) \prod_{\bar{k}\in\partial\bar{j}} \overline{m}_{\bar{k}\to\bar{j}}(\overline{x}'_{\bar{j}}) \tag{4.12}$$

for a non-copy node, and for a copy node as

$$\overline{Z}_{\bar{j}}^{(x_L)}(\overline{x}'_{\bar{j}}) = \begin{cases} \overline{\Phi}_{\bar{j}}(\overline{x}'_{\bar{j}}) \prod_{\bar{k}\in\partial\bar{j}} \overline{m}_{\bar{k}\to\bar{j}}(\overline{x}'_{\bar{j}}) & x'_j = (x_L)_j \\ 0 & \text{else} \end{cases} \tag{4.13}$$

Once the beliefs $\overline{Z}_{\bar{j}}^{(x_L)}$ have been computed for all nodes $\bar{j} \in \bar{V}$ for all valid configurations on $\bar{L}$, these are summed using (4.5) and (4.6) to give the beliefs for nodes in the original graph G.



(a) (b) (c)

Figure 4.2: (a) A cycle with loop cutset nodes in red; (b), (c) Components of $\overline{G}$.

4.2.1.2 $\overline{G}$ Disconnected

If $\overline{G}$ is disconnected, then we can compute beliefs for a node $\overline{j}$ by running BP on each component of $\overline{G}$ and combining them. In Section 2.4.3, the components of the forest are independent, and therefore, the probability distribution for a given node j can be computed simply by running BP on the component containing j. However, in a disconnected $\overline{G}$ the components are not independent because different copies of the same loop cutset node might lie on different components and must assume the same value. The components of $\overline{G}$ are conditionally independent, though, which means that we can run BP on each component independently of the other components. To compute the beliefs for a given node, however, we must combine the belief computed for that node on its component with the (conditional) partition functions of other components. Figure 4.2 (a) shows an original cycle while (b) and (c) show the two components of $\overline{G}$.

Let $C_{\overline{j}}$ be the component of $\overline{G}$ containing $\overline{j}$. By again beginning the BP algorithm at the leaves of $\overline{G}$ and applying the recursive message passing updates, for each node $\overline{j}$ we compute

$$\hat{\overline{Z}}_{\overline{j}}^{(C_{\overline{j}},\overline{x}_{\overline{L}})}(\overline{x}_{\overline{j}}) = \overline{\Psi}_{\overline{j}}(\overline{x}_{\overline{j}}) \prod_{\overline{k}\in\partial\overline{j}} \overline{m}_{\overline{k}\to\overline{j}}(\overline{x}_{\overline{j}}), \tag{4.14}$$

where the additional superscript $C_{\overline{j}}$ indicates that it is for the component $C_{\overline{j}}$. To compute the full belief $\hat{\overline{Z}}_{\overline{j}}^{(\overline{x}_{\overline{L}})}(\overline{x}_{\overline{j}})$ we must combine $\hat{\overline{Z}}_{\overline{j}}^{(C_{\overline{j}},\overline{x}_{\overline{L}})}(\overline{x}_{\overline{j}})$ with the partition

functions from the other components of $\overline{G}$, as described in the following theorem.

Theorem IV.2. *Let $\bar{C}_1, \bar{C}_2, \ldots, \bar{C}_M$ denote the components of $\overline{G}$. Then,*

for $\overline{j} \notin \bar{L}$,

$$\overline{Z}_{\overline{j}}^{(\overline{x}_{\bar{L}})}(\overline{x}_{\overline{j}}) = \hat{\overline{Z}}_{\overline{j}}^{(C_{\overline{j}}, \overline{x}_{\bar{L}})}(\overline{x}_{\overline{j}}) \prod_{i:\overline{j} \notin \bar{C}_i} Q_{\bar{C}_i}^{(x_L)}, \tag{4.15}$$

and for $\overline{j} \in \bar{L}$,

$$\overline{Z}_{\overline{j}}^{(\overline{x}_{\bar{L}})}(\overline{x}_{\overline{j}}) = \begin{cases} \prod_{\overline{j}' \in \rho^{-1}(j)} \hat{\overline{Z}}_{\overline{j}'}^{(C_{\overline{j}}, \overline{x}_{\bar{L}})}(\overline{x}_{\overline{j}'}) \prod_{i:\overline{j} \notin \bar{C}_i} Q_{\bar{C}_i}^{(x_L)}, & \overline{x}_{\overline{j}} = (x_L)_j \\ 0 & \textit{else} \end{cases} \tag{4.16}$$

In the above, $Q_{\bar{C}_i}^{(x_L)}$ is the partition function on component $\bar{C}_i$ with the copy nodes of $\bar{C}_i$ fixed to the values determined by x_L.

Proof. Let $C_1, C_2, \ldots, C_M$ denote the components of $\overline{G}$. There must be at least one non-copy node on each component. If this were not true for some component, then that component would consist of a single copy node, and since there is no edge this singleton can be removed. Or, there would be two adjacent copy nodes which violates the restriction imposed in Section 4.1. Let $k_1, k_2, \ldots, k_M$ denote a non-copy node on each of the respective components.

For $\overline{j} \notin \bar{L}$,

$$\begin{aligned}
\overline{Z}_{\overline{j}}^{(x_L)}(\overline{x}_{\overline{j}}) &= \sum_{\overline{x}_{\bar{V} \setminus \bar{L}}} \prod_{\overline{i} \in \bar{V}} \overline{\Phi}_{\overline{i}}(\overline{x}_{\overline{i}}) \prod_{\{\overline{i},\overline{k}\}} \overline{\Psi}_{\overline{i},\overline{k}}(\overline{x}_{\overline{i}}, \overline{x}_{\overline{k}}) \\
&= \sum_{\overline{x}_{C_{\overline{j}}}} \prod_{\overline{i} \in C_{\overline{j}}} \overline{\Phi}_{\overline{i}}(\overline{x}_{\overline{i}}) \prod_{\overline{i},\overline{k} \in C_{\overline{j}}} \overline{\Psi}_{\overline{i},\overline{k}}(\overline{x}_{\overline{i}}, \overline{x}_{\overline{k}}) \prod_{i:\overline{j} \notin \bar{C}_i} \sum_{\overline{x}'_{\overline{k}_i}} \hat{\overline{Z}}_{\overline{k}_i}^{(\overline{x}_{\bar{L}})}(\overline{x}_{\overline{k}_i}) \qquad (4.17) \\
&= \hat{\overline{Z}}_{\overline{j}}^{(\overline{x}_{\bar{L}})}(\overline{x}_{\overline{j}}) \prod_{i:\overline{j} \notin \bar{C}_i} \sum_{\overline{x}'_{\overline{k}_i}} \sum_{\overline{x}'_{\overline{k}_i}} \hat{\overline{Z}}_{\overline{k}_i}^{(\overline{x}_{\bar{L}})}(\overline{x}_{\overline{k}_i}) \qquad (4.18) \\
&= \hat{\overline{Z}}_{\overline{j}}^{(\overline{x}_{\bar{L}})}(\overline{x}_{\overline{j}}) \prod_{i:\overline{j} \notin \bar{C}_i} Q_{\bar{C}_i}^{(x_L)}. \qquad (4.19)
\end{aligned}$$

For $\bar{j} \in \bar{L}$, $\overline{Z}_{\bar{j}}^{(x_L)}(\overline{x}_{\bar{j}}) = 0$ if $\overline{x}_{\bar{j}}$ is not induced by x_L, since under the valid configuration $\overline{x}_{\bar{j}}$ any other value of $\overline{x}_{\bar{j}}$ is not possible. If $\overline{x}_{\bar{j}}$ is induced by x_L, then

$$\begin{aligned}
\overline{Z}_{\bar{j}}^{(x_L)}(\overline{x}_{\bar{j}}) &= \prod_{k:\bar{j}\notin\bar{C}_k} \sum_{\overline{x}_{C_k}} \prod_{\bar{i}\in C_{\bar{j}}} \overline{\Phi}_{\bar{i}}(\overline{x}_{\bar{i}}) \prod_{\bar{i},\bar{k}\in C_{\bar{j}}} \overline{\Psi}_{\bar{i},\bar{k}}(\overline{x}_{\bar{i}}, \overline{x}_{\bar{k}}) \prod_{i:\bar{j}\notin\bar{C}_i} \sum_{\overline{x}'_{\bar{k}_i}} \hat{\overline{Z}}_{\bar{k}_i}^{(\overline{x}_{\bar{L}})}(\overline{x}_{\bar{k}_i}) \quad (4.20) \\
&= \hat{\overline{Z}}_{\bar{j}}^{(\overline{x}_{\bar{L}})}(\overline{x}_{\bar{j}}) \prod_{i:\bar{j}\notin\bar{C}_i} Q_{\bar{C}_i}^{(x_L)}. \quad (4.21)
\end{aligned}$$

□

Example IV.3. *Consider a binary MRF on the cycle in Figure 4.2 with $L = \{1, 4\}$. Let C_1 denote the component in (b) and C_2 the component in (c). As an example of computing the probability distribution of a node not in L we compute the probability that $X_5 = 0$ as*

$$\begin{aligned}
\Pr(X_5 = 0) &= \sum_{x_L} \Pr(X_5 = 0, X_L = x_L) \\
&= \sum_{x_L} \frac{Z_{5\cup L}(0, x_L)}{\sum_{x'_L}\sum_{x_5} Z_{5\cup L}(x_5, x'_L)} \\
&= \sum_{x_L} \frac{\overline{Z}_5^{(x_L)}(0)}{\sum_{x'_L}\sum_{x_5} \overline{Z}_5^{(x'_L)}(x_5)} \quad (4.22) \\
&= \sum_{x_L} \frac{\overline{Z}_5^{(C_2,x_L)}(0) Q^{(C_1,x_L)}}{\sum_{x'_L}\sum_{x_5} \overline{Z}_5^{(C_2,x'_L)}(x_5) Q^{(C_1,x'_L)}},
\end{aligned}$$

where (4.22) follows from Theorem IV.1. Since $Q^{(C_1,x_L)}$ does not cancel with $Q^{(C_1,x'_L)}$ for all values of x'_L, we see that we need to combine the partition functions from other components to get the belief for a given node.

Now as an example of computing the distribution of a node in L we see

$$\begin{aligned}
\Pr(X_4 = 0) &= \sum_{x_L} \Pr(X_4 = 0, X_L = x_L) \\
&= \sum_{x_L} \frac{Z_L(0, x_L)}{\sum_{x'_L} \sum_{x_4} Z_L(x_4, x'_L)} \\
&= \sum_{x_L} \frac{\overline{Z}_4^{(x_L)}(0)\delta(0, (x_L)_4)}{\sum_{x'_L} \sum_{x_4} \overline{Z}_4^{(x'_L)}(x_4)\delta(x_4, (x_L)_4)} \qquad (4.23) \\
&= \sum_{x_L} \frac{\overline{Z}_4^{(C_2, x_L)}(0)\overline{Z}_4^{(C_1, x_L)}(0)\delta(0, (x_L)_4)}{\sum_{x'_L} \sum_{x_4} \overline{Z}_4^{(C_2, x'_L)}(x_4) Z_4^{(C_1, x'_L)}(x_4)\delta(x_4, (x_L)_4)},
\end{aligned}$$

where in the expression on the right side of equality (4.23) we use the fact that the value at node 4 must agree with the valid configuration on x_L, and as above, (4.23) follows from Theorem IV.1.

The complexity of serial conditioning is on the order of $|E||\mathcal{X}|^{|L|+2}$.

4.2.2 Parallel Conditioning

In addition to conditioning on loop cutset configurations in a serial or iterative manner, we will show that one can simultaneously condition on all possible assignments x_L to the loop cutset, and run BP on the corresponding valid configurations of the loop cutset nodes. Performing the conditioning in a parallel manner means that matrix-valued messages will be passed, a given column corresponding to the message over the particular edge under a specific configuration to the copy nodes. We will refer to these matrix-valued messages as *super messages*. Here, a copy node of the edge cover will no longer be held fixed to a given value but rather can be viewed as a processor having access to all of the possible values the copy node can assume. Once the message passing is complete, matrix-valued beliefs are computed, a column of which is computed from the corresponding columns of the incoming messages,

together with the self-potential. Parallel computation of the conditional messages requires that each copy node of $\overline{G}$ "be aware" of the configuration on the set of copy nodes corresponding to each column of the incoming message matrices, because the self-potential on the copy node will depend the corresponding valid configuration. Such awareness would be achieved, for instance, through a predetermined ordering of the columns of the super message, so that each copy node would "know" the corresponding configuration on x_L and adjust its self-potential accordingly. Non copy nodes, however, will simply process the incoming messages in the usual way.

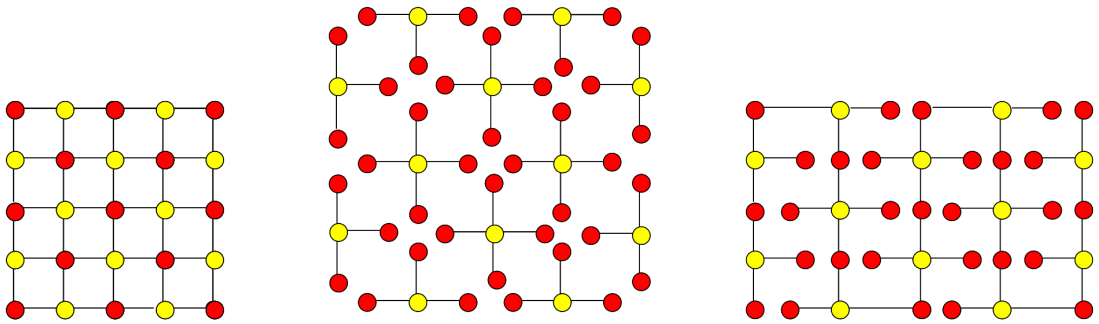
Though it is possible to perform parallel conditioning on a disconnected $\overline{G}$, using Theorem IV.2 as in the serial case above, there are advantages to using a connected unwrapping $\overline{G}$. One is that fewer overall computations will be required since we will not have to combined BP-conditional beliefs computed on different components. More importantly, though, a connected unwrapping $\overline{G}$ is required for the Local Conditioning algorithm discussed in the next chapter. Therefore, for the remainder of the chapter we assume that a connected unwrapping $\overline{G}$ is used. The following theorem states that it is always possible to find a connected acyclic edge cover based on a given loop cutset. Figure 4.3 (a) shows a grid graph and a checkerboard loop cutset, (b) a complete unwrapping, and (c) a partial unwrapping.

Theorem IV.4. *Let $G = (V, E)$ be a connected cyclic graph and let $L \subset V$ be a loop cutset of G. Then there exists a connected acyclic edge cover $\overline{G}$ based on L.*

For ease of notation, we enumerate the different configurations on $\bar{L}$ as $0, 1, \ldots,$ $|\mathcal{X}_L| - 1$. The nodes of $\overline{G}$ will pass the following matrix-valued messages

$$\overline{M}^{(L)}_{\bar{k}\to\bar{i}} \stackrel{\Delta}{=} \left[\overline{m}^{(x_L=0)}_{\bar{k}\to\bar{i}}, \overline{m}^{(x_L=1)}_{\bar{k}\to\bar{i}}, \ldots, \overline{m}^{(x_L=|\mathcal{X}_L|-1)}_{\bar{k}\to\bar{i}}\right],$$

where each $\overline{m}^{(x_L)}_{\bar{k}\to\bar{i}}$ of $\overline{M}^{(\bar{L})}_{\bar{k}\to\bar{i}}$ is a column vector representing the message values for all choices of $\overline{x}_{\bar{i}}$ conditioned on the configuration listed in the superscript. Moreover,



(a) (b) (c)

Figure 4.3: (a) A graph with fixed node indicated in red; (b) complete unwrapping; (c) partial unwrapping.

each column of $\overline{M}^{(\bar{L})}_{\bar{k}\to\bar{i}}$, which is a conditional message, is created in the same way as in (4.11) and is used in the same way to create further outgoing messages. The conditional beliefs for each node will be computed as

$$\overline{Z}^{(\bar{L})}_{\bar{j}} \triangleq \left[\overline{Z}^{(\bar{x}_{\bar{L}}=0)}_{\bar{j}}(\overline{x}_{\bar{j}}), \overline{Z}^{(\bar{x}_{\bar{L}}=1)}_{\bar{j}}(\overline{x}_{\bar{j}}), \ldots, \overline{Z}^{(\bar{x}_{\bar{L}}=|\bar{\mathcal{X}}|-1)}_{\bar{j}}(\overline{x}_{\bar{j}})\right]$$

where each column of $\overline{Z}^{(\bar{L})}_{\bar{j}}$ is computed by applying (4.14) to the corresponding columns of the incoming messages $\overline{M}^{(\bar{L})}_{\bar{k}\to\bar{j}}$, $\overline{k} \in \partial\overline{j}$.

Since each column of a super message is a message conditioned on a configuration of loop cutset nodes, when a fixed copy node $l^{(i)}$ of $\overline{G}$ processes a particular column of the incoming super messages, it modifies its self-potential as in (2.19) to correspond with the value assigned to l in that configuration of loop cutset nodes. The number of columns is exponential in the size of L. This is referred to as Global Conditioning since each message is conditioned on all loop cutset nodes and the complexity is on the order of $|E||\mathcal{X}|^{|L|+2}$.

4.3 Local Conditioning

In this section we describe Local Conditioning and show how it can result in dramatic computational savings over Global Conditioning. The Local Conditioning (LC) conditioning algorithm involves messages of the form $\overline{M}^{(A)}_{\bar{k}\to\bar{i}}$, where A is some subset of the loop cutset nodes upon which the messages are conditioned, and there is a column in $\overline{M}$ for every possible configuration of the nodes in A. The set A is referred to as the *conditioning set* for the message $\overline{M}^{(A)}$ and since complexity is proportional to the number of columns and the number of columns is exponential in the size of A, reducing the conditioning set from L to a strict subset A causes an exponential decrease in complexity. That is, the savings obtained through LC stem from there being fewer columns in the super messages over those of Global Conditioning. Sometimes we will refer to a message passed over an edge without wishing to mention the subset A. In this case we will refer to the message simply as $\overline{M}_{\bar{k}\to\bar{i}}$. In the following we will show how to choose A as small as possible in a way that nevertheless permits the beliefs to be computed.

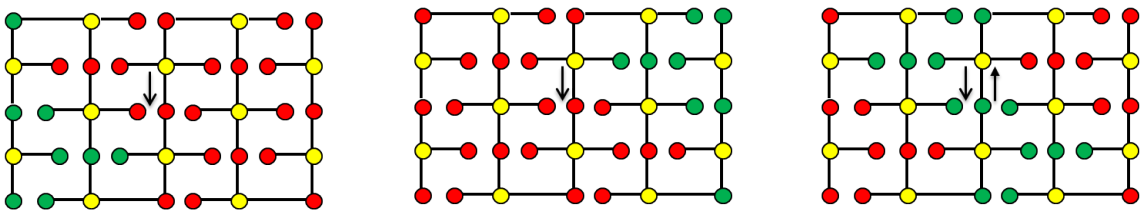
4.3.1 Setup

Let $\overline{G} = \overline{G}(L, \rho)$ be a connected acyclic unwrapping of G based on loop cutset L and splitting ρ^{-1}. We define $\overline{G}_{\bar{k}\setminus\bar{i}}$ and $\overline{G}_{\bar{i}\setminus\bar{k}}$ similarly to $G_{k\setminus i}$ and $G_{i\setminus k}$ in Section 2.4.1, to be the component trees containing k, respectively i, after removing edge $\{i, k\}$ from tree $\overline{G}$.

Definition IV.5. *A node $\bar{j} \in \bar{V}$ is said to be* upstream *of message $\overline{M}_{\bar{k}\to\bar{i}}$ if $\bar{j} \in \overline{G}_{\bar{k}\setminus\bar{i}}$ and* downstream *of message $\overline{M}_{\bar{k}\to\bar{i}}$ if $\bar{j} \in \overline{G}_{\bar{i}\setminus\bar{k}}$.*

For edge $\{\bar{i}, \bar{k}\} \in \bar{E}$, the set of loop cutset nodes in G whose copies in $\overline{G}$ are all upstream of $M_{\bar{i}\to\bar{k}}$ (equivalently downstream of $M_{\bar{k}\to\bar{i}}$) will be denoted by $L_{\bar{i}\setminus\bar{k}}$[3]. As

[3]Dependence on $\overline{G}$ is implicit.



(a) (b) (c)

Figure 4.4: Unwrapping of 5×5 grid with green nodes indicating (a) downstream nodes; (b) upstream nodes; (c) relevant loop cutset nodes.

we will see, the method of Local Conditioning reduces the complexity of computing a message $\overline{M}_{\bar{k}\to\bar{i}}$ from $\bar{k}$ to $\bar{i}$ by identifying those loop cutset nodes all copies of which are upstream or all copies of which are downstream of $\overline{M}_{\bar{k}\to\bar{i}}$. For edge $\{\bar{i},\bar{k}\} \in \bar{E}$, we let $R_{\bar{k}\bar{i}} = R_{\bar{i}\bar{k}}$ denote the set $L \setminus (L_{\bar{i}\setminus\bar{k}} \cup L_{\bar{k}\setminus\bar{i}})$ of loop cutset nodes that are neither upstream nor downstream and it is referred to as the *relevant* set of loop cutset nodes for $\{\bar{i},\bar{k}\}$. For node $\bar{i}$, the set

$$R_{\bar{i}} = \bigcup_{\bar{k}\in\partial\bar{i}} R_{\bar{k}\bar{i}}$$

is the *relevant* set for $\bar{i}$. If a loop cutset node l belongs to $L_{\bar{k}\setminus\bar{i}}$ for some $\bar{k} \in \partial\bar{i}$, then l is not relevant for edge $\{\bar{k},\bar{i}\}$ and hence is not relevant for $\bar{i}$. Therefore, we can express the relevant set $R_{\bar{i}}$ as

$$R_{\bar{i}} = L \setminus (\bigcup_{\bar{k}\in\partial\bar{i}} L_{\bar{k}\setminus\bar{i}}). \tag{4.24}$$

Moreover, for two distinct neighbors $\bar{k}_1$ and $\bar{k}_2$ of $\bar{i}$, a loop cutset node cannot belong to both $L_{\bar{k}_1\setminus\bar{i}}$ and $L_{\bar{k}_2\setminus\bar{i}}$. In other words, the sets $L_{\bar{k}\setminus\bar{i}}$, $\bar{k} \in \partial\bar{i}$, partition the loop cutset nodes $L \setminus R_{\bar{i}}$ that are not relevant for a node $\bar{i}$. However, there does not exist a decomposition similar to (4.24) for loop cutset nodes in the sets $L_{\bar{i}\setminus\bar{k}}$, $\bar{k} \in \partial\bar{i}$. The reason is that a loop cutset node may be downstream of a message $M_{\bar{k}\to\bar{i}}$ yet still be relevant for the edge connecting $\bar{i}$ to another of its neighbors, and hence relevant

for $\bar{i}$ itself. In sum, for an edge the relevant loop cutset nodes are determined by redundant and summable loop cutset nodes, whereas for a node, it is the summable nodes that determine the relevant set.

For loop cutset node $l \in L$, we define $\overline{G}_L^{(l)}$ to be the subgraph of $\overline{G}$ consisting of all nodes and edges contained in non-backtracking paths connecting copies of l in $\overline{G}$.[4] We can see that $R_{\bar{i}}$ is the set of loop cutset nodes l such that $\bar{i}$ lies on the subtree $\overline{G}_L^{(l)}$. For instance, if $\overline{k}$ is a neighbor of $\bar{i}$ and if $l^{(m)}$ and $l^{(n)}$ are two copies of a given loop cutset node $l \in L$ such that $l^{(m)}$ is downstream of $\overline{M}_{\overline{k}\to\bar{i}}$ and $l^{(n)}$ is upstream of $\overline{M}_{\overline{k}\to\bar{i}}$, then the path connecting $l^{(n)}$ and $l^{(m)}$ goes through the edge $\{\overline{k}, \bar{i}\}$, and therefore node $\bar{i}$. It is the sizes of the sets $\{R_{\overline{i}\overline{k}}\}$ and $\{R_{\bar{i}}\}$ that determine the complexity of Local Conditioning.

4.3.2 Reduced and Summed Out Loop Cutset Nodes

In this section we show how the notions of upstream and downstream nodes provide a means of reducing the complexity from that of Global Conditioning.

Definition IV.6. *A loop cutset node $l \in L$ is said to be* redundant *with respect to message $\overline{M}_{\overline{k}\to\bar{i}}$ if each copy $l^{(m)}$ is downstream of $\overline{M}_{\overline{k}\to\bar{i}}$. In other words, the set $L_{\bar{i}\setminus\overline{k}}$ is the set of redundant loop cutset nodes with respect to $\overline{M}_{\overline{k}\to\bar{i}}$.*

If l is redundant with respect to $\overline{M}_{\overline{k}\to\bar{i}}$, i.e., all copies of cutset node l are downstream of message $\overline{M}_{\overline{k}\to\bar{i}}$, then for each configuration x_L, the conditional message $\overline{m}_{\overline{k}\to\bar{i}}^{(x_L)}$ is not a function of x_l. In other words, for two configurations x_L and x'_L on the loop cutset nodes that agree everywhere except at a redundant loop cutset node l, the message vectors $\overline{m}_{\overline{k}\to\bar{i}}^{(x_L)}$ and $\overline{m}_{\overline{k}\to\bar{i}}^{(x'_L)}$ will be identical. This is because, by Definition II.2, $\overline{m}_{\overline{k}\to\bar{i}}^{(x_L)}$ is function consisting of sums and products of potentials for nodes and edges contained in the subtree $\overline{G}_{\overline{k}\setminus\bar{i}}$, and no copy of a redundant node l is in the subtree $\overline{G}_{\overline{k}\setminus\bar{i}}$. This leads to the following proposition.

[4] There is only one non-backtracking path connecting any two copies of a given loop cutset node.

Proposition IV.7 (Redundant Loop Cutset Nodes). Let $B \subset L_{\bar{i}\setminus\bar{k}}$ be a subset of loop cutset nodes that are redundant with respect to $\overline{M}_{\bar{k}\to\bar{i}}$, and let x_L and x'_L be configurations such that $x_{L\setminus B} = x'_{L\setminus B}$ and $x_B \neq x'_B$. Then,

$$\overline{m}_{\bar{k}\to\bar{i}}^{(x_L)} = \overline{m}_{\bar{k}\to\bar{i}}^{(x'_L)}.$$

For a given configuration $x_{L\setminus L_{\bar{i}\setminus\bar{k}}}$ on loop cutset nodes that are not redundant with respect to $\overline{M}_{\bar{k}\to\bar{i}}$, there are $|\mathcal{X}|^{|L_{\bar{i}\setminus\bar{k}}}$ identical message column vectors all corresponding to this configuration. Therefore, we send *reduced* super messages by sending only one of these message columns and we let $\widetilde{M}_{\bar{k}\to\bar{i}}$ denote the reduced super message. Since the reduced message array $\widetilde{M}_{\bar{k}\to\bar{i}}^{(L\setminus B)}$ has fewer columns than $M_{k\to i}^{(L)}$, the resulting BP complexity is $\mathcal{O}(|\mathcal{X}|^{|L|-|B|})$ rather than $\mathcal{O}(|\mathcal{X}|^{|L|})$. However, this creates the need for an operation that allows us to "fill out" incoming messages that have been reduced so that they can be combined appropriately.

Example IV.8. *Consider the* 2×3 *grid graph shown in Figure 4.5. Because* 1 *is redundant for* $\overline{M}_{3\to 2}$*, the message from node 3 to node 2 is a function of loop cutset node 6 only and therefore reduces to the form*

$$\widetilde{M}_{3\to 2}^{(\{1,6\})} = \left[m_{3\to 2}^{(x_6=0)}, m_{3\to 2}^{(x_6=1)}\right]. \tag{4.25}$$

On the other hand, the message from node 1 to node 2 is a function of node 1 only and of the form

$$\widetilde{M}_{1\to 2}^{(\{1,6\})} = \left[m_{1\to 2}^{(x_1=0)}, m_{1\to 2}^{(x_1=1)}\right]. \tag{4.26}$$

Upon receiving these two messages, node 2 applies an expansion operator $W(\cdot)$ *to*

produce "filled-out" messages having a column for every possible pair of x_1, x_6 values.

$$\begin{aligned} W(\widetilde{M}_{3\to 2}^{(\{1,6\})}) &= \left[m_{3\to 2}^{(x_6=0)}, m_{3\to 2}^{(x_6=1)}, m_{3\to 2}^{(x_6=0)}, m_{3\to 2}^{(x_6=1)}\right] \\ &= \left[m_{3\to 2}^{(x_1x_6=00)}, m_{3\to 2}^{(x_1x_6=01)}, m_{3\to 2}^{(x_1x_6=10)}, m_{3\to 2}^{(x_1x_6=11)}\right]. \end{aligned}$$

and

$$\begin{aligned} W(\widetilde{M}_{1\to 2}^{(\{1,6\})}) &= \left[m_{1\to 2}^{(x_1=0)}, m_{1\to 2}^{(x_1=0)}, m_{1\to 2}^{(x_1=1)}, m_{1\to 2}^{(x_1=1)}\right], \\ &= \left[m_{1\to 2}^{(x_1x_6=00)}, m_{1\to 2}^{(x_1x_6=01)}, m_{1\to 2}^{(x_1x_6=10)}, m_{1\to 2}^{(x_1x_6=11)}\right]. \end{aligned}$$

respectively. To form the message from 2 to 5, the matrices $W(\widetilde{M}_{3\to 2}^{(\{1,6\})})$ and $W(\widetilde{M}_{1\to 2}^{(\{1,6\})})$ are multiplied together component-wise and combined with the self-potential for node 2 to form the matrix

$$\left[\Phi_1 m_{1\to 2}^{(x_1=0)} m_{3\to 2}^{(x_6=0)}, \Phi_1 m_{1\to 2}^{(x_1=0)} m_{3\to 2}^{(x_6=1)}, \Phi_1 m_{1\to 2}^{(x_1=1)} m_{3\to 2}^{(x_6=0)}, \Phi_1 m_{1\to 2}^{(x_1=1)} m_{3\to 2}^{(x_6=1)}\right] \quad (4.27)$$

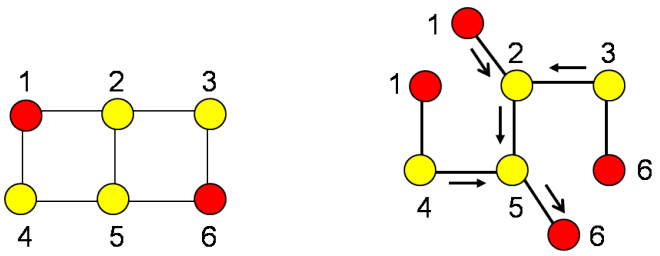
which is then multiplied by the potential matrix

$$A_{2,5} = \begin{bmatrix} \Psi_{2,5}(0,0) & \Psi_{2,5}(1,0) \\ \Psi_{2,5}(0,1) & \Psi_{2,5}(1,1) \end{bmatrix} \quad (4.28)$$

to form the outgoing message $\widetilde{M}_{2\to 5}^{\{1,6\}}$. □

As seen in this example, a node $\bar{i}$ may receive incoming messages that are conditioned on different subsets of L. To form an outgoing message $\widetilde{M}_{\bar{i}\to\bar{k}}$ we have to expand each of the incoming messages to the same size.

Definition IV.9. *For $A \subset B \subset V$, we define W_A^B to be the operator that maps*



(a) (b)

Figure 4.5: (a) 2×3 grid, loop cutset nodes in red; (b) Unwrapping based on $\{1, 6\}$.

message array $\tilde{M}_{k\to i}^{(X_A)}$ *to message array* $\tilde{M}_{k\to i}^{(X_A, X_{B\setminus A})}$ *through the identity*

$$m_{k\to i}^{(x_A, x_{B\setminus A})} = m_{k\to i}^{(x_A)} \tag{4.29}$$

for all $x_A \in \mathcal{X}_A$ *and* $x_{B\setminus A} \in \mathcal{X}_{\mathcal{B}\setminus\mathcal{A}}$.

The operator $W_A^B(\cdot)$ allows us to take a reduced message $\widetilde{M}^{(A)}$ and expand it to reduced message $\widetilde{M}^{(B)}$, where $A \subset B$, by simply creating duplicates of message column vectors of $\widetilde{M}^{(A)}$. In particular, if a node $\bar{i}$ receives incoming messages $\widetilde{M}_{\bar{k}_n\to\bar{i}}^{(A_n)}$ and wishes to form from these an outgoing reduced message, then each of the incoming reduced messages need to be filled out so that the effective conditioning set B is the union of the conditioning sets A_n of the incoming messages. Namely, for each incoming message $\widetilde{M}_{\bar{k}_n\to\bar{i}}^{(A_n)}$ we apply the expansion $W_{A_n}^{\bigcup_n A_n}(\cdot)$.

Complementary to redundant loop cutset nodes, and the other source of computational savings from LC, are *summable loop cutset nodes*.

Definition IV.10. *A loop cutset node* l *is said to be* summable *with respect to message* $M_{\bar{k}\to\bar{i}}$ *if each copy* $l^{(m)}$ *is upstream of* $M_{\bar{k}\to\bar{i}}$. *In other words, the set* $L_{\bar{k}\setminus\bar{i}}$ *is the set of summable loop cutset nodes with respect to* $M_{\bar{k}\to\bar{i}}$.

If loop cutset node $l \in L$ is upstream of $\overline{M}_{\bar{k}\to\bar{i}}$ then $\overline{M}_{\bar{k}\to\bar{i}}$ does not need to be conditioned on X_l because, as we will explain, message columns corresponding to configurations of the loop cutset nodes differing only in the values of summable nodes

can be added together, thus further reducing the number of message columns that must be transmitted in a given message.

Definition IV.11. *For a subset $B \subset L_{\bar{i}\setminus\bar{k}}$ of the loop cutset nodes that are summable with respect to message $M_{k\to i}$, the* summed out message $\widehat{m}_{\bar{k}\to\bar{i}}^{(x_{L\setminus B})}$ *is defined as*

$$\widehat{m}_{\bar{k}\to\bar{i}}^{(x_{L\setminus B})}(\overline{x}_{\bar{i}}) = \sum_{x'_B} \overline{m}_{\bar{k}\to\bar{i}}^{(x_{L\setminus B}, x_B)}(\overline{x}_{\bar{i}}). \tag{4.30}$$

No loss of information takes place through this summing out process, in that subsequent summed out messages can be computed from incoming summed out messages, as summarized in the following.

Theorem IV.12 (Summed Out Message Recursion)**.** *Let $B \subset L_{\bar{k}\setminus\bar{i}}$ be a subset of loop cutset nodes that are summable with respect to message $\overline{M}_{\bar{k}\to\bar{i}}$. For each neighbor $\bar{j}_n \in \partial\bar{k} \setminus \bar{i}$, let $A_n \subset L_{\bar{j}_n\setminus\bar{k}}$ be a subset of loop cutset nodes summable with respect to message $\overline{M}_{\bar{j}_n\setminus\bar{k}}$ such that $A_n \subset B$ for all n. Then*

$$\widehat{m}_{\bar{k}\to\bar{i}}^{(x_{L\setminus B})}(\overline{x}_{\bar{i}}) = \sum_{\bar{x}'_{\bar{k}}} \overline{\Psi}_{\bar{i}\bar{k}}(\overline{x}_{\bar{i}}, \overline{x}'_{\bar{k}}) \overline{\Phi}_{\bar{k}}(\overline{x}'_{\bar{k}}) \sum_{x'_{B\setminus(\bigcup_n A_n)}} \prod_{\bar{j}_n \in \partial\bar{k}\setminus\bar{i}} \widehat{m}_{\bar{j}_n\to\bar{k}}^{(x_{L\setminus A_n}, x_{A_n})}(\overline{x}'_{\bar{k}}) \tag{4.31}$$

This theorem says that if a node $\bar{k}$ receives summed out messages from its neighbors, it can use them to compute outgoing summed out messages. The condition that each of the A_n be contained in the set B is required, in that if for some m, A_m is not contained in B, then certain loop cutset nodes will have been summed out in the message $\hat{m}_{\bar{j}_m\to\bar{k}}$ whereas they should not be summed out in summed out message $\hat{m}_{\bar{k}\to\bar{i}}$. Summed out messages reduce the complexity of a message $\overline{M}_{\bar{k}\to\bar{i}}$ in much the same way as reduced messages. In reduced super messages, message columns corresponding to different configurations on redundant loop cutset nodes are all replicates of each other and therefore can be identified with any of them. In super messages where summed out messages $\hat{M}_{\bar{k}\to\bar{i}}$ are used, message columns corresponding

to different configurations on summable loop cutset nodes can be added to together, leaving a single summed out message vector. Because summing out summable loop cutset nodes will reduce the size of the super messages being passed, expansion will be needed to combine incoming summed out messages from different neighbors.

Example IV.13. *Consider the graph shown in Figure 4.6. The message from node 4 to node 3 is*

$$\overline{M}_{4\to 3}^{(\{1,8\})} = \left[m_{4\to 3}^{(x_1x_8=00)},\ m_{4\to 3}^{(x_1x_8=01)}, m_{4\to 3}^{(x_1x_8=10)},\ m_{4\to 3}^{(x_1x_8=11)}\right] \tag{4.32}$$

and the message from node 7 to node 3 is

$$\overline{M}_{7\to 3}^{(\{8\})} = \left[m_{7\to 3}^{(x_1x_8=00)},\ m_{7\to 3}^{(x_1x_8=01)}, m_{7\to 3}^{(x_1x_8=10)},\ m_{7\to 3}^{(x_1x_8=11)}\right]. \tag{4.33}$$

These two incoming messages are first combined through component-wise multiplication to yield the outgoing super message from 3 to 2

$$\overline{M}_{3\to 2}^{(\{1,8\})} = \left[m_{3\to 2}^{(x_1x_8=00)},\ m_{3\to 2}^{(x_1x_8=01)}, m_{3\to 2}^{(x_1x_8=10)},\ m_{3\to 2}^{(x_1x_8=11)}\right]. \tag{4.34}$$

Since node 8 is upstream of the message from 3 to 2, we sum out node 8 to form the summed out message

$$\hat{M}_{3\to 2}^{(\{1\})} = \left[m_{3\to 2}^{(x_1x_8=00)} + m_{3\to 2}^{(x_1x_8=10)},\ m_{3\to 2}^{(x_1x_8=01)} + m_{3\to 2}^{(x_1x_8=11)}\right]. \tag{4.35}$$

As a word of contrast, with reduced messages, the complexity reducing operation is on the super message, by identifying identical columns so that fewer messages have to be computed. This complexity reducing operation is invertible, by replicating columns to fill out the super message to a super message conditioned on a larger subset of loop cutset nodes. With summed out messages, the complexity reducing

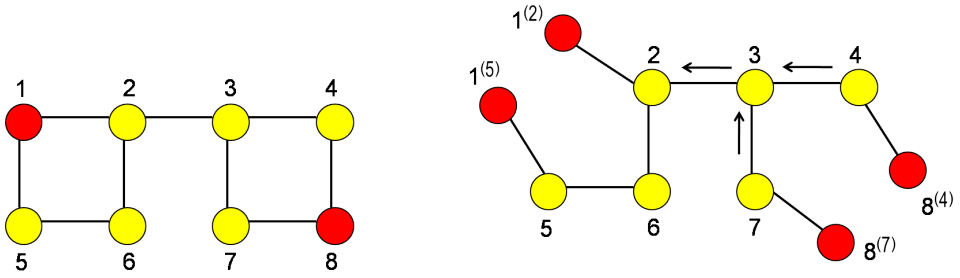


Figure 4.6: (a) 2×3 grid, loop cutset nodes in red; (b) Unwrapping based on $\{1, 6\}$.

operation is on columns of the super message, adding certain of them together. This operation is not invertible and is the reason for the inclusion constraint in Theorem IV.12.

Combining reduced and summed out messages yields the method of Local Conditioning. This achieves a great computational advantage over Global Conditioning. For a given message $\overline{M}_{\overline{k}\to\overline{i}}$, the subset of summable loop cutset nodes B is chosen to be the entire set of summable nodes with respect to $\overline{M}_{\overline{k}\to\overline{i}}$, which is $L_{\overline{k}\setminus\overline{i}}$. Then, summed out messages will be of the form $\widehat{m}_{\overline{k}\to\overline{i}}^{(\overline{x}_{R_{\overline{ik}}},\overline{x}_{L_{\overline{i}\setminus\overline{k}}})}$. By Proposition IV.7, for distinct configurations $\overline{x}_{L_{\overline{i}\setminus\overline{k}}}$ and $\overline{x}'_{L_{\overline{i}\setminus\overline{k}}}$ on the redundant loop cutset nodes for $\overline{M}_{\overline{k}\to\overline{i}}$, $\widehat{m}_{\overline{k}\to\overline{i}}^{(\overline{x}_{R_{\overline{ik}}},\overline{x}_{L_{\overline{i}\setminus\overline{k}}})} = \widehat{m}_{\overline{k}\to\overline{i}}^{(\overline{x}_{R_{\overline{ik}}},\overline{x}'_{L_{\overline{i}\setminus\overline{k}}})}$. Therefore the message sent over edge $\{\overline{k}, \overline{i}\}$ should be conditioned only on the relevant loop cutset nodes $R_{\overline{ik}}$ for the edge, and therefore denoted as $\hat{m}_{\overline{k}\to\overline{i}}^{(x_{R_{\overline{ik}}})}$.

The following theorem is similar to Theorem IV.12, the difference being that all nodes in $L_{\overline{k}\to\overline{i}}$ are being summed out and we are taking into account the redundant nodes $L_{\overline{i}\to\overline{k}}$.

Theorem IV.14 (Local Conditioning Message Recursion)**.** *Let $R_{\overline{ik}} = R_{\overline{ki}}$ be the set of relevant nodes for edge $\{\overline{i}, \overline{k}\}$. Then,*

$$\hat{m}_{\overline{k}\to\overline{i}}^{(x_{R_{\overline{ik}}})} = \sum_{x'_{\overline{k}}} \overline{\Psi}_{\overline{ik}}(x_{\overline{i}}, x'_{\overline{k}}) \overline{\Phi}_{\overline{k}}(x'_{\overline{k}}) \sum_{x'_{L_{\overline{k}\setminus\overline{i}}}} \prod_{\overline{j}\in\partial\overline{k}\setminus\overline{i}} \hat{m}_{\overline{j}\to\overline{k}}^{(x_{R_{\overline{jk}}})}(x'_k) \tag{4.36}$$

We can now combine summed out messages as with standard messages before to get what we call *Local Conditioning beliefs*.

Definition IV.15 (Local Conditioning Beliefs)**.** *Using the summed out messages* $\hat{m}_{\overline{k}\to\overline{i}}^{(x_{R_{\overline{ik}}})}$*, the* Local Conditioning beliefs *are defined as*

$$\widehat{Z}_{\overline{i}}^{(x_{R_{\overline{i}}})}(\overline{x}_{\overline{i}}) \stackrel{\Delta}{=} \overline{\Phi}_{\overline{i}}(\overline{x}_{\overline{i}}) \prod_{\overline{k}\in\partial\overline{i}} \hat{m}_{\overline{k}\to\overline{i}}^{(x_{R_{\overline{ik}}})}(\overline{x}_{\overline{i}}). \tag{4.37}$$

The following theorem states that the original beliefs for nodes in G can be computed from the Local Conditioning beliefs. It is analogous to combining (4.1) and (4.2) with Theorem IV.1 above.

Theorem IV.16 (Beliefs from Local Conditioning Beliefs)**.** *The beliefs can be computed from the summed out messages as*

$$Z_i(x_i) = \sum_{x_{R_{\overline{i}}}} \widehat{Z}_{\overline{i}}^{(x_{R_{\overline{i}}})}(\overline{x}_{\overline{i}}). \tag{4.38}$$

The following new theorem establishes the LC propagation formulas for undirected cyclic graphs analogous to (2.17) and (2.18) for undirected acyclic graphs.

Theorem IV.17. *The LC formulas (in matrix notation) are*

$$\widehat{Z}_{\overline{i}}^{(R_{\overline{i}})} = W_i^{R_{\overline{i}}}(\overline{\Phi}_{\overline{i}}) \odot \prod_{\overline{k}\in\partial\overline{i}}^{\odot} W_{R_{ki}}^{R_i}(\widehat{M}_{\overline{k}\to\overline{i}}^{(R_{\overline{ki}})}).$$

and

$$\widehat{M}_{\overline{i}\to\overline{k}}^{(R_{\overline{ik}})} = A_{i,k} \cdot \left[W_{\overline{i}}^{R_{\overline{i}}}(\overline{\Phi}_{\overline{i}}) \odot \prod_{\overline{j}\in\partial\overline{i}\setminus\overline{k}}^{\odot} W_{R_{\overline{ji}}}^{R_{\overline{i}}}(\widehat{M}_{\overline{j}\to\overline{i}}^{(R_{\overline{ji}})}) \right] \cdot s_i^k,$$

where s_i^k *is a* $|R_i| \times |R_{ik}|$ *matrix that performs the 'summing out' over the loop cutset nodes in* $\cup_{j\in\partial i\setminus k} R_{ji}$.

4.3.3 Complexity of LC

The complexity of LC on a given graph G is no longer strictly a matter of the number of loop cutset nodes. It is the topology of the (connected) unwrapping $\overline{G}$ based on L that determines the relevant set for a particular edge, and therefore the complexity of performing LC on $\overline{G}$.

Theorem IV.18. *The complexity of LC using loop cutset L is $\mathcal{O}(|E||\mathcal{X}|^{c(L)})$, where we say the* cost *of L is*

$$c(L) = \max_{i \in \overline{V}} |R_i|. \tag{4.39}$$

Proof. Without loss of generality, assume that each node assumes values in a common alphabet $\mathcal{X}$. By examining Theorem IV.17 one can see that the number of multiplications that are required in combining the incoming messages is $|\mathcal{X}|^{|R_i|}$. □

4.4 Example: Checkerboard Loop Cutset on 4 pt. Grid Graph

In this section we use Theorem VI.8 to evaluate the complexity of Local Conditioning with three different *unwrapping patterns* based on this loop cutset. Consider an $m \times n$ 4 pt. grid graph G and the checkerboard loop cutset shown in Figure 4.3 (a). We want to look at unwrapping patterns rather than simply unwrappings at this point since different values of m and n will define different graphs and therefore unwrappings of these graphs will be different even if they follow the same pattern and hence have complexity which is the same function of m and n. The first unwrapping pattern that we will look at is illustrated in Figure 4.3 (c) and again in Figure 4.7 and will be denoted in this section $\overline{G}_1$. It can be seen as "unzipping" the original grid graph, moving up and then down adjacent columns. The next unwrapping pattern is shown in Figure 4.8 (a) and will be indicated by $\overline{G}_2$. It can be seen as anchoring the grid graph at the top row, and then separating adjacent columns straight down from

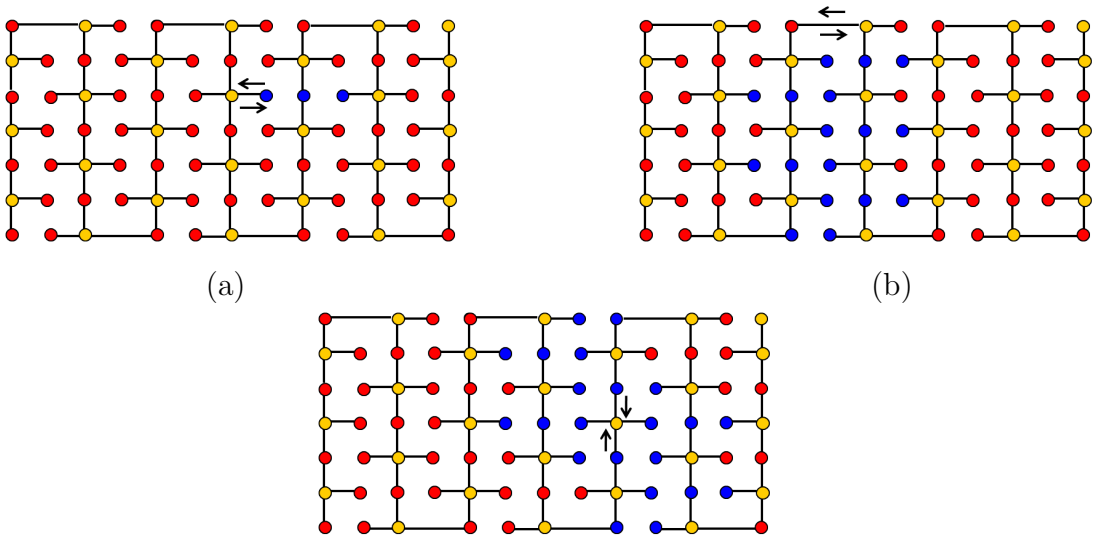


Figure 4.7: (a)-(b) Relevant sets for different horizontal edges; (c) Relevant set for a node.

the top row. The third unwrapping pattern is displayed in Figure 4.9 (b) and will be denoted by $\overline{G}_3$. It is similar to $\overline{G}_2$ except that the middle row is anchored. Note that each of these unwrappings is a partial unwrapping, as the complete unwrapping $\overline{G}^*$ based on the checkerboard loop cutset is disconnected, as shown in Figure 4.3 (b). Also, for each of these unwrappings we note further that the unwrapping can leave each of the columns connected or each of the rows connected. We will assume throughout that $m < n$ and that the columns are connected in the unwrappings.

Theorem IV.19. *Let G be an $m \times n$ $(m < n)$ 4 pt. grid graph. The complexity of Local Conditioning using the unwrapping pattern $\overline{G}_1$ of G based on the checkerboard loop cutset is $\mathcal{O}(|E||\mathcal{X}|^m)$*

Proof. Looking at Figure 4.4 (c) one can observe that with columns connected in unwraping $\overline{G}_1$, each vertical edge of $\overline{G}_1$ has a relevant node in each row. Two vertical edges incident to the same node have all but one of their relevant nodes in common, so the relevant set for that node is again the number of rows of $\overline{G}_1$ plus one. The horizontal edges of $\overline{G}_1$ either likewise have a relevant node in each row or have only

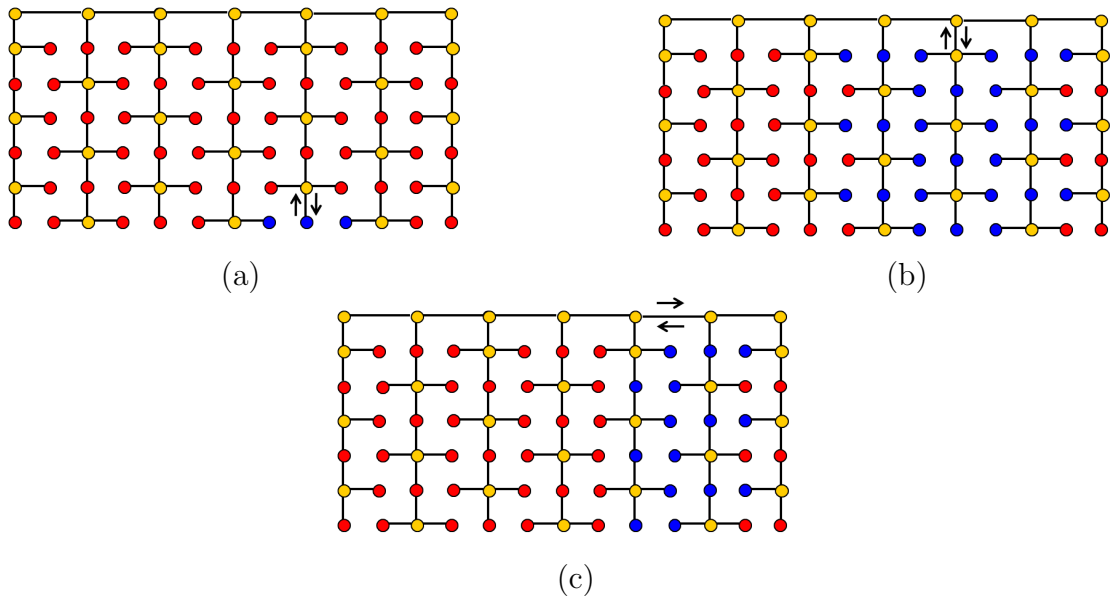


Figure 4.8: (a) Unwrapped graph $\overline{G}_2$. (b) Unwrapped graph $\overline{G}_3$.

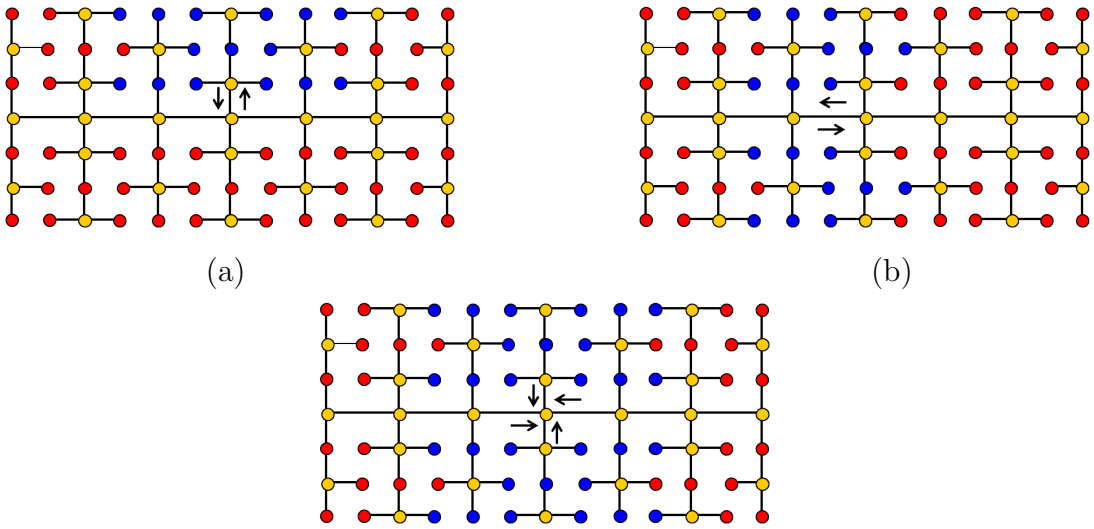
one relevant node. A horizontal edge and vertical edge that are incident to the same node also have the same relevant set, so the relevant set for the common node is again the same size. □

In $\overline{G}_2$ and $\overline{G}_3$, not only are the unwrappings partial, but certain nodes of the checkerboard loop cutset have not been split. The split graphs $\overline{G}_2$ and $\overline{G}_3$ can still be thought of as unwrappings based on the checkerboard loop cutset where the mapping ρ^{-1} is simply a one-to-one correspondence for these nodes, or one can identify a smaller loop cutset on which $\overline{G}_2$ and $\overline{G}_3$ are based.

Theorem IV.20. *Let G be an $m \times n$ $(m < n)$ 4 pt. grid graph. The complexity of Local Conditioning using the unwrapping pattern $\overline{G}_2$ of G based on the checkerboard loop cutset is $\mathcal{O}(|E||\mathcal{X}|^{(3/2)m})$*

Proof. In this unwrapping there is one connected row. From Figure 4.8 (a) we can see that in $\overline{G}_2$ with columns connected, the size of the relevant set for a vertical edge depends on how close the vertical is to the connected row. For a vertical edge that is incident to the connected row, roughly half of the nodes in the same column and half

of the nodes in each of the two adjacent columns are relevant. Each horizontal edge along the top row of the unwrapping has relevant nodes in half of the two nearest columns. Therefore the size of the largest relevant set is (approximately) one-and-a-half times the number of rows. □



(a) (b) (c)

Figure 4.9: (a) Unwrapped graph $\overline{G}_2$. (b) Unwrapped graph $\overline{G}_3$.

Theorem IV.21. *Let G be an $m \times n$ $(m < n)$ 4 pt. grid graph. The complexity of Local Conditioning using the unwrapping pattern $\overline{G}_3$ of G based on the checkerboard loop cutset is $\mathcal{O}(|E||\mathcal{X}|^{(3/2)m})$*

Proof. As with $\overline{G}_2$ there is one connected row, and furthermore, the size of the relevant set for a vertical edge depends on how close it is to the connected row. For a vertical edge incident to the connected row, the size of the relevant set is roughly 3/4 times the number of rows, but arguments similar to the case of $\overline{G}_2$. For a node on the connected row, the union of the relevant sets for the two incident vertical edges has size 3/2 times the number of rows, and the relevant set for a horizontal edge on the connected row is a subset of this union. □

We can see from these examples that while the unwrapping patterns $\overline{G}_2$ and $\overline{G}_3$ are based on loop cutsets that are strict subsets of the checkerboard loop cutset, the exponent in the order of complexity is greater. However, there is a more varied distribution of complexity over the edges of the unwrapping. This is because in unwrapping pattern $\overline{G}_1$ all vertical edges have the same complexity, whereas in $\overline{G}_2$ vertical edges that are furthest from the connected row have complexity proportional to $|\mathcal{X}|^2$.

4.5 Proofs

4.5.1 Proof of Theorem IV.1

Proof. Assume that $j \in L$. If $\mathbf{x}$ is a configuration on G and $\bar{\mathbf{x}}$ is the corresponding configuration on $\bar{G}$, then

$$
\begin{aligned}
\bar{Z}^{(x_L)}(\bar{x}_{j^{(m)}}) &= \sum_{\bar{x}'_{(\bar{V}\setminus\bar{L})}} \prod_{\substack{\{i,k\}:\\ i,k\notin\bar{L}}} \bar{\Psi}_{i,k}(\bar{x}'_i,\bar{x}'_k) \prod_{i\notin\bar{L}} \bar{\Phi}_i(\bar{x}'_i) \prod_{\bar{l}\in\bar{L}\setminus\rho^{-1}(j)} \bar{\Phi}_{\bar{l}}(\bar{x}_{\bar{l}}) \prod_{k\in\partial\bar{l}} \bar{\Psi}_{\bar{l},k}(\bar{x}_{\bar{l}},\bar{x}'_k) \bar{\Phi}_{j^{(m)}}(\bar{x}_{j^{(m)}}) \\
&\quad \cdot \prod_{k\in\partial j^{(m)}} \bar{\Psi}_{j^{(m)},k}(\bar{x}_{j^{(m)}},\bar{x}_k) \prod_{j^{(n)}\in\rho^{-1}(j)\setminus j^{(m)}} \bar{\Phi}_{j^{(n)}}(\bar{x}_{j^{(n)}}) \prod_{k\in\partial j^{(n)}} \bar{\Psi}_{j^{(n)},k}(\bar{x}_{j^{(n)}},\bar{x}_k) \\
&= \sum_{\bar{x}'_{(\bar{V}\setminus\bar{L})}} \prod_{\substack{\{i,k\}:\\ i,k\notin\bar{L}}} \bar{\Psi}_{i,k}(\bar{x}'_i,\bar{x}'_k) \prod_{i\notin\bar{L}} \bar{\Phi}_i(\bar{x}'_i) \prod_{\bar{l}\in\bar{L}\setminus\rho^{-1}(j)} \prod_{m=1}^{k(l)} \bar{\Phi}_{l^{(m)}}(\bar{x}_{l^{(m)}}) \prod_{\bar{j}\in\partial l^{(m)}} \bar{\Psi}_{l^{(m)},\bar{j}}(\bar{x}_{l^{(m)}},\bar{x}'_{\bar{j}}) \\
&\quad \cdot \prod_{j^{(n)}\in\rho^{-1}(j)} \bar{\Phi}_{j^{(n)}}(\bar{x}_{j^{(n)}}) \prod_{k\in\partial j^{(n)}} \bar{\Psi}_{j^{(n)},k}(\bar{x}_{j^{(n)}},\bar{x}_k) \\
&= \sum_{\bar{x}'_{(\bar{V}\setminus\bar{L})}} \prod_{\substack{\{\bar{i},\bar{j}\}:\\ i,j\notin L}} \bar{\Psi}_{\bar{i},\bar{j}}(\bar{x}'_{\bar{i}},\bar{x}'_{\bar{j}}) \prod_{i\notin L} \bar{\Phi}_{\bar{i}}(\bar{x}_{\bar{i}}) \prod_{l\in L\setminus j} \prod_{j\in\partial l} \Psi_{l,j}(x_l,x''_j) \prod_{m=1}^{k(l)} \Phi_l^{e(m)}(x_l) \\
&\quad \cdot \prod_{j^{(n)}\in\rho^{-1}(j)} \Phi_j(\bar{x}_j)^{e(n)} \prod_{k\in\partial j^{(n)}} \Psi_{j,k}(x_j,x_k) \\
&= \sum_{x''_{(V\setminus L)}} \prod_{\substack{\{i,j\}:\\ i,j\notin L}} \Psi_{i,j}(x''_i,x''_j) \prod_{i\notin L} \Phi_i(x''_i) \prod_{l\in L} \prod_{j\in\partial l} \Psi_{l,j}(x_l,x''_j) \prod_{m=1}^{k(l)} \Phi_l^{e(m)}(x_l) \qquad (4.40) \\
&= \sum_{x''_{(V\setminus L)}} \prod_{\substack{\{i,j\}:\\ i,j\notin L}} \Psi_{i,j}(x''_i,x''_j) \prod_{i\notin L} \Phi_i(x''_i) \prod_{l\in L} \prod_{j\in\partial l} \Psi_{l,j}(x_l,x''_j) \Phi_l(x_l) \\
&= Z_L(x_j,x_L),
\end{aligned}
$$

where the equality in 4.41 follows from the fact that there is a one-one correspondence between configurations on $\bar{G}$ and configurations on G, and because for each corresponding pair of configurations the respective summands are equal. A very similar argument establishes the case where $j \notin L$.

$$
\begin{aligned}
\bar{Z}^{(\bar{x}_{\bar{L}})}(\bar{x}_{\bar{j}}) &= \sum_{\bar{x}'_{\bar{V}\setminus(\bar{L}\cup\{\bar{j}\})}} \prod_{\substack{\{\bar{i},\bar{j}\}:\\ \bar{i},\bar{j}\notin\bar{L}\cup j}} \bar{\Psi}_{\bar{i},\bar{j}}(\bar{x}'_{\bar{i}},\bar{x}'_{\bar{j}}) \prod_{\bar{i}\notin\bar{L}\cup j} \bar{\Phi}_{\bar{i}}(\bar{x}'_{\bar{i}}) \prod_{\bar{l}\in\bar{L}} \bar{\Phi}_{\bar{l}}(\bar{x}_{\bar{l}}) \prod_{\bar{j}\in\partial\bar{l}} \bar{\Psi}_{\bar{l},\bar{j}}(\bar{x}_{\bar{l}},\bar{x}'_{\bar{j}}) \\
&\quad \cdot\bar{\Phi}_j(\bar{x}_j) \prod_{\bar{k}\in\partial\bar{j}} \bar{\Psi}_{j,\bar{k}}(\bar{x}_j,\bar{x}_{\bar{k}}) \\
&= \sum_{\bar{x}'_{\bar{V}\setminus(\bar{L}\cup\{\bar{j}\})}} \prod_{\substack{\{\bar{i},\bar{j}\}:\\ i,j\notin L\cup j}} \bar{\Psi}_{\bar{i},\bar{j}}(\bar{x}'_{\bar{i}},\bar{x}'_{\bar{j}}) \prod_{i\notin L\cup j} \bar{\Phi}_{\bar{i}}(\bar{x}_{\bar{i}}) \prod_{\bar{l}\in\bar{L}} \prod_{m=1}^{k(l)} \bar{\Phi}_{l^{(m)}}(\bar{x}_{l^{(m)}}) \prod_{\bar{j}\in\partial l^{(m)}} \bar{\Psi}_{l^{(m)},\bar{j}}(\bar{x}_{l^{(m)}},\bar{x}'_{\bar{j}}) \\
&\quad \cdot\bar{\Phi}_j(\bar{x}_j) \prod_{\bar{k}\in\partial\bar{j}} \bar{\Psi}_{j,\bar{k}}(\bar{x}_j,\bar{x}_{\bar{k}}) \\
&= \sum_{x''_{V\setminus(L\cup\{j\})}} \prod_{\substack{\{i,j\}:\\ i,j\notin L\cup j}} \Psi_{i,j}(x''_i,x''_j) \prod_{i\notin L\cup j} \Phi_i(x''_i) \prod_{l\in L} \prod_{j\in\partial l} \Psi_{l,j}(x_l,x''_j) \prod_{m=1}^{k(l)} \Phi_l^{e(m)}(x_l) \qquad (4.41) \\
&= \sum_{x''_{V\setminus(L\cup\{j\})}} \prod_{\substack{\{i,j\}:\\ i,j\notin L\cup j}} \Psi_{i,j}(x''_i,x''_j) \prod_{i\notin L\cup j} \Phi_i(x''_i) \prod_{l\in L} \prod_{j\in\partial l} \Psi_{l,j}(x_l,x''_j)\Phi_l(x_l) \\
&= Z_L(x_j,x_L),
\end{aligned}
$$

□

4.5.2 Proof of connected acyclic edge cover

Proof. We let $\overline{G}$ be an unwrapping of G based on some loop cutset L. If $\overline{G}$ is connected the theorem is proved so we assume that $\overline{G}$ is a forest. We enumerate the components of $\overline{G}$ as $C_1, C_2, \cdots, C_p$. We will form a tree iteratively by adding components one-by-one as follows. Letting $\overline{G}(n)$ denote the tree formed after n iterations, we will 're-identify' a copy $l^{(a)}$ of a loop cutset node on a component C_i not in $\overline{G}(n)$ with a copy $l^{(b)}$ of the same loop cutset node on the tree $\overline{G}(n)$. We will denote by $A_n \subset \{1,2,\ldots,p\}$ the set of component indices connected after n iterations and by $\bar{L}(n) \subset L$ the set of cutset nodes copies of which are in $\overline{G}$.

Choose an arbitrary $i \in \{1,2,\cdots,p\}$ and set $A_1 = i$. For $n \in \{2,\cdots,p\}$, suppose

there exists a copy node $l^{(m)} \in \bigcup_{j \in A_n} C_j$ and a component C_k, $k \notin A_n$, containing another copy $l^{(t)}$ of the same loop cutset node. Then, we attach C_k to the tree $\overline{G}(n)$ by identifying $l^{(m)}$ and $l^{(t)}$, forming $\overline{G}(n+1)$. We then set $A_{n+1} = A_n \cup \{k\}$, and in this case, $\bar{L}(n+1) = \bar{L}(n)$. Now suppose that for all loop cutset nodes $l \in \bar{L}(n)$ and all $k \notin A_n$, C_k does not contain a copy of l. This implies that $\overline{G}(n)$ corresponds to its own graph, that the original graph G was disconnected, a contradiction. Therefore, we can always find component C_k that can be attached to $\overline{G}(n)$, hence we can form a tree. □

4.5.3 Proof of Proposition IV.7

Proof. Let L be a loop cutset of the graph G and $\overline{G}$ an unwrapping of G based on L. Expressing $\overline{m}^{(x_L)}_{\overline{k} \to \overline{i}}$ in terms of potentials we get

$$\overline{m}^{(x_L)}_{\overline{k} \to \overline{i}}(\overline{x}_{\overline{i}}) \;=\; \sum_{\overline{x}_{\overline{k}}} \overline{\Phi}_{\overline{i}}(\overline{x}_{\overline{i}}) \overline{\Psi}_{\overline{k},\overline{i}}(\overline{x}_{\overline{k}}, \overline{x}_{\overline{i}}) \overline{Z}^{(x_L)}_{\overline{k} \setminus \overline{i}}(\overline{x}_{\overline{k}}).$$

Since $\overline{Z}^{(x_L)}_{\overline{k} \setminus \overline{i}}(\overline{x}_{\overline{k}})$ is the belief of node $\overline{k}$ in the subgraph $\overline{G}_{\overline{k} \setminus \overline{i}}$ it is a function of potentials defined on nodes and edges within $\overline{G}_{\overline{k} \setminus \overline{i}}$. Since all copies of l are downstream of $\overline{M}_{\overline{k} \to \overline{i}}$, no copy l is contained in $\overline{G}_{\overline{k} \setminus \overline{i}}$, hence $\overline{m}^{(x_{L \setminus l}, x_l)}_{\overline{k} \to \overline{i}}$ is not a function of x_l. This completes the proof.

□

4.5.4 Proof of Theorem IV.12

Proof. First, note that for each $\overline{j}_n \in \partial \overline{k} \setminus \overline{i}$, the set of loop cutset nodes $L_{\overline{j}_n \setminus \overline{k}}$ is fully contained in $L_{\overline{k} \setminus \overline{i}}$. Note that for $\overline{j}_m$ and $\overline{j}_n$ in $\partial \overline{k} \setminus \overline{i}$, the sets $L_{\overline{j}_m \setminus \overline{k}}$ and $L_{\overline{j}_n \setminus \overline{k}}$ are

disjoint. Through direct computation we have

$$\begin{aligned}
\hat{m}_{\overline{k}\to\overline{i}}^{(x_{L\setminus B})}(\overline{x}_{\overline{i}}) &= \sum_{x'_B} \overline{m}_{\overline{k}\to\overline{i}}^{(x_{L\setminus B},x'_B)}(\overline{x}_{\overline{i}}) \\
&= \sum_{x'_{B\setminus\bigcup_n A_n}} \sum_{x'_{A_1}} \cdots \sum_{x'_{A_{|\partial k|-1}}} \overline{m}_{\overline{k}\to\overline{i}}^{(x_{L\setminus B},x'_B)}(\overline{x}_{\overline{i}}) \\
&= \sum_{x'_{B\setminus\bigcup_n A_n}} \sum_{x'_{A_1}} \cdots \sum_{x'_{A_{|\partial k|-1}}} \sum_{\overline{x}'_{\overline{k}}} \overline{\Psi}_{\overline{i}\overline{k}}(\overline{x}_{\overline{i}},\overline{x}'_{\overline{k}})\overline{\Phi}_k(x'_k) \prod_{n=1}^{|\partial k|-1} \overline{m}_{\overline{j}_n\to\overline{k}}^{(x_{L\setminus B},x'_B)}(\overline{x}'_{\overline{k}}) \\
&= \sum_{x'_{B\setminus\bigcup_n A_n}} \sum_{\overline{x}'_{\overline{k}}} \overline{\Psi}_{\overline{i}\overline{k}}(\overline{x}_{\overline{i}},\overline{x}'_{\overline{k}})\overline{\Phi}_{\overline{k}}(\overline{x}'_{\overline{k}}) \sum_{x'_{A_1}} \cdots \sum_{x'_{A_{|\partial k|-1}}} \prod_{n=1}^{|\partial k|-1} \overline{m}_{\overline{j}_n\to\overline{k}}^{(x_{L\setminus B},x'_B)}(\overline{x}'_{\overline{k}}) \\
&= \sum_{x'_{B\setminus\bigcup_n A_n}} \sum_{\overline{x}'_{\overline{k}}} \overline{\Psi}_{\overline{i}\overline{k}}(\overline{x}_{\overline{i}},\overline{x}'_{\overline{k}})\overline{\Phi}_{\overline{k}}(\overline{x}'_{\overline{k}}) \prod_{n=1}^{|\partial k|-1} \sum_{x'_{A_n}} \overline{m}_{\overline{j}_n\to k}^{(x_{L\setminus B},x'_{B\setminus A_n}x'_{A_n})}(\overline{x}'_{\overline{k}}) \\
&= \sum_{x'_{B\setminus\bigcup_n A_n}} \sum_{\overline{x}'_{\overline{k}}} \overline{\Psi}_{\overline{i}\overline{k}}(\overline{x}_{\overline{i}},\overline{x}'_{\overline{k}})\overline{\Phi}_{\overline{k}}(\overline{x}'_{\overline{k}}) \prod_{n=1}^{|\partial k|-1} \sum_{x'_{A_n}} \overline{m}_{\overline{j}_n\to k}^{(x''_{L\setminus A_n},x'_{A_n})}(\overline{x}'_{\overline{k}}) \\
&= \sum_{x'_{B\setminus\bigcup_n A_n}} \sum_{\overline{x}'_{\overline{k}}} \overline{\Psi}_{\overline{i}\overline{k}}(\overline{x}_{\overline{i}},\overline{x}'_{\overline{k}})\overline{\Phi}_{\overline{k}}(\overline{x}'_{\overline{k}}) \prod_{\overline{j}_n\in\partial k\setminus i} \hat{m}_{\overline{j}_n\to\overline{k}}^{(x''_{L\setminus A_n})}(\overline{x}'_{\overline{k}})
\end{aligned}$$

where $x''_{L\setminus A_n} = (x_{L\setminus B}, x'_{B\setminus A_n})$. □

4.5.5 Proof of Theorem IV.14

Proof. First, note that for each $j \in \partial k \setminus i$, the set of loop cutset nodes $L_{j\setminus k}$ is fully contained in $L_{k\setminus i}$. Next, let $\bar{L}_{k\setminus i} = L_{k\setminus i} \setminus \left(\bigcup_{j\in\partial k\setminus i}\right)$. Note that for j_m and j_n in $\partial k \setminus i$, the sets $L_{j_m\setminus k}$ and $L_{j_n\setminus k}$ are disjoint. Also note that for each $j_m \in \partial k \setminus i$, the sets

$L_{j_m \setminus k}$ and $\bar{L}_{k \setminus i}$ are disjoint. Through direct computation we have

$$\begin{aligned}
\tilde{m}_{k \to i}^{(x_{R_{ik}})} &= \sum_{x'_{L_{k \setminus i}}} m_{k \to i}^{(x_{L_{k \setminus i}}, x_{R_{ik}})}(x_i) \\
&= \sum_{x'_{L_{k \setminus i}}} \sum_{x''_k} \Psi_{ik}(x_i, x''_k) \Phi_k(x''_k) \prod_{j \in \partial k \setminus i} m_{j \to k}^{(x_{L_{j \setminus k}}, x_{R_{jk}})}(x''_k) \\
&= \sum_{x''_k} \Psi_{ik}(x_i, x''_k) \Phi_k(x''_k) \sum_{x'_{L_{k \setminus i}}} \prod_{j \in \partial k \setminus i} m_{j \to k}^{(x_{L_{j \setminus k}}, x_{R_{jk}})}(x''_k) \\
&= \sum_{x''_k} \Psi_{ik}(x_i, x''_k) \Phi_k(x''_k) \sum_{x'_{\bar{L}_{k \setminus i}}} \prod_{j \in \partial k \setminus i} \sum_{x'_{L_{j \setminus k}}} m_{j \to k}^{(x_{L_{j \setminus k}}, x_{R_{jk}})}(x''_k) \\
&= \sum_{x''_k} \Psi_{ik}(x_i, x''_k) \Phi_k(x''_k) \sum_{x'_{\bar{L}_{k \setminus i}}} \prod_{j \in \partial k \setminus i} \tilde{m}_{j \to k}^{(x_{R_{jk}})}(x''_k)
\end{aligned}$$

□

4.5.6 Proof of Theorem IV.16

Proof. Let L be a loop cutset for a graph G, $\overline{G}$ an unwrapping of G based on L, and $\overline{L}$ the corresponding set of copy nodes in $\overline{G}$. We first note that, for a given node $\bar{i}$, the set of loop cutset nodes L can be partitioned as $L = (R_{\bar{i}}, L_{\bar{k}_1 \setminus \bar{i}}, L_{\bar{k}_2 \setminus \bar{i}}, \ldots, L_{\bar{k}_n \setminus \bar{i}})$. Then, we have

$$\begin{aligned}
Z_i(x_i) &= \sum_{x_L} \overline{Z}_{\bar{i}}^{(x_L)}(\overline{x}_{\bar{i}}) \\
&= \sum_{x_L} \overline{\Phi}_{\bar{i}}(\overline{x}_{\bar{i}}) \prod_{\bar{k} \in \partial \bar{i}} \overline{m}_{\bar{k} \to \bar{i}}^{(x_L)}(\overline{x}'_{\bar{i}}) \\
&= \sum_{x_{R_{\bar{i}}}} \overline{\Phi}_{\bar{i}}(\overline{x}_{\bar{i}}) \sum_{x_{L \setminus R_{\bar{i}}}} \prod_{\bar{k} \in \partial \bar{i}} \overline{m}_{\bar{k} \to \bar{i}}^{(x_L)}(\overline{x}'_{\bar{i}}) \\
&= \sum_{x_{R_{\bar{i}}}} \overline{\Phi}_{\bar{i}}(\overline{x}_{\bar{i}}) \prod_{\bar{k} \in \partial \bar{i}} \sum_{x_{L_{\bar{k} \setminus \bar{i}}}} \overline{m}_{\bar{k} \to \bar{i}}^{(x_{R_{\bar{k}\bar{i}}}, x_{L_{\bar{k} \setminus \bar{i}}})}(\overline{x}'_{\bar{i}}) \\
&= \sum_{x_{R_{\bar{i}}}} \overline{\Phi}_{\bar{i}}(\overline{x}_{\bar{i}}) \prod_{\bar{k} \in \partial \bar{i}} \widehat{m}_{\bar{k} \to \bar{i}}^{(x_{R_{\bar{k}\bar{i}}})}(\overline{x}'_{\bar{i}})
\end{aligned}$$

□

CHAPTER V

Monotonicity and Reduction in MRFs

In Chapter III we presented the general idea of lossy cutset coding and gave explicit MAP reconstruction results when applied to the class of homogenous Ising models with no external field. Later, in Section 6.2, we present a lossless cutset coding method for general MRFs in which a cutset of nodes is losslessly encoded and then the remaining nodes are losslessly encoded conditioned on the cutset. The rationale behind this approach is that by losslessly encoding the values X_U on a cutset, processing (i.e., encoding or estimating) the remaining variables is simplified due to the conditional independence properties of Markov random fields. This still leaves the generally intractable task of encoding the cutset X_U, for even if U itself is relatively sparse, marginalizing over the remaining $|V \setminus U|$ variables has complexity of exponential order. And herein lies the tradeoff: in order to guarantee tractable solutions at the decoder approximations have to be made at the encoder in that the cutset X_U must be encoded using a proxy distribution $\tilde{p}_U$ instead of the true marginal distribution p_U.

Due to the convenient parametrization of an MRF[1] as a labeling of an associated graph, the most natural proxy distributions are those derived by *thinning* or *reducing* the model by eliminating edges of the graph. The effect of this is that the new, reduced proxy distribution is more tractable to inference (or, for that matter, parameter

[1] And graphical models more generally.

estimation) algorithms. For example, removing all edges from a fully connected graph yields a memoryless distribution, which is trivial to process. Eliminating all edges is unnecessary, of course, since there will be many proper subgraphs whose connectivity lends them to available inference algorithms. The exponential representation of MRFs introduced in Section 2.1.5 allows one to analyze such approximations as the removal of an edge is effectively accomplished by setting the associated exponential coordinate to zero.

Using an approximate distribution obtained by thinning or removing edges is common for MRFs and other probability distributions on graphs [33, 55, 31]. The exponential family representation and information geometry has been used to analyze problems in multi-terminal information theory [26] and approximations to the full MRF distribution by model-thinning [2]. Moreover, significant progress has been made in understanding Belief Propagation through a spanning tree-based analysis that uses information geometry [55]. In [31] an approximate inference algorithm for Gaussian MRFs is introduced in which a cutset is used to permit exact inference on the remaining, tractable components, while requiring that approximations be made in the inference for the cutset itself. The approximations in [31] are analyzed using information geometry.

In this chapter we will consider proxy distributions that are *reduced* MRFs, which as introduced in Section 2.1.9, are MRFs defined on proper subgraphs of the original graph. In particular, most of our attention will be on MRFs defined on the subgraph induced by the cutset U though we will also consider proper subgraphs of the induced subgraph. As a note, we mention that while it may be tempting to refer to an MRF defined on the induced subgraph as *the induced MRF*, the exponential coordinates for edges in this graph can vary leading to different MRF models. Since in this thesis the principal concern is compression, the quantities that we will be looking at and comparing between marginal and reduced MRF distributions will be those directly

concerning data compression. Namely, we will examine entropy and divergence, and have a little discussion of mutual information.

There is certainly the option of using an approximate inference method to compute the marginal distribution of the cutset. One reason against this approach is that approximate methods offer little guarantee as to what they actually compute. We prefer to compute exactly, something which is not correct, but in such a way that allows us to quantify the loss due to the approximation. This was a motivating reason for pursuing the exact inference method of Local Conditioning in the previous chapter, as opposed to one of the many approximate inference algorithms available for MRFs, and for the suboptimal lossless coding method presented in Section 6.2 of the following chapter.

In Section 5.1 we introduce the necessary background material for this chapter. In Section 5.2 we discuss monotonicity properties of MRFs with respect to the exponential parameters. In Section 5.3 we discuss reduced Markov random fields, in particular a Pythagorean decomposition comparisons between marginal and reduced MRF entropies. The proofs are given either inline or in the last section of the chapter.

5.1 Preliminaries

In this chapter we consider a family of MRFs based on a statistic t, which we assume throughout to be minimal, on a graph G. We can and do assume that G is connected, for if it were not then any MRF defined on G could be decomposed into a collection of independent MRFs each defined on a connected graph. For two exponential parameter vectors θ_1 and θ_2, we say $\theta_2 \succeq \theta_1$ if $(\theta_2)_{ij} \geq (\theta_1)_{ij}$ for all $\{i,j\} \in E$ and $\theta_2 \succ \theta_1$ if $\theta_2 \succeq \theta_1$ and $(\theta_2)_{ij} > (\theta_1)_{ij}$ for some edge $\{i,j\} \in E$. For edge $\{i,j\}$ the coordinate μ_{ij} is the expected value $\mathbb{E}_\theta\left[t_{ij}(X_i, X_j)\right]$ of the statistic t_{ij} defined on that edge, and similarly for the coordinate μ_i as the expected value of statistic t_i. By $\mathrm{cov}_\theta\left(t_{ij}, t_{kl}\right)$ we denote the covariance $\mathbb{E}_\theta\left[t_{ij}(X_i, X_j)t_{kl}(X_k, X_l)\right] -$

$\mathbb{E}_\theta\left[t_{ij}(X_i, X_j)\right]\mathbb{E}_\theta\left[t_{kl}(X_k, X_l)\right]$ between the edge statistics t_{ij} and t_{kl}, and similarly for the respective covariances $\mathrm{cov}_\theta\left(t_i, t_{kl}\right)$ and $\mathrm{cov}_\theta\left(t_i, t_j\right)$ between a node statistic t_i and an edge statistic t_{kl}, and between two node statistics t_i and t_j.

The log-partition function plays an important role in the analysis of MRFs generally [53, 54] and especially in our analysis of monotonicity that follows. It is straightforward to show the following result, which is well known [56].

Proposition V.1.

$$\begin{aligned}
\frac{\partial}{\partial\theta_{ij}}\log Q_G(\theta) &= \mu_{ij} \\
\frac{\partial}{\partial\theta_i}\log Q_G(\theta) &= \mu_i \\
\frac{\partial^2}{\partial\theta_{kl}\partial\theta_{ij}}\log Q_G(\theta) &= \mathrm{cov}_\theta\left(t_{ij}, t_{kl}\right) \\
\frac{\partial^2}{\partial\theta_{kl}\partial\theta_i}\log Q_G(\theta) &= \mathrm{cov}_\theta\left(t_i, t_{kl}\right) \\
\frac{\partial^2}{\partial\theta_j\partial\theta_i}\log Q_G(\theta) &= \mathrm{cov}_\theta\left(t_i, t_j\right)
\end{aligned}$$

Since this shows that the Hessian of the log-partition function $\Phi(\theta)\stackrel{\Delta}{=}\log Q_G(\theta)$ is a covariance matrix, and therefore nonnegative definite, we conclude from the above that the log-partition function is convex, in that for all θ_1, θ_2 and $\lambda \in [0, 1]$, we have $\Phi(\lambda\theta_1 + (1-\lambda)\theta_2) \leq \lambda\Phi(\theta_1) + (1-\lambda)\Phi(\theta_2)$. Since t is minimal, it is known that $\Phi(\cdot)$ is *strictly* convex, in that the above inequality is strict for all θ_1, θ_2 [4]. The *gradient inequality* for a convex functions shows that for all θ_1, θ_2

$$\Phi(\theta_1) \;\geq\; \Phi(\theta_2) + \nabla\Phi(\theta_2)^T(\theta_1 - \theta_2) \stackrel{\Delta}{=} \hat{\Phi}_{\theta_2}(\theta_1),$$

where we recognize $\hat{\Phi}_{\theta_2}(\theta_1)$ as the first-order Taylor series expansion of $\Phi(\cdot)$ about the point θ_2. In other words, the gradient inequality states that the Taylor series

approximation is a global underestimator of $\Phi(\cdot)$. The gradient inequality is strict for $\theta_1 \neq \theta_2$ if $\Phi(\cdot)$ is strictly convex.

A statistic t is said to be *positively correlated* if for $\theta \succ 0$ the MRF induced by θ has cov $(t_{ij}, t_{lk}) \geq 0$, cov $(t_{ij}, t_m) \geq 0$, and cov $(t_m, t_n) \geq 0$, for all $\{i,j\}, \{l,k\} \in E$ and all $n, m \in V$. Again, since we assume t to be minimal, the preceding covariance inequalitites are strict for a positively correlated statistic t. For example, the statistic used to generate an Ising model with no external field is positively correlated [24]. Since the entries of the Hessian $\nabla^2\Phi(\theta)$ are covariances between the t_{ij}'s, between the t_i's, and between the t_i's and t_{ij}'s, if t is positively correlated, then for $\theta_2 \succ \theta_1 \succeq 0$, $\nabla\Phi(\theta_2) \succeq \nabla\Phi(\theta_1)$. In other words, the moment parameter μ is monotone increasing in the exponential parameter θ.

5.2 Monotonicity

In this section we consider how properties of an MRF vary when making monotone adjustments to the exponential coordinates assigned as weights to the nodes and edges of the corresponding graph. It is intuitive, for instance, that for an MRF based on a positively correlated statistic t, increasing the exponential parameter on an edge $\{i,j\}$ will "pull" the pair of random variables X_i, X_j towards the configurations x_i', x_j' that maximize the value of the statistic $t_{ij}(\cdot,\cdot)$. For example, for a given i and x_i', for each $k \in \partial i \setminus j$, the positive correlation of the statistic is going to draw the value of each k towards that value that maximizes the value of the statistic $t_{ik}(x_i', \cdot)$, and so on throughout the network. Such biasing of the random variables decreases the entropy. This effect would be even more pronounced if we increase the exponential coordinates associated with multiple edges. Likewise, increasing the exponential coordinate associated with the statistic for an individual node i is going to pull the random variable X_i towards the particular value or values that maximize t_i and have a similar propagating effect on the bias of the remaining nodes in the

network. Now looking at the divergence between an original MRF and another, it is reasonable to assume that by "moving" the second MRF even "further away", for instance by increasing the differences in the exponential coordinates, we should get an increase in the divergence. Finally, if we consider two subsets of the random variables that partition the graph, we would expect that increasing the weight of an edge connecting the two would result in an increase in the mutual information between the two subsets of random variables.

5.2.1 Entropy

In this subsection we show that the entropy of a family of MRFs is monotone increasing in the exponential parameters. As developers of source codes we have an obvious interest in the entropy of an MRF, as it gives the lower bound to all lossless coding schemes for the model. For example, by thinning a model to something that is more tractable[2], we can use the entropy of the simplified model as a substitute or approximation to that of the original joint distribution. By showing that entropy is monotone decreasing in the exponential parameters, we know that the entropy of a thinned model is an upper bound to the entropy of the original MRF. The utility of monotonicity is further demonstrated in Section 5.3 by showing that the entropy of the reduced MRF with matching exponential parameters is greater than that of the reduced MRF with matching moment coordinates. While monotonicity is an intuitive idea, we were surprised not to have found the result in the literature. It follows directly from the observation that the entropy of an MRF induced by exponential parameter

[2]Though we show monotonicity with respect to both edge and node potentials, tractability depends on the presence of edges, so it is the exponential parameters associated with edges that matter in "thinning" applications.

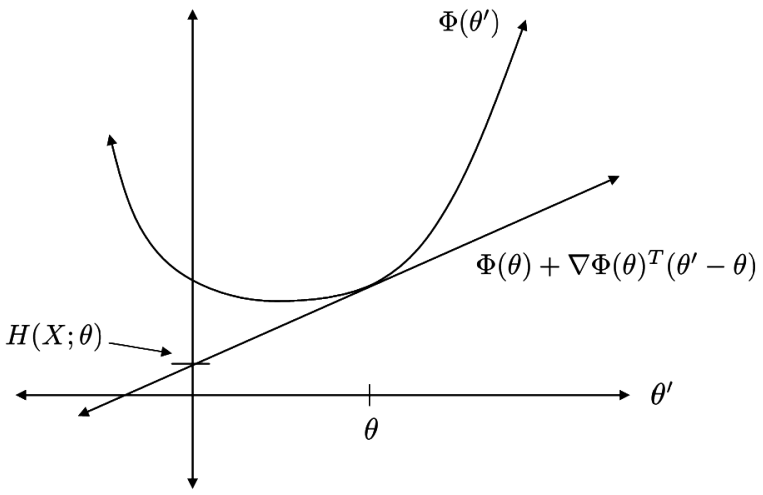


Figure 5.1: The entropy of a Markov random field can be expressed as a Taylor series approximation of a convex function $\Phi(\cdot)$.

θ is expressed as

$$\begin{aligned} H(\mathbf{X};\theta) &= \Phi(\theta) - \nabla\Phi(\theta)^T\theta \\ &= \Phi(\theta) + \nabla\Phi(\theta)^T(0-\theta), \end{aligned}$$

i.e., as the first-order Taylor series expansion of $\Phi(\cdot)$ evaluated at the point 0. This is illustrated in Figure 5.1. The following is a new information inequality for Markov random fields.

Theorem V.2 (Monotonicity of Entropy)**.** *Let $X \sim p(\cdot;\theta)$ be an MRF on a graph G based on positively correlated statistic t. Then, for $\theta_1, \theta_2 \in \Theta$, $0 \preceq \theta_1 \prec \theta_2$, we have that*

$$H_G(X;\theta_1) > H_G(X;\theta_2).$$

Proof. Assume t is positively correlated. Then, as explained earlier, we know that

for $\theta_2 \succeq \theta_1 \succeq 0$, $\nabla\Phi(\theta_2) \succeq \nabla\Phi(\theta_1)$. Therefore,

$$\begin{aligned} H(X;\theta_2) &= \Phi(\theta_2) - \nabla\Phi(\theta_2)^T\theta_2 \\ &= \Phi(\theta_2) + \nabla\Phi(\theta_2)^T(\theta_1 - \theta_2) - \nabla\Phi(\theta_2)^T\theta_1 \\ &< \Phi(\theta_1) - \nabla\Phi(\theta_2)^T\theta_1 \qquad (5.1) \\ &< \Phi(\theta_1) - \nabla\Phi(\theta_1)^T\theta_1 \qquad (5.2) \\ &= H(X;\theta_1) \end{aligned}$$

where (5.1) follows from the gradient inequality of convex functions and (5.2) from the positive correlation assumption. This demonstrates the desired monotonicity. □

The previous proof shows that monotonicity of entropy follows from both the convexity of Φ and positive correlation of the statistic t. One might wonder if the positive correlation assumption is too strict, i.e., if there are families of MRFs based on non-positively correlated statistics that also exhibit monotonicity of entropy. The following, a converse of the preceding theorem, states that positive correlation is in fact required.

Theorem V.3 (Converse: Monotonicity implies Positive Correlation)**.** *Let t be the statistic for a family of MRFs on a graph G such that $H_G(X;\theta_1) > H_G(X;\theta_2)$ for $0 \preceq \theta_1 \prec \theta_2$. Then, t is positively correlated.*

Proof. By assumption the entropy is monotone decreasing in the exponential coordinates. Therefore, it is decreasing with respect to an incremental increase in a single

exponential coordinate. Thus, we have that for any $\theta \succ 0$,

$$\begin{aligned} 0 &> \frac{\partial}{\partial \theta_{ij}} H(X;\theta) \\ &= \frac{\partial}{\partial \theta_{ij}} \Phi(\theta) - \frac{\partial}{\partial \theta_{ij}} \sum_{m \in V} \theta_m \mu_m - \frac{\partial}{\partial \theta_{ij}} \sum_{\{k,l\}} \theta_{kl} \mu_{kl} \\ &= \mu_{ij} - \sum_{m \in V} \theta_i \text{cov}\ (t_{ij}, t_m) - \sum_{\{k,l\} \neq \{i,j\}} \theta_{kl} \text{cov}\ (t_{ij}, t_{kl}) - (\mu_{ij} + \theta_{ij} \text{cov}\ (t_{ij}, t_{ij})) \\ &= -\sum_{m \in E} \theta_m \text{cov}\ \ (t_{ij}, t_m) - \sum_{\{k,l\} \in E} \theta_{kl} \text{cov}\ \ (t_{ij}, t_{kl})\,. \end{aligned} \tag{5.3}$$

Since this holds for arbitrary values of θ each of the covariances in (5.3) must be positive. Likewise, for the exponential coordinate on the statistic of an individual node,

$$\begin{aligned} 0 &> \frac{\partial}{\partial \theta_i} H(X;\theta) \\ &= -\sum_{m \in V} \theta_m \text{cov}\ \ (t_i, t_m) - \sum_{\{k,l\} \in E} \theta_{kl} \text{cov}\ \ (t_i, t_{kl}) \end{aligned} \tag{5.4}$$

implies that each of the covariances in (5.4) must also be positive. Since the above arguments hold for an arbitrary edge $\{i, j\}$ and arbitrary node i, the statistic t is positively correlated. □

The previous two theorems provide a characterization of those families of MRFs in which entropy is monotone decreasing in the exponential parameters. To say that increasing the weights on the edges and nodes of an MRF increases the coupling of the random variables and the bias of the node distributions, and thus decreases the entropy, is to say that the components of the statistic on which the MRF is based all have positive covariance with one another. Thus the family of Ising models is not only the prototypical example of a family of MRFs, but also of one exhibiting monotonicity of entropy.

Example V.4. *Let T_N be an Ising tree on N nodes with edge weights $\{\theta_{ij}\}$ and no*

external field. The entropy, denoted by $H_T^N(X;\theta)$*, can be shown to satisfy (expressions for* Q *and* μ *can be found in [5])*

$$\begin{aligned} H_T^N(X;\theta) &= \ln\left[2^N \prod_{\{i,j\}\in E} \cosh\theta_{ij}\right] - \sum_{\{i,j\}\in E} \theta_{ij}\tanh\theta_{ij} && (5.5)\\ &= N\ln 2 + \sum_{\{i,j\}\in E} [\ln\cosh\theta_{ij} - \theta_{ij}\tanh\theta_{ij}] && (5.6)\\ &= N\ln 2 + \sum_{\{i,j\}\in E} c(\theta_{ij}), && (5.7) \end{aligned}$$

where

$$c(\theta_{ij}) = \ln\cosh\theta_{ij} + \theta_{ij}\tanh\theta_{ij}.$$

To see the effect on entropy of increasing the weight θ_{ij} *attached to a given edge we plot* $c(\theta_{ij})$ *as a function of* θ_{ij}*, shown in Figure 5.2. From the plot one can see that if all other exponential parameters are kept the same, the entropy decreases as* θ_{ij} *is increased. Moreover, it can be shown that the derivative of* $c(\theta_{ij})$ *with respect to* θ_{ij} *is negative. From Figure 5.2 we can also see that* $c(\theta_{ij})$ *is nonpositive. This makes sense as the first term on in (5.6) is the entropy of an independent equiprobable distribution. Thus each summand* $c(\theta_{ij})$ *is the reduction in entroy by connecting nodes* i *and* j *by an edge with weight* θ_{ij}*.*

An immediate application of Monotonicity is that we can upper bound the entropy of a Markov random field by setting some of the exponential parameters to zero, which in graphical terms amounts to removing edges from the graph.

Corollary V.5. *Let* $\theta \in \Theta$ *be an exponential parameter vector for an MRF on* $G = (V,E)$*, let* A *be a subset of edges of* E*, and let* $\bar{\theta} = (\theta_{E\setminus A}, 0)$ *be the coordinate vector obtained by setting the components for edges in* A *to zero. Then,*

$$H_G(X;\theta) < H_G(X;\bar{\theta}).$$

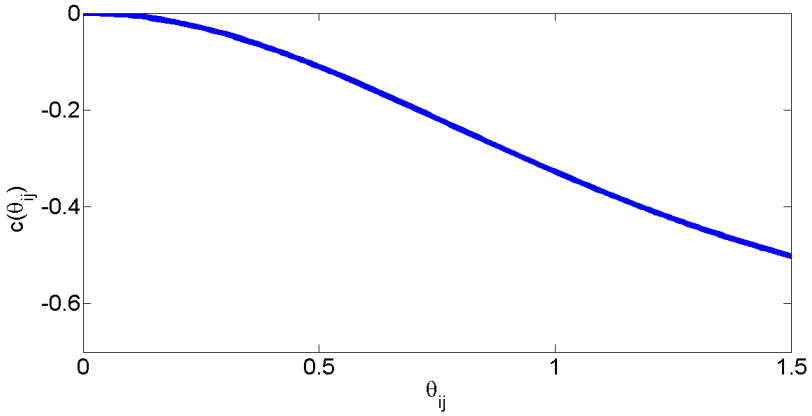


Figure 5.2: A plot of $c(\theta_{ij})$.

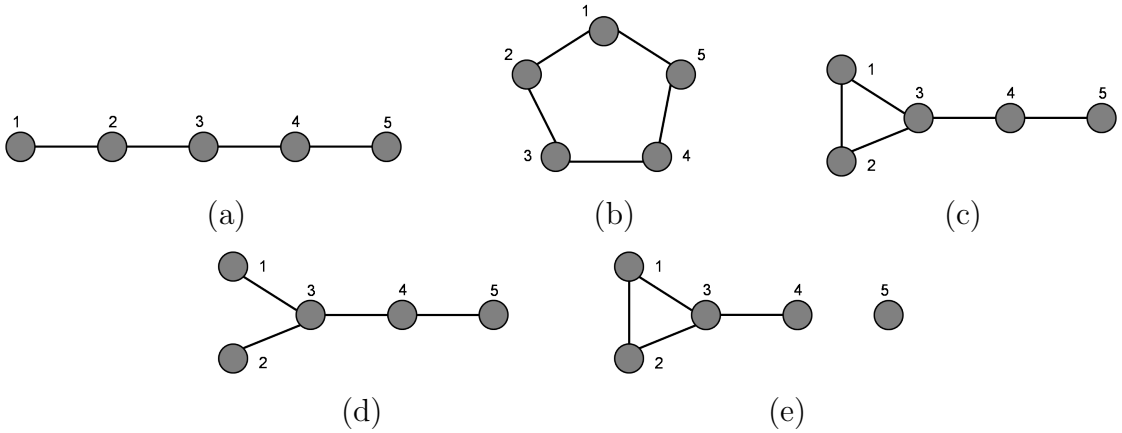


Figure 5.3: Different graphs on five nodes.

Obviously, monotonicity can also be used to analyze situations where an edge is added to a graph. We have seen theoretically and through an example that increasing the weight of an edge in an MRF results in a new MRF with lower entropy. However, how much the entropy decreases depends in large part on the significance of the edge in coupling the random variables.

Example V.6. *Let C denote a graph with M nodes $\{1, 2, \ldots, M\}$ connected successively in a cycle. It can be shown (for example, by substituting expressions for Q and*

μ *found in [5]) that the entropy of a cycle* C *on* M *nodes is*

$$H_C^M(X;\theta) = M\ln 2 + \ln\left[\prod_{i,i+1}\cosh\theta_{i,i+1} + \prod_{i,i+1}\sinh\theta_{i,i+1}\right] - \frac{\sum_{i,i+1}\theta_{i,i+1}\left[\cosh\theta_{i,i+1}\prod_{j\neq i}\sinh\theta_{j,j+1} + \sinh\theta_{i,i+1}\prod_{j\neq i}\cosh\theta_{j,j+1}\right]}{2\left[\prod_{i,i+1}\cosh\theta_{i,i+1} + \prod_{i,i+1}\sinh\theta_{i,i+1}\right]}.$$

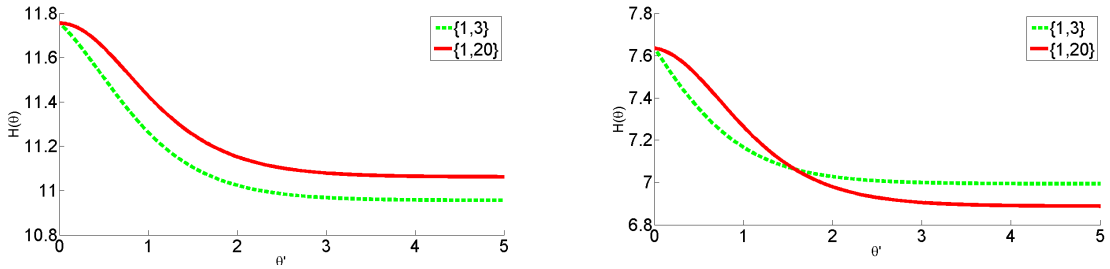
One can therefore show that the entropy of an Ising model on a single cycle graph *consisting of a single cycle* C *connected to a tree* T *is (proof given in Section 5.4)*

$$H_{C,T}^{M,N-M} = H_C^M + H_T^{N-M} - \ln 2, \tag{5.8}$$

where the cycle C *consists of* M *nodes, the tree* T *consists of* $N-M$ *nodes, and they intersect at one node so that the total number of nodes is* $N-1$*. It does not matter whether they intersect at a leaf or interior node of the tree.*

Now consider a chain graph of N *nodes, for example as shown in Figure 5.3 (a), with a homogeneous Ising model defined with common edge weight* θ*. Now consider adding an edge between node 1 and either node* N *or node* 3*, resulting respectively in, for example, the graphs in (b) and (c). In Figure 5.4 we show plots of the entropy of the resulting Ising models when* $N = 20$ *and the new edge weight* θ' *varies from 0 to 5. In (a)* $\theta = 0.5$ *on the original edges and in (b)* $\theta = 1$ *on the original edges. We can see that when the original parameter* θ *is the larger value 1 and* θ' *is small, forming the tighter 3-node cycle lowers the entropy more than forming the* 20*-node cycle. But for larger values of* θ'*, the entropy of the 20-node cycle becomes lower. For the lower value* 0.5 *of* θ*, the 3-node cycle has lower entropy than the 20-node cycle for all values of the new edge weight. As* $\theta' \longrightarrow \infty$*, the edge connecting either* $\{1,3\}$ *or* $\{1,20\}$ *the edge effectively contracts until the endpoints are identically equal. In*

the model with the 3-node cycle, then, as $\theta' \longrightarrow \infty$ *nodes 1 and 3 effectively become the same node. Since each is already connected to node 2, the entropy tends to that of a 19-node chain where all edges have weight* θ *and one edge (at the beginning of the chain) has edge weight* 2θ*. For the 20-node cycle, as* $\theta' \longrightarrow \infty$ *the entropy tends to that of a 19-node cycle with uniform edge weight* θ*. An explanation for the different behavior observed in the plots is that for lower values of* θ *having an edge with twice the weight lowers the entropy more, while at higher values of* θ *the increased weight on an edge matters less in terms of decrease in entropy.*



(a) (b)

Figure 5.4: Plots of entropy adding edge for (a) $\theta = 0.5$; (b) $\theta = 1$

Example V.7. *This example is similar to Example V.6 except that here we begin with the graph created by adding the edge* $\{1, 3\}$ *in the example above, for example, as shown in Figure 5.3 (c). This time we consider decreasing the weight of either edge* $\{1, 3\}$ *or edge* $\{N - 1, N\}$*, until the edge is effectively removed with* $\theta' = 0$*. The effects on the entropy of the resulting graph by varying edge weight associated with either of these two edges from the common edge weight* θ *to 0 are shown in Figure 5.5 for different values of the uniform weight* θ*. We can see that for* $\theta = .5$*, lowering the weight of edge* $\{1, 3\}$ *results in a greater increase in entropy that lowering the coordinate for edge* $\{19, 20\}$*. For* $\theta = 1$*, lowering the weight for edge* $\{19, 20\}$ *increases the entropy more than doing so for edge* $\{1, 3\}$ *up to a certain point. After this point, breaking the cycle results in a greater increase in entropy than creating the isolated node. For* $\theta = 1.5$*,*

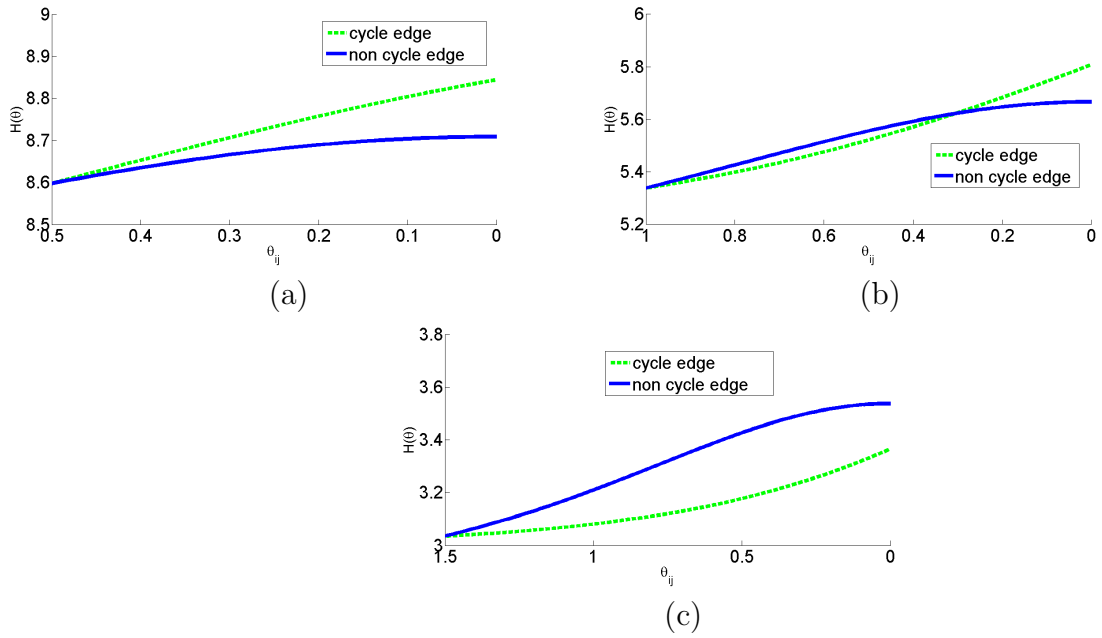


Figure 5.5: Plots of entropy adding edge for (a) $\theta = 1$; (b) $\theta = .5$; (c) $\theta = 1.5$.

the entropy is so low, that it is more beneficial (in terms of increasing entropy) to create the isolated node than to create the tree.

5.2.2 Divergence

In this section we explore the monotonicity of divergence between two MRFs. Whereas with entropy we can begin to understand how increasing and decreasing the weight on different edges leads to different decreases and increases, respectively, of the entropy, with the divergence we can look at how the adjustment to specific edges affects the divergence between a new MRF and an original MRF.

The divergence between an MRF induced by exponential parameter θ_1 and an MRF induced by parameter θ_2 is denoted in shorthand by $D(\theta_1||\theta_2)$ and can be expressed as

$$\begin{aligned} D(\theta_1||\theta_2) &= \Phi(\theta_2) - \Phi(\theta_1) + \mu_1^T(\theta_1 - \theta_2) \\ &= \Phi(\theta_2) - \hat{\Phi}_{\theta_1}(\theta_2). \end{aligned}$$

That is, the divergence between $p_G(\theta_1)$ and $p_G(\theta_2)$ is the error in the first-order Taylor series approximation of $\Phi(\theta)$ about θ_1. We have the following theorem regarding the monotonicity of divergence.

Theorem V.8. *The $\mathcal{X}$ be a Markov random field based on a positively correlated, minimal statistic t. For arbitrary $\theta_1 \prec \theta_2 \prec \theta_3$, we have*

$$D(\theta_1||\theta_2) < D(\theta_1||\theta_3). \tag{5.9}$$

Proof. Taking the gradient of $D(\theta_1||\theta)$ with respect to θ gives

$$\nabla D(\theta_1||\theta) = \mu - \mu_1,$$

where μ and μ_1 are the moment coordinates for exponential parameters θ and θ_1, respectively. Since for a positively correlated and minimal statistic t, the gradient of $\Phi(\theta)$, which equals the moment coordinates μ, is component-wise increasing, this shows that the divergence is as well, in the second argument. □

Since in source coding scenarios the first argument of $D(\cdot||\cdot)$ corresponds to the original or true model and the second argument to the proxy, the theorem as stated aligns with the case of approximating an original MRF by a "heavier" proxy model and by considering monotone increases in divergence by going to an even heavier proxy model. The typical situation, of course, is the reverse: modeling the original MRF by a lighter model and asking what happens by going to an even lighter model. The theorem holds in this direction as well, for if $\theta \prec \theta_1$, then for a positively correlated and minimal statistic, $\mu \prec \mu_1$ so the gradient of $D(\theta_1||\theta)$ with respect to θ is component-wise negative, which means that decreasing θ results in greater divergence.

Example V.9. *Consider the graphs in Figure 5.6 (a), (b), (c), and (e). There is*

a uniform Ising distribution with common edge weight $\theta = .5$ in all three cases. By comparing the first four lines or the last three lines of Table 5.1, we can see that the divergence increases as additional edges are removed from the graph.

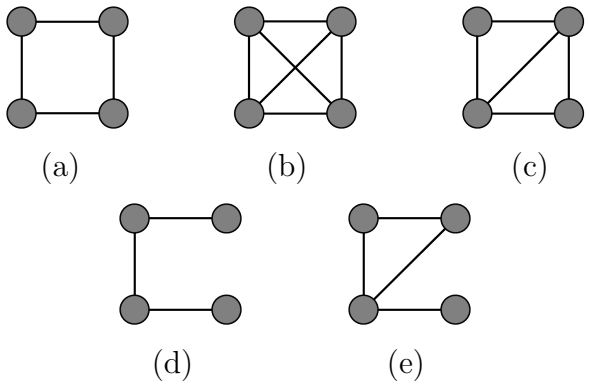


Figure 5.6: Graphs with different edge sets on four nodes.

Example V.10. *Consider again the graphs in Figure 5.6. Looking at the divergences $D(c||a)$ and $D(c||e)$ suggests that the relative entropy from a given MRF to a second MRF whose edge set (exponential parameters) contains that of the first depends on the location of the removed edge(s). For instance, an explanation for why the divergence $D(c||a)$ is smaller than $D(c||e)$ is that the graph in (e) has a tree component that makes it "farther" from (c) than (a), in which all nodes are bound in a cycle.*

It is well-known that relative entropy is asymmetric. In the case of MRFs, where the exponential parameters associated with the two MRFs are related through component-wise inequality, it is interesting to consider whether there is a pattern to the asymmetry. In the example graphs considered above, we use the same edge weight $\theta = .5$ on all edges, so the divergences in some sense are measuring the contribution of additional or missing edges in the "distance" from one MRF to another. Observing that for two graphs where one edge set is contained in the other, the relative entropy from the "smaller" MRF to the "larger" MRF is greater than that from "larger" to "smaller". This leads the author to offer the following conjecture.

θ_1	θ_2	$D(\theta_1 \| \| \theta_2)$	$D(\theta_2 \| \| \theta_1)$
(b)	(c)	0.0902	0.1121
(b)	(a)	0.2321	0.3079
(b)	(e)	0.3035	.4158
(b)	(d)	0.5591	0.7556
(c)	(a)	0.1055	0.1281
(c)	(d)	0.3438	0.4351
(c)	(e)	0.1245	0.1463

Table 5.1: Divergences between pairs of graphs from Figure 5.6.

Conjecture V.11. *Let* $\mathbf{X}$ *be a minimal, positively correlated MRF. If* $\theta_1 \prec \theta_2$*, then* $D(\theta_1||\theta_2) > D(\theta_2||\theta_1)$*.*

If this is true then it offers some insight into the asymmetry of relative entropy. This can be understood in an extreme case. Suppose the coordinates of θ_2 are very large, much larger than those of θ_1. In this case, there will be many configurations $\mathbf{x}$ for which $p_G(\theta_2)$ is very close to zero, due to the high coupling of the random variables. For such configurations, the term $\log p_G(\mathbf{x};\theta_1)/p_G(\mathbf{x};\theta_2)$ is going to be very large and is going to dominate the terms contributing to the two divergences in question.

5.2.3 Mutual Information

It is just as intuitive that for such an MRF model, decreasing the exponential coordinate for an edge $\{i, j\}$, where $i \in B$ and $j \in \partial B$, should decrease the mutual information between X_B and $X_{\partial B}$. We let $I(X_B; X_{\partial B}||\theta)$ denote the mutual information between the random subfields X_B and $X_{\partial B}$ in the MRF distribution induced by exponential parameter θ. Then, it can be shown that

$$\begin{aligned} I(X_B; X_{\partial B}; \theta) &= D(p_G(\theta) \;||\; p_G^B(\theta) p_G^{\partial B}(\theta)) \\ &= \log Q_G(\theta) + \mathbb{E}_\theta \left[\log \frac{\bar{\Psi}_{B,\partial B}}{m_{B\to\partial B} m_{\partial B \to B}} \right], \end{aligned}$$

where

$$\begin{aligned}\bar{\Psi}_{U,\partial U}(x_U, x_{\partial U}) &= \prod_{\substack{\{i,j\}\in E \\ i\in U, j\in\partial U}} \Psi_{ij}(x_i, x_j) \\ &= \prod_{\substack{\{i,j\}\in E \\ i\in U, j\in\partial U}} \exp\{\theta_{ij} t_{ij}(x_i, x_j)\} \\ &= \exp\{\sum_{k\in E_{U,\partial U}} \theta_k t_k(x_k)\},\end{aligned}$$

and for subsets $A, B \subset V$,

$$m_{A\to B}(x_B) = \sum_{x'_A} \bar{\Psi}_{A,B}(x'_A, x_B) \prod_{i\in A} \Phi_i(x_i).$$

Conjecture V.12. *For a minimal, positively correlated MRF, the mutual information is monotone increasing in the exponential parameter.*

5.3 Reduced Markov Random Fields

In this section we consider an MRF based on a (minimal) statistic t, whose components correspond to subsets of the nodes and edges of the graph G. For a given subset $U \subset V$ of sites, we are interested in MRF distributions for X_U. As mentioned in Section 2.1.9, for a subset U of sites the subvectors t_U, θ_U, and μ_U correspond, respectively, to the statistic components, exponential coordinates, and moment coordinates for nodes and edges contained within the subset U. A reduced MRF on the subgraph G_U based on t_U is expressed in terms of exponential coordinates $\theta_U \in \Theta(G)$ as

$$p_{G_U}(x_U; \theta_U) = \exp\{\langle \theta_U, t_U \rangle - \Phi_U(\theta_U)\}. \tag{5.10}$$

Alternatively, it can be indexed by the moment coordinates as $\mu_U \in \mathcal{M}(G_U)$. Since t

is minimal for the family of MRFs on G, the subvector t_U is minimal for the family of MRFs $\mathcal{F}(G_U)$ on the induced subgraph G_U, so $\Theta(G_U)$ and $\mathcal{M}(G_U)$ are dual coordinate systems for MRFs on G_U. For $\mu_U \in \mathcal{M}(G_U)$, the reduced MRF on G_U is denoted by $p_{G_U}(\mu_U)$.

The first step in analyzing reduced MRFs on G_U is to isolate the induced subgraph G_U from the remainder of the graph. The reason for isolating G_U is that we want to consider MRFs on G_U alone, with no edges connecting it to other nodes. Since θ is a coordinate system for the collection of MRFs $\mathcal{F}$ on G based on statistic t, we can isolate G_U by setting to zero those exponential coordinates corresponding to edges one of whose endpoints is in U, the other not. However, to simplify the coming analysis we choose to set to zero the exponential coordinates $\backslash\theta_U$ for all nodes and edges outside of U. That is, we consider the *e-flat submanifold* [3]

$$\mathcal{F}'_U(0) \;=\; \{p' \in \mathcal{F} : \backslash\theta'_U = 0\}$$

of MRFs on G. Because of the zeroed out exponential coordinates, this e-flat submanifold can be seen as the family of MRF on the induced subgraph G_U together with $\mid V \backslash U \mid$ isolated nodes, where the isolated nodes are independent and uniformly distributed.

5.3.1 Moment-matching Reduced MRF

Given an MRF p_G on G, it is not immediately clear whether the subvector μ_U corresponding to moment coordinates for nodes and edges inside a subset U is a valid moment parameter for MRFs on G_U based on t_U, i.e., it is not clear if it is in $\mathcal{M}(G_U)$. We now use the fact that the moment parameters μ provide a coordinate system for $\mathcal{F}$ and consider the *m-flat submanifold* [3]

$$\mathcal{F}''_U(\mu_U) \;=\; \{p'' \in \mathcal{F} : \mu''_U = \mu_U\}$$

of all MRFs whose moment coordinates μ'' for edges and nodes in U are equal to the corresponding coordinates μ_U from the original MRF $p \sim \mu$. In partition form, an MRF $p_G(\cdot; \mu'') \in \mathcal{F}''_U$ is parameterized by a moment vector $\mu'' = (\mu_U, \backslash\mu''_U)$.

In the information geometry literature, the submanifolds $\mathcal{F}'_U$ and $\mathcal{F}''_U$ are known as *orthogonal* submanifolds [3, 2] because the retained moment coordinates and zeroed out exponential coordinates partition the set of coordinate indices. It follows that $\mathcal{F}'_U$ and $\mathcal{F}''_U$ intersect uniquely at an MRF p^*_G with mixed coordinates $(\mu_U, \backslash\theta^*_U = 0)$ and exponential coordinates $\theta^*(\mu) = (\theta^*_U(\mu), 0)$. This is called the *m-projection* of $p_G(\cdot; \theta)$ onto the the thinned model [3, 2]. Figure 5.7 gives a schematic of the manifold $\mathcal{F}$ of MRFs on G based on t and submanifolds $\mathcal{F}''_U$ and $\mathcal{F}'_U$. Though $p^*_G = p_G(\cdot; \theta^*)$ is defined on the original graph G, the subfields X_U and $X_{V\backslash U}$ are independent under p^*_G, so that

$$p^U_G(X_U; \theta^*) = p_{G_U}(X_U; \theta^*_U).$$

Since t is minimal for the original family of MRFs, t_U is minimal for reduced MRFs on G_U, and in this case, for any MRF on G, $\mu_U \in \mathcal{M}(G_U)$, as these are the unique moment coordinates to induce the reduced MRF distribution $p_{G_U}(\theta^*)$, thus making clear what was not clear earlier, that indeed μ_U is a valid moment parameter for an MRF on G_U.

5.3.2 Pythagorean Decomposition for Reduced MRFs

Let θ' be an exponential parameter for an MRF in $\mathcal{F}'_U(0)$, and given some MRF in $\mathcal{F}$ with moment parameter μ, let θ^* be the moment-matching exponential parameter for the MRF in the intersection of $\mathcal{F}'_U(0)$ and $\mathcal{F}''_U(\mu_U)$. Then for any θ inducing an

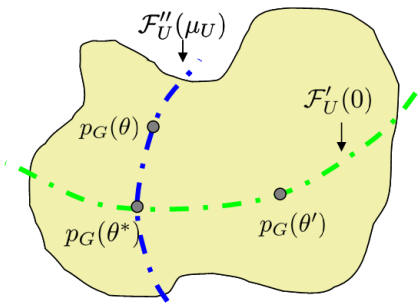


Figure 5.7: Orthogonal submanifolds $\mathcal{F}'_U(0)$ and $\mathcal{F}''_U(\mu_U)$ in manifold $\mathcal{F}$ of MRFs on G.

MRF in $\mathcal{F}$, the divergence between $p_G(\theta)$ and $p_G(\theta')$ can be decomposed as [2]

$$D(p_G(\theta)||p_G(\theta')) = D(p_G(\theta)||p_G(\theta^*)) + D(p_G(\theta^*)||p_G(\theta')). \tag{5.11}$$

This is a well-known Pythagorean relation of information geometry. This shows that of all MRFs $p(\theta')$ in the e-flat submanifold $\mathcal{F}'_U$ the one with minimum reverse divergence with the original MRF $p(\theta)$ is the moment-matching MRF $p(\theta^*)$.

We can use the above Pythagorean formula to simplify the divergence between $p^U_G(\theta)$ and $p_{G_U}(\theta'_U)$ by first noting that MRFs in $\mathcal{F}'_U$ have a particularly simple form. The MRF $p_G(\theta')$ induced by θ' is simply the product of the reduced MRF $p_{G_U}(\theta'_U)$ and the product of the independent distributions for the isolated nodes. Given a second MRF $p_G(\theta'')$ in $\mathcal{F}'_U$, the divergence between $p_G(\theta')$ and $p_G(\theta'')$ can be decomposed as the divergence between the reduced MRFs $p_{G_U}(\theta'_U)$ and $p_{G_U}(\theta''_U)$, plus the divergence between the two independent parts. But the independent parts are identical so the divergence is simply the divergence between the two reduced MRFs, as summarized in the following lemma.

Lemma V.13. *Let $\theta', \theta'' \in \mathcal{F}'_U$. Then,*

$$D_G(\theta'||\theta'') = D_{G_U}(\theta'_U||\theta''_U). \tag{5.12}$$

(a) (b)

Figure 5.8: (a) A graph with subsets indicated in yellow and red; (b) induced subgraphs for respective subsets.

Because of their simple form, we can decompose the entropy of an MRF $p_G(\theta')$ in the e-flat submanifold $\mathcal{F}'_U$ in the following way.

Lemma V.14. *Let θ' induce an MRF in $\mathcal{F}'_U$. Then,*

$$H_G(\theta') = H_{G_U}(\theta'_U) + |V \setminus U| \log |\mathcal{X}|. \tag{5.13}$$

Again owing to the simple form of MRFs in $\mathcal{F}'_U$, we can decompose the divergence between the original MRF $p_G(\theta)$ and an MRF $p_G(\theta')$ in $\mathcal{F}'_U$ as follows.

Lemma V.15. *Let θ be given, and let θ' induce an MRF in $\mathcal{F}'_U$. Then,*

$$D(\theta||\theta') = D(p_G^U(\theta)||p_{G_U}^U(\theta'_U)) + |V \setminus U| \log |\mathcal{X}| - H_G^{V\setminus U|U}(\theta). \tag{5.14}$$

We substitute the decompositions from the above two lemmas into (5.11) to derive a decomposition for the divergence between the marginal distribution $p_G^U(\theta)$ and the reduced MRF $p_{G_U}(\theta'_U)$.

Theorem V.16. *Let θ induce an MRF in $\mathcal{F}$ and let μ be the corresponding moment coordinate vector and let θ' be an exponential coordinate vector for an MRF in $\mathcal{F}'_U(0)$. Furthermore, let U be a nonempty subset of sites. Then*

$$D(p_G^U(\theta)||p_{G_U}^U(\theta'_U)) = D(p_G^U(\theta)||p_{G_U}^U(\mu_U)) + D(p_{G_U}(\mu_U)||p_{G_U}(\theta'_U)). \tag{5.15}$$

(a) (b)

Figure 5.9: (a) A graph with subsets indicated in yellow and red; (b) induced subgraphs for respective subsets.

5.3.3 Reduced MRF Entropy

In Section 5.2.1 we showed that removing edges from a graph, by way of setting the corresponding exponential coordinates to zero, increases the entropy of an MRF. In this section we again assume that θ induces and original MRF with corresponding moment parameter μ, and we consider a subset U of V and show that

$$H_{G_U}(\theta_U) \geq H_G^U(\theta). \tag{5.16}$$

In other words, we show that the reduced MRF entropy $H_{G_U}(\theta_U)$ is an upper bound to the marginal entropy $H_G^U(\theta)$. That is, isolating G_U by eliminating edges connected to U cannot decrease the entropy of X_U. We show this in two steps – first showing that $H_{G_U}(\mu_U) \geq H_G^U(\theta)$, and then demonstrating that $H_{G_U}(\theta_U) \geq H_{G_U}(\mu_U)$.

Given an exponential parameter θ, let $\theta' = (\theta_U, 0)$ be the exponential parameter obtained by setting coordinates for edges outside of U to zero. We know from Theorem V.3 that $H_G(\theta') > H_G(\theta)$, however this does not give us a comparison between $H_{G_U}(\theta'_U)$ and $H_G^U(\theta)$. Since μ_U is a valid moment parameter for reduced MRFs on G_U, the first step is accomplished by using the well-known maximum entropy principle for exponential families [9], a slight variation of which is given below.

Proposition V.17 (Maximum Entropy). Let $U \subset V$ be a subset of nodes, let $\mu_U \in \mathcal{M}(G_U)$, and let $\mathcal{P}_{\mu_U}$ be the set of (not necessarily Markov) probability distributions

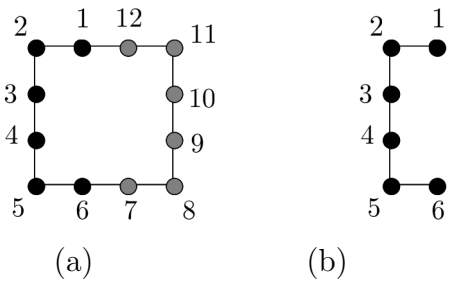


Figure 5.10: (a) Original cycle on which Ising model defined. Subset U indicated in black. (b) Induced subgraph G_U on which upper bound $H_{G_U}(X_U; \theta_U)$ is based.

on X_U satisfying

$$\mathbb{E}_p\left[t_U(X)\right] = \mu_U.$$

If $p \in \mathcal{P}_{\mu_U}$, then

$$H_p(X_U) \leq H_{G_U}(X_U; \mu_U),$$

with equality if and only if $p = p_{G_U}(X_U; \mu_U)$.

We now conclude from this proposition that $H_{G_U}(\mu_U) \geq H_G^U(\theta)$, which completes the first step.

To do the second and final step we show that the exponential coordinate vector θ_U^* for the moment-matching MRF on G_U is component-wise larger than the subvector θ_U of the original exponential parameter. This is stated in the following theorem.

Theorem V.18. *Let $G = (V, E)$ be an undirected graph, let t be a positively correlated, minimal statistic for MRFs on G, and let $\theta \in \Theta$ be an exponential parameter for MRFs on G with corresponding mean parameter vector μ. For a subset of nodes $U \subset V$, let θ_U^* be the exponential parameter vector for MRFs on G_U with corresponding moment vector μ_U. Then*

$$\theta_U^* \succeq \theta_U,$$

with equality if and only if $\theta_U^* = \theta_U$.

Proof. We do this by first showing that $\nabla\Phi_U(\theta_U^*) \succeq \nabla\Phi_U(\theta_U)$. Let $\theta' = (\theta_U, 0)$ be

the exponential parameter for an MRF obtained by zeroing out coordinates for nodes and edges outside of U. Then,

$$\begin{aligned} \nabla\Phi_U(\theta^*_U) &= \mu_U && (5.16)\\ &= [\nabla\Phi(\theta)]_U && (5.17)\\ &\succ [\nabla\Phi(\theta')]_U && (5.18)\\ &= \nabla\Phi_U(\theta_U) && (5.19) \end{aligned}$$

where (5.16) and (5.17) follow from the moment-matching constraint on θ^*, (5.18) follows from the fact that the gradient $\nabla\Phi(\cdot)$ is increasing for positively correlated t, and (5.19) is due to $p_G(\theta')$ being the reduced MRF on G_U with exponential parameter θ_U, times an independent part.

We now let μ'_U be the moment parameter for the MRF on G_U corresponding to exponential parameter θ_U. Consider the mapping $\Lambda : \Theta \longrightarrow \mathcal{M}$ from exponential coordinates to moment coordinates. Since $\mu_U \succ \mu'_U$, the theorem is proved if we can show that the inverse map $\Lambda^{-1} : \mathcal{M} \longrightarrow \Theta$ exists and is increasing. By the positive correlation assumption, Λ is strictly increasing which means that it is invertible, hence Λ^{-1} exists. Moreover, since the derivative of the inverse function is simply the reciprocal of the derivative of the original function, we see that indeed, Λ^{-1} is strictly increasing as well. Therefore, $\theta^*_U \succ \theta_U$ and the theorem is proved. □

We now summarize in the following theorem.

Theorem V.19. *Let* $\mathbf{X}$ *be a minimal, positively correlated Markov random field induced by exponential parameter* θ *and corresponding moment parameter* μ*. Then,*

$$H^U_G(\theta) \leq H_{G_U}(\mu_U) \leq H_{G_U}(\theta_U). \tag{5.20}$$

We now quantify the gap of the first inequality in (5.20).

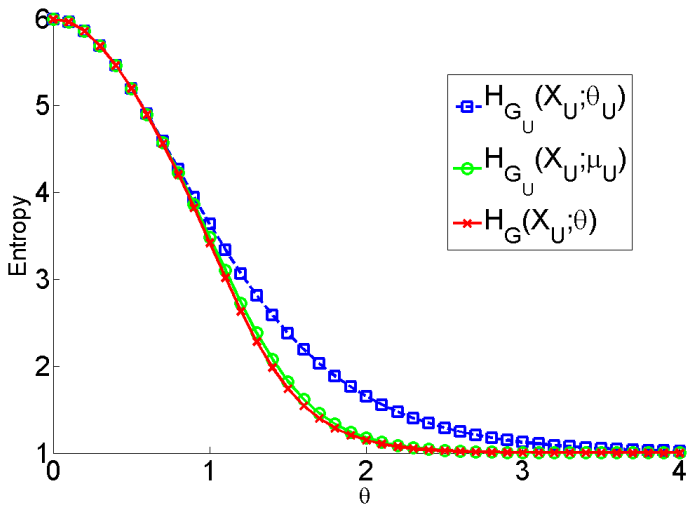


Figure 5.11: Entropy of marginal, moment-matching and exponential-matching reduced MRFs for cycle graph shown in Figure 5.10.

In the previous subsection we discussed how, given an exponential parameter θ and a cutset U, the MRF in the e-flat submanifold $\mathcal{F}'_U$ with minimum reverse divergence to $p(\theta)$ is induced by the moment-matching exponential parameter θ^*. It is well-known and straightforward to show that

$$H_G(\theta^*) = H_G(\theta) + D(p_G(\theta)||p_G(\theta^*)). \tag{5.21}$$

Substituting the results of Lemmas V.14 and V.15 into (5.21) gives us the following theorem.

Theorem V.20. *Let $G = (V, E)$ be an undirected graph on which MRF X is defined. For an arbitrary subset $U \subset V$,*

$$H_{G_U}(\mu_U) = H^U_G(\theta) + D(p^U_G(\theta)||p_{G_U}(\mu_U)). \tag{5.22}$$

5.4 Proofs

5.4.1 Proof of Equation (5.8)

Proof. We enumerate the nodes on the cycle as $1, 2, \cdots, m$, and then enumerate the nodes on the subtree in such a way that along any path toward the leaves, the numbers get bigger. Then,

$$\begin{aligned}
H(X_1, \cdots, X_{m+n}; \theta) &= H_C^m(X_1, \cdots, X_m; \beta) + \sum_{i=m+1}^{m+n} H(X_i \mid X_{\pi(i)}; \beta) \\
&= H_C^m + \sum_{i=m+1}^{m+n} H(X_i \mid X_{\pi(i)}; \beta) \\
&= H_C^m + \sum_{i=m+1}^{m+n} H_{G(\{i,j\})}(X_i \mid X_{\pi(i)}; \beta_{i\pi(i)}) \\
&= H_C^m + \sum_{\{i,j\} \in E(T)} \left[-\frac{e^{\beta_{ij}}}{e^{\beta_{ij}} + e^{-\beta_{ij}}} \ln \frac{e^{\beta_{ij}}}{e^{\beta_{ij}} + e^{-\beta_{ij}}} \right] \\
&\quad + \sum_{\{i,j\} \in E(T)} \left[-\frac{e^{-\beta_{ij}}}{e^{\beta_{ij}} + e^{-\beta_{ij}}} \ln \frac{e^{-\beta_{ij}}}{e^{\beta_{ij}} + e^{-\beta_{ij}}} \right] \\
&= H_C^m + \sum_{\{i,j\} \in E(T)} \left[\ln (2 \cosh \beta_{ij}) - \theta_{ij} \tanh \beta_{ij} \right] \\
&= H_C^m + (n-1) \left[\ln 2 + \ln (\cosh \beta_{ij}) - \beta_{ij} \tanh \beta_{ij} \right] \\
&= H_C^m + H_T^n - \ln 2
\end{aligned}$$

□

5.4.2 Proof of Lemma V.13

Proof. Let θ' and θ'' induce MRFs in the e-flat manifold $\mathcal{F}'_U$. Then,

$$\begin{aligned} D_G(\theta'||\theta'') &= \sum_{\mathbf{x}} p_G(\mathbf{x};\theta') \log \frac{p_G(\mathbf{x};\theta')}{p_G(\mathbf{x};\theta'')} \\ &= \sum_{x_U} p_{G_U}(x_U;\theta'_U) \log \frac{p_{G_U}(x_U;\theta'_U)}{p_{G_U}(x_U;\theta''_U)} + \sum_{\mathbf{x}} p_G(\mathbf{x};\theta') \log \frac{p_{G_{V\setminus U}}(x_{V\setminus U};\theta'_{V\setminus U})}{p_{G_{V\setminus U}}(x_{V\setminus U};\theta''_{V\setminus U})} \\ &= D_{G_U}(\theta'_U||\theta''_U), \end{aligned} \tag{5.23}$$

where equality (5.23) follows from the fact that θ' and θ'' agree on coordinates for $V \setminus U$. □

5.4.3 Proof of Lemma V.14

Proof. Let θ' induce an MRF in $\mathcal{F}'_U$. Then,

$$\begin{aligned} H_G(\theta') &= H_G(x_U;\theta') + H_G^{V\setminus U|U}(\theta') \\ &= H_{G_U}(\theta'_U) + |V \setminus U| \log |\mathcal{X}|, \end{aligned} \tag{5.24}$$

where equality (5.24) follows from the fact the under θ', $V \setminus U$ is independent and identically distributed with uniform distribution. □

5.4.4 Proof of Lemma V.15

Proof. Let θ be given, and let θ' induce an MRF in $\mathcal{F}'_U$. Then,

$$\begin{aligned} D(\theta||\theta') &= \sum_{\mathbf{x}} p_G(\mathbf{x};\theta) \log \frac{p_G(\mathbf{x};\theta)}{p_G(\mathbf{x};\theta')} \\ &= \sum_{x_U} p_G^U(x_U;\theta) \log \frac{p_G(x_U;\theta)}{p_{G_U}(x_U;\theta'_U)} + \sum_{\mathbf{x}} p_G(\mathbf{x};\theta') \log \frac{p_{G_{V\setminus U|U}}(x_{V\setminus U};\theta)}{p_{G_{V\setminus U}}(x_{V\setminus U};\theta'_{V\setminus U})} \\ &= D(p_G^U(\theta)||p_{G_U}^U(\theta'_U)) + |V \setminus U| \log |\mathcal{X}| - H_G^{V\setminus U|U}(\theta). \end{aligned}$$

□

CHAPTER VI

Lossless Compression of MRFs

In this chapter we discuss lossless compression of Markov random fields based on Arithmetic Encoding (AC). As discussed in Section 2.3.2, the redundancy in rate in using Arithmetic Encoding is the divergence between the original distribution and the coding distribution determined by the product of the coding distributions for the individual symbols. This means that to achieve optimal compression of an MRF $\mathbf{X}$, one must be able to perform exact inference. The connection between optimal compression and exact inference has been known for some time [42, 43], but in the context of Markov random fields this has a direct implication: since Belief Propagation is the standard method for performing inference in an MRF, it is only possible to optimally losslessly compress an MRF if the associated graph is amenable to exact inference by BP. Since most sources can be described as Markov random fields on some graph, it is interesting to note that the existence of practical, efficient algorithms for optimally losslessly compressing that source depends on how "interconnected" the source is. In this chapter we discuss optimal compression for acyclic and EASY cyclic MRFs, and introduce a method for suboptimal lossless compression of HARD cyclic MRFs.

Recall from Section 2.3.2 that Arithmetic Encoding involves forming a scan $x_1, x_2, \ldots, x_{|V|}$ of the pixel values and then for each i, computing a coding distribution

$f_i(x'_i)$, which is passed along with the pixel value x_i to the encoder. The decoder reverses the process, recomputing the coding distributions to perfectly reproduce the scan of pixel values. When the MRF is defined on an acyclic graph, optimal direct encoding can be performed in two stages. In the first, which will be referred to as the *upward* pre-encoding phase, the graph is viewed as a tree with the first scanned pixel as its root. The Belief Propagation (BP) algorithm is run, starting at the leaves, with *messages* being passed from children to parents, until messages are received at the root. These messages do not depend on the pixel values and are of the form indicated in Proposition II.4. In the second stage, the *downward* encoding phase, a *lineal scan* is chosen, which means that every parent is scanned before its children. Then, for $i = 1, \ldots, N$, the optimal coding distribution $p_{i|*}$ is computed from the upward messages into node i from its children and the value of the edge potential $\Psi_{i,\pi(i)}(x_i, \bar{x}_{\pi(i)})$ between i and its parent $x_{\pi(i)}$, where $\bar{x}_{\pi(i)}$ is the observed value of the parent node. The values in $x_1, \ldots, x_{i-1}$ other than $x_{\pi(i)}$ are irrelevant in computing $p_{i|*}$ due to the tree structure and the Markov property. Overall, the complexity is proportional to $|V|$ times the alphabet size squared. The algorithm for encoding an acyclic MRF is the basis for both the optimal and suboptimal methods for AC encoding a cyclic graph discussed in this chapter.

In an MRF defined on a cyclic graph, the cycles prevent efficient computation of the $p_{i|*}$ using the standard BP algorithm. Thus the problem of optimally compressing a cyclic MRF boils down to the familiar tactics used to perform inference in a cyclic MRF. For some graphs it may be possible to form a cluster tree with reasonable sized clusters or there may be a loop cutset and associated unwrapping where the largest relevant set is of reasonable size. Such MRFs are called *EASY*. In the former case, one can group nodes together to form a cluster tree $\widehat{G}$ and then run the acyclic encoding algorithm, and in the latter, one can choose a loop-cutset L and an unwrapped graph $\overline{G}$ based on L, then apply Local Conditioning to encode an image $\mathbf{x}$. For example, an

MRF on a 10×100 grid graph is EASY, since the columns can be grouped into clusters of size 10 and, as shown in Section 4.4, the checkerboard cutset has an unwrapping with largest relevant set size 10. In each of these methods, a lineal scan is chosen, an initial round of upward messages toward the root, followed by a downward pass in which the current node of the scan uses the upward messages from its children combined with conditional information from nodes encoded before it.

A Markov random field on a graph for which neither of the two previous approaches lead to methods with reasonable complexity is called a *HARD* MRF. For instance, an MRF on a 100×100 grid graph is HARD. For HARD cyclic Markov random fields, one can of course use the Sum-Product variant of Loopy Belief Propagation. However, as is well known [37], very little is understood about the answers, that is, beliefs, that LBP produces. Indeed, in few cases can we say exactly what LBP is computing, for example in Gaussian MRFs [59] or finite-valued MRFs defined on a single-cycle [57, 58]. For any application requiring an inference algorithm this is sure to be frustrating, but perhaps more so in the source coding problem because of the operational significance of the divergence between the true distribution and what the inference algorithm is actually computing. In Section 6.2 we propose *Reduced Cutset Coding* (RCC), a suboptimal method for HARD cyclic MRFs, in which we can explicitly quantify the divergence between the original and coding distributions. A cutset U is chosen so that the subgraph induced by the cutset and the components of $G \setminus U$ are all tractable to the clustering methods or to Local Conditioning. If the components of $G \setminus U$ are EASY, then the remaining sites $V \setminus U$ can be encoded optimally conditioned on the cutset. If the induced subgraph G_U is tractable, we can exactly compute a reduced MRF coding distribution for X_U. This is suboptimal since we would not be using the marginal distribution of X_U. However, we can quantify the redundancy and even optimize with respect to reduced MRFs on G_U. This approach of using suboptimal inference on a cutset in order to guarantee optimal performance on

the remaining components is similar to the suboptimal inference method for Gaussian MRFs in [31].

In Section 6.1 we discuss optimal coding of acyclic MRFs and EASY cyclic MRFs. In Section 6.2 we discuss the Reduced Cutset Coding method. In Section 6.3 we demonstrate the RCC method on homogeneous Ising models on both the 4 pt. and 8 pt. graphs. For these models, the performance of this method is near optimal. In Section 6.4 we model bilevel images as Ising models on a 4 pt. grid graphs and apply RCC. Proofs are in the last section of the chapter.

6.1 Optimal AC Encoding of MRFs

As can be seen from formula (2.11) for the average bitstring length produced by an AC encoder, optimal compression occurs when exact inference is possible. For MRFs defined on acyclic graphs, this is always the case. For cyclic MRFs, this is the case when defined on EASY graphs.

6.1.1 Acyclic MRFs

First, a lineal scan $1, 2, \ldots, |V|$ of the nodes is formed, and we denote by A_i the first i nodes of the scan. The optimal coding distribution for node i is then computed as

$$\begin{aligned} p_{i|*}(x_i|x_{A_{i-1}}) &= \frac{p(x_{A_i})}{p(x_{A_{i-1}})} \\ &= \frac{Z_{A_i}(x_{A_i})}{Z_{A_{i-1}}(x_{A_{i-1}})} \end{aligned}$$

using the usual messages from the BP algorithm. Since the scan is lineal, the set A_i is connected. Therefore, the beliefs in the numerator and denominator above can each be computed using a formula analogous to (2.14) [30]. However, due to the Markov property and the lineal scan, the optimal coding distribution $p_{i|*}(x_i|x_{A_i-1})$ can be

computed more simply as

$$\begin{aligned} p_{i|*}(x_i|x_{A_{i-1}}) &= \frac{p(x_{\pi(i)}, x_i)}{p(x_{\pi(i)})} \\ &= \frac{Z_{\{\pi(i),i\}}(x_{\pi(i)}, x_i)}{Z_{\pi(i)}(x_{\pi(i)})}, \end{aligned}$$

since X_i is conditionally independent of $X_{A_{i-1}} \setminus X_{\pi(i)}$ conditioned on $X_{\pi(i)}$. The following theorem shows how the $p_{i|*}$'s can be efficiently computed using the BP messages.

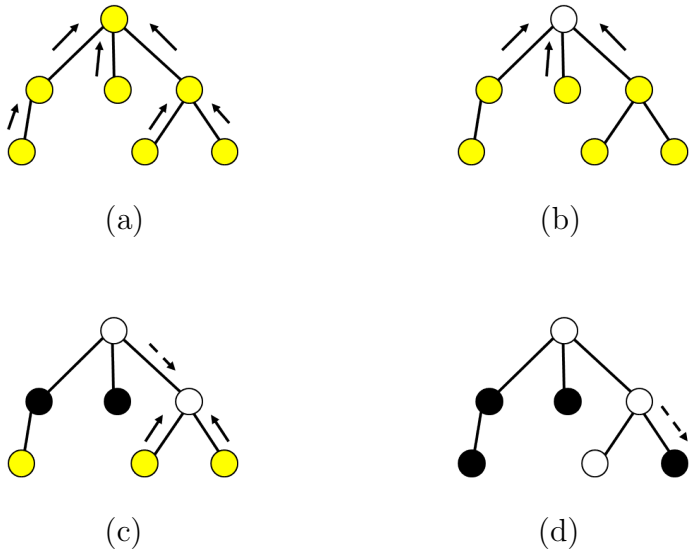


Figure 6.1: (a) Upward messages to root node; (b) Encoding root node with messages from children; (c) Continue encoding with messages from children and conditional message from parent; (d) Encoding final node of scan. The black and white nodes indicate observed pixel values.

Theorem VI.1. *Suppose the graph $G = (V, E)$ of an MRF $\mathbf{X}$ is acyclic, and suppose that a lineal scan of the nodes in G is chosen. Then,*

$$p_{i|*}(x_i|x_{A_i}) = \frac{\Psi_{\pi(i),i}(x_{\pi(i)}, x_i)\Phi_i(x_i) \prod_{j \in \sigma(i)} m_{j \to i}(x_i)}{\sum_{x'_i} \Psi_{\pi(i),i}(x_{\pi(i)}, x'_i)\Phi_i(x'_i) \prod_{j \in \sigma(i)} m_{j \to i}(x'_i)}. \tag{6.1}$$

Notice that this theorem says that the optimal coding distribution for node i

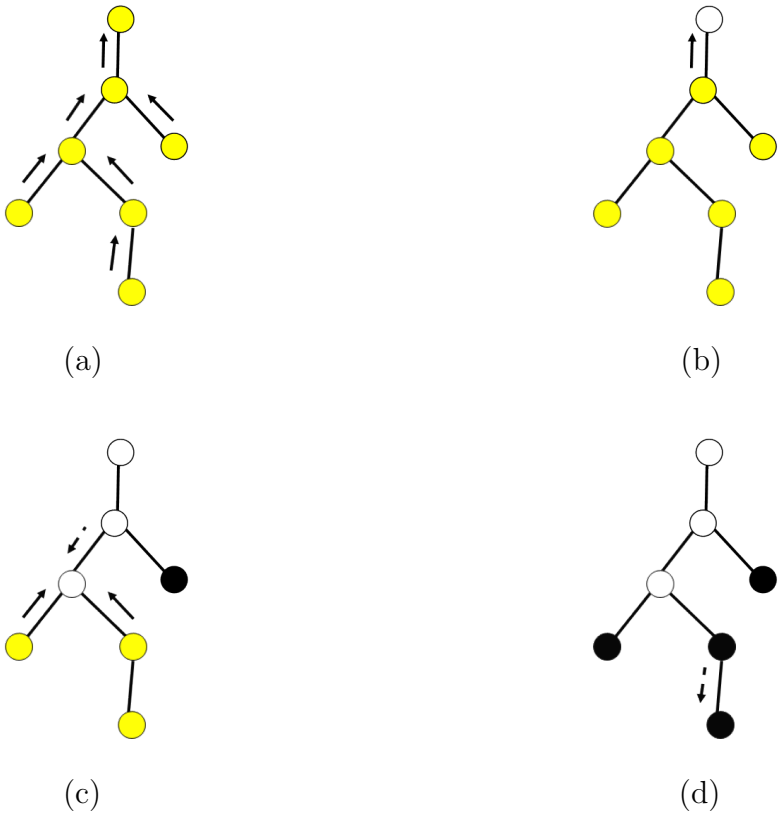


Figure 6.2: (a) Upward messages to root node; (b) Encoding root node with messages from children; (c) Continue encoding with messages from children and conditional message from parent; Encoding final node of scan.

depends only on messages from its children and the value of its parent. Therefore, one can efficiently encode an image $\mathbf{x}$ with the two-stage process described in the Introduction. The first pixel to be scanned is chosen arbitrarily. The graph is viewed as a tree with this pixel as its root. The scan order for the remaining pixels can be any lineal scan, where parent-child relationships are determined by the choice of root. In the first stage, an upward pre-encoding phase starting at the leaves, BP successively computes and passes messages from children to parents, using (2.15) and (2.16), up to the root node of the scan. In the second stage, a downward encoding phase from the root to the leafs takes place, where for $i = 1$ to N, Theorem VI.1 is used to compute $p_{i|*}(\cdot)$. The factor $\Psi_{\pi(i),i}(x_{\pi(i)}, x_i)$ can be viewed as a conditional message from node $\pi(i)$ to its child i. One feeds it along with x_i to the AC encoder.

Note that we did not need to actually compute the individual node beliefs nor the messages from parents to children. The complexity of this method is on the same order as that of ordinary BP message passing on the acyclic graph. More specifically, the total number of computations is roughly half that of regular BP, since only one message per edge is computed rather than the two when beliefs at all nodes is the objective.

Example VI.2. *Consider the acyclic graph shown in Figure 6.1. The encoding is done with one scan, while in Figure 6.2 the encoding is done with a different scan. In particular, the first nodes of the respective scans are different. As long as both scans are connected, then the compression will be optimal. However, as one can observe, the root node in Figure 6.1 has a smaller greatest distance to the leaves, which means that if the message passing is partitioned among parallel processors, running the encoding algorithm on the graph in Figure 6.1 will require less time than on the graph in Figure 6.2.*

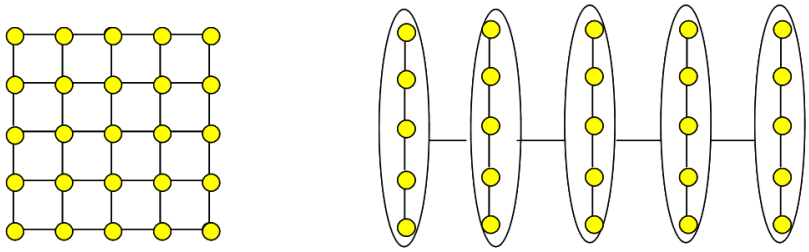
6.1.2 EASY Cyclic MRFs

If the graph G has cycles, then it is still possible to perform optimal compression, provided the conditions for performing exact inference are met. This means that a cluster tree can be formed where the size of the largest cluster is of reasonable size, or that there is a loop cutset and an acyclic edge cover based on it such that the largest relevant set is not too large.

6.1.2.1 Cluster Tree Based Coding

Suppose the nodes of the original cyclic graph G can be grouped into clusters resulting in a cluster tree $\widehat{G}$ where the largest cluster is of reasonable size. Then the acyclic algorithm of the previous section can be applied to the resulting MRF on the cluster tree. Since the clusters of the cluster tree partition the nodes of the original

graph, the nodes of G can be encoded together as groups or blocks of nodes.



(a) (b)

Figure 6.3: (a) Original 4-pt. grid graph; (b) Cluster graph.

Example VI.3. *Consider the* 2×3 *grid graph shown in Figure 6.3 (a). A cluster tree can be formed by lumping the columns together, as shown in (b). The resulting MRF on this cluster tree can be encoded, for example, by beginning at the rightmost cluster, passing usual BP messages to the leftmost node, and the starting the downward phase, encoding the cluster nodes from left to right.*

6.1.2.2 Local Conditioning Based Coding

In this section we losslessly compress an MRF on a cyclic graph $G = (V, E)$. The nodes of V will be scanned as $1, 2, \ldots, |V|$, and we let A_i indicate the first i nodes of the scan, i.e., the first i nodes to be encoded. The optimal coding distribution could in principle be brute force computed as

$$\begin{aligned} p_{i|*}(x_i) &= \frac{p(x_{A_i})}{p(x_{A_{i-1}})} \\ &= \frac{Z_{A_i}(x_{A_i})}{Z_{A_{i-1}}(x_{A_{i-1}})}. \end{aligned} \tag{6.2}$$

Of course, since G is cyclic such computation of the numerator and denominator is prohibitively complex. However, if there exists a loop cutset L of G, together with a connected unwrapping $\overline{G}$ based on L with a reasonably sized $\max_{\{i,j\}} |R_{ij}|$ and $\max_i |R_{\overline{A}_i}| - |L \cap A_i|$, then one can compute the optimal coding distributions with

reasonable complexity using Local Conditioning. In this case, Local Conditioning is performed on the unwrapped graph $\overline{G}$. Using the generalization of Theorem IV.1 to subsets of arbitrary size and the subsequent decomposition of beliefs given by (4.9), the optimal coding distribution can be expressed in terms of the conditional beliefs produced by Local Conditioning as

$$\begin{aligned} p_{i|*}(x_i) &= \frac{\sum_{x_{L\setminus A_i}} Z_{L\cup A_i}(x_{A_i}, x_{L\setminus A_i})}{\sum_{x_{L\setminus A_{i-1}}} Z_{L\cup A_{i-1}}(x_{A_{i-1}}, x_{L\setminus A_{i-1}})} \\ &= \frac{\sum_{x_{L\setminus A_i}} \overline{Z}^{(x_L)}_{\overline{A}_i}(\overline{x}_{\overline{A}_i})}{\sum_{x_{L\setminus A_{i-1}}} \overline{Z}^{(x_L)}_{\overline{A}_{i-1}}(\overline{x}_{\overline{A}_{i-1}})}, \end{aligned}$$

where $\overline{A}_i$ is a set of nodes in $\overline{G}$ corresponding to the set A_i such that for each loop cutset node $l \in A_i \cap L$, there is exactly one copy $\overline{l}$ of l in $\overline{A}_i$.

To simplify the algorithm and analysis, we want the $\overline{A}_i$ to form a lineal *sub*-scan $\overline{1}, \overline{2}, \ldots, \overline{|V|}$ of the nodes in $\overline{G}$ with the following properties: for each non loop cutset node $i \notin L$, the corresponding node $\overline{i} = \rho^{-1}(i)$ is in the sub-scan; for each loop cutset node $l \in L$, there is exactly one copy $\overline{l} \in \rho^{-1}(l)$ of l in the sub-scan; and for each $\overline{i}$ in the sub-scan, its parent $\pi(\overline{i})$ precedes it. The following proposition, whose proof is in the last section of this chapter, tells us that by choosing the scan $1, 2, \ldots, |V|$ to be connected at each iteration i, it is possible to construct a lineal sub-scan of nodes in $\overline{G}$.

Proposition VI.4. Let $1, 2, \ldots, |V|$ be a scan of nodes in G. If for each i, the set A_i is connected in G, then there exists a corresponding $\overline{A}_i$ that is connected in $\overline{G}$. Therefore there exists a lineal sub-scan $\overline{1}, \overline{2}, \ldots, \overline{|V|}$ of the nodes of $\overline{G}$ such that $\overline{i} = i$ or $\overline{i}$ is a copy of i.

Given the acyclic graph $\overline{G}$, the initial node $\overline{1}$ of the lineal sub-scan defines parent-child relations throughout $\overline{G}$ as described in Section 2.1.1, and is thus the root of the

resulting tree. Note that it does not matter whether this initial node of the lineal sub-scan is a copy node or not. As in the acyclic case, the upward pre-encoding phase depends only on knowledge of the root, while the downward encoding phase depends on knowledge of the entire lineal sub-scan.

Recall from Section 2.1.1 that $\sigma(\bar{i})$ is the set of neighbors of $\bar{i}$ that lie on non-backtracking paths between i and the leaves of $\overline{G}$. By construction, the set $\overline{A}_i$ contains the root of the sub-scan on $\overline{G}$ used to encode the original cyclic MRF. Therefore the boundary $\partial\overline{A}_i$ of $\overline{A}_i$ is also the set of children $\sigma(\overline{A}_i)$ of $\overline{A}_i$ and thus denoted by

$$\sigma(\overline{A}_i) = \left\{ \overline{k} \notin \overline{A}_i \mid \pi(\overline{k}) \in \overline{A}_i \right\}. \tag{6.3}$$

Just as the relevant set for a node $\bar{i}$ in $\overline{G}$ can be expressed as $R_{\bar{i}} = L \setminus (\bigcup_{\bar{k} \in \partial \bar{i}} L_{\bar{k} \setminus \bar{i}})$ we can likewise define the relevant set of loop cutset nodes for the connected set $\overline{A}_i$ as follows

$$R_{\overline{A}_i} = L \setminus \left(\bigcup_{\bar{k} \in \sigma(\overline{A}_i)} L_{\bar{k} \setminus \pi(\bar{k})} \right). \tag{6.4}$$

That is, for a connected set of nodes in $\overline{G}$, the relevant set of loop cutset nodes are those that are not summable with respect to a message incoming to that set. The reason that "summability" rather than "redundancy" determines the relevant loop cutset nodes for the set $R_{\overline{A}_i}$ is that a loop cutset node may be redundant for a message $M_{\overline{k} \to \pi(\bar{i})}$ incoming to $R_{\overline{A}_i}$ yet still be relevant for $R_{\overline{A}_i}$.

The nodes in $R_{\overline{A}_i}$ that have already been encoded through the ith iteration of the encoding will be denoted $R^{e}_{\overline{A}_i}$, while those that have not been encoded are indicated by $R^{u}_{\overline{A}_i}$. Similarly for the relevant set for an edge $\{\overline{k}, \overline{j}\}$. The following theorem states the formulas for the optimal coding distributions for a node i in an EASY cyclic MRF. As well, Section 2.1.1 defines the surface of the set $\overline{A}_i$ to be the subset of nodes $\gamma(\overline{A}_i)$ one of whose neighbors are not in $\overline{A}_i$.

Theorem VI.5. *Suppose the graph $G = (V, E)$ of an MRF $\mathbf{X}$ is cyclic, and that $\overline{G}$ is a connected unwrapping of G based on a loop cutset L. Furthermore, let $1, 2, \ldots, |V|$ be a scan of V such that there is a corresponding sequence of nodes $\overline{1}, \overline{2}, \ldots, \overline{|V|}$ in $\overline{G}$ is a lineal sub-scan of nodes in $\overline{G}$. Then, for $i \notin L$,*

$$p_{i|*}(x_i|x_{A_{i-1}}) =$$

$$\frac{\overline{\Phi}_{\bar{i}}(\overline{x}_{\bar{i}})\overline{\Psi}_{\bar{i},\pi(\bar{i})}(\overline{x}_{\bar{i}}, \overline{x}_{\pi(\bar{i})}) \sum_{y_{R^u_{\overline{A}_{i-1}}}} \prod_{\bar{k} \in \partial \overline{A}_{i-1} \setminus \bar{i}} \widehat{m}^{(y_{R^u_{\bar{k}\pi(\bar{k})}}, x_{R^e_{\bar{k}\pi(\bar{k})}})}_{\bar{k} \to \pi(\bar{k})}(\overline{x}_{\pi(i)}) \prod_{\bar{k} \in \sigma(\bar{i})} \widehat{m}^{(y_{R^u_{\bar{k}\bar{i}}}, x_{R^e_{\bar{k}\bar{i}}})}_{\bar{k} \to \bar{i}}(\overline{x}_{\bar{i}})}{\sum_{\overline{x}''_{\bar{i}}} \overline{\Phi}_{\bar{i}}(\overline{x}''_{\bar{i}})\overline{\Psi}_{\bar{i},\pi(\bar{i})}(\overline{x}''_{\bar{i}}, \overline{x}_{\pi(\bar{i})}) \sum_{y_{R^u_{\overline{A}_{i-1}}}} \prod_{\bar{k} \in \partial \overline{A}_{i-1} \setminus \bar{i}} \widehat{m}^{(y_{R^u_{\bar{k}\pi(\bar{k})}}, x_{R^e_{\bar{k}\pi(\bar{k})}})}_{\bar{k} \to \pi(\bar{k})}(\overline{x}_{\pi(i)}) \prod_{\bar{k} \in \sigma(\bar{i})} \widehat{m}^{(y_{R^u_{\bar{k}\bar{i}}}, x_{R^e_{\bar{k}\bar{i}}})}_{\bar{k} \to \bar{i}}(\overline{x}''_{\bar{i}})}$$

where $R^u_{\overline{A}_i} = R_{\overline{A}_i} \setminus A_i$ are the relevant nodes for $\overline{A}_i$ that have not been encoded and $R^e_{\overline{A}_i} = R_{\overline{A}_i} \cap A_i$ are those that have been encoded. Similarly, for $R^u_{\bar{k}\bar{i}}$ and $R^e_{\bar{k}\bar{i}}$ for edge $\{\bar{k}, \bar{i}\}$. For $i \in L$,

$$p_{i|*}(x_i|x_{A_{i-1}}) =$$

$$\frac{\overline{\Phi}_{\bar{i}}(\overline{x}_{\bar{i}})\overline{\Psi}_{\bar{i},\pi(\bar{i})}(\overline{x}_{\bar{i}}, \overline{x}_{\pi(\bar{i})}) \sum_{y_{R^u_{\overline{A}_{i-1}}}} \prod_{\bar{k} \in \partial \overline{A}_{i-1} \setminus \bar{i}} \widehat{m}^{(y_{R^u_{\bar{k}\pi(\bar{k})}}, x_{R^e_{\bar{k}\pi(\bar{k})}})}_{\bar{k} \to \pi(\bar{k})}(\overline{x}_{\pi(\bar{k})}) \prod_{\bar{k} \in \sigma(\bar{i})} \widehat{m}^{(x_{R^u_{\bar{k}\bar{i}}}, x_{R^e_{\bar{k}\bar{i}}})}_{\bar{k} \to \bar{i}}(\overline{x}_{\bar{i}})}{\sum_{y_i} \sum_{\overline{x}''_{\bar{i}}} \overline{\Phi}_{\bar{i}}(\overline{x}''_{\bar{i}})\overline{\Psi}_{\bar{i},\pi(\bar{i})}(\overline{x}''_{\bar{i}}, \overline{x}_{\pi(\bar{i})}) \sum_{y_{R^u_{\overline{A}_{i-1}}} \setminus i} \prod_{\bar{k} \in \partial \overline{A}_{i-1} \setminus \bar{i}} \widehat{m}^{(y_{R^u_{\bar{k}\pi(\bar{k})}}, x_{R^e_{\bar{k}\pi(\bar{k})}})}_{\bar{k} \to \pi(\bar{k})}(\overline{x}_{\pi(\bar{k})}) \prod_{\bar{k} \in \sigma\bar{i}} \widehat{m}^{(y_{R^u_{\bar{k}\bar{i}}}, x_{R^e_{\bar{k}\bar{i}}})}_{\bar{k} \to \bar{i}}(\overline{x}''_{\bar{i}})}$$

First, a word on the denominators of the equations in Theorem VI.5. As is made clearer in the proof, given in the last section of the chapter, the denominator, properly, should include a summation over $R_{\overline{A}_{i-1}}$ rather than $R_{\overline{A}_i}$. However, if node i is not in L, then $R_{\overline{A}_i} = R_{\overline{A}_{i-1}}$. And if i is in L, then we can simply sum over x_i first, as in the denominator of the second equation above. Notice that in the case where $i \in L$ the denominator includes two summations over the values at node i. The first, $\sum_{x_i}$, is over the values of the loop cutset node i in the original graph G. Each such value, combined with the various configurations on the other loop cutset nodes $L \setminus i$, induce configurations on the copy nodes of $\overline{G}$ and thereby a particular conditional MRF on

the acyclic graph $\overline{G}$. The second summation, $\sum_{\overline{x}_{\bar{i}}}$, is over the values the copy $\bar{i}$ of i that is included in the lineal sub-scan. The range of values that $\overline{x}_{\bar{i}}$ can assume in $\overline{G}$ is the same range as for x_i in G. However, when we sum over $\overline{x}_{\bar{i}}$, one factor of the summand is the self-potential on $\bar{i}$, which is implicitly a function of the particular configuration x_L, hence of x_i. In other words, for each value of x_i, only the induced value $\overline{x}_{\bar{i}}$ contributes to the summand within $\sum_{x_i}$.

As in the acyclic case, the formula for the optimal coding distributions suggests the algorithm to carry out the computations. First, though, the formula and thus the algorithm for computing the optimal coding distributions in an EASY cyclic MRF differs from an acyclic MRF in the following critical way. In an acyclic MRF the optimal coding distribution computed at a node i is function of the self-potential on i, the edge potential between i and its parent $\pi(i)$, and messages incoming to node i. As stated above in Theorem VI.5, in a cyclic MRF not only is the optimal coding distribution computed at a node $\bar{i}$ a function of the self-potential on node $\bar{i}$, the edge potential between $\bar{i}$ and its parent $\pi(\bar{i})$, and the (LC) messages incoming to $\bar{i}$, but additionally, of (LC) messages into the remaining nodes on the surface of $\overline{A}_i$. This difference is due to the fact that, in the acyclic case the optimal coding distribution is a ratio of beliefs on an acyclic MRF and therefore messages incoming to $\overline{A}_i \setminus \bar{i}$ cancel in the numerator and denominator; whereas in the EASY cyclic case the optimal coding distribution is a ratio of sums of beliefs on acyclic MRFs, and because messages incoming to $\overline{A}_i \setminus \bar{i}$ are functions of the different values on loop cutset nodes, over which the summations are taking place, these cannot be canceled.

The upward pre-encoding phase begins at the leaves of $\overline{G}$, with LC super messages $\widehat{M}^{(R_{\bar{k}\bar{i}})} = [\widehat{m}_{\bar{k}\to\bar{i}}^{(x_{R_{\bar{k}\bar{i}}})}]$ passed up to the root $\overline{1}$ of the sub-scan, where the summed out message $\widehat{m}_{\bar{k}\to\bar{i}}^{(x_{R_{\bar{k}\bar{i}}})}$ is as defined in Section 4.3. As defined in Section 4.3, $\widehat{m}^{(R_{\bar{k}\bar{i}})}$ is the summed out and reduced message in which the redundant loop cutset nodes for edge $\{\bar{k}, \bar{i}\}$ have been ignored and the summable loop cutset nodes have already been

summed out. A column of $\widehat{M}^{(R_{\bar{k}\bar{i}})} = [\widehat{m}_{\bar{k}\to\bar{i}}^{(x_{R_{\bar{k}\bar{i}}})}]$, passed over edge $\{\overline{k}, \overline{i}\}$, is a BP message conditioned on a particular configuration of $R_{\overline{ki}}$.

Theorem VI.6. *Let $G = (V, E)$ be a cyclic graph and let $\overline{G} = (\overline{V}, \overline{E})$ be a connected unwrapping based on loop cutset L. The complexity of the upward pre-encoding phase is then $\mathcal{O}(|E||\mathcal{X}|^{c'(L)})$, where we say the* cost *of L is*

$$c'(L) = \max_{\bar{i}\in\overline{V}} \left| \bigcup_{\overline{k}\in\sigma(\bar{i})} R_{\overline{ik}} \right|. \tag{6.5}$$

Proof. For a given node $\bar{i} \in \overline{V}$ the incoming LC super messages $\widehat{M}_{\overline{k}\to\bar{i}}^{(R_{\overline{ik}})}$, $\overline{k} \in \sigma(\bar{i})$, are filled out to account for configurations on the set $\bigcup_{\overline{k}\in\sigma(\bar{i})} R_{\overline{ik}}$, so that the number of multiplications required to combine each of these filled out messages is proportional to a number that is exponential in the size of this set. □

Similar to the algorithm for optimal AC encoding of an acyclic MRF, a downward encoding phase begins with the root of the sub-scan combining the incoming super messages with the self-potential for node $\overline{1}$, as in Theorem IV.17, to compute the optimal coding distribution for the initial node in the scan of G. Following this, subsequent nodes of the lineal sub-scan are visited and at each node $\bar{i}$ in $\overline{G}$ the optimal coding distribution $p_{i|*}(\cdot)$ is computed for the corresponding node $i = \rho(\bar{i})$ in G. As can be seen from the two equations of Theorem VI.5, this coding distribution is computed the with LC messages incoming to the set $\overline{A}_i$. These incoming messages will in general need to be filled out so that they are conditioned on $X_{R_{\overline{A}_i}}$ rather than some $X_{\bar{k}\bar{j}_{\bar{k}}}$. Once these messages are combined, the columns corresponding to different configurations on $R_{\overline{A}_i}$ are summed together to give the belief vector for node $\bar{i}$ in $\overline{G}$ which is used to compute the optimal coding distribution for i in G.

It is natural to ask whether the set $R_{\overline{A}_i}$ is monotonic in i, whether the size is increasing or decreasing as the encoding proceeds. The following theorem tells us

that, in fact, the $R_{\overline{A}_i}$ are monotone nondecreasing in size.

Theorem VI.7. *Let $G = (V, E)$ be a cyclic graph and let $\overline{G} = (\overline{V}, \overline{E})$ be a connected unwrapping based on loop cutset L. Let $1, 2, \ldots, |V|$ be a scan of V with $\overline{1}, \overline{2}, \ldots, \overline{|V|}$ the corresponding lineal sub-scan of $\overline{V}$, and $R_{\overline{A}_i}$ the relevant set for the first i nodes of the lineal sub-scan. Then,*

$$R_{\overline{A}_i} \subset R_{\overline{A}_{i+1}}. \tag{6.6}$$

Proof. Using (6.4), we see that

$$R_{\overline{A}_i} = L \setminus \left(\left(\bigcup_{\overline{k} \in \sigma(\overline{A}_i) \setminus \overline{i}+1} L_{\overline{k} \setminus \pi(\overline{k})} \right) \bigcup \left(L_{\overline{i}+1 \setminus \pi(\overline{i}+1)} \right) \right) \tag{6.7}$$

and

$$R_{\overline{A}_{i+1}} = L \setminus \left(\left(\bigcup_{\overline{k} \in \sigma(\overline{A}_i) \setminus \overline{i}+1} L_{\overline{k} \setminus \pi(\overline{k})} \right) \bigcup \left(\bigcup_{\overline{k} \in \sigma(\overline{i}+1)} L_{\overline{k} \setminus \overline{i}+1} \right) \right). \tag{6.8}$$

It is straightforward to see that $\left(\bigcup_{\overline{k} \in \sigma(\overline{i}+1)} L_{\overline{k} \setminus \overline{i}+1} \right) \subset L_{\overline{i}+1 \setminus \pi(\overline{i}+1)}$, from which the theorem statement can easily be deduced. □

Because the relevant sets for the $\overline{A}_i$ it might seem that the complexity of the downward phase is increasing with i. This is not necessarily the case, as the complexity of the downward encoding phase is actually determined by the unencoded relevant nodes $R^u_{\overline{A}_i}$, since the LC super messages incoming to the surface of $\overline{A}_i$ need to be filled out to the size of the alphabet for this set. The following theorem states this formally.

Theorem VI.8. *Let $G = (V, E)$ be a cyclic graph and let $\overline{G} = (\overline{V}, \overline{E})$ be a connected unwrapping based on loop cutset L. Let $1, 2, \ldots, |V|$ be a scan of V with $\overline{1}, \overline{2}, \ldots, \overline{|V|}$ the corresponding lineal sub-scan of $\overline{V}$. The complexity of the downward encoding phase of LC based AC encoding using is $\mathcal{O}(|E||\mathcal{X}|^{c''(L)})$, where the* cost *of L is*

$$c''(L) = \max_{i\in\{1,2,\ldots,|V|\}} |R^u_{\overline{A}_i}| \tag{6.9}$$

$$= \max_{i\in\{1,2,\ldots,|V|\}} |R_{\overline{A}_i}| - |A_i \cap L|. \tag{6.10}$$

Proof. Without loss of generality, assume that each node assumes values in a common alphabet $\mathcal{X}$. Each of the incoming LC super messages $\widehat{M}^{(R_{\overline{k}\pi(\overline{k})})}_{\overline{k}\to\pi(\overline{k})}$, $\overline{k} \in \sigma(\overline{A}_i)$, must be "filled out" to the set $R^u_{\overline{A}_i}$. The number of multiplications that are required in combining these filled out incoming messages is $|\mathcal{X}|^{|R^u_{\overline{A}_i}|}$. □

Example VI.9. *Consider a binary MRF on the* 2×3 *grid graph* G *shown in Figure 6.4 (a). Loop cutset* $L = \{2, 6\}$ *is indicated in red. In (b) we see an unwrapped graph* $\overline{G}$ *based on* L*. Note that a copy of node 2 is internal to* $\overline{G}$*. Also, (b) illustrates the upward phase of super message passing to the root node 1. In (c) the messages into 1 are used to encode the observed value* $X_1 = 1$ *(black). The second node in the scan is the copy node* $2^{(1,3)}$*, and (d) illustrates the encoding of the observed value* $X_2 = 0$ *(white). Once the coding distribution is computed for this copy node and the value of the original node in* G *is encoded, all copies corresponding to this original node become fixed to that (induced) value. For the encoding of non-copy node epicted in (e), since a copy of node 2 is already encoded of node 4 must be conditioned on the observed value of* X_2*. On the other hand, even though loop cutset node 6 is relevant for node* 4*, no copy of it has been encoded, so encoding of node 4 will not be conditioned on any particular value of* X_6*. The upward messages used to encode node 4 will be* $\overline{m}^{(X_2=0,X_6=0)}_{5\to4}$ *and* $\overline{m}^{(X_2=0,X_6=1)}_{5\to4}$ *from node 5 to 4 and* $\overline{m}^{(X_2=0,X_6=0)}_{3\to2^{(1,3)}}$ *and* $\overline{m}^{(X_2=0,X_6=1)}_{3\to2^{(1,3)}}$ *from node 3 to node* $2^{(1,3)}$*. These will be combined according to the value of* X_6 *and then the two resulting columns will be summed over the value of* X_6*. This will yield a belief vector for node 4 conditioned on* $X_2 = 0$ *from which the optimal coding distribution will be computed. In (f), node 3 is being encoded. Again, the value of* X_2 *is fixed to its observed value, the incoming messages are matched up according to the value of* X_6*, then the two columns corresponding to the different values of* X_6

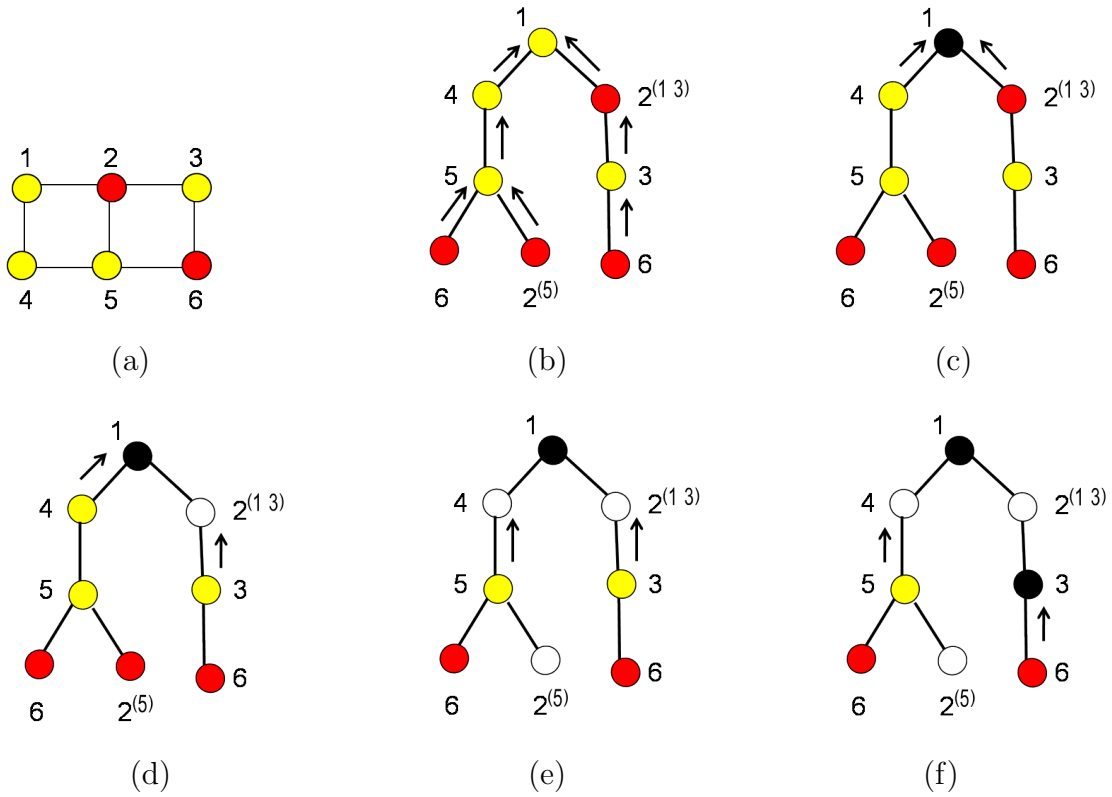


Figure 6.4: (a) Original graph; (b) Upward (super) message passing; (c) - (f) Stages of the downward encoding phase, using the lineal sub-scan $1, 2^{(1,3)}, 4, 3$ followed by either $6^{(3)}, 5$; $5, 6^{(3)}$; or $5, 6^{(5)}$.

are summed together to give a belief vector for node 3 conditioned on $X_2 = 0$.

We summarize the above results of optimal compression of MRFs in the following theorem.

Theorem VI.10. *Let* $\mathbf{X}$ *be an MRF defined on a graph* $G = (V, E)$ *that is either acyclic or EASY cyclic. Then, using the algorithms given above, optimal compression is obtainable with the following complexity.*

A) If G *is acyclic,* $|E||\mathcal{X}|^2$,

B) If G *is cyclic,*

(I) and the cluster tree algorithm is used, $|\hat{E}||\mathcal{X}|^{\max_i |K_i|}$, *where* $\hat{E}$ *is the set of edges in the cluster tree and* $\max_i |K_i|$ *is the size of the largest cluster,*

(II) and the LC based algorithm is used, $|E||\mathcal{X}|^{c'(L)}$ *for the pre-encoding phase,*

where

$$c'(L) = \max_{\bar{i} \in \overline{V}} \left| \bigcup_{\overline{k} \in \sigma(\bar{i})} R_{\overline{ik}} \right|, \tag{6.11}$$

and $|\mathcal{X}|^{c''(L)}$ *for the encoding phase, where*

$$c''(L) = \max_{i \in \{1,2,\ldots,|V|\}} |R^u_{\overline{A}_i}|. \tag{6.12}$$

6.2 Reduced Cutset Coding of HARD Cyclic MRFs

In this section we introduce Reduced Cutset Coding (RCC), a suboptimal method for losslessly compressing a HARD Markov random field. A cutset of sites U is chosen, X_U is AC encoded and the remainder $X_{V\setminus U}$ is AC encoded conditioned on X_U. The rate for encoding X_U is

$$R_U \;=\; \frac{1}{|U|}\overline{R}_U, \tag{6.13}$$

where $\overline{R}_U \;=\; \mathbb{E}\left[l(X_U)\right]$ is the expected codeword length for X_U. The remaining variables $X_{V\setminus U}$ are encoded at rate

$$R_{V\setminus U} \;=\; \frac{1}{|V \setminus U|}\overline{R}_{V\setminus U}, \tag{6.14}$$

where $\overline{R}_{V\setminus U} \;=\; \mathbb{E}\left[l(X_{V\setminus U})\right]$. The total encoding rate for $\mathbf{X}$ is

$$R = \frac{1}{|V|}\overline{R} \;=\; \frac{|U|}{|V|}R_U + \frac{|V \setminus U|}{|V|}R_{V\setminus U}. \tag{6.15}$$

The quantities $\overline{R}_U$, $\overline{R}_{V\setminus U}$, and $\overline{R}$ will be referred to a the *total* rates for encoding X_U, $X_{V\setminus U}$, and $\mathbf{X}$, respectively.

In RCC, the cutset U is selected so that the induced subgraph G_U and all of the components $C_1, \ldots, C_K$ of $G\setminus U$ are either acyclic or EASY, in that they are tractable to either the cluster or LC based encoding methods for cyclic graphs. From (2.11)

we see that X_U would be encoded optimally if the coding distribution f had been chosen to be the marginal distribution $p_G^U(\theta)$ of X_U. However, by assumption in this section G is HARD, so computing $p_G^U(\theta)$ is intractable. By choosing G_U to be EASY, X_U can be efficiently encoded with an optimal distribution computed for an MRF on G_U, which in the language of Chapter V, is a reduced MRF on G_U. However, we can choose the potentials on G_U in a way to optimize minimize the divergence between the computed coding distribution and the marginal distribution $p_G^U(\theta)$ for X_U. Since the components $C_1, \ldots, C_K$ are tractable, and since the conditional distribution of a set of sites C_i conditioned on its boundary is an MRF on the induced subgraph G_{C_i} with the self potentials appropriately modified to account for the configuration on the boundary ∂C_i, the components are encoded optimally conditioned on the values X_U of the cutset.

Because in encoding X_U, we compute the optimal distribution for on MRF on G_U, i.e. we cut the edges from U to ∂U, to permit analysis it is best to use the exponential representation of MRFs. This analysis will then allow us to choose the optimal potentials for G_U. For the remainder of this section we assume a fixed t and a distribution p, indexed by either a fixed but arbitrary exponential vector $\theta \in \Theta$ or the corresponding moment vector $\mu = \mu(\theta)$.

6.2.1 Encoding the Cutset

Because the cutset U is chosen so that the induced subgraph G_U is tractable to either the clustering method or LC, we can efficiently compute any AC coding distribution $f = p_{G_U}(\theta_U^c)$, where θ_U^c is an exponential coordinate used to encode X_U. In other words, we have the freedom to choose any collection of potentials on G_U.

Also from (2.11) we see that the redundancy incurred by encoding X_U with reduced MRF coding distribution $p_{G_U}(\theta_U^c)$ instead of its marginal distribution is

$D(p_G^U(\theta)||p_{G_U}(\theta_U^c)$. Therefore, the rate of encoding X_U in this way is

$$R_U = \frac{1}{|U|}\left[H_G(X_U) + D(p_G^U(\theta)||p_{G_U}(\theta^c))\right]. \tag{6.16}$$

We showed in Section 5.3.2 that the redundancy decomposes as

$$D(p_G^U(\theta)||p_{G_U}(\theta_U^c) = D(p_G^U(\theta)||p_{G_U}(\mu_U)) + D(p_{G_U}(\mu_U)||p_{G_U}(\theta_U^c)),$$

which implies that the choice of θ^c that minimizes this divergence and therefore induces the optimal reduced MRF coding distribution, is the one, θ_U^*, that preserves the moments of U. To determine the optimal θ_U^* the first step is to compute the moment coordinates for nodes and edges in U. For acyclic MRFs, the moment coordinates for nodes i and edge $\{i, j\}$ can be computed straightforwardly, using (2.14) and its generalization to edges. Once these moment coordinates are obtained, there are straightforward formulas for the moment-matching exponential coordinates for nodes and edges in U [55, 56].

For cyclic graphs, however, one can use the cluster or Local Conditioning based methods to make analogous computations. However, if the graph is HARD, then an approximate inference method, such as Loopy Belief Propagation could be used instead. However, we note that these moment calculations are part of the "design" and not the running of the algorithm. That is, they are done only once. Thus even if they are HARD, they don't contribute to operational complexity. Once the moment parameter μ_U or an approximation $\hat{\mu}_U$ is obtained, one can use Iterative Proportional Fitting [10], Iterative Scaling [12], or variations thereof, to compute the adjusted exponential coordinates θ_U^* for the moment-preserving reduced MRF on G_U.

Another issue in the choice of U is that we would like the divergence in (6.16) to be small. This would likely have to be determined empirically for a given application, though one might be able to show an inequality indicating which of two or more

candidate cutsets should be used.

6.2.2 Conditional Component Coding

As mentioned, the cutset U is also chosen so that the disjoint subgraphs $G_{C_1}, G_{C_2}, \ldots, G_{C_K}$ induced by the components of $G \setminus U$ are tractable to either JC or LC. Therefore, if each of the disjoint subgraphs of $G \setminus U$ is tractable, then LC or the cluster method can be used to optimally encode the random variables X_{C_i} on each component of $G \setminus U$ conditioned on X_U.

Before analyzing conditional coding, we consider how well it would work if we encoded $X_{V \setminus U}$ independently. If we encode each component X_{C_i} independently of X_U with a reduced MRF coding distribution $p_{G_{C_i}}(\theta'_{C_i})$ then the total rate for encoding the remainder $X_{V \setminus U}$ would be

$$\begin{aligned}
\overline{R}_{V \setminus U} &= H_G^{V \setminus U} + \sum_{C_i} D(p_G^{C_i}(\theta) || p_{G_{C_i}}(\theta'_{C_i})) \\
&= H_G^{V \setminus U | U}(\theta) + I(X_{V \setminus U}; X_U || \theta) + \sum_{C_i} D(p_G^{C_i}(\theta) || p_{G_{C_i}}(\theta'_{C_i})) \\
&= \sum_{C_i} \frac{|C_i|}{|V \setminus U|} R_{C_i} + I(X_{V \setminus U}; X_U || \theta) + \sum_{C_i} D(p_G^{C_i}(\theta) || p_{G_{C_i}}(\theta'_{C_i})),
\end{aligned}$$

where R_{C_i} is the rate of encoding component C_i. We see here that the redundancy can be broken into a penalty for independently encoding $X_{V \setminus U}$, which is the mutual information between X_U and $X_{V \setminus U}$ plus a divergence penalty for encoding each component X_{C_i} with a reduced MRF coding distribution rather than its marginal distribution.

If we instead encode each component X_{C_i} with its marginal distribution $p_G^{C_i}(\theta)$ independently of X_U, the total rate for the remainder $X_{V \setminus U}$ would be

$$\overline{R}_{V \setminus U} = H_G^{V \setminus U}(\theta) = H_G^{V \setminus U | U}(\theta) + I(X_{V \setminus U}; X_U; ; \theta).$$

Here, we do not pay the divergence penalty since we are using the marginal distribution. However, there is still the mutual information penalty for encoding $X_{V\setminus U}$ independently of X_U.

We now return to what we actually do in RCC, conditional coding of $X_{V\setminus U}$. We choose to encode each component X_{C_i} with its conditional distribution given the configuration $X_{\partial C_i}$ on its boundary. As noted in Section 2.1.4, the conditional distribution for component X_{C_i} given X_U is a reduced MRF distribution on G_{C_i} with modified potentials on the surface of C_i. The total rate is the conditional entropy $H_G^{V\setminus U|U}(\theta)$ of the remainder conditioned on the cutset.

This means that the remainder is encoded optimally, and that the only suboptimality in the RCC method is in the encoding of X_U with a reduced MRF coding distribution. We now summarize with the following theorem on the performance of RCC:

Theorem VI.11. *Let $G = (V, E)$ be an undirected graph on which an MRF $p_G(\theta)$ is defined and let μ be the moment coordinates corresponding to exponential parameter θ. If a cutset $U \subset V$ is encoded using coding distribution $p_{G_U}(\theta_U^c)$, then the rates with Reduced Cutset Coding are*

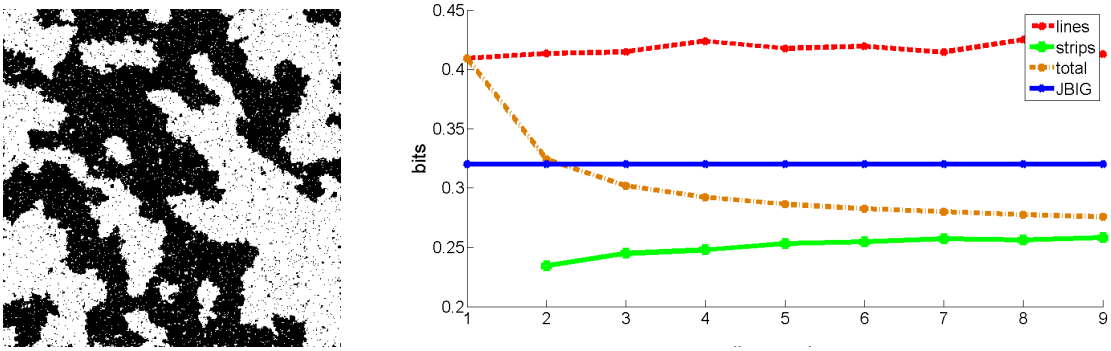
$$R_U = \tfrac{1}{|U|}\left[H_G^U(\theta) + D(p_G^U(\theta)||p_{G_U}(\mu_U)) + D(p_{G_U}(\mu_U)||p_{G_U}(\theta_U^c))\right],$$

$$R_{V\setminus U} = \tfrac{1}{|V\setminus U|}\left[H_G(X_{V\setminus U}|X_U)\right] = \sum_{C_i} \tfrac{|C_i|}{|V\setminus U|}\tfrac{1}{|C_i|} H_G(X_{C_i}|X_{\partial C_i}) = \sum_{C_i} \tfrac{|C_i|}{|V\setminus U|} R_{C_i},$$

where the redundancy is $D(p_G^U(\theta)||p_{G_U}(\mu_U)) + D(p_{G_U}(\mu_U)||p_{G_U}(\theta_U^c))$ and can be minimized by choosing θ_U^c to match the moments μ_U on nodes and edges in U, and the decomposition of $R_{V\setminus U}$ is due to U being a cutset of G.

6.3 Example: Ising Model on $N \times N$ Grid

In this section we consider RCC in the specific example of a homogeneous Ising model on an $N \times N$ grid graph, either 4 pt. or 8 pt. The graph is either the 4 pt. or



(a) (b)

Figure 6.5: (a) Typical image $\mathbf{x}_{GS}$ on 4 pt. graph, $\theta = .5$; (b) Coding rates: R_U (crosses), $R_{V\setminus U}$ (squares), total rate R (circles), JBIG (solid line).

8 pt. grid. The cutset U consists of every Mth row of the graph, and the components of $G \setminus U$ are therefore $(M-1) \times N$ rectangular strips. For the conditional coding, we can use the cluster method, grouping the columns of a strip into clusters. Or, we can use the Local Conditioning based method, for example, with the checkerboard loop cutset and one of the unwrapped graphs discussed in Section 4.4. Thus, the line spacing M should be chosen to have a moderate value, say, 10 or less.

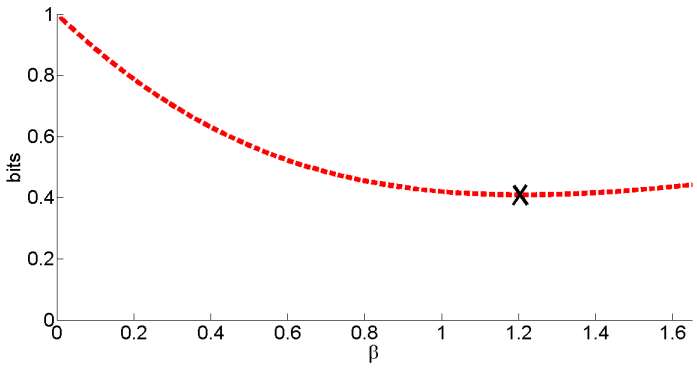
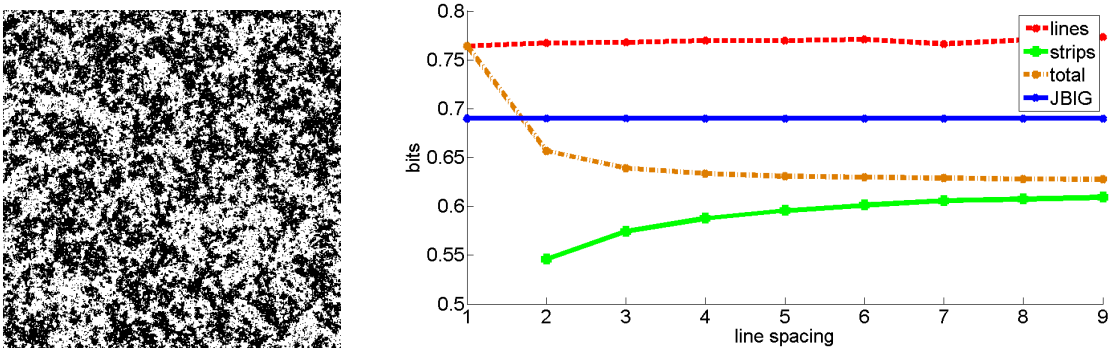


Figure 6.6: Plot of encoding rate for cutset of lines vs. θ^c used for the coding distribution.

We test the method on a sample image $\mathbf{x}_{GS}$ generated by the Gibbs Sampler. The lines of the cutset are encoded independently using a homogeneous distribution induced by an exponential coordinate $\overline{\theta}^*$. This is an approximation to the true moment-matching distribution based on the assumption that the grid graph is a fi-

nite portion of an infinite graph, in which case the distributions, hence the moments, would be stationary on the edges. We determine $\overline{\theta}^*$ by using Arithmetic Encoding to losslessly compress the subgraph induced by the cutset of lines using different values for the uniform exponential coordinate on the edges. Figure 6.6 shows a plot of the bit rate for different values of θ. Once the lines are encoded, these nodes are fixed to their observed values. Then the strips are encoded one at a time, conditioning on the lines above and below the strip. By the properties of the Gibbs Sampler [21], the image $\mathbf{x}_{GS}$ is typical for the respective Ising model; hence the empirical rate $R_{V\setminus U}$ of encoding the strips of $\mathbf{x}_{GS}$ should be very close to the rate specified in Theorem VI.11, while the observed rate R_U should actually be very close to a quantity slightly higher than that indicated in Theorem VI.11, due to the fact that the coding distribution is an approximation to the moment-matching reduced MRF distribution. Figure 6.5 (a) shows a typical image for the 4 pt. graph with coupling parameter $\theta = .5$. In (b) we show plots for the rates, in bits per pixel, for the lines of the cutset using the reduced MRF coding distribution and the optimal conditional coding of the strips. We show the overall coding rate of RCC for the image.



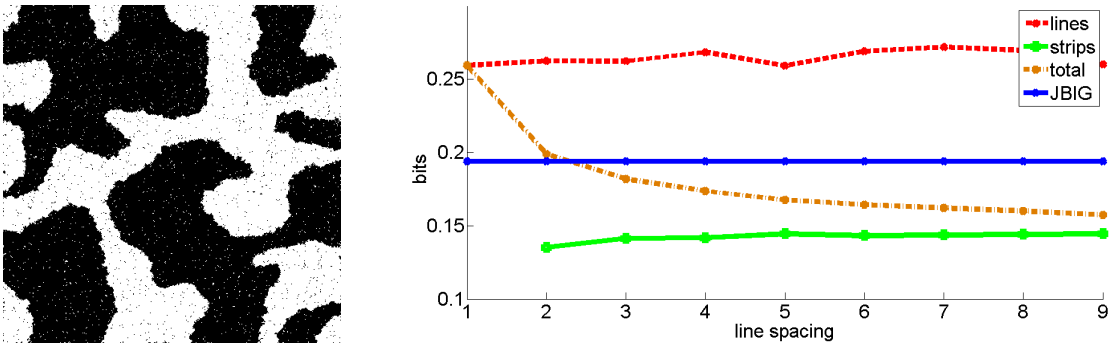
(a) (b)

Figure 6.7: (a) Typical image $\mathbf{x}_{GS}$ on 4 pt. graph and $\theta = .4$; (c) Coding rates: R_U (crosses), $R_{V\setminus U}$ (squares), total rate R (circles), JBIG (solid line).

We see that the rate R_U for the lines is essentially constant. This is because the same coding distribution is used for each line and because the lines are roughly

stationary. In addition, as the line spacing increases, the strip coding rate $R_{V\setminus U}$ increases because the ineriors of the strips have decreasing dependence on X_U. Note further that RCC achieves nearly its least rate with line spacing two, in which case the cutset consists of every other row, which leads to a very simple encoding of the single-row strips.

For any line spacing, the normalized encoding rate of the strips, $R_{V\setminus U}$, is a lower bound to the entropy of the Ising MRF, and the total encoding rate is an upper bound. From Figure 6.5 (b), we see that these are close and become closer as line spacing increases, which indicates that the coding is very nearly optimal. In Figure 6.7 (a) we show a typical image for the 4 pt. model with parameter $\theta = .4$ and (b) shows the rate performance of RCC. In Figure 6.8 (a) and (b) we show, respectively, a typical image on the 8 pt. graph with parameter $\theta = .28$ and the corresponding rate performance of RCC.



(a) (b)

Figure 6.8: (a) Typical image $\mathbf{x}_{GS}$ on 8 pt. graph and $\theta = .28$; (c) Coding rates: R_U (crosses), $R_{V\setminus U}$ (squares), total rate R (circles), JBIG (solid line).

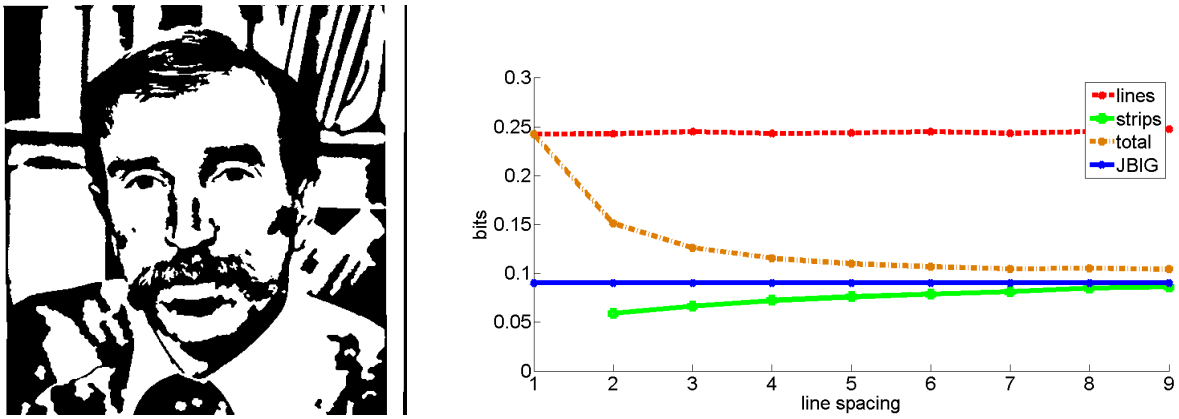
For comparison, we encoded $\mathbf{x}_{GS}$ using JBIG, a state-of-the-art bilevel image compression method [27, 45], and found that RCC outperforms JBIG. This is not surprising in that RCC is essentially optimal for the MRF, whereas JBIG was not at all designed for MRFs. Moreover, RCC with line spacing two is essentially optimal. However, as seen in Figures 6.5, 6.7, and 6.8 (a), the $\mathbf{x}_{GS}$ is a relatively simple image

with large homogenous regions so the that perhaps it is not so surprising that JBIG works as well as it does. We also applied RCC and JBIG to images generated by values of θ ranging from 0.1 to 1, and found similar results. For example, we found line spacing two was essentially as good as any other line spacing.

The order of complexity for RCC in this example is determined by the complexity of encoding the strips. For strip height M, the complexity is exponential in M. Since the performance of RCC was nearly as good for strips of height 1, using this strip height is both efficient and close to optimal. Moreover, the same routines used to encode the cutset of lines can be used to encode the remaining strips.

6.4 Example: RCC on Real-World Image

In this section we apply the RCC method presented in the last section to a real-world image. Essentially, we are modeling "blobby" bilevel images such as the one in Figure 6.9 (a) as instantiations of an Ising model on either the 4 pt. or 8 pt. topologies. We note that since we demonstrated in the last section that RCC is nearly optimal for the 4 pt. and 8 pt. models, the success of this approach to compressing images as in Figure 6.9 (a) depends on how faithful the model is in capturing the dependencies of the image. Figure 6.9 (b) shows results for encoding (a) with RCC modeled as an Ising model on the 4 pt. graph with $\theta = .6$. We found this θ value by searching for the one that gave the best overall coding performance. We can see that the overall coding rate achieved by RCC is slightly higher than that for JBIG.



(a) (b)

Figure 6.9: (a) Bilevel image to be encoded; (b) Coding performance of RCC using 4 pt. graph and $\theta = .6$ and JBIG.

6.5 Proofs

6.5.1 Proof of Theorem VI.1

Proof. Let $1, \ldots, |V|$ be a lineal scan of the nodes. It follows that

$$
\begin{aligned}
p_{i|*}(x_i) &= \frac{p(x_{i-1}, x_i)}{p(x_{i-1})} \\
&= \frac{Z_{\{i-1,i\}}(x_{i-1}, x_i)}{Z_{i-1}(x_{i-1})} \\
&= \frac{\Phi_{\pi(i)}(x_{\pi(i)})\Psi_{i\pi(i)}(x_i, x_{\pi(i)})\Phi_i(x_i) \prod_{k\in\sigma i} m_{k\to i}(x_i) \prod_{j\in\partial\pi(i)\setminus i} m_{j\to\pi(i)}(x_{\pi(i)})}{\sum_{x'_i} \Phi_{\pi(i)}(x_{\pi(i)})\Psi_{i\pi(i)}(x'_i, x_{\pi(i)})\Phi_i(x'_i) \prod_{k\in\sigma i} m_{k\to i}(x'_i) \prod_{j\in\partial\pi(i)\setminus i} m_{j\to\pi(i)}(x_{\pi(i)})} \\
&= \frac{\Phi_{\pi(i)}(x_{\pi(i)})\Psi_{i\pi(i)}(x_i, x_{\pi(i)})\Phi_i(x_i) \prod_{k\in\sigma i} m_{k\to i}(x_i) \prod_{j\in\partial\pi(i)\setminus i} m_{j\to\pi(i)}(x_{\pi(i)})}{\Phi_{\pi(i)}(x_{\pi(i)}) \prod_{j\in\partial\pi(i)\setminus i} m_{j\to\pi(i)}(x_{\pi(i)}) \sum_{x'_i} \Psi_{i\pi(i)}(x'_i, x_{\pi(i)})\Phi_i(x'_i) \prod_{k\in\sigma i} m_{k\to i}(x'_i)} \\
&= \frac{\Psi_{i\pi(i)}(x_i, x_{\pi(i)})\Phi_i(x_i) \prod_{k\in\sigma i} m_{k\to i}(x_i)}{\sum_{x'_i} \Psi_{i\pi(i)}(x'_i, x_{\pi(i)})\Phi_i(x'_i) \prod_{k\in\sigma i} m_{k\to i}(x'_i)},
\end{aligned}
$$

where the second equality follows from the Markov property and the third equality from [30]. □

6.5.2 Proof of Proposition VI.4

Proof. Let $1, 2, \ldots, |V|$ be a scan of nodes in G such that for each i the set A_i is connected. Furthermore, let $\overline{G}$ be an unwrapping of G based on a loop cutset L. By construction, there is a one-to-one correspondence between edges of G and edges of $\overline{G}$. Now suppose k and j are in A_i for some i. Since we assume A_i to be connected, there is a sequence of nodes $p_1, p_2, \ldots, p_M \in A_i$ such that $\{k, p_1\}, \{p_1, p_2\}, \ldots, \{p_M, j\}$ are edges in E. Let $p_0 = k$ and $p_{M+1} = j$. Thus, by construction, there exists unique edges $\{\overline{p}_0, \overline{p}_1, \}, \{\overline{p}_1, \overline{p}_2\}, \ldots, \{\overline{p}_M, \overline{p}_{M+1}\}$ in $\overline{E}$ such that $\rho(\overline{p_i}) = p_i$ for $i = 0, 1, \ldots, M+1$.

Suppose that for some $i = 0, 1, \ldots, M+1$, p_i is a loop cutset node. Then the copy of p_i chosen to be in $\overline{A}_i$ is the unique node $\overline{p}_i$ indicated above. Thus if A_i is connected, then one can choose a corresponding $\overline{A}_i$ that is connected. That is, one can find a lineal sub-scan of $\overline{G}$. □

6.5.3 Proof of Proposition VI.5

Proof. Let G be a cyclic graph and $\overline{G}$ a connected unwrapping based on a loop cutset L. Let $A_i = \{1, \ldots, i\}$ be the set of the first i encoded nodes in G and $\overline{A}_i$ be the corresponding lineal sub-scan in $\overline{G}$. Furthermore, as in the proof of Lemma II.6 we use the shorthand

$$\overline{\Phi}_{\bar{A}}(\overline{x}_{\overline{A}}) = \prod_{\bar{i} \in \overline{A}} \Phi_{\bar{i}}(\overline{x}_{\bar{i}}) \prod_{\{\bar{i},\bar{j}\} \subset \overline{A}} \overline{\Psi_{\bar{i},\bar{j}}}(\overline{x}_{\bar{i}}, \overline{x}_{\bar{j}}).$$

for the product of the potentials on nodes and edges contained within the subset. We suppress the arguments for space and clarity, though these should of course be apparent from the subscripts on the potentials and messages. The optimal coding

distribution for a node i of the original graph is computed as

$$
\begin{aligned}
p_{i|*}(x_i) &= \frac{Z_{A_i}(x_{A_i})}{Z_{A_{i-1}}(x_{A_{i-1}})} \\
&= \frac{\sum_{x_{L\setminus A_i}} Z_{L\cup A_i}(x_{A_i}, x_{L\setminus A_i})}{\sum_{x_{L\setminus A_{i-1}}} Z_{L\cup A_{i-1}}(x_{A_{i-1}}, x_{L\setminus A_{i-1}})} \\
&= \frac{\sum_{x_{L\setminus A_i}} \overline{Z}^{(x_L)}_{\overline{A}_i}(\overline{x}_{\overline{A}_i})}{\sum_{x_{L\setminus A_{i-1}}} \overline{Z}^{(x_L)}_{\overline{A}_{i-1}}(\overline{x}_{\overline{A}_{i-1}})} \\
&= \frac{\sum_{x_{L\setminus A_i}} \prod_{\bar{j}\in\overline{A}_i} \overline{\Phi}_{\bar{j}} \prod_{\{\bar{j},\bar{k}\}\subset A_i} \overline{\Psi}_{\bar{j},\bar{k}} \prod_{\bar{k}\in\delta\overline{A}_i} \overline{m}^{(x_L)}_{\bar{k}\to\overline{A}_i}}{\sum_{x_{L\setminus A_{i-1}}} \prod_{\bar{j}\in\overline{A}_{i-1}} \overline{\Phi}_{\bar{j}} \prod_{\{\bar{j},\bar{k}\}\subset A_{i-1}} \overline{\Psi}_{\bar{j},\bar{k}} \prod_{\bar{k}\in\delta\overline{A}_{i-1}} \overline{m}^{(x_L)}_{\bar{k}\to\overline{A}_{i-1}}} \qquad (6.17) \\
&= \frac{\overline{\Phi}_{\bar{i}}\overline{\Psi}_{\bar{i},\pi(\bar{i})}\overline{\Phi}_{\overline{A}_{i-1}} \sum_{x_{L\setminus A_i}} \prod_{\bar{k}\in\partial\overline{A}_{i-1}\setminus\bar{i}} \overline{m}^{(x_L)}_{\bar{k}\to\bar{j}_{\bar{k}}} \prod_{\bar{k}\in\sigma\bar{i}} \overline{m}^{(x_L)}_{\bar{k}\to\bar{i}}}{\overline{\Phi}_{\overline{A}_{i-1}} \sum_{x_{L\setminus A_{i-1}}} \prod_{\bar{k}\in\partial\overline{A}_{i-1}\setminus\bar{i}} \overline{m}^{(x_L)}_{\bar{k}\to\bar{j}_{\bar{k}}} \overline{m}^{(x_L)}_{\bar{i}\to\pi(\bar{i})}} \qquad (6.18) \\
&= \frac{\overline{\Phi}_{\bar{i}}\overline{\Psi}_{\bar{i},\pi(\bar{i})} \sum_{x_{L\setminus A_i}} \prod_{\bar{k}\in\partial\overline{A}_{i-1}\setminus\bar{i}} \overline{m}^{(x_L)}_{\bar{k}\to\bar{j}_{\bar{k}}} \prod_{\bar{k}\in\sigma\bar{i}} \overline{m}^{(x_L)}_{\bar{k}\to\bar{i}}}{\sum_{x_{L\setminus A_{i-1}}} \prod_{\bar{k}\in\partial\overline{A}_{i-1}\setminus\bar{i}} \overline{m}^{(x_L)}_{\bar{k}\to\bar{j}_{\bar{k}}} \overline{m}^{(x_L)}_{\bar{i}\to\pi(\bar{i})}} \\
&= \frac{\overline{\Phi}_{\bar{i}}\overline{\Psi}_{\bar{i},\pi(\bar{i})} \sum_{x_{L\setminus A_i}} \prod_{\bar{k}\in\partial\overline{A}_{i-1}\setminus\bar{i}} \overline{m}^{(x_L)}_{\bar{k}\to\bar{j}_{\bar{k}}} \prod_{\bar{k}\in\sigma\bar{i}} \overline{m}^{(x_L)}_{\bar{k}\to\bar{i}}}{\sum_{x_{L\setminus A_{i-1}}} \prod_{\bar{k}\in\partial\overline{A}_{i-1}\setminus\bar{i}} \overline{m}^{(x_L)}_{\bar{k}\to\bar{j}_{\bar{k}}} \sum_{\overline{x}''_{\bar{i}}} \overline{\Phi}_{\bar{i}}\overline{\Psi}_{\bar{i},\pi(\bar{i})} \prod_{\bar{k}\in\sigma\bar{i}} \overline{m}^{(x_L)}_{\bar{k}\to\bar{i}}}, \qquad (6.19)
\end{aligned}
$$

where (6.17) follows from (4.14); (6.18) from the fact that configuration on $\overline{A}_i$ and $\overline{A}_{i-1}$ are constant for different values on $x_{L\setminus A_i}$ and $x_{L\setminus A_{i-1}}$, respectively; and (6.19) by substituting for the message $\overline{m}^{(x_L)}_{\bar{i}\to\pi(\bar{i})}$.

At this point we have made the basic substitutions in order to compute the desired ratio of beliefs in the original graph G, and the expression in (6.19) can be carried out with Global Conditioning. Moreover, the above holds whether or not i is a loop

cutset node. To complete the proof we now take into account the topology of the unwrapping $\overline{G}$. In particular, we recognize that the summations in the numerator and denominator of (6.19) can be distributed through the products of messages, because a number of the loop cutset nodes in $L \setminus A_i$ or $L \setminus A_{i-1}$ will be summable with respect to the appropriate messages incoming to $\overline{A}_i$.

We define

$$L_{\sigma(\overline{A})\setminus\overline{A}} = \bigcup_{\bar{k}\in\sigma(\overline{A})} L_{\bar{k}\setminus\bar{j}_{\bar{k}}} \tag{6.20}$$

The set of loop cutset nodes can therefore be expressed as

$$L = \left(R_{\overline{A}}, L_{\sigma(\overline{A})\setminus\overline{A}}\right) \tag{6.21}$$

Recall that for distinct $\bar{k}_1, \bar{k}_2 \in \partial\overline{A}$, the sets $L_{\bar{k}_1\setminus\bar{i}}$ and $L_{\bar{k}_2\setminus\bar{i}}$ are disjoint. In addition, we need to subtract out the loop cutset nodes that are also in A_i. This "set complementing" is implicit in the sets of loop cutset nodes that follow. In the lines that follow we consider the case that $i \notin L$. Note that under this assumption, the sets $L \setminus A_i$ and $L \setminus L_{i-1}$ are equal. Therefore, resuming with line (6.19),

$$
\begin{aligned}
p_{i|*}(x_i) &= \frac{\overline{\Phi}_{\bar{i}}\overline{\Psi}_{\bar{i},\pi(\bar{i})}\sum_{x_{L\setminus A_i}}\prod_{\bar{k}\in\partial\overline{A}_{i-1}\setminus\bar{i}}\overline{m}^{(x_L)}_{\bar{k}\to\bar{j}_{\bar{k}}}\prod_{\bar{k}\in\sigma\bar{i}}\overline{m}^{(x_L)}_{\bar{k}\to\bar{i}}}{\sum_{x_{L\setminus A_{i-1}}}\prod_{\bar{k}\in\partial\overline{A}_{i-1}\setminus\bar{i}}\overline{m}^{(x_L)}_{\bar{k}\to\bar{j}_{\bar{k}}}\sum_{\overline{x}''_{\bar{i}}}\overline{\Phi}_{\bar{i}}\overline{\Psi}_{\bar{i},\pi(\bar{i})}\prod_{\bar{k}\in\sigma\bar{i}}\overline{m}^{(x_L)}_{\bar{k}\to\bar{i}}} \\
&= \frac{\overline{\Phi}_{\bar{i}}\overline{\Psi}_{\bar{i},\pi(\bar{i})}\sum_{x_{L\setminus A_i}}\prod_{\bar{k}\in\partial\overline{A}_{i-1}\setminus\bar{i}}\overline{m}^{(x_L)}_{\bar{k}\to\bar{j}_{\bar{k}}}\prod_{\bar{k}\in\sigma\bar{i}}\overline{m}^{(x_L)}_{\bar{k}\to\bar{i}}}{\sum_{x_{L\setminus A_i}}\prod_{\bar{k}\in\partial\overline{A}_{i-1}\setminus\bar{i}}\overline{m}^{(x_L)}_{\bar{k}\to\bar{j}_{\bar{k}}}\sum_{\overline{x}''_{\bar{i}}}\overline{\Phi}_{\bar{i}}\overline{\Psi}_{\bar{i},\pi(\bar{i})}\prod_{\bar{k}\in\sigma\bar{i}}\overline{m}^{(x_L)}_{\bar{k}\to\bar{i}}} \qquad (6.22) \\
&= \frac{\overline{\Phi}_{\bar{i}}\overline{\Psi}_{\bar{i},\pi(\bar{i})}\sum_{x_{R_{\overline{A}_i}}}\sum_{x_{L_{\sigma(\overline{A}_i)\setminus\overline{A}_i}}}\prod_{\bar{k}\in\partial\overline{A}_{i-1}\setminus\bar{i}}\overline{m}^{(x_L)}_{\bar{k}\to\bar{j}_{\bar{k}}}\prod_{\bar{k}\in\sigma\bar{i}}\overline{m}^{(x_L)}_{\bar{k}\to\bar{i}}}{\sum_{\overline{x}''_{\bar{i}}}\overline{\Phi}_{\bar{i}}\overline{\Psi}_{\bar{i},\pi(\bar{i})}\sum_{x_{R_{\overline{A}_i}}}\sum_{x_{L_{\sigma(\overline{A}_i)\setminus\overline{A}_i}}}\prod_{\bar{k}\in\partial\overline{A}_{i-1}\setminus\bar{i}}\overline{m}^{(x_L)}_{\bar{k}\to\bar{j}_{\bar{k}}}\prod_{\bar{k}\in\sigma\bar{i}}\overline{m}^{(x_L)}_{\bar{k}\to\bar{i}}} \qquad (6.23) \\
&= \frac{\overline{\Phi}_{\bar{i}}\overline{\Psi}_{\bar{i},\pi(\bar{i})}\sum_{x_{R_{\overline{A}_i}}}\sum_{x_{L_{\sigma(\bar{i})\setminus\bar{i}}}}\prod_{\bar{k}\in\sigma\bar{i}}\overline{m}^{(x_L)}_{\bar{k}\to\bar{i}}\prod_{\bar{k}\in\partial\overline{A}_{i-1}\setminus\bar{i}}\sum_{x_{L_{\bar{k}\setminus\bar{j}_{\bar{k}}}}}\overline{m}^{(x_L)}_{\bar{k}\to\bar{j}_{\bar{k}}}}{\sum_{\overline{x}''_{\bar{i}}}\overline{\Phi}_{\bar{i}}\overline{\Psi}_{\bar{i},\pi(\bar{i})}\sum_{x_{R_{\overline{A}_i}}}\sum_{x_{L_{\sigma(\bar{i})\setminus\bar{i}}}}\prod_{\bar{k}\in\sigma\bar{i}}\overline{m}^{(x_L)}_{\bar{k}\to\bar{i}}\prod_{\bar{k}\in\partial\overline{A}_{i-1}\setminus\bar{i}}\sum_{x_{L_{\bar{k}\setminus\bar{j}_{\bar{k}}}}}\overline{m}^{(x_L)}_{\bar{k}\to\bar{j}_{\bar{k}}}} \qquad (6.24) \\
&= \frac{\overline{\Phi}_{\bar{i}}\overline{\Psi}_{\bar{i},\pi(\bar{i})}\sum_{x_{R_{\overline{A}_i}}}\sum_{x_{L_{\sigma(\bar{i})\setminus\bar{i}}}}\prod_{\bar{k}\in\sigma\bar{i}}\overline{m}^{(x_L)}_{\bar{k}\to\bar{i}}\prod_{\bar{k}\in\partial\overline{A}_{i-1}\setminus\bar{i}}\widehat{m}^{(x_{R_{\bar{k}\bar{j}_{\bar{k}}}})}_{\bar{k}\to\bar{j}_{\bar{k}}}}{\sum_{\overline{x}''_{\bar{i}}}\overline{\Phi}_{\bar{i}}\overline{\Psi}_{\bar{i},\pi(\bar{i})}\sum_{x_{R_{\overline{A}_i}}}\sum_{x_{L_{\sigma(\bar{i})\setminus\bar{i}}}}\prod_{\bar{k}\in\sigma\bar{i}}\overline{m}^{(x_L)}_{\bar{k}\to\bar{i}}\prod_{\bar{k}\in\partial\overline{A}_{i-1}\setminus\bar{i}}\widehat{m}^{(x_{R_{\bar{k}\bar{j}_{\bar{k}}}})}_{\bar{k}\to\bar{j}_{\bar{k}}}} \qquad (6.25) \\
&= \frac{\overline{\Phi}_{\bar{i}}\overline{\Psi}_{\bar{i},\pi(\bar{i})}\sum_{x_{R_{\overline{A}_i}}}\prod_{\bar{k}\in\partial\overline{A}_{i-1}\setminus\bar{i}}\widehat{m}^{(x_{R_{\bar{k}\bar{j}_{\bar{k}}}})}_{\bar{k}\to\bar{j}_{\bar{k}}}\sum_{x_{L_{\sigma(\bar{i})\setminus\bar{i}}}}\prod_{\bar{k}\in\sigma\bar{i}}\overline{m}^{(x_L)}_{\bar{k}\to\bar{i}}}{\sum_{\overline{x}''_{\bar{i}}}\overline{\Phi}_{\bar{i}}\overline{\Psi}_{\bar{i},\pi(\bar{i})}\sum_{x_{R_{\overline{A}_i}}}\prod_{\bar{k}\in\partial\overline{A}_{i-1}\setminus\bar{i}}\widehat{m}^{(x_{R_{\bar{k}\bar{j}_{\bar{k}}}})}_{\bar{k}\to\bar{j}_{\bar{k}}}\sum_{x_{L_{\sigma(\bar{i})\setminus\bar{i}}}}\prod_{\bar{k}\in\sigma\bar{i}}\overline{m}^{(x_L)}_{\bar{k}\to\bar{i}}} \\
&= \frac{\overline{\Phi}_{\bar{i}}\overline{\Psi}_{\bar{i},\pi(\bar{i})}\sum_{x_{R_{\overline{A}_i}}}\prod_{\bar{k}\in\partial\overline{A}_{i-1}\setminus\bar{i}}\widehat{m}^{(x_{R_{\bar{k}\bar{j}_{\bar{k}}}})}_{\bar{k}\to\bar{j}_{\bar{k}}}\prod_{\bar{k}\in\sigma\bar{i}}\sum_{x_{L_{\bar{k}\setminus\bar{i}}}}\overline{m}^{(x_L)}_{\bar{k}\to\bar{i}}}{\sum_{\overline{x}''_{\bar{i}}}\overline{\Phi}_{\bar{i}}\overline{\Psi}_{\bar{i},\pi(\bar{i})}\sum_{x_{R_{\overline{A}_i}}}\prod_{\bar{k}\in\partial\overline{A}_{i-1}\setminus\bar{i}}\widehat{m}^{(x_{R_{\bar{k}\bar{j}_{\bar{k}}}})}_{\bar{k}\to\bar{j}_{\bar{k}}}\prod_{\bar{k}\in\sigma\bar{i}}\sum_{x_{L_{\bar{k}\setminus\bar{i}}}}\overline{m}^{(x_L)}_{\bar{k}\to\bar{i}}} \qquad (6.26) \\
&= \frac{\overline{\Phi}_{\bar{i}}\overline{\Psi}_{\bar{i},\pi(\bar{i})}\sum_{x_{R_{\overline{A}_i}}}\prod_{\bar{k}\in\partial\overline{A}_{i-1}\setminus\bar{i}}\widehat{m}^{(x_{R_{\bar{k}\bar{j}_{\bar{k}}}})}_{\bar{k}\to\bar{j}_{\bar{k}}}\prod_{\bar{k}\in\sigma\bar{i}}\widehat{m}^{(x_{R_{\bar{k}\bar{i}}})}_{\bar{k}\to\bar{i}}}{\sum_{\overline{x}''_{\bar{i}}}\overline{\Phi}_{\bar{i}}\overline{\Psi}_{\bar{i},\pi(\bar{i})}\sum_{x_{R_{\overline{A}_i}}}\prod_{\bar{k}\in\partial\overline{A}_{i-1}\setminus\bar{i}}\widehat{m}^{(x_{R_{\bar{k}\bar{j}_{\bar{k}}}})}_{\bar{k}\to\bar{j}_{\bar{k}}}\prod_{\bar{k}\in\sigma\bar{i}}\widehat{m}^{(x_{R_{\bar{k}\bar{i}}})}_{\bar{k}\to\bar{i}}}, \qquad (6.27)
\end{aligned}
$$

where (6.22) follows from the assumption that $i \notin L$; (6.23) from the partitioning of

the loop cutset nodes; (6.24) from distributing the summation of loop cutset nodes that are children of $\overline{A}_i$; (6.25) follows from (IV.14); and (6.26) and (6.27) follow for the same reasons as (6.24) and (6.25).

Now, if $i \in L$, we can obviously express the summation $\sum_{x_{L\setminus A_{i-1}}}$ as $\sum_i \sum_{x_{L\setminus A_i}}$. In other words, everything will be the same as above except that in the denominators there will be a $\sum_i$ at the far left. So, for $i \in L$, the optimal coding distribution is

$$
\begin{aligned}
p_{i|*}(x_i) &= \frac{\overline{\Phi}_{\bar{i}}\overline{\Psi}_{\bar{i},\pi(\bar{i})} \sum_{x_{L\setminus A_i}} \prod_{\bar{k}\in\partial\overline{A}_{i-1}\setminus\bar{i}} \overline{m}^{(x_L)}_{\bar{k}\to\bar{j}_{\bar{k}}} \prod_{\bar{k}\in\sigma\bar{i}} \overline{m}^{(x_L)}_{\bar{k}\to\bar{i}}}{\sum_{x_{L\setminus A_{i-1}}} \prod_{\bar{k}\in\partial\overline{A}_{i-1}\setminus\bar{i}} \overline{m}^{(x_L)}_{\bar{k}\to\bar{j}_{\bar{k}}} \sum_{\overline{x}''_{\bar{i}}} \overline{\Phi}_{\bar{i}}\overline{\Psi}_{\bar{i},\pi(\bar{i})} \prod_{\bar{k}\in\sigma\bar{i}} \overline{m}^{(x_L)}_{\bar{k}\to\bar{i}}} \\
&= \frac{\overline{\Phi}_{\bar{i}}\overline{\Psi}_{\bar{i},\pi(\bar{i})} \sum_{x_{L\setminus A_i}} \prod_{\bar{k}\in\partial\overline{A}_{i-1}\setminus\bar{i}} \overline{m}^{(x_L)}_{\bar{k}\to\bar{j}_{\bar{k}}} \prod_{\bar{k}\in\sigma\bar{i}} \overline{m}^{(x_L)}_{\bar{k}\to\bar{i}}}{\sum_{x_i}\sum_{x_{L\setminus A_{i-1}}} \prod_{\bar{k}\in\partial\overline{A}_{i-1}\setminus\bar{i}} \overline{m}^{(x_L)}_{\bar{k}\to\bar{j}_{\bar{k}}} \sum_{\overline{x}''_{\bar{i}}} \overline{\Phi}_{\bar{i}}\overline{\Psi}_{\bar{i},\pi(\bar{i})} \prod_{\bar{k}\in\sigma\bar{i}} \overline{m}^{(x_L)}_{\bar{k}\to\bar{i}}} \qquad (6.28) \\
&= \frac{\overline{\Phi}_{\bar{i}}\overline{\Psi}_{\bar{i},\pi(\bar{i})} \sum_{x_{R_{\overline{A}_i}}} \prod_{\bar{k}\in\partial\overline{A}_{i-1}\setminus\bar{i}} \widehat{m}^{(x_{R_{\bar{k}\bar{j}_{\bar{k}}}})}_{\bar{k}\to\bar{j}_{\bar{k}}} \prod_{\bar{k}\in\sigma\bar{i}} \widehat{m}^{(x_{R_{\bar{k}\bar{i}}})}_{\bar{k}\to\bar{i}}}{\sum_{x_i}\sum_{\overline{x}''_{\bar{i}}} \overline{\Phi}_{\bar{i}}\overline{\Psi}_{\bar{i},\pi(\bar{i})} \sum_{x_{R_{\overline{A}_i}}} \prod_{\bar{k}\in\partial\overline{A}_{i-1}\setminus\bar{i}} \widehat{m}^{(x_{R_{\bar{k}\bar{j}_{\bar{k}}}})}_{\bar{k}\to\bar{j}_{\bar{k}}} \prod_{\bar{k}\in\sigma\bar{i}} \widehat{m}^{(x_{R_{\bar{k}\bar{i}}})}_{\bar{k}\to\bar{i}}}, \qquad (6.29)
\end{aligned}
$$

where the sequence of steps between (6.28) and (6.29) are the same as those between (6.22) and (6.27) in the case where $i \notin L$. □

CHAPTER VII

Concluding Remarks

This thesis began by motivating the source coding or compression problem of observing an image $\mathbf{x}$, encoding $\mathbf{x}$ into a string of bits to be stored on a computer, then recovering a reconstruction $\hat{\mathbf{x}}$ of the original image from the stored sequence of 1s and 0s. We introduced both lossless and lossy compression, in which the reconstruction $\hat{\mathbf{x}}$ is, respectively, identical and (in general) not identical to the original image $\mathbf{x}$. Furthermore, we stated that the primary goal was to develop a framework for the source coding problem when the image $\mathbf{x}$ is drawn from a Markov random field (MRF) distribution $\mathbf{X}$ defined on a graph $G = (V, E)$, but that along the way we would encounter problems that are of interest in their own right.

There are two reasons for focusing on MRFs in this thesis. The first and most basic is that methods for compression of MRFs had not been pursued before in the literature. The second and more important is that MRFs are an increasingly significant class of probability distributions, finding manifold application and having been studied extensively in many of the areas that appear in the analytical and algorithmic aspects of the source coding problem. In this thesis we succeeded in providing a foundation for studying the compression of MRFs, and indeed, many of the contributions in this thesis apply to areas of MRF research that extend beyond source coding.

In the following sections we discuss the contributions of this thesis, suggest areas where the results in this thesis can be pursued further and mention possible approaches, and discuss connections to other areas.

7.1 Contributions

From a high-level perspective, the principal contribution of this thesis is the development of a framework for compressing a Markov random field distribution. This was a topic that had not been taken up in the literature. Although groups had looked at existential limits of lossy source coding of MRFs under different scenarios, no one had looked at definite approaches to doing so. The key feature of our method, the use of a cutset to simplify processing, has been used in other contexts [31], and indeed, is likely the most natural one to consider in view of the underlying graphical structure associated with an MRF and the accompanying Markov property. In particular, we presented either general rules or examples of the three key aspects of the cutset based coding method: lossless compression of the cutset; processing the remainder conditioned on the cutset; analysis of the approximation. Moreover, as components of the lossless and processing stages, we develop an exact inference algorithm for cyclic graphs. And while the emphasis is on cutset based coding methods, the author strongly suspects that the tools presented in this thesis will be useful in other (non cutset based) source coding techniques for MRFs.

We focused on Arithmetic Encoding (AC) as the means of compressing an MRF, and this requires an inference algorithm to compute the required coding distributions. For acyclic MRFs, the standard Belief Propagation (BP) algorithm is used to compute the optimal AC coding distributions in a two-step process. The acyclic encoding algorithm is the basis for the two optimal AC encoding algorithms for EASY cyclic MRFs presented. The two such methods discussed in this thesis are based on converting the original cyclic graph into a new graph that does not contain any cycles.

In the first, a well-known clustering algorithm is used to form an acyclic graph of supernodes and then the acyclic encoding algorithm is used. In the second, the LC algorithm is used to compute the optimal coding distributions, essentially a matter of running the acyclic encoding algorithm on an unwrapped graph. For a HARD cyclic MRF, a cutset is chosen such that the subgraphs induced by the cutset and the remaining components are EASY; the cutset is losslessly AC encoded; then the remaining components are losslessly AC encoded conditioned on the cutset. In this way our approach to lossless encoding of an MRF is hierarchical, with the acyclic encoding algorithm as the starting point.

For cyclic graphs we rigorously developed the Local Conditioning (LC) inference algorithm through a solid presentation of the basic Global Conditioning algorithm. The principles of the LC algorithm presented in this thesis are essentially the same as those in LC algorithm previously established for Bayesian networks [38, 46, 16, 14, 20]. One of the main differences is that MRFs are defined on undirected graphs, whereas Bayesian networks are defined on graphs with directed edges, and it is the lack of direction on the edges that makes the analysis more straightforward. Our analysis and resulting formulas allow us to compare different unwrappings for a given loop cutset on a particular graph. Moreover, the author believes that the analysis afforded by considering undirected graph makes the underlying principles of splitting nodes more accessible. While we presented LC in the context of the Sum-Product algorithm, it holds equally well for Max-Product, and indeed applies to any problem that can be cast in the framework of the Generalized Distributive Law [1]. As MRFs are used to model systems that are necessarily distributed, Local Conditioning is a much more natural exact inference algorithm to employ than a clustering algorithm, as grouping nodes together may not be practical or even make sense in some cases.

For lossy compression, the idea proposed is to again losslessly compress a cutset of sites, but instead of encoding the remaining components we estimate them condi-

tioned on their boundaries. In this thesis we considered the homogeneous Ising model. We gave analytical solutions for the MAP reconstruction of a block conditioned on its boundary under the more commonly occurring special cases of two or fewer runs of black (white) pixels. We did this for both the 4 pt. and 8 pt. grid graphs. In addition to the potential application of these reconstruction results to other sampling problems, they were used in this thesis to demonstrate the optimality of Max-Product LBP as a tool for interpolating from the boundary. When one considers that the performance of LBP is generally unknown, this is significant because it might be used not only to show LBP's performance in this particular problem, but also to gain insight into the algorithm in general. In addition, we touched on the relationship between cutset based coding and erasure entropy [51, 52] and how the latter may be used to analyze the former.

In both the (general) lossless and lossy coding methods discussed in this thesis, a cutset of sites is initially encoded. In the lossless RCC algorithm we explicitly stated that we wanted to encode the cutset using a reduced MRF coding distribution, but this can also be done in the lossy case. Encoding the cutset in this way necessitates an analysis of the approximation between the marginal distribution of the cutset and the reduced MRF coding distribution. The key tool used in analyzing these approximations is the exponential family representation of an MRF, which has proven to be quite useful in dealing with MRFs [53]. We established a number of useful results for situations where a thinned MRF is used, and in particular, a reduced MRF model is used as an approximation to the marginal distribution for a given subset of nodes. We gave the first results of which the author is aware for the monotonicity of entropy, which was later used to show a reduced MRF entropy inequality. We also showed that in order to preserve the moments on a reduced MRF, the corresponding exponential coordinates are component-wise greater than those in the original specification. We likewise showed monotonicity of divergence and pythagorean decomposition for

reduced MRF compared to marginal distribution. We stated conjectures about the monotonicity of mutual information and the assymmetry of divergence. The analytical methods, in particular the Pythagorean decomposition, was used in optimizing the suboptimal efficient method, Reduced Cutset Coding, for lossless compression of HARD cyclic MRFs. Furthermore, the theoretical guarantees allow us estimate upper and lower bounds to the entropy of the MRF being compressed. In addition, the gap between these estimates gives us an estimate to the divergence between the marginal distribution on the cutset and the reduced MRF used to encode it.

In addition to establishing the algorithmic and analytical theory behind cutset based compression of MRFs, we also demonstrated the performance of our techniques on simulated Ising images. As the Ising model is the oldest MRF studied and the simplest example, exhibiting the performance of our method on the Ising model gives our presentation a completeness that is important in laying the groundwork for future study of the compression of MRFs. From the theoretical guarantees of Section 5.3.2 and application of the RCC method, we obtained estimates of upper and lower bounds to the entropy of an Ising model, and observed empirically the these bounds are very close, giving an estimate to the entropy of the Ising model, a quantity that has great information-theoretic significance but due to computational difficulty has eluded MRF researchers up to this point. In addition, we demonstrated the applicability of the lossy cutset coding method to real-world images and showed that its performance was competitive compared to other lossy bilevel coding methods. In particular, the grid cutset preserves edges from the original image so that this technique of keeping grid lines appears as though it may be useful in encoding non-binary images. We also demonstrated a simple adaptive method for losslessly encoding the grid cutset of a general binary image.

7.2 Suggestions for Future Work

While the author views this thesis as successful in laying a foundation for the compression of Markov random fields, more questions were raised in the process than specific answers provided. And perhaps this is as it should be. This is a very exciting time in the research of MRFs in general, with many groups around the world investigating model selection, erasure entropy and related quantities, inference algorithms for MRFs and more. This thesis has added source coding to the list of areas in which MRF research is being conducted. And as many of the specific results within this thesis concern general MRF-related problems, we feel that there is much to be gained by pursuing some of the results in this thesis further.

7.2.1 Grid Cutset Encoding

In the lossy cutset coding method presented for the Ising model there are many things that are of practical and theoretical interest that can be explored further.

7.2.1.1 Lossless Grid Compression with RCC

Another area of further exploration is to apply the lossless coding method presented in Section 6.2 to the grid cutset. This can be done by first noting that the subgraph induced by the grid is *homeomorphic* to the 4 pt. grid. Stated simply, we can obtain the 4 pt. grid graph from the grid cutset induced graph by removing nodes of degree two, which in the context of MRFs corresponds to summing out or marginalizing the random variables associated with these nodes. By marginalizing or deleting nodes of degree two from the original grid induced subgraph, we will have a 4 pt. grid graph, the nodes of which will correspond to those nodes of the original grid that are contained in both a row and a column of the grid. For convenience we will refer to these nodes as *intersection* nodes.

This means that we can model the subset of intersection nodes as an MRF on

the 4 pt. grid graph, and then apply the RCC encoding method described in Section 6.2 to this model. Then, once the intersection nodes are encoded, the remaining grid cutset nodes can be encoded conditioned on the intersection nodes. The overall redundancy can be decomposed as the redundancy for encoding the intersection nodes and the redundancy for encoding the remaining grid cutset nodes. The redundancy on the intersection nodes is the divergence between the marginal distribution on the intersection nodes and the MRF on the 4 pt. grid used to encode them. The redundancy on the remaining grid cutset nodes is the divergence between the true conditional distribution of the remaining grid cutset nodes given the intersection nodes and the conditional distribution of said nodes in the reduced MRF on the grid cutset induced graph.

7.2.1.2 Rate Distortion Analysis

We discussed briefly in Section 3.6 that the concept of erasure entropy can be used to analyze the rate of lossy cutset coding by noting that the entropy of, say, an Ising model, is the entropy of the grid plus the conditional entropies of the remaining blocks conditioned on the grid. A straightforward thing that one can do is to use the Gibbs Sampler to obtain a very close estimate to the conditional entropy of a block given its boundary. Combining this with estimates for the entropy of the model, obtained via the lossless RCC method of Section 6.2, we get an estimate for the entropy of the grid cutset, and comparing this to the encoding rates using either the simple adaptive method presented in Section or the RCC method applied to the grid.

Another thing that would be interesting is to apply the concept of erasure rate distortion to determine, at least approximately, the erasure rate distortion function [51, 52] for encoding a block from its boundary. At one extreme is the case of lossless encoding of a block given its boundary and at the other is the situation presently pursued, sending no extra information and simply estimating. One benefit from such

analysis is that it could possibly be used to quantify the suboptimality in cutset based lossy coding by comparing against bounds to the rate distortion function for the Ising model [36].

7.2.1.3 Lossy Compression of Grid

Instead of reconstructing the cutset perfectly at the decoder, we can allow some distortion between the original and decoded grids. This could give more rate-distortion tradeoff, perhaps allowing the basic cutset based coding method to approach optimal lossy performance. Another potential benefit is that it could further reduce the required bit rate without substantial loss in the quality of the reconstruction. For instance, in the lossy coding of the homogeneous Ising model the only parameter that matters in the estimation is the number of odd bonds. Therefore, one can imagine allowing the values on the grid to be altered slightly in a way that optimizing the block interiors with respect to the new grid results in reconstructions that are structurally very similar to the reconstructions that are possible through MAP reconstructing the interiors from the original grid. Moreover, as we see in the reconstructions in Figure 3.11, if the original image is rather noisy[1], then there will be perceptible distortion between the MAP reconstruction of the entire image and the original. This is because the MAP reconstruction algorithm produces smooth block interiors, while the noise on the original will be preserved, creating a discontinuity. If the grid were lossy encoded/decoded in a way that, for example, resulted in a smoother grid, then the resulting MAP estimated of the remainder would be more "compatible".

7.2.1.4 Hierarchical Grid Encoding

The lossy cutset coding method presented in Chapter III uses a fixed size grid, where the block size is an algorithm parameter that can be chosen according to rate-

[1]By this we mean simply that there are regions of the image in which there are many odd bonds.

distortion performance requirements. One could also use an adaptive grid with a given fixed starting size. Then, at the encoder, if a block is observed to have more than two boundary runs[2], the encoder would subdivide the block, encoding the column and row required to bisect the original block. The additional bits produced by the encoder for subdividing blocks would be stored in a separate bit array. The decoder, likewise observing that the block boundary has more than two runs, will access the additional rows and column bits from the appropriate bit string and reconstruct the subdivided block.

7.2.2 Max-Product BP for Ising Interpolation

In Section 3.3 we discussed the use of Max-Product BP to do the interpolation of the block interiors from the block boundaries. We showed empirically, using the results of Section 3.2, that for block boundaries consisting of the common boundary configurations having two or fewer runs, Max-Product BP converged and did so to the correct values. A very interesting problem directly motivated by these results is a proof of the correctness of Max-Product BP, for either the limited boundary configuration cases mentioned above, or as stated in Conjecture III.11, for arbitrary boundary configurations. A line of attack that may prove fruitful is the *unwrapped graph*[3] introduced in [58] as a means of analyzing the iterates of LBP. Using the unwrapped graph approach, the beliefs at a given site i at iteration n are known to equal the beliefs computed at the root of the depth n tree unwrapped at node i. Because the Max-Product beliefs converge after a finite number of iterations, it is clear that after some number of iterations, the addition of subtrees to the unwrapped graph has no effect on the beliefs computed at the root. Moreover, because the converged beliefs are correct, the leaves of some depth n' unwrapped graph capture the boundary information required to exactly determine the beliefs. But the analytical solutions

[2]Or meet some other performance-based criterion for that matter.

[3]The unwrapped graph in [58] is different from that discussed in Chapter IV.

derived in Section 3.3 looked at the boundary as a whole, or rather the pattern of runs on the boundary. Therefore, it seems a viable proof of LBP's correctness will arise through demonstration that the leaves of a particular unwrapped graph capture this run information sufficiently. The author feels strongly that such a proof is possible. Once an understanding of the LBP dynamics in the homogenous interpolation problem is obtained, further steps would be the cases where uniform self potentials and non-uniform edge potentials are incorporated into the model specification.

7.2.3 Optimal Loop Cutsets for 4 pt. Grid Graph

In Section 4.4 used the results of Theorem VI.8 to compare the complexity of performing LC on a 4 pt. grid graph using the checkerboard loop cutset with three different unwrappings. An interesting problem to consider is finding an optimal pair of loop cutset and unwrapped graph based on the loop cutset for the 4 pt. grid graph. For instance, one can consider the family of (infinite) subgrids defined by

$$\Lambda_k = \{(i2^k + 1, j2^k + 1) : i, j \geq 0\}. \tag{7.1}$$

Letting $I_n = \{1, \ldots, n\}^2$, we can express the checkerboard loop cutset as

$$L_1 = I_n \cap (\Lambda_1 \cup (\Lambda_1 + (1, 1))) . \tag{7.2}$$

A second loop cutset L_2 can be defined similarly as

$$L_2 = I_n \cup (\Lambda_1 \cup (\Lambda_2 + (1, 1)) \cup (\Lambda_2 + (3, 3))) , \tag{7.3}$$

and in this way we can define a family of loop cutsets for the 4 pt. grid graph, based on the subgrid Λ_i. Continuing in this fashion, we can see that subsequent loop cutsets will be sparser.

7.2.4 Monotonicity

Some very interesting monotonicity results for MRFs remain to be shown. Conjecture V.11, which states that the asymmetry of the divergence between two MRFs follows a monotonic relationship, remains to be shown. This would be a very useful result as it would shed light on the general asymmetry of divergence between two probability distributions. The line of attack that seems most apparent to the author would be to use the log-partition function and first order Taylor series expansion. Another conjecture that was made that requires a proof is that the Mutual Information between two mutually exclusive and collectively exhaustive subsets of nodes $U, W \subset V$ is monotone increasing in the exponential parameters. This question can be approached in a couple of different ways, by considering the increased exponential parameter to weight an edge contained within one of the subsets or connecting a node in one to a node in the other. If the MRF is positively correlated and the edge weights are nonnegative, then monotonicity likely holds in either case. It would be an interesting additional exercise to demonstrate that the increase in Mutual Information is greater if the increased exponential parameter connects a node in one subset to a node in the other. In addition to these conjectures, one could also try and show that the marginal entropy of a subset of nodes is monotone in the exponential parameters.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] S. M. Aji and R. J. McCliece. "the generalized distributive law". *IEEE Trans. Info. Thy.*, 46(2):325–343, March 2000.

[2] S. Amari. "information geometry on hierarchy of probability distributions". *IEEE Trans. Info. Thy.*, 47(5):1701–1711, July 2001.

[3] S. Amari and H. Nagoaka. *"Methods of Information Geometry"*. Oxford University Press, 1993.

[4] O. E. Barndorrf-Nielson. *"Information and Exponential Families"*. Wiley, Chichester, U. K., 1978.

[5] R. J. Baxter. *"Exactly Solved Models in Statistical Mechanics"*. 1982.

[6] A. Becker and D. Geiger. "approximation algorithms for the loop cutset problem". In *Journal of Artificial Intelligence Research*, pages 60–68. Morgan Kauffman, 1994.

[7] H. A. Bethe. "statistical theory of superlattices". *Proc. Roy. Soc.*, (A150):552–575, 1935.

[8] "F. Comets". "on consistency of a class of estimators for exponential families of markov random fields on the lattice". *The Annals of Statistics*, 20(1):455–468, 1992.

[9] T. M. Cover and J. A. Thomas. *"Elements of Information Theory"*. Wiley Inter Science, 2nd edition, 2006.

[10] I. Csiszar. "i-divergence geometry of probability distributions and minimization problems". *The Annals of Probability*, 3(1):146–158, January 1975.

[11] I. Csiszar and Z. Talata. "consistent estimation of the basic neighborhood structure of markov random fields". *Th'e Annals of Statistics*, 34(1):123–145, 2006.

[12] J. N. Darroch and D. Ratcliff. "generalized iterative scaling for log-linear models". *Annals of Mathematical Statistics*, 43(5):1470–1480, 1972.

[13] Adnan Darwiche. "conditioning methods for exact and approximate inference in causal networks". In *"In Proceedings of the 11th Conference on Uncertainty in Artificial Intelligence"*, pages 99–107. Morgan Kauffman, 1995.

[14] Adnan Darwiche. "recursive conditioning". *Artificial Intelligence*, 126:5–41, February 2001.

[15] X. Descombes, R. D. Morris, J. Zerubia, and M. Berthod. "estimation of markov random field prior parameters using markov chain monte carlo maximum likelihood". *IEEE Trans. Image Proc.*, 8(7):954–963, July 1999.

[16] F.J. Diez. "local conditioning in bayesian networks". *Artificial Intelligence*, 87:1–20, 1996.

[17] R. L. Dobrushin. "the description of a random field by means of conditional probabilities and conditions of its regularity". *Theor. Prob. Appl.*, 13:197–224, 1968.

[18] R. L. Dobrushin. "prescribing a system of random variables by conditional distributions". *Theor. Prob. Appl.*, 15:458–486, 1970.

[19] R. L. Dobrushin. "markov processes with a large number of locally interacting components". *Problems of Information Transmission*, 7:149–164, 235–241, 1971.

[20] A. Fay and J. Jaffray. "a justification for local conditioning in bayesian networks". *Intnl. Jrnl. Approximate Reasoning*, 24(1):59–81, April 2000.

[21] S. Geman and D. Geman. "stochastic relaxation, gibbs distributions, and the bayesian restoration of images". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):721–741, 1984.

[22] S. Geman and C. Graffigne. "markov random field image models and their application to computer vision". In *Proceddings of the Intl. Congress of Mathematicians*, pages 1496–1516, Berkeley, CA, 1986.

[23] C. J. Geyer and E. A. Thompson. "constrained monte carlo maximum likelihood for dependent data". *Jrnl. Royal Stat. Soc. B*, 54(3):657–699, 1992.

[24] R. B. Griffiths. "correlations in ising ferromagnets i.". *Jrnl. Math. Physics*, 8(3):478, March 1967.

[25] J. M. Hammersley and P. Clifford. "markov fields on finite graphs and lattices". 1971.

[26] T. S. Han and S. Amari. "statistical inference under multiterminal data compression". *IEEE Trans. Info. Thy.*, 44(6):2300–2323, October 1998.

[27] P. G. Howard, F. Kossentini, B. Martins, S. Forchhammer, and W. J. Rucklidge. "the emerging jbig2 standard". *IEEE Trans. on Circuits and Systems for Video Technology*, 8(7):838–848, 1998.

[28] K. Culik II and V. Valenta. "finite automata based compression of bi-level and simple color images". *Computers and Graphics*, 21(1):61–68, 1997.

[29] E. Ising. "beitrag sur theorie des ferromagnetismus". *Zeit. fur Physik*, 31:253–258, 1925.

[30] F. V. Jensen. *"An Introduction to Bayesian Networks"*. Springer, 1997.

[31] J. K. Johnson and A. S. Willsky. "a recursive model-reduction method for approximate inference in gaussian markov random fields". *IEEE Trans. Image Proc.*, 17(1):70–83, January 2008.

[32] R. Kikuchi. "a theory of cooperative phenomena". *Physical Review*, 81, 1951.

[33] U. Kjaerulff. "approximation of bayesian networks thorugh edge removals". Technical Report DK-9220, Aalborg University, Aalborg, Denmark, August 1993.

[34] I. Kontoyiannis. "pattern matching and lossy data compression on random fields". *IEEE Trans. Info. Thy.*, 49(4):1047–1051, April 2003.

[35] S. Lauritzen. *"Graphical Models"*. Oxford University Press, 1996.

[36] T. A. Lee. "on the rate distortion function of the ising model". Master's, Cornell, 1984.

[37] K. P. Murphy, Y. Weiss, and M. I. Jordan. "loopy belief propagation for approximate inference: An empirical study". In *In Proceedings of Uncertainty in AI*, pages 467–475, 1999.

[38] J. Pearl. "a constraint-propagation approach to probabilisitc reasoning". CS-850020 R-44, UCLA, Los Angelos, CA, October 1985.

[39] J. Pearl. *"Probabilisitic Reasoning in Intelligent Systems"*. Morgan Kauffman Publishers, San Francisco, CA, USA, 1988.

[40] R. E. Peierls. "on ising's ferromagnet model". *Proc. Camb. Phil. Soc.*, 32:477–481, 1936.

[41] C. J. Preston. *"Gibbs States on Countable Sets"*. Cambridge University Press, 1974.

[42] J. Rissanen. "a universal data compression system". *IEEE Trans. Info. Thy.*, 29(5):656–664, September 1983.

[43] J. Rissanen. "a universal prior for integers and estimation by minimum description length". *The Annals of Statistics*, 11(2):416–431, 1983.

[44] J. Rissanen and G. G. Langdon. "arithmetic codng". *IBM Jrnl. Res. Dev.*, 23(2):149–162, 1979.

[45] K. Sayood. *"Introduction to Data Compression"*. Academic Press, 2nd edition, 200.

[46] R. D. Schachter, S. K. Anderson, and P. Szolovits. "global conditioning for probabilistic inference in bayesian networks". pages 514–522, San Mateo, CA, 1994. Morgan Kaufmann.

[47] C. E. Shannon. "communication in the presence of noise". *Proc. Institute of Radio Engineers*, 37(1):10–21, January 1949.

[48] F. Spitzer. "random fields and interacting particle systems". M.A.A. Summer Seminar Notes, 1971.

[49] H. J. Suermondt and Gregory F. Cooper. "probabilistic inference in multiply connected belief networks using loop cutsets". *Int. Nat. Jrnl. App. Reasoning*, 4:283–306, July 1990.

[50] S. Tatikonda and M. I. Jordan. "loopy belief propagation and gibbs measures". In *Uncertainty in Artificial Intelligence*, Alberta, Canada, August 2002.

[51] S. Verdu and T. Weissman. "erasure entropy". Proc. 2006 IEEE Int. Symp. Information Theory, 2006.

[52] S. Verdu and T. Weissman. "the information lost in erasures". *IEEE Trans. Inf. Theory*, 54:5030, 2008.

[53] M. Wainwright. *"Stochastic processes on graphs with cycles: geometric and variational approaches"*. PhD thesis, MIT, Cambridge, MA, January 2002.

[54] M. J. Wainwright. "estimating the wrong graphical model: Benefits in the computation-limited setting". *Journal of Machine Learning Research*, 7:1829–1859, Septmber 2006.

[55] M. J. Wainwright, T. S. Jaakkola, and A. S. Willsky. "tree-based reparameterizaton framework for analysis of sum-product and related algorithms". *IEEE Trans. Info. Thy.*, 49(5):1120–1145, May 2003.

[56] M. J. Wainwright and M. I. Jordan. "graphical models, exponential families, and variational inference". Techinical Report 649, U. C. Berkeley, 2003.

[57] Y. Weiss. "belief propagation and revision in networks with loops". A.I. Memo 1616, MIT AI Lab, Cambridge, MA, November 1997.

[58] Y. Weiss. "correctness of local probability propagation in graphical models with loops". *Neural Computation*, 12:1–41, 2000.

[59] Yair Weiss. "correctness of gaussian belief propagation in graphical models of arbitrary topology". *Neural Computation*, 13(10), October 2001.

[60] T. Weissman and S. Mannor. "on universal compression of multi-dimensional data arrays using self-similar curves", 2000.

[61] S. Winkler. *"Uniqueness of Gibbs measures with application to Gibbs sampling and the Sum-Product algorithm"*. Phd, Yale, 2007.

[62] J. S. Yedidia, W. T. Freeman, and Y. Weiss. "understanding belief propagation and its generalization". Tr-2001-22, MERL, Cambridge, MA, January 2002.