DISTRIBUTED ITERATIVE DECODING AND ESTIMATION VIA EXPECTATION PROPAGATION: PERFORMANCE AND CONVERGENCE

A Dissertation

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

by

John MacLaren Walsh

May 2006

DISTRIBUTED ITERATIVE DECODING AND ESTIMATION VIA EXPECTATION PROPAGATION: PERFORMANCE AND CONVERGENCE

John MacLaren Walsh, Ph.D.

Cornell University 2006

This dissertation discusses the performance and convergence of a family of algorithms for distributed iterative approximate statistical inference called expectation propagation algorithms [1, 2]. Notable examples of expectation propagation include the turbo decoder[3, 4, 5, 6, 7], Gallager's algorithm for the soft decoding of LDPC codes[8, 9, 10], the Kalman filter [11], the forward backward algorithm[12], and belief propagation [13, 14]. These algorithms were heuristically proposed and neither their convergence behavior nor the mechanism behind their good performance is well understood. After covering special cases in which the algorithms can be shown to converge to the optimal values, we provide a novel performance framework which shows that the stationary points of the expectation propagation algorithms solve a constrained maximum likelihood optimization problem. We also show that the stationary points of expectation propagation are critical points of a constrained statistics based Bethe free energy. We then discuss duality and the relationship between the two generic optimality frameworks. Continuing on, we next study the mechanism of convergence behind expectation propagation, and discover that it may be interpreted both as a nonlinear block Gauss Seidel method [15, 16], on the gradient of the Lagrangian as well as a variant of Dykstra's algorithm [17] with iterated Bregman projections[18]. Some limited convergence results

are provided via the nonlinear block Gauss Seidel interpretation. Throughout the dissertation, we take care to apply the abstract theory to particular well known members of the expectation propagation family, most notably the belief propagation decoder and the turbo decoder. The dissertation concludes with a discussion of several avenues that the work therein has opened up for further research.

BIOGRAPHICAL SKETCH

John MacLaren Walsh was born on September 23, 1981 in Carbondale, Illinois. At the age of 8, he and his parents moved to Virginia Beach, VA, where he attended Kingston elementary school. John then attended Norfolk Academy for middle and high school, where he learned a love for German and theater. Since Fall 1999, John has been a student at Cornell University, where he earned his BS magna cum laude in May 2002. John also earned his MS (January 2004) and PhD (May 2006) at Cornell University under the supervision of C. Richard Johnson, Jr. During his graduate studies John, with the aid of Dr. Johnson and research collaborator Phillip A. Regalia (Catholic University of America and INT Evry, France), was fortunate enough to travel across the globe, presenting his research at conferences and universities in France, Italy, Portugal, and Australia. John is a member of Tau Beta Pi and Eta Kappa Nu, and he loves to play the guitar.

To my family.

ACKNOWLEDGEMENTS

I want to thank my parents, Ken and Cathy, and my sister, Elizabeth, for their constant love and support and for their patience as I spent the long hours drafting this dissertation, tucked away in a room in their house over winter break. Their love and encouragement continues to provide me with the youthful ability to rediscover myself without fear of the consequences.

I want to thank Christina for her continuing meaningful companionship and for putting up with me, especially as I spent the self-absorbed time necessary to complete the research for this dissertation.

I want to thank Cyp, Stefan, Andy, and Rick M. for their friendship and for their patience with my idiosyncracies as an officemate/housemate/friend.

This work, and indeed my graduate career, would not have been enjoyable or successful without my advisor, C. Richard Johnson, Jr. Rick provided the unique combination of appropriate amounts of friendliness, technical guidance, patience, and exciting opportunities that allowed my graduate experience to flourish in a way that I never could have even expected, let alone hoped for. He has been a profoundly positive influence on my life, even beyond my career as a researcher. I also want to thank Dr. Phillip Regalia for his help and comments on much of this research as a collaborator, and for his positive influence on my graduate education.

The extensive travel to conferences at remote parts of the globe supported by two grants from the National Science Foundation and generous gifts from Applied Signal Technology and Texas Instruments to Dr. Johnson's research group, have allowed me to form relationships with other researchers and exchange ideas which greatly influenced this dissertation and will have a profoundly positive impact on my future career in academics.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

Introduction

Methods for approximate distributed statistical inference have become of increased interest in engineering in the past decades due to increasing performance demands and the desire to accurately estimate and decode data using algorithms which require a limited amount of delay and computation complexity. Within the context of digital communications especially, the high cost of bandwidth and the increasing data rate needs of users have driven researchers to search for means of increasing spectral efficiency through better error control coding. Again within that arena, the turbo decoder[3, 4, 5, 6, 7], one such method for approximate distributed statistical inference, met with great success, bringing spectral efficiencies closer than ever before to the theoretical limits set out by Shannon[24]. The discovery of the equivalence of turbo decoding with belief propagation and the subsequence rebirth and increased interest in the soft iterative decoding of low density parity check (LDPC) codes[8, 9, 10], another method for approximate statistical inference, also provided communications engineers with decoding algorithms that were implementable in hardware that could achieve performance very near to theoretical limits. Turbo equalization [25, 26, 27] and synchronization [28, 29, 30], again algorithms for distributed approximate statistical inference, allowed receiver designers to cope with the non-idealities of real fading channels. Within the signal processing for communications and wireless communications research communities, the increased research focus on sensor networks also warranted attention to distributed, but coupled, estimation problems in which the capability for better performance via collaboration amongst the sensors on a joint estimation problem begs for a dis-

tributed computationally efficient algorithm for statistical inference which makes efficient use of communications resources. Indeed, it has become clear, certainly in the context of signal processing for communications, that systems designers need in their arsenal decoding and estimation algorithms which are distributed, require a small amount of iterative processing, and achieve performance comparable to theoretical limits.

The academic community has answered this need with the outpouring of algorithms, and generalizations of algorithms, and generalizations of generalizations of algorithms promising approximate distributed decoding and/or estimation usually justified by empirical simulation based evidence. However, the sense in which these algorithms are "approximate", that is, the sense of optimality of their outputs, is in many cases unknown. Furthermore, in the case of iterative algorithms, it is often unclear if and when the algorithms will converge. Again, some of the best examples are the turbo decoder, the turbo equalizer/synchronizer, and the belief propagation decoder. These algorithms were proposed heuristically, with their performance and convergence behavior being characterized in an "on average" manner via Monte Carlo simulation. Naturally, since their inception, countless theoreticians have attempted to prove convergence properties of these algorithms, but this too has proven to be a daunting task, partially because the sense of optimality is unknown.

In this dissertation, we will analyze the performance and convergence of a family of such iterative algorithms for approximate distributed statistical inference, called expectation propagation[1, 2]. Notable members of this family include the turbo decoder[3, 4, 5, 6, 7], Gallager's algorithm for the soft decoding of LDPC codes[8, 9, 10], the Kalman filter [11], the forward backward algorithm[12], and

belief propagation [13, 14]. Members of this family of algorithms whose performance and convergence behavior is already well understood include the Kalman filter and the forward backward algorithm for the maximum a posteriori decoding of convolutional codes. After introducing the family of algorithms in Chapter 2 and discussing various members of the family, we will begin our analysis in Section 3 by covering some special cases in which these algorithms can be proven to converge to the maximum a posteriori parameters. Then, in Chapter 4 we extend and develop some results concerning the optimality of any expectation propagation algorithm. Some of the most important ideas in the dissertation occur in Section 4.2, where we show that expectation propagation is an iterative method seeking a solution of a constrained maximum likelihood estimation problem. Chapter 5 is then devoted to convergence frameworks which may be applied to the dynamics of expectation propagation. We make some novel connections in this chapter between expectation propagation and the Gauss Seidel iteration [16, 15] and between expectation propagation and Dykstra's algorithm [17] for Bregman projections on convex sets [18]. Throughout the dissertation, we take care to apply our general theory to particular well known members of the family of expectation propagation algorithms such as the turbo decoder and belief propagation. The theoretical tools necessary for the work contained in this dissertation stretch across several disciplines, and thus we have included the fundamental results that we used (often without proof) and references to appropriate material in the appendices.

1.1 Orientation and Organization for the Non-Specialist

A great many of the ideas in this dissertation have to do with connections between bodies of theoretical work. A diagram highlighting some of the connections we

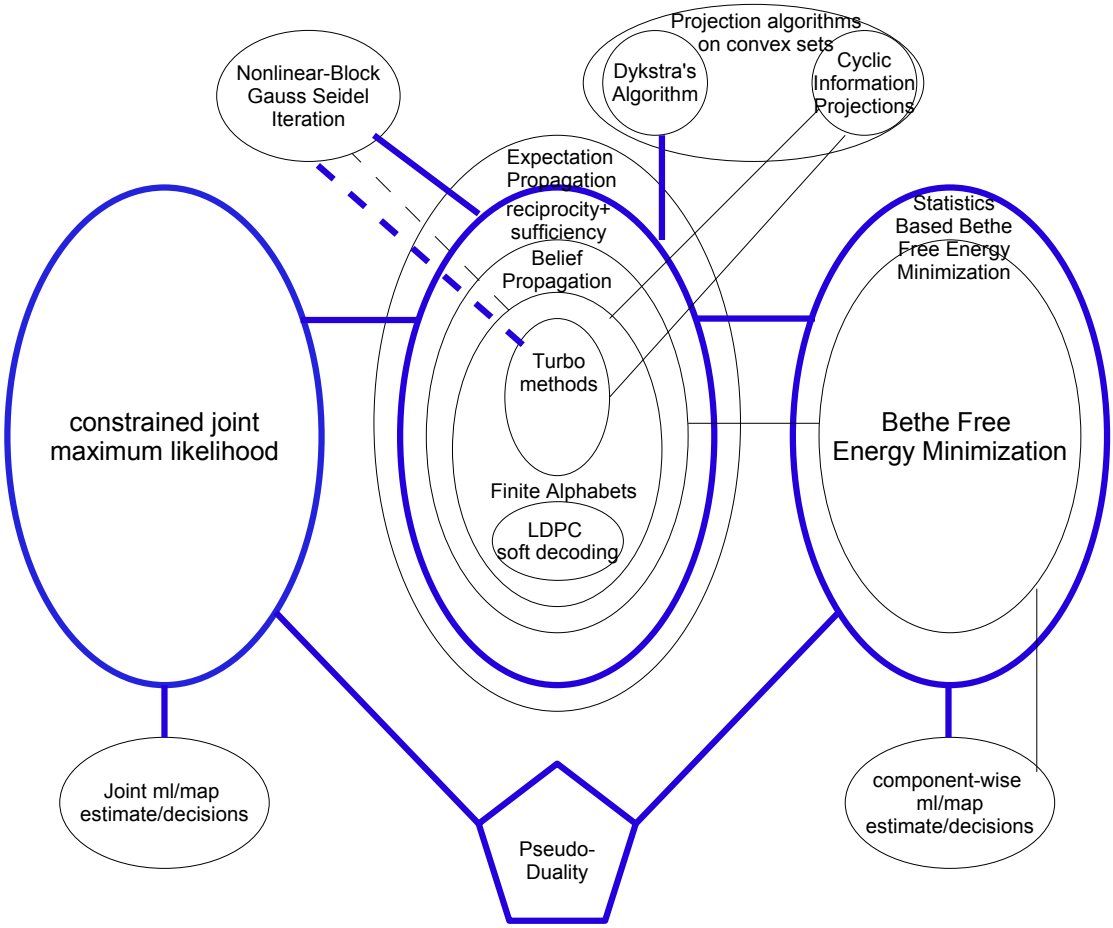

Figure 1.1: A visual organization of many of the concepts in this dissertation. Bolded lines and shapes indicate work that we innovated in developing this dissertation. A line connects two ovals if we develop theory in this dissertation relating the ideas which the two ovals represent. Different, but related bodies of theory are drawn separately and connections between these bodies are shown via lines. Containment of one oval within another oval indicates that the idea represented by the latter is a special case of the idea represented by the former.

will discuss is shown in Figure 1.1. Items which we innovated in creating this dissertation and the research leading up to it have bolded lines, while items which we cover but we did not innovate have thinner lines. The dashed lines represent work we developed at the same time as other researchers.

In Chapter 2 we will discuss a family of algorithms, expectation propagation (situated in the center of Figure 1.1), that attempt to infer some unknown parameters from some observations given a model for the observations which depends on the parameters. Within the context of the family of algorithms, we wish to determine some well known algorithms which fall into this family, and which of those algorithms are special cases of other algorithms. This gives rise to the containment within the expectation propagation oval in Figure 1.1 of several other algorithms and families of algorithms. Out of the family of expectation propagation algorithms, we will select a special subfamily which satisfies a couple of assumptions which we call reciprocity and sufficiency. These assumptions allow for further analysis, and include all interesting algorithms falling within the expectation propagation family developed to date.

Next, we will turn our attention in Chapters 3 and 4 to the performance that these algorithms can attain by studying their stationary points, which are the locations at which their values would not change if we initialized the algorithms at them. We develop two novel optimality frameworks (depicted to the right and to the left of the expectation propagation oval in Figure 1.1) which apply to expectation propagation algorithms under the sufficiency and reciprocity assumptions. One of these frameworks we totally innovate, and the other we develop by extending existing results for belief propagation to the wider class of expectation propagation algorithms with sufficiency and reciprocity.

In Chapter 5, we investigate convergence frameworks for the expectation propagation algorithms by relating them to iterative methods developed for other, more mathematically abstract problems. The Gauss Seidel method, shown in the upper left corner of Figure 1.1 is an iterative numerical method for solving a system of equations. We show in Section 5.1 that expectation propagation under the sufficiency and reciprocity assumption may be seen as particular instances of the Gauss Seidel method. This fact allows us to apply convergence theory for Gauss Seidel iterations to the expectation propagation algorithms. Dykstra's algorithm, shown in the upper right corner of Figure 1.1 is a broad family of algorithms made from projections (i.e. minimization of distance like quantities) on convex sets. We show in Section 5.2 how expectation propagation may be related to Dykstra's algorithm with Bregman projections under the reciprocity and sufficiency assumptions.

1.2 Basic Notation and Preliminaries

To aid the reader, we have used typed typographical notation in this dissertation which is described in Table 1.1. We will attempt to further aid the reader by providing a key to the new notation introduced in each chapter.

Of central interest in this dissertation will be problems of statistical inference of some random parameters $\boldsymbol{\theta} \in \Theta$ from some random observations $\mathbf{r} \in \mathcal{Y}$. The parameter space Θ will be the Cartesian product of $\mathtt{N}$ subsets of the real numbers

$$\Theta := \Theta_1 \times \cdots \times \Theta_{\mathtt{N}}, \quad \Theta_{\mathtt{i}} \subseteq \mathbb{R} \ \forall \mathtt{i} \in \{1, \ldots, \mathtt{N}\}$$

so that $\boldsymbol{\theta}$ is a vector

$$\boldsymbol{\theta} := [\theta_1, \ldots, \theta_{\mathtt{N}}], \quad \theta_{\mathtt{i}} \in \Theta_{\mathtt{i}} \ \forall \mathtt{i} \in \{1, \ldots, \mathtt{N}\}$$

Table 1.1: Typographical notation used in this dissertation.

$\mathcal{A}$	set
$\mathbb{R}$	set of real numbers
x	deterministic scalar variable
$\mathbf{x}$	deterministic vector
$\mathbf{A}$	deterministic matrix
$\mathtt{i}$	integer counting index
x	random scalar
P_x	cumulative distribution function for x
p_x	probability density function for x
$\mathbb{E}_x$	expectation with respect to the pdf p_x
$\mathbb{P}$	probability
$\mathbf{x}$	random vector
f	scalar valued function
$\mathbf{g}$	vector valued function
$\mathfrak{D}$	a functional

Similarly, the received space will be the Cartesian product of $\mathtt{L}$ subsets of the real numbers

$$\mathcal{Y} := \mathcal{Y}_1 \times \cdots \mathcal{Y}_{\mathtt{L}}, \quad \mathcal{Y}_{\mathtt{i}} \subseteq \mathbb{R} \; \forall \mathtt{i} \in \{1, \ldots, \mathtt{L}\}$$

so that $\mathbf{r}$ is a vector

$$\mathbf{r} := [r_1, \ldots, r_{\mathtt{L}}], \quad r_{\mathtt{i}} \in \mathcal{Y}_{\mathtt{i}} \; \forall \mathtt{i} \in \{1, \ldots, \mathtt{L}\}$$

Specified will be a joint probability model for $\mathbf{r}$ and $\boldsymbol{\theta}$, so that we have a probability measure which assigns probabilities $\mathbb{P}\left[(\mathbf{r}, \boldsymbol{\theta}) \in \mathcal{A}\right]$ for sets $\mathcal{A} \subseteq \mathcal{Y} \times \Theta$. We will represent this probability model in terms of a joint probability density function, $\mathsf{p}_{\mathbf{r},\boldsymbol{\theta}}(\mathbf{r}, \boldsymbol{\theta})$, which is the Radon Nikodym derivative of the probability measure $\mathbb{P}\left[(\mathbf{r}, \boldsymbol{\theta}) \in \mathcal{A}\right]$ with respect to a reference measure $\mathsf{d}\mathbf{r}\mathsf{d}\boldsymbol{\theta}$. The reference measure will be a suitable product measure of two measures $\mathsf{d}\mathbf{r}$ and $\mathsf{d}\boldsymbol{\theta}$ whose support will contain $\mathcal{Y}$ and Θ respectively. In fact, due to the Cartesian product nature of the received and parameter spaces, $\mathcal{Y} = \mathcal{Y}_1 \times \cdots \times \mathcal{Y}_{\mathtt{L}}$ and $\Theta = \Theta_1 \times \cdots \times \Theta_{\mathtt{N}}$, we will choose $\mathsf{d}\mathbf{r}$ and $\mathsf{d}\boldsymbol{\theta}$ to be product measures themselves, so that

$$\mathsf{d}\mathbf{r} = \prod_{\mathtt{i}=1}^{\mathtt{L}} \mathsf{d}r_{\mathtt{i}}$$

and

$$\mathsf{d}\boldsymbol{\theta} = \prod_{\mathtt{i}=1}^{\mathtt{N}} \mathsf{d}\theta_{\mathtt{i}}$$

Due to the fact that the component spaces are subsets of the real numbers $\mathbb{R}$ for each $\mathtt{i} \in \{1, \ldots, \mathtt{L}\}$, we will choose $\mathsf{d}r_{\mathtt{i}}$ to be Lebesgue measure if $\mathcal{Y}_{\mathtt{i}}$ is uncountable, or $\mathsf{d}r_{\mathtt{i}}$ to be a counting measure with support $\mathcal{Y}_{\mathtt{i}}$ if $\mathcal{Y}_{\mathtt{i}}$ is countable (finite or countably infinite). Similarly, for each $\mathtt{i} \in \{1, \ldots, \mathtt{N}\}$, $\mathsf{d}\theta_{\mathtt{i}}$ will be Lebesgue measure if $\Theta_{\mathtt{i}}$ is uncountable and will be a counting measure with support $\Theta_{\mathtt{i}}$ if $\Theta_{\mathtt{i}}$ is countable. We will then be able to calculate probabilities using

$$\mathbb{P}\left[(\mathbf{r}, \boldsymbol{\theta}) \in \mathcal{A}\right] = \int_{\mathcal{A}} \mathsf{p}_{\mathbf{r},\boldsymbol{\theta}}(\mathbf{r}, \boldsymbol{\theta}) \mathsf{d}\mathbf{r}\mathsf{d}\boldsymbol{\theta}$$

Chapter 2

Expectation Propagation

Expectation propagation, first proposed in [1, 2], defines a family of algorithms for approximate Bayesian statistical inference which exploit structure in the joint probability density function $\mathsf{p}_{\mathbf{r},\boldsymbol{\theta}}(\mathbf{r},\boldsymbol{\theta})$ for $\mathbf{r}$ and $\boldsymbol{\theta}$. In particular, it is assumed that the $\mathsf{p}_{\mathbf{r},\boldsymbol{\theta}}(\mathbf{r},\boldsymbol{\theta})$ factors multiplicatively

$$\mathsf{p}_{\mathbf{r},\boldsymbol{\theta}}(\mathbf{r},\boldsymbol{\theta}) \propto \prod_{\mathtt{a}=1}^{\mathtt{M}} \mathsf{f}_{\mathtt{a},\mathbf{r}}(\boldsymbol{\theta}_{\mathtt{a}}), \quad \boldsymbol{\theta}_{\mathtt{a}} \subseteq \boldsymbol{\theta} \tag{2.1}$$

where the factors $\mathsf{f}_{\mathtt{a},\mathbf{r}}$ implicitly are functions of $\mathbf{r}$ and have range $[0,\infty)$ and where $\boldsymbol{\theta}_{\mathtt{a}}$ is a vector formed by taking a subset of the elements of $\boldsymbol{\theta}$. Expectation propagation aims at iteratively approximating the joint density as the product of $\mathtt{M}$ minimal standard exponential family densities

$$\mathsf{p}_{\mathbf{r},\boldsymbol{\theta}}(\mathbf{r},\boldsymbol{\theta}) \approx \prod_{\mathtt{a}=1}^{\mathtt{M}} \mathsf{g}_{\mathtt{a},\boldsymbol{\lambda}_{\mathtt{a}}}(\boldsymbol{\theta}_{\mathtt{a}})$$

The minimal standard exponential family densities $\mathsf{g}_{\mathtt{a},\boldsymbol{\lambda}_{\mathtt{a}}}(\boldsymbol{\theta}_{\mathtt{a}})$ are the Radon Nikodym derivative of a standard exponential family measure with respect to the reference measure $\mathsf{d}\boldsymbol{\theta}_{\mathtt{a}}$ formed by the product of measures $\mathsf{d}\theta_{\mathtt{i}}$ for each $\theta_{\mathtt{i}}$ appearing in $\boldsymbol{\theta}_{\mathtt{a}}$. These standard exponential family densities may be parameterized in terms of a vector of real valued parameters $\boldsymbol{\lambda}_{\mathtt{a}}$ and sufficient statistics $\mathbf{t}_{\mathtt{a}}(\boldsymbol{\theta}_{\mathtt{a}})$ as

$$\mathsf{g}_{\mathtt{a},\boldsymbol{\lambda}_{\mathtt{a}}}(\boldsymbol{\theta}_{\mathtt{a}}) := \exp\left(\mathbf{t}_{\mathtt{a}}(\boldsymbol{\theta}_{\mathtt{a}}) \cdot \boldsymbol{\lambda}_{\mathtt{a}} - \psi_{\mathbf{t}_{\mathtt{a}}}(\boldsymbol{\lambda}_{\mathtt{a}})\right)$$

where ψ is defined as

$$\psi_{\mathbf{t}_{\mathtt{a}}}(\boldsymbol{\lambda}_{\mathtt{a}}) := \log\left(\int_{\Theta_{\mathtt{a}}} \exp(\mathbf{t}_{\mathtt{a}}(\boldsymbol{\theta}_{\mathtt{a}}) \cdot \boldsymbol{\lambda}_{\mathtt{a}}) \mathsf{d}\boldsymbol{\theta}_{\mathtt{a}}\right) \tag{2.2}$$

Table 2.1: The notation introduced in Section 2, part 1.

$\mathtt{k}$	a time index
r	a sample of an element of the vector of observations
θ	a sample value of an element in the vector of parameters
$\mathbf{r}$	a sample value of the vector of observations
$\boldsymbol{\theta}$	a sample value of the vector of parameters
r	an element of the vector of observations
θ	an element of the vector of parameters
$\mathtt{N}$	the number of parameters
$\mathtt{L}$	the number of observations
Θ	the parameter space
$\mathcal{Y}$	the observation space
$\mathbf{r}$	the observations
$\boldsymbol{\theta}$	the parameters
$\mathsf{p}_{\mathbf{r},\boldsymbol{\theta}}$	the joint pdf of the observations and parameters
$\mathtt{a}, \mathtt{c}$	an index to the factors
$\mathtt{M}$	the number of factors
$\mathtt{f}$	a factor function
$\mathtt{g}$	an approximating factor
ψ	the partition function
$\mathtt{d}$	the supporting measure
$\boldsymbol{\lambda}$	the parameters of a minimal standard exponential family
$\mathbf{t}$	the sufficient statistics of a minimal standard exponential family

Table 2.2: The notation introduced in Section 2, part 2.

$\mathbb{P}$	the probability operator
$\mathfrak{D}$	the Kullback Leibler divergence
$\mathsf{v}, \mathsf{q}, \mathsf{t}, \mathsf{u}$	some pdfs
μ	the induced (change of variables) Radon Nikodym derivative

The $\mathsf{g}_{\mathsf{a},\boldsymbol{\lambda}_\mathsf{a}}$ are iteratively refined in order to approximate $\mathsf{f}_{\mathsf{a},\boldsymbol{\lambda}_\mathsf{a}}$ by minimizing the Kullback Leibler distance

$$\mathsf{g}_{\mathsf{a},\boldsymbol{\lambda}_\mathsf{a}} = \arg\min_{\mathsf{g}_{\mathsf{a},\boldsymbol{\lambda}_\mathsf{a}}} \mathfrak{D}\left(\frac{\mathsf{f}_{\mathsf{a},\mathbf{r}}(\boldsymbol{\theta}_\mathsf{a}) \prod_{\mathsf{c}\neq\mathsf{a}} \mathsf{g}_{\mathsf{c},\boldsymbol{\lambda}_\mathsf{c}}(\boldsymbol{\theta}_\mathsf{c})}{\int_\Theta \mathsf{f}_{\mathsf{a},\mathbf{r}}(\boldsymbol{\theta}_\mathsf{a}) \prod_{\mathsf{c}\neq\mathsf{a}} \mathsf{g}_{\mathsf{c},\boldsymbol{\lambda}_\mathsf{c}}(\boldsymbol{\theta}_\mathsf{c}) \mathsf{d}\boldsymbol{\theta}} \middle\| \frac{\prod_{\mathsf{c}=1}^{\mathtt{M}} \mathsf{g}_{\mathsf{c},\boldsymbol{\lambda}_\mathsf{c}}(\boldsymbol{\theta}_\mathsf{c})}{\int_\Theta \prod_{\mathsf{c}=1}^{\mathtt{M}} \mathsf{g}_{\mathsf{c},\boldsymbol{\lambda}_\mathsf{c}}(\boldsymbol{\theta}_\mathsf{c}) \mathsf{d}\boldsymbol{\theta}} \right) \tag{2.3}$$

This minimization has a unique solution due to the log convexity of the Kullback Leibler distance in the second argument and the minimality of each of the representations of the standard exponential families $\mathsf{g}_{\mathsf{a},\boldsymbol{\lambda}_\mathsf{a}}$. The minima may be found by taking derivatives with respect to $\boldsymbol{\lambda}_\mathsf{a}$ to get

$$\nabla_{\boldsymbol{\lambda}_\mathsf{a}} \mathfrak{D} = \mathbb{E}_\mathsf{q}\left[\mathbf{t}_\mathsf{a}(\boldsymbol{\theta}_\mathsf{a})\right] - \mathbb{E}_\mathsf{v}\left[\mathbf{t}_\mathsf{a}(\boldsymbol{\theta}_\mathsf{a})\right] \tag{2.4}$$

where the probability density functions $\mathsf{v}(\boldsymbol{\theta})$ and $\mathsf{q}(\boldsymbol{\theta})$ are

$$\mathsf{v}(\boldsymbol{\theta}) := \frac{\mathsf{f}_{\mathsf{a},\mathbf{r}}(\boldsymbol{\theta}_\mathsf{a}) \prod_{\mathsf{c}\neq\mathsf{a}} \mathsf{g}_{\mathsf{c},\boldsymbol{\lambda}_\mathsf{c}}(\boldsymbol{\theta}_\mathsf{c})}{\int_\Theta \mathsf{f}_{\mathsf{a},\mathbf{r}}(\boldsymbol{\theta}_\mathsf{a}) \prod_{\mathsf{c}\neq\mathsf{a}} \mathsf{g}_{\mathsf{c},\boldsymbol{\lambda}_\mathsf{c}}(\boldsymbol{\theta}_\mathsf{c}) \mathsf{d}\boldsymbol{\theta}}$$

and

$$\mathsf{q}(\boldsymbol{\theta}) := \frac{\prod_{\mathsf{c}=1}^{\mathtt{M}} \mathsf{g}_{\mathsf{c},\boldsymbol{\lambda}_\mathsf{c}}(\boldsymbol{\theta}_\mathsf{c})}{\int_\Theta \prod_{\mathsf{c}=1}^{\mathtt{M}} \mathsf{g}_{\mathsf{c},\boldsymbol{\lambda}_\mathsf{c}}(\boldsymbol{\theta}_\mathsf{c}) \mathsf{d}\boldsymbol{\theta}}$$

Next, another $\mathsf{g}_{\mathsf{a},\boldsymbol{\lambda}_\mathsf{a}}$ is selected to refine, usually according to some iteration order, and the algorithm continues until it converges.

The family of algorithms may then be summarized by the following

- initialize $\boldsymbol{\lambda}_{\mathtt{a}}$ for all $\mathtt{a} \in \{1, \ldots, \mathtt{M}\}$
- repeat 1 and 2 until convergence
  1. select a $\boldsymbol{\lambda}_{\mathtt{a}}$ to refine for some $\mathtt{a} \in \{1, \ldots, \mathtt{M}\}$
  2. update $\boldsymbol{\lambda}_{\mathtt{a}}$ according to

$$\boldsymbol{\lambda}_{\mathtt{a}} \leftarrow \arg\min_{\boldsymbol{\lambda}_{\mathtt{a}}} \mathfrak{D}\left(\frac{\mathsf{f}_{\mathtt{a},\mathbf{r}}(\boldsymbol{\theta}_{\mathtt{a}}) \prod_{\mathtt{c} \neq \mathtt{a}} \mathsf{g}_{\mathtt{c},\boldsymbol{\lambda}_{\mathtt{c}}}(\boldsymbol{\theta}_{\mathtt{c}})}{\int_{\Theta} \mathsf{f}_{\mathtt{a},\mathbf{r}}(\boldsymbol{\theta}_{\mathtt{a}}) \prod_{\mathtt{c} \neq \mathtt{a}} \mathsf{g}_{\mathtt{c},\boldsymbol{\lambda}_{\mathtt{c}}}(\boldsymbol{\theta}_{\mathtt{c}}) \mathsf{d}\boldsymbol{\theta}} \middle\| \frac{\prod_{\mathtt{c}=1}^{\mathtt{M}} \mathsf{g}_{\mathtt{c},\boldsymbol{\lambda}_{\mathtt{c}}}(\boldsymbol{\theta}_{\mathtt{c}})}{\int_{\Theta} \prod_{\mathtt{c}=1}^{\mathtt{M}} \mathsf{g}_{\mathtt{c},\boldsymbol{\lambda}_{\mathtt{c}}}(\boldsymbol{\theta}_{\mathtt{c}}) \mathsf{d}\boldsymbol{\theta}} \right)$$

The order according to which $\boldsymbol{\lambda}_{\mathtt{a}}$ is selected in step 1, which we call scheduling, varies in implementations. Several possibilities include

- **parallel scheduling** In this case, we update all of the parameters in parallel, so that if we denote the value $\boldsymbol{\lambda}_{\mathtt{a}}^{\mathtt{k}}$ to be the value of $\boldsymbol{\lambda}_{\mathtt{a}}$ at time instant $\mathtt{k} \in \{0, 1, \ldots, \infty\}$ we have

$$\boldsymbol{\lambda}_{\mathtt{a}}^{\mathtt{k}+1} = \arg\min_{\boldsymbol{\lambda}_{\mathtt{a}}} \mathfrak{D}\left(\frac{\mathsf{f}_{\mathtt{a},\mathbf{r}}(\boldsymbol{\theta}_{\mathtt{a}}) \prod_{\mathtt{c} \neq \mathtt{a}} \mathsf{g}_{\mathtt{c},\boldsymbol{\lambda}_{\mathtt{c}}^{\mathtt{k}}}(\boldsymbol{\theta}_{\mathtt{c}})}{\int_{\Theta} \mathsf{f}_{\mathtt{a},\mathbf{r}}(\boldsymbol{\theta}_{\mathtt{a}}) \prod_{\mathtt{c} \neq \mathtt{a}} \mathsf{g}_{\mathtt{c},\boldsymbol{\lambda}_{\mathtt{c}}^{\mathtt{k}}}(\boldsymbol{\theta}_{\mathtt{c}}) \mathsf{d}\boldsymbol{\theta}} \middle\| \frac{\mathsf{g}_{\mathtt{a},\boldsymbol{\lambda}_{\mathtt{a}}}(\boldsymbol{\theta}_{\mathtt{a}}) \prod_{\mathtt{c} \neq \mathtt{a}} \mathsf{g}_{\mathtt{c},\boldsymbol{\lambda}_{\mathtt{c}}^{\mathtt{k}}}(\boldsymbol{\theta}_{\mathtt{c}})}{\int_{\Theta} \mathsf{g}_{\mathtt{a},\boldsymbol{\lambda}_{\mathtt{a}}}(\boldsymbol{\theta}_{\mathtt{a}}) \prod_{\mathtt{c} \neq \mathtt{a}} \mathsf{g}_{\mathtt{c},\boldsymbol{\lambda}_{\mathtt{c}}^{\mathtt{k}}}(\boldsymbol{\theta}_{\mathtt{c}}) \mathsf{d}\boldsymbol{\theta}} \right)$$

  for all $\mathtt{a} \in \{1, \ldots, \mathtt{M}\}$.

- **serial scheduling** In this case, we update each of the parameters in serial, so that if we denote the value $\boldsymbol{\lambda}_{\mathtt{a}}^{\mathtt{k}}$ to be the value of $\boldsymbol{\lambda}_{\mathtt{a}}$ at iteration $\mathtt{k} \in \{0, 1, \ldots, \infty\}$ we have

$$\boldsymbol{\lambda}_{\mathtt{a}}^{\mathtt{k}+1} = \arg\min_{\boldsymbol{\lambda}_{\mathtt{a}}} \mathfrak{D}\left(\mathsf{v} \| \mathsf{q}\right)$$

with

$$v(\boldsymbol{\theta}) := \frac{f_{a,\mathbf{r}}(\boldsymbol{\theta}_a)\prod_{c<a} g_{c,\boldsymbol{\lambda}_c^{k+1}}(\boldsymbol{\theta}_c)\prod_{c>a} g_{c,\boldsymbol{\lambda}_c^{k}}(\boldsymbol{\theta}_c)}{\int_\Theta f_{a,\mathbf{r}}(\boldsymbol{\theta}_a)\prod_{c<a} g_{c,\boldsymbol{\lambda}_c^{k+1}}(\boldsymbol{\theta}_c)\prod_{c>a} g_{c,\boldsymbol{\lambda}_c^{k}}(\boldsymbol{\theta}_c)d\boldsymbol{\theta}}$$

and

$$q(\boldsymbol{\theta}) := \frac{g_{a,\boldsymbol{\lambda}_a}(\boldsymbol{\theta}_a)\prod_{c<a} g_{c,\boldsymbol{\lambda}_c^{k+1}}(\boldsymbol{\theta}_c)\prod_{c>a} g_{c,\boldsymbol{\lambda}_c^{k}}(\boldsymbol{\theta}_c)}{\int_\Theta g_{a,\boldsymbol{\lambda}_a}(\boldsymbol{\theta}_a)\prod_{c<a} g_{c,\boldsymbol{\lambda}_c^{k+1}}(\boldsymbol{\theta}_c)\prod_{c>a} g_{c,\boldsymbol{\lambda}_c^{k}}(\boldsymbol{\theta}_c)d\boldsymbol{\theta}}$$

for all $a \in \{1,\ldots,M\}$.

- **random scheduling** Here, we decide which $\boldsymbol{\lambda}_a$ to update by drawing a uniformly from the set $\{1,\ldots,M\}$.

We will be primarily interested in studying the performance and convergence under the parallel and serial schedules in the remainder of the thesis.

2.1 Examples of Expectation Propagation

To provide motivation, and to allow for continued pragmatic application of the ideas throughout the remainder of the dissertation, we discuss in the next few examples some well known members of the expectation propagation family of algorithms, as well as introduce some new ones.

2.1.1 Parallel Turbo Decoding

A parallel turbo encoder is depicted in Figure 2.1. A block of message bits

$$\boldsymbol{\xi} = [\xi_1,\ldots,\xi_N] \in \{0,1\}^N$$

is encoded by a rate R_0 recursive systematic convolutional code to get a binary sequence $\boldsymbol{\zeta}(\boldsymbol{\xi}) \in \{0,1\}^{\frac{N}{R_0}}$. Then $\boldsymbol{\xi}$ is interleaved by permuting the order of the

Table 2.3: Notation used in connection with the turbo decoder, part 1.

$\boldsymbol{\xi}, \mathbf{w}, \mathbf{z}$	random binary vectors
ξ	a bit in a random message
$\boldsymbol{\xi}$	a random binary message
$\mathtt{R}_0$	the rate of the first component code
$\mathtt{R}_1$	the rate of the second component code
$\mathtt{N}$	the turbo block length
$\mathbf{r}_s$	sample rec'vd inf. from systematic bits
$\mathbf{r}_0$	sample rec'vd inf. from parity check bits of 1 component code
$\mathbf{r}_1$	sample rec'vd inf. from parity check bits of 2 component code
$\mathbf{r}_s$	rec'vd inf. from systematic bits
$\mathbf{r}_0$	rec'vd inf. from parity check bits of 1 component code
$\mathbf{r}_1$	sample rec'vd inf. from parity check bits of 2 component code
$\boldsymbol{\pi}$	the interleaver
ζ	a bit of the output of the first encoder
χ	a bit of the output of the second encoder
$\boldsymbol{\zeta}$	output of the first encoder
$\boldsymbol{\chi}$	output of the second encoder
$\mathbf{c}_0$	function producing parity check bits of first code
$\mathbf{c}_1$	function producing parity check bits of second code
λ	a log likelihood ratio
$\boldsymbol{\rho}_0$	log coordinates of the first decoder's likelihood function
$\boldsymbol{\rho}_1$	log coordinates of the second decoder's likelihood function

Table 2.4: Notation used in connection with the turbo decoder, part 2.

$\boldsymbol{\alpha}$	extrinsic information from first decoder
$\boldsymbol{\beta}$	extrinsic information from second decoder
ρ	log coordinate of likelihood function
$\boldsymbol{\rho}$	log coordinates of likelihood function
$\boldsymbol{\lambda}$	vector of log likelihood ratios
$\mathcal{C}$	codebook
1	indicator function
$\boldsymbol{\theta}$	log coordinates of a pmf
$\mathbf{B}$	matrix of all binary words
$\mathcal{F}$	set of all pmfs on binary words of appropriate length
$\mathcal{M}$	set of pmfs which factor into product of bitwise marginals
$\boldsymbol{\eta}$	expectation coordinates for a pmf
sign	function returning the sign of a real number
$\mathbf{J}$	the Fisher information matrix
$\mathbf{0}$	a matrix of all zeros
p	likelihood func. of 1 component decoder weighted by priors
q	likelihood func. of 2 component decoder weighted by priors
r	posterior density formed from the sum of the extr. inf.s
$\mathbf{p}_{\boldsymbol{\theta}}$	bitwise probabilities associated log-pmf $\boldsymbol{\theta}$
$\mathbf{P}_{\boldsymbol{\theta}}$	2 bit joint probabilities associated log-pmf $\boldsymbol{\theta}$
Diag	creates diagonal matrix from vector
$\mathbf{1}$	vector of all ones
$\mathbf{I}$	the identity matrix

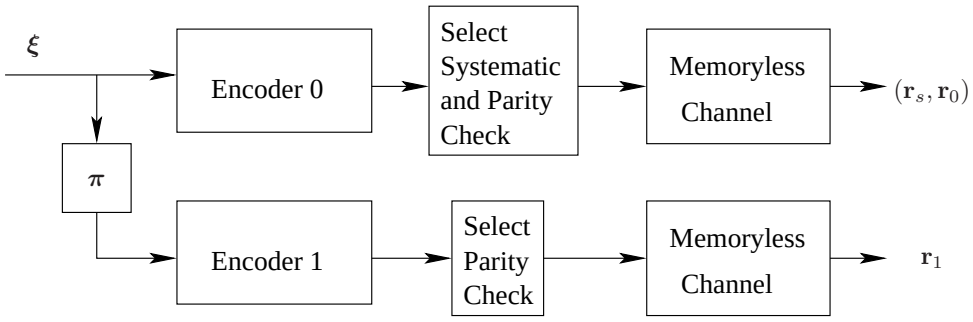

Figure 2.1: Parallel Concatenation of two convolutional codes with interleavers.

bits with some permutation $\boldsymbol{\pi}$ and encoded by a second (possibly different) rate $\mathtt{R}_1$ recursive systematic convolutional code to get a binary sequence $\boldsymbol{\chi}(\boldsymbol{\pi}(\boldsymbol{\xi})) \in \{0,1\}^{\frac{\mathtt{N}}{\mathtt{R}_1}}$. Systematic codes have the property that the message bits $\boldsymbol{\xi}$ must appear at fixed locations in the encoded messages $\boldsymbol{\zeta}$ and $\boldsymbol{\chi}$, and the remaining (non-systematic) bits in $\boldsymbol{\zeta}$ and $\boldsymbol{\chi}$ are called *parity check* bits. Some subset of bits among the systematic bits $\boldsymbol{\xi}$, the parity check bits $\mathbf{c}_0(\boldsymbol{\xi})$ from the first encoder, and the parity check bits $\mathbf{c}_1(\boldsymbol{\pi}(\boldsymbol{\xi}))$ are then selected for transmission via a *puncturing pattern*, and then transmitted over a memoryless channel to yield the observations $\mathbf{r}_s$, $\mathbf{r}_0$, and $\mathbf{r}_1$ at the receiver, respectively.

The optimal bit by bit decoder, in terms of minimizing the probability of bit error for each bit[31], would base its decisions on the a posteriori probability ratios

$$\frac{\mathbb{P}[\xi_{\mathtt{i}} = 1 | \mathbf{r}_s, \mathbf{r}_0, \mathbf{r}_1]}{\mathbb{P}[\xi_{\mathtt{i}} = 0 | \mathbf{r}_s, \mathbf{r}_0, \mathbf{r}_1]} = \frac{\sum_{\boldsymbol{\xi} \in \{0,1\}^{\mathtt{N}} | \xi_{\mathtt{i}} = 1} \mathtt{p}_{\mathbf{r}_s, \mathbf{r}_0, \mathbf{r}_1 | \boldsymbol{\xi}}(\mathbf{r}_s, \mathbf{r}_0, \mathbf{r}_1 | \boldsymbol{\xi}) \prod_{\mathtt{j}=1}^{\mathtt{M}} \mathbb{P}[\xi_{\mathtt{j}}]}{\sum_{\boldsymbol{\xi} \in \{0,1\}^{\mathtt{N}} | \xi_{\mathtt{i}} = 0} \mathtt{p}_{\mathbf{r}_s, \mathbf{r}_0, \mathbf{r}_1 | \boldsymbol{\xi}}(\mathbf{r}_s, \mathbf{r}_0, \mathbf{r}_1 | \boldsymbol{\xi}) \prod_{\mathtt{j}=1}^{\mathtt{M}} \mathbb{P}[\xi_{\mathtt{j}}]} \tag{2.5}$$

where it has been assumed that the prior density $\mathbb{P}[\boldsymbol{\xi}]$ for $\boldsymbol{\xi}$ has the bits $\xi_{\mathtt{i}}$ inde-

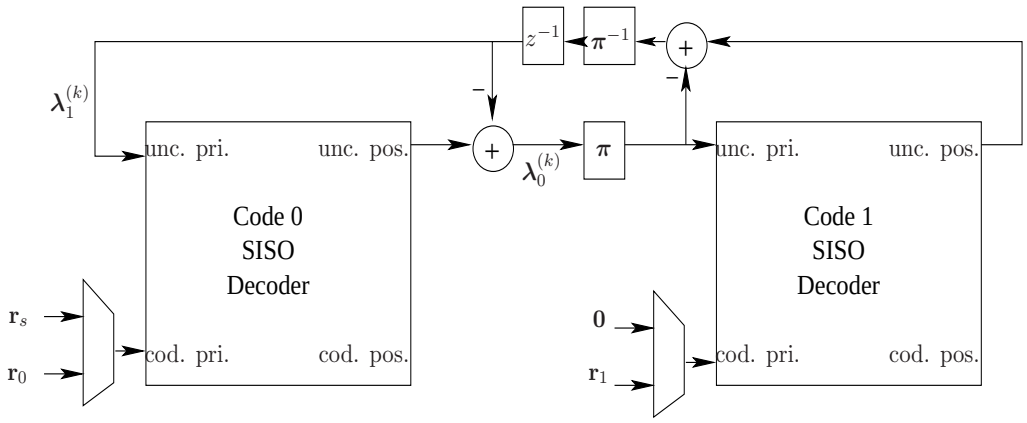

Figure 2.2: The Parallel Concatenated Turbo Decoder.

pendent, so that

$$\mathbb{P}[\boldsymbol{\xi}] = \prod_{\mathtt{j}=1}^{\mathtt{M}} \mathbb{P}[\xi_{\mathtt{j}}]$$

Unfortunately, due to the random interleaver $\boldsymbol{\pi}$, the complexity of the decoder (2.5) grows exponentially in the block length $\mathtt{N}$, and thus is not computationally feasible[32]. Given only $\mathbf{r}_s$ and $\mathbf{r}_0$, however, the forward backward algorithm [12] can calculate bitwise a posteriori probabilities for $\boldsymbol{\xi}$ with computational complexity growing only linearly in the block length $\mathtt{N}$. Furthermore, given only $\mathbf{r}_1$, the forward backward algorithm can calculate the bitwise a posteriori probabilities for $\boldsymbol{\xi}$ with computational complexity growing only linearly in the block length. The turbo decoder, depicted in Figure 2.2 and introduced in [3], then makes use of the computational efficiency of these two component decoders to iteratively approximate (2.5).

In order to describe the operation of the turbo decoder, we must first split the bitwise a posteriori probability ratios for the two component decoders into two

parts

$$\frac{\mathbb{P}[\xi_{\mathtt{i}}=1|\mathbf{r}_s,\mathbf{r}_0]}{\mathbb{P}[\xi_{\mathtt{i}}=0|\mathbf{r}_s,\mathbf{r}_0]}=\left(\frac{\mathbb{P}[\xi_{\mathtt{i}}=1]}{\mathbb{P}[\xi_{\mathtt{i}}=0]}\right)\underbrace{\left(\frac{\sum_{\boldsymbol{\xi}\in\{0,1\}^{\mathtt{N}}|\xi_{\mathtt{i}}=1}\mathsf{p}_{\mathbf{r}_s,\mathbf{r}_0|\boldsymbol{\xi}}(\mathbf{r}_s,\mathbf{r}_0|\boldsymbol{\xi})\prod_{\mathtt{j}=1,\mathtt{j}\neq\mathtt{i}}^{\mathtt{M}}\mathbb{P}[\xi_{\mathtt{j}}]}{\sum_{\boldsymbol{\xi}\in\{0,1\}^{\mathtt{N}}|\xi_{\mathtt{i}}=0}\mathsf{p}_{\mathbf{r}_s,\mathbf{r}_0|\boldsymbol{\xi}}(\mathbf{r}_s,\mathbf{r}_0,\mathbf{r}_1|\boldsymbol{\xi})\prod_{\mathtt{j}=1,\mathtt{j}\neq\mathtt{i}}^{\mathtt{M}}\mathbb{P}[\xi_{\mathtt{j}}]}\right)}_{:=\mathsf{t}_{\mathtt{i}}(\xi_{\mathtt{i}}=1)/\mathsf{t}_{\mathtt{i}}(\xi_{\mathtt{i}}=0)}\tag{2.6}$$

The term on the right is labelled the *extrinsic information ratio* and determines a probability mass function $\mathsf{t}(\boldsymbol{\xi}):=\prod_{\mathtt{j}=1}^{\mathtt{N}}\mathsf{t}_{\mathtt{j}}(\xi_{\mathtt{j}})$ such that $\mathsf{t}_{\mathtt{j}}(1)+\mathsf{t}_{\mathtt{j}}(0)=1$. The idea behind turbo decoding is to use this extrinsic information as a pseudo prior in the second component decoder, by replacing $\mathbb{P}[\boldsymbol{\xi}]$ in

$$\frac{\mathbb{P}[\xi_{\mathtt{i}}=1|\mathbf{r}_1]}{\mathbb{P}[\xi_{\mathtt{i}}=0|\mathbf{r}_1]}=\left(\frac{\mathbb{P}[\xi_{\mathtt{i}}=1]}{\mathbb{P}[\xi_{\mathtt{i}}=0]}\right)\underbrace{\left(\frac{\sum_{\boldsymbol{\xi}\in\{0,1\}^{\mathtt{N}}|\xi_{\mathtt{i}}=1}\mathsf{p}_{\mathbf{r}_1|\boldsymbol{\xi}}(\mathbf{r}_1|\boldsymbol{\xi})\prod_{\mathtt{j}=1,\mathtt{j}\neq\mathtt{i}}^{\mathtt{M}}\mathbb{P}[\xi_{\mathtt{j}}]}{\sum_{\boldsymbol{\xi}\in\{0,1\}^{\mathtt{N}}|\xi_{\mathtt{i}}=0}\mathsf{p}_{\mathbf{r}_1|\boldsymbol{\xi}}(\mathbf{r}_1|\boldsymbol{\xi})\prod_{\mathtt{j}=1,\mathtt{j}\neq\mathtt{i}}^{\mathtt{M}}\mathbb{P}[\xi_{\mathtt{j}}]}\right)}_{:=\mathsf{u}_{\mathtt{i}}(\xi_{\mathtt{i}}=1)/\mathsf{u}_{\mathtt{i}}(\xi_{\mathtt{i}}=0)}\tag{2.7}$$

with $\mathsf{t}(\boldsymbol{\xi})$ from (2.6) to get $\mathsf{u}(\boldsymbol{\xi}):=\prod_{\mathtt{j}=1}^{\mathtt{N}}\mathsf{u}_{\mathtt{j}}(\xi_{\mathtt{j}})$. Then $\mathbb{P}[\boldsymbol{\xi}]$ in (2.6) is replaced by $\mathsf{u}(\boldsymbol{\xi})$, which then defines a new $\mathsf{t}(\boldsymbol{\xi})$ which is used to get a new $\mathsf{u}(\boldsymbol{\xi})$, and so on as the process iterates. Succinctly, this may be written as

$$\frac{\mathsf{t}_{\mathtt{i}}(\xi_{\mathtt{i}}=1)}{\mathsf{t}_{\mathtt{i}}(\xi_{\mathtt{i}}=0)}\leftarrow\frac{\sum_{\boldsymbol{\xi}\in\{0,1\}^{\mathtt{N}}|\xi_{\mathtt{i}}=1}\mathsf{p}_{\mathbf{r}_s,\mathbf{r}_0|\boldsymbol{\xi}}(\mathbf{r}_s,\mathbf{r}_0|\boldsymbol{\xi})\prod_{\mathtt{j}=1,\mathtt{j}\neq\mathtt{i}}^{\mathtt{M}}\mathsf{u}_{\mathtt{j}}(\xi_{\mathtt{j}})}{\sum_{\boldsymbol{\xi}\in\{0,1\}^{\mathtt{N}}|\xi_{\mathtt{i}}=0}\mathsf{p}_{\mathbf{r}_s,\mathbf{r}_0|\boldsymbol{\xi}}(\mathbf{r}_s,\mathbf{r}_0,\mathbf{r}_1|\boldsymbol{\xi})\prod_{\mathtt{j}=1,\mathtt{j}\neq\mathtt{i}}^{\mathtt{M}}\mathsf{u}_{\mathtt{j}}(\xi_{\mathtt{j}})}\tag{2.8}$$

$$\frac{\mathsf{u}_{\mathtt{i}}(\xi_{\mathtt{i}}=1)}{\mathsf{u}_{\mathtt{i}}(\xi_{\mathtt{i}}=0)}\leftarrow\frac{\sum_{\boldsymbol{\xi}\in\{0,1\}^{\mathtt{N}}|\xi_{\mathtt{i}}=1}\mathsf{p}_{\mathbf{r}_1|\boldsymbol{\xi}}(\mathbf{r}_1|\boldsymbol{\xi})\prod_{\mathtt{j}=1,\mathtt{j}\neq\mathtt{i}}^{\mathtt{M}}\mathsf{t}_{\mathtt{j}}(\xi_{\mathtt{j}})}{\sum_{\boldsymbol{\xi}\in\{0,1\}^{\mathtt{N}}|\xi_{\mathtt{i}}=0}\mathsf{p}_{\mathbf{r}_1|\boldsymbol{\xi}}(\mathbf{r}_1|\boldsymbol{\xi})\prod_{\mathtt{j}=1,\mathtt{j}\neq\mathtt{i}}^{\mathtt{M}}\mathsf{t}_{\mathtt{j}}(\xi_{\mathtt{j}})}\tag{2.9}$$

The authors of [3, 4] then noted that this iteration was capable of delivering performance very near the theoretical limits on the performance of optimal decoding for simple component codes and reasonably large block lengths.

The publication most often credited for realizing that turbo decoding was an instance of a more abstract belief propagation algorithm (which, in turn in an instance of our expectation propagation algorithm) is [33].

To recognize turbo decoding as an instance of expectation propagation first identify the parameters $\boldsymbol{\theta}:=\boldsymbol{\xi}$, the parameter space $\Theta=\{0,1\}^{\mathtt{N}}$, and the observations $\mathbf{r}:=[\mathbf{r}_s,\mathbf{r}_0,\mathbf{r}_1]$. Owing to the memoryless nature of the channel, we have

a factorization of the likelihood function

$$\mathsf{p}_{\mathbf{r}|\boldsymbol{\theta}}(\mathbf{r}|\boldsymbol{\theta}) = \underbrace{\mathsf{p}_{\mathbf{r}_s,\mathbf{r}_0|\boldsymbol{\theta}}(\mathbf{r}_s,\mathbf{r}_0|\boldsymbol{\theta})}_{\mathsf{f}_0(\boldsymbol{\theta})}\underbrace{\mathsf{p}_{\mathbf{r}_1|\boldsymbol{\theta}}(\mathbf{r}_1|\boldsymbol{\theta})}_{\mathsf{f}_1(\boldsymbol{\theta})}$$

The next step is to recognize that the family of wordwise probability mass functions on $\boldsymbol{\xi}$ which factor into the product of their marginals are a minimal standard exponential family with sufficient statistics $\mathbf{t}(\boldsymbol{\theta}) = \mathbf{t}(\boldsymbol{\xi}) := \boldsymbol{\xi}$

$$\mathbb{P}[\boldsymbol{\xi}] := \prod_{\mathtt{i}}^{\mathtt{N}} \mathbb{P}[\xi_{\mathtt{i}}] = \exp(\boldsymbol{\xi}\cdot\boldsymbol{\lambda} - \psi_{\mathbf{t}}(\boldsymbol{\lambda}))$$

where the parameters

$$\boldsymbol{\lambda} := [\lambda_1,\ldots,\lambda_{\mathtt{N}}], \lambda_{\mathtt{i}} := \log\left(\frac{\mathbb{P}[\xi_{\mathtt{i}}=1]}{\mathbb{P}[\xi_{\mathtt{i}}=0]}\right)$$

are the vector of log bitwise probability ratios.

The approximating exponential densities are then

$$\mathsf{g}_0(\boldsymbol{\theta}) := \exp(\boldsymbol{\theta}\cdot\boldsymbol{\lambda}_0 - \psi_{\mathbf{t}}(\boldsymbol{\lambda}_0))$$

and

$$\mathsf{g}_1(\boldsymbol{\theta}) := \exp(\boldsymbol{\theta}\cdot\boldsymbol{\lambda}_1 - \psi_{\mathbf{t}}(\boldsymbol{\lambda}_1))$$

In this case the refinement (2.4) from expectation propagation takes the form

$$\begin{aligned}&\frac{\int_{\boldsymbol{\theta}\in\Theta}\boldsymbol{\theta}\mathsf{p}_{\mathbf{r}_s,\mathbf{r}_0|\boldsymbol{\theta}}(\mathbf{r}_s,\mathbf{r}_0|\boldsymbol{\theta})\exp(\boldsymbol{\theta}\cdot\boldsymbol{\lambda}_1-\psi_{\mathbf{t}}(\boldsymbol{\lambda}_1))\mathsf{d}\boldsymbol{\theta}}{\int_{\boldsymbol{\theta}\in\Theta}\mathsf{p}_{\mathbf{r}_s,\mathbf{r}_0|\boldsymbol{\theta}}(\mathbf{r}_s,\mathbf{r}_0|\boldsymbol{\theta})\exp(\boldsymbol{\theta}\cdot\boldsymbol{\lambda}_1-\psi_{\mathbf{t}}(\boldsymbol{\lambda}_1))\mathsf{d}\boldsymbol{\theta}}\\ &= \frac{\int_{\boldsymbol{\theta}\in\Theta}\boldsymbol{\theta}\exp(\boldsymbol{\theta}\cdot(\boldsymbol{\lambda}_0+\boldsymbol{\lambda}_1)-\psi(\boldsymbol{\lambda}_0)-\psi(\boldsymbol{\lambda}_1))\mathsf{d}\boldsymbol{\theta}}{\int_{\boldsymbol{\theta}\in\Theta}\exp(\boldsymbol{\theta}\cdot(\boldsymbol{\lambda}_0+\boldsymbol{\lambda}_1)-\psi(\boldsymbol{\lambda}_0)-\psi(\boldsymbol{\lambda}_1))\mathsf{d}\boldsymbol{\theta}}\end{aligned} \tag{2.10}$$

The ith element of these vectors may be recognized as the probability, according to the appropriate measure, that the ith bit of $\boldsymbol{\xi}$ is equal to one. Forming the probability ratio from this bitwise probability, and identifying

$$\left[\frac{\mathsf{t}_{\mathsf{j}}(\xi_{\mathsf{j}}=1)}{\mathsf{t}_{\mathsf{j}}(\xi_{\mathsf{j}}=0)}\right] := \exp(\boldsymbol{\lambda}_0)$$

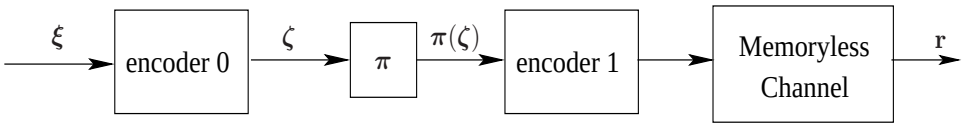

Figure 2.3: The serially concatenated turbo encoder.

and

$$\left[\frac{\mathtt{u}_\mathtt{j}(\xi_\mathtt{j}=1)}{\mathtt{u}_\mathtt{j}(\xi_\mathtt{j}=0)}\right] := \exp(\boldsymbol{\lambda}_1)$$

we get an equivalent condition to (2.10)

$$\frac{\sum_{\boldsymbol{\xi}|\xi_\mathtt{i}=1} \mathtt{p}_{\mathbf{r}_s,\mathbf{r}_0|\boldsymbol{\xi}}(\mathbf{r}_s,\mathbf{r}_0|\boldsymbol{\xi}) \prod_{\mathtt{j}=1}^{\mathtt{N}} \mathtt{u}_\mathtt{j}(\xi_\mathtt{j})}{\sum_{\boldsymbol{\xi}|\xi_\mathtt{i}=0} \mathtt{p}_{\mathbf{r}_s,\mathbf{r}_0|\boldsymbol{\xi}}(\mathbf{r}_s,\mathbf{r}_0|\boldsymbol{\xi}) \prod_{\mathtt{j}=1}^{\mathtt{N}} \mathtt{u}_\mathtt{j}(\xi_\mathtt{j})} = \frac{\mathtt{t}_\mathtt{i}(\xi_\mathtt{i}=1)\mathtt{u}_\mathtt{i}(\xi_\mathtt{i}=1)}{\mathtt{t}_\mathtt{i}(\xi_\mathtt{i}=0)\mathtt{u}_\mathtt{i}(\xi_\mathtt{i}=0)} \quad \forall \mathtt{i} \in \{1,\ldots,\mathtt{N}\}$$

From which we see that performing the turbo step (2.8) is equivalent to refining $\boldsymbol{\lambda}_0$ according to expectation propagation. A nearly identical progression of steps then shows that performing the turbo step (2.9) is equivalent to refining $\boldsymbol{\lambda}_1$ according to expectation propagation.

2.1.2 Serial Turbo Decoding

A serial turbo encoder[5, 34, 6, 7] is depicted in (2.3). A block of message bits

$$\boldsymbol{\xi} := [\xi_1,\ldots,\xi_\mathtt{N}] \in \{0,1\}^\mathtt{N}$$

is encoded with a convolutional encoder to get $\boldsymbol{\zeta}(\boldsymbol{\xi}) \in \{0,1\}^{\frac{\mathtt{N}}{\mathtt{R}_0}}$, which is then interleaved and encoded with another convolutional encoder to get $\boldsymbol{\chi}(\boldsymbol{\pi}(\boldsymbol{\zeta}(\boldsymbol{\xi})))$. The output of this second encoder is then modulated and passed over a memoryless channel to get $\mathbf{r}$.

The optimal bit by bit decoder in the sense of minimizing the probability of selecting the wrong bit would base its decisions on the a posteriori bitwise

probability ratios

$$\frac{\mathbb{P}[\xi_i = 1|\mathbf{r}]}{\mathbb{P}[\xi_i = 0|\mathbf{r}]} = \frac{\sum_{\boldsymbol{\xi}|\xi_i=1} \mathsf{p}_{\mathbf{r}|\boldsymbol{\chi}}(\mathbf{r}|\boldsymbol{\chi}(\boldsymbol{\pi}(\boldsymbol{\zeta}(\boldsymbol{\xi}))))\mathbb{P}[\boldsymbol{\xi}]}{\sum_{\boldsymbol{\xi}|\xi_i=0} \mathsf{p}_{\mathbf{r}|\boldsymbol{\chi}}(\mathbf{r}|\boldsymbol{\chi}(\boldsymbol{\pi}(\boldsymbol{\zeta}(\boldsymbol{\xi}))))\mathbb{P}[\boldsymbol{\xi}]} \tag{2.11}$$

Once again, due to the interleaver, the complexity of a decoder calculating (2.11) grows exponentially in the block length. On the other hand, due to the convolutional structure of the component codes, soft decoding only one of the component codes has complexity which is only linear in the block length. The serial turbo decoder aims to exploit this by inferring $\boldsymbol{\zeta}$ by exchanging information between the component decoders until convergence, and then inferring $\boldsymbol{\xi}$ from $\boldsymbol{\zeta}$. The update equations take a similar form to that for the parallel turbo decoder

$$\frac{\mathsf{t}_i(\zeta_i = 1)}{\mathsf{t}_i(\zeta_i = 0)} \leftarrow \frac{\sum_{\boldsymbol{\zeta}|\zeta_i=1} 1_{\mathcal{C}}(\boldsymbol{\zeta}) \prod_{j=1,j\neq i}^{N} \mathsf{u}_j(\zeta_j)}{\sum_{\boldsymbol{\zeta}|\zeta_i=0} 1_{\mathcal{C}}(\boldsymbol{\zeta}) \prod_{j=1,j\neq i}^{N} \mathsf{u}_j(\zeta_j)} \tag{2.12}$$

where $1_{\mathcal{C}}(\cdot)$ is the indicator function for the outer codebook, and,

$$\frac{\mathsf{u}_i(\zeta_i = 1)}{\mathsf{u}_i(\zeta_i = 0)} \leftarrow \frac{\sum_{\boldsymbol{\zeta}|\zeta_i=1} \mathsf{p}_{\mathbf{r}|\boldsymbol{\chi}}(\mathbf{r}|\boldsymbol{\chi}(\boldsymbol{\pi}(\boldsymbol{\zeta}))) \prod_{j=1,j\neq i}^{N} \mathsf{t}_j(\zeta_j)}{\sum_{\boldsymbol{\zeta}|\zeta_i=0} \mathsf{p}_{\mathbf{r}|\boldsymbol{\chi}}(\mathbf{r}|\boldsymbol{\chi}(\boldsymbol{\pi}(\boldsymbol{\zeta}))) \prod_{j=1,j\neq i}^{N} \mathsf{t}_j(\zeta_j)} \tag{2.13}$$

We can now see that the serial turbo decoder is member of the family of expectation propagation algorithms by identifying the factorization of the likelihood function

$$\mathsf{p}_{\mathbf{r}|\boldsymbol{\zeta}}(\mathbf{r}|\boldsymbol{\zeta}) = \underbrace{\mathsf{p}_{\mathbf{r}|\boldsymbol{\chi}}(\mathbf{r}|\boldsymbol{\chi}(\boldsymbol{\pi}(\boldsymbol{\zeta})))}_{\mathsf{f}_0(\boldsymbol{\theta})} \underbrace{1_{\mathcal{C}}(\boldsymbol{\zeta})}_{\mathsf{f}_1(\boldsymbol{\theta})}$$

where the parameters are $\boldsymbol{\theta} := \boldsymbol{\zeta}$. Next, identifying the messages passed between the component decoders with the parameters of minimal standard exponential families with sufficient statistics $\mathbf{t}(\boldsymbol{\theta}) := \boldsymbol{\theta}$

$$\left[\frac{\mathsf{t}_i(\zeta_i = 1)}{\mathsf{t}_i(\zeta_i = 0)}\right] =: \exp(\boldsymbol{\lambda}_1)$$

$$\left[\frac{\mathsf{u}_i(\zeta_i = 1)}{\mathsf{u}_i(\zeta_i = 0)}\right] =: \exp(\boldsymbol{\lambda}_2)$$

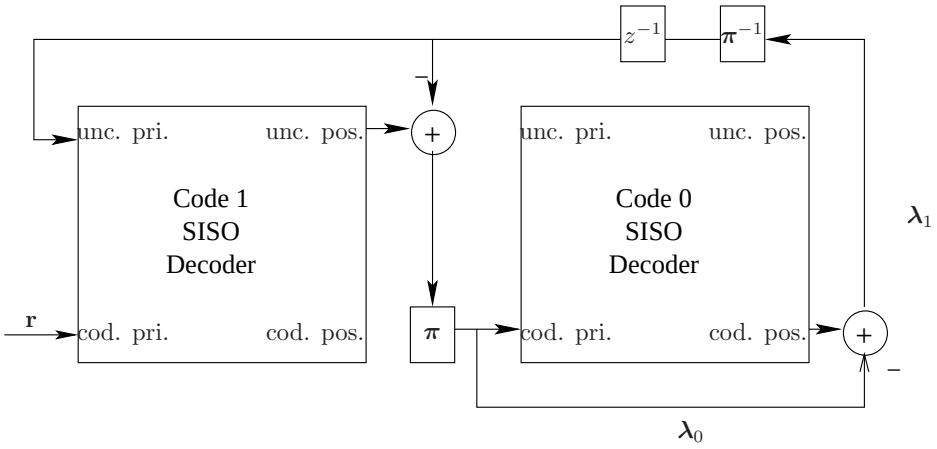

Figure 2.4: The serially concatenated turbo decoder.

gives us the approximating minimal standard exponential families as

$$\mathsf{g}_{\mathsf{a},\boldsymbol{\lambda}_\mathsf{a}}(\theta) := \exp(\theta \cdot \boldsymbol{\lambda}_\mathsf{a} - \psi_\mathbf{t}(\boldsymbol{\lambda}_\mathsf{a})),\ \mathsf{a} \in \{1, 2\}$$

after which the turbo updates (2.12) and (2.13) may be identified with expectation propagation refinement of $\boldsymbol{\lambda}_1$ and $\boldsymbol{\lambda}_2$.

2.1.3 Joint Data Decoding and Detection with Turbo Equalization

Note that the inner code in the serial turbo decoder setup does not necessarily have to be an error control/correction code. Instead, it could result from the convolutional structure of a frequency selective channel. In this instance, one can still use the forward backward algorithm for the inner component decoder in a serial turbo "decoder". The turbo "decoder" which results, first proposed in [25] is often called a turbo equalizer. Because our results in the serial turbo decoder example did not require that the outer code be an error control code, this form of turbo

Table 2.5: Notation used for the Gallager's algorithm for the soft decoding of LDPC codes.

$\mathbf{H}$	parity check matrix
$\mathbf{h}$	row in the parity check matrix
$\mathtt{L}$	number of rows in parity check matrix
$\mathtt{N}$	number of columns in the parity check matrix
$\mathtt{j}$	number of ones in a row
$\mathtt{k}$	number of ones in a column
$\mathbf{0}$	a vector of all zeros
$\mathcal{N}_p$	number of parity check equations
$\mathcal{N}_b$	number of bits
fact	function which returns product of bitwise marginal pdfs
$\mathbf{1}$	a matrix of all ones
$\mathtt{c}$	a constraint equation (e.g. parity check eqn)

equalization may also be considered an instance of the expectation propagation algorithm.

2.1.4 Gallager's Soft Decoding Algorithm for LDPC Codes

In [9] and in his doctoral dissertation, Gallager proposed a family of linear block codes known as low density parity check (LDPC) codes[9, 8, 10]. These codes have the property that their parity check matrices are sparse. For regular LDPC codes (the codes that Gallager considered) each column in the parity check matrix $\mathbf{H} \in \{0,1\}^{\mathtt{L}\times\mathtt{N}}$ has a fixed number $\mathtt{j}$ of 1s and each row has a fixed number $\mathtt{k}$ of 1s. For irregular LDPC codes, the sparse nature of the parity check matrix remains,

but the number of 1s in each column vary according to a degree distribution. Valid codewords lie in the null space of the parity check matrix over the binary field GF(2), so that the code book (the set of codewords) may be written as

$$\mathcal{C} := \{\boldsymbol{\xi} \in \{0,1\}^{\mathtt{N}} | \mathbf{H}\boldsymbol{\xi} = \mathbf{0}\}$$

(where the arithmetic is done in GF(2)). A codeword, $\boldsymbol{\xi}$, is selected at random from this codebook and then transmitted over a memoryless channel to yield the received vector $\mathbf{r}$. Let the set of parity check equations containing $\xi_{\mathtt{i}}$ (i.e. rows in $\mathbf{H}$ containing a 1 at the $\mathtt{i}$th column) be denoted by $\mathcal{N}_p(\mathtt{i})$, and let the set of bits contained in the $\mathtt{a}$th parity check equation (i.e. columns in $\mathbf{H}$ with a 1 in the $\mathtt{a}$th row) be denoted by $\mathcal{N}_b(\mathtt{a})$. Gallager's soft decoding algorithm gives an iterative equation for a probability density on $\boldsymbol{\xi}$ which factors into the product of its bitwise marginals, call it $\tilde{\mathbb{P}}$. The pseudo probability associated with a particular digit $\tilde{\mathbb{P}}(\xi_{\mathtt{i}})$ is separated into components associated with each parity check equation $\mathtt{a}$ associated with that bit (i.e. $\mathtt{a} \in \mathcal{N}_p(\mathtt{i})$), so that

$$\frac{\tilde{\mathbb{P}}(\xi_{\mathtt{i}}=1)}{\tilde{\mathbb{P}}(\xi_{\mathtt{i}}=0)} = \frac{\mathsf{p}_{r_{\mathtt{i}}|\xi_{\mathtt{i}}}(r_{\mathtt{i}}|\boldsymbol{\xi}_{\mathtt{i}}=1)\mathbb{P}[\xi_{\mathtt{i}}=1]}{\mathsf{p}_{r_{\mathtt{i}}|\xi_{\mathtt{i}}}(r_{\mathtt{i}}|\boldsymbol{\xi}_{\mathtt{i}}=0)\mathbb{P}[\xi_{\mathtt{i}}=0]} \prod_{\mathtt{a}\in\mathcal{N}_p(\mathtt{i})} \frac{1-\prod_{\mathtt{j}\in\mathcal{N}_b(\mathtt{a})\setminus\{\mathtt{i}\}}(1-2\mathsf{u}_{\mathtt{a},\mathtt{j}}(\xi_{\mathtt{j}}=1))}{1+\prod_{\mathtt{j}\in\mathcal{N}_b(\mathtt{a})\setminus\{\mathtt{i}\}}(1-2\mathsf{u}_{\mathtt{a},\mathtt{j}}(\xi_{\mathtt{j}}=1))}$$

Where one repeatedly updates u according to the equation

$$\frac{\mathsf{u}_{\mathtt{a},\mathtt{i}}(\xi_{\mathtt{i}}=1)}{\mathsf{u}_{\mathtt{a},\mathtt{i}}(\xi_{\mathtt{i}}=0)} \leftarrow \frac{\mathsf{p}_{r_{\mathtt{i}}|\xi_{\mathtt{i}}}(r_{\mathtt{i}}|\boldsymbol{\xi}_{\mathtt{i}}=1)\mathbb{P}[\xi_{\mathtt{i}}=1]}{\mathsf{p}_{r_{\mathtt{i}}|\xi_{\mathtt{i}}}(r_{\mathtt{i}}|\boldsymbol{\xi}_{\mathtt{i}}=0)\mathbb{P}[\xi_{\mathtt{i}}=0]} \prod_{\mathtt{c}\in\mathcal{N}_p(\mathtt{i})\setminus\{\mathtt{a}\}} \frac{1-\prod_{\mathtt{j}\in\mathcal{N}_b(\mathtt{c})\setminus\{\mathtt{i}\}}(1-2\mathsf{u}_{\mathtt{c},\mathtt{j}}(\xi_{\mathtt{j}}=1))}{1+\prod_{\mathtt{j}\in\mathcal{N}_b(\mathtt{c})\setminus\{\mathtt{i}\}}(1-2\mathsf{u}_{\mathtt{c},\mathtt{j}}(\xi_{\mathtt{j}}=1))} \tag{2.14}$$

this procedure is repeated until either $\tilde{\mathbb{P}}$ converges or a fixed number of iterations have elapsed.

To see how this algorithm is an instance of expectation propagation, we must

first note that the products on the far right of (2.14)

$$\frac{1-\prod_{\mathtt{j}\in\mathcal{N}_b(\mathtt{c})\setminus\{\mathtt{i}\}}(1-2\mathtt{u}_{\mathtt{c},\mathtt{j}}(\xi_\mathtt{j}=1))}{1+\prod_{\mathtt{j}\in\mathcal{N}_b(\mathtt{c})\setminus\{\mathtt{i}\}}(1-2\mathtt{u}_{\mathtt{c},\mathtt{j}}(\xi_\mathtt{j}=1))} \tag{2.15}$$

may be interpreted as the probability, according to $\mathtt{u}_\mathtt{c}$, that an odd number of bits among $\mathcal{N}_b(\mathtt{c})\setminus\{\mathtt{a}\}$ are one divided by the probability that an even number of bits among $\mathcal{N}_b(\mathtt{c})\setminus\{a\}$ are one. This follows from the clever proof of [9], lemma 1. Denoting by $1_{\mathbf{h}_\mathtt{c}}(\boldsymbol{\xi})$ the indicator function for $\boldsymbol{\xi}$ satisfying the $\mathtt{c}$th parity check equation (i.e. $1_{\mathbf{h}_\mathtt{c}}(\boldsymbol{\xi})=1$ if $\mathbf{h}_\mathtt{c}^T\boldsymbol{\xi}=0$ over GF(2) and 0 otherwise), (2.15) may be rewritten as

$$\frac{\sum_{\boldsymbol{\xi}|\xi_\mathtt{i}=1}1_{\mathbf{h}_\mathtt{c}}(\boldsymbol{\xi})\prod_{\mathtt{j}=1,\mathtt{j}\neq\mathtt{i}}^{\mathtt{N}}\mathtt{u}_{\mathtt{c},\mathtt{j}}(\xi_\mathtt{j})}{\sum_{\boldsymbol{\xi}|\xi_\mathtt{i}=0}1_{\mathbf{h}_\mathtt{c}}(\boldsymbol{\xi})\prod_{\mathtt{j}=1,\mathtt{j}\neq\mathtt{i}}^{\mathtt{N}}\mathtt{u}_{\mathtt{c},\mathtt{j}}(\xi_\mathtt{j})}$$

Now, to aid with notation, define the intermediate densities $\mathtt{t}_\mathtt{a}(\boldsymbol{\xi}):=\prod_{\mathtt{i}=1}^{\mathtt{N}}\mathtt{t}_{\mathtt{a},\mathtt{i}}(\xi_\mathtt{i})$ with

$$\frac{\mathtt{t}_{\mathtt{c},\mathtt{i}}(\xi_\mathtt{i}=1)}{\mathtt{t}_{\mathtt{c},\mathtt{i}}(\xi_\mathtt{i}=0)}\leftarrow\frac{\sum_{\boldsymbol{\xi}|\xi_\mathtt{i}=1}1_{\mathbf{h}_\mathtt{c}}(\boldsymbol{\xi})\prod_{\mathtt{j}=1,\mathtt{j}\neq\mathtt{i}}^{\mathtt{N}}\mathtt{u}_{\mathtt{c},\mathtt{j}}(\xi_\mathtt{j})}{\sum_{\boldsymbol{\xi}|\xi_\mathtt{i}=0}1_{\mathbf{h}_\mathtt{c}}(\boldsymbol{\xi})\prod_{\mathtt{j}=1,\mathtt{j}\neq\mathtt{i}}^{\mathtt{N}}\mathtt{u}_{\mathtt{c},\mathtt{j}}(\xi_\mathtt{j})} \tag{2.16}$$

so that we may rewrite (2.14) as

$$\frac{\mathtt{u}_{\mathtt{a},\mathtt{i}}(\xi_\mathtt{i}=1)}{\mathtt{u}_{\mathtt{a},\mathtt{i}}(\xi_\mathtt{i}=0)}=\frac{\mathtt{p}_{r_\mathtt{i}|\xi_\mathtt{i}}(r_\mathtt{i}|\boldsymbol{\xi}_\mathtt{i}=1)\mathbb{P}[\xi_\mathtt{i}=1]}{\mathtt{p}_{r_\mathtt{i}|\xi_\mathtt{i}}(r_\mathtt{i}|\boldsymbol{\xi}_\mathtt{i}=0)\mathbb{P}[\xi_\mathtt{i}=0]}\prod_{\mathtt{c}\in\mathcal{N}_p(\mathtt{i})\setminus\{\mathtt{a}\}}\frac{\mathtt{t}_{\mathtt{c},\mathtt{i}}(\xi_\mathtt{i}=1)}{\mathtt{t}_{\mathtt{c},\mathtt{i}}(\xi_\mathtt{i}=0)}$$

It is now easier to see that this is expectation propagation with a joint density function factorization as

$$\mathtt{p}_{\mathbf{r},\boldsymbol{\xi}}(\mathbf{r},\boldsymbol{\xi})=\left(\prod_{\mathtt{a}=1}^{\mathtt{L}}\underbrace{1_{\mathbf{h}_\mathtt{a}}(\boldsymbol{\xi})}_{\mathtt{f}_\mathtt{a}(\boldsymbol{\xi})}\right)\underbrace{\left(\prod_{\mathtt{i}=1}^{\mathtt{N}}\mathtt{p}_{r_\mathtt{i}|\xi_\mathtt{i}}(r_\mathtt{i}|\xi_\mathtt{i})\mathbb{P}[\xi_\mathtt{i}]\right)}_{\mathtt{f}_{\mathtt{L}+1}(\boldsymbol{\xi})}$$

and approximating minimal standard exponential families

$$\mathtt{g}_\mathtt{a}(\boldsymbol{\xi}):=\exp\left(\mathbf{t}_\mathtt{a}(\boldsymbol{\xi})\cdot\boldsymbol{\lambda}_\mathtt{a}-\psi_{\mathbf{t}_\mathtt{a}}(\boldsymbol{\lambda}_\mathtt{a})\right)\quad\forall\mathtt{i}\in\{1,\ldots,\mathtt{L}+1\}$$

with sufficient statistics

$$\mathbf{t}_\mathtt{a}(\boldsymbol{\xi}):=\boldsymbol{\xi}_{\mathcal{N}_b(\mathtt{a})}\quad\forall\mathtt{a}\in\{1,\ldots,\mathtt{L}\}$$

and

$$\mathbf{t}_{\mathtt{N}+1}(\boldsymbol{\xi}) := \boldsymbol{\xi}$$

where we have used $\boldsymbol{\xi}_{\mathcal{N}_b(\mathtt{a})}$ to specify the vector of digits of $\boldsymbol{\xi}$ which are involved in the $\mathtt{a}$th parity check equation (i.e. columns of $\mathbf{H}$ with a 1 in the $\mathtt{a}$th row)

$$\boldsymbol{\xi}_{\mathcal{N}_b(\mathtt{a})} := [\xi_{\mathtt{i}}]_{\mathtt{i}\in\mathcal{N}_b(\mathtt{a})}$$

Now, consider the refinement equation (2.4), writing it as

$$\frac{\sum_{\boldsymbol{\xi}} \mathbf{t}_{\mathtt{a}}(\boldsymbol{\xi}) \prod_{\mathtt{c}=1}^{\mathtt{L}} \exp(\mathbf{t}_{\mathtt{c}}(\boldsymbol{\xi})\cdot\boldsymbol{\lambda}_{\mathtt{c}} - \psi_{\mathbf{t}_{\mathtt{c}}}(\boldsymbol{\lambda}_{\mathtt{c}}))}{\sum_{\boldsymbol{\xi}} \prod_{\mathtt{c}=1}^{\mathtt{L}} \exp(\mathbf{t}_{\mathtt{c}}(\boldsymbol{\xi})\cdot\boldsymbol{\lambda}_{\mathtt{c}} - \psi_{\mathbf{t}_{\mathtt{c}}}(\boldsymbol{\lambda}_{\mathtt{c}}))} = \frac{\sum_{\boldsymbol{\xi}} \mathbf{t}_{\mathtt{a}}(\boldsymbol{\xi})\mathsf{f}_{\mathtt{a}}(\boldsymbol{\xi}) \prod_{\mathtt{c}\neq\mathtt{a}} \exp(\mathbf{t}_{\mathtt{c}}(\boldsymbol{\xi})\cdot\boldsymbol{\lambda}_{\mathtt{c}})}{\sum_{\boldsymbol{\xi}} \mathsf{f}_{\mathtt{a}}(\boldsymbol{\xi}) \prod_{\mathtt{c}\neq\mathtt{a}} \exp(\mathbf{t}_{\mathtt{c}}(\boldsymbol{\xi})\cdot\boldsymbol{\lambda}_{\mathtt{c}})} \tag{2.17}$$

allows us to see that the expectation of the sufficient statistics are actually the bitwise marginal probabilities in this instance. Writing (2.17) in terms of bitwise probability ratios, then, yields

$$\begin{aligned}&\frac{\sum_{\boldsymbol{\xi}|\xi_{\mathtt{i}}=1} \prod_{\mathtt{c}=1}^{\mathtt{L}} \exp(\mathbf{t}_{\mathtt{c}}(\boldsymbol{\xi})\cdot\boldsymbol{\lambda}_{\mathtt{c}} - \psi_{\mathbf{t}_{\mathtt{c}}}(\boldsymbol{\lambda}_{\mathtt{c}}))}{\sum_{\boldsymbol{\xi}|\xi_{\mathtt{i}}=0} \prod_{\mathtt{c}=1}^{\mathtt{L}} \exp(\mathbf{t}_{\mathtt{c}}(\boldsymbol{\xi})\cdot\boldsymbol{\lambda}_{\mathtt{c}} - \psi_{\mathbf{t}_{\mathtt{c}}}(\boldsymbol{\lambda}_{\mathtt{c}}))} \\ &= \frac{\sum_{\boldsymbol{\xi}|\xi_{\mathtt{i}}=1} \mathsf{f}_{\mathtt{a}}(\boldsymbol{\xi}) \prod_{\mathtt{c}=1,\mathtt{c}\neq\mathtt{a}}^{\mathtt{L}} \exp(\mathbf{t}_{\mathtt{c}}(\boldsymbol{\xi})\cdot\boldsymbol{\lambda}_{\mathtt{c}} - \psi_{\mathbf{t}_{\mathtt{c}}}(\boldsymbol{\lambda}_{\mathtt{c}}))}{\sum_{\boldsymbol{\xi}|\xi_{\mathtt{i}}=0} \mathsf{f}_{\mathtt{a}}(\boldsymbol{\xi}) \prod_{\mathtt{c}=1,\mathtt{c}\neq\mathtt{a}}^{\mathtt{L}} \exp(\mathbf{t}_{\mathtt{c}}(\boldsymbol{\xi})\cdot\boldsymbol{\lambda}_{\mathtt{c}} - \psi_{\mathbf{t}_{\mathtt{c}}}(\boldsymbol{\lambda}_{\mathtt{c}}))}\end{aligned} \tag{2.18}$$

If we then identify

$$\boldsymbol{\lambda}_{\mathtt{a}} := \left[\log\left(\frac{\mathsf{t}_{\mathtt{a},\mathtt{i}}(\xi_{\mathtt{i}}=1)}{\mathsf{t}_{\mathtt{a},\mathtt{i}}(\xi_{\mathtt{i}}=0)}\right)\right]$$

we can rewrite (2.18) as

$$\prod_{\mathtt{c}\in\mathcal{N}_p(\mathtt{a})} \frac{\mathsf{t}_{\mathtt{c},\mathtt{i}}(\xi_{\mathtt{i}}=1)}{\mathsf{t}_{\mathtt{c},\mathtt{i}}(\xi_{\mathtt{i}}=0)} = \left(\prod_{\mathtt{c}\in\mathcal{N}_p(\mathtt{a})\setminus\{\mathtt{a}\}} \frac{\mathsf{t}_{\mathtt{c},\mathtt{i}}(\xi_{\mathtt{i}}=1)}{\mathsf{t}_{\mathtt{c},\mathtt{i}}(\xi_{\mathtt{i}}=0)}\right) \frac{\sum_{\boldsymbol{\xi}|\xi_{\mathtt{i}}=1} \mathsf{f}_{\mathtt{a}}(\boldsymbol{\xi}) \prod_{\mathtt{j}\neq\mathtt{i}} \mathsf{u}_{\mathtt{a},\mathtt{j}}(\xi_{\mathtt{j}})}{\sum_{\boldsymbol{\xi}|\xi_{\mathtt{i}}=0} \mathsf{f}_{\mathtt{a}}(\boldsymbol{\xi}) \prod_{\mathtt{j}\neq\mathtt{i}} \mathsf{u}_{\mathtt{a},\mathtt{j}}(\xi_{\mathtt{j}})}$$

which, after dividing by some common terms, gives

$$\frac{\mathsf{t}_{\mathtt{a},\mathtt{i}}(\xi_{\mathtt{i}}=1)}{\mathsf{t}_{\mathtt{a},\mathtt{i}}(\xi_{\mathtt{i}}=0)} = \frac{\sum_{\boldsymbol{\xi}|\xi_{\mathtt{i}}=1} \mathsf{f}_{\mathtt{a}}(\boldsymbol{\xi}) \prod_{\mathtt{j}\neq\mathtt{i}} \mathsf{u}_{\mathtt{a},\mathtt{j}}(\xi_{\mathtt{j}})}{\sum_{\boldsymbol{\xi}|\xi_{\mathtt{i}}=0} \mathsf{f}_{\mathtt{a}}(\boldsymbol{\xi}) \prod_{\mathtt{j}\neq\mathtt{i}} \mathsf{u}_{\mathtt{a},\mathtt{j}}(\xi_{\mathtt{j}})}$$

This shows that refining $\boldsymbol{\lambda}_{\mathtt{a}}$ according to expectation propagation is equivalent to the LDPC decoding step (2.16), and thus that Gallager's algorithm for the soft

Table 2.6: The notation introduced for belief propagation.

$\mathcal{FN}$	the factor neighbors of a parameter
$\mathcal{PN}$	the parameter neighbors of a factor
n	a message passed from a parameter to a factor
m	a message passed from a factor to a parameter

decoding of LDPC codes is a member of the family of expectation propagation algorithms.

2.1.5 Belief Propagation-Factor Graph Formulation

Belief propagation is a method for statistical inference originally given that name by Pearl[13]. It may be considered to be equivalent to a number of other independently developed algorithms, as discussed in [14], and as an instance of the sum product algorithm (which need not involve statistical inference). In this section we show how belief propagation is an instance of expectation propagation. We will use the factor graph form of belief propagation found in [35] and [14]. For readers who are more familiar with the Bayesian network or markov random field formulation, we refer them to [14] which shows the equivalence of the different descriptions.

Like expectation propagation, of central interest in belief propagation is a joint likelihood function for some observations $\mathbf{r}$ and some parameters $\boldsymbol{\theta}$ which we would like to infer which factors into functions of subsets of the variables $\boldsymbol{\theta}$. This is mathematically described as

$$\mathsf{p}_{\mathbf{r},\boldsymbol{\theta}}(\mathbf{r},\boldsymbol{\theta}) := \prod_{\mathsf{a}=1}^{\mathtt{M}} \mathsf{f}_{\mathsf{a}}(\boldsymbol{\theta}_{\mathsf{a}}) \quad \boldsymbol{\theta}_{\mathsf{a}} \subset \boldsymbol{\theta}_{\mathsf{a}}$$

where the factors f_{a} are implicitly functions of $\mathbf{r}$, but we have observed a particular value of $\mathbf{r}$ and thus are using that throughout the rest of the discussion, hence

dropping it from the notation. We will assume the parameters $\boldsymbol{\theta}$ to have been drawn from some finite set Θ.

In its factor graph formulation, belief propagation may be interpreted as a message passing algorithm on the parameter factor graph to be discussed in Section 2.2. The parameter factor graph is a bipartite graph whose $\mathtt{N}$ "left" nodes correspond to the parameters $\theta_{\mathtt{i}}$ $\mathtt{i} \in \{1, \ldots, \mathtt{N}\}$ and whose $\mathtt{M}$ "right" nodes correspond to the factors $\mathsf{f}_{\mathtt{a}}$ $\mathtt{a} \in \{1, \ldots, \mathtt{M}\}$. The messages passed from the variable nodes are defined by the updates

$$\mathsf{n}_{\mathtt{i}\to\mathtt{a}}(\theta_{\mathtt{i}}) \leftarrow \frac{\prod_{\mathtt{a}\in\mathcal{PN}(\mathtt{i})\setminus\{\mathtt{a}\}} \mathsf{m}_{\mathtt{a}\to\mathtt{i}}(\theta_{\mathtt{i}})}{\int \prod_{\mathtt{a}\in\mathcal{PN}(\mathtt{i})\setminus\{\mathtt{a}\}} \mathsf{m}_{\mathtt{a}\to\mathtt{i}}(\theta'_{\mathtt{i}})\mathsf{d}\theta'_{\mathtt{i}}}$$

where here $\mathcal{PN}(\mathtt{i})$ denotes the factor nodes that neighbor (i.e. share an edge with) the parameter node $\theta_{\mathtt{i}}$ and $\mathsf{m}_{\mathtt{a}\to\mathtt{i}}(\theta_{\mathtt{i}})$ is the previous message passed from factor node $\mathtt{a}$ to parameter node $\theta_{\mathtt{i}}$. The messages from the factor nodes the variable nodes, in turn, are defined by

$$\mathsf{m}_{\mathtt{a}\to\mathtt{i}}(\theta_{\mathtt{i}}) = \frac{\int \mathsf{f}_{\mathtt{a}}(\boldsymbol{\theta}_{\mathtt{a}}) \prod_{\mathtt{j}\in\mathcal{FN}(\mathtt{a})\setminus\{\mathtt{i}\}} \mathsf{n}_{\mathtt{j}\to\mathtt{a}}(\theta_{\mathtt{i}})\mathsf{d}\boldsymbol{\theta}_{\mathtt{a}} \setminus \boldsymbol{\theta}_{\mathtt{i}}}{\int \mathsf{f}_{\mathtt{a}}(\boldsymbol{\theta}_{\mathtt{a}}) \prod_{\mathtt{j}\in\mathcal{FN}(\mathtt{a})\setminus\{\mathtt{i}\}} \mathsf{n}_{\mathtt{j}\to\mathtt{a}}(\theta_{\mathtt{i}})\mathsf{d}\boldsymbol{\theta}_{\mathtt{a}}} \tag{2.19}$$

Because Θ is a finite set, we do not actually have to pass functions, but instead only need to pass parameters of an exponential family with sufficient statistics $\mathbf{t} : \mathbb{R} \to \mathbb{R}^{|\Theta_{\mathtt{i}}|-1}$. Denote the possible values $\Theta_{\mathtt{i}}$ of $\theta_{\mathtt{i}}$ by $\{\alpha_1, \ldots, \alpha_{\mathtt{V}}\}$ then the sufficient statistics are

$$\mathbf{t}_{\mathtt{i}}(\theta_{\mathtt{i}}) := \begin{cases} \mathbf{e}_{\mathtt{j}} & \theta_{\mathtt{i}} = \alpha_{\mathtt{j}},\ \mathtt{j} \neq 1 \\ \mathbf{0} & \text{otherwise} \end{cases}$$

where $\mathbf{e}_{\mathtt{j}}$ is the $\mathtt{j}$th column of the identity matrix of dimension $|\Theta_{\mathtt{i}}| - 1$.

At any given time, our estimate of the a posteriori density for $\theta_{\mathtt{i}}$ is the product of the two messages on any of its adjacent edges

$$\mathsf{q}_{\mathtt{i}}(\theta_{\mathtt{i}}) := \mathsf{n}_{\mathtt{i}\to\mathtt{a}}(\theta_{\mathtt{i}})\mathsf{m}_{\mathtt{a}\to\mathtt{i}}(\theta_{\mathtt{i}}) = \prod_{\mathtt{a}\in\mathcal{N}(\mathtt{i})} \mathsf{m}_{\mathtt{a}\to\mathtt{i}}(\theta_{\mathtt{i}})$$

To see how belief propagation may be interpreted as an instance of expectation propagation, note that the approximating densities $\mathsf{g}_\mathtt{a}$ are exponential families with parameters that are the concatenation of $\mathbf{t}_\mathtt{i}(\theta_\mathtt{i})$ over all $\mathtt{i} \in \mathcal{PN}(\mathtt{a})$, so that

$$\mathbf{t}_\mathtt{a}(\boldsymbol{\theta}_\mathtt{a}) := [\mathbf{t}_\mathtt{i}(\theta_\mathtt{i})]_{\mathtt{i} \in \mathcal{PN}(\mathtt{a})}$$

Note that the dependence of $\mathbf{t}_\mathtt{i}$ on only one parameter $\theta_\mathtt{i}$ yields the $\mathbf{t}_\mathtt{i}$'s independent of each other within the exponential family, so that for any $\mathtt{i} \in \mathcal{PN}(\mathtt{a})$

$$\begin{aligned} &\int_{\Theta_\mathtt{a}} \mathbf{t}_\mathtt{i}(\boldsymbol{\theta}_\mathtt{i}) \exp\left(\mathbf{t}_\mathtt{a}(\boldsymbol{\theta}_\mathtt{a}) \cdot \boldsymbol{\lambda}_\mathtt{a} - \psi_{\mathbf{t}_\mathtt{a}}(\boldsymbol{\lambda}_\mathtt{a})\right) \mathsf{d}\boldsymbol{\theta}_\mathtt{a} = \\ &\int_{\Theta_\mathtt{i}} \mathbf{t}_\mathtt{i}(\boldsymbol{\theta}_\mathtt{i}) \exp\left(\mathbf{t}_\mathtt{i}(\boldsymbol{\theta}_\mathtt{i}) \cdot [\boldsymbol{\lambda}_\mathtt{a}]_\mathtt{i} - \psi_{\mathbf{t}_\mathtt{i}}([\boldsymbol{\lambda}_\mathtt{a}]_\mathtt{i})\right) \mathsf{d}\boldsymbol{\theta}_\mathtt{i} \end{aligned} \tag{2.20}$$

Now, identify the message functions in terms of the parameters from the expectation propagation algorithm

$$\mathsf{n}_{\mathtt{i}\to\mathtt{a}}(\theta_\mathtt{i}) := \exp\left(\mathbf{t}_\mathtt{i}(\theta_\mathtt{i}) \cdot \left(\sum_{\mathtt{c} \in \mathcal{FN}(\mathtt{i}) \setminus \{\mathtt{a}\}} [\boldsymbol{\lambda}_\mathtt{c}]_\mathtt{i}\right) - \psi_{\mathbf{t}_\mathtt{i}}\left(\sum_{\mathtt{c} \in \mathcal{FN}(\mathtt{i}) \setminus \{\mathtt{a}\}} [\boldsymbol{\lambda}_\mathtt{c}]_\mathtt{i}\right)\right)$$

and

$$\mathsf{m}_{\mathtt{a}\to\mathtt{i}}(\theta_\mathtt{i}) := \exp\left(\mathbf{t}_\mathtt{i}(\theta_\mathtt{i}) \cdot [\boldsymbol{\lambda}_\mathtt{a}]_\mathtt{i} - \psi_{\mathbf{t}_\mathtt{i}}([\boldsymbol{\lambda}_\mathtt{a}]_\mathtt{i})\right)$$

Next, multiply the belief propagation update equation (2.19) through by $\mathsf{m}_{\mathtt{a}\to\mathtt{i}}(\theta_\mathtt{i})$ to get

$$\begin{aligned} &\exp\left(\mathbf{t}_\mathtt{i}(\theta_\mathtt{i}) \cdot \left(\sum_{\mathtt{a} \in \mathcal{FN}(\mathtt{i})} [\boldsymbol{\lambda}_\mathtt{a}]_\mathtt{i}\right) - \psi_{\mathbf{t}_\mathtt{i}}\left(\sum_{\mathtt{a} \in \mathcal{FN}(\mathtt{i})} [\boldsymbol{\lambda}_\mathtt{a}]_\mathtt{i}\right)\right) \\ &= \frac{\int \mathsf{f}_\mathtt{a}(\boldsymbol{\theta}_\mathtt{a}) \exp\left(\sum_{\mathtt{c} \neq \mathtt{a}} \boldsymbol{\lambda}_\mathtt{c} \cdot \mathbf{t}_\mathtt{c}(\boldsymbol{\theta}_\mathtt{c})\right) \mathsf{d}\boldsymbol{\theta} \setminus \theta_\mathtt{i}}{\int_\Theta \mathsf{f}_\mathtt{a}(\boldsymbol{\theta}_\mathtt{a}) \exp\left(\sum_{\mathtt{c} \neq \mathtt{a}} \boldsymbol{\lambda}_\mathtt{c} \cdot \mathbf{t}_\mathtt{c}(\boldsymbol{\theta}_\mathtt{c})\right) \mathsf{d}\boldsymbol{\theta}} \end{aligned} \tag{2.21}$$

which is equivalent to the expectation propagation update equation (2.3), establishing that belief propagation is an instance of expectation propagation.

Table 2.7: Notation introduced for section 2.2.

perm	corresponding permutation
v	elementary statistics vectors
n	message from elementary statistic vector to factor
m	message from factor to elementary statistic vector
$\mathcal{SN}$	statistics which neighbor a factor/parameter node
T	number of sufficient statistics
V	number of elementary statistics vectors
$\mathcal{T}$	set of sufficient statistics
$\mathcal{V}$	set of elementary statistics vectors
$\mathcal{F}$	set of factors
t	vector containing all statistics
e	an edge
$\mathcal{E}$	set of edges
$\mathcal{V}$	set of nodes
L	number of edges
$\mathcal{EN}$	el. stat. vec.s which neighbor a factor/parameter node

2.2 Message Passing and Reciprocity

Under some special conditions satisfied by all of our previous examples, expectation propagation may be identified as a message passing algorithm on a statistics factor graph. We will focus exclusively on these cases for the remainder of the dissertation, so here we outline these conditions as assumptions for the remainder of the dissertation and discuss their implications including the message passing framework.

The first intuitively reasonable assumption is that the sufficient statistics $\mathbf{t}_{\mathtt{a}}(\boldsymbol{\theta}_{\mathtt{a}})$ for the approximating density $\mathsf{g}_{\mathtt{a},\boldsymbol{\lambda}_{\mathtt{a}}}$ are also sufficient statistics for the factors $\mathsf{f}_{\mathtt{a}}$ so that $\mathsf{f}_{\mathtt{a}}(\boldsymbol{\theta}_{\mathtt{a}}) = \hat{\mathsf{f}}_{\mathtt{a}}(\mathbf{t}_{\mathtt{a}}(\boldsymbol{\theta}_{\mathtt{a}}))\ \forall \boldsymbol{\theta}_{\mathtt{a}} \in \Theta_{\mathtt{a}}$. Because we are using $\mathsf{g}_{\mathtt{a},\boldsymbol{\lambda}_{\mathtt{a}}}$ to approximate $\mathsf{f}_{\mathtt{a}}$, this assumption seems reasonable, since it requires that all of the information that $\mathsf{f}_{\mathtt{a}}$ depends on be present in the vector $\mathbf{t}_{\mathtt{a}}$.

As. 1 (Sufficiency of Approximating Statistics)**:** The sufficient statistics $\mathbf{t}_{\mathtt{a}}(\boldsymbol{\theta}_{\mathtt{a}})$ for the approximating density $\mathsf{g}_{\mathtt{a},\boldsymbol{\lambda}_{\mathtt{a}}}$ are also sufficient statistics for the factors $\mathsf{f}_{\mathtt{a}}$.

Next, we construct the statistics factor graph. In most cases, the same statistic will occur in several places (i.e. $[\mathbf{t}_{\mathtt{a}}]_{\mathtt{i}} = [\mathbf{t}_{\mathtt{c}}]_{\mathtt{j}}$ for some $\mathtt{a}, \mathtt{c}, \mathtt{i}, \mathtt{j}$). Collect, then, all of the statistics $\mathtt{t}_{\mathtt{i}} : \Theta \rightarrow \mathbb{R}$ used in the sufficient statistic vectors $\{\mathbf{t}_{\mathtt{a}}(\boldsymbol{\theta}_{\mathtt{a}}) | \mathtt{a} \in \{1, \ldots, \mathtt{M}\}\}$ for the approximating functions/factors, into the set $\mathcal{T} = \{\mathtt{t}_{\mathtt{i}}(\boldsymbol{\theta}_{\mathtt{i}}), \mathtt{i} \in \{1, \ldots, \mathtt{T}\}\}$ without replication. Collect all of the elements of $\mathcal{T}$ into a vector $\mathbf{t}$.

Next, partition $\mathcal{T}$ up into the disjoint union $\mathcal{T} = \bigcup_{\mathtt{j}} \mathcal{T}_{\mathtt{j}}$ such that $\mathtt{t}_{\mathtt{i}} \in \mathcal{T}_{\mathtt{j}}$ and $\mathtt{t}_{\mathtt{i}} \in \mathbf{t}_{\mathtt{a}}$ implies that for all $\mathtt{t}_{\mathtt{k}} \in \mathcal{T}_{\mathtt{j}}$ we have $\mathtt{t}_{\mathtt{k}} \in \mathbf{t}_{\mathtt{a}}$. For each $\mathtt{j}$ collect the elements of $\mathcal{T}_{\mathtt{j}}$ into a vector $\mathbf{v}_{\mathtt{j}}$. Our construction now allows us to write each vector $\mathbf{t}_{\mathtt{a}}$ as the concatenation of several $\mathbf{v}$'s. Once again, it is likely that the same $\mathbf{v}_{\mathtt{i}}$ will

appear in several $\mathbf{t}$'s. We can now depict this replication of a statistic $\mathbf{v}_j$ in several $\mathbf{t}_a$'s via a factor graph [14].

Def. 1 (Factor Graph)**:** A factor graph $(\mathcal{V}, \mathcal{E})$, is undirected bipartite graph representing a particular multiplicative factorization of a function, e.g.

$$f(\mathbf{x}) := \prod_a f_a(\mathbf{x}_a), \quad \mathbf{x}_a \subset \mathbf{x}$$

(where $\mathbf{x}_a \subset \mathbf{x}$ indicates that the elements of the former vector are a subset of the elements of the latter). The graph, like any other undirected graph, is an ordered pair of two sets, a set of nodes $\mathcal{V}$ and a set of edges $\mathcal{E}$, which are unordered pairs of nodes. The nodes in the graph are divided into two sets, a set $\mathcal{V} = \{x_1, \ldots, x_T\}$ of *variable nodes*, and a set $\mathcal{F} = \{f_1, \ldots, f_M\}$ of factor nodes. An edge $e \in \mathcal{E}$ may only connect a variable node to a factor node and appears only if the variable associated with the variable node is an argument of the factor associated with the factor node.

In our case, the variable node set $\mathcal{V} := \{\mathbf{v}_j\}$ and the factor node set is $\mathcal{F} := \left\{\hat{f}_a | a \in \{1, \ldots, M\}\right\}$. An edge connects $\mathbf{v}_j$ with $\hat{f}_a$ if $\hat{f}_a$ has as an argument $\mathbf{v}_j$, which of course happens if $\mathbf{v}_j$ is a vector formed from a subset of the elements of $\mathbf{t}_a$ (i.e. $\mathbf{v}_j \subset \mathbf{t}_a$). We call the graph that results a statistics factor graph, an example of which is shown in Figure 2.5.

We can also form a second factor graph that we will call the parameter statistics graph, an example of which is shown in Figure 2.6. This is again a bipartite graph with a set of "left" nodes which are the elements $\{\theta_i\}$ of the parameter vector $\boldsymbol{\theta}$ and a set of "right" nodes which are the elementary statistics vectors $\{\mathbf{v}_j\}$. Each of the elementary statistics vectors $\mathbf{v} = \mathbf{v}_j(\boldsymbol{\theta}_{\mathbf{v},j})$ depends on a subset $\boldsymbol{\theta}_{\mathbf{v},j}$ of the original parameters $\boldsymbol{\theta}$. An edge occurs in the parameter statistics graph between a parameter θ_i and an elementary statistics vector $\mathbf{v}_j$ if and only if $\mathbf{v}_j$ depends on

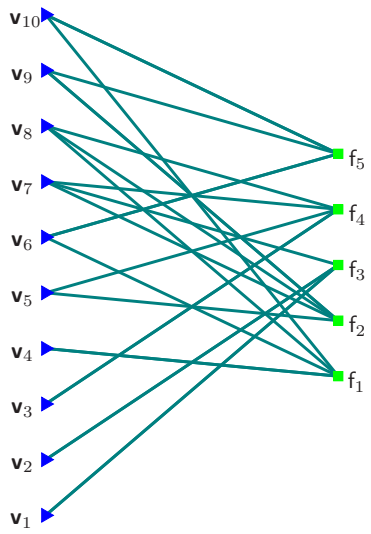

Figure 2.5: An example of a statistics factor graph.

$\theta_{\mathtt{i}}$, which of course happens if and only if $\theta_{\mathtt{i}} \in \boldsymbol{\theta}_{\mathbf{v},\mathtt{j}}$.

Note that we may concatenate the parameter statistics graph with the statistics factor graph in a natural manner, since the "right" nodes of the parameter statistics graph are exactly the "left" nodes of the statistics factor graph. We call the graph that results the parameter statistics factor graph.

Our next assumption enforces a sort of regularity in the parameter statistics graph.

As. 2 (Reciprocity)**:** The Radon Nikodym derivative μ of the Lebesgue measure of inverse image $\mathbf{t}^{-1}(\mathcal{A})$ with respect to the Lebesgue measure of $\mathcal{A}$ factors into the product of functions of $\mathbf{v}_{\mathtt{j}}$

$$\mu(\mathbf{t}) = \prod_{\mathtt{j}} \mu_{\mathtt{j}}(\mathbf{v}_{\mathtt{j}}) \tag{2.22}$$

A particularly simple way to ensure the reciprocity assumption is to have each

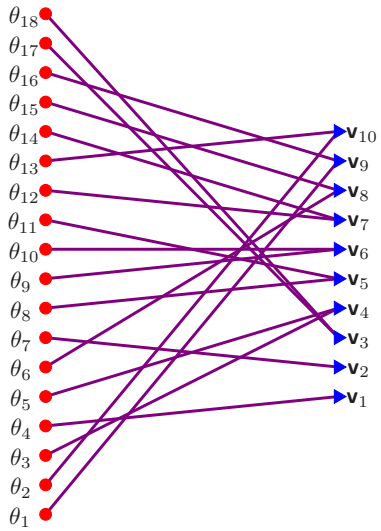

Figure 2.6: An example of a parameter statistics graph.

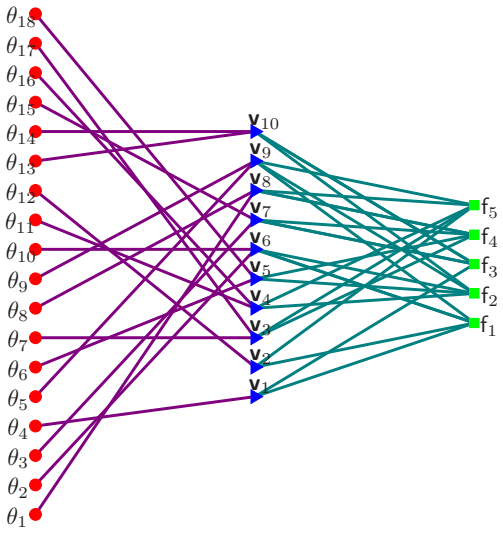

Figure 2.7: An example of a parameter statistics factor graph.

parameter appear as an argument for only one elementary statistics vector.

Prop. 1 (Meaning of Reciprocity)**:** The reciprocity assumption holds if the degree of each parameter (i.e. left node) in the parameter statistics graph is 1.

Proof: If the degree of each parameter node $\theta_{\mathtt{i}}$ is one, then it may only appear in one elementary statistics vector $\mathbf{v}_{\mathtt{j}}$. Thus, we may partition the vector of parameters $\boldsymbol{\theta}$ into the disjoint concatenation and reordering of the set of vectors $\boldsymbol{\theta}_{\mathbf{v},\mathtt{j}}$ which are the arguments of $\mathbf{v}_{\mathtt{j}}$ for all $\mathtt{j} \in \{1, \ldots, \mathtt{V}\}$. Let the corresponding permutation, which takes the concatenated $param_{\mathbf{v},\mathtt{j}}$ and reorders them into the order found in $\boldsymbol{\theta}$ be denoted by **perm**. There is then a one to one correspondence between subsets $\mathcal{A} \subset \Theta$ and subsets $\hat{\mathcal{A}} = \mathbf{perm}(\mathcal{A})$ by simple reordering of the indices of the vector space.

$$\{\boldsymbol{\theta} | \mathbf{v}(\boldsymbol{\theta}) \in \mathcal{A}_1 \times \cdots \times \mathcal{A}_{\mathtt{i}} \times \cdots \times \mathcal{A}_{\mathtt{V}}\} =$$
$$\mathbf{perm}\left(\{\boldsymbol{\theta}_{\mathbf{v},1} | \mathbf{v}_1(\boldsymbol{\theta}_{\mathbf{v},1}) \in \mathcal{A}_1\} \times \cdots \times \{\boldsymbol{\theta}_{\mathbf{v},\mathtt{V}} | \mathbf{v}_{\mathtt{V}}(\boldsymbol{\theta}_{\mathbf{v},\mathtt{V}}) \in \mathcal{A}_{\mathtt{V}}\}\right) \tag{2.23}$$

This, together with the product nature of Lebesgue measure on $\mathbb{R}^{\mathtt{V}}$ and Lebesgue measure on $\mathbb{R}^{\mathtt{N}}$ implies the desired factorization. ■

We can intuitively summarize the reciprocity condition as requiring that everywhere that we approximate $\theta_{\mathtt{i}}$ we approximate it with the same statistics and the same exponential family distribution. In other words, we will not approximate $\theta_{\mathtt{i}}$ as a Gaussian distribution in one context, and then simultaneously an exponential distribution in another context. This too, seems intuitively reasonable.

The additional benefit of the reciprocity condition is that now we may consider expectation propagation to be a message passing algorithm on the statistics factor graph. In particular, due to the reciprocity condition (2.22) the EP update

equation (2.4) simplifies to

$$\frac{\int_{\Theta_\mathtt{a}} \mathbf{t}_\mathtt{a}(\boldsymbol{\theta}_\mathtt{a})\hat{\mathsf{f}}_\mathtt{a}(\mathbf{t}_\mathtt{a}(\boldsymbol{\theta}_\mathtt{a}))\exp\left(\mathbf{t}_\mathtt{a}(\boldsymbol{\theta}_\mathtt{a})\cdot\boldsymbol{\lambda}_{\text{in}}\right)\mathsf{d}\boldsymbol{\theta}_\mathtt{a}}{\int_{\Theta_\mathtt{a}} \hat{\mathsf{f}}_\mathtt{a}(\mathbf{t}_\mathtt{a}(\boldsymbol{\theta}_\mathtt{a}))\exp\left(\mathbf{t}_\mathtt{a}(\boldsymbol{\theta}_\mathtt{a})\cdot\boldsymbol{\lambda}_{\text{in}}\right)\mathsf{d}\boldsymbol{\theta}_\mathtt{a}} = \frac{\int_{\Theta_\mathtt{a}} \mathbf{t}_\mathtt{a}(\boldsymbol{\theta}_\mathtt{a})\exp\left(\mathbf{t}_\mathtt{a}(\boldsymbol{\theta}_\mathtt{a})\cdot(\boldsymbol{\lambda}_{\text{in}}+\boldsymbol{\lambda}_\mathtt{a})\right)\mathsf{d}\boldsymbol{\theta}_\mathtt{a}}{\int_{\Theta_\mathtt{a}} \exp\left(\mathbf{t}_\mathtt{a}(\boldsymbol{\theta}_\mathtt{a})\cdot(\boldsymbol{\lambda}_{\text{in}}+\boldsymbol{\lambda}_\mathtt{a})\right)\mathsf{d}\boldsymbol{\theta}_\mathtt{a}} \tag{2.24}$$

where

$$[\boldsymbol{\lambda}_{\text{in}}]_\mathtt{i} := \sum_{\mathtt{c}\in\mathcal{N}(\mathtt{i})\setminus\{\mathtt{a}\}} [\boldsymbol{\lambda}_\mathtt{c}]_\mathtt{i} =: [\mathbf{n}_{\mathtt{j}\to\mathtt{a}}]_\mathtt{i} \tag{2.25}$$

where $\mathcal{N}(\mathtt{i})$ are the factors in the statistics factor graph that are neighbors (i.e. share an edge) with the elementary statistics vector $\mathbf{v}_\mathtt{j}$ containing $\mathtt{t}_\mathtt{i}$, and $[\boldsymbol{\lambda}_\mathtt{c}]_\mathtt{i}$ is the parameter of the $\mathtt{c}$th approximating function multiplying the statistic $\mathtt{t}_\mathtt{i}$. This shows that it suffices to assume $\mathbf{t}_\mathtt{a} = \mathbf{u}_\mathtt{a}$ ($\mathbf{u}_\mathtt{a}$ are the statistics whose exponential family weight the factor $\mathsf{f}_\mathtt{a}(\boldsymbol{\theta}_\mathtt{a})$ in the expectation propagation update), since any statistics in $\mathbf{t}$ other than $\mathbf{t}_\mathtt{a}$ are independent of $\mathbf{t}_\mathtt{a}$ and thus drop out of the a posteriori density for $\boldsymbol{\theta}_\mathtt{a}$.

From (2.24) and (2.25) we can determine the equivalence between expectation propagation and a message passing algorithm on the statistics factor graph. The message passing algorithm may only send messages along edges in the factor graph. The messages from factor node $\mathsf{f}_\mathtt{a}$ to the elementary statistics vector $\mathbf{v}_\mathtt{j}$ are based on the solutions $\boldsymbol{\lambda}_\mathtt{a}$ to (2.24) and take the form

$$\mathbf{m}_{\mathtt{j}\leftarrow\mathtt{a}} := [[\boldsymbol{\lambda}_\mathtt{a}]_\mathtt{i} \,|\mathtt{t}_\mathtt{i} \in \mathbf{v}_\mathtt{j}]$$

that is, they are the components of $\boldsymbol{\lambda}_\mathtt{a}$ which multiply $\mathbf{v}_\mathtt{j} \subset \mathbf{t}_\mathtt{a}$. The messages going in the reverse direction from the elementary statistics vector $\mathbf{v}_\mathtt{j}$ to the factor $\mathsf{f}_\mathtt{a}$ are defined by (2.25). Of course, due to the simple vector addition form of (2.25), the computation may be done either collectively at the single node $\mathbf{v}_\mathtt{j}$, or equivalently $\mathbf{v}_\mathtt{j}$ may be split into nodes that are its components $\mathtt{t}_\mathtt{i} \in \mathbf{v}_\mathtt{j}$ and the element-wise addition may be done there separately.

Now that we have clarified how expectation propagation may be interpreted as a message passing algorithm on the statistics factor graph under the reciprocity and sufficiency assumptions, we will move on to discuss some special cases satisfying these assumptions in which the stationary points of expectation propagation are "optimal".

Chapter 3

Special Cases: Convergence and Optimality

In this section, we cover some special cases in which the stationary points of expectation propagation may be easily seen to provide the correct a posteriori values of the parameters.

3.1 Matching Sufficient Statistics

Prop. 2 (Perfectly Matched Case)**:** In the special case that the actual factors f_a are minimal standard exponential family densities *whose sufficient statistics match those of the approximating densities* $\mathsf{g}_{\mathsf{a},\boldsymbol{\lambda}_\mathsf{a}}$, the aggregate approximate density

$$\prod_{\mathsf{a}=1}^{\mathtt{M}} \mathsf{g}_{\mathsf{a},\boldsymbol{\lambda}_\mathsf{a}}(\boldsymbol{\theta}_\mathsf{a})$$

converges to the true density as soon as all of the factors $\mathsf{g}_{\mathsf{a},\boldsymbol{\lambda}_\mathsf{a}}$ have been refined at least once.

Proof: To see this, suppose the parameters of the factors f_a are $\boldsymbol{\lambda}_\mathsf{a}^*$. Then, consider the refinement step (2.3) and note that we can make

$$\frac{\mathsf{f}_\mathsf{a}(\boldsymbol{\theta}_\mathsf{a}) \prod_{\mathsf{c}\neq\mathsf{a}} \mathsf{g}_{\mathsf{c},\boldsymbol{\lambda}_\mathsf{c}}(\boldsymbol{\theta}_\mathsf{c})}{\int_\Theta \mathsf{f}_\mathsf{a}(\boldsymbol{\theta}_\mathsf{a}) \prod_{\mathsf{c}\neq\mathsf{a}} \mathsf{g}_{\mathsf{c},\boldsymbol{\lambda}_\mathsf{c}}(\boldsymbol{\theta}_\mathsf{c})} = \frac{\prod_\mathsf{c} \mathsf{g}_{\mathsf{c},\boldsymbol{\lambda}_\mathsf{c}}(\boldsymbol{\theta}_\mathsf{c})}{\int_\Theta \prod_\mathsf{c} \mathsf{g}_{\mathsf{c},\boldsymbol{\lambda}_\mathsf{c}}(\boldsymbol{\theta}_\mathsf{c})}$$

by choosing $\boldsymbol{\lambda}_\mathsf{a} = \boldsymbol{\lambda}_\mathsf{a}^*$. This then implies that for $\boldsymbol{\lambda}_\mathsf{a} = \boldsymbol{\lambda}_\mathsf{a}^*$

$$\mathfrak{D}\left(\frac{\mathsf{f}_\mathsf{a}(\boldsymbol{\theta}_\mathsf{a}) \prod_{\mathsf{c}\neq\mathsf{a}} \mathsf{g}_{\mathsf{c},\boldsymbol{\lambda}_\mathsf{c}}(\boldsymbol{\theta}_\mathsf{c})}{\int_\Theta \mathsf{f}_\mathsf{a}(\boldsymbol{\theta}_\mathsf{a}) \prod_{\mathsf{c}\neq\mathsf{a}} \mathsf{g}_{\mathsf{c},\boldsymbol{\lambda}_\mathsf{c}}(\boldsymbol{\theta}_\mathsf{c})} \middle\| \frac{\prod_\mathsf{c} \mathsf{g}_{\mathsf{c},\boldsymbol{\lambda}_\mathsf{c}}(\boldsymbol{\theta}_\mathsf{c})}{\int_\Theta \prod_\mathsf{c} \mathsf{g}_{\mathsf{c},\boldsymbol{\lambda}_\mathsf{c}}(\boldsymbol{\theta}_\mathsf{c})} \right) = 0$$

Due to the fact, then, that the Kullback Leibler distance is non-negative definite, $\boldsymbol{\lambda}_\mathsf{a} = \boldsymbol{\lambda}_\mathsf{a}^*$ must be the global minima of the refinement step (2.3). Thus, as soon as

we have refined a approximating factor $\mathsf{g}_{\mathsf{a},\boldsymbol{\lambda}_\mathsf{a}}$'s parameters $\boldsymbol{\lambda}_\mathsf{a}$ we have $\mathsf{g}_{\mathsf{a},\boldsymbol{\lambda}_\mathsf{a}} = \mathsf{f}_\mathsf{a}$, which implies that after all of the factors have been refined at least once the aggregate approximating density matches the true density. ■

For the parallel scheduling, for instance, this occurs after only one iteration, and for the cyclic serial iteration this occurs after a single iterative cycle.

Ex. 1 (Application to Turbo Decoding)**:** A particularly simple case in which many authors have noted (see, e.g. [32, 36, 37]) that the turbo decoder converges to the true a posteriori probabilities occurs when the component decoder likelihood functions factor into the product of functions of the individual bits. In this case, the likelihood function for the first decoder $\mathsf{p}_{\mathbf{r}_s,\mathbf{r}_0|\boldsymbol{\xi}}$ is proportional to an exponential family density (see Appendix D) whose sufficient statistics $\mathbf{t}(\mathbf{x}) := \mathbf{x}$ are the message bits themselves. The likelihood function for the other component decoder admits the same form. Of course, the approximating function has the same sufficient statistics as the factors in this instance, and by Proposition 2, the turbo decoder will converge after a single iteration to the correct a posteriori probabilities. This happens, for instance, if the component codes are repetition codes, because then the component likelihood functions admit the necessary factorization.

3.2 Cycle-free Dependence

More generally, the sufficient statistics $\mathbf{t}_\mathsf{a}(\boldsymbol{\theta}_\mathsf{a})$ for the approximating density $\mathsf{g}_{\mathsf{a},\boldsymbol{\lambda}_\mathsf{a}}$ are also sufficient statistics for the factors f_a so that $\mathsf{f}_\mathsf{a}(\boldsymbol{\theta}_\mathsf{a}) = \hat{\mathsf{f}}_\mathsf{a}(\mathbf{t}_\mathsf{a}(\boldsymbol{\theta}_\mathsf{a}))\ \forall \boldsymbol{\theta}_\mathsf{a} \in \Theta_\mathsf{a}$, with the factors f_a not necessarily being members of the minimal standard exponential families. This holds for our setup because we assumed in it Section 2.2.

The special case of interest occurs when the statistics factor graph that we constructed in Section 2.2 has no loops.

Def. 2 (Cycle-free)**:** A factor graph is cycle-free if there does not exist a sequence of distinct connected edges $\{e_1, \ldots, e_L\}$ which connect a node with itself. In other words, if there do not exist $\{e_1, \ldots, e_L\}$ with $e_i \neq e_j \forall i, j$ and for all i $e_i = \{c, a\}$, $e_{i+1} = \{j, a\}$ for some c, a, j and with $e_1 = \{c, a\}$ and $e_L = \{c, j\}$.

Of course, a graph can be cycle-free, but not be connected. To specify whether or not the cycle-free graph is connected it is customary ([38] pp. 5) to call a single connected component within a cycle-free graph a tree, and then a cycle-free graph is referred to as a forest ([38] pp. 6).

We are now ready to prove the following theorem.

Thm. 1 (Cycle-Free Optimality)**:** If

1. The sufficiency assumption As. 1 and the reciprocity assumption As. 2 hold.

2. The marginal density

$$\frac{\int_{\mathbf{t} \setminus \mathbf{v}_j} \hat{f}_a(\mathbf{t}_a) \prod_{c \neq a} \hat{g}_{c, \boldsymbol{\lambda}_c}(\mathbf{t}_c) \mu(\mathbf{t}) d\mathbf{t}}{\int_{\mathbf{t}} \hat{f}_a(\mathbf{t}_a) \prod_{c \neq a} \hat{g}_{c, \boldsymbol{\lambda}_c}(\mathbf{t}_c) \mu(\mathbf{t}) d\mathbf{t}}$$

   is a minimal standard exponential family with sufficient statistics $\mathbf{v}_j$ for all $\mathbf{v}_j \in \mathbf{t}_a$ and for all $a \in \{1, \ldots, M\}$.

3. The statistics factor graph is cycle-free (i.e. is a forest).

Then, after a finite number of iterations, the aggregate approximate density $\prod_a g_{a, \boldsymbol{\lambda}_a}(\boldsymbol{\theta})$ gives the same expected values of the sufficient statistics $\mathbb{E}[\mathbf{t}(\boldsymbol{\theta})]$ as the true density $\prod_a f_a(\boldsymbol{\theta})$. Furthermore, the marginal distributions for the statistics vectors $\mathbf{v}_j$ derived

from the aggregate approximate density match the marginal distributions of the exact density, so that

$$\frac{\int_{\mathbf{t}\setminus\mathbf{v}_j}\prod_a \hat{g}_{a,\boldsymbol{\lambda}_a}(\mathbf{t}_a)\mu(\mathbf{t})d\mathbf{t}}{\int_{\mathbf{t}}\prod_a \hat{g}_{a,\boldsymbol{\lambda}_a}(\mathbf{t}_a)\mu(\mathbf{t})d\mathbf{t}} = \frac{\int_{\mathbf{t}\setminus\mathbf{v}_j}\prod_a \hat{f}_{a,\boldsymbol{\lambda}_a}(\mathbf{t}_a)\mu(\mathbf{t})d\mathbf{t}}{\int_{\mathbf{t}}\prod_a \hat{f}_{a,\boldsymbol{\lambda}_a}(\mathbf{t}_a)\mu(\mathbf{t})d\mathbf{t}}$$

Proof: This is not as much a theorem as it is just a reshuffling of calculations. We mimic the construction used in [14] to prove the same result for the belief propagation algorithm in the form of the more abstract sum product algorithm. Construct the statistics factor graph. Select an elementary statistics vector $\mathbf{v}_i$ to verify that expectation propagation calculated the a posteriori density for correctly. Redraw the statistics factor graph as a tree rooted at $\mathbf{v}_i$ and descending the page vertically. At each factor node, cut the edge between it and its parent elementary statistics vector node and insert an integration operation over all of the statistics that are children of that factor node. This integration integrates the function passed to it from the its child (the factor node). Each elementary statistics vector $\mathbf{v}_j$ passes to its parent any function passed to it from its children multiplied by the change of variable ($\boldsymbol{\theta} \rightarrow \mathbf{t}(\boldsymbol{\theta})$) Radon Nikodym derivative component from the reciprocity assumption As. 2 $\mu_j(\mathbf{v}_i)$. Starting at $\mathbf{v}_i$ and working our way down the tree, we write out the expression for the marginal a posterior distribution for $\mathbf{v}_i$ (multiplied by a scalar which is removed if we divide by the expression's integral over $\mathbf{v}_i$). But under the assumptions of the theorem the messages we have passed in this algorithm are just the densities associated with the parameter messages passed in the expectation propagation algorithm (after a finite number of "garbage" steps as the correct messages work their way up the tree to the node of interest). ■

In addition to the example applications of this theory provided below, [14] used

the version of this theorem for belief propagation to elegantly prove convergence of the Kalman filter[39] for linear (least squares) estimation of signals with known linear dynamics, as well as the forward backward algorithm [12] for the decoding of convolutional codes.

Ex. 2 (Application to LDPC Decoding)**:** The Tanner graph (see [40] pp. 139) for a LDPC code is the bipartite graph whose adjacency matrix is derived from the parity check matrix. Thus, there are left nodes, which are the transmitted bits output from the error control coder, and there are right nodes, which correspond to the parity check equations. If this graph is cycle-free, then Gallager's algorithm for the soft decoding of LDPC codes will converge in a finite number of steps to the bitwise a posteriori probabilities for the coded bits. This is because the factor graph which expectation propagation is running on contains a component, corresponding to the parity check equations, that is the Tanner graph, and the remaining factor nodes have degree one. If the code is also systematic, then we will have the bitwise a posteriori probabilities for the message bits.

Chapter 4

Generic Optimality Frameworks

In this section we discuss two constrained optimization problems which yield the expectation propagation stationary points as their interior critical points. The first one, called statistics based Bethe free energy minimization, discussed in Section 4.1 where we generalize its version from belief propagation to general expectation propagation, has an objective function which is only an approximation of the true function we wish to minimize. That inspires us in Section 4.2 to develop a novel intuitive constrained maximum likelihood optimization problem which yields the expectation propagation stationary points as its critical points. We then apply the constrained maximum likelihood optimization framework to particular instances of expectation propagation, the turbo decoder and the belief propagation decoder, in order to illuminate the theory. We conclude the chapter by connecting the two optimization frameworks with a pseudo-duality framework. The reader not familiar with nonlinear programming or the calculus of variations should read Appendix C before tackling this chapter.

4.1 Free Energy Minimization

We begin by reviewing some important quantities in information theory and statistical thermodynamics. Given some equilibrium distribution with pdf $\mathsf{p}(\boldsymbol{\theta})$ on $\boldsymbol{\theta}$, one defines the *energy* associated with a particular value of $\boldsymbol{\theta}$ to be

$$\mathsf{E}(\boldsymbol{\theta}) := -\log\left(\mathsf{p}(\boldsymbol{\theta})\right)$$

Table 4.1: Notation introduced in the context of free energy minimization.

E	energy function of parameters
$\mathfrak{U}$	average energy functional of distribution on parameters
$\mathfrak{E}$	entropy functional of distribution on parameters
$\mathfrak{G}$	free energy functional of distribution on parameters
p	reference distribution
q	a pdf
$\mathcal{R}$	a region in a statistics factor graph
$\mathscr{R}$	a collection of regions
$\mathcal{N}$	neighbors of a node in the statistics factor graph
$\mathbf{q}$	a collection of several pdfs
μ	(Lagrange) multiplier
$\boldsymbol{\mu}$	vector of multipliers
P	number of constraints
L	Lagrangian of the statistics based Bethe free energy minimization
a.e.	almost everywhere (term from measure theory)

Given a particular distribution on $\boldsymbol{\theta}$ with pdf $\mathsf{q}(\boldsymbol{\theta})$, we then are able to define the average energy of this distribution as

$$\mathfrak{U}(\mathsf{q}) := \int_{\Theta} \mathsf{q}(\boldsymbol{\theta}) \mathsf{E}(\boldsymbol{\theta}) \mathsf{d}\boldsymbol{\theta}$$

The entropy of a distribution is defined as

$$\mathfrak{E}(\mathsf{q}) := -\int_{\Theta} \mathsf{q}(\boldsymbol{\theta}) \log\left(\mathsf{q}(\boldsymbol{\theta})\right) \mathsf{d}\boldsymbol{\theta}$$

The *free energy* of a distribution is then its average energy minus its entropy

$$\mathfrak{G}(\mathsf{q}) := \mathfrak{U}(\mathsf{q}) - \mathfrak{E}(\mathsf{q})$$

which allows us to identify the free energy as the Kullback Leibler divergence between q and the equilibrium distribution p

$$\mathfrak{D}(\mathsf{q}||\mathsf{p}) := \int_{\Theta} \mathsf{q}(\boldsymbol{\theta}) \log\left(\frac{\mathsf{q}(\boldsymbol{\theta})}{\mathsf{p}(\boldsymbol{\theta})}\right) \mathsf{d}\boldsymbol{\theta} = \mathfrak{U}(\mathsf{q}) - \mathfrak{E}(\mathsf{q}) = \mathfrak{G}(\mathsf{q})$$

Owing to the convexity of the Kullback Leibler divergence in its first argument, the free energy is convex, so that for any two distributions q_0 and q_1 and $\lambda \in [0, 1]$, we have

$$\mathfrak{G}(\lambda \mathsf{q}_0 + (1-\lambda)\mathsf{q}_1) \leq \lambda \mathfrak{G}(\mathsf{q}_0) + (1-\lambda)\mathfrak{G}(\mathsf{q}_1)$$

Furthermore, the free energy is non-negative definite with equality to zero if and only if the argument distribution is equal to the equilibrium distribution $\mathsf{q} = \mathsf{p}$ a.e. q.

$$\mathfrak{G}(\mathsf{q}) \geq 0, \mathfrak{G}(\mathsf{q}) = 0 \iff \mathsf{q} = \mathsf{p} \text{ a.e. } \mathsf{q}$$

This motivates using free energy minimization to determine the equilibrium distribution, since we know that the equilibrium distribution is a global minimum over all distributions of the free energy.

Unfortunately, exact free energy minimization often has computational complexity that is too high, and so we must settle for minimizing approximations to the free energy. One class of approximations discussed in [35] are *region based approximations*, which are relevant in a context where the parameter space Θ is finite. We generalize that idea here, allowing Θ to possibly be uncountably infinite and for explicitly structured consistency constraints. We will be working with the parameter factor graph and under the sufficiency assumption As. 1 and reciprocity assumption As. 2. The parameter factor graph, of course, is the bipartite graph whose "left" nodes $\mathcal{V}$ are the parameters $\theta_{\mathtt{i}} \in \mathcal{V} := \{\theta_{\mathtt{i}} | \mathtt{i} \in \{1, \ldots, \mathtt{N}\}\}$ and whose "right" nodes $\mathcal{F}$ are the factors $\mathsf{f}_{\mathtt{a}} \in \{\mathsf{f}_{\mathtt{a}} | \mathtt{a} \in \{1, \ldots, \mathtt{M}\}\}$. An edge connects $\theta_{\mathtt{i}}$ and $\mathsf{f}_{\mathtt{a}}$ if $\mathsf{f}_{\mathtt{a}}$ has $\theta_{\mathtt{i}}$ as an argument, or equivalently if $\theta_{\mathtt{i}} \in \boldsymbol{\theta}_{\mathtt{a}}$. An example parameter factor graph is depicted in Figure 4.1. Note that an edge occurs in the parameter factor graph between $\theta_{\mathtt{i}}$ and $\mathsf{f}_{\mathtt{a}}$ if and only if there is an elementary statistics vector $\mathbf{v}_{\mathtt{j}}$ in the parameter statistics factor graph with an edge between $\theta_{\mathtt{i}}$ and $\mathbf{v}_{\mathtt{j}}$ (i.e. $\theta_{\mathtt{i}} \in \mathbf{v}_{\mathtt{j}}$) as well as an edge between $\mathbf{v}_{\mathtt{j}}$ and $\mathsf{f}_{\mathtt{a}}$ (i.e. $\mathbf{v}_{\mathtt{j}} \subseteq \mathbf{t}_{\mathtt{a}}$).

A *region* $\mathcal{R}$ of the parameter factor graph is defined as the ordered pair $\mathcal{R} := (\mathcal{V}_{\mathcal{R}}, \mathcal{F}_{\mathcal{R}})$ of a set $\mathcal{V}_{\mathcal{R}}$ of parameter nodes and a set of factor nodes $\mathcal{F}_{\mathcal{R}}$ such that if a factor node $\mathsf{f}_{\mathtt{a}}$ belongs to $\mathcal{F}_{\mathcal{R}}$ then all of the parameter nodes connected to that factor node are in $\mathcal{V}_{\mathcal{R}}$. Given a region $\mathcal{R}$, we can then define a region based energy

$$\mathsf{E}_{\mathcal{R}}(\boldsymbol{\theta}_{\mathcal{R}}) \quad := \quad -\sum_{\mathtt{a} \in \mathcal{F}_{\mathcal{R}}} \ln\left(\mathsf{f}_{\mathtt{a}}(\boldsymbol{\theta}_{\mathtt{a}})\right) \tag{4.1}$$

where $\boldsymbol{\theta}_{\mathcal{R}}$ is the vector formed by concatenating the elements of $\boldsymbol{\theta}_{\mathtt{a}}$ for each $\mathtt{a} \in \mathcal{F}_{\mathcal{R}}$, or equivalently the vector whose elements are the set $\mathcal{V}_{\mathcal{R}}$. Then, given a distribution on the variable nodes within the region, $\mathsf{q}_{\mathcal{R}}$, we can define a region based average

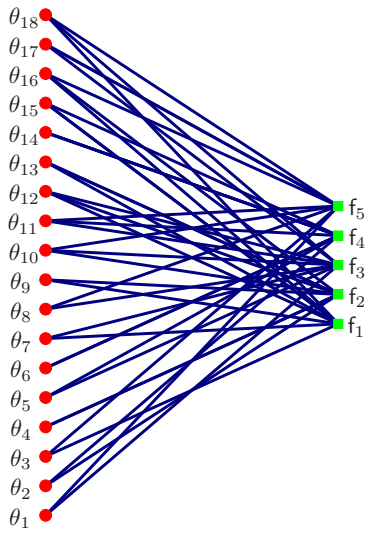

Figure 4.1: An example of a parameter factor graph.

energy

$$\mathfrak{U}_{\mathcal{R}}(\mathsf{q}_{\mathcal{R}}) \;:=\; \int_{\Theta_{\mathcal{R}}} \mathsf{q}_{\mathcal{R}}(\boldsymbol{\theta}_{\mathcal{R}})\mathsf{E}_{\mathcal{R}}(\boldsymbol{\theta}_{\mathcal{R}})\mathsf{d}\boldsymbol{\theta}_{\mathcal{R}} \tag{4.2}$$

where $\Theta_{\mathcal{R}} := \Theta_{\mathtt{i}_1} \times \cdots \times \Theta_{\mathtt{i}_{|\mathcal{V}_{\mathcal{R}}|}}$ is the subset of the parameter space associated with the parameters in $\boldsymbol{\theta}_{\mathcal{R}} = \left[\theta_{\mathtt{i}_1}, \ldots, \theta_{\mathtt{i}_{|\mathcal{V}_{\mathcal{R}}|}}\right]$.

We can also define a region based entropy

$$\mathfrak{E}_{\mathcal{R}}(\mathsf{q}_{\mathcal{R}}) := -\int_{\Theta_{\mathcal{R}}} \mathsf{q}_{\mathcal{R}}(\boldsymbol{\theta}_{\mathcal{R}}) \ln\left(\mathsf{q}_{\mathcal{R}}(\boldsymbol{\theta}_{\mathcal{R}})\right) \mathsf{d}\boldsymbol{\theta}_{\mathcal{R}}$$

The idea now is to use a collection $\mathscr{R}$ of regions to cover the original statistics factor graph, cleverly summing up the region based entropies and region based average energies so as to hopefully well-approximate the entropy of the graph. Part of summing the regions cleverly can be summarized by requiring that each parameter node and each factor node only gets counted once via the use of counting

numbers. In particular one forms the approximate entropy

$$\mathfrak{E}_{\mathscr{R}}(\mathbf{q}_{\mathscr{R}}) := \sum_{\mathcal{R}\in\mathscr{R}} \mathsf{c}_{\mathcal{R}}\mathfrak{E}_{\mathcal{R}}(\mathsf{q}_{\mathcal{R}})$$

the approximate average energy

$$\mathfrak{U}_{\mathscr{R}}(\mathbf{q}_{\mathscr{R}}) := \sum_{\mathcal{R}\in\mathscr{R}} \mathsf{c}_{\mathcal{R}}\mathfrak{U}_{\mathcal{R}}(\mathsf{q}_{\mathcal{R}})$$

and the approximate free energy

$$\mathfrak{G}_{\mathscr{R}}(\mathbf{q}_{\mathscr{R}}) := \mathfrak{U}_{\mathscr{R}}(\mathbf{q}_{\mathscr{R}}) - \mathfrak{E}_{\mathscr{R}}(\mathbf{q}_{\mathscr{R}}) \tag{4.3}$$

where the counting numbers $\mathsf{c}_{\mathcal{R}}$ are chosen so that each node only gets counted once, so that

$$\sum_{\mathcal{R}\in\mathscr{R}} \mathsf{c}_{\mathcal{R}}\mathbf{1}_{\mathsf{i}\in\mathcal{R}} = 1 \ \forall \mathsf{i} \in \mathcal{F}\bigcup\mathcal{V}$$

and where we denoted for notational simplicity $\mathbf{q}_{\mathscr{R}}$ to be the collection of densities defined on $\mathcal{R} \in \mathscr{R}$

$$\mathbf{q}_{\mathscr{R}} := [\mathsf{q}_{\mathcal{R}} | \mathcal{R} \in \mathscr{R}]$$

Of course, we will need to enforce that the $\mathsf{q}_{\mathcal{R}}$'s are probability density functions

$$\int_{\Theta_{\mathcal{R}}} \mathsf{q}_{\mathcal{R}}\mathsf{d}\boldsymbol{\theta}_{\mathcal{R}} = 1, \quad \mathsf{q}_{\mathcal{R}}(\boldsymbol{\theta}_{\mathcal{R}}) \geq 0 \ \ \forall\boldsymbol{\theta}_{\mathcal{R}} \in \Theta_{\mathcal{R}}, \quad \forall\mathcal{R} \in \mathscr{R} \tag{4.4}$$

It will also be necessary to enforce some sense of consistency between the $\mathsf{q}_{\mathcal{R}}$'s, because they are supposed to be approximating marginal densities of some single joint density $\mathsf{q}(\boldsymbol{\theta})$. The sense of consistency we will use in the following development is consistency in the means of the statistics common to different regions.

$$\mathbb{E}_{\mathsf{q}_{\mathcal{R}_1}}\left[\mathsf{t}_{\mathsf{i}}(\boldsymbol{\theta}_{\mathbf{v},\mathsf{i}})\right] = \mathbb{E}_{\mathsf{q}_{\mathcal{R}_2}}\left[\mathsf{t}_{\mathsf{i}}(\boldsymbol{\theta}_{\mathbf{v},\mathsf{i}})\right] \quad \forall\mathcal{R}_1,\mathcal{R}_2 \in \mathscr{R}, \ \forall\mathsf{t}_{\mathsf{i}} \in \mathcal{SN}(\mathcal{V}_{\mathcal{R}_1})\bigcap\mathcal{SN}(\mathcal{V}_{\mathcal{R}_2}) \tag{4.5}$$

where $\mathcal{SN}(\mathcal{A})$ for $\mathcal{A} \subset \mathcal{V}$ is the set of statistics t_{i} contained in the elementary statistics vectors $\mathbf{v}_{\mathsf{j}}$ which share an edge with a parameter in $\mathcal{A}$ in the parameter

statistics graph. Equivalently, $\mathcal{SN}(\mathcal{A})$ are the set of statistics that depend on (i.e. have as an argument) a parameter $\theta_{\mathtt{i}} \in \mathcal{A}$. It is thus through these consistency constraints that the "statistics" part of the parameter statistics factor graph are relevant.

We call the free energy approximation (4.3) together with the consistency constraints (4.5) and density constraints (4.4) a *statistics based approximation* to distinguish it from the region based approximations that inspired it which had different consistency constraints.

The particular statistics based approximation we will consider may be related to the one proposed by Hans Bethe in the context of studying the Curie temperature in ferromagnets (see [35] for the relevant references). One selects

$$\mathscr{R}_{\text{Bethe}} := \{\{\boldsymbol{\theta}_{\mathtt{v},\mathtt{i}}\} | \, \mathtt{i} \in \{1, \ldots, \mathtt{V}\}\} \bigcup \{\{\mathsf{f}_{\mathtt{a}}\} \cup \{\boldsymbol{\theta}_{\mathtt{v},\mathtt{i}} | \, \boldsymbol{\theta}_{\mathtt{v},\mathtt{i}} \subseteq \boldsymbol{\theta}_{\mathtt{a}}\}\}$$

in order to define

$$\mathfrak{E}_{\text{Bethe}}\left(\mathbf{q}_{\mathscr{R}_{\text{Bethe}}}\right) := \sum_{\mathtt{i}=1}^{\mathtt{V}} (1 - |\mathcal{N}(\mathtt{i})|) \mathfrak{E}_{\mathtt{v},\mathtt{i}}(\mathsf{q}_{\mathtt{v},\mathtt{i}}) + \sum_{\mathtt{a}=1}^{\mathtt{N}} \mathfrak{E}_{\mathtt{a}}(\mathsf{q}_{\mathtt{a}})$$

where

$$\mathfrak{E}_{\mathtt{v},\mathtt{i}}(\mathsf{q}_{\mathtt{v},\mathtt{i}}) := -\int_{\Theta_{\mathtt{v},\mathtt{i}}} \mathsf{q}_{\mathtt{v},\mathtt{i}}(\boldsymbol{\theta}_{\mathtt{v},\mathtt{i}}) \log\left(\mathsf{q}_{\mathtt{v},\mathtt{i}}(\boldsymbol{\theta}_{\mathtt{v},\mathtt{i}})\right) \mathsf{d}\boldsymbol{\theta}_{\mathtt{v},\mathtt{i}}$$

and

$$\mathfrak{E}_{\mathtt{a}}(\mathsf{q}_{\mathtt{a}}) := -\int_{\Theta_{\mathtt{a}}} \mathsf{q}_{\mathtt{a}}(\boldsymbol{\theta}_{\mathtt{a}}) \log\left(\mathsf{q}_{\mathtt{a}}(\boldsymbol{\theta}_{\mathtt{a}})\right) \mathsf{d}\boldsymbol{\theta}_{\mathtt{a}}$$

and $\mathcal{N}(\mathtt{i})$ are the factor nodes which neighbor the elementary statistics vector containing $\mathtt{i}$. The corresponding approximate average energy is

$$\mathfrak{U}_{\text{Bethe}}\left(\mathbf{q}_{\mathscr{R}_{\text{Bethe}}}\right) := -\sum_{\mathtt{a} \in \mathcal{F}} \int_{\Theta_{\mathtt{a}}} \mathsf{q}_{\mathtt{a}}(\boldsymbol{\theta}_{\mathtt{a}}) \ln\left(\mathsf{f}_{\mathtt{a}}(\boldsymbol{\theta}_{\mathtt{a}})\right) \mathsf{d}\boldsymbol{\theta}_{\mathtt{a}}$$

which then, of course, yields an approximate free energy of

$$\mathfrak{G}_{\text{Bethe}}\left(\mathbf{q}_{\mathscr{R}_{\text{Bethe}}}\right) := \mathfrak{U}_{\text{Bethe}}\left(\mathbf{q}_{\mathscr{R}_{\text{Bethe}}}\right) - \mathfrak{E}_{\text{Bethe}}\left(\mathbf{q}_{\mathscr{R}_{\text{Bethe}}}\right) \tag{4.6}$$

It turns out that the expectation propagation algorithm under the sufficiency assumption As. 1 and the reciprocity assumption As. 2 is intimately related to the Bethe approximation to the free energy together with the statistics consistency constraints (4.5), as we show with the following theorem.

Thm. 2 (Statistics Based Bethe Free Energy Minimization)**:** The stationary points of the expectation propagation algorithm are interior critical points of the Lagrangian[1] for the optimization problem

$$\mathbf{q}^{*}_{\mathscr{R}_{\text{Bethe}}} = \arg \min_{\mathbf{q}_{\mathscr{R}_{\text{Bethe}}} \in \mathcal{Z}} \mathfrak{G}_{\text{Bethe}} \left(\mathbf{q}_{\mathscr{R}_{\text{Bethe}}} \right)$$

where the constraint set $\mathcal{Z}$ requires that the candidate q_{a}s and q_{i}s must be probability density functions and obey the consistency constraints.

$$\mathcal{Z} := \left\{ \mathbf{q}_{\mathscr{R}_{\text{Bethe}}} \;\middle|\; \begin{array}{c} \mathsf{q}_{\mathsf{a}}(\boldsymbol{\theta}_{\mathsf{a}}) \geq 0 \;\forall \boldsymbol{\theta}_{\mathsf{a}} \in \Theta_{\mathsf{a}} \\ \mathsf{q}_{\mathbf{v},\mathsf{i}}(\boldsymbol{\theta}_{\mathbf{v},\mathsf{i}}) \geq 0 \;\forall \boldsymbol{\theta}_{\mathbf{v},\mathsf{i}} \in \Theta_{\mathbf{v},\mathsf{i}} \\ \int_{\Theta_{\mathsf{a}}} \mathsf{q}_{\mathsf{a}}(\boldsymbol{\theta}_{\mathsf{a}}) \mathrm{d}\boldsymbol{\theta}_{\mathsf{a}} = 1 \\ \int_{\Theta_{\mathbf{v},\mathsf{i}}} \mathsf{q}_{\mathbf{v},\mathsf{i}}(\boldsymbol{\theta}_{\mathbf{v},\mathsf{i}}) \mathrm{d}\boldsymbol{\theta}_{\mathbf{v},\mathsf{i}} = 1 \\ \mathbb{E}_{\mathsf{q}_{\mathsf{a}}} \left[\mathbf{v}_{\mathsf{i}}(\boldsymbol{\theta}_{\mathbf{v},\mathsf{i}}) \right] = \mathbb{E}_{\mathsf{q}_{\mathbf{v},\mathsf{i}}} \left[\mathbf{v}_{\mathsf{i}}(\boldsymbol{\theta}_{\mathbf{v},\mathsf{i}}) \right] \end{array} \right\}$$

Here by *critical point*, we mean a location $\mathbf{q}_{\mathscr{R}_{\text{Bethe}}}$ where the variation [41] of the Lagrangian is zero. The Lagrangian function is the objective function (i.e. the function we wish to minimize) plus the constraints, each multiplied by a separate real number called a Lagrange multiplier. Note that the variation of the Lagrangian being equal to zero is a necessary condition for an extrema of the constrained free

[1] Here, we use *Lagrangian* to mean the objective function(al) plus the constraint function(al)s weighted by real numbers called the *Lagrange multipliers*. This is a separate meaning from what is occasionally used in the context of the calculus of variations.

energy functional as long as the variation of the constraints are linearly independent (see [41] Thm. 1, pp. 43 for the one constraint version). Note that for the purposes of our discussion, it will suffice for the reader not familiar with the calculus of variations to consider the variation to be like a partial derivative and the integral to be like a sum. We will assume interior critical points, so that we consider $\mathbf{q}_{\mathscr{R}_{\text{Bethe}}}$s such that

$$\mathsf{q}_{\mathsf{a}}(\boldsymbol{\theta}_{\mathsf{a}}) > 0 \; \forall \boldsymbol{\theta}_{\mathsf{a}} \in \Theta_{\mathsf{a}} \tag{4.7}$$

and

$$\mathsf{q}_{\mathbf{v},\mathsf{i}}(\boldsymbol{\theta}_{\mathbf{v},\mathsf{i}}) > 0 \; \forall \boldsymbol{\theta}_{\mathbf{v},\mathsf{i}} \in \Theta_{\mathbf{v},\mathsf{i}} \tag{4.8}$$

Proof: Since with (4.7) and (4.8) we are assuming that the inequality constraints are inactive, their Lagrange multipliers must all be zero, and thus our Lagrangian L will contain only the objective function and the equality constraints multiplied by the Lagrange multipliers, which we stack into a vector $\boldsymbol{\mu}$ for brevity.

$$\begin{aligned}
\mathsf{L} \;:=\; & \mathfrak{G}_{\text{Bethe}}(\mathbf{q}_{\mathscr{R}_{\text{Bethe}}}) - \sum_{\mathsf{a}=1}^{\mathsf{N}} \mu_{\mathsf{a}} \left(\int_{\Theta_{\mathsf{a}}} \mathsf{q}_{\mathsf{a}}(\boldsymbol{\theta}_{\mathsf{a}}) \mathsf{d}\boldsymbol{\theta}_{\mathsf{a}} - 1 \right) \\
& - \sum_{\mathsf{i}=1}^{\mathsf{V}} \mu_{\mathbf{v},\mathsf{i}} \left(\int_{\Theta_{\mathbf{v},\mathsf{i}}} \mathsf{q}_{\mathbf{v},\mathsf{i}}(\boldsymbol{\theta}_{\mathbf{v},\mathsf{i}}) \mathsf{d}\boldsymbol{\theta}_{\mathbf{v},\mathsf{i}} - 1 \right) \\
& - \sum_{\mathsf{a}=1}^{\mathsf{M}} \sum_{\mathsf{i} \in \mathcal{EN}(\mathsf{a})} \boldsymbol{\mu}_{\mathsf{a},\mathsf{i}} \cdot \left(\mathbb{E}_{\mathsf{q}_{\mathsf{a}}} \left[\mathbf{v}_{\mathsf{i}}(\boldsymbol{\theta}_{\mathbf{v},\mathsf{i}}) \right] - \mathbb{E}_{\mathsf{q}_{\mathbf{v},\mathsf{i}}} \left[\mathbf{v}_{\mathsf{i}}(\boldsymbol{\theta}_{\mathbf{v},\mathsf{i}}) \right] \right)
\end{aligned} \tag{4.9}$$

The variation of the Lagrangian with respect to q_{a} is

$$\frac{\delta \mathsf{L}}{\delta \mathsf{q}_{\mathsf{a}}(\boldsymbol{\theta}_{\mathsf{a}})} = -\ln\left(\mathsf{f}_{\mathsf{a}}(\boldsymbol{\theta}_{\mathsf{a}})\right) + \ln\left(\mathsf{q}_{\mathsf{a}}(\boldsymbol{\theta}_{\mathsf{a}})\right) + 1 - \mu_{\mathsf{a}} - \sum_{\mathsf{i} \in \mathcal{EN}(\mathsf{a})} \boldsymbol{\mu}_{\mathsf{a},\mathsf{i}} \cdot \mathbf{v}_{\mathsf{i}}(\boldsymbol{\theta}_{\mathbf{v},\mathsf{i}}) \tag{4.10}$$

where $\mathcal{EN}(\mathsf{a})$ are the set of indices for the elementary statistics vector nodes sharing an edge in the statistics factor graph with the factor f_{a}. Here we have used $\frac{\delta \mathsf{L}}{\delta \mathsf{q}_{\mathsf{a}}(\boldsymbol{\theta}_{\mathsf{a}})}$ to represent the variation if the reference measure has uncountable support and to represent the normal partial derivative if the reference measure has countable

support. Similarly, the variation of the Lagrangian with respect to $\mathsf{q}_{\mathbf{v},\mathtt{i}}$ is

$$\frac{\delta \mathsf{L}}{\delta \mathsf{q}_{\mathbf{v},\mathtt{i}}(\boldsymbol{\theta}_{\mathbf{v},\mathtt{i}})} = (1 - |\mathcal{N}(\mathtt{i})|)\left(\ln\left(\mathsf{q}_{\mathbf{v},\mathtt{i}}(\boldsymbol{\theta}_{\mathbf{v},\mathtt{i}})\right) + 1\right) - \mu_{\mathbf{v},\mathtt{i}} + \sum_{\mathtt{a}\in\mathcal{N}(\mathtt{i})} \boldsymbol{\mu}_{\mathtt{a},\mathtt{i}} \cdot \mathbf{v}_{\mathtt{i}}(\boldsymbol{\theta}_{\mathbf{v},\mathtt{i}}) \quad (4.11)$$

Setting (4.10) and (4.11) equal to zero and solving for $\mathsf{q}_{\mathtt{a}}(\boldsymbol{\theta}_{\mathtt{a}})$ and $\mathsf{q}_{\mathbf{v},\mathtt{i}}(\boldsymbol{\theta}_{\mathbf{v},\mathtt{i}})$ respectively yields

$$\mathsf{q}_{\mathtt{a}}(\boldsymbol{\theta}_{\mathtt{a}}) = \mathsf{f}_{\mathtt{a}}(\boldsymbol{\theta}_{\mathtt{a}}) \exp\left(\sum_{\mathtt{i}\in\mathcal{EN}(\mathtt{a})} \boldsymbol{\mu}_{\mathtt{a},\mathtt{i}} \cdot \mathbf{v}_{\mathtt{i}}(\boldsymbol{\theta}_{\mathbf{v},\mathtt{i}}) + \mu_{\mathtt{a}} - 1\right) \quad (4.12)$$

and

$$\mathsf{q}_{\mathbf{v},\mathtt{i}}(\boldsymbol{\theta}_{\mathbf{v},\mathtt{i}}) = \exp\left(\frac{\sum_{\mathtt{a}\in\mathcal{N}(\mathtt{i})} \boldsymbol{\mu}_{\mathtt{a},\mathtt{i}} \cdot \mathbf{v}_{\mathtt{i}}(\boldsymbol{\theta}_{\mathbf{v},\mathtt{i}}) - \mu_{\mathbf{v},\mathtt{i}}}{|\mathcal{N}(\mathtt{i})| - 1} - 1\right) \quad (4.13)$$

Now, if we make the substitutions

$$\boldsymbol{\mu}_{\mathtt{a},\mathtt{i}} := \mathbf{n}_{\mathtt{i}\to\mathtt{a}}$$

where the messages $\mathbf{n}_{\mathtt{i}\to\mathtt{a}}$ were defined in (2.25), and if we use $\mu_{\mathbf{v},\mathtt{i}}$s and $\mu_{\mathtt{a}}$s to satisfy the integrate to unity constraints, we see that the consistency constraints (4.5) are equivalent to the stationary points of the expectation propagation refinement equation (2.24). This shows that the EP stationary points are critical points of the Lagrangian for the statistics based Bethe free energy minimization. ■

Note that special cases of this result were proven for turbo decoding in [42] and for belief propagation decoding (detection) in [35, 43, 44, 45]. [42] uses the connection in order to predict the location of the phase transition (i.e. the location of the waterfall) in turbo codes, whereas [45] focuses on attempting to make better approximations to the variational free energy. Note that one of the ways to make these better approximations is to use better approximating sufficient statistics (which is exactly the notion of expectation propagation separate from belief propagation), which turns out to be a pedagogically clean and quick route of exposition and is the one chosen from the beginning in this dissertation.

Table 4.2: Notation introduced in Section 4.2.

$\mathbf{x}$	output of repetition code
$\mathbf{y}$	input to factor nodes
δ	Dirac delta function
$\mathbf{u}$	sufficient statistics for incoming messages at factors
$\boldsymbol{\gamma}$	incoming messages at factors
ϵ	value of constraint
$\mathcal{C}$	constraint set
ρ	radius of small ball
V	volume of ball
L	Lagrangian

4.2 Constrained Maximum Likelihood

In this section, we show how the stationary points of expectation propagation are solutions to a constrained maximum likelihood estimation problem, but first it will be essential to understand how one may find a maximum likelihood or maximum a posteriori estimate by selecting a prior density. In particular, put aside the expectation propagation framework for the moment and simply imagine that one has some observations $\mathbf{r} \in \mathcal{Y}$, some parameters $\boldsymbol{\theta} \in \Theta$ and a model $\mathsf{f}(\mathbf{r}, \boldsymbol{\theta})$ representing the likelihood of observing $\mathbf{r} = \mathbf{r}$ and $\boldsymbol{\theta} = \boldsymbol{\theta}$. In the case that $\boldsymbol{\theta}$ is truly random, the model f will be a joint density function, and if it is instead deterministic or has an unknown probability distribution, the model f will be a probability density function for $\mathbf{r}$ parameterized by $\boldsymbol{\theta}$. In either case, we have the following proposition.

Prop. 3: One can go about determining the maximum a posteriori or maximum

likelihood estimate (when it exists) $\hat{\boldsymbol{\theta}}_{\mathrm{ML}} := \arg\max_{\boldsymbol{\theta}\in\Theta} \mathsf{f}(\mathbf{r},\boldsymbol{\theta})$ by asking for the most likely prior density for $\boldsymbol{\theta}$, that is, by searching for the $\mathsf{q}(\boldsymbol{\theta})$ such that $\mathsf{q}(\boldsymbol{\theta}) \geq 0$ for all $\boldsymbol{\theta} \in \Theta$ and $\int_\Theta \mathsf{q}_{\boldsymbol{\theta}}(\boldsymbol{\theta})\mathrm{d}\boldsymbol{\theta} = 1$ which maximizes the likelihood function

$$\mathsf{q}_{\mathbf{r}|\mathsf{q}_{\boldsymbol{\theta}}}(\mathbf{r}|\mathsf{q}_{\boldsymbol{\theta}}) = \int_\Theta \mathsf{f}(\mathbf{r},\boldsymbol{\theta})\mathsf{q}_{\boldsymbol{\theta}}(\boldsymbol{\theta})\mathrm{d}\boldsymbol{\theta}$$

In particular, if the maximum a posteriori (or likelihood) estimate for $\boldsymbol{\theta}$ exists, then the pdf which maximizes this expression is a δ function (i.e. a Dirac δ if Θ is continuous and a Kronecker δ if Θ is discrete) with all of its probability mass at $\hat{\boldsymbol{\theta}}_{\mathrm{ML}}$

$$\hat{\mathsf{q}}_{\boldsymbol{\theta},\mathrm{ML}}(\boldsymbol{\theta}) := \delta(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}_{\mathrm{ML}})$$

We can then find the maximum a posteriori (or likelihood) estimate by taking the expectation of $\boldsymbol{\theta}$ with respect to the most likely prior density

$$\hat{\boldsymbol{\theta}}_{\mathrm{ML}} = \mathbb{E}_{\mathsf{q}_{\boldsymbol{\theta},\mathrm{ML}}}[\boldsymbol{\theta}]$$

Proof: This is due to the Hölder inequality, which tells us

$$\int_\Theta \mathsf{f}(\mathbf{r},\boldsymbol{\theta})\mathsf{q}_{\boldsymbol{\theta}}(\boldsymbol{\theta})\mathrm{d}\boldsymbol{\theta} \leq \|\mathsf{f}(\mathbf{r},\boldsymbol{\theta})\|_\infty \|\mathsf{q}_{\boldsymbol{\theta}}(\boldsymbol{\theta})\|_1 = \|\mathsf{f}(\mathbf{r},\boldsymbol{\theta})\|_\infty$$

where the last equality follows from the fact that q is a pdf, and $\mathsf{f}(\mathbf{r},\boldsymbol{\theta})$ is being regarded as a function of $\boldsymbol{\theta}$. Because the density which attains the maximum is the δ function, we can then find the maximum a posteriori (or likelihood) estimate by taking the expectation of $\boldsymbol{\theta}$ with respect to the most likely prior density

$$\hat{\boldsymbol{\theta}}_{\mathrm{ML}} = \mathbb{E}_{\mathsf{q}_{\boldsymbol{\theta},\mathrm{ML}}}[\boldsymbol{\theta}]$$

■

Proposition 3 validates the roundabout method of determining the maximum a posteriori/likelihood parameters $\boldsymbol{\theta}$ by first determining the maximum likelihood prior density for $\boldsymbol{\theta}$. It turns out that this roundabout method is related to the method which expectation propagation is using to approximate the maximum likelihood/a posteriori detector/estimate.

Returning now to the expectation propagation setup, we begin by introducing extra parameters $\mathbf{x}_a \in \Theta$ and $\mathbf{y}_a \in \Theta$ representing the outputs of the parameter nodes and the inputs to the statistics nodes to rewrite the joint pdf for the parameters and the received data $\mathbf{r}$ as

$$q_{\mathbf{r},\mathbf{x},\mathbf{y}}(\mathbf{r},\boldsymbol{\theta},\mathbf{x},\mathbf{y}) := \prod_{a=1}^{M} f_a(\mathbf{y}_a)\delta(\mathbf{x}_a - \boldsymbol{\theta}) \prod_{a=1}^{M} \delta(\mathbf{x}_a - \mathbf{y}_a) \tag{4.14}$$

where δ is the Dirac δ distribution if Θ is uncountable and is the Kronecker δ function if Θ is finite. Note that the $\prod_{a=1}^{M} \delta(\mathbf{x}_a - \boldsymbol{\theta})$ is enforcing that all of the $\mathbf{x}_a$'s are equal to the parameters $\boldsymbol{\theta}$ and the factor $\prod_{a=1}^{M} \delta(\mathbf{x}_a - \mathbf{y}_a)$ is enforcing that $\mathbf{x}_a = \mathbf{y}_a$ for all $a \in \{1, \ldots, M\}$. As we just established, asking for a prior density on $(\mathbf{x}, \mathbf{y})$ which maximizes this likelihood function provides a method for determining the maximum likelihood estimate for $\mathbf{x}$ and $\mathbf{y}$ and thus for $\boldsymbol{\theta}$. Suppose, for instance, that we wish to do so, choosing the prior distribution from the set of standard exponential family distributions which have $\mathbf{x}_a$ and $\mathbf{y}_a$ chosen independently from each other with sufficient statistics

Now, suppose that we relaxed the requirement that $\mathbf{x}_a = \mathbf{y}_a$ for all a, by dropping the component of the likelihood function

$$\prod_{a=1}^{M} \delta(\mathbf{x}_a - \mathbf{y}_a)$$

and approximating the joint distribution for $\mathbf{x}, \mathbf{y}$ by a distribution specified by choosing $\mathbf{x}_a$ and $\mathbf{y}_a$ independently according to standard exponential family den-

sities with sufficient statistics $\mathbf{t}_a(\mathbf{x}_a)$ and $\mathbf{u}_a(\mathbf{y}_a) := [\mathbf{t}_c(\mathbf{y}_a)]_{c\neq a}$

$$(\mathbf{x},\mathbf{y}) \sim \prod_{a=1}^{M} \exp\left(\mathbf{t}_a(\mathbf{x}_a)\cdot\boldsymbol{\lambda}_a - \psi_{\mathbf{t}_a}(\boldsymbol{\lambda}_a)\right)\exp\left(\mathbf{u}_a(\mathbf{y}_a)\cdot\boldsymbol{\gamma}_a - \psi_{\mathbf{u}_a}(\boldsymbol{\gamma}_a)\right)$$

thereby approximating the true likelihood function $q_{\mathbf{r},\mathbf{x},\mathbf{y},\boldsymbol{\theta}}(\mathbf{r},\mathbf{x},\mathbf{y},\boldsymbol{\theta})$ defined in (4.14) with

$$q_{\mathbf{r},\mathbf{x},\mathbf{y},\boldsymbol{\theta}}(\mathbf{r},\mathbf{x},\mathbf{y},\boldsymbol{\theta}) \approx \hat{q}_{\mathbf{r},\mathbf{x},\mathbf{y},\boldsymbol{\theta}}(\mathbf{r},\mathbf{x},\mathbf{y},\boldsymbol{\theta}) \tag{4.15}$$

$$\begin{aligned}\hat{q}_{\mathbf{r},\mathbf{x},\mathbf{y},\boldsymbol{\theta}}(\mathbf{r},\mathbf{x},\mathbf{y},\boldsymbol{\theta}) := &\prod_{a=1}^{M} f_a(\mathbf{y}_a)\delta(\mathbf{x}_a-\boldsymbol{\theta})\prod_{a=1}^{M}\exp\left(\mathbf{t}_a(\mathbf{x}_a)\cdot\boldsymbol{\lambda}_a - \psi_{\mathbf{t}_a}(\boldsymbol{\lambda}_a)\right)\\ &\exp\left(\mathbf{u}_a(\mathbf{y}_a)\cdot\boldsymbol{\gamma}_a - \psi_{\mathbf{u}_a}(\boldsymbol{\gamma}_a)\right)\end{aligned} \tag{4.16}$$

This approximation, in turn, yields an approximate likelihood function for $\boldsymbol{\lambda},\boldsymbol{\gamma}$ via

$$\hat{q}_{\mathbf{r}|\boldsymbol{\lambda},\boldsymbol{\gamma}}(\mathbf{r}|\boldsymbol{\lambda},\boldsymbol{\gamma}) := \int_{\Theta}\int_{\Theta^M}\int_{\Theta^M} \hat{q}_{\mathbf{r},\mathbf{x},\mathbf{y},\boldsymbol{\theta}}(\mathbf{r},\mathbf{x},\mathbf{y},\boldsymbol{\theta})\mathrm{d}\mathbf{x}\mathrm{d}\mathbf{y}\mathrm{d}\boldsymbol{\theta} \tag{4.17}$$

Of course, to make this approximation (4.15) which replaces the requirement $\mathbf{x}=\mathbf{y}$ with independence of $\mathbf{x},\mathbf{y}$ accurate, we will have to consider only distributions for $\mathbf{x},\mathbf{y}$ which have a high probability of selecting $\mathbf{x}=\mathbf{y}$. We can control the error introduced by this approximation by constraining the $\boldsymbol{\lambda}$ and $\boldsymbol{\gamma}$ considered to lie within the constraint set described by the equation

$$\mathcal{C}_\epsilon := \left\{ (\boldsymbol{\lambda},\boldsymbol{\gamma}) \;\middle|\; \sum_{a=1}^{M}\log\left(\frac{\int_\Theta \exp(\mathbf{t}_a(\boldsymbol{\theta})\cdot\boldsymbol{\lambda}_a + \mathbf{u}_a(\boldsymbol{\theta})\cdot\boldsymbol{\gamma}_a)\mathrm{d}\boldsymbol{\theta}}{\int_\Theta \exp(\mathbf{t}_a(\mathbf{x}_a)\cdot\boldsymbol{\lambda}_a)\mathrm{d}\mathbf{x}_a \int_\Theta \exp(\mathbf{u}_a(\mathbf{y}_a)\cdot\boldsymbol{\gamma}_a)\mathrm{d}\mathbf{y}_a}\right) = \log(\epsilon) \right\} \tag{4.18}$$

In particular, if Θ is finite, so that the measure we are using is a counting measure, this constraint set may be interpreted as the set of $(\boldsymbol{\lambda},\boldsymbol{\gamma})$ such that the probability that $\mathbf{x}_a = \mathbf{y}_a\ \forall a \in \{1,\ldots,M\}$ is equal to ϵ. Choosing, within the finite context, a constraint value ϵ equal to one, we recover $\mathbf{x}_a = \mathbf{y}_a$ with probability one according to the joint measure on $\mathbf{x}$ and $\mathbf{y}$.

If, on the other hand Θ is uncountably infinite, the constraint set more closely reflects $\mathbf{x} \approx \mathbf{y}$ for large $\epsilon \gg 1$, since the probability that two continuous (independent) random variables are equal is zero. To get a more rigorous handle on what we mean by $\mathbf{x} \approx \mathbf{y}$ consider the probability that $\mathbf{x}$ and $\mathbf{y}$ are no more than ρ apart

$$\begin{aligned}
\mathbb{P}\left[\|\mathbf{x}-\mathbf{y}\| \leq \rho\right] &= \int_{\{(\mathbf{x},\mathbf{y})\in\Theta^2 | \|\mathbf{x}-\mathbf{y}\|\leq\rho\}} \mathsf{q}_{\mathbf{x}}(\mathbf{x})\mathsf{q}_{\mathbf{y}}(\mathbf{y})\mathsf{d}\mathbf{x}\mathsf{d}\mathbf{y} \\
&= \int_{\{(\mathbf{x},\mathbf{y}) | \|\mathbf{x}-\mathbf{y}\|\leq\rho\}} \mathsf{q}_{\mathbf{x}}(\mathbf{x})\left(\mathsf{q}_{\mathbf{y}}(\mathbf{x}) + (\nabla_{\mathbf{y}}\mathsf{q}_{\mathbf{y}})^T(\mathbf{y}-\mathbf{x}) + \mathsf{o}(\rho)\right)\mathsf{d}\mathbf{x}\mathsf{d}\mathbf{y} \\
&= \mathsf{V}(\rho)\left(\int_{\Theta} \mathsf{q}_{\mathbf{x}}(\mathbf{x})\mathsf{q}_{\mathbf{y}}(\mathbf{x})\mathsf{d}\mathbf{x} + \mathsf{o}(\rho)\right) \\
&= \mathsf{V}(\rho)\left(\epsilon + \mathsf{o}(\rho)\right) \qquad (4.19)
\end{aligned}$$

where $\mathsf{V}(\rho)$ is the Lebesgue measure of the ball $\{\mathbf{x} | \|\mathbf{x}\| \leq \rho\}$ and the last equality holds for $(\mathsf{q}_{\mathbf{x}}, \mathsf{q}_{\mathbf{y}}) \in \mathcal{C}_\epsilon$. Thus, making ϵ very large, we are forcing higher probabilities of $\mathbb{P}\left[\|\mathbf{x}-\mathbf{y}\| \leq \rho\right]$ for small ρ.

We now show how expectation propagation is intimately related to maximizing the approximate likelihood function (4.17) subject to the constraints (4.18).

Thm. 3 (Maximum Likelihood Interpretation of Expectation Propagation)**:** The stationary points of expectation propagation solve the first order necessary conditions for the constrained optimization problem

$$(\boldsymbol{\lambda}^*, \boldsymbol{\gamma}^*) := \arg \max_{(\boldsymbol{\lambda},\boldsymbol{\gamma})\in\mathcal{C}_\epsilon} \log(\mathsf{q}_{\mathbf{r}|\boldsymbol{\lambda},\boldsymbol{\gamma}}(\mathbf{r}|\boldsymbol{\lambda},\boldsymbol{\gamma}))$$

subject to a Lagrange multiplier $\mu = -1$.

Proof: The proof of this theorem is logically direct. First one forms the Lagrangian by adding the objective function to the constraining function times a Lagrange multiplier μ.

$$\mathsf{L} := \log\left(\mathsf{q}_{\mathbf{r}|\boldsymbol{\lambda},\boldsymbol{\gamma}}(\mathbf{r}|\boldsymbol{\lambda},\boldsymbol{\gamma})\right) + \mu \log\left(\frac{\int_\Theta \exp(\mathbf{t}_{\mathsf{a}}(\boldsymbol{\theta})\cdot\boldsymbol{\lambda}_{\mathsf{a}} + \mathbf{u}_{\mathsf{a}}(\boldsymbol{\theta})\cdot\boldsymbol{\gamma}_{\mathsf{a}})\mathsf{d}\boldsymbol{\theta}}{\int_\Theta \exp(\mathbf{t}_{\mathsf{a}}(\mathbf{x}_{\mathsf{a}})\cdot\boldsymbol{\lambda}_{\mathsf{a}})\mathsf{d}\mathbf{x}_{\mathsf{a}} \int_\Theta \exp(\mathbf{u}_{\mathsf{a}}(\mathbf{y}_{\mathsf{a}})\cdot\boldsymbol{\gamma}_{\mathsf{a}})\mathsf{d}\mathbf{y}_{\mathsf{a}}}\right)$$

Substituting in the Lagrange multiplier $\mu = -1$ gives

$$\begin{aligned} \mathsf{L} &= \sum_{\mathtt{a}=1}^{\mathtt{M}} \log\left(\int_{\Theta} \mathsf{f}_{\mathtt{a}}(\mathbf{y}_{\mathtt{a}}) \exp\left(\mathbf{u}_{\mathtt{a}}(\mathbf{y}_{\mathtt{a}}) \cdot \boldsymbol{\gamma}_{\mathtt{a}}\right) \mathrm{d}\mathbf{y}_{\mathtt{a}}\right) \\ &\quad + \log\left(\int_{\Theta} \exp\left(\sum_{\mathtt{c}=1}^{\mathtt{M}} \mathbf{t}_{\mathtt{c}}(\boldsymbol{\theta}_{\mathtt{c}}) \cdot \boldsymbol{\lambda}_{\mathtt{c}}\right) \mathrm{d}\boldsymbol{\theta}\right) \\ &\quad - \sum_{\mathtt{a}=1}^{\mathtt{M}} \log\left(\int_{\Theta} \exp\left(\mathbf{t}_{\mathtt{a}}(\boldsymbol{\theta}_{\mathtt{a}}) \cdot \boldsymbol{\lambda}_{\mathtt{a}} + \mathbf{u}_{\mathtt{a}}(\boldsymbol{\theta}_{\mathtt{a}}) \cdot \boldsymbol{\gamma}_{\mathtt{a}}\right) \mathrm{d}\boldsymbol{\theta}_{\mathtt{a}}\right) \end{aligned} \tag{4.20}$$

The first order necessary conditions stipulate that the gradient of the Lagrangian with respect to the parameters must be zero. The gradient of the Lagrangian, subject to $\mu = -1$, in this instance is

$$\begin{aligned} \nabla_{\boldsymbol{\lambda}_{\mathtt{a}}} \mathsf{L} &= \frac{\int_{\Theta} \mathbf{t}_{\mathtt{a}}(\boldsymbol{\theta}_{\mathtt{a}}) \exp\left(\sum_{\mathtt{c}=1}^{\mathtt{M}} \mathbf{t}_{\mathtt{c}}(\boldsymbol{\theta}_{\mathtt{c}}) \cdot \boldsymbol{\lambda}_{\mathtt{c}}\right) \mathrm{d}\boldsymbol{\theta}}{\int_{\Theta} \exp\left(\sum_{\mathtt{c}=1}^{\mathtt{M}} \mathbf{t}_{\mathtt{c}}(\boldsymbol{\theta}_{\mathtt{c}}) \cdot \boldsymbol{\lambda}_{\mathtt{c}}\right) \mathrm{d}\boldsymbol{\theta}} \\ &\quad - \frac{\int_{\Theta} \mathbf{t}_{\mathtt{a}}(\boldsymbol{\theta}) \exp\left(\mathbf{t}_{\mathtt{a}}(\boldsymbol{\theta}) \cdot \boldsymbol{\lambda}_{\mathtt{a}} + \mathbf{u}_{\mathtt{a}}(\boldsymbol{\theta}) \cdot \boldsymbol{\gamma}_{\mathtt{a}}\right) \mathrm{d}\boldsymbol{\theta}}{\int_{\Theta} \exp\left(\mathbf{t}_{\mathtt{a}}(\boldsymbol{\theta}) \cdot \boldsymbol{\lambda}_{\mathtt{a}} + \mathbf{u}_{\mathtt{a}}(\boldsymbol{\theta}) \cdot \boldsymbol{\gamma}_{\mathtt{a}}\right) \mathrm{d}\boldsymbol{\theta}} \end{aligned} \tag{4.21}$$

$$\begin{aligned} \nabla_{\boldsymbol{\gamma}_{\mathtt{a}}} \mathsf{L} &= \frac{\int_{\Theta} \mathbf{u}_{\mathtt{a}}(\mathbf{y}_{\mathtt{a}}) \mathsf{f}_{\mathtt{a}}(\mathbf{y}_{\mathtt{a}}) \exp\left(\mathbf{u}_{\mathtt{a}}(\mathbf{y}_{\mathtt{a}}) \cdot \boldsymbol{\gamma}_{\mathtt{a}}\right) \mathrm{d}\mathbf{x}_{\mathtt{a}}}{\int_{\Theta} \mathsf{f}_{\mathtt{a}}(\mathbf{y}_{\mathtt{a}}) \exp\left(\mathbf{u}_{\mathtt{a}}(\mathbf{y}_{\mathtt{a}}) \cdot \boldsymbol{\gamma}_{\mathtt{a}}\right) \mathrm{d}\mathbf{x}_{\mathtt{a}}} \\ &\quad - \frac{\int_{\Theta} \mathbf{u}_{\mathtt{a}}(\boldsymbol{\theta}) \exp\left(\mathbf{t}_{\mathtt{a}}(\boldsymbol{\theta}) \cdot \boldsymbol{\lambda}_{\mathtt{a}} + \mathbf{u}_{\mathtt{a}}(\boldsymbol{\theta}) \cdot \boldsymbol{\gamma}_{\mathtt{a}}\right) \mathrm{d}\boldsymbol{\theta}}{\int_{\Theta} \exp\left(\mathbf{t}_{\mathtt{a}}(\boldsymbol{\theta}) \cdot \boldsymbol{\lambda}_{\mathtt{a}} + \mathbf{u}_{\mathtt{a}}(\boldsymbol{\theta}) \cdot \boldsymbol{\gamma}_{\mathtt{a}}\right) \mathrm{d}\boldsymbol{\theta}} \end{aligned} \tag{4.22}$$

Subject to the sufficiency assumption As. 1 and to the reciprocity assumption As. 2, $\mathbf{t}_{\mathtt{a}}$ and $\mathbf{u}_{\mathtt{a}}$ may be interchanged in these equations because they are always equal (we discussed this in Section 2.2). Thus, setting (4.21) equal to zero and solving for $\boldsymbol{\gamma}_{\mathtt{a}}$ ensures the equivalence between solving (4.22) set equal to zero for $\boldsymbol{\lambda}_{\mathtt{a}}$ and refining $\boldsymbol{\lambda}_{\mathtt{a}}$ according to the EP refinement rule (2.24). In fact, the parameters $\boldsymbol{\lambda}_{\mathtt{a}}$ can be identified with the messages (2.25) from the factor nodes to the elementary statistics vector nodes and the parameters $\boldsymbol{\gamma}_{\mathtt{a}}$ may be identified with the messages (2.24) from the elementary statistics vector nodes and the factor nodes. Thus, when the parameters are equal to those messages and are such that $\nabla_{\boldsymbol{\lambda}_{\mathtt{a}}} \mathsf{L} = \mathbf{0}$ and $\nabla_{\boldsymbol{\gamma}_{\mathtt{a}}} \mathsf{L} = \mathbf{0}$ for all $\mathtt{a} \in \{1, \ldots, \mathtt{M}\}$, then the messages correspond to a stationary point of expectation propagation. ■

Note that it is rather atypical to set a Lagrange multiplier in a problem, usually one sets a constraint value and then gets a resulting Lagrange multiplier. We will see the theoretical significance of choosing -1 in Section 4.3. Here we address its practical significance in the case that Θ is finite. In particular, choosing the Lagrange multiplier to be -1 ensures that the gradient of the likelihood function under the false independence assumption is equal to the gradient of the constraint at a critical point.

$$\nabla \mathsf{L} = \mathbf{0} \Leftrightarrow \nabla \hat{\mathsf{q}}_{\mathbf{r}|\boldsymbol{\lambda},\boldsymbol{\gamma}}(\mathbf{r}|\boldsymbol{\lambda},\boldsymbol{\gamma}) = \nabla \mathbb{P}\left[\mathbf{x} = \mathbf{y}|\boldsymbol{\lambda},\boldsymbol{\gamma}\right]$$

This means that, to first order, the change in the constraint and the change in the likelihood function are equal as one moves away from a stationary point. In the finite Θ context often one takes decisions on the convergent values of the parameters $(\boldsymbol{\lambda}^*, \boldsymbol{\gamma}^*)$ by selecting a particular $\mathbf{x} = \mathbf{y}$ as the most likely candidate and throwing all other information away (take for instance the binary case where $\boldsymbol{\lambda}$ and $\boldsymbol{\gamma}$ are vectors of log likelihood ratios). Because this forces $\mathbf{x} = \mathbf{y}$ and puts all probability on one value $\mathbf{x} = \mathbf{x}^* = \mathbf{y}$, it is accompanied by a large (in fact, the largest possible, since probabilities must be less than one) increase in the value of the constraint. The Lagrange multiplier being -1, then, ensures that this large increase in the constraint is accompanied by an equivalently large increase in likelihood under the false independence assumption, to first order.

In the next section, we apply the results we developed to the turbo decoder and the section following that applies the results to the belief propagation decoder.

4.2.1 Application to Turbo Decoding

An accurate justification for why the turbo decoding strategy performs as well as it does is still lacking. While it has been proven that turbo codes have good distance

properties, which would be relevant for maximum likelihood decoding, a proper connection of the suboptimal turbo decoder with maximum likelihood decoding has been lacking. This is exacerbated by the fact that the turbo decoder, unlike most of the designs in modern communications systems engineering, was not originally introduced as a solution to an optimization problem. This has made explaining just why the turbo decoder performs as well as it does very difficult. Significant progress has been made with EXIT style analysis [46] and density evolution [47], but these techniques ultimately appeal to results which become valid only when the block length grows rather large. Other attempts, such as connections to factor graphs [14] and belief propagation [33], have been largely unsuccessful at showing convergence due to loops in the turbo coding graph. The information geometric attempts [48], [49], [36], [50], and [51], in turn have been inhibited by an inability to efficiently describe extrinsic information extraction as an information projection.

None of these convergence frameworks, so far, have identified the optimization problem that the decoder is attempting to solve. The connection between the stationary points of belief propagation and the critical points of the constrained Bethe free energy, discussed for various cases in this dissertation in Section 4.1 and in [44, 45, 52, 43, 35] does recognize that belief propagation is related to an approximation in statistical physics which is a constrained optimization, but it is difficult to gain intuition as to why one would like to minimize the objective function, since the Bethe free energy is not exact in factor graphs with loops. In this subsection, we will show that the turbo decoder admits an exact interpretation as attempting to find a solution to a particular intuitively pleasing constrained optimization problem by applying the result we just developed for expectation propagation in Section 4.2. In our formulation of the constrained optimization problem, it will

become clear that the turbo decoder is calculating a constrained maximum likelihood sequence detection. This is to be contrasted to the claims in [44, 45, 52, 43] that belief propagation decoding results in estimates of the maximum likelihood bitwise detection, owing to the fact that the existing proofs of convergence within that arena were for the product density case in which the maximum likelihood sequence detection and the maximum likelihood bitwise detections corresponded. Of course, the constraints change the problem from pure word-wise maximum likelihood detection to something that is generically (i.e. in factor graphs with loops) neither wordwise nor bitwise detection.

After interpreting what Proposition 3 means in the context of decoding, we will apply Theorem 3 to the turbo decoder. Actually, rather than simply copying the theorem to the turbo decoder, we state the same essential idea, using notation native to the turbo decoder. In particular, the parameters in our optimization problem will be the usual parameters used in the turbo decoder. This means that the notation will be slightly different and there will be slight differences in appearances, but fundamentally the theorem remains the same. In order to more deeply understand the constrained maximum likelihood framework developed in Section 4.2 as it applies to the turbo decoder, we must first consider a system which, although it is not exactly equal to that in which the turbo decoder operates, is the system which the turbo decoder assumes in its sense of optimality. We will then provide the optimization interpretation of the turbo decoder. See the notation Tables 2.3 and 2.4.

Maximum Likelihood Decoding

In a typical decoding situation, a binary message $\boldsymbol{\xi}$ is encoded and transmitted over a channel to create the received data $\mathbf{r}$. Given $\mathbf{r}$ we would like to reconstruct the original message $\boldsymbol{\xi}$. There are two senses of "optimal" when it comes to decoding $\boldsymbol{\xi}$ in the situation where we have no prior probabilities for the outcomes $\{\boldsymbol{\xi} = \mathbf{B}_i\}$. In one situation, we wish to minimize the probability of selecting the wrong sequence $\boldsymbol{\xi}$ so as to minimize the block error rate. In another situation we wish to minimize the probability of selecting the wrong bit ξ_i so as to minimize the bit error rate. We call the decisions which yield the former the *maximum likelihood sequence detection*, and the decisions which yield the later the *maximum likelihood bitwise detection*. The maximum likelihood sequence detection is then

$$\hat{\boldsymbol{\xi}}_{\text{MLSD}} = \arg \max_{\boldsymbol{\xi} \in \{0,1\}^N} p(\mathbf{r}|\boldsymbol{\xi})$$

and the maximum likelihood bitwise detection is

$$\hat{\xi}_{i,\text{MLBD}} = \arg \max_{\xi_i \in \{0,1\}} \sum_{\mathbf{v}|x_i=\xi_i} p(\mathbf{r}|\mathbf{v})$$

where $p(\mathbf{r}|\boldsymbol{\xi})$ is the likelihood function which results from concatenating the encoder with the channel.

As we discussed in Proposition 3, one can set up maximum likelihood parameter estimation problems which yield these detectors as their solutions. In particular, consider a parameter estimation problem where we are trying to determine the parameters of a factorizable prior PMF on $\boldsymbol{\xi}$ which yields the maximum a posteriori probability of having received $\mathbf{r}$. We know from Appendix D that to parameterize the set $\mathcal{M}$, it is sufficient to use a vector of log probability ratios $\boldsymbol{\lambda}$, since the set of factorizable prior PMFs for a binary word $\boldsymbol{\xi}$ is an exponential family with the

trivial sufficient statistics $\mathbf{t}(\boldsymbol{\xi}) = \boldsymbol{\xi}$. Denoting $\mathbf{B}_{\mathtt{i}}$ to be the binary representation of the integer $\mathtt{i}$, Proposition 3 tell us that to estimate $\boldsymbol{\xi}$ given $\mathbf{r}$ one can solve the problem

$$\hat{\boldsymbol{\lambda}}_{\mathrm{ML}} = \arg \max_{\boldsymbol{\lambda} \in (\mathbb{R} \bigcup \{\pm\infty\})^N} p(\mathbf{r}|\boldsymbol{\lambda}) \tag{4.23}$$

$$= \arg \max_{\boldsymbol{\lambda} \in (\mathbb{R} \bigcup \{\pm\infty\})^N} \sum_{i=0}^{2^N-1} p(\mathbf{r}|\boldsymbol{\xi} = \mathbf{B}_i)\mathbb{P}(\mathbf{B}_i|\boldsymbol{\lambda}) \tag{4.24}$$

In particular, since we know that for any $\boldsymbol{\lambda}$ we must have

$$\sum_{i=0}^{2^N-1} \mathbb{P}(\mathbf{B}_i|\boldsymbol{\lambda}) = 1$$

we see that the $\mathbb{P}$ that maximizes (4.23) takes the form

$$\mathbb{P}(\mathbf{B}_i|\boldsymbol{\lambda}) = \begin{cases} 1 & \mathbf{B}_i = \hat{\boldsymbol{\xi}}_{\mathrm{MLSD}} \\ 0 & \text{otherwise} \end{cases}$$

because putting any probability mass on any other word would not yield as high a likelihood. This then implies that $\hat{\boldsymbol{\lambda}}_{\mathrm{ML}}$ are the infinite log probability ratios associated with the word $\hat{\boldsymbol{\xi}}_{\mathrm{MLSD}}$. In this manner we have seen that the maximum likelihood sequence detector may be viewed as the answer to a maximum likelihood estimation problem.

One can also set up a set of maximum likelihood parameter estimation problems whose answers are the maximum likelihood bitwise detections. In particular,

consider the set of estimation problems

$$\hat{\lambda}_{i,\mathrm{ML}} = \arg \max_{\lambda \in \mathbb{R} \bigcup \{\pm\infty\}} \sum_{\boldsymbol{\xi} \in \{0,1\}^N} p(\mathbf{r}|\boldsymbol{\xi}) \mathbb{P}[\xi_i|\lambda] \tag{4.25}$$

$$= \arg \max_{\lambda \in \mathbb{R} \bigcup \{\pm\infty\}} \left\{ \mathbb{P}[\xi_i = 1|\lambda] \left(\sum_{\boldsymbol{\xi} \in \{0,1\}^N | \xi_i = 1} p(\mathbf{r}|\boldsymbol{\xi}) \right) + \mathbb{P}[\xi_i = 0|\lambda] \left(\sum_{\boldsymbol{\xi} \in \{0,1\}^N | \xi_i = 0} p(\mathbf{r}|\boldsymbol{\xi}) \right) \right\} \tag{4.26}$$

From the latter form it is evident that

$$\mathbb{P}[\boldsymbol{\xi}_i = 1|\hat{\lambda}_{i,\mathrm{ML}}] = \begin{cases} 1 & \sum_{\boldsymbol{\xi} \in \{0,1\}^N | \xi_i = 1} p(\mathbf{r}|\boldsymbol{\xi}) > \sum_{\boldsymbol{\xi} \in \{0,1\}^N | \xi_i = 0} p(\mathbf{r}|\boldsymbol{\xi}) \\ 0 & \text{otherwise} \end{cases}$$

which shows that $\hat{\lambda}_{i,\mathrm{ML}}$ is the (infinite) log probability ratio corresponding to the maximum likelihood bitwise detection.

In general both of these detectors can have high computational complexity for large block lengths unless some structure in the likelihood function can be exploited. We will show rigorously in the next few sections that the turbo decoder, introduced in Section 2.1.1 and 2.1.2, is a computationally efficient means of approximating these optimal detectors.

A Convenient Independence Assumption

In describing the operation of the turbo decoder, it will be advantageous to describe a system like that in which the parallel turbo decoder operates, but in which the messages $\mathbf{x}$ and $\mathbf{y}$ are chosen completely independently of one another as depicted in Figure 4.2 for the parallel case and in Figure 4.3 for the serial case. Under the false assumption that $\mathbf{x}$ and $\mathbf{y}$ were chosen independently of one another according to factorizable PMFs with bitwise log probability ratios $\boldsymbol{\lambda}_1$ and $\boldsymbol{\lambda}_2$, the likelihood

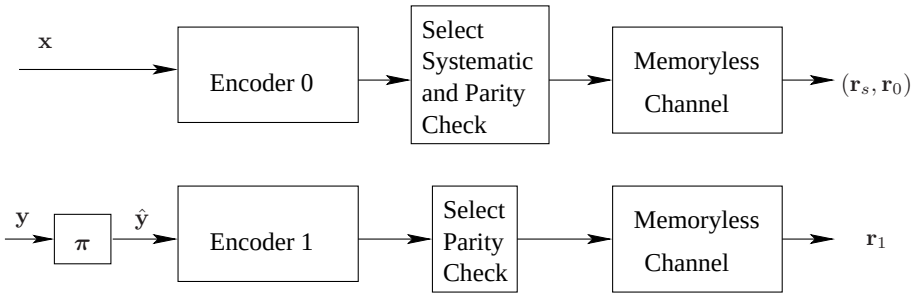

Figure 4.2: The system which the parallel turbo decoder assumes.

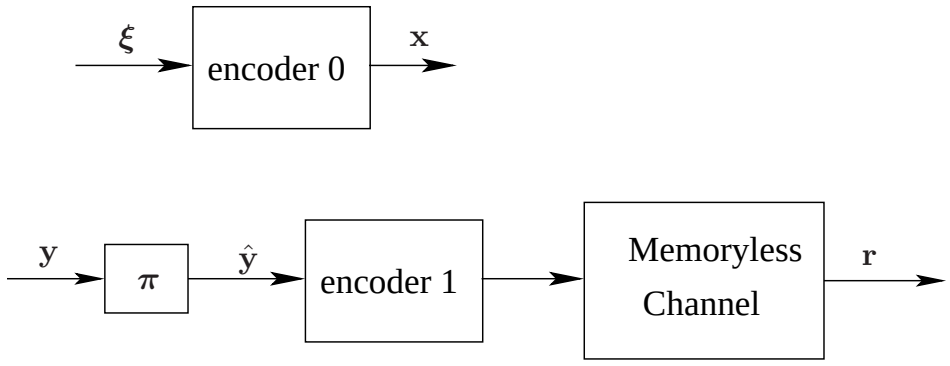

Figure 4.3: The system which the serial turbo decoder assumes.

function for the received data would be

$$\begin{aligned} \mathsf{p}_{\mathbf{r}_s,\mathbf{r}_0,\mathbf{r}_1|\boldsymbol{\alpha},\boldsymbol{\beta}}(\mathbf{r}_s,\mathbf{r}_0,\mathbf{r}_1|\boldsymbol{\alpha},\boldsymbol{\beta}) &= \left(\sum_i p(\mathbf{r}_s,\mathbf{r}_0|\mathbf{x}=\mathbf{B}_i)\mathbb{P}(\mathbf{x}=\mathbf{B}_i|\boldsymbol{\alpha})\right) \\ &\quad \left(\sum_j p(\mathbf{r}_1|\mathbf{y}=\mathbf{B}_j)\mathbb{P}(\mathbf{y}=\mathbf{B}_j|\boldsymbol{\beta})\right) \qquad (4.27) \\ &= \frac{\|\exp(\mathbf{B}\boldsymbol{\alpha}+\boldsymbol{\rho}_0)\|_1}{\|\exp(\mathbf{B}\boldsymbol{\alpha})\|_1}\frac{\|\exp(\mathbf{B}\boldsymbol{\beta}+\boldsymbol{\rho}_1)\|_1}{\|\exp(\mathbf{B}\boldsymbol{\beta})\|_1} \qquad (4.28) \end{aligned}$$

in the case of parallel concatenation. This is (4.17) applied to the turbo decoder, where the integrals have now become sums, due to the finite nature of the support measure $\mathsf{d}\boldsymbol{\theta}$ in this instance[2]. In (4.28) we used the notation

$$\boldsymbol{\rho}_0 := \left[\log\left(\mathsf{p}_{\mathbf{r}_s,\mathbf{r}_0|\mathbf{x}=\mathbf{B}_i}(\mathbf{r}_s,\mathbf{r}_0|\mathbf{x})\right)\right]$$

and

$$\boldsymbol{\rho}_1 := \left[\log\left(\mathsf{p}_{\mathbf{r}_s,\mathbf{r}_0|\mathbf{x}=\mathbf{B}_i}(\mathbf{r}_s,\mathbf{r}_0|\mathbf{x})\right)\right]$$

In the case of serial concatentation we have

$$\begin{aligned} p(\mathbf{r}|\boldsymbol{\alpha},\boldsymbol{\beta}) &= \left(\sum_i p(\mathbf{r}|\boldsymbol{\xi}=\mathbf{B}_i)\mathbb{P}(\boldsymbol{\xi}=\mathbf{B}_i|\boldsymbol{\alpha})\right)\sum_j \phi(\mathbf{B}_j)\mathbb{P}(\mathbf{y}=\mathbf{B}_j|\boldsymbol{\beta}) \qquad (4.29) \\ &= \frac{\|\exp(\mathbf{B}\boldsymbol{\alpha}+\boldsymbol{\rho}_0)\|_1}{\|\exp(\mathbf{B}\boldsymbol{\alpha})\|_1}\frac{\|\exp(\mathbf{B}\boldsymbol{\beta}+\boldsymbol{\rho}_1)\|_1}{\|\exp(\mathbf{B}\boldsymbol{\beta})\|_1} \qquad (4.30) \end{aligned}$$

Thus, recalling the definition of $\psi_{\mathbf{x}}$ from (2.2), the log likelihood function in either (serial or parallel) case is

$$\log(p(\mathbf{r}|\boldsymbol{\alpha},\boldsymbol{\beta})) = -\psi_{\mathbf{x}}(\mathbf{B}\boldsymbol{\alpha}) - \psi_{\mathbf{x}}(\mathbf{B}\boldsymbol{\beta}) + \psi_{\mathbf{x}}(\mathbf{B}\boldsymbol{\alpha}+\boldsymbol{\rho}_0) + \psi_{\mathbf{x}}(\mathbf{B}\boldsymbol{\beta}+\boldsymbol{\rho}_1)$$

Furthermore, the decision statistics which the parallel turbo decoder uses to make its decisions, which are the bitwise posterior probabilities at the output of one of

[2]Note that in this special case where there are only two component decoders, we have chosen to leave off the repetition unit, using the constraints to add in the probability that the messages are the same. This simply removes some parameters, which one typically works with during expectation propagation or belief propagation, but are usually not used in the turbo decoder.

the component decoders and have the log probability ratio vector $\boldsymbol{\alpha} + \boldsymbol{\beta}$, may be interpreted under this notation as

$$[\mathbb{P}(\mathbf{x}_i = \mathbf{y}_i = 1 | \mathbf{x} = \mathbf{y}, \boldsymbol{\alpha}, \boldsymbol{\beta})] = \mathbf{p}_{\mathbf{B}(\boldsymbol{\alpha}+\boldsymbol{\beta})}$$

or the collection over all the bits of the probability that the ith bit of $\mathbf{x}$ and $\mathbf{y}$ are equal to 1, given that $\mathbf{x} = \mathbf{y}$ and $\mathbf{x}$ and $\mathbf{y}$ are chosen with factorizable PMFs with log probability ratios $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, respectively. Since we have assumed that the first component encoder in the serial turbo encoder is systematic, the serial turbo decoder uses the same statistics for its decisions, but only the elements in $\mathbf{p}_{\mathbf{B}(\boldsymbol{\alpha}+\boldsymbol{\beta})}$ that correspond to systematic bits from that first encoder are needed.

To control this false assumption of independently choosing $\mathbf{x}$ and $\mathbf{y}$, we will be interested in choosing $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ so that the two densities so chosen have a large probability of selecting the same word. This is because we know a priori that $\mathbf{x}$ and $\mathbf{y}$ are the same, and thus, although we are relaxing the constraint that they be exactly the same in doing the decoding, we want to enforce that they be similar. Thus, at the decoder, we will be interested in the constraint set

$$\mathcal{C} = \{(\boldsymbol{\alpha}, \boldsymbol{\beta}) | \mathbb{P}(\mathbf{x} = \mathbf{y} | \boldsymbol{\alpha}, \boldsymbol{\beta}) = c\}$$

which is the set such that the probability of choosing the same message for $\mathbf{x}$ and $\mathbf{y}$, given that we chose them independently according to the factorizable densities whose bitwise marginals are $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ respectively, is fixed to c. We will now use the notation

$$\mathbb{P}(\mathbf{x} = \mathbf{B}_i | \boldsymbol{\alpha}) = \frac{\exp(\mathbf{B}_i \boldsymbol{\alpha})}{\| \exp(\mathbf{B}\boldsymbol{\alpha}) \|_1}$$
$$\mathbb{P}(\mathbf{y} = \mathbf{B}_i | \boldsymbol{\beta}) = \frac{\exp(\mathbf{B}_i \boldsymbol{\beta})}{\| \exp(\mathbf{B}\boldsymbol{\beta}) \|_1}$$

in order to rewrite

$$\mathbb{P}(\mathbf{x}=\mathbf{y}|\boldsymbol{\alpha},\boldsymbol{\beta}) = \sum_i \mathbb{P}(\mathbf{x}=\mathbf{B}_i|\boldsymbol{\alpha})\mathbb{P}(\mathbf{y}=\mathbf{B}_i|\boldsymbol{\beta}) \tag{4.31}$$

$$= \frac{\|\exp(\mathbf{B}(\boldsymbol{\alpha}+\boldsymbol{\beta}))\|_1}{\|\exp(\mathbf{B}\boldsymbol{\alpha})\|_1\|\exp(\mathbf{B}\boldsymbol{\beta})\|_1} \tag{4.32}$$

so that

$$\log(\mathbb{P}(\mathbf{x}=\mathbf{y}|\boldsymbol{\alpha},\boldsymbol{\beta})) = \log(\|\exp(\mathbf{B}(\boldsymbol{\alpha}+\boldsymbol{\beta}))\|_1) - \log(\|\exp(\mathbf{B}\boldsymbol{\alpha})\|_1) - \log(\|\exp(\mathbf{B}\boldsymbol{\beta})\|_1) \tag{4.33}$$

$$= \psi_{\mathbf{x}}(\mathbf{B}(\boldsymbol{\alpha}+\boldsymbol{\beta})) - \psi_{\mathbf{x}}(\mathbf{B}\boldsymbol{\alpha}) - \psi_{\mathbf{x}}(\mathbf{B}\boldsymbol{\beta}) \tag{4.34}$$

and so that we can write the constraint set as

$$\mathcal{C} = \{(\boldsymbol{\alpha},\boldsymbol{\beta})|\psi_{\mathbf{x}}(\mathbf{B}(\boldsymbol{\alpha}+\boldsymbol{\beta})) - \psi_{\mathbf{x}}(\mathbf{B}\boldsymbol{\alpha}) - \psi_{\mathbf{x}}(\mathbf{B}\boldsymbol{\beta}) = \log(c)\}$$

An Exact Characterization of the Turbo Decoder

In the next theorem we show that the turbo decoder is an iterative attempt to find the maximum likelihood estimate for bitwise log probability ratios $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ under the false assumption that $\mathbf{x}$ and $\mathbf{y}$ where chosen independently of one another according to the bitwise factorizable PMFs with log probability ratios $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, respectively. Furthermore, this optimization is performed subject to the constraint that the probability of selecting the same message when selecting $\mathbf{x}$ and $\mathbf{y}$ in this manner is held constant.

Thm. 4 (Turbo Decoder Stationary Points)**:** The turbo decoder stationary points are in a one to one correspondence with the critical points of the Lagrangian of the optimization problem

$$(\boldsymbol{\alpha}^*,\boldsymbol{\beta}^*) = \arg\max_{(\boldsymbol{\alpha},\boldsymbol{\beta})\in\mathcal{C}} \log(p(\mathbf{r}|\boldsymbol{\alpha},\boldsymbol{\beta}))$$

subject to a Lagrange multiplier of -1.

Proof: Form the Lagrangian

$$\begin{aligned} \mathsf{L}(\boldsymbol{\alpha},\boldsymbol{\beta}) &= \log(p(\mathbf{r}|\boldsymbol{\alpha},\boldsymbol{\beta})) + \mu\left(\log\left(\mathbb{P}\left[\mathbf{x}=\mathbf{y}|\boldsymbol{\alpha},\boldsymbol{\beta}\right]\right) - \log(c)\right) && (4.35)\\ &= -\psi_{\mathbf{x}}(\mathbf{B}\boldsymbol{\alpha}) - \psi_{\mathbf{x}}(\mathbf{B}\boldsymbol{\beta}) + \psi_{\mathbf{x}}(\mathbf{B}\boldsymbol{\alpha}+\boldsymbol{\rho}_0) + \psi_{\mathbf{x}}(\mathbf{B}\boldsymbol{\beta}+\boldsymbol{\rho}_1) && (4.36)\\ &\quad +\mu\left(-\psi_{\mathbf{x}}(\mathbf{B}\boldsymbol{\alpha}) - \psi_{\mathbf{x}}(\mathbf{B}\boldsymbol{\beta}) + \psi_{\mathbf{x}}(\mathbf{B}(\boldsymbol{\alpha}+\boldsymbol{\beta}))\right) && (4.37) \end{aligned}$$

Take its gradient

$$\begin{aligned} \nabla_{\boldsymbol{\alpha}}\mathsf{L} &= -\mathbf{p}_{\mathbf{B}\boldsymbol{\alpha}} + \mathbf{p}_{\mathbf{B}\boldsymbol{\alpha}+\boldsymbol{\rho}_0} + \mu\left(-\mathbf{p}_{\mathbf{B}\boldsymbol{\alpha}} + \mathbf{p}_{\mathbf{B}(\boldsymbol{\alpha}+\boldsymbol{\beta})}\right) && (4.38)\\ \nabla_{\boldsymbol{\beta}}\mathsf{L} &= -\mathbf{p}_{\mathbf{B}\boldsymbol{\beta}} + \mathbf{p}_{\mathbf{B}\boldsymbol{\beta}+\boldsymbol{\rho}_1} + \mu\left(-\mathbf{p}_{\mathbf{B}\boldsymbol{\beta}} + \mathbf{p}_{\mathbf{B}(\boldsymbol{\alpha}+\boldsymbol{\beta})}\right) && (4.39) \end{aligned}$$

Selecting a Lagrange multiplier of $\mu = -1$, we have

$$\begin{aligned} \nabla_{\boldsymbol{\alpha}}\mathsf{L} &= \mathbf{p}_{\mathbf{B}\boldsymbol{\alpha}+\boldsymbol{\rho}_0} - \mathbf{p}_{\mathbf{B}(\boldsymbol{\alpha}+\boldsymbol{\beta})} && (4.40)\\ \nabla_{\boldsymbol{\beta}}\mathsf{L} &= \mathbf{p}_{\mathbf{B}\boldsymbol{\beta}+\boldsymbol{\rho}_1} - \mathbf{p}_{\mathbf{B}(\boldsymbol{\alpha}+\boldsymbol{\beta})} && (4.41) \end{aligned}$$

Identifying $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ with the extrinsic information vectors from the turbo decoder, we see that the $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ which solve $\nabla\mathsf{L} = \mathbf{0}$ are stationary points of the turbo decoder. This is because $\nabla\mathsf{L} = \mathbf{0}$ implies that the bitwise posteriors from the sum of the two extrinsic information vectors are equal to the bitwise posteriors from both of the two component decoders. ■

An important fact which sets the turbo decoder apart from most constrained optimization problems is that one does not get to specify the value of $c = \mathbb{P}[\mathbf{x} = \mathbf{y}|\boldsymbol{\alpha},\boldsymbol{\beta}]$ before decoding. Instead, if the turbo decoder converges, one may evaluate the constraint function $-\psi_{\mathbf{x}}(\mathbf{B}\boldsymbol{\alpha})-\psi_{\mathbf{x}}(\mathbf{B}\boldsymbol{\beta})+\psi_{\mathbf{x}}(\mathbf{B}(\boldsymbol{\alpha}+\boldsymbol{\beta}))$ at the convergent $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ to get the $\log(c)$ for which the turbo decoder stationary points are critical points of the optimization problem. Note also that if we have $c = \mathbb{P}[\mathbf{x} = \mathbf{y}|\boldsymbol{\alpha},\boldsymbol{\beta}] = 1$ so

that $\log(c) = 0$, then the globally optimal solution to the optimization problem is actually the maximum likelihood sequence detection. This then suggests that for convergent values of $\log(c)$ close to 0, the turbo decoder will provide a detection close to a critical point of the wordwise likelihood function, provided continuity in $\log(c)$ of the solutions to the optimization problem.

Given the importance of selecting the Lagrange multiplier $\mu = -1$, from here on, we will always select $\mu = -1$ when we refer to the Lagrangian. Because the condition that the gradient of the Lagrangian is equal to zero is necessary, but not always sufficient, for a point to be the global optima, we must characterize the type of critical points which are possible. In particular, we wish to know whether or not the critical points of the Lagrangian are at least local maxima of the constrained optimization problem. Generally speaking one can converge to either a maximum or minimum, although if one replaces the Lagrangian with the expectation of the Lagrangian over the received data one can guarantee that there is only a global maximum.

Thm. 5 (Critical Point Characterization)**:** The expected Lagrangian has only one critical point which is a maximum of the expected constrained optimization problem. Here, the expectation is taken over the received information $\mathbf{r}$ given $\boldsymbol{\alpha}, \boldsymbol{\beta}$.

Proof: To see this, consider the value of the Lagrangian within the constraint space

$$\begin{aligned} \mathsf{L} &= -\psi_{\mathbf{x}}(\mathbf{B}(\boldsymbol{\alpha}+\boldsymbol{\beta})) + \psi_{\mathbf{x}}(\mathbf{B}\boldsymbol{\alpha}+\boldsymbol{\rho}_0) + \psi_{\mathbf{x}}(\mathbf{B}\boldsymbol{\beta}+\boldsymbol{\rho}_1) && (4.42) \\ &= -\psi_{\mathbf{x}}(\mathbf{B}\boldsymbol{\alpha}) - \psi_{\mathbf{x}}(\mathbf{B}\boldsymbol{\beta}) + C + \psi_{\mathbf{x}}(\mathbf{B}\boldsymbol{\alpha}+\boldsymbol{\rho}_0) + \psi_{\mathbf{x}}(\mathbf{B}\boldsymbol{\beta}+\boldsymbol{\rho}_1) && (4.43) \end{aligned}$$

where in the latter equation we substituted in the constraint. Now, note from this

that L thus is the sum of two log likelihood functions within the constraint space

$$\mathsf{L} = (-\psi_{\mathbf{x}}(\mathbf{B}\boldsymbol{\alpha}) + \psi_{\mathbf{x}}(\mathbf{B}\boldsymbol{\alpha} + \boldsymbol{\rho}_0)) + (-\psi_{\mathbf{x}}(\mathbf{B}\boldsymbol{\beta}) + \psi_{\mathbf{x}}(\mathbf{B}\boldsymbol{\beta} + \boldsymbol{\rho}_1)) \quad (4.44)$$

$$= \log(p(\mathbf{r}_s, \mathbf{r}_0|\boldsymbol{\alpha})) + \log(p(\mathbf{r}_1|\boldsymbol{\beta})) \quad (4.45)$$

This then implies that the second derivative of the Lagrangian within the constraint set has a mean which is the negative of the Fisher information matrix, call it $\mathbf{J}$, since the Fisher information matrix is defined as

$$\mathbf{J} = -\begin{pmatrix} \int \nabla^2_{\boldsymbol{\alpha},\boldsymbol{\alpha}}\{\log(p(\mathbf{r}|\boldsymbol{\alpha},\boldsymbol{\beta}))\}p(\mathbf{r}|\boldsymbol{\alpha},\boldsymbol{\beta})d\mathbf{r} & \mathbf{0} \\ \mathbf{0} & \int \nabla^2_{\boldsymbol{\beta},\boldsymbol{\beta}}\{\log(p(\mathbf{r}|\boldsymbol{\alpha},\boldsymbol{\beta}))\}p(\mathbf{r}|\boldsymbol{\alpha},\boldsymbol{\beta})d\mathbf{r} \end{pmatrix}$$

where we have used $\nabla^2\{\cdot\}$ here to denote the operator which takes a function to its Hessian matrix of second order partial derivatives. The well known fact, then, that the Fisher information matrix is positive semi-definite, then shows that the expectation of the Hessian matrix of the Lagrangian L is negative semi-definite. This implies then, by Theorem 4.5 of [53], the function $\mathbb{E}[\mathsf{L}]$ is concave, where $\mathbb{E}$ denotes expectation with respect to $p(\mathbf{r}|\boldsymbol{\alpha},\boldsymbol{\beta})$, and thus has a unique global maxima. Roughly speaking, this means that L is concave, and thus has a unique global maximum, on average.■

Two important distinctions are made in the previous theorem. First of all, while the expected value of the Lagrangian is concave within the constraint space, and thus has a unique global maximum and no spurious local maxima, we are not guaranteed that for a particular sample $\mathbf{r}_s, \mathbf{r}_0, \mathbf{r}_1$ there is only one solution to $\nabla\mathsf{L} = \mathbf{0}$. In fact, Figure 4.4 shows that, even for $N = 2$, it is possible, depending on $\mathbf{r}_s, \mathbf{r}_0, \mathbf{r}_1$ for this convergent point to be either a maximum or minimum. Another important distinction is that while the expected value of the Lagrangian is concave within the constraint space, it is not necessarily concave outside of the constraint

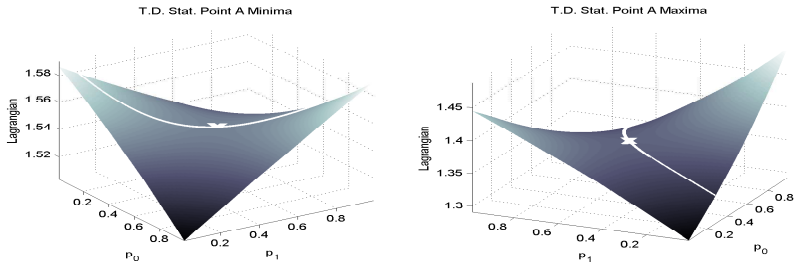

Figure 4.4: L for two different sample values of $\boldsymbol{\rho}_0$ and $\boldsymbol{\rho}_1$ for $N = 2$ bits and simple parity check codes. Here, the line indicates $\mathcal{C}$, and x marks the spot to which the Turbo Decoder converges given an initialization at $(\frac{1}{2}, \frac{1}{2})$. The p_0 and p_1 axes are the bitwise marginal probabilities associated with $\boldsymbol{\alpha}$, and we are always selecting $\boldsymbol{\beta} = \pi(\mathbf{B}\boldsymbol{\alpha} + \boldsymbol{\rho}_0) - \boldsymbol{\alpha}$. In one instance we have converged to a local minimum along $\mathcal{C}$, and in another to a local maximum along $\mathcal{C}$.

space. In fact, one can show that outside of the constraint space

$$\nabla^2_{\boldsymbol{\alpha},\boldsymbol{\alpha}}\mathsf{L} = \mathbf{P}_{\mathbf{B}\boldsymbol{\alpha}+\boldsymbol{\rho}_0} - \mathbf{p}_{\mathbf{B}\boldsymbol{\alpha}+\boldsymbol{\rho}_0}\mathbf{p}^T_{\mathbf{B}\boldsymbol{\alpha}+\boldsymbol{\rho}_0} - \left(\mathbf{P}_{\mathbf{B}(\boldsymbol{\alpha}+\boldsymbol{\beta})} - \mathbf{p}_{\mathbf{B}(\boldsymbol{\alpha}+\boldsymbol{\beta})}\mathbf{p}^T_{\mathbf{B}(\boldsymbol{\alpha}+\boldsymbol{\beta})}\right) \quad (4.46)$$

$$\nabla^2_{\boldsymbol{\alpha},\boldsymbol{\beta}}\mathsf{L} = -\left(\mathbf{P}_{\mathbf{B}(\boldsymbol{\alpha}+\boldsymbol{\beta})} - \mathbf{p}_{\mathbf{B}(\boldsymbol{\alpha}+\boldsymbol{\beta})}\mathbf{p}^T_{\mathbf{B}(\boldsymbol{\alpha}+\boldsymbol{\beta})}\right) \quad (4.47)$$

$$\nabla^2_{\boldsymbol{\beta},\boldsymbol{\beta}}\mathsf{L} = \mathbf{P}_{\mathbf{B}\boldsymbol{\alpha}+\boldsymbol{\rho}_0} - \mathbf{p}_{\mathbf{B}\boldsymbol{\alpha}+\boldsymbol{\rho}_0}\mathbf{p}^T_{\mathbf{B}\boldsymbol{\alpha}+\boldsymbol{\rho}_0} - \left(\mathbf{P}_{\mathbf{B}(\boldsymbol{\alpha}+\boldsymbol{\beta})} - \mathbf{p}_{\mathbf{B}(\boldsymbol{\alpha}+\boldsymbol{\beta})}\mathbf{p}^T_{\mathbf{B}(\boldsymbol{\alpha}+\boldsymbol{\beta})}\right) \quad (4.48)$$

where $\mathbf{P}_{\boldsymbol{\theta}}$ is the matrix whose i, jth entry is the probability that both the ith and jth bits are one, according to the wordwise measure whose log coordinates are $\boldsymbol{\theta}$. From this we can see that at a Turbo decoder stationary point, the diagonal elements of $\nabla^2\mathsf{L}$ are all zero. Thus, the trace of the Hessian matrix is zero, and, provided the Hessian is not identically zero, the stationary points are saddle points of the Lagrangian when one does not restrict oneself to the constraint space.

4.2.2 Application to Belief Propagation (e.g. LDPC) Decoding

We introduced belief propagation decoding in the context of LDPC decoding in Section 2.1.4. In order to consider a specific case of the wide ranging result of Theorem 3, we will consider our parameters from expectation propagation $\boldsymbol{\theta}$ to be a binary vector and use the simplest sufficient statistics $\mathbf{t}(\boldsymbol{\theta}) := \boldsymbol{\theta}$ so $\mathbf{t}_{\mathtt{a}}(\boldsymbol{\theta}) := \boldsymbol{\theta}_{\mathtt{a}}$. These choices are the ones used in belief propagation decoding.

Belief propagation is often described as a method for obtaining the marginal probabilities for the states $\theta_{\mathtt{i}}$, $\mathtt{i} \in \{1, \ldots, \mathtt{N}\}$ of the variable nodes in a factor graph. While this certainly holds true for factor graphs that are trees, the breakthrough applications of the algorithm to the decoding of turbo [33, 14] and LDPC codes [9] involve networks with loops. In these cases, many authors have noted that the convergent values often do not approximate the marginal probabilities well, and several authors, believing that this must be a problem, have proposed and analyzed more complex algorithms which more accurately calculate the marginal probabilities [43, 36, 45, 52, 35]. This path is justifiable, because the (variable) nodewise marginal probabilities are the correct statistics to use in making decisions in an algorithm seeking to minimize the probability of (variable) nodewise error [12]. Yet, the unmodified belief propagation decoding algorithm still yields marginal values in those breakthrough applications which allow for good performance in a decoder, suggesting that the algorithm needs no modification in important cases. Rather than attempting to fix a decoder which was already working well, we strive here to accurately explain why the convergent points of belief propagation decoding can offer good performance. To do so, it will be necessary to identify the sense in which the convergent points of belief propagation decoding are optimal,

since it is clear that in instances where there are loops in the graph the convergent points no longer minimize the probability of nodewise error. Other authors [42, 35] have attempted to determine the sense of optimality of the convergent points by pointing out that the message passing dynamics can be viewed as resulting from minimizing the Bethe approximation to the variational free energy, and minimizing the variational free energy gives marginalization. Those authors have noted, however, that the Bethe approximation does not yield the exact variational free energy in loopy factor graphs, and it is thus still not clear why minimizing it yields good performance in situations where it is not exact. One can determine a great many optimization problems which yield the same answers, and usually one wishes to choose an objective function and constraints which have clear intuitive meaning. This motivates us to search for other optimizations whose objective functions and constraints have deep intuitive meaning and which yield the belief propagation decoding stationary points as their solutions. We are also motivated to explain the optimality of the stationary points of belief propagation decoding using analytical tools which are likely to be familiar to the wide range of engineers and scientists that use the belief propagation decoding algorithm, and we believe maximum likelihood estimation to be such a tool.

Here, we argue that belief propagation decoding may be viewed as a detector which iteratively attempts to minimize the blockwise error probability, that is the probability of selecting the wrong $\boldsymbol{\theta} = [\theta_1, \ldots, \theta_{\mathtt{N}}]^T$, subject to specific constraints. After reviewing some preliminaries in Section 4.2.2, we do this by showing in Section 4.2.2 that the stationary points of belief propagation decoding are critical points of an intuitively pleasing constrained blockwise maximum likelihood estimation problem. If the convergent value of the constraint, which is easy to measure

with low complexity in practice, is equal to zero, then the corresponding stationary point of the belief propagation decoding algorithm is a critical point of the blockwise-likelihood function. We then show via simulation that in the situations where the belief propagation decoding algorithm is applied, that is, in situations where the decisions provided by belief propagation decoding yield low probability of error, these constraints are close to zero, which suggests, via continuity, that the convergent points of belief propagation decoding are close to critical points of the network-wide likelihood function and thus network-wide maximum likelihood detections.

Belief Propagation in Factor Graphs

In this article, we will consider the use of the belief propagation decoding (aka sum product) algorithm in factor graphs of a likelihood function corresponding to the likelihood of decoding some observations $\mathbf{r}$, given a vector of $\mathtt{N}$ bits $\boldsymbol{\theta} \in \{0,1\}^{\mathtt{N}}$. We have observed $\mathbf{r}$ and we would like to determine the $\boldsymbol{\theta} = [\theta_1, \ldots, \theta_{\mathtt{N}}]^T$ which gave rise to our observations. Furthermore, the likelihood function factors into the product of functions of smaller subsets of variables

$$p(\mathbf{r}|\boldsymbol{\theta}) = \frac{1}{Z} \prod_{a=1}^{\mathtt{M}} \mathsf{f}_{\mathsf{a}}(\boldsymbol{\theta}_{\mathsf{a}}) \tag{4.49}$$

where $\boldsymbol{\theta}_{\mathsf{a}}$ is a vector containing the elements of some subset of $\{\theta_1, \ldots, \theta_{\mathtt{N}}\}$ and Z is a constant defined by

$$Z = \sum_{\boldsymbol{\theta} \in \{0,1\}^{\mathtt{N}}} \prod_{a=1}^{\mathtt{M}} \mathsf{f}_{\mathsf{a}}(\boldsymbol{\theta}_{\mathsf{a}})$$

The factor graph encodes the factorization in (4.49) into a bipartite graph of variable nodes representing the elements of the vector $\boldsymbol{\theta}$, and factor nodes representing the functions f_{a}. The graph is bipartite, because an edge may connect only a variable node with a factor node so that there can be no edges incident on two variable

nodes and there can be no edges incident on two factor nodes. In fact, if an edge connects a variable node $\theta_{\mathtt{i}}$ with a factor node $\mathsf{f}_{\mathtt{a}}$, it means that $\mathsf{f}_{\mathtt{a}}$ depends on $\theta_{\mathtt{i}}$. The belief propagation decoding algorithm lists a set of message passing rules along edges between adjacent factor nodes and variable nodes. In particular, denote the message passed from the variable node $\theta_{\mathtt{i}}$ to the factor node $\mathsf{f}_{\mathtt{a}}$ by $\mathsf{n}_{n,a}$ and denote the message passed from the factor node $\mathsf{f}_{\mathtt{a}}$ to the variable node $\theta_{\mathtt{i}}$ by $\mathsf{m}_{a,n}$, then the message passing rules of belief propagation decoding are

$$\mathsf{n}_{n,a}(\theta_{\mathtt{i}}) \leftarrow \frac{1}{C} \prod_{c \in \mathcal{N}(n) \backslash a} \mathsf{m}_{c,n}(\theta_{\mathtt{i}})$$

with

$$C = \sum_{\theta_{\mathtt{i}} \in \{0,1\}} \prod_{c \in \mathcal{N}(n) \backslash a} \mathsf{m}_{c,n}(\theta_{\mathtt{i}})$$

and

$$\mathsf{m}_{a,n}(\theta_{\mathtt{i}}) \leftarrow \frac{1}{D} \sum_{\boldsymbol{\theta}_{\mathtt{a}} \backslash \theta_{\mathtt{i}}} \mathsf{f}_{\mathtt{a}}(\boldsymbol{\theta}_{\mathtt{a}}) \prod_{n' \in \mathcal{N}(a) \backslash n} \mathsf{n}_{n',a}(\theta_{n'})$$

with

$$D = \sum_{\boldsymbol{\theta}_{\mathtt{a}}} \mathsf{f}_{\mathtt{a}}(\boldsymbol{\theta}_{\mathtt{a}}) \prod_{n' \in \mathcal{N}(a) \backslash n} \mathsf{n}_{n',a}(\theta_{n'})$$

where we have replicated the notation from [35]. Because we are considering only variable nodes which can take on binary values, these messages may be summarized by passing log likelihood ratios instead of single bit probability mass functions. Given a log probability ratio, we can determine the corresponding single bit probability mass function. In other words, we can equivalently consider belief propagation decoding to be an algorithm which passes the log likelihood ratios

$$\lambda_{n,a} = \log \left(\frac{\mathsf{n}_{n,a}(1)}{\mathsf{n}_{n,a}(0)} \right)$$

and

$$\gamma_{a,n} = \log \left(\frac{\mathsf{m}_{a,n}(1)}{\mathsf{m}_{a,n}(0)} \right)$$

between the nodes. Taking this a step further, define

$$\rho_{\mathtt{a}}(\boldsymbol{\theta}) = \log\left(\frac{\mathsf{f}_{\mathtt{a}}(\boldsymbol{\theta}_{\mathtt{a}})}{\mathsf{f}_{\mathtt{a}}(\mathbf{B}_0)}\right)$$

and list this function into a vector

$$\boldsymbol{\rho}_{\mathtt{a}} = [\rho_{\mathtt{a}}(\mathbf{B}_0), \rho_{\mathtt{a}}(\mathbf{B}_1), \ldots, \rho_{\mathtt{a}}(\mathbf{B}_{2^{\mathtt{N}}-1})]$$

Similarly, define the vectors of log likelihood ratios

$$\boldsymbol{\alpha}_{\mathtt{a}} = [\lambda_{1,a}, \lambda_{2,a}, \ldots, \lambda_{\mathtt{N},a}]^T$$

and

$$\boldsymbol{\gamma}_{\mathtt{a}} = [\gamma_{a,1}, \gamma_{a,2}, \ldots, \gamma_{a,\mathtt{N}}]$$

with the understanding that if there is no edge connecting $\theta_{\mathtt{i}}$ and $\mathsf{f}_{\mathtt{a}}$, then the values $\lambda_{n,a}$ and $\gamma_{a,n}$, whatever they may be, are meaningless in terms of the factor graph. We can now rewrite the message passing that occurs in the belief propagation decoding algorithm as

$$\boldsymbol{\alpha}_{\mathtt{a}} \leftarrow \mathsf{fact}\left(\mathbf{B}\boldsymbol{\gamma}_{\mathtt{a}} + \boldsymbol{\rho}_{\mathtt{a}}\right) - \boldsymbol{\gamma}_{\mathtt{a}} \tag{4.50}$$

and

$$\boldsymbol{\gamma}_{\mathtt{a}} \leftarrow \sum_{c \neq a} \boldsymbol{\alpha}_c$$

or equivalently

$$\boldsymbol{\gamma}_{\mathtt{a}} \leftarrow \mathsf{fact}\left(\sum_{c=1}^{\mathtt{M}} \mathbf{B}\boldsymbol{\alpha}_c\right) - \boldsymbol{\lambda}_{\mathtt{a}} \tag{4.51}$$

where we used fact to represent the map which takes a wordwise PMF's log coordinates to its bitwise log probability ratios

$$\mathsf{fact}(\boldsymbol{\rho}) = \log(\mathbf{B}^T \exp(\boldsymbol{\rho})) - \log((\mathbf{1} - \mathbf{B})^T \exp(\boldsymbol{\rho}))$$

Now, with regards to locations in the vectors $\boldsymbol{\lambda}_{\mathsf{a}}$ corresponding to non-existent edges, we can see from the independence of $\rho_{\mathsf{a}}(\boldsymbol{\theta})$ on θ_{i} for n not in $\mathcal{N}(a)$ that $\boldsymbol{\lambda}_{n,a} = 0$, since for these n $\mathsf{fact}_{\mathsf{i}}(\mathbf{B}\boldsymbol{\gamma}_{\mathsf{a}} + \boldsymbol{\rho}_{\mathsf{a}}) = \gamma_{a,n}$.

Since the map between a bit's log probability ratio and the probability that the bit is one is a bijective map, we can also write these relations with respect to the bitwise probabilities. In particular, we could have equivalently determined the message passing rules by stating

$$\boldsymbol{\gamma}_{\mathsf{a}} \leftarrow \boldsymbol{\lambda} \text{ such that } \mathbf{p}_{\mathbf{B}(\boldsymbol{\lambda}+\boldsymbol{\lambda}_{\mathsf{a}})} - \mathbf{p}_{\mathbf{B}\sum_{c=1}^{\mathtt{M}}\boldsymbol{\lambda}_c} = \mathbf{0} \tag{4.52}$$

$$\boldsymbol{\lambda}_{\mathsf{a}} \leftarrow \boldsymbol{\lambda} \text{ such that } \mathbf{p}_{\mathbf{B}(\boldsymbol{\gamma}_{\mathsf{a}}+\boldsymbol{\lambda})} - \mathbf{p}_{\mathbf{B}\boldsymbol{\gamma}_{\mathsf{a}}+\boldsymbol{\rho}_{\mathsf{a}}} = \mathbf{0} \tag{4.53}$$

It is this way of writing the belief propagation decoding dynamics which allows us to most easily recognize their optimality.

A Factorization Related to Turbo Decoding

Writing the belief propagation decoder in the form (4.50) and (4.51) allows us to recognize another duality between belief propagation decoding and turbo decoding. It has already been shown that the turbo decoder may be thought of as an instance of the belief propagation decoding algorithm [33, 14]. This is certainly the correct ordering when one is considering the chronological order of invention of the two algorithms in the context of the decoding of error correction codes [9, 3]. Writing the belief propagation decoding algorithm in the form (4.50) and (4.51), however, emphasizes that belief propagation decoding may be viewed as an instance of turbo decoding, and identifies that the two ideas are equivalent. The mindset used in looking at belief propagation decoding in this manner will be helpful in developing intuition for our results later, so we will describe this duality here. To see this, consider the factors $\mathsf{f}_{\mathsf{a}}(\boldsymbol{\theta}_{\mathsf{a}})$ of the original likelihood function $p(\mathbf{r}|\boldsymbol{\theta})$ to be

likelihood functions themselves. In this way each $\mathtt{f_a}(\boldsymbol{\theta}_\mathtt{a})$ may be regarded as the likelihood function $\tilde{f}_\mathtt{a}(\boldsymbol{\theta})$ for $\boldsymbol{\theta}$ from some code $\mathtt{c_a}$ given our observations $\mathbf{r}$ (where $\tilde{f}_\mathtt{a}(\boldsymbol{\theta})$ is constant with respect to $\theta_\mathtt{i}$ for i not in $\mathcal{N}(a)$). We also know that the $\boldsymbol{\theta}$ in each $\tilde{f}_\mathtt{a}(\boldsymbol{\theta})$ must be the same, so we can consider this fact as arising from a repetition code for $\boldsymbol{\theta}$. Thinking about belief propagation decoding in this manner inspires us to consider the original factor graph as actually arising from the serial concatenation of a repetition code with $\mathtt{M}$ parallel "codes", where $\mathtt{M}$ is the number of factors in the original factor graph. Running belief propagation decoding on the general factor graph in Figure 4.5 (a) is thus equivalent to running the turbo decoder for Figure 4.5 (b), and is therefore no more mathematically general than turbo decoding if one allows the concatenation of multiple codes.

The spirit of the idea here is an alternate factorization of the likelihood function $p(\mathbf{r}|\boldsymbol{\theta})$ related to belief propagation decoding on the factor graph from the original factorization. In particular, if we introduce extra variables $\mathbf{x}_\mathtt{a}$ for all $a \in \{1, \ldots, \mathtt{M}\}$ and the functions $\tilde{f}_\mathtt{a}(\boldsymbol{\theta}) = \mathtt{f_a}(\boldsymbol{\theta}_\mathtt{a})\ \forall \boldsymbol{\theta}$, then we can rewrite the likelihood function as

$$p(\mathbf{r}|\mathbf{x}_\mathtt{:}) = \frac{1}{Z} g(\mathbf{x}_1, \ldots, \mathbf{x}_\mathtt{M}) \prod_{a=1}^{\mathtt{M}} \tilde{f}_\mathtt{a}(\mathbf{x}_\mathtt{a})$$

where we have the new factor g which enforces that all of the $\mathbf{x}_\mathtt{a}$ be equal (to $\boldsymbol{\theta}$, in particular)

$$g(\mathbf{x}_1, \ldots, \mathbf{x}_\mathtt{M}) = \prod_{a=1}^{\mathtt{M}} \delta(\mathbf{x}_1 - \mathbf{x}_\mathtt{a})$$

where δ is the multidimensional Kronecker delta function which is defined as

$$\delta(\boldsymbol{\theta}) = \begin{cases} 1 & \boldsymbol{\theta} = \mathbf{0} \\ 0 & \text{otherwise} \end{cases}$$

Note that choosing the $(\mathbf{x}_\mathtt{:})$ to maximize $p(\mathbf{r}|\mathbf{x}_\mathtt{:})$ yields the same result for $\boldsymbol{\theta} = \mathbf{x}_\mathtt{a}$ for any a as maximizing $p(\mathbf{r}|\boldsymbol{\theta})$. Furthermore, marginalizing $p(\mathbf{r}|\mathbf{x}_\mathtt{:})$ for $\mathbf{x}_{a,n}$ also

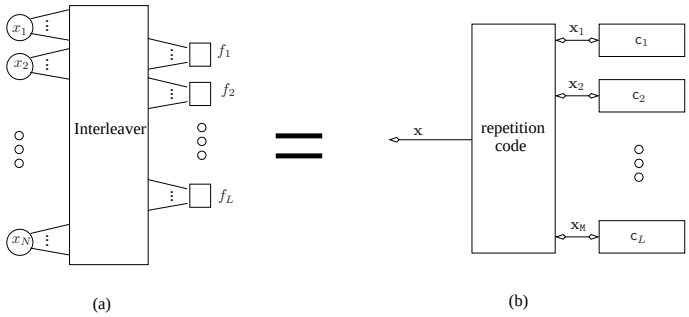

Figure 4.5: An equivalent interpretation of a factor graph arising from a likelihood function. In particular, we can consider each factor $\mathsf{f_a}$ to be a likelihood function itself, and the factor graph emphasizes the way that the arguments $\boldsymbol{\theta}_\mathsf{a}$ must all arise from subsets of a common vector $\boldsymbol{\theta}$. This may be represented as in (b), as the serial concatenation of a repetition code which makes L copies of the vector $\boldsymbol{\theta}$, and constraints $\mathsf{c_i}$ which represent the model for the way the observations are generated according to the function $\mathsf{f_a}$. Each constraint $\mathsf{c_a}$ corresponds to a model which, together with the observations $\mathbf{r}$, gives $\mathsf{f_a}$ as a likelihood function for $\boldsymbol{\theta}$.

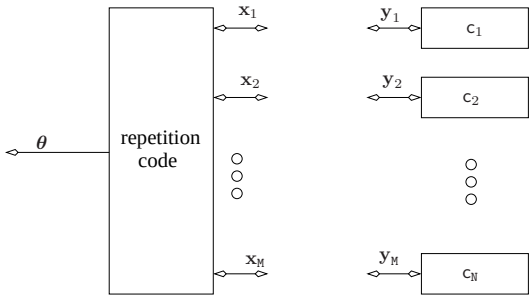

Figure 4.6: The system which the belief propagation decoding algorithm assumes. In particular, a false assumption is made that the input $\mathbf{y}_{\mathtt{a}}$ to the ath constraint $\mathtt{c}_{\mathtt{a}}$ is independently chosen from the output $\mathbf{x}_{\mathtt{a}}$ of the repetition code via a factorizable pmf with log probability ratios $\boldsymbol{\gamma}_{\mathtt{a}}$. Also, it is assumed that the output of the repetition code $\mathbf{x}_{\mathtt{a}}$ is chosen with a factorizable density whose log probability ratios are $\boldsymbol{\lambda}_{\mathtt{a}}$.

yields the same result as marginalizing $p(\mathbf{r}|\boldsymbol{\theta})$ for $\theta_{\mathtt{i}}$.

We now may view g as being the likelihood function for the repetition code depicted in Figure 4.5, whose single turbo decoding step is represented by (4.51) and $\prod_{a=1}^{\mathtt{M}} \tilde{f}_{\mathtt{a}}(\mathbf{x}_{\mathtt{a}})$ to be the likelihood function resulting from the codes on the right in Figure 4.5 and whose single turbo decoding step is represented by (4.50).

Belief Propagation as Iterative Maximum Likelihood

To help reinterpret belief propagation decoding, it will be useful to consider a system, shown in Figure 4.6, which is related to the system in which the belief propagation decoding algorithm is operating. In particular we will ignore the fact that the output of the repetition code is connected to the constraints $\mathtt{c}_{\mathtt{a}}$, or equivalently that the output of the interleaver in the factor graph is connected to

the factor nodes. Instead assume that for each constraint $\mathtt{c}_\mathtt{a}$ (i.e. the "code" giving rise to the likelihood function $\mathtt{f}_\mathtt{a}$ given our observations) there are two independent messages $\mathbf{x}_\mathtt{a}$ and $\mathbf{y}_\mathtt{a}$ chosen from factorizable densities with log probability ratios $\boldsymbol{\lambda}_\mathtt{a}$ and $\boldsymbol{\gamma}_\mathtt{a}$, at the output of the repetition unit and the input to the constraint, respectively. This is the same as assuming that there are $\mathtt{M}$ parallel independent factor graphs, where the ath one has variable nodes $\mathbf{x}_\mathtt{a}$ and only has those edges which are adjacent to the factor $\mathtt{f}_\mathtt{a}$.

This independence assumption may be related to yet another refactorization of the likelihood function $p(\mathbf{r}|\boldsymbol{\theta})$ by introducing extra variables $\mathbf{x}_\mathtt{a}$ and $\mathbf{y}_\mathtt{a}$

$$p(\mathbf{r}|\boldsymbol{\theta}, \mathbf{x}_:, \mathbf{y}_:) = \frac{1}{Z} g(\mathbf{x}_1, \ldots, \mathbf{x}_\mathtt{M}) h(\mathbf{x}_:, \mathbf{y}_:) \prod_{a=1}^{\mathtt{M}} \tilde{f}_\mathtt{a}(\mathbf{y}_\mathtt{a})$$

where, once again

$$g(\mathbf{x}_1, \ldots, \mathbf{x}_\mathtt{M}) = \prod_{a=1}^{\mathtt{M}} \delta(\mathbf{x}_1 - \mathbf{x}_\mathtt{a})$$

and we have introduced a new factor

$$h(\mathbf{x}_:, \mathbf{y}_:) = \prod_{a=1}^{\mathtt{M}} \delta(\mathbf{x}_\mathtt{a} - \mathbf{y}_\mathtt{a})$$

which enforces that $\mathbf{x}_\mathtt{a}$ and $\mathbf{y}_\mathtt{a}$ must be equal. Once again, maximizing $p(\mathbf{r}|\mathbf{x}_:, \mathbf{y}_:)$ trivially leads to the same $\boldsymbol{\theta} = \mathbf{x}_\mathtt{a} = \mathbf{y}_\mathtt{a}$ for any a as that which maximizes $p(\mathbf{r}|\boldsymbol{\theta})$ and marginalizing $p(\mathbf{r}|\boldsymbol{\theta}, \mathbf{x}_:, \mathbf{y}_:)$ leads to the same marginals for $\mathbf{x}_{a,n} = \mathbf{y}_{b,n}$ as marginalizing $p(\mathbf{r}|\boldsymbol{\theta})$.

Using this new factorization, the false independence assumption may then be regarded as approximating the true likelihood function $p(\mathbf{r}|\boldsymbol{\theta}, \mathbf{x}_:, \mathbf{y}_:)$ with

$$p_{app}(\mathbf{r}|\mathbf{x}_:, \mathbf{y}_:) = \frac{1}{Z} g(\mathbf{x}_1, \ldots, \mathbf{x}_\mathtt{M}) \prod_{a=1}^{\mathtt{M}} \tilde{f}_\mathtt{a}(\mathbf{y}_\mathtt{a})$$

Writing this in terms of the log probability ratios $\boldsymbol{\lambda}_:, \boldsymbol{\gamma}_:$ then gives

$$\begin{aligned}
\log(p_{app}(\mathbf{r}|\boldsymbol{\lambda}_:, \boldsymbol{\gamma}_:)) &= \log\left(\sum_{\mathbf{x}_:, \mathbf{y}_:} p_{app}(\mathbf{r}|\mathbf{x}_:, \mathbf{y}_:) \prod_{a=1}^{\mathtt{M}} \mathbb{P}[\mathbf{x}_\mathtt{a}|\boldsymbol{\lambda}_\mathtt{a}]\mathbb{P}[\mathbf{y}_\mathtt{a}|\boldsymbol{\gamma}_\mathtt{a}]\right) \\
&= \log\left(\mathbb{P}[\mathbf{x}_\mathtt{a} = \boldsymbol{\xi}\ \forall a|\boldsymbol{\lambda}_:] \prod_{a=1}^{\mathtt{M}} \sum_{i=0}^{2^\mathtt{N}-1} \tilde{f}_\mathtt{a}(\mathbf{B}_\mathtt{i})\mathbb{P}[\mathbf{y}_\mathtt{a} = \mathbf{B}_\mathtt{i}|\boldsymbol{\gamma}_\mathtt{a}]\right) \\
&= \log\left(\frac{\sum_{i=0}^{2^\mathtt{N}-1} \exp(\sum_{a=1}^{\mathtt{M}} \mathbf{B}_\mathtt{i}^T \boldsymbol{\lambda}_\mathtt{a})}{\prod_{a=1}^{\mathtt{M}} \sum_{i=0}^{2^\mathtt{N}-1} \exp(\mathbf{B}_\mathtt{i}^T \boldsymbol{\lambda}_\mathtt{a})}\right) \\
&\quad + \log\left(\prod_{a=1}^{\mathtt{M}} \frac{\sum_{i=0}^{2^\mathtt{N}-1} \exp(\mathbf{B}_\mathtt{i}^T \boldsymbol{\gamma}_\mathtt{a} + \boldsymbol{\rho}_\mathtt{a}(\mathbf{B}_\mathtt{i}))}{\sum_{i=0}^{2^\mathtt{N}-1} \exp(\mathbf{B}_\mathtt{i}^T \boldsymbol{\gamma}_\mathtt{a}) \sum_{i=0}^{2^\mathtt{N}-1} \exp(\boldsymbol{\rho}_\mathtt{a}(\mathbf{B}_\mathtt{i}))}\right) \\
&= \psi_\mathbf{x}\left(\sum_{a=1}^{\mathtt{M}} \mathbf{B}\boldsymbol{\lambda}_\mathtt{a}\right) + \sum_{a=1}^{\mathtt{M}} \psi_\mathbf{x}(\mathbf{B}\boldsymbol{\gamma}_\mathtt{a} + \boldsymbol{\rho}_\mathtt{a}) - \sum_{a=1}^{\mathtt{M}} \psi_\mathbf{x}(\mathbf{B}\boldsymbol{\lambda}_\mathtt{a}) \\
&\quad - \sum_{a=1}^{\mathtt{M}} \psi_\mathbf{x}(\mathbf{B}\boldsymbol{\gamma}_\mathtt{a}) - \sum_{a=1}^{\mathtt{M}} \psi_\mathbf{x}(\boldsymbol{\rho}_\mathtt{a})
\end{aligned}$$

Of course, we want to minimize the effect of dropping h from the factorization. One way to do this is to control the false independence assumption between $\mathbf{x}_\mathtt{a}$ and $\mathbf{y}_\mathtt{a}$ by constraining the probability that the $\mathbf{x}_\mathtt{a}$ and $\mathbf{y}_\mathtt{a}$ so chosen are the same for each a to be a constant c.

$$\mathbb{P}[\mathbf{x}_\mathtt{a} = \mathbf{y}_\mathtt{a}\ \forall a \in \{1, \ldots, \mathtt{M}\}|\boldsymbol{\lambda}_:, \boldsymbol{\gamma}_:] = \prod_{a=1}^{\mathtt{M}} \frac{\sum_{i=0}^{2^\mathtt{N}-1} \exp(\mathbf{B}_\mathtt{i}^T(\boldsymbol{\lambda}_\mathtt{a} + \boldsymbol{\gamma}_\mathtt{a}))}{\sum_{i=0}^{2^\mathtt{N}-1} \exp(\mathbf{B}_\mathtt{i}^T \boldsymbol{\lambda}_\mathtt{a}) \sum_{i=0}^{2^\mathtt{N}-1} \exp(\mathbf{B}_\mathtt{i}^T \boldsymbol{\gamma}_\mathtt{a})} = c$$

This may be rewritten equivalently in the log domain

$$\sum_{a=1}^{\mathtt{M}} \psi_\mathbf{x}(\mathbf{B}(\boldsymbol{\lambda}_\mathtt{a} + \boldsymbol{\gamma}_\mathtt{a})) - \psi_\mathbf{x}(\mathbf{B}\boldsymbol{\lambda}_\mathtt{a}) - \psi_\mathbf{x}(\mathbf{B}\boldsymbol{\gamma}_\mathtt{a}) = \log(c)$$

In particular, if c is (nearly) one, then $\mathbf{x}_\mathtt{a}$ will be (nearly) equal to $\mathbf{y}_\mathtt{a}$ for each a, and thus we will not suffer from dropping h from the likelihood function.

With these definitions, we are ready to prove the following theorem.

Thm. 6 (Optimality of Belief Propagation Stationary Points)**:** The stationary points of belief propagation decoding solve the first order necessary conditions for the

following constrained maximum likelihood problem

$$(\boldsymbol{\lambda}^*_{:}, \boldsymbol{\gamma}^*_{:}) = \arg \max_{(\boldsymbol{\lambda}_{:}, \boldsymbol{\gamma}_{:}) \in \mathcal{C}} \log \left(p_{app}(\mathbf{r} | \boldsymbol{\lambda}_{:}, \boldsymbol{\gamma}_{:}) \right) \tag{4.54}$$

where the constraint set $\mathcal{C}$ is

$$\mathcal{C} = \{ (\boldsymbol{\lambda}_{:}, \boldsymbol{\gamma}_{:}) | \log \left(\mathbb{P}[\mathbf{x}_{\mathsf{a}} = \mathbf{y}_{\mathsf{a}} \ \forall a | \boldsymbol{\lambda}_{:}, \boldsymbol{\gamma}_{:}] \right) = \log(c) \ \forall a \} \tag{4.55}$$

subject to a Lagrange multiplier of -1.

Proof: A direct application of Theorem 3. ■

The expression (4.54) under maximization is the log likelihood function for choosing the input sequence if, for each constraint c_{a}, the input $\mathbf{y}_{\mathsf{a}}$ were chosen independently of the repetition $\mathbf{x}_{\mathsf{a}}$; these are parameterized by log probability ratios $\boldsymbol{\lambda}_{\mathsf{a}}$ and $\boldsymbol{\gamma}_{\mathsf{a}}$, respectively. The constraint set (4.55) may be viewed as enforcing a "sufficiently high" probability that the $\mathbf{y}_{\mathsf{a}}$ terms in fact agree, according to the constraint probability c.

Let us rehash the intuitive nature of the optimization. In calculating the likelihood function, we make a false assumption of independence between all of the $\mathbf{x}_{\mathsf{a}}$s and $\mathbf{y}_{\mathsf{a}}$s, but then we control this false independence assumption by restricting the probability that the $\mathbf{x}_{\mathsf{a}}$s and $\mathbf{y}_{\mathsf{a}}$s are equal to be constant, with values closer to one indicating higher confidence. What the result shows is that, if the belief propagation decoding converges to a stationary point $(\boldsymbol{\lambda}^*_{:}, \boldsymbol{\gamma}^*_{:})$, then this stationary point has a corresponding probability $\mathbb{P}[\mathbf{x}_{\mathsf{a}} = \mathbf{y}_{\mathsf{a}} \forall a | \boldsymbol{\lambda}^*_{:}, \boldsymbol{\gamma}^*_{:}] = c$, and this stationary point is a critical point of the likelihood function within the constraint space $\{ (\boldsymbol{\lambda}_{:}, \boldsymbol{\gamma}_{:}) | \log(\mathbb{P}[\mathbf{x}_{\mathsf{a}} = \mathbf{y}_{\mathsf{a}} \ \forall a | \boldsymbol{\lambda}_{:}, \boldsymbol{\gamma}_{:}]) = \log(c) \}$. Before the decoder has converged, however, the value of the constraint may vary widely.

Via the sensitivity interpretation of Lagrange multipliers [54], we may consider

choosing the Lagrange multiplier of -1 as indicating that it is equally important to us to have a large $\mathbb{P}[\mathbf{x}_{\mathtt{a}} = \mathbf{y}_{\mathtt{a}} \ \forall a | \boldsymbol{\lambda}_{:}, \boldsymbol{\gamma}_{:}] = c$ as to have a large $p_{app}(\mathbf{r}|\boldsymbol{\lambda}_{:}, \boldsymbol{\gamma}_{:})$.

Before interpreting this result further, it is important to reinterpret the quantities used in the belief propagation decoder for decisions in terms of the intuitive quantities in this theorem. Now, recall from Section 4.1 that the belief at node $\theta_{\mathtt{i}}$ at time k is given by

$$\log\left(\frac{\mathsf{q}_{\mathtt{i}}(1)}{\mathsf{q}_{\mathtt{i}}(0)}\right) = \boldsymbol{\lambda}_{n,a}^{(k)} + \boldsymbol{\gamma}_{a,n}^{(k+1)}$$

This allows us to reinterpret the belief as

$$\mathsf{q}_{\mathtt{i}}(x) = \mathbb{P}\left[\mathbf{x}_{a,n} = \mathbf{y}_{a,n} = x | \mathbf{x}_{\mathtt{a}} = \mathbf{y}_{\mathtt{a}}, \boldsymbol{\lambda}_{\mathtt{a}}, \boldsymbol{\gamma}_{\mathtt{a}}\right]$$

which is intuitively reasonable, since we know a priori that $\mathbf{x}_{\mathtt{a}} = \mathbf{y}_{\mathtt{a}}$, and so we want to exploit that in making our decisions. Now, also note that if $c = 1$, then all of the messages $\mathbf{x}_{\mathtt{a}}$ and $\mathbf{y}_{\mathtt{a}}$ are forced to be equal. Thus, if the belief propagation decoding converges to a stationary point which has $c = 1$, then it has converged to an extremum of the true blockwise likelihood function. Of course, the global extrema of the blockwise likelihood function is defined as the maximum likelihood sequence detector. Provided continuity of the solutions of the optimization problem with respect to c, then for c close to 1, we should have solutions close to extrema of the network-wide likelihood functions, and if they are initialized correctly, close to the network-wide maximum likelihood detector.

In the next section, we show for some key examples that when the belief propagation decoding algorithm is giving decisions on the network states with a low probability of error, c is close to 1. This then completes the argument that in instances where the belief propagation decoding algorithm is commonly used, that is, in regimes with a low observed probability of node-error, it is providing that

low probability of node-error by iteratively computing a close approximation to the network-wide maximum likelihood detector.

Simulations

We have seen in the preceding section that the belief propagation decoding maximizes an approximation to the true blockwise (belief network wide) likelihood function within a constraint set $\mathcal{C}$ which fixes the probability $\mathbb{P}[\mathbf{x}_\mathtt{i} = \mathbf{y}_\mathtt{i}]$ to a constant. Furthermore, we saw that if this constant is one then the approximation is exact within the constraint space. Here we give some simulated examples which show that the value of this constraint usually converges to 1 with successive iterations for SNRs which provide a low probability of bit error at the output of the decoder. This then suggests that in situations where the belief propagation decoder is likely to be used, that is, in situations where the decisions from the decoder provide a low probability of bit error, the belief propagation decoder is providing decisions which are close to the maximum likelihood blockwise detection. This then justifies the proper operation of the belief propagation decoding in instances (such as the one simulated) where there are loops in the belief graph.

Figure 4.7 depicts the results for the 10^3-block length optimized code from [19], with the exception that we did not avoid cycles when picking a random interleaver. One can easily discern that for low bit error rates the constraint probability $\mathbb{P}[\mathbf{x}_\mathtt{a} = \mathbf{y}_\mathtt{a} \forall a | \boldsymbol{\lambda}_{:,}, \boldsymbol{\gamma}_{:}]$ converges to one with successive iterations. The same phenomenon is shown for the 10^4 block length optimized code from [19] in Figure 4.8. We can conclude then, that the constraint probability converges closely to 1 with successive iterations, which suggests that the stationary points are close to the critical points of the real likelihood function within the constraint space.

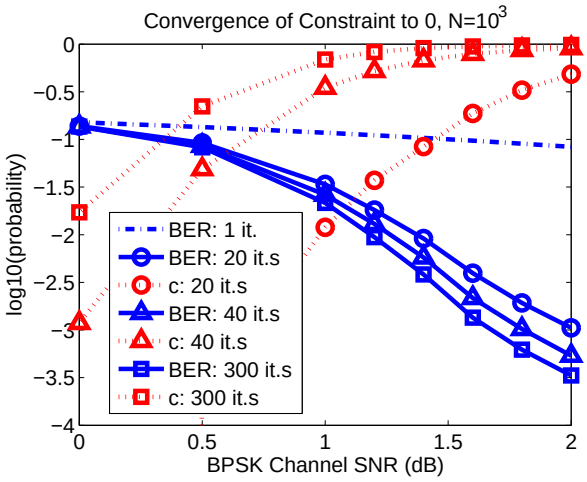

Figure 4.7: A $\mathtt{N} = 10^3$ optimized code from [19]. The value of the constraint near to 0 when the decoder is providing a low bit error rate, suggesting that the decoder is performing decisions close to the blockwise ml decisions.

Note that c is a quantity which is capable of being monitored at the decoder with low complexity calculations over the iterations, thus one could use it to predict convergence or use "c is close to 1" as a stopping criterion and an indicator for successful decoding, although we do not investigate this possibility here.

Conclusions

While the belief propagation decoding algorithm has been previously shown to converge to the true a posteriori probabilities in belief networks that are trees, many of the recent breakthrough applications of the algorithm, such as for the decoding of LDPC and turbo codes, involve belief networks with loops. While the belief propagation decoding stationary points have been shown to minimize the Bethe approximation to the variational free energy, and thus equal the true

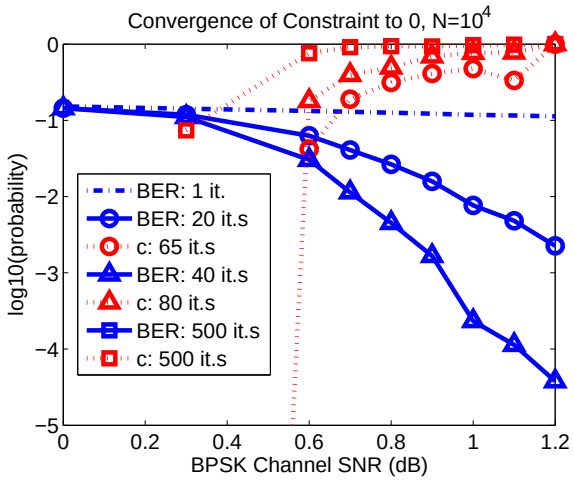

Figure 4.8: A $\mathtt{N} = 10^4$ optimized code from [19]. The value of the constraint near to 0 when the decoder is providing a low bit error rate, suggesting that the decoder is performing decisions close to the blockwise ml decisions.

bitwise a posteriori probabilities for factor graphs which are trees, this approximation is generally inexact for factor graphs which are not trees, and thus it is not entirely clear why minimizing it yields statistics which give decisions with good performance. Here, we have connected belief propagation decoding to constrained maximum likelihood sequence detection by showing that the stationary points of belief propagation decoding are critical points of an approximation to the likelihood function within a constraint space. For a constraint probability of 1, the approximation is exact, and we observed via simulation that the constraint probability converges to 1 with successive iterations in instances where the decoder is providing a low bit error rate. We argue that these results suggest that the belief propagation decoder is providing a low bit error rate in networks with loops because it is iteratively minimizing the probability of blockwise error.

4.3 Relation Between the Two Generic Optimality Frameworks

In the previous two sections we provided two different constrained optimization problems which yielded the stationary points of belief propagation as their critical points. It turns out that these two frameworks are intimately related to each other, because the Lagrangian of the constrained maximum likelihood framework is a reparameterization of the pseudo-dual to the constrained Bethe free energy. The intuitive justification provided for the constrained maximum likelihood framework then allows us to see the motivation for minimizing the Bethe free energy in contexts where there are loops in the statistics factor graph. Here we have used the (perhaps somewhat quirky) terminology

Def. 3 (Pseudo-Dual)**:** Consider the Lagrangian of a variational constrained functional optimization problem of the form (C.7) where each of the functionals is of the form (C.6). If, for each set of Lagrange multipliers $\boldsymbol{\mu}$, there is a unique function f^* which sets the variation equal to zero, we call the value of the Lagrangian at f^* regarded as a function of $\boldsymbol{\mu}$ the pseudo-dual function to the optimization problem (C.7).

We are now ready to prove the following theorem.

Thm. 7 (Pseudo-Duality of the Two Optimality Frameworks)**:** The constrained maximum likelihood optimization problem from Theorem 3 is a reparameterization of the negative of the pseudo-dual of the constrained Bethe free energy minimization problem from Theorem 2 within the set of $\{\mu_{\mathtt{a}}, \mu_{\mathtt{i}}\}$ that yield probability densities in (4.10) and (4.11) that integrate to one.

Proof: From (4.10) and (4.11), given a fixed set of multipliers $\boldsymbol{\mu}$, the variation

of the Lagrangian with respect to q_{a} and $\mathsf{q}_{\mathbf{v},\mathsf{i}}$ is equal to zero if and only if q_{a} and $\mathsf{q}_{\mathbf{v},\mathsf{i}}$ are of the form in (4.12) and (4.13). Substituting these $\mathsf{q}^*_{\mathsf{a}}$'s and $\mathsf{q}^*_{\mathbf{v},\mathsf{i}}$'s into the Lagrangian (4.9), then, gives us the pseudo-dual function

$$\begin{aligned}
\mathsf{d}(\boldsymbol{\mu}) \;=\; & \sum_{\mathsf{a}=1}^{\mathtt{M}} \int_{\Theta_{\mathsf{a}}} \mathsf{f}_{\mathsf{a}}(\boldsymbol{\theta}_{\mathsf{a}}) \exp\left(\sum_{\mathsf{i}\in\mathcal{EN}(\mathsf{a})} \boldsymbol{\mu}_{\mathsf{a},\mathsf{i}} \cdot \mathbf{v}_{\mathsf{i}}(\boldsymbol{\theta}_{\mathbf{v},\mathsf{i}}) + \mu_{\mathsf{a}} - 1 \right) \\
& \left(\sum_{\mathsf{i}\in\mathcal{EN}(\mathsf{a})} \boldsymbol{\mu}_{\mathsf{a},\mathsf{i}} \cdot \mathbf{v}_{\mathsf{i}}(\boldsymbol{\theta}_{\mathbf{v},\mathsf{i}}) + \mu_{\mathsf{a}} - 1 \right) \mathsf{d}\boldsymbol{\theta}_{\mathsf{a}} \\
& - \sum_{\mathsf{i}=1}^{\mathtt{V}} (|\mathcal{N}(\mathsf{i})| - 1) \int_{\Theta_{\mathbf{v},\mathsf{i}}} \exp\left(\frac{\sum_{\mathsf{a}\in\mathcal{N}(\mathsf{i})} \boldsymbol{\mu}_{\mathsf{a},\mathsf{i}} \cdot \mathbf{v}_{\mathsf{i}}(\boldsymbol{\theta}_{\mathbf{v},\mathsf{i}}) - \mu_{\mathbf{v},\mathsf{i}}}{|\mathcal{N}(\mathsf{i})| - 1} - 1 \right) \\
& \left(\frac{\sum_{\mathsf{a}\in\mathcal{N}(\mathsf{i})} \boldsymbol{\mu}_{\mathsf{a},\mathsf{i}} \cdot \mathbf{v}_{\mathsf{i}}(\boldsymbol{\theta}_{\mathbf{v},\mathsf{i}}) - \mu_{\mathbf{v},\mathsf{i}}}{|\mathcal{N}(\mathsf{i})| - 1} - 1 \right) \mathsf{d}\boldsymbol{\theta}_{\mathbf{v},\mathsf{i}} \\
& - \sum_{\mathsf{a}=1}^{\mathtt{M}} \mu_{\mathsf{a}} \left(\int_{\Theta_{\mathsf{a}}} \mathsf{f}_{\mathsf{a}}(\boldsymbol{\theta}_{\mathsf{a}}) \exp\left(\sum_{\mathsf{i}\in\mathcal{EN}(\mathsf{a})} \boldsymbol{\mu}_{\mathsf{a},\mathsf{i}} \cdot \mathbf{v}_{\mathsf{i}}(\boldsymbol{\theta}_{\mathbf{v},\mathsf{i}}) + \mu_{\mathsf{a}} - 1 \right) \mathsf{d}\boldsymbol{\theta}_{\mathsf{a}} - 1 \right) \\
& - \sum_{\mathsf{i}=1}^{\mathtt{V}} \mu_{\mathbf{v},\mathsf{i}} \left(\int_{\Theta_{\mathbf{v},\mathsf{i}}} \exp\left(\frac{\sum_{\mathsf{a}\in\mathcal{N}(\mathsf{i})} \boldsymbol{\mu}_{\mathsf{a},\mathsf{i}} \cdot \mathbf{v}_{\mathsf{i}}(\boldsymbol{\theta}_{\mathbf{v},\mathsf{i}}) - \mu_{\mathbf{v},\mathsf{i}}}{|\mathcal{N}(\mathsf{i})| - 1} - 1 \right) \mathsf{d}\boldsymbol{\theta}_{\mathbf{v},\mathsf{i}} - 1 \right) \\
& - \sum_{\mathsf{i}=1}^{\mathtt{V}} \sum_{\mathsf{a}\in\mathcal{N}(\mathsf{i})} \boldsymbol{\mu}_{\mathsf{a},\mathsf{i}} \cdot \left(\mathbb{E}_{\mathsf{q}_{\mathsf{a}}}\left[\mathbf{v}_{\mathsf{i}}(\boldsymbol{\theta}_{\mathbf{v},\mathsf{i}})\right] - \mathbb{E}_{\mathsf{q}_{\mathbf{v},\mathsf{i}}}\left[\mathbf{v}_{\mathsf{i}}(\boldsymbol{\theta}_{\mathbf{v},\mathsf{i}})\right] \right)
\end{aligned} \tag{4.56}$$

Simplifying this, we have

$$\begin{aligned}
\mathsf{d}(\boldsymbol{\mu}) \;=\; & - \sum_{\mathsf{a}=1}^{\mathtt{M}} \int_{\Theta_{\mathsf{a}}} \mathsf{f}_{\mathsf{a}}(\boldsymbol{\theta}_{\mathsf{a}}) \exp\left(\sum_{\mathsf{i}\in\mathcal{EN}(\mathsf{a})} \boldsymbol{\mu}_{\mathsf{a},\mathsf{i}} \cdot \mathbf{v}_{\mathsf{i}}(\boldsymbol{\theta}_{\mathbf{v},\mathsf{i}}) + \mu_{\mathsf{a}} - 1 \right) \mathsf{d}\boldsymbol{\theta}_{\mathsf{a}} \\
& + \sum_{\mathsf{i}=1}^{\mathtt{V}} (|\mathcal{N}(\mathsf{i})| - 1) \int_{\Theta_{\mathbf{v},\mathsf{i}}} \exp\left(\frac{\sum_{\mathsf{a}\in\mathcal{N}(\mathsf{i})} \boldsymbol{\mu}_{\mathsf{a},\mathsf{i}} \cdot \mathbf{v}_{\mathsf{i}}(\boldsymbol{\theta}_{\mathbf{v},\mathsf{i}}) - \mu_{\mathbf{v},\mathsf{i}}}{|\mathcal{N}(\mathsf{i})| - 1} - 1 \right) \mathsf{d}\boldsymbol{\theta}_{\mathbf{v},\mathsf{i}} \\
& + \sum_{\mathsf{a}=1}^{\mathtt{M}} \mu_{\mathsf{a}} + \sum_{\mathsf{i}=1}^{\mathtt{V}} \mu_{\mathbf{v},\mathsf{i}}
\end{aligned} \tag{4.57}$$

It is particularly of interest to consider the value of this pseudo-dual function within the constraint space $\mathcal{C}_{\boldsymbol{\mu}}$ of multipliers $\boldsymbol{\mu}$ which via (4.12) and (4.13) give distributions $\{\mathsf{q}_{\mathsf{a}}\}$ and $\{\mathsf{q}_{\mathsf{i}}\}$ that are probability distributions which integrate to

one. Within this constraint space, the pseudo-dual function simplifies to

$$\begin{aligned}
\mathsf{d}(\boldsymbol{\mu})|_{\boldsymbol{\mu}\in\mathcal{C}_\mu} &= -\mathtt{M} + \sum_{\mathtt{i}=1}^{\mathtt{V}} (|\mathcal{N}(\mathtt{i})| - 1) + \sum_{\mathtt{a}=1}^{\mathtt{M}} \mu_{\mathtt{a}} + \sum_{\mathtt{i}=1}^{\mathtt{V}} \mu_{\mathbf{v},\mathtt{i}} && (4.58)\\
&= -\mathtt{M} + \sum_{\mathtt{i}=1}^{\mathtt{V}} (|\mathcal{N}(\mathtt{i})| - 1) + \sum_{\mathtt{i}=1}^{\mathtt{V}} (|\mathcal{N}(\mathtt{i})| - 1) \\
&\quad \log\left(\int_{\Theta_{\mathbf{v},\mathtt{i}}} \exp\left(\frac{\sum_{\mathtt{a}\in\mathcal{N}(\mathtt{i})} \boldsymbol{\mu}_{\mathtt{a},\mathtt{i}} \cdot \mathbf{v}_{\mathtt{i}}(\boldsymbol{\theta}_{\mathbf{v},\mathtt{i}})}{|\mathcal{N}(\mathtt{i})| - 1} - 1 \right) \mathsf{d}\boldsymbol{\theta}_{\mathbf{v},\mathtt{i}} \right) \\
&\quad - \sum_{\mathtt{a}=1}^{\mathtt{M}} \log\left(\int_{\Theta_{\mathtt{a}}} \mathsf{f}_{\mathtt{a}}(\boldsymbol{\theta}_{\mathtt{a}}) \exp\left(\sum_{\mathtt{i}\in\mathcal{EN}(\mathtt{a})} \boldsymbol{\mu}_{\mathtt{a},\mathtt{i}} \cdot \mathbf{v}_{\mathtt{i}}(\boldsymbol{\theta}_{\mathbf{v},\mathtt{i}}) - 1 \right) \mathsf{d}\boldsymbol{\theta}_{\mathtt{a}} \right) && (4.59)\\
&= \sum_{\mathtt{i}=1}^{\mathtt{V}} (|\mathcal{N}(\mathtt{i})| - 1) \log\left(\int_{\Theta_{\mathbf{v},\mathtt{i}}} \exp\left(\frac{\sum_{\mathtt{a}\in\mathcal{N}(\mathtt{i})} \boldsymbol{\mu}_{\mathtt{a},\mathtt{i}} \cdot \mathbf{v}_{\mathtt{i}}(\boldsymbol{\theta}_{\mathbf{v},\mathtt{i}})}{|\mathcal{N}(\mathtt{i})| - 1} \right) \mathsf{d}\boldsymbol{\theta}_{\mathbf{v},\mathtt{i}} \right) \\
&\quad - \sum_{\mathtt{a}=1}^{\mathtt{M}} \log\left(\int_{\Theta_{\mathtt{a}}} \mathsf{f}_{\mathtt{a}}(\boldsymbol{\theta}_{\mathtt{a}}) \exp\left(\sum_{\mathtt{i}\in\mathcal{EN}(\mathtt{a})} \boldsymbol{\mu}_{\mathtt{a},\mathtt{i}} \cdot \mathbf{v}_{\mathtt{i}}(\boldsymbol{\theta}_{\mathbf{v},\mathtt{i}}) \right) \mathsf{d}\boldsymbol{\theta}_{\mathtt{a}} \right) && (4.60)
\end{aligned}$$

It turns out that this is related to the Lagrangian (4.20) for the optimization problem in Section 4.2. In particular, if we substitute the relation

$$[\boldsymbol{\gamma}_{\mathtt{a}}]_{\mathtt{i}} := \sum_{\mathtt{c}\in\mathcal{N}(\mathtt{i})\setminus\{\mathtt{a}\}} [\boldsymbol{\lambda}_{\mathtt{a}}]_{\mathtt{i}} \quad \forall \mathtt{i} \in \mathcal{N}(\mathtt{a}) \quad \forall \mathtt{a} \in \{1, \ldots, \mathtt{M}\}$$

which, incidentally, solves $\nabla_{\boldsymbol{\lambda}_{\mathtt{a}}} \mathsf{L} := \mathbf{0}$ in terms of $\boldsymbol{\gamma}_{\mathtt{a}}$, then (4.20) becomes

$$\begin{aligned}
\mathsf{L} &= \sum_{\mathtt{a}=1}^{\mathtt{M}} \log\left(\int_{\Theta} \mathsf{f}_{\mathtt{a}}(\mathbf{y}_{\mathtt{a}}) \exp\left(\mathbf{u}_{\mathtt{a}}(\mathbf{y}_{\mathtt{a}}) \cdot \boldsymbol{\gamma}_{\mathtt{a}} \right) \mathsf{d}\mathbf{y}_{\mathtt{a}} \right) \\
&\quad - \sum_{\mathtt{i}=1}^{\mathtt{V}} (|\mathcal{N}(\mathtt{i})| - 1) \log\left(\int_{\Theta_{\mathbf{v},\mathtt{i}}} \exp\left(\mathbf{v}_{\mathtt{i}}(\boldsymbol{\theta}_{\mathbf{v},\mathtt{i}}) \cdot \sum_{\mathtt{a}\in\mathcal{N}(\mathtt{i})} [\boldsymbol{\lambda}_{\mathtt{a}}]_{\mathtt{i}} \right) \mathsf{d}\boldsymbol{\theta}_{\mathbf{v},\mathtt{i}} \right) && (4.61)\\
&= \sum_{\mathtt{a}=1}^{\mathtt{M}} \log\left(\int_{\Theta} \mathsf{f}_{\mathtt{a}}(\mathbf{y}_{\mathtt{a}}) \exp\left(\mathbf{u}_{\mathtt{a}}(\mathbf{y}_{\mathtt{a}}) \cdot \boldsymbol{\gamma}_{\mathtt{a}} \right) \mathsf{d}\mathbf{y}_{\mathtt{a}} \right) \\
&\quad - \sum_{\mathtt{i}=1}^{\mathtt{V}} (|\mathcal{N}(\mathtt{i})| - 1) \log\left(\int_{\Theta_{\mathbf{v},\mathtt{i}}} \exp\left(\mathbf{v}_{\mathtt{i}}(\boldsymbol{\theta}_{\mathbf{v},\mathtt{i}}) \cdot \sum_{\mathtt{a}\in\mathcal{N}(\mathtt{i})} [\boldsymbol{\gamma}_{\mathtt{a}}]_{\mathtt{i}} \right) \mathsf{d}\boldsymbol{\theta}_{\mathbf{v},\mathtt{i}} \right) && (4.62)
\end{aligned}$$

Then, if we identify

$$\boldsymbol{\gamma}_{\mathtt{a},\mathtt{i}} := \boldsymbol{\mu}_{\mathtt{a},\mathtt{i}} \quad \forall \mathtt{i} \in \mathcal{EN}(\mathtt{a})$$

we see that (4.62) is equal to the negative of (4.60). ■

One might wonder why it is necessary to distinguish between a pseudo-dual and a dual. This has to due with the necessity, but not sufficiency, of the requirement that the variation of the Lagrangian be equal to zero to have the unique infimum of the Lagrangian regarded as a functional of f (i.e. with fixed Lagrange multipliers $\boldsymbol{\mu}$). Even though the Lagrangian might have a unique function which makes its variation zero, this function does not necessarily attain the infimum of the Lagrangian, that is, it is not necessarily the minimum. We can conjure up at least two examples here via analogy to basic calculus. First of all, there is the possibility that the infimum is $-\infty$ (take, for instance x^3 which has zero derivative only at $x = 0$ but is unbounded below). Second of all, there is the possibility that the infimum is finite but not attained (take for instance $\exp(-x^3)$ which has zero derivative only at $x = 0$ and which has an infimum of zero which no real number x can attain). On the other hand, we can be confident that if the infimum is both finite and attained, then a unique solution to the first order variation conditions will attain the infimum.

Prop. 4 (Duality of the Two Optimality Frameworks)**:** Let $\mathcal{A}$ be the set of Lagrange multipliers $\boldsymbol{\mu}$ for which the infimum of the Lagrangian of the constrained Bethe free energy with respect to $\mathbf{q}_{\mathscr{R}_{\text{Bethe}}}$ is finite and attained and for which the pdfs in $\mathbf{q}_{\mathscr{R}_{\text{Bethe}}}$ integrate to unity. Then, for $\boldsymbol{\mu} \in \mathcal{A}$ the Lagrangian of the constrained maximum likelihood optimization problem is the negative of the dual of the constrained Bethe free energy.

Proof: When $\boldsymbol{\mu} \in \mathcal{A}$ the unique solution to the first order necessary conditions that we found must attain the infimum, erasing the difference between the pseudo-dual and the dual. ■

Chapter 5

Convergence Frameworks

In this chapter we introduce two novel convergence frameworks for the family of expectation propagation algorithms. We first show how both the serial and parallel update schedules of the expectation propagation algorithm are nonlinear block Gauss Seidel iterations on the first order necessary conditions from the constrained maximum likelihood optimization problem in Section 4.2. This allows us to extract two convergence theorems from the numerical analysis literature and apply them to expectation propagation. We then take the theory that results and apply it to the turbo decoder, meeting with mixed practical success. Starting anew, we next introduce another convergence framework for the expectation propagation family of algorithms via analogy to Dykstra's algorithm to solve a convex feasibility problem with iterated Bregman projections. We then discuss some of the possibilities of convergence theories based on this framework. We recommend the reader not well versed in numerical analysis and the canonical representations of exponential families read Appendices A and D.4 before attacking this section.

5.1 Iterative Numerical Methods

In this section we explain how expectation propagation algorithms may be viewed as iterative numerical methods bent on finding the critical points of the generic constrained maximum likelihood optimization problem and the statistics based Bethe free energy minimization problem discussed in the previous chapter.

5.1.1 Common Iterative Methods of Solution

Of central interest in this section, due to their relevance to expectation propagation, will be various iterative numerical methods for solving systems of nonlinear equations. In particular, suppose we have a system of equations described by

$$\mathbf{h}(\mathbf{x}) = \mathbf{0} \tag{5.1}$$

where $\mathbf{h} : \mathbb{R}^{\mathtt{N}} \rightarrow \mathbb{R}^{\mathtt{N}}$ is a vector valued function of $\mathbf{x} \in \mathbb{R}^{\mathtt{N}}$. Suppose further that we had a method which was capable of solving subsets of the system of equations for subsets of the variables. In particular, divide up the equations in the system into groups by rewriting

$$\mathbf{h}(\mathbf{x}) = \left[\mathbf{h}_1^T(\mathbf{x}), \mathbf{h}_2^T(\mathbf{x}), \cdots, \mathbf{h}_{\mathtt{M}}^T(\mathbf{x})\right]^T$$

and divide the variables to solve for up into the same groups $\mathbf{x} := [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_{\mathtt{M}}]$, so that $\mathbf{x}_{\mathtt{i}} \in \mathbb{R}^{\mathtt{N}_{\mathtt{i}}}$ and $\mathbf{h}_{\mathtt{i}}^T : \mathbb{R}^{\mathtt{N}} \rightarrow \mathbb{R}^{\mathtt{N}_{\mathtt{i}}}$ for all $\mathtt{i} \in \{1, \ldots, \mathtt{M}\}$ and $\sum_{\mathtt{i}} \mathtt{N}_{\mathtt{i}} = \mathtt{N}$. Suppose that we are capable of solving

$$\mathbf{h}_{\mathtt{i}}(\mathbf{x}) = \mathbf{0} \tag{5.2}$$

for $\mathbf{x}_{\mathtt{i}}$ in terms of $\mathbf{x} \setminus \mathbf{x}_{\mathtt{i}} := \left[\mathbf{x}_1^T, \cdots, \mathbf{x}_{\mathtt{i}-1}^T, \mathbf{x}_{\mathtt{i}+1}^T, \cdots, \mathbf{x}_{\mathtt{M}}^T\right]^T$. That is, suppose that for each $\mathtt{i} \in \{1, \ldots, \mathtt{M}\}$ we had a function $\mathbf{u}_{\mathtt{i}}(\mathbf{x} \setminus \mathbf{x}_{\mathtt{i}})$ which satisfied

$$\mathbf{h}_{\mathtt{i}}(\mathbf{x}_1, \cdots, \mathbf{x}_{\mathtt{i}-1}, \mathbf{u}_{\mathtt{i}}(\mathbf{x} \setminus \mathbf{x}_{\mathtt{i}}), \mathbf{x}_{\mathtt{i}+1}, \cdots, \mathbf{x}_{\mathtt{M}}) = \mathbf{0} \quad \forall \mathbf{x} \setminus \mathbf{x}_{\mathtt{i}}$$

There are several traditional methods of solving the original system of equations (5.1) using the implicit functions $\mathbf{u}_{\mathtt{i}}$. For brevity, we mention only two here, the Gauss Seidel method and the Jacobi method, and expound on the connections between expectation propagation and the Gauss Seidel method. The Gauss-Seidel procedure has been discussed most widely for linear systems of equations [55, pp.

480–483] , [56, pp. 251–257], [57, pp. 510–511], but can be generalized to nonlinear systems of equations as well [16, pp. 131–133; pp. 185–197], [15, p. 225], the most generalized form being referred to as a *nonlinear block Gauss Seidel method*. In a nonlinear block Gauss Seidel method we attempt to solve the system by iterating

$$\mathbf{x}_{\mathtt{i}}^{(\mathtt{k}+1)} = \mathbf{u}_{\mathtt{i}}(\mathbf{x}_{1}^{(\mathtt{k}+1)}, \cdots, \mathbf{x}_{\mathtt{i}-1}^{(\mathtt{k}+1)}, \mathbf{x}_{\mathtt{i}+1}^{(\mathtt{k})}, \cdots, \mathbf{x}_{\mathtt{M}}^{(\mathtt{k})}) \quad \forall \mathtt{i} \in \{1, \ldots, \mathtt{M}\}$$

and in a nonlinear block Jacobi method we attempt to solve the system by iterating

$$\mathbf{x}_{\mathtt{i}}^{(\mathtt{k}+1)} = \mathbf{u}_{\mathtt{i}}(\mathbf{x}_{1}^{(\mathtt{k})}, \cdots, \mathbf{x}_{\mathtt{i}-1}^{(\mathtt{k})}, \mathbf{x}_{\mathtt{i}+1}^{(\mathtt{k})}, \cdots, \mathbf{x}_{\mathtt{M}}^{(\mathtt{k})}) \quad \forall \mathtt{i} \in \{1, \ldots, \mathtt{M}\}$$

We will now see how the Gauss Seidel iterative method corresponds to the two different schedules of the expectation propagation algorithm for the system of equations which are the first order necessary conditions of Lagrange for our constrained optimization problem.

Prop. 5 (Parallel Expectation Propagation as Gauss Seidel)**:** Expectation propagation under parallel scheduling is equivalent to a nonlinear block Gauss Seidel iteration attempting to zero the gradient of the Lagrangian to the constrained maximum likelihood optimization problem from Section 4.2 with

$$\mathbf{x}_1 := \boldsymbol{\lambda} \quad \mathbf{x}_2 := \boldsymbol{\gamma}$$

and

$$\mathbf{h}_1 := \nabla_{\boldsymbol{\gamma}} \mathsf{L} \quad \mathbf{h}_2 := \nabla_{\boldsymbol{\lambda}} \mathsf{L}$$

Proof: In the proof for Theorem 3, we showed that solving $\nabla_{\boldsymbol{\lambda}_\mathsf{a}} \mathsf{L} = \mathbf{0}$ is equivalent to updating the messages $\boldsymbol{\gamma}_\mathsf{a}$ from the statistics nodes to the factor nodes, and solving $\nabla_{\boldsymbol{\gamma}_\mathsf{a}} \mathsf{L} = \mathbf{0}$ is equivalent to updating the messages $\boldsymbol{\lambda}_\mathsf{a}$ from the factor nodes

to the statistics nodes. Thus, scheduling the solutions in this fashion updates the statistics to factor node messages all at one, and the factor to statistics messages all at once. This is what is referred to as parallel scheduling. ■

Prop. 6 (Serial Expectation Propagation as Gauss Seidel)**:** Expectation propagation under serial scheduling is equivalent to a nonlinear block Gauss Seidel iteration attempting to zero the gradient of the Lagrangian to the constrained maximum likelihood optimization problem from Section 4.2 with

$$\mathbf{x}_{2\mathsf{a}-1} := \boldsymbol{\lambda}_{\mathsf{a}} \quad \mathbf{x}_{2\mathsf{a}} := \boldsymbol{\gamma}_{\mathsf{a}} \quad \forall \mathsf{a} \in \{1, \ldots, \mathtt{M}\}$$

and

$$\mathbf{h}_{2\mathsf{a}-1} := \nabla_{\boldsymbol{\gamma}_{\mathsf{a}}} \mathsf{L} \quad \mathbf{h}_{2\mathsf{a}} := \nabla_{\boldsymbol{\lambda}_{\mathsf{a}}} \mathsf{L}$$

Proof: In the proof for Theorem 3, we showed that solving $\nabla_{\boldsymbol{\lambda}_{\mathsf{a}}} \mathsf{L} = \mathbf{0}$ is equivalent to updating the messages $\boldsymbol{\gamma}_{\mathsf{a}}$ from the statistics nodes to the factor nodes, and solving $\nabla_{\boldsymbol{\gamma}_{\mathsf{a}}} \mathsf{L} = \mathbf{0}$ is equivalent to updating the messages $\boldsymbol{\lambda}_{\mathsf{a}}$ from the factor nodes to the statistics nodes. Thus, scheduling the solutions in this fashion updates the statistics to factor node a messages, followed by the factor a to statistics messages next. This is what is referred to as serial scheduling. ■

We should also mention that we were not the only ones to note the correspondence between belief propagation (a special case of expectation propagation) and a nonlinear block Gauss Seidel method on the gradient of (a function related to) the dual to the Bethe free energy. Just before we presented our result about the constrained optimization framework [58] for turbo decoding, and at the same conference (ISIT 2005) as our original result concerning the Gauss Seidel iteration [59], [60] was also presented. In [60] and [61], the authors noted the relevance of the Gauss Seidel iterations and the Jacobi iterations on the dual of the Bethe free en-

ergy, providing some elementary (but in the latter instance, incorrect) convergence results.

The convergence of nonlinear Gauss Seidel methods received some significant attention in the numerical analysis literature during the early 1970s. Relevant references include [62], [63], and [64]. Our next theorem is an application of a theorem from [64], and the terminology used in it is included in the appendix A.

Thm. 8 (Convergence of Expectation Propagation Algorithms (Mixed Parameter Spaces))**:** If, when regarded as a function of $[\boldsymbol{\lambda}^T, \boldsymbol{\gamma}^T]^T$, the function $[(\nabla_{\boldsymbol{\lambda}}\mathsf{L})^T, (\nabla_{\boldsymbol{\gamma}}\mathsf{L})^T]^T$ is an m-function and is surjective onto $\mathbb{R}^{2\mathsf{V}}$ the expectation propagation algorithm converges to the unique solution of $\nabla\mathsf{L} = 0$.

Proof: This is just Theorem 19 from Appendix A with (4.21)= $\mathbf{0}$ and (4.22)= $\mathbf{0}$ as the system to be solved. ■

One of the difficulties prohibiting direct application of (8) to practical instances of the expectation propagation algorithms is that the gradients $\nabla\mathsf{L}$ are usually not surjective onto $\mathbb{R}^{2\mathsf{V}}$. In binary decoding problems, for instance, $\nabla\mathsf{L}$ is a difference between two vectors of bitwise marginal probabilities and is therefore necessarily limited to the range $[-1, 1]$. One way around this, taken in Section 5.1.2, the so called "application" of this theory to the turbo decoder, is to attempt to change the theorem and proof a bit to ensure that the algorithm remains within a set within which the $\nabla\mathsf{L}$ is an m-function. This method, however, has not met with a great deal of success, since the conditions that results are confusing at best, and certainly hard to check in anything even remotely resembling a practical instance of the algorithm.

Another way around this difficulty is to work with the natural parameters $\boldsymbol{\lambda}$

instead of the expectation parameters $\mathbb{E}[\mathbf{t}]$. In particular, since we used minimal representations we can use the inverse of the one to one map (discussed in Appendix D) between $\boldsymbol{\lambda}$ and $\mathbb{E}[\mathbf{t}]$, call it $\boldsymbol{\Lambda}$, and rewrite (4.21) and (4.22) as

$$(\boldsymbol{\lambda}_a + \boldsymbol{\gamma}_a) - \boldsymbol{\Lambda}\left(\frac{\int_\Theta \mathbf{t}_a(\boldsymbol{\theta}_a)\exp\left(\sum_{c=1}^{M}\mathbf{t}_c(\boldsymbol{\theta}_c)\cdot\boldsymbol{\lambda}_c\right)d\boldsymbol{\theta}}{\int_\Theta \exp\left(\sum_{c=1}^{M}\mathbf{t}_c(\boldsymbol{\theta}_c)\cdot\boldsymbol{\lambda}_c\right)d\boldsymbol{\theta}}\right) = \mathbf{0} \tag{5.3}$$

$$(\boldsymbol{\lambda}_a + \boldsymbol{\gamma}_a) - \boldsymbol{\Lambda}\left(\frac{\int_\Theta \mathbf{u}_a(\mathbf{y}_a)f_a(\mathbf{y}_a)\exp\left(\mathbf{u}_a(\mathbf{y}_a)\cdot\boldsymbol{\gamma}_a\right)d\mathbf{x}_a}{\int_\Theta f_a(\mathbf{y}_a)\exp\left(\mathbf{u}_a(\mathbf{y}_a)\cdot\boldsymbol{\gamma}_a\right)d\mathbf{x}_a}\right) = \mathbf{0} \tag{5.4}$$

Thm. 9 (Convergence of Expectation Propagation Algorithms (Single Parameter Space))**:** If, when regarded as a function of $[\boldsymbol{\lambda}^T, \boldsymbol{\gamma}^T]^T$, the vector function set equal to zero in the system of equations (5.3) and (5.4) is an m-function, continuous, and surjective (onto) $\mathbb{R}^{2V}$, the expectation propagation algorithm converges to the unique solution of (5.3) and (5.4) and thus the unique interior critical point of the constrained maximum likelihood optimization problem.

Proof: This is just theorem 19 with (5.3) and (5.4) as the system to be solved. ■

This second theorem allows for somewhat neater application to the turbo decoder in 5.1.3.

5.1.2 Application to Turbo Decoding

In this section we apply Prop. 5 and Thm. 8 to the turbo decoder using the slightly differing formulation from Theorem 4 and a particular set of conditions under which a function is a m-function.

Thm. 10 (Turbo Decoder Dynamics)**:** The turbo decoder is exactly a nonlinear block Gauss Seidel iteration bent on finding the solution to the constrained optimization

problem

$$(\boldsymbol{\alpha}^*, \boldsymbol{\beta}^*) = \arg \max_{(\boldsymbol{\alpha},\boldsymbol{\beta})\in\mathcal{C}} \log(p(\mathbf{r}|\boldsymbol{\alpha}, \boldsymbol{\beta}))$$

In particular, the turbo decoder stationary points are in a one to one correspondence with the critical points of the Lagrangian of this optimization problem with a Lagrange multiplier of -1. The turbo decoder is then a nonlinear block Gauss Seidel iteration on the gradient of this Lagrangian.

Proof: Form the Lagrangian

$$\begin{aligned}
\mathsf{L}(\boldsymbol{\alpha}, \boldsymbol{\beta}) &= \log(p(\mathbf{r}|\boldsymbol{\alpha}, \boldsymbol{\beta})) + \mu \left(\log\left(\mathbb{P}\left[\mathbf{x} = \mathbf{y}|\boldsymbol{\alpha}, \boldsymbol{\beta}\right]\right) - \log(c)\right) && (5.5)\\
&= -\psi_{\mathbf{x}}(\mathbf{B}\boldsymbol{\alpha}) - \psi_{\mathbf{x}}(\mathbf{B}\boldsymbol{\beta}) + \psi_{\mathbf{x}}(\mathbf{B}\boldsymbol{\alpha} + \boldsymbol{\rho}_0) + \psi_{\mathbf{x}}(\mathbf{B}\boldsymbol{\beta} + \boldsymbol{\rho}_1) && (5.6)\\
&\quad +\mu \left(-\psi_{\mathbf{x}}(\mathbf{B}\boldsymbol{\alpha}) - \psi_{\mathbf{x}}(\mathbf{B}\boldsymbol{\beta}) + \psi_{\mathbf{x}}(\mathbf{B}(\boldsymbol{\alpha} + \boldsymbol{\beta}))\right) && (5.7)
\end{aligned}$$

Take its gradient

$$\begin{aligned}
\nabla_{\boldsymbol{\alpha}}\mathsf{L} &= -\mathbf{p}_{\mathbf{B}\boldsymbol{\alpha}} + \mathbf{p}_{\mathbf{B}\boldsymbol{\alpha}+\boldsymbol{\rho}_0} + \mu\left(-\mathbf{p}_{\mathbf{B}\boldsymbol{\alpha}} + \mathbf{p}_{\mathbf{B}(\boldsymbol{\alpha}+\boldsymbol{\beta})}\right) && (5.8)\\
\nabla_{\boldsymbol{\beta}}\mathsf{L} &= -\mathbf{p}_{\mathbf{B}\boldsymbol{\beta}} + \mathbf{p}_{\mathbf{B}\boldsymbol{\beta}+\boldsymbol{\rho}_1} + \mu\left(-\mathbf{p}_{\mathbf{B}\boldsymbol{\beta}} + \mathbf{p}_{\mathbf{B}(\boldsymbol{\alpha}+\boldsymbol{\beta})}\right) && (5.9)
\end{aligned}$$

Selecting a Lagrange multiplier of $\mu = -1$, we have

$$\begin{aligned}
\nabla_{\boldsymbol{\alpha}}\mathsf{L} &= \mathbf{p}_{\mathbf{B}\boldsymbol{\alpha}+\boldsymbol{\rho}_0} - \mathbf{p}_{\mathbf{B}(\boldsymbol{\alpha}+\boldsymbol{\beta})} && (5.10)\\
\nabla_{\boldsymbol{\beta}}\mathsf{L} &= \mathbf{p}_{\mathbf{B}\boldsymbol{\beta}+\boldsymbol{\rho}_1} - \mathbf{p}_{\mathbf{B}(\boldsymbol{\alpha}+\boldsymbol{\beta})} && (5.11)
\end{aligned}$$

If we break this system of equations into two parts

$$\begin{aligned}
\mathsf{F}_0(\boldsymbol{\alpha}, \boldsymbol{\beta}) &= \nabla_{\boldsymbol{\alpha}}\mathsf{L} = \mathbf{p}_{\mathbf{B}\boldsymbol{\alpha}+\boldsymbol{\rho}_0} - \mathbf{p}_{\mathbf{B}(\boldsymbol{\alpha}+\boldsymbol{\beta})} && (5.12)\\
\mathsf{F}_1(\boldsymbol{\alpha}, \boldsymbol{\beta}) &= \nabla_{\boldsymbol{\beta}}\mathsf{L} = \mathbf{p}_{\mathbf{B}\boldsymbol{\beta}+\boldsymbol{\rho}_1} - \mathbf{p}_{\mathbf{B}(\boldsymbol{\alpha}+\boldsymbol{\beta})} && (5.13)
\end{aligned}$$

Then we can see that the turbo decoder solves F_0 for $\boldsymbol{\alpha}$ given a fixed $\boldsymbol{\beta} = \boldsymbol{\beta}^{(k)}$ to

get $\boldsymbol{\alpha}^{(k)}$, and then solves F_1 for $\boldsymbol{\beta}$ given a fixed $\boldsymbol{\alpha} = \boldsymbol{\alpha}^{(k)}$ to get $\boldsymbol{\beta}^{(k+1)}$, that is

$$\boldsymbol{\alpha}^{(k)} = \boldsymbol{\alpha} \text{ such that } \mathsf{F}_0(\boldsymbol{\alpha}, \boldsymbol{\beta}^{(k)}) = \mathbf{0} \tag{5.14}$$

$$\boldsymbol{\beta}^{(k+1)} = \boldsymbol{\beta} \text{ such that } \mathsf{F}_1(\boldsymbol{\alpha}^{(k)}, \boldsymbol{\beta}) = \mathbf{0} \tag{5.15}$$

This is exactly the form of a nonlinear block Gauss Seidel iteration. Furthermore, the system of equations it is trying to solve are the necessary conditions for finding a solution of the provided constrained optimization problem, which are

$$\nabla_{\boldsymbol{\alpha}} \mathsf{L} = \mathbf{0} \tag{5.16}$$

$$\nabla_{\boldsymbol{\beta}} \mathsf{L} = \mathbf{0} \tag{5.17}$$

subject to a Lagrange multiplier of $\mu = -1$. ■

From here on, to compact notation, and to continue relation with the F_0 and F_1 notation used in the proof of Theorem 10, we will denote $\nabla \mathsf{L}$ by F. Here we will elaborate on the connection to the Gauss Seidel iteration, and use it to gain some convergence conditions for the turbo decoder. We will use the componentwise ordering, so that $\mathbf{x} \leq \mathbf{y} \iff x_i \leq y_i \forall i$.

Thm. 11 (Region of Convergence for the Turbo Decoder)**:** Define the measures p whose θ coordinates are $\mathbf{B}\boldsymbol{\lambda} + \boldsymbol{\rho}_0$, q whose θ coordinates are $\mathbf{B}\boldsymbol{\gamma} + \boldsymbol{\rho}_1$, and r whose θ

coordinates are $\mathbf{B}(\boldsymbol{\lambda}+\boldsymbol{\gamma})$. Define the set $\mathcal{D} \subseteq \mathbb{R}^N \times \mathbb{R}^N$ as

$$\left\{(\boldsymbol{\lambda},\boldsymbol{\gamma}) \in \mathbb{R}^N \times \mathbb{R}^N |\ \exists \mathbf{x} \in \mathbb{R}^{2N} \text{ such that } \forall i \in \{1,\ldots,N\}\right.$$

$$x_i \mathsf{r}\left[\xi_i = 1\right] \mathsf{r}\left[\xi_i = 0\right] > \textstyle\sum_{j\neq i} x_{N+j} |\mathsf{p}[\xi_i = 1]\mathsf{p}[\xi_j = 1] - \mathsf{p}\left[\xi_i = 1 \cap \xi_j = 1\right]|$$

$$+x_{N+i}|\mathsf{r}[\xi_i = 1]\mathsf{r}[\xi_i = 0] - \mathsf{p}[\xi_i = 1]\mathsf{p}[\xi_i = 0]|$$

and

$$x_{N+i} \mathsf{r}\left[\xi_i = 1\right] \mathsf{r}\left[\xi_i = 0\right] > \textstyle\sum_{j\neq i} x_j |\mathsf{q}[\xi_i = 1]\mathsf{q}[\xi_j = 1] - \mathsf{q}\left[\xi_i = 1 \cap \xi_j = 1\right]|$$

$$+x_i|\mathsf{r}[\xi_i = 1]\mathsf{r}[\xi_i = 0] - \mathsf{q}[\xi_i = 1]\mathsf{q}[\xi_i = 0]|$$

and

$$\mathsf{p}\left[\xi_i = 1\right] \mathsf{p}\left[\xi_j = 1\right] \leq \mathsf{p}\left[\xi_i = 1 \cap \xi_j = 1\right] \ \forall j \neq i, \text{ and}$$

$$\mathsf{q}\left[\xi_i = 1\right] \mathsf{q}\left[\xi_j = 1\right] \leq \mathsf{q}\left[\xi_i = 1 \cap \xi_j = 1\right] \ \forall j \neq i, \text{ and}$$

$$\mathsf{r}\left[\xi_i = 1\right] \mathsf{r}\left[\xi_i = 0\right] \leq \mathsf{p}\left[\xi_i = 1\right] \mathsf{p}\left[\xi_i = 0\right], \text{and}$$

$$\left.\mathsf{r}\left[\xi_i = 1\right] \mathsf{r}\left[\xi_i = 0\right] \leq \mathsf{q}\left[\xi_i = 1\right] \mathsf{q}\left[\xi_i = 0\right]\right\}$$

Consider any open set $\mathcal{C} \subseteq \mathcal{D}$. Then, given an initialization $\boldsymbol{\lambda}^{(0)} = (\boldsymbol{\lambda}^{(0)}, \boldsymbol{\gamma}^{(0)}) \in \mathcal{C}$ such that the $\mathbf{a}$ and $\mathbf{b}$ defined by

$$\begin{aligned} \mathbf{x} &= \mathbf{F}(\boldsymbol{\lambda}^{(0)}) \\ a_i &= \min(\mathbf{x}_i, 0) \quad \forall i \in \{1,\ldots,N\} \\ b_i &= \max(\mathbf{x}_i, 0) \quad \forall i \in \{1,\ldots,N\} \end{aligned}$$

are in $\mathbf{F}(\mathcal{C})$, and the set $\{\mathbf{x}|\mathbf{F}^{-1}(\mathbf{a}) \leq \mathbf{x} \leq \mathbf{F}^{-1}(\mathbf{b})\} \subseteq \mathcal{C}$ the turbo decoder converges to a unique fixed point in $\mathcal{C}$.

Proof: The outline of our proof parallels that of [64]:

- The conditions for $\mathcal{D}$ make the Jacobian of the nonlinear system of equations

$$\mathbf{F}(\boldsymbol{\lambda} = \begin{bmatrix} \boldsymbol{\lambda} \\ \boldsymbol{\gamma} \end{bmatrix}) = \begin{bmatrix} \mathbf{p}_{\mathbf{B}(\boldsymbol{\lambda}+\boldsymbol{\gamma})} - \mathbf{p}_{\mathbf{B}\boldsymbol{\lambda}+\boldsymbol{\rho}_0} \\ \mathbf{p}_{\mathbf{B}(\boldsymbol{\lambda}+\boldsymbol{\gamma})} - \mathbf{p}_{\mathbf{B}\boldsymbol{\gamma}+\boldsymbol{\rho}_1} \end{bmatrix}$$

strictly diagonally dominant anywhere within $\mathcal{D}$. This, in turn makes the Jacobian an M-matrix [65].

- Since the Jacobian is an M-matrix everywhere within $\mathcal{D}$, it follows that $\mathbf{F}$ is an M-function [64], [63].

- Since $\mathbf{F}$ is an M-function everywhere in $\mathcal{D}$, the Gauss Seidel block iteration on $\mathbf{F}$ converges. Hence, the turbo decoder converges.

Begin by determining the Jacobian of $\mathbf{F}$ with respect to $\boldsymbol{\lambda}$ and $\boldsymbol{\gamma}$. To facilitate this, define $\boldsymbol{\lambda} = \left[\boldsymbol{\gamma}^T, \boldsymbol{\lambda}^T\right]^T$, so

$$\nabla_{\boldsymbol{\lambda}} \mathbf{F} = \begin{bmatrix} \mathbf{Q}_0 & \mathbf{Q}_0 - \mathbf{Q}_1 \\ \mathbf{Q}_0 - \mathbf{Q}_2 & \mathbf{Q}_0 \end{bmatrix}$$

where

$$\mathbf{Q}_0 = \mathbf{P}_{\mathbf{B}(\boldsymbol{\lambda}+\boldsymbol{\gamma})} - \mathbf{p}_{\mathbf{B}(\boldsymbol{\lambda}+\boldsymbol{\gamma})} \mathbf{p}^T_{\mathbf{B}(\boldsymbol{\lambda}+\boldsymbol{\gamma})} \tag{5.18}$$

$$\mathbf{Q}_1 = \mathbf{P}_{\mathbf{B}\boldsymbol{\lambda}+\boldsymbol{\rho}_0} - \mathbf{p}_{\mathbf{B}\boldsymbol{\lambda}+\boldsymbol{\rho}_0} \mathbf{p}^T_{\mathbf{B}\boldsymbol{\lambda}+\boldsymbol{\rho}_0} \tag{5.19}$$

$$\mathbf{Q}_2 = \mathbf{P}_{\mathbf{B}\boldsymbol{\gamma}+\boldsymbol{\rho}_1} - \mathbf{p}_{\mathbf{B}\boldsymbol{\gamma}+\boldsymbol{\rho}_1} \mathbf{p}^T_{\mathbf{B}\boldsymbol{\gamma}+\boldsymbol{\rho}_1} \tag{5.20}$$

Next, note that the block diagonal components of this matrix are indeed diagonal matrices, since the PMF whose θ coordinates are $\mathbf{B}(\boldsymbol{\lambda} + \boldsymbol{\gamma})$ is a product measure. Thus we have that $\mathbf{Q}_0 = \mathsf{Diag}[\mathbf{p}_{\mathbf{B}(\boldsymbol{\lambda}+\boldsymbol{\gamma})}]\mathsf{Diag}[1 - \mathbf{p}_{\mathbf{B}(\boldsymbol{\lambda}+\boldsymbol{\gamma})}]$. Now, suppose $(\boldsymbol{\lambda}, \boldsymbol{\gamma}) \in \mathcal{D}$ and also that they are finite (i.e. $\boldsymbol{\lambda}, \boldsymbol{\gamma} \in \mathbb{R}^N$), so that $\mathbf{p}_{\mathbf{B}(\boldsymbol{\lambda}+\boldsymbol{\gamma})}$ and $1 - \mathbf{p}_{\mathbf{B}(\boldsymbol{\lambda}+\boldsymbol{\gamma})}$ have no elements which are equal to zero so that the diagonal elements of $\nabla_{\boldsymbol{\lambda}} \mathbf{F}$ are all positive. Then, the last four conditions for $\mathcal{D}$, imply that the off-diagonal components of $\nabla_{\boldsymbol{\lambda}} \mathbf{F}$ are non-positive. The other two conditions for $\mathcal{D}$, together with the positivity of the diagonal components and the negativity of the off diagonal components, then imply that $\nabla_{\boldsymbol{\lambda}} \mathbf{F}$ is generalized diagonally dominant. This, in

turn, implies, together with its positive diagonal elements and non-positive off diagonal elements, that $\nabla_{\boldsymbol{\lambda}}\mathbf{F}$ is an M-matrix (see [65] pp. 205 Theorem 6.5). This then means that the function $L(\boldsymbol{\lambda}, \mathbf{x}) = (\nabla_{\boldsymbol{\lambda}}\mathbf{F})\mathbf{x}$ is inverse isotone with respect to $\mathbf{x}$ for any fixed $\boldsymbol{\lambda} \in \mathcal{D}$, so that $L(\boldsymbol{\lambda}, \mathbf{x}_0) \leq L(\boldsymbol{\lambda}, \mathbf{x}_1) \implies \mathbf{x}_0 \leq \mathbf{x}_1$. Furthermore, $L(\boldsymbol{\lambda}, \mathbf{x}_0)$ for a fixed $\boldsymbol{\lambda} \in \mathcal{D}$ is also off diagonally antitone, so that, if we define the ith component of $L(\boldsymbol{\lambda}, \mathbf{x})$ by $L_i(\boldsymbol{\lambda}, \mathbf{x})$, and if we consider $\mathbf{y}, \mathbf{x}$ such that $\mathbf{x} \leq \mathbf{y}$ and $y_i = x_i$, then we have $L_i(\boldsymbol{\lambda}, \mathbf{y}) \geq L_i(\boldsymbol{\lambda}, \mathbf{x})$. Differentiability of $\mathbf{F}$, and thus the theory of best affine approximations, gives this same property to $\mathbf{F}(\boldsymbol{\lambda})$ at any point in the interior of $\mathcal{D}$ [64]. Thus, on the interior of $\mathcal{D}$, $\mathbf{F}(\boldsymbol{\lambda})$ is inverse isotone and off-diagonally antitone.

Now, consider the function which relates $\boldsymbol{\lambda}^{(k+1)} = [\boldsymbol{\lambda}^{(k+1)}; \boldsymbol{\gamma}^{(k+1)}]$ to $\boldsymbol{\lambda}^{(k)} = [\boldsymbol{\lambda}^{(k)}; \boldsymbol{\gamma}^{(k)}]$ and whose solution

$$\mathbf{G}(\boldsymbol{\lambda}^{(k+1)}, \boldsymbol{\lambda}^{(k)}) = 0$$

describes the evolution of the turbo decoder. In particular, since the turbo decoder is a block Gauss Seidel iteration on $\mathbf{F} = 0$, we have

$$\mathbf{G}(\boldsymbol{\lambda}^{(k+1)}, \boldsymbol{\lambda}^{(k)}) = \begin{bmatrix} \mathbf{p}_{\mathbf{B}(\boldsymbol{\lambda}+\boldsymbol{\gamma})}(\boldsymbol{\lambda}^{(k)}, \boldsymbol{\gamma}^{(k+1)}) - \mathbf{p}_{\mathbf{B}\boldsymbol{\lambda}+\boldsymbol{\rho}_0}(\boldsymbol{\lambda}^{(k)}) \\ \mathbf{p}_{\mathbf{B}(\boldsymbol{\lambda}+\boldsymbol{\gamma})}(\boldsymbol{\lambda}^{(k+1)}, \boldsymbol{\gamma}^{(k+1)}) - \mathbf{p}_{\mathbf{B}\boldsymbol{\gamma}+\boldsymbol{\rho}_1}(\boldsymbol{\gamma}^{(k+1)}) \end{bmatrix}$$

To see that the turbo decoder solves this iteration, just recall the Jacobian of $\mathbf{F}$, and note that the block diagonal components which were $\mathsf{Diag}[\mathbf{p}_{\mathbf{B}(\boldsymbol{\lambda}+\boldsymbol{\gamma})}]\mathsf{Diag}[\mathbf{1}-\mathbf{p}_{\mathbf{B}(\boldsymbol{\lambda}+\boldsymbol{\gamma})}]$ are injective for any real valued $\boldsymbol{\lambda}, \boldsymbol{\gamma}$, so that there are no zero or one bitwise marginal probabilities. The implicit function theorem applied to the top half of $\mathbf{G} = \mathbf{0}$, then gives that $\boldsymbol{\gamma}^{(k+1)}$ can be written as a function of $\boldsymbol{\lambda}^{(k)}$ and is unique for any particular $\boldsymbol{\lambda}^{(k)}$, so that no two $\boldsymbol{\gamma}^{(k+1)}$s satisfy the top half of $\mathbf{G} = \mathbf{0}$. Repeating this argument of the bottom half of $\mathbf{G} = \mathbf{0}$ shows that $\boldsymbol{\lambda}^{(k+1)}$ can be

written as a function of $\boldsymbol{\gamma}(k+1)$ and is unique for that $\boldsymbol{\gamma}^{(k+1)}$, which shows that it is unique given $\boldsymbol{\lambda}^{(k)}$. Thus, we have an equation whose repeated solution describes the behavior of the turbo decoder. We have

$$\mathbf{G}(\boldsymbol{\lambda}^{(k+1)}, \boldsymbol{\lambda}^{(k)}) = \begin{bmatrix} \mathbf{F}_0(\boldsymbol{\gamma}^{(k+1)}, \boldsymbol{\lambda}^{(k)}) \\ \mathbf{F}_1(\boldsymbol{\gamma}^{(k+1)}, \boldsymbol{\lambda}^{(k+1)}) \end{bmatrix}$$

This form makes it easy to see that, given that $\mathbf{F}$ is off diagonal antitone and inverse isotone on the interior of $\mathcal{D}$, $\mathbf{G}(\cdot, \mathbf{x})$ is inverse isotone for a fixed $\mathbf{x}$, and $\mathbf{G}(\mathbf{y}, \cdot)$ is antitone for a fixed $\mathbf{x}$ (all considered in the interior of $\mathcal{D}$). This in turn implies the implicit function $\mathbf{h}$, which satisfies

$$\mathbf{G}(\mathbf{h}(\mathbf{x}), \mathbf{x}) = 0$$

is isotonic. To see this, consider $\mathbf{x}_0 \leq \mathbf{x}_1$. Now, antitonicity of $\mathbf{G}$ gives

$$\mathbf{G}(\mathbf{h}(\mathbf{x}_1), \mathbf{x}_0) \geq \mathbf{G}(\mathbf{h}(\mathbf{x}_1), \mathbf{x}_1) = 0 = \mathbf{G}(\mathbf{h}(\mathbf{x}_0), \mathbf{x}_0)$$

Now, inverse isotonicity of $\mathbf{G}$ means that

$$\mathbf{G}(\mathbf{h}(\mathbf{x}_0), \mathbf{x}_0) \leq \mathbf{G}(\mathbf{h}(\mathbf{x}_1), \mathbf{x}_0) \implies \mathbf{h}(\mathbf{x}_0) \leq \mathbf{h}(\mathbf{x}_1)$$

We now have a function $\mathbf{h}$ that describes the turbo decoder through

$$\boldsymbol{\lambda}^{(k+1)} = \mathbf{h}(\boldsymbol{\lambda}^{(k)})$$

Of course, we could have written this function from the turbo decoder iterations when we started, but defining it this way made it easy to see some of its properties, namely that it is isotone in $\mathcal{D}$.

Now, consider some initialization $\boldsymbol{\lambda}^{(0)}$ in the interior of $\mathcal{D}$, and set $\mathbf{x} = \mathbf{F}(\boldsymbol{\lambda}^{(0)})$.

$$\begin{aligned} a_i &= \min(\mathbf{x}_i, 0),\ b_i = \max(\mathbf{x}_i, 0) \quad \forall i \in \{1, \ldots, N\} \\ \mathbf{y} &= \mathbf{F}^{-1}(\mathbf{a}),\ \mathbf{z} = \mathbf{F}^{-1}(\mathbf{b}) \end{aligned}$$

The inverse function F^{-1} is guaranteed to exist within the interior of $\mathcal{D}$, because $\nabla_{\boldsymbol{\lambda}}\mathsf{F}$ is an M-matrix and thus non-singular ([65] lemma 6.1 page 202). This, in turn, allows the inverse function theorem to be applied ([66] Theorem 9.24 page 221). Then we have both $\mathbf{a} \leq \mathbf{0} \leq \mathbf{b}$ and by inverse isotonicity of F, we have $\mathbf{y} \leq \boldsymbol{\lambda}^{(0)} \leq \mathbf{z}$. Isotonicity of h then implies that the sequences defined by

$$\mathbf{y}^{(k+1)} = \mathsf{h}(\mathbf{y}^{(k)}), \;\; \mathbf{z}^{(k+1)} = \mathsf{h}(\mathbf{z}^{(k)})$$

will obey $\mathbf{y}^{(k)} \leq \boldsymbol{\lambda}^{(k)} \leq \mathbf{z}^{(k)}$, $\mathbf{y}^{(k+1)} \geq \mathbf{y}^{(k)}$, and $\mathbf{z}^{(k+1)} \leq \mathbf{z}^{(k)}$. This shows that $\mathbf{y}^{(k)}, \mathbf{x}^{(k)}, \boldsymbol{\lambda}^{(k)}$ all converge to fixed points. Furthermore, F is injective because it was inverse isotone, which is because it was an M-function ([64] Theorem 3.2(a) page 511). The injectivity of F then implies that this fixed point must be unique in $\mathcal{C}$. This proves the theorem. ■

5.1.3 Application to the Turbo Decoder, II

Let us now try to apply the existing theory in a new way to the turbo decoder. In our ISIT paper [59] we worked in mixed domains. To be specific, we considered the turbo decoder as solving the bitwise marginal probability equations in terms of the log-likelihood ratios. In fact, in terms of directly applying the existing theory, this was probably not the best choice. Here, we've decided to work in the log domain because in that domain we can work with $\mathbb{R}^N$ in both the input space and output space. In particular, let us consider the turbo decoder as attempting to find a simultaneous solution to

$$\boldsymbol{\lambda} + \boldsymbol{\gamma} - \log\left(\mathbf{B}^T \exp(\mathbf{B}\boldsymbol{\lambda} + \boldsymbol{\rho}_0)\right) + \log\left((\mathbf{1} - \mathbf{B})^T \exp(\mathbf{B}\boldsymbol{\lambda} + \boldsymbol{\rho}_0)\right) = \mathbf{0}$$

and

$$\boldsymbol{\lambda} + \boldsymbol{\gamma} - \log\left(\mathbf{B}^T \exp(\mathbf{B}\boldsymbol{\gamma} + \boldsymbol{\rho}_1)\right) + \log\left((\mathbf{1} - \mathbf{B})^T \exp(\mathbf{B}\boldsymbol{\gamma} + \boldsymbol{\rho}_1)\right) = \mathbf{0}$$

We can now prove the following theorem.

Thm. 12 (Global Convergence of the Turbo Decoder)**:** Define the measures p whose θ coordinates are $\mathbf{B}\boldsymbol{\lambda} + \boldsymbol{\rho}_0$ and q whose θ coordinates are $\mathbf{B}\boldsymbol{\gamma} + \boldsymbol{\rho}_1$. Suppose that for all $\boldsymbol{\lambda}, \boldsymbol{\gamma} \in \mathbb{R}^N$, we have that there exists a $\mathbf{x} \in \mathbb{R}^{2N}$ such that $\forall i \in \{1, \ldots, N\}$

$$x_i > \sum_{j \neq i} x_{N+j} |\mathsf{p}\left[\xi_j = 1 | \xi_i = 1\right] - \mathsf{p}\left[\xi_j = 1 | \xi_i = 0\right]|$$

and

$$x_{N+i} > \sum_{j \neq i} x_j |\mathsf{q}\left[\xi_j = 1 | \xi_i = 1\right] - \mathsf{q}\left[\xi_j = 1 | \xi_i = 0\right]|$$

as well as

$$\mathsf{p}\left[\xi_j = 1 | \xi_i = 1\right] \geq \mathsf{p}\left[\xi_j = 1 | \xi_i = 0\right]$$

and

$$\mathsf{q}\left[\xi_j = 1 | \xi_i = 1\right] \geq \mathsf{q}\left[\xi_j = 1 | \xi_i = 0\right]$$

Then the turbo decoder converges to a unique solution from any initialization.

The outline of our proof is as follows.

- Our assumptions are equivalent to generalized strict diagonal dominance of $\nabla\mathbf{F}$ together with requiring that the off diagonal elements of $\nabla\mathbf{F}$ are non-positive for all $\boldsymbol{\lambda}, \boldsymbol{\gamma} \in \mathbb{R}^N$. Then, Thm. 14 shows that $\nabla\mathbf{F}$ is an M-matrix.
- Prop. 13 then shows that $\mathbf{F}$ is an M-function everywhere in $\mathbb{R}^{2N}$.
- We can linearly lower bound $\mathbf{F}$ componentwise, and thus use Thm. 16 to prove surjectivity of $\mathbf{F}$.
- We finally have that $\mathbf{F}$ is a surjective M-function. This allows us to apply Thm. 19 to prove convergence.

First, calculate the Jacobian $\nabla \mathbf{F}$

$$\nabla \mathbf{F} = \begin{bmatrix} \mathbf{I} & \mathbf{I} - \mathbf{C} \\ \mathbf{I} - \mathbf{D} & \mathbf{I} \end{bmatrix}$$

where

$$[\mathbf{C}]_{i,j} = \mathsf{p}[\xi_j = 1 | \xi_i = 1] - \mathsf{p}[\xi_j = 1 | \xi_i = 0]$$

and

$$[\mathbf{D}]_{i,j} = \mathsf{q}[\xi_j = 1 | \xi_i = 1] - \mathsf{q}[\xi_j = 1 | \xi_i = 0]$$

We now see that the first two conditions of the theorem force $\nabla \mathbf{F}$ to be generalized strictly diagonally dominant while the second two conditions force the off diagonal elements of $\nabla \mathbf{F}$ to be non-positive. This then allows us to apply Thm. 14 to show that $\nabla \mathbf{F}$ is an M-matrix for all $\boldsymbol{\lambda}, \boldsymbol{\gamma} \in \mathbb{R}^N$. Then Prop. 13 shows that $\mathbf{F}$ is an M-function. Next, consider that

$$\begin{aligned} -\log\left(\frac{\mathbf{B}^T \exp(\mathbf{B}\boldsymbol{\lambda} + \boldsymbol{\rho}_0)}{(\mathbf{1} - \mathbf{B})^T \exp(\mathbf{B}\boldsymbol{\lambda} + \boldsymbol{\rho}_0)}\right) &\geq -\log \frac{\mathbf{B}^T \exp(\mathbf{B}\boldsymbol{\lambda})}{(\mathbf{1} - \mathbf{B})^T \exp(\mathbf{B}\boldsymbol{\lambda})} + \mathbf{1} \log\left(\frac{\alpha_0}{\beta_0}\right) \\ &\geq -\boldsymbol{\lambda} + \log\left(\frac{\alpha_0}{\beta_0}\right) \end{aligned}$$

where

$$\alpha_0 = \min_i [\exp(\boldsymbol{\rho}_0)]_i, \quad \beta_0 = \max_i [\exp(\boldsymbol{\rho}_0)]_i$$

and we used the componentwise division notation so that $\frac{\mathbf{a}}{\mathbf{b}}$ for $\mathbf{a}, \mathbf{b} \in \mathbb{R}^N$ is the vector whose ith element is a_i/b_i. Similarly, we have

$$-\log\left(\frac{\mathbf{B}^T \exp(\mathbf{B}\boldsymbol{\gamma} + \boldsymbol{\rho}_1)}{(\mathbf{1} - \mathbf{B})^T \exp(\mathbf{B}\boldsymbol{\gamma} + \boldsymbol{\rho}_1)}\right) \geq -\boldsymbol{\gamma} + \mathbf{1} \log\left(\frac{\alpha_1}{\beta_1}\right)$$

where

$$\alpha_1 = \min_i [\exp(\boldsymbol{\rho}_1)]_i, \quad \beta_1 = \max_i [\exp(\boldsymbol{\rho}_1)]_i$$

Table 5.1: Some of the notation introduced in Section 5.2.

B_{h}	Bregman divergence from convex function h
$\overleftarrow{\mathbf{p}}$	Left Bregman projection
$\overrightarrow{\mathbf{p}}$	Right Bregman projection
$\mathcal{B}$	Parent exponential family
$\mathcal{P}$	marginal Exponential family
$\mathcal{Q}$	subfamily with $\mathbb{P}[\mathbf{x}^1 = \cdots = \mathbf{x}^{\mathtt{M}}] = 1$
h	a convex function
$\mathbf{s}$	sufficient statistics of $\mathcal{B}$
$\mathbf{k}$	repetition of $\mathbf{t}$ over $\mathtt{a}$

Putting these two facts together, we have

$$\mathbf{F}(\boldsymbol{\lambda}, \boldsymbol{\gamma}) \geq \begin{bmatrix} \boldsymbol{\gamma} + \mathbf{1} \log\left(\frac{\alpha_0}{\beta_0}\right) \\ \boldsymbol{\lambda} + \mathbf{1} \log\left(\frac{\alpha_1}{\beta_1}\right) \end{bmatrix}$$

which shows that

$$\lim_{k \to \infty} \|\mathbf{F}(\boldsymbol{\lambda}_k)\| = \infty$$

whenever $\|\boldsymbol{\lambda}_k\| \to \infty$ and either $\boldsymbol{\lambda}_k \geq \boldsymbol{\lambda}_{k+1}$ or $\boldsymbol{\lambda}_{k+1} \geq \boldsymbol{\lambda}_k$. This then implies that $\mathbf{F}$ is surjective by Thm. 16. Thus, $\mathbf{F}$ is a surjective M-function, and Thm. 19 proves that the Gauss Seidel block iteration on $\mathbf{F}$ is globally convergent. ■

5.2 Alternating Projections on Information Spaces

In this subsection we will discuss various iterative methods used for solving projection problems on convex sets with Bregman divergences. We will then see how the expectation propagation algorithm may be related to these algorithms, although some approximations will be required for the connection.

5.2.1 Bregman Divergences

To begin, it will be useful to recall that a differentiable convex function is always lower bounded by its first order Taylor approximation. In particular, if $\mathsf{h} : \mathbb{R}^N \to \mathbb{R}$ is a convex function of Legendre type (see Appendix B and the references therein) then

$$\mathsf{h}(\mathbf{x}) \geq \mathsf{h}(\mathbf{y}) + \nabla \mathsf{h}(\mathbf{y}) \cdot (\mathbf{x} - \mathbf{y}) \tag{5.21}$$

with equality if and only if $\mathbf{x} = \mathbf{y}$. Given, then, such a convex function h, we can define a second function which has many of the properties of a notion of a distance. In particular, the *Bregman Divergence* [18], B_h associated with h is defined as

$$\mathsf{B}_\mathsf{h}(\mathbf{x}, \mathbf{y}) := \mathsf{h}(\mathbf{x}) - \mathsf{h}(\mathbf{y}) - \nabla \mathsf{h}(\mathbf{y}) \cdot (\mathbf{x} - \mathbf{y})$$

This function has some of the properties of a distance. In particular, we see from (5.21) that

$$\mathsf{B}_\mathsf{h}(\mathbf{x}, \mathbf{y}) \geq 0 \quad \mathsf{B}_\mathsf{h}(\mathbf{x}, \mathbf{y}) = 0 \Leftrightarrow \mathbf{x} = \mathbf{y}$$

On the other hand, B_h will not necessarily satisfy the triangle inequality in general and will not necessarily be symmetric. We are interested in Bregman divergences of this sort because we are primarily interested in this dissertation with standard exponential families. As we discuss in Appendix D, there are two natural parameter spaces for exponential families related through the Legendre transformation between the entropy and the log partition function. These two potential functions, the entropy and the log partition function, yield the Kullback Leibler divergence, also known as the relative entropy, as their Bregman divergence. Since the expectation propagation algorithm is defined as an iterative algorithm minimizing the Kullback Leibler divergences between several measures, we can see the relevance between projection algorithms using Bregman divergences and expectation

propagation. We explore these connections here after first reviewing some of the common types of projection algorithms that are related to the expectation propagation algorithm.

5.2.2 Projection Algorithms on Convex Sets

Endowing the space now with this distance like function we are now free to consider projection algorithms using B_h as the "distance" to minimize. In particular, since B_h is not in general symmetric, given a point $\mathbf{x}$ and a convex set $\mathcal{C}$ we can define two different projections of $\mathbf{x}$ onto $\mathcal{C}$ depending on which argument of B_h we minimize.

$$\overleftarrow{\mathbf{p}}_{\mathcal{C}}\mathbf{x} := \arg\min_{\mathbf{y}\in\mathcal{C}} \mathsf{B}_\mathsf{h}(\mathbf{y},\mathbf{x})$$

$$\overrightarrow{\mathbf{p}}_{\mathcal{C}}\mathbf{x} := \arg\min_{\mathbf{y}\in\mathcal{C}} \mathsf{B}_\mathsf{h}(\mathbf{x},\mathbf{y})$$

One particular algorithm, called the *method of cyclic Bregman projections*[67][68], aims to solve a convex feasibility problem in which one wishes to find a point common to a finite number of convex sets $\mathcal{C}_1,\ldots,\mathcal{C}_\mathtt{S}$. In order to find such a point in their intersection $\bigcap_{\mathtt{i}=1}^{\mathtt{S}}\mathcal{C}_\mathtt{i}$, the algorithm iteratively projects the candidate point $\mathbf{x}^{(\mathtt{k})}$ onto one of the convex sets, choosing which convex set to project onto in a cyclic manner. Of course, there are two algorithms which fall into this category, one if the projections are done in the first argument of the divergence

$$\mathbf{x}^{(\mathtt{k}+1)} := \overleftarrow{\mathbf{p}}_{\mathcal{C}_{\mathtt{k} \bmod \mathtt{S}}}\mathbf{x}^{(\mathtt{k})}$$

and another if the projections are done in the second argument of the divergence

$$\mathbf{x}^{(\mathtt{k}+1)} := \overrightarrow{\mathbf{p}}_{\mathcal{C}_{\mathtt{k} \bmod \mathtt{S}}}\mathbf{x}^{(\mathtt{k})}$$

Instead of choosing the convex set to project on cyclicly, one may also choose it randomly [69, 70].

A close variant of the method of cyclic Bregman projections, often called *Dykstra's algorithm* [18, 71] after its inventor [17], adds some extra processing between the projections. In particular, denoting by h^* the convex conjugate of h, Dykstra's algorithm is the iteration

$$\mathbf{x}^{(\mathtt{k}+1)} := \overleftarrow{\mathbf{p}}_{\mathcal{C}} \circ \nabla \mathsf{h}^* \left(\nabla \mathsf{h}(\mathbf{x}^{(\mathtt{k})}) + \mathbf{q}^{(\mathtt{k}+1-\mathtt{s})} \right)$$

together with

$$\mathbf{q}^{(\mathtt{k}+1)} := \nabla \mathsf{h}(\mathbf{x}^{(\mathtt{k})}) + \mathbf{q}^{(\mathtt{k}+1-\mathtt{s})} - \nabla \mathsf{h}(\mathbf{x}^{(\mathtt{k}+1)})$$

where we initialize $\mathbf{q}^{(-\mathtt{s}+1)}, \ldots, \mathbf{q}^{(0)} = \mathbf{0}$. This algorithm, under some assumptions, can be shown [18] to solve the best approximation problem, in which one is seeking the point in $\mathcal{C} := \bigcap_{\mathtt{i}=1}^{\mathtt{s}} \mathcal{C}_{\mathtt{i}}$ which minimizes the Bregman divergence B_{h} in the first argument from the initial point $\mathbf{x}^{(0)}$. That is, the sequence $\{\mathbf{x}^{(\mathtt{k})}\}$ converges to $\overleftarrow{\mathbf{p}}_{\mathcal{C}} \mathbf{x}^{(0)}$.

In another situation, it is desirable to find points in two different convex sets $\mathcal{P}$ and $\mathcal{Q}$ which minimize the Bregman divergence B_{h} between these two sets. The projection algorithm that is often employed in this case is the *method of alternating projections* [72, 73, 74] which may be described via the iteration

$$\mathbf{x}^{(\mathtt{k})} := \overleftarrow{\mathbf{p}}_{\mathcal{P}} \mathbf{y}^{(\mathtt{k})}, \quad \mathbf{y}^{(\mathtt{k}+1)} := \overrightarrow{\mathbf{p}}_{\mathcal{Q}} \mathbf{x}^{(\mathtt{k})}$$

The expectation maximization algorithm for statistical inference in the presence of nuisance parameters may be interpreted, for instance, as an alternating projection algorithm using the Kullback Leibler divergence at the Bregman divergence [72, 75]. [76] discusses some properties and well-posedness of projections when the Bregman divergence is the Kullback Leibler divergence.

5.2.3 Expectation Propagation as Iterated Projections

For the convergence framework that we are about to apply to expectation propagation to be valid, we will need to make a rather benign extra assumption about the structure of the true a posteriori distribution $\mathsf{p}_{\boldsymbol{\theta}|\mathbf{r}}(\boldsymbol{\theta}|\mathbf{r})$ for the parameters $\boldsymbol{\theta}$ given the observations $\mathbf{r}$. In particular, we must assume that it is a standard exponential family distribution, say with sufficient statistics $\mathbf{k}(\boldsymbol{\theta})$, so that

$$\mathsf{p}_{\boldsymbol{\theta}|\mathbf{r}}(\boldsymbol{\theta}|\mathbf{r}) \in \{\exp\left(\boldsymbol{\kappa} \cdot \mathbf{k}(\boldsymbol{\theta}) - \psi_{\mathbf{k}}(\boldsymbol{\kappa})\right)\}$$

Furthermore, we need to make the extra assumption that the approximating standard exponential family $\mathsf{g}(\boldsymbol{\theta})$ with sufficient statistics $\mathbf{t}$ is a subfamily of the standard exponential family with sufficient statistics $\mathbf{k}$, so that

$$\{\exp\left(\mathbf{t}(\boldsymbol{\theta}) \cdot \boldsymbol{\lambda} - \psi_{\mathbf{t}}(\boldsymbol{\lambda})\right)\} \subseteq \{\exp\left(\mathbf{k}(\boldsymbol{\theta}) \cdot \boldsymbol{\kappa} - \psi_{\mathbf{k}}(\boldsymbol{\kappa})\right)\}$$

These conditions will hold, for instance, if the parameters $\boldsymbol{\theta}$ are drawn from a finite set.

Consider now standard exponential family distributions on $\Theta^{\mathtt{M}}$, that is probability distributions on the product space of $\mathtt{M}$ copies of the original parameter space $\boldsymbol{\theta}$. We will denote an element of this space by $\mathbf{x} := [\mathbf{x}^1, \ldots, \mathbf{x}^{\mathtt{M}}]$, so that we are considering probability distributions on $\mathbf{x}$ (the superscript notation is to avoid confusion with the subscript notation which meant subsets of the original vector instead of different vectors). In particular we will be interested in the set $\mathcal{B}$ of standard exponential family distributions on $\mathbf{x}$ with sufficient statistics $\mathbf{s}(\mathbf{x}) := \left[\mathbf{k}(\mathbf{x}^1)^T, \cdot, \mathbf{k}(\mathbf{x}^{\mathtt{M}})^T\right]^T$ (and reference measure which the product of $\mathtt{M}$ copies of the original reference measure). It is subsets of this family which we will consider in our discussion of expectation propagation algorithms as projection algorithms.

Of particular interest will be the set $\mathcal{P}$ of distributions from $\mathcal{B}$ which give $\mathbb{P}\left[\mathbf{x}_1 = \cdots = \mathbf{x}_M\right] = 1$ and the set $\mathcal{Q}$ of exponential family distributions with sufficient statistics $\hat{\mathbf{t}}(\mathbf{x}) := [\mathbf{t}(\mathbf{x}_1), \ldots, \mathbf{t}(\mathbf{x}_M)]$.

$$\mathcal{P} := \{q \in \mathcal{B} | \mathbb{P}_q \left[\mathbf{x}_1 = \cdots = \mathbf{x}_M\right] = 1\} \tag{5.22}$$

$$\mathcal{Q} := \left\{q | q = \exp\left(\boldsymbol{\lambda} \cdot \hat{\mathbf{t}}(\mathbf{x}) - \psi_{\hat{\mathbf{t}}}(\boldsymbol{\lambda})\right),\ \boldsymbol{\lambda} \in \mathbb{R}^{MV},\ |\psi_{\hat{\mathbf{t}}}(\boldsymbol{\lambda})| < \infty\right\} \tag{5.23}$$

Finally, we will be working with the negative entropy function H on $\mathcal{B}$ written as a function of the expectation parameters $\boldsymbol{\eta} := \mathbb{E}\left[\mathbf{s}(\mathbf{x})\right]$.

Oftentimes, the desired result of applying expectation propagation is an approximating density whose expected values of the sufficient statistics match that of the true a posteriori density, so that

$$\mathbb{E}_g\left[\mathbf{t}(\boldsymbol{\theta})\right] = \mathbb{E}_{\boldsymbol{\theta}|\mathbf{r}}\left[\mathbf{t}(\boldsymbol{\theta})\right] \tag{5.24}$$

The next proposition connects this ideal solution as a composition of two Bregman projections using the Kullback Leibler divergence.

Prop. 7 (Optimal Solution as Two Projections)**:** Consider the Bregman divergence generated using the negative entropy H function on $\mathcal{B}$ written in terms of $\mathbb{E}[\mathbf{s}(\mathbf{x})]$ as its convex function. Then, the Bregman projection defined by

$$\overrightarrow{\mathbf{p}}_{\mathcal{P}} \circ \overleftarrow{\mathbf{p}}_{\mathcal{Q}} \left(\frac{\int_{\Theta^M} \mathbf{s}(\mathbf{x}) \prod_{a=1}^{M} f_a(\mathbf{x}_a) d\mathbf{x}}{\int_{\Theta^M} \prod_{a=1}^{M} f^a(\mathbf{x}_a) d\mathbf{x}} \right) \tag{5.25}$$

yields expectation parameters $\mathbb{E}[\mathbf{s}]$ corresponding to a probability distribution $g(\mathbf{x}) \in \mathcal{B}$ whose components satisfy (5.24), so that

$$\mathbb{E}_g[\mathbf{t}(\mathbf{x}_a)] = \mathbb{E}_{\boldsymbol{\theta}|\mathbf{r}}\left[\mathbf{t}(\boldsymbol{\theta})\right] \quad \forall a \in \{1, \ldots, M\}$$

Proof: The first projection $\overleftarrow{\mathbf{p}}_{\mathcal{Q}}$ yields as its result

$$\overleftarrow{\mathbf{p}}_{\mathcal{Q}} \left(\prod_{a=1}^{M} f_a(\mathbf{x}_a) \right) = \frac{\prod_{a=1}^{M} f_a(\mathbf{x}_1)}{\int_{\Theta} \prod_{a=1}^{M} f_a(\mathbf{x}_1) d\mathbf{x}_1} =: p_{\boldsymbol{\theta}|\mathbf{r}}(\mathbf{x}_1|\mathbf{r})$$

since this attains a Kullback Leibler divergence of zero (which is the global minimum value). The second projection $\vec{\mathbf{p}}_{\mathcal{P}}$ then, calculates the expectation of sufficient statistics (as we showed by taking derivatives in Section 2) to choose the approximating distribution (and thus result of the projection). Thus, defining

$$\mathsf{f} := \vec{\mathbf{p}}_{\mathcal{P}} \circ \overleftarrow{\mathbf{p}}_{\mathcal{Q}} \left(\prod_{\mathtt{a}=1}^{\mathtt{M}} \mathsf{f}^{\mathtt{a}}(\mathbf{x}_{\mathtt{a}}) \right)$$

implies

$$\mathbb{E}_{\mathsf{f}}[\mathbf{t}(\boldsymbol{\theta})] = \mathbb{E}_{\boldsymbol{\theta}|\mathbf{r}}[\mathbf{t}(\boldsymbol{\theta})]$$

■

Of course, expectation propagation does not perform the projection (5.25) directly, but if it is to be a method for approximately satisfying (5.24), then it may be considered an iterative projection method attempting to approximately solve the projection problem (5.25).

Prop. 8 (Expectation Propagation as Information Projections)**:** The expectation propagation algorithm under parallel scheduling is equivalent to the following Dykstra-like iterated Bregman projections algorithm

$$\mathbf{z}_0, \mathbf{q}_0 = \mathbf{0}, \quad \mathbf{x}_0 = \frac{\int_{\Theta^{\mathtt{M}}} \mathbf{s}(\mathbf{x}) \prod_{\mathtt{a}=1}^{\mathtt{M}} \mathsf{f}_{\mathtt{a}}(\mathbf{x}^{\mathtt{a}}) \mathrm{d}\mathbf{x}}{\int_{\Theta^{\mathtt{M}}} \prod_{\mathtt{a}=1}^{\mathtt{M}} \mathsf{f}_{\mathtt{a}}(\mathbf{x}^{\mathtt{a}}) \mathrm{d}\mathbf{x}} \tag{5.26}$$

$$\mathtt{k} \in \{0, 1, \ldots, \} \tag{5.27}$$

$$\mathbf{y}_{\mathtt{k}} := \vec{\mathbf{p}}_{\mathcal{Q}} \circ \nabla \mathsf{h}^* \left(\nabla \mathsf{h}(\mathbf{x}_{\mathtt{k}}) + \mathbf{q}_{\mathtt{k}} \right) \tag{5.28}$$

$$\mathbf{q}_{\mathtt{k}+1} := \nabla \mathsf{h}(\mathbf{x}_{\mathtt{k}}) + \mathbf{q}_{\mathtt{k}} - \nabla \mathsf{h}(\mathbf{y}_{\mathtt{k}}) \tag{5.29}$$

$$\mathbf{x}_{\mathtt{k}+1} := \vec{\mathbf{p}}_{\mathcal{Q}} \circ \overleftarrow{\mathbf{p}}_{\mathcal{P}} \circ \nabla \mathsf{h}^* \left(\nabla \mathsf{h}(\mathbf{y}_{\mathtt{k}}) + \mathbf{z}_{\mathtt{k}} \right) \tag{5.30}$$

$$\mathbf{z}_{\mathtt{k}+1} := \nabla \mathsf{h}(\mathbf{y}_{\mathtt{k}}) + \mathbf{z}_{\mathtt{k}} - \nabla \mathsf{h}(\mathbf{x}_{\mathtt{k}+1}) \tag{5.31}$$

where the Bregman divergence is built from the convex function h which is the negative entropy H function on $\mathcal{B}$ written in terms of $\mathbb{E}[\mathbf{s}(\mathbf{x})]$, and $\mathcal{P}$ $\mathcal{Q}$ are the convex and

log convex sets defined in (5.22) and (5.23), respectively. In this equivalence $\mathbf{q}_k$ are the negative of the messages passed from the factor nodes to the elementary statistics vectors, and $\mathbf{z}_k$ are the negative of the messages passed from the elementary statistics vectors to the factor nodes.

Proof: Begin by recalling from Appendix D.4 that the gradient $\nabla \mathsf{h}$ of the negative Shannon entropy takes the expectation parameters of a pdf in $\mathcal{B}$ to its canonical parameters $\boldsymbol{\kappa}$, and the gradient of its conjugate (the partition function) is its inverse. The pre-projection processing in (5.28) and (5.30), then finds the pdf in $\mathcal{B}$ whose canonical parameters $\boldsymbol{\kappa}$ are the sum of $\mathbf{q}$ with that of the density represented by $\mathbf{x}$. As we have discussed previously, the result of the right projection $\overrightarrow{\mathbf{p}}_{\mathcal{P}}$ are the expectation coordinates ($\mathbb{E}[\mathbf{s}]$) of the density in $\mathcal{P}$ with $\mathbb{E}[\mathbf{k}]$ matching that of its argument (that this density is unique is guaranteed by the fact that $\mathcal{P}$ is a minimal standard exponential family with sufficient statistics $\mathbf{k}$). Now, this establishes the equivalence of the first iteration of (5.28) followed by (5.29) with (5.3) solved for $\boldsymbol{\lambda}$. Furthermore, if $\mathbf{q}$ is $-\boldsymbol{\gamma}$ from (5.3) for subsequent iterations, then solving (5.28) and (5.29) will be equivalent to (5.3) for subsequent iterations as well. Then, $\overrightarrow{\mathbf{p}}_{\mathcal{Q}} \circ \overleftarrow{\mathbf{p}}_{\mathcal{P}}$, when operating on the expectation coordinates of a pdf $\mathsf{q}(\mathbf{x}) = \prod_{\mathtt{a}} \mathsf{q}_{\mathtt{a}}(\mathbf{x}^{\mathtt{a}})$ which is input to it, calculates

$$\frac{\int_{\Theta} \mathbf{t}(\boldsymbol{\theta}) \prod^{\mathtt{a}} \mathsf{q}_{\mathtt{a}}(\boldsymbol{\theta}) \mathsf{d}\boldsymbol{\theta}}{\int_{\Theta} \prod_{\mathtt{a}} \mathsf{q}_{\mathtt{a}}(\boldsymbol{\theta}) \mathsf{d}\boldsymbol{\theta}} \tag{5.32}$$

Now, if we make the identification $\mathsf{q}_{\mathtt{a}}(\boldsymbol{\theta}) := \exp(\mathbf{t}_{\mathtt{a}}(\boldsymbol{\theta}_{\mathtt{a}}) \cdot \boldsymbol{\lambda}_{\mathtt{a}})$, we see that (5.32) followed by the subtraction in (5.31) is equivalent to the (natural parameter domain) update (5.4) over all $\mathtt{a} \in \{1, \ldots, \mathtt{M}\}$, which we saw in Section 5.1 to be equivalent to half of a parallel refinement in EP. This establishes the equivalence between the $\mathbf{z}$'s and $-\boldsymbol{\gamma}$. Cycling back to our argument for (5.28)/(5.29) then shows the

equivalence between the iteration described by (5.28), (5.29), (5.30), and (5.31) and (5.3), (5.4), which we showed in Section 5.1 to be equivalent to expectation propagation. ■

Note that there are several marked differences between expectation propagation and Dykstra's algorithm. To begin with Dykstra's algorithm with cyclic Bregman projections is built with only left proximity operators, and it is somewhat unusual for one of the projections to not have any preprocessing. One can consider alternating projections with Dykstra preprocessing ([73] did so for Hilbert spaces with orthogonal projections instead of Bregman projections), but one must use different preprocessing based on the projection to follow it. The preprocessing used in expectation propagation is the preprocessing for a left projection, but it is followed by a right projection. Still, the similarity between the two algorithms is striking, and it suggests that convergence results might be developed via this analogy.

We are not the first to note the relevance of projection algorithms to specific instances of expectation propagation. Very early on [48] noted the relevance of information projections to turbo decoding. Later, [50, 36] and [43] considered belief propagation and turbo decoding to be projection algorithms on information sets, and this work inspired our alternative information projection interpretation of turbo decoding in [51].

Chapter 6

Conclusions

In this dissertation we discussed the performance and convergence of a broadly reaching family of algorithms for distributed iterative decoding and estimation known as expectation propagation. We showed that the family of expectation propagation algorithms include both the serial and parallel turbo decoders, Gallager's algorithm for the soft decoding of LDPC codes, and the belief propagation algorithm. While these particular algorithms had been shown to converge to the optimal solutions in some very special cases, those proofs did not apply to many of their breakthrough applications to the decoding of error control codes, and the algorithms were proposed heuristically with their performance being proven via simulation. Indeed, neither the sense of optimality of their stationary points nor the mechanism of their frequently occurring convergence was fully understood.

In order to explain the empirically observed capability of expectation propagation to perform well, we discussed two generic optimality frameworks which described the optimization problem that the expectation propagation algorithm is trying to solve. For one of the optimality frameworks, we extended results from belief propagation to show that the stationary points of expectation propagation are interior critical points of a statistics based constrained minimization of the Bethe free energy. Because the Bethe free energy is only an approximation to the variational free energy, however, and the sense in which it was approximate was not fully understood, it was not clear why one would wish to minimize this objective function. To solve this, we provided a novel constrained maximum likelihood optimization problem with an intuitive objective function and constraints, which, after

selecting a Lagrange multiplier of -1, yielded the stationary points of expectation propagation as its critical points. In order to highlight the theory, we discussed in depth two applications to specific expectation propagation algorithms: the turbo decoder and the belief propagation decoder. Further analysis then showed that the two generic optimality frameworks were pseudo-duals of each other, allowing us to connect the two optimality interpretations with some beautiful duality theory from nonlinear programming.

We then moved on to discuss the convergence of expectation propagation by identifying some relevant convergence frameworks from numerical and convex analysis which already have hefty literature containing various convergence proofs. To begin with, we showed that both the serial and parallel schedules for the expectation propagation algorithm may be interpreted as a Gauss Seidel iteration on the first order necessary conditions for both of the generic optimality frameworks we discussed in Chapter 4. This allowed us to apply a couple of convergence theorems from numerical analysis to the expectation propagation algorithm, albeit with mixed practical success. Starting afresh with a clean slate, we showed how expectation propagation may be related to Dykstra's algorithm for solving a convex feasibility problem with cyclic Bregman projections.

The ideas and themes begun in this dissertation are far from complete. It was our intention to open up promising avenues for further analysis of particular instances of expectation propagation. Some immediate extensions of the theory are to bound the performance of the global optimum of constrained maximum likelihood with respect to the value of the constraint, using this in practical instances of the algorithm to accurately predict when the algorithm has converged to a stationary point which is likely to provide accurate estimates. One could set

out to compare the performance bound obtained via this technique with that of the Cramer Rao bound to argue for the room (or lack there of) for improvement of the expectation propagation algorithm in particular cases. One could attempt to show how to predict which approximate sufficient statistics should be picked for specific instances of the algorithm, again attempting to extract a general theory.

In another line of extension, one could set out looking for new applications of expectation propagation in sensor networks, cellular networks, and other wireless networks. Expectation propagation's distributed iterative nature gives it a definite edge over algorithms which do not collaborate on estimation problems that will clearly be coupled, as many are likely to be in these contexts. One could apply expectation propagation to training diagnosis predictors from medical information databases. The possibilities in the applications arena are simply too widespread to even begin to enumerate.

The more analytically driven and ambitious reader could attempt to use the convergence frameworks developed here to obtain an overarching convergence theory tailored to the needs of typical cases of expectation propagation. While it appears unlikely that simple unadulterated or slightly modified application of the convergence theory from the arenas from which the convergence frameworks were drawn will be directly applicable to typical instances of expectation propagation, these new frameworks ought to be good place to start for the fundamental convergence ideas. Along different lines, given the pseudo-duality between the two optimality frameworks, one could investigate conditions for duality and then use convexity properties of the dual to apply a different sort of Gauss Seidel convergence framework from iterative optimization, as was the approach in [60]. Another clear convergence framework, typical in the context of applications of expectation

propagation such as belief propagation and turbo decoding, but not appearing in this dissertation, is that of density evolution [46, 77, 78, 79, 19]. Density evolution uses the law of large numbers and the central limit theorem along with ergodicity results to consider cases in which the parameter vector is infinite, focussing on the average performance of the individual estimation problems. This approach is likely to be, and indeed has been, successful in situations where the factors may be considered to be samples from a random distribution and the statistics factor graph is well approximated by a randomly chosen bipartite graph without short cycles. This theory could be, with a little effort, generalized to expectation propagation.

Indeed, there are many possibilities, and the author hopes that the reader will join him in exploring them. In that vein, the bibliography contains numerous articles [80]-[165] which were not explicitly used in this dissertation, but would be relevant for the reader wishing to extend these results.

Appendix A

Numerical Methods

Here, we collect the existing theory about nonlinear block Gauss Seidel iteration's convergence. Most of the material comes from [64], although we will use some of the theory from [65] as well.

Def. 4 (Ordering)**:** In this paper we will use the componentwise ordering of $\mathbb{R}^N$, so that

$$\mathbf{x} \geq \mathbf{y} \iff x_i \geq y_i \forall i \in \{0, \ldots, N-1\}, \quad \mathbf{x} > \mathbf{y} \iff x_i > y_i \forall i \in \{0, \ldots, N-1\}$$

and, similarly

$$\mathbf{x} \leq \mathbf{y} \iff x_i \leq y_i \, \forall i \in \{0, \ldots, N-1\}, \quad \mathbf{x} < \mathbf{y} \iff x_i \leq y_i \, \forall i \in \{0, \ldots, N-1\}$$

Def. 5 (isotone)**:** A mapping $\mathbf{F} : \mathcal{D} \subseteq \mathbb{R}^N \rightarrow \mathbb{R}^N$ is isotone on $\mathcal{D}$ if $\mathbf{x} \leq \mathbf{y}$ for $\mathbf{x}, \mathbf{y} \in \mathcal{D}$ implies that $\mathbf{F}(\mathbf{x}) \leq \mathbf{F}(\mathbf{y})$.

Def. 6 (antitone)**:** A mapping $\mathbf{F} : \mathcal{D} \subseteq \mathbb{R}^N \rightarrow \mathbb{R}^N$ is antitone on $\mathcal{D}$ if $\mathbf{x} \leq \mathbf{y}$ for $\mathbf{x}, \mathbf{y} \in \mathcal{D}$ implies that $\mathbf{F}(\mathbf{x}) \geq \mathbf{F}(\mathbf{x})$.

Def. 7 (inverse isotone)**:** The mapping $\mathbf{F} : \mathcal{D} \subseteq \mathbb{R}^N \rightarrow \mathbb{R}^N$ is inverse isotone on $\mathcal{D}$ if $\mathbf{F}(\mathbf{x}) \leq \mathbf{F}(\mathbf{y})$ for $\mathbf{x}, \mathbf{y} \in \mathcal{D}$ implies that $\mathbf{x} \leq \mathbf{y}$.

Def. 8 (Off-diagonally Antitone)**:** Suppose $\mathbf{F} : \mathcal{D} \subseteq \mathbb{R}^N \rightarrow \mathbb{R}^N$ has component functions $\mathsf{F}_0, \mathsf{F}_1, \ldots$ each of which is thus a map taking $\mathbb{R}^N \rightarrow \mathbb{R}$. Then $\mathbf{F}$ is off-diagonally antitone if for any $\mathbf{x} \in \mathbb{R}^N$ we have that

$$\phi_{i,j} : \mathbb{R} \rightarrow \mathbb{R}, \quad \phi_{i,j}(t) = \mathsf{F}_i(\mathbf{x} + t\mathbf{e}_j)$$

where $\mathbf{e}$ is the jth column in the identity matrix of order N, is antitone whenever $i \neq j$. That is

$$i \neq j,\ t_1 \leq t_2 \implies \phi_{i,j}(t_1) \geq \phi_{i,j}(t_2)$$

Def. 9 (M-function)**:** An inverse isotone and off-diagonally antitone mapping $\mathbf{F} : \mathcal{D} \subseteq \mathbb{R}^N \to \mathbb{R}^N$ is an M-function.

Def. 10 (M-Matrix)**:** A real square matrix $\mathbf{A} = [a_{i,j}] \in \mathbb{R}^{N\times N}$ is called an M-matrix if $a_{i,j} \leq 0$ for $i \neq j$ and $\mathbf{A}^{-1} \geq 0$ (in other words $\mathbf{Ax} \geq 0 \implies \mathbf{x} \geq 0$).

Thm. 13 (Properties of M-Functions)**:** ([64] Thm. 3.6) Let $\mathbf{F} : \mathbb{R}^N \to \mathbb{R}^N$ be Frechet differentiable on all of $\mathbb{R}^N$

- If for all $\mathbf{x} \in \mathbb{R}^N$ the derivative $\mathbf{F}'$ of $\mathbf{F}$ is a M-matrix, then $\mathbf{F}$ is an M-function.
- If $\mathbf{F}$ is an M-function, then the derivative $\mathbf{F}'$ of $\mathbf{F}$ is a M-matrix whenever it is nonsingular.

Def. 11 (Generalized Strictly Diagonally Dominant)**:** ([65] Def. 6.3 pp. 202) A matrix $\mathbf{A} \in \mathbb{R}^{N\times N}$ is generalized strictly diagonally dominant if there is some $\mathbf{x} \in \mathbb{R}^N$ such that

$$|a_{i,i}|x_i > \sum_{j\neq i} |a_{i,j}|x_j, \quad i \in \{1, \ldots, N\}$$

Thm. 14 (Generalized Diagonal Dominance and M-Matrix)**:** ([65] Thm. 6.5 pp. 205) A matrix $\mathbf{A} \in \mathbb{R}^{N\times N}$ such that $a_{i,j} \leq 0,\ j \neq i$ is a M-Matrix if and only if it is generalized strictly diagonally dominant and $a_{i,i} \geq 0$.

Def. 12 (Order Coercive)**:** A function $\mathbf{F} : \mathbb{R}^N \to \mathbb{R}^N$ is order coercive if

$$\lim_{k\to\infty} \|\mathbf{F}(\mathbf{x}_k)\| = \infty \text{ whenever } \lim_{k\to\infty} \|\mathbf{x}_k\| = \infty \text{ and either } \mathbf{x}_k \leq \mathbf{x}_{k+1} \text{ or } \mathbf{x}_k \geq \mathbf{x}_{k+1}$$

Thm. 15 (Surjectivity 1)**:** ([64] Lemma 4.3) A continuous, inverse isotone mapping $\mathbf{F} : \mathbb{R}^N \rightarrow \mathbb{R}^N$ is surjective if and only if

$$\{\mathbf{x} | \mathbf{x} = \mathbf{z} + t\mathbf{v}, -\infty < t < \infty\} \subset \mathbf{F}(\mathbb{R}^N)$$

for some $\mathbf{z}, \mathbf{v} \in \mathbb{R}^N$ such that $\mathbf{v} > 0$.

Thm. 16 (Surjectivity 2)**:** ([64] Thm. 4.4) A continuous, inverse isotone mapping $\mathbf{F} : \mathbb{R}^N \rightarrow \mathbb{R}^N$ is surjective if and only if it is order-coercive.

Thm. 17 (Surjectivity 3)**:** ([64] Thm. 4.5) Suppose that $\mathbf{F} : \mathbb{R}^N \rightarrow \mathbb{R}^N$ is continuously Frechet differentiable and that, for any $\mathbf{x} \in \mathbb{R}^N$, $\mathbf{F}'(\mathbf{x})$ is non-singular and satisfies $\mathbf{F}(\mathbf{x})^{-1} \geq 0$. Then $\mathbf{F}$ is inverse isotone and surjective if and only if it is order-coercive.

Def. 13 (regular iteration function)**:** A mapping $\mathbf{G} : \mathcal{D}_0 \times \mathcal{D}_0 \subseteq \mathbb{R}^N \times \mathbb{R}^N \rightarrow \mathbb{R}^N$ is a regular iteration function for $\mathbf{F} : \mathcal{D} \subseteq \mathbb{R}^N \rightarrow \mathbb{R}^N$ on the subset $\mathcal{D}_0 \subseteq \mathcal{D}$ if

1. $\mathbf{G}(\mathbf{x}, \mathbf{x}) = \mathbf{F}(\mathbf{x})$ for all $\mathbf{x} \in \mathcal{D}_0$,
2. $\mathbf{G}(\cdot, \mathbf{x}) : \mathcal{D}_0 \rightarrow \mathbb{R}^N$ is inverse isotone for any fixed $\mathbf{x} \in \mathcal{D}_0$, and
3. $\mathbf{G}(\mathbf{y}, \cdot) : \mathcal{D}_0 \rightarrow \mathbb{R}^N$ is antitone for any fixed $\mathbf{y} \in \mathcal{D}_0$.

Thm. 18 (Iteration Function for Block Nonlinear Gauss Seidel)**:** ([64] Thm. 6.5) Let $\mathbf{F} : \mathbb{R}^N \rightarrow \mathbb{R}^N$ be a M-function; then the mapping $\mathbf{G} : \mathbb{R}^N \times \mathbb{R}^N \rightarrow \mathbb{R}^N$ defined by

$$\mathbf{G}(\mathbf{y}, \mathbf{x}) = \begin{bmatrix} \mathbf{F}_0(\mathbf{y}_0, \mathbf{x}_1) \\ \mathbf{F}_1(\mathbf{y}_0, \mathbf{y}_1) \end{bmatrix}$$

is a regular iteration function for $\mathbf{F}$ on $\mathbb{R}^N$. If, in addition, $\mathbf{F}$ is continuous and onto $(\mathbb{R}^N)$, then $\mathbf{G}(\cdot, \mathbf{x}) : \mathbb{R}^N \rightarrow \mathbb{R}^N$ is onto $(\mathbb{R}^N)$ for any fixed $\mathbf{x} \in \mathbb{R}^N$.

Thm. 19 (Convergence of Block Nonlinear Gauss Seidel)**:** ([64] Thm. 6.4) Let $\mathbf{F} : \mathbb{R}^N \rightarrow \mathbb{R}^N$ be continuous, inverse isotone, and onto ($\mathbb{R}^N$). Suppose, further, that $\mathbf{G} : \mathbb{R}^N \times \mathbb{R}^N \rightarrow \mathbb{R}^N$ is a regular iteration function for $\mathbf{F}$ on $\mathbb{R}^N$ with the property that $\mathbf{G}(\cdot, \mathbf{x}) : \mathbb{R}^N \rightarrow \mathbb{R}^N$ is onto ($\mathbb{R}^N$) for any fixed $\mathbf{x} \in \mathbb{R}^N$. Then, for any $\mathbf{z} \in \mathbb{R}^N$ and any initial point $\mathbf{x}^{(0)} \in \mathbb{R}^N$, the process

$$\mathbf{G}(\mathbf{x}^{(k+1)}, \mathbf{x}^{(k)}) = \mathbf{z}$$

converges to the unique solution $\mathbf{x}^* \in \mathbb{R}^N$ of $\mathbf{F}(\mathbf{x}) = \mathbf{z}$.

Appendix B

Convex Analysis

Perhaps the most widely cited rigorous coverage of convex analysis is [53]. Good texts for engineers, who are likely to wish for a more pragmatic approach, include [80] and [81].

Def. 14 (convex set)**:** ([53] pp. 10) A subset $\mathcal{A} \subseteq \mathbb{R}^N$ is convex if for all $\mathbf{x}, \mathbf{y} \in \mathcal{A}$ and $\lambda \in [0,1]$ $\mathbf{z} := \lambda\mathbf{x} + (1-\lambda)\mathbf{y}$ is in $\mathcal{A}$.

Def. 15 (dom (h))**:** dom (h) is the set of points $\mathbf{x}$ in the domain of h which have $\mathsf{h}(\mathbf{x}) < \infty$.

Def. 16 (convex function, I)**:** [53] A function $\mathsf{h} : \mathbb{R}^N \rightarrow \mathbb{R}$ is convex if its epigraph

$$\{(u, \mathbf{x}) \in \mathbb{R}^{N+1} | u \geq \mathsf{h}(\mathbf{x})\}$$

is a convex subset of $\mathbb{R}^{N+1}$.

Def. 17 (convex function, II)**:** ([81] pp. 67) A function $\mathsf{h} : \mathbb{R}^N \rightarrow (\mathbb{R} \bigcup \{\infty\})$ is convex if dom h is a convex set and if for all $\mathbf{x}, \mathbf{y} \in$ dom (h) and $\lambda \in [0,1]$ we have

$$\mathsf{h}(\lambda\mathbf{x} + (1-\lambda)\mathbf{y}) \leq \lambda\mathsf{h}(\mathbf{x}) + (1-\lambda)\mathsf{h}(\mathbf{y})$$

Def. 18 (Interior int)**:** ([81] pp. 638) The interior of a set $\mathcal{A} \subseteq \mathbb{R}^N$ is the union of all open (with respect to $\mathbb{R}^N$) sets contained in $\mathcal{A}$.

Def. 19 (Boundary bd)**:** ([81] pp. 638) The boundary of a set is the difference between its closure and its interior.

Def. 20 (Convex Conjugate)**:** The convex conjugate of a closed convex function

$\mathsf{h} : \mathbb{R}^N \to \mathbb{R}$ is the function $\mathsf{h}^* : \mathbb{R}^N \to \mathbb{R} \bigcup \infty$ defined by

$$\mathsf{h}^*(\mathbf{x}^*) = - \inf_{\mathbf{x} \in \mathbb{R}^N} \{\mathbf{x} \cdot \mathbf{x}^* - \mathsf{h}(\mathbf{x})\}$$

Def. 21 (Convex Function of Legendre Type)**:** [18] We say h is a convex function of Legendre type if it is a closed convex proper function on $\mathbb{R}^K$ with $\mathsf{int}(\mathrm{dom}\ \mathsf{h}) \neq \emptyset$ satisfies every one of the following conditions

1. h is differentiable on $\mathsf{int}(\mathrm{dom}\ \mathsf{h})$
2. $\lim_{t \to 0^+} \langle \nabla \mathsf{h}(\mathbf{x} + t(\mathbf{y} - \mathbf{x})), \mathbf{y} - x \rangle = -\infty$ for all $\mathbf{x} \in \mathrm{bd}\ (\mathrm{dom}\ \mathsf{h})$ and for all $\mathbf{y} \in \mathsf{int}(\mathrm{dom}\ \mathsf{h}))$.
3. h is strictly convex on $\mathsf{int}(\mathrm{dom}\ \mathsf{h})$.

Prop. 9 (Convex Conjugate Gradient Inverse)**:** ([18] cites from [53] thm. 26.5) A convex function h is Legendre if and only if its conjugate h^* is. In this case, the gradient mapping

$$\nabla \mathsf{h} : \mathsf{int}(\mathrm{dom}\ (\mathsf{h})) \to \mathsf{int}(\mathrm{dom}\ (\mathsf{h}^*)) : \mathbf{x} \mapsto \nabla \mathsf{h}(\mathbf{x})$$

is a topological isomorphism with inverse mapping $(\nabla \mathsf{h})^{-1} = \nabla \mathsf{h}^*$.

Appendix C

Nonlinear Programming and the Calculus of Variations

C.1 Finite Dimensional Constrained Optimization: Nonlinear Programming

This section of the appendix collects the theory from nonlinear programming and the calculus of variations necessary for the theoretical ideas presented in this dissertation. Of particular interest will be finite dimensional optimization problems of the form

$$\min_{\mathbf{x}\in\mathcal{C}} \mathsf{f}(\mathbf{x}) \tag{C.1}$$

where

$$\mathcal{C} := \left\{\mathbf{x} \in \mathbb{R}^{\mathtt{N}} | \mathsf{h}_{\mathtt{i}}(\mathbf{x}) = 0 \ \forall \mathtt{i} \in \{1, \ldots, \mathtt{H}\},\ \mathsf{g}_{\mathtt{j}}(\mathbf{x}) \leq 0 \ \forall \mathtt{j} \in \{1, \ldots, \mathtt{G}\}\right\}$$

and f, h, and g are continuously differentiable functions from $\mathbb{R}^{\mathtt{N}}$ to $\mathbb{R}$. Later, for notational convenience, we will use the vector function notation

$$\mathbf{h}(\mathbf{x}) := [\mathsf{h}_1(\mathbf{x}), \ldots, \mathsf{g}_{\mathtt{H}}(\mathbf{x})]^T$$

and $\mathbf{g}(\mathbf{x}) := [\mathsf{g}_1(\mathbf{x}), \ldots, \mathsf{g}_{\mathtt{G}}(\mathbf{x})]$.

Def. 22 (Local Minimum)**:** A point $\mathbf{x}^* \in \mathcal{C}$ is said to be a local minimum of (C.1) if there exists an $\epsilon \geq 0$ such that

$$\forall \mathbf{x} \in \{\mathbf{y} \in \mathbb{R}^{\mathtt{N}} | \|\mathbf{y} - \mathbf{x}^*\| \leq \epsilon\} \bigcap \mathcal{C}, \quad \mathsf{f}(\mathbf{x}) \geq \mathsf{f}(\mathbf{x}^*)$$

Thm. 20 (Fritz John Necessary Conditions)**:** ([54] Prop. 3.3.5, pp. 324) Let $\mathbf{x}^*$ be local minimum of the problem (C.1). Then there exist a scalar λ_0^* and multipliers $\boldsymbol{\lambda}^* := [\lambda_1^*, \ldots, \lambda_H^*]$ and $\boldsymbol{\mu}^* := [\mu_1^*, \ldots, \mu_G^*]$ with $\lambda_0^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*$ not all equal to zero, such that

$$\lambda_0^* \nabla_{\mathbf{x}} f(\mathbf{x}^*) + \sum_{i=1}^{H} \lambda_i^* \nabla_{\mathbf{x}} h_i(\mathbf{x}^*) + \sum_{j=1}^{G} \mu_j^* \nabla_{\mathbf{x}} g_j(\mathbf{x}^*) = \mathbf{0}$$

and

$$\mu_j^* \geq 0 \quad \forall j \in \{1, \ldots, G\}$$

and such that for every neighborhood $\mathcal{O}(\mathbf{x}^*)$ of $\mathbf{x}^*$ there is an $\mathbf{x} \in \mathcal{O}(\mathbf{x}^*)$ such that $\lambda_i^* h_i(\mathbf{x}) > 0$ for all i such that $\lambda_i^* \neq 0$ and $\mu_j^* g_j(\mathbf{x}) > 0$ for all j such that $\mu_j^* \neq 0$.

Def. 23 (Dual Function)**:** The dual function to the constrained optimization problem (C.1) is the function $d : \mathbb{R}^H \times \mathbb{R}^G \to \mathbb{R} \bigcup \{\pm\infty\}$ defined by

$$d(\boldsymbol{\lambda}, \boldsymbol{\mu}) := \inf_{\mathbf{x} \in \mathbb{R}^N} f(\mathbf{x}) + \boldsymbol{\lambda} \cdot \mathbf{h}(\mathbf{x}) + \boldsymbol{\mu} \cdot \mathbf{g}(\mathbf{x}) \tag{C.2}$$

The dual has the nice property that its effective domain,

$$\text{dom } d := \{(\boldsymbol{\lambda}, \boldsymbol{\mu}) | d(\boldsymbol{\lambda}, \boldsymbol{\mu}) > -\infty\}$$

is convex, and furthermore that d is concave within the effective domain.

Prop. 10 (Properties of the Dual)**:** The effective domain of of the dual function is convex, and the dual function is concave on its effective domain.

Def. 24 (Geometric Multiplier)**:** ([54] Defn. 5.1.1, pp. 488) A vector $(\boldsymbol{\lambda}^*, \boldsymbol{\mu}^*)$ is said to be a geometric multiplier for the primal problem (C.1) if $\boldsymbol{\mu}^* \geq \mathbf{0}$ and

$$\inf_{\mathbf{x} \in \mathbb{R}^N} f(\mathbf{x}) + \boldsymbol{\lambda}^* \cdot \mathbf{h}(\mathbf{x}) + \boldsymbol{\mu}^* \cdot \mathbf{g}(\mathbf{x}) = \inf_{\mathbf{x} \in \mathcal{C}} f(\mathbf{x})$$

It turns out that the dual function (C.2) is intimately related to the optimization problem (C.1).

Thm. 21 (Weak Duality Theorem)**:** ([54] Prop. 5.1.3, 5.1.4 pp. 495) The supremum of the dual function underbounds the optimum value of the primal optimization problem (C.1).

$$\sup_{\boldsymbol{\lambda}\in\mathbb{R}^{H},\boldsymbol{\mu}\in[0,\infty)^{G}} \mathsf{d}(\boldsymbol{\lambda},\boldsymbol{\mu}) \leq \inf_{\mathbf{x}\in\mathcal{C}} \mathsf{f}(\mathbf{x}) \tag{C.3}$$

Def. 25 (Duality Gap)**:** If equality is attained in (C.3) we say that there is *no duality gap*, likewise if the inequality in (C.3) is strict, we say that there is *a duality gap*.

Prop. 11 (Duality Gap and Geometric Multipliers)**:** ([54], Prop. 5.1.4 pp. 495) If there is no duality gap, then the set of geometric multipliers is equal to the set of optimal dual solutions

$$\left\{(\boldsymbol{\lambda},\boldsymbol{\mu})\in\mathbb{R}^{H}\times[0,\infty)^{G}\,\middle|\,\inf_{\mathbf{x}\in\mathbb{R}^{N}}\mathsf{f}(\mathbf{x})+\boldsymbol{\lambda}\cdot\mathbf{h}(\mathbf{x})+\boldsymbol{\mu}\cdot\mathbf{g}(\mathbf{x})=\inf_{\mathbf{x}\in\mathcal{C}}\mathsf{f}(\mathbf{x})\right\}=$$
$$\left\{(\boldsymbol{\lambda},\boldsymbol{\mu})\in\mathbb{R}^{H}\times[0,\infty)^{G}\,\middle|\,\mathsf{d}(\boldsymbol{\lambda},\boldsymbol{\mu})=\sup_{(\boldsymbol{\lambda}',\boldsymbol{\mu}')\in\mathbb{R}^{H}\times[0,\infty)^{G}}\mathsf{d}(\boldsymbol{\lambda}',\boldsymbol{\mu}')\right\} \tag{C.4}$$

and if there is a duality gap the set of geometric multipliers is empty.

C.2 Constrained Optimization w.r.t. a Function: Calculus of Variations

In this section of the appendix, we seek to generalize the results for nonlinear programming problems which involve a finite number of optimization variables to situations where we seek to minimize a functional over a family of multivariate functions which are required to satisfy a finite number of functional constraints. We will be working with functions in a normed linear space of functions.

Def. 26 (Linear Space)**:** (e.g. [41] pp. 5) A linear space is a set $\mathcal{L}$ of elements for

which addition and multiplication by a real number is defined and obeys the following axioms $\forall \mathsf{h}, \mathsf{g} \in \mathcal{L}$ and $\forall \alpha, \beta \in \mathbb{R}$:

1. $\mathsf{h} + \mathsf{g} = \mathsf{g} + \mathsf{h}$
2. $(\mathsf{h} + \mathsf{g}) + z = \mathsf{h} + (\mathsf{g} + z)$
3. There exists $0 \in \mathcal{L}$ such that $\mathsf{h} + 0 = \mathsf{h}$ for any $\mathsf{h} \in \mathcal{L}$.
4. For each $\mathsf{h} \in \mathcal{L}$, there exists an element $-\mathsf{h}$ such that $\mathsf{h} + (-\mathsf{h}) = 0$.
5. $1\mathsf{h} = \mathsf{h}$
6. $\alpha(\beta\mathsf{h}) = (\alpha\beta)\mathsf{h}$
7. $(\alpha + \beta)\mathsf{h} = \alpha\mathsf{h} + \beta\mathsf{h}$
8. $\alpha(\mathsf{h} + \mathsf{g}) = \alpha\mathsf{h} + \alpha\mathsf{g}$.

Def. 27 (Normed Linear Space)**:** (e.g. [41] pp. 6) A linear space $\mathcal{L}$ is said to be *normed*, if each element $\mathsf{h} \in \mathcal{L}$ is assigned a nonnegative number $\|\mathsf{h}\|$, called the norm of h such that

1. $\|\mathsf{h}\| = 0$ if and only if $\mathsf{h} = 0$.
2. $\|\alpha\mathsf{h}\| = |\alpha|\|\mathsf{h}\|$.
3. $\|\mathsf{h} + \mathsf{g}\| \leq \|\mathsf{h}\| + \|\mathsf{g}\|$.

Def. 28 (Functional)**:** A functional $\mathfrak{F}$ is a map $\mathfrak{F} : \mathcal{D} \to \mathbb{R}$ which associates every member f of a subset of a normed linear space $\mathcal{D} \subseteq \mathcal{L}$ with a real number $\mathfrak{F}[\mathsf{f}] \in \mathbb{R}$.

We see then, that a functional is really only a real valued function whose domain is a space of functions. A special subset of the class of functionals are linear functionals.

Def. 29 (Linear Functional)**:** ([41] pp. 8) A functional $\mathfrak{F}$ is said to be a (continuous) linear functional if

1. $\mathfrak{F}[\alpha \mathsf{f}] = \alpha \mathfrak{F}[\mathsf{f}]$ for all $\alpha \in \mathbb{R}$ and $\mathsf{f} \in \mathcal{D}$.
2. $\mathfrak{F}[\mathsf{h} + \mathsf{g}] = \mathfrak{F}[\mathsf{h}] + \mathfrak{F}[\mathsf{g}]$ for any $\mathsf{h}, \mathsf{g} \in \mathcal{D}$.
3. $\mathfrak{F}$ is continuous, so that $\forall \mathsf{f} \in \mathcal{D}$ for any real $\epsilon > 0$ there exists a $\delta > 0$ such that

$$\|\mathsf{h} - \mathsf{f}\| < \delta \implies |\mathfrak{F}[\mathsf{h}] - \mathfrak{F}[\mathsf{g}]| < \epsilon$$

A notion which extends the idea of a derivative for a real valued function on Euclidean space to functionals on function spaces is the variation.

Def. 30 (Variation of a Functional)**:** ([41] Sec. 3.2 pp. 11) Fix a $\mathsf{h} \in \mathcal{L}$ and suppose that

$$\mathfrak{F}[\mathsf{h} + \mathsf{g}] = \mathfrak{d}[\mathsf{h}; \mathsf{g}] + \epsilon \|\mathsf{g}\| \tag{C.5}$$

with $\epsilon \to 0$ as $\mathsf{g} \to 0$, and $\mathfrak{d}$ a linear functional in the second argument with the first argument fixed. Then $\mathfrak{d}$ is called the variation of $\mathfrak{F}$, and is denoted by $\frac{\delta \mathfrak{F}}{\delta \mathsf{h}}[\mathsf{h}; \mathsf{g}]$.

If the variation of a function exists, so that (C.5) holds, we say that $\mathfrak{F}[\mathsf{h}]$ is differentiable.

Def. 31 (Relative Extremum)**:** ([41] pp. 13) We say that h is a relative extremum of $\mathfrak{F}$ if $\mathfrak{F}[\mathsf{h}] - \mathfrak{F}[\mathsf{g}]$ does not change sign for all g in some neighborhood of h.

Thm. 22 (Variation and Relative Extrema)**:** ([41] Thm. 2, pp. 13) A necessary condition for the differentiable functional $\mathfrak{F}$ to have an extremum at h is for its variation to vanish at h, that is

$$\frac{\delta \mathfrak{F}}{\delta \mathsf{h}}[\mathsf{h}; \mathsf{g}] = 0 \quad \forall \mathsf{g}$$

Prop. 12 (Multidimensional Integral Functional)**:** ([82] XV.3, pp. 462) When the functional takes the special form

$$\mathfrak{F}[\mathsf{f}] := \int_{\mathcal{A}} \mathsf{J}(\mathbf{x}, \mathsf{f}(\mathbf{x}))\mathrm{d}\mathbf{x} \tag{C.6}$$

where $\mathsf{J} : \mathbb{R}^{\mathtt{N}} \times \mathbb{R} \to \mathbb{R}$ is some continuously differentiable function, the variation $\delta\mathfrak{F}$ is equal to zero if and only if

$$\nabla_2 \mathsf{J}(\mathbf{x}, \mathsf{f}(\mathbf{x})) = \mathbf{0} \quad \forall \mathbf{x} \in \mathcal{A}$$

In the previous theorems, we have used the notation $\nabla_2 \mathsf{J}(\mathbf{x}, \mathsf{f}(\mathbf{x}))$ to represent the vector of partial derivatives with respect to the dependent variables of the function J that are not being integrated over in the functional $\mathfrak{F}$, so that $\nabla_2 \mathsf{J}(\mathbf{x}, \mathbf{y}) := \nabla_{\mathbf{y}} \mathsf{J}(\mathbf{x}, \mathbf{y})$. We will continue to use this notation for the remainder of this appendix.

We will also be interested in optimization of functionals with respect to functions which must satisfy some functional constraints. That is, we will be interested in problems of the form

$$\inf_{\mathsf{f} \in \mathcal{C}} \mathfrak{J}(\mathsf{f}) := \int_{\mathcal{A}} \mathsf{J}(\mathbf{x}, \mathsf{f}(\mathbf{x}))\mathrm{d}\mathbf{x} \tag{C.7}$$

where the constraint set is defined as

$$\mathcal{C} = \left\{ \mathsf{f} \,\middle|\, \mathfrak{C}_{\mathtt{i}}(\mathsf{f}) := \int_{\mathcal{A}} \mathsf{C}_{\mathtt{i}}(\mathbf{x}, \mathsf{f}(\mathbf{x}))\mathrm{d}\mathbf{x} - c_{\mathtt{i}} = 0 \quad \forall \mathtt{i} \in \{1, \ldots, \mathtt{H}\} \right\}$$

Thm. 23 (Constrained Functional Extrema)**:** ([82] XV.5, pp. 487, [41] pp. 45) Suppose we wish to solve (C.7) where each of the functionals is of the form (C.6), so that

$$\mathfrak{J}(\mathsf{f}) := \int_{\mathcal{A}} \mathsf{J}(\mathbf{x}, \mathsf{f}(\mathbf{x}))\mathrm{d}\mathbf{x}$$

and

$$\mathfrak{C}_{\mathtt{i}}(\mathsf{f}) := \int_{\mathcal{A}} \mathsf{C}_{\mathtt{i}}(\mathbf{x}, \mathsf{f}(\mathbf{x}))\mathrm{d}\mathbf{x} - c_{\mathtt{i}} \quad \forall \mathtt{i} \in \{1, \ldots, \mathtt{H}\}$$

Then a necessary condition for a local extremum at f^* is the existence of Lagrange multipliers $\boldsymbol{\lambda} \in \mathbb{R}^{\mathtt{H}}$ such that

$$\nabla_2 \mathsf{J}(\mathbf{x}, \mathsf{f}^*(\mathbf{x})) + \sum_{\mathtt{i}=1}^{\mathtt{H}} \lambda_{\mathtt{i}} \nabla_2 \mathsf{C}_{\mathtt{i}}(\mathbf{x}, \mathsf{f}^*(\mathbf{x})) = \mathbf{0} \quad \forall \mathbf{x} \in \mathcal{A}$$

provided that f^* is not an extremal of any $\mathfrak{C}_{\mathtt{i}}$.

Def. 32 (Dual of a Isoperimetric Variational Problem)**:** The dual function of the constrained variational optimization (C.7) is a function $\mathsf{d} : \mathbb{R}^{\mathtt{H}} \rightarrow \mathbb{R} \bigcup \{\pm\infty\}$ defined by

$$\mathsf{d}(\boldsymbol{\lambda}) := \inf_{\mathsf{f} \in \mathcal{L}} \mathfrak{J}(\mathsf{f}) + \sum_{\mathtt{i}=1}^{\mathtt{H}} \lambda_{\mathtt{i}} \mathfrak{C}_{\mathtt{i}}(\mathsf{f}) \tag{C.8}$$

It is claimed on pp. 508 of [80] part (f) that Propositions 10 and 11 and Theorem 21 for the finite dimensional nonlinear programming case carry over to the variational case "with essentially no modification." This motivates interest in the dual function.

Appendix D

Exponential Family Distributions and Information Geometry

This section of the appendix collects the elements of the theory of exponential family distributions and information geometry that were necessary for the theoretical ideas presented in this dissertation.

Def. 33 (k dimensional exponential family)**:** [20] Let $\mathcal{F} = \{\mathsf{F}_i | i \in \mathcal{I}\}$ be a family of distributions on a probability (i.e. measurable) space $(\mathcal{X}, \mathscr{X})$. Suppose that F_i is absolutely continuous with respect to ν ($\mathsf{F}_i \ll \nu$) for all $i \in \mathcal{I}$. Suppose further that there exist functions $\mathsf{C} : \mathcal{I} \to (0, \infty)$, $\mathbf{R} : \mathcal{I} \to \mathbb{R}^k$, $\mathsf{h} : \mathcal{X} \to [0, \infty)$ (Borel measurable), $\mathbf{T} : \mathcal{X} \to \mathbb{R}^k$ (Borel measurable), such that

$$\mathsf{f}_i(y) := \frac{d\mathsf{F}_i}{d\mu} = \mathsf{C}(i)\mathsf{h}(y) \exp\left(\mathbf{R}(i) \cdot \mathbf{T}(y)\right)$$

Then $\mathcal{F}$ is called a *k dimensional family of distributions* and $\{\mathsf{f}_i | i \in \mathcal{I}\}$ is called a *k dimensional family of densities*.

D.1 Standard Exponential Families

Def. 34 (k dimensional standard exponential family)**:** [20] Let ν be a σ-finite measure on the Borel subsets of $\mathbb{R}^k$. Define the *natural parameter space*

$$\mathcal{N} := \{\boldsymbol{\theta} : \int \exp(\boldsymbol{\theta} \cdot \mathbf{x}) \nu(d\mathbf{x}) < \infty\}$$

and define the *cumulant generating function* aka the *log partition function* as

$$\psi(\boldsymbol{\theta}) := \log \left(\int \exp(\boldsymbol{\theta} \cdot \mathbf{x}) \nu(d\mathbf{x}) \right)$$

Let $\Theta \subseteq \mathcal{N}$ and for each $\boldsymbol{\theta} \in \Theta$ define the probability densities

$$\mathsf{p}_{\boldsymbol{\theta}}(\mathbf{x}) := \exp(\boldsymbol{\theta} \cdot \mathbf{x} - \psi(\boldsymbol{\theta}))$$

Then the family of probability densities

$$\{\mathsf{p}_{\boldsymbol{\theta}}(\mathbf{x}) : \boldsymbol{\theta} \in \Theta\}$$

is called a k-dimensional standard exponential family of probability densities.

The family is called *full* if $\Theta = \mathcal{N}$ and the family is called *regular* if $\mathcal{N}$ is open, so that

$$\mathcal{N} = \mathsf{int}\mathcal{N} := \left\{ \bigcup \mathcal{A} | \mathcal{A} \subset \mathcal{N}, \mathcal{A} \text{ is open} \right\}$$

Prop. 13: Any k dimensional exponential family can be reduced by sufficiency, reparameterization, and proper choice of ν to a k dimensional standard exponential family. The sufficient statistic is $\mathbf{x} = \mathbf{T}(y)$ and its distributions form an exponential family with canonical parameter $\boldsymbol{\theta} = \mathbf{R}(i)$.

D.2 Minimal Standard Exponential Families

With respect to the definition of the standard exponential family, define the support of ν, $\mathsf{supp}\ \nu$ to be the minimal closed set $\mathcal{S} \subset \mathbb{R}^k$ such that $\nu(\mathcal{S}^c) = 0$. Then define the *convex support* of ν to be

$$\mathcal{K} = \mathsf{clo\ conv\ supp}\ \nu$$

Then the dimension $\dim \mathcal{K}$ is defined as the dimension of the linear space spanned by the vectors $\{(\mathbf{v}_1 - \mathbf{v}_2) | \mathbf{v}_1, \mathbf{v}_2 \in \mathcal{K}\}$.

Def. 35 (Minimal Standard Exponential Family)**:** A k dimensional standard exponential family is called minimal if $\dim \mathcal{N} = \dim \mathcal{K} = k$.

Thm. 24 (Reduction to a Minimal Standard Family)**:** Any k dimensional exponential family can be reduced by sufficiency, reparametrization, and proper choice of ν to a m dimensional minimal standard exponential family for some $m \leq k$. This reduction is unique up to any (bijective) affine transformation on the parameters with a corresponding affine transformation on the sufficient statistics.

Minimal standard exponential families have the property that the map between their natural parameters and the expectation of their sufficient statistics is one to one within the relative interior of $\mathcal{N}$.

D.3 Members of the Exponential Families

In this section we show how many common parameterized densities/distributions are exponential families of distributions, and we reduce them to a minimal standard exponential family, providing the sufficient statistics and parameters. Members of the group of exponential exponential families include the normal distribution, multivariate normal distribution, exponential and gamma distributions, the beta distribution, the binomial and multinomial distributions, the geometric and negative binomial distributions, the Poisson distribution, the Log normal distribution, the Weibull distribution, any finite discrete distribution (i.e. distribution with non-zero measure only at a finite number of points), any collection of independent random variables distributed according to exponential families, sums of random variables drawn from distributions from exponential families, and linear functions of vectors drawn from standard exponential families. Some of these member's minimal standard representations are worked out in Table D.3.

Table D.1: Some common exponential family distributions. Table entries compiled from [20, 21, 22, 23]

distribution	**min. suff. stat.**	**ref. measure** $\mathsf{d}\boldsymbol{\theta}$	**supp. set**
normal	$[x, x^2]^T$	Lebesgue	$\mathbb{R}$
multivariate normal	$\left[\mathbf{x}^T, [x_{\mathtt{i}} x_{\mathtt{j}}]_{\mathtt{j} \geq \mathtt{i}}\right]^T$	Lesbegue	$\mathbb{R}^{\mathtt{N}}$
exponential	$[-x]$	Lebesgue	$(0, \infty)$
beta	$[\log(x), \log(1-x)]^T$	Lebesgue	$(0, 1)$
Poisson	$[x]$	counting $1/(x!)$	$\{0, 1, \ldots, \}$
gamma	$[x, \ln(x)]^T$	Lebesgue	$(0, \infty)$
bin. (fix N**)**	$[x]$	counting $\binom{N}{x}$	$\{0, \ldots, N\}$
neg. bin., geom. (fix. α**)**	$[x]$	counting $\frac{\Gamma(x+\alpha)}{(\Gamma(x+1)\Gamma(\alpha))}$	$\{0, 1, \ldots\}$
multinom. (fix N	$\mathbf{x}$	counting $\frac{N!}{x_1! x_2! \cdots x_n!}$	$x_i \in \{0, \ldots, N\}$ and $\sum_i x_i = N$
log normal	$[\ln(x), \ln(x)^2]^T$	Lebesgue	$(0, \infty)$

One particularly important type of exponential family distribution not mentioned in Table D.3 are the discrete distributions which have Θ finite. For these the sufficient statistics take the form $\mathbf{t} : \mathbb{R} \rightarrow \mathbb{R}^{|\Theta|-1}$. Denote the possible values Θ of θ by $\{\alpha_1, \ldots, \alpha_{\mathtt{V}}\}$ then the sufficient statistics are

$$\mathbf{t}(\theta) := \begin{cases} \mathbf{e}_{\mathtt{j}} & \theta = \alpha_{\mathtt{j}},\ \mathtt{j} \neq 1 \\ \mathbf{0} & \text{otherwise} \end{cases}$$

where $\mathbf{e}_{\mathtt{j}}$ is the $\mathtt{j}$th column of the identity matrix of dimension $|\Theta| - 1$. The natural parameters $\boldsymbol{\lambda}$ are then the log likelihood ratios

$$[\boldsymbol{\lambda}]_{\mathtt{i}} = \log\left(\frac{\mathbb{P}[\theta = \alpha_{\mathtt{j}}]}{\mathbb{P}[\theta = \alpha_1]}\right)$$

which are often called the log coordinates of the pmf. Furthermore, the expectation of the sufficient statistics are the probabilities

$$\mathbb{E}\left[[\mathbf{t}]_{\mathtt{j}}(\theta)\right] = \mathbb{P}[\theta = \alpha_{\mathtt{j}}] \quad \forall \mathtt{j} \in \{2, \ldots, \mathtt{V}\}$$

Another set of important standard exponential family densities are those that are formed by considering independent standard exponential families jointly. Of course, the pdf in this case is just the product of pdfs of the different standard exponential families, and the joint log partition function is just the sum of the individual log partition functions.

D.4 Natural Coordinate Systems and Legendre Transform

One of the most important characteristics of the exponential family of densities is that of the existence of two "canonical" coordinate systems for any exponential family. These coordinates are "dual" to each other under the Legendre transform, and are the gradients of two important "potential" functions, the entropy and the

log partition function. This subject has been widely studied under various names and in various contexts, [83, 84] are some of the early references on the subject, [75] may be credited for collecting a lot of the important results and expounding on new ideas and applications, and [23] collects a lot of the fundamental ideas in an advanced manner (which humbles greatly the author of this dissertation). Of course, the duality between the entropy and the log partition function may be seen as a particular instance of a more abstract duality between convex functions, often called Fenchel conjugacy [53] (see Appendix B). Because of the sophistication used in [23], and the general accessibility of that report, we follow the same progression of results as presented there, and refer the reader interested in proofs to that document as well.

We begin by considering any standard exponential family distribution on a random vector $\boldsymbol{\theta} \in \Theta \subseteq \mathbb{R}^{\mathtt{N}}$ with sufficient statistics $\mathbf{t}(\boldsymbol{\theta}) \in \mathbb{R}^{\mathtt{V}}$, parameters $\boldsymbol{\lambda}$, and supporting measure $\mathsf{d}\boldsymbol{\theta}$. The log partition function $\psi_{\mathbf{t}} : \mathcal{K} \rightarrow \mathbb{R} \bigcup \{\infty\}$ is defined as

$$\psi_{\mathbf{t}}(\boldsymbol{\lambda}) := \log \left(\int_{\Theta} \exp \left(\boldsymbol{\lambda} \cdot \mathbf{t}(\boldsymbol{\theta}) \right) \mathsf{d}\boldsymbol{\theta} \right)$$

where

$$\mathcal{K} := \left\{ \boldsymbol{\lambda} \in \mathbb{R} | \int_{\Theta} \exp \left(\boldsymbol{\lambda} \cdot \mathbf{t}(\boldsymbol{\theta}) \right) \mathsf{d}\boldsymbol{\theta} < \infty \right\}$$

Prop. 14 (Convexity of the log partition function)**:** The log partition function is convex and is strictly convex if the sufficient statistics are minimal.

A fact which we have used ad nauseam in this dissertation is that the gradient of the log partition function is the expectation of the sufficient statistics

$$\nabla \psi_{\mathbf{t}} = \mathbb{E}_{\exp(\mathbf{t} \cdot \boldsymbol{\lambda} - \psi_{\mathbf{t}}(\boldsymbol{\lambda}))} \left[\mathbf{t} \right]$$

Furthermore, the mapping $\boldsymbol{\lambda}$ and $\nabla \psi_{\mathbf{t}}$ is one to one (and thus invertible) if and

only if the sufficient statistics are minimal.

Another key potential function is the (negative of) the Shannon entropy of the exponential family distribution written in terms of the expectations of its sufficient statistics $\boldsymbol{\eta} := \mathbb{E}[\mathbf{t}(\boldsymbol{\theta})]$, which is defined as

$$\mathsf{H}(\boldsymbol{\eta}) := \int_{\Theta} \exp\left(\mathbf{t}(\boldsymbol{\theta}) \cdot \boldsymbol{\lambda} - \psi_{\mathbf{t}}(\boldsymbol{\lambda})\right) \left(\mathbf{t}(\boldsymbol{\theta}) \cdot \boldsymbol{\lambda} - \psi_{\mathbf{t}}(\boldsymbol{\lambda})\right) \mathsf{d}\boldsymbol{\theta}$$

from which the nice relation drops out

$$\mathsf{H}(\boldsymbol{\eta}) = \boldsymbol{\eta} \cdot \boldsymbol{\lambda} - \psi_{\mathbf{t}}(\boldsymbol{\lambda})$$

In fact, we have that

$$\nabla_{\boldsymbol{\eta}} \mathsf{H} = \boldsymbol{\lambda}$$

so that $\nabla_{\boldsymbol{\eta}} \mathsf{H}$ and $\nabla_{\boldsymbol{\lambda}} \psi$ furnish inverses of each other. Still even more than this is true. Let $\boldsymbol{\eta}_0$ be the expectation of the sufficient statistics of the minimal standard exponential family q_0 and $\boldsymbol{\lambda}_1$ be the natural parameters of another member of the same minimal standard exponential family. Then we have

$$\mathsf{H}(\boldsymbol{\eta}_0) + \psi(\boldsymbol{\lambda}_1) = \boldsymbol{\eta}_0 \cdot (\boldsymbol{\lambda}_0 - \boldsymbol{\lambda}_1) - \psi_{\mathbf{t}}(\boldsymbol{\lambda}_0) + \psi_{\mathbf{t}}(\boldsymbol{\lambda}_1) = \mathfrak{D}(\mathsf{q}_0 \| \mathsf{q}_1) + \boldsymbol{\eta}_0 \cdot \boldsymbol{\lambda}_1$$

Which shows that

$$\mathsf{H}(\boldsymbol{\eta}_0) + \psi(\boldsymbol{\lambda}_1) \geq \boldsymbol{\eta}_0 \cdot \boldsymbol{\lambda}_1$$

with equality if and only if $\mathsf{q}_0 = \mathsf{q}_1$. This is one (oversimplified) representation of the fact that $\psi(\boldsymbol{\lambda})$ and $\mathsf{H}(\boldsymbol{\eta})$ are Fenchel (convex) conjugates of each other which suffices for the purposes of this dissertation.

BIBLIOGRAPHY

[1] T. P. Minka, "A Family of Algorithms for Approximate Bayesian Inference," Ph.D. dissertation, Massachusetts Institute of Technology, 2001.

[2] ——, "Expectation propagation for approximate Bayesian inference," in *Uncertainty in AI'01*, 2001.

[3] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes," in *IEEE International Conference on Communications*, vol. 2, Geneva, May 1993, pp. 1064–1070.

[4] C. Berrou and A. Glavieux, "Near optimum error correction coding and decoding: Turbo codes," *IEEE Trans. Commun.*, vol. 44, pp. 1262–1271, Oct. 1996.

[5] S. Benedetto and G. Montorsi, "Iterative decoding of serially concatenated convolutional codes," *Electronics Letters*, vol. 32, p. 1186 1188, June 1996.

[6] ——, "Unveiling turbo codes: Some results on parallel concatenated coding structures," *IEEE Trans. Inform. Theory*, vol. 42, pp. 409–428, Mar. 1996.

[7] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara, "Serial concatenation of interleaved codes: performance analysis, design, and iterative decoding," *IEEE Trans. Inform. Theory*, vol. 44, pp. 909–926, May 1998.

[8] R. G. Gallager, *Low Density Parity-Check Codes*. Cambridge, MA: MIT Press, 1963.

[9] ——, "Low-density parity-check codes," *IRE Trans. Information Theory*, vol. 2, pp. 21–28, 1962.

[10] D. J. C. MacKay, "Good error-correcting codes based on very sparse matrices," *IEEE Trans. Inform. Theory*, vol. 45, pp. 399–431, Mar. 1999.

[11] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the ASME–Journal of Basic Engineering*, vol. 82, no. Series D, pp. 35–45, 1960.

[12] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inform. Theory*, vol. 20, pp. 284–287, Mar. 1974.

[13] J. Pearl, *Probabilistic reasoning in intelligent systems : networks of plausible inference*. Morgan Kaufmann Publishers, 1988.

[14] F. R. Kshischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inform. Theory*, vol. 47, pp. 498–519, Feb. 2001.

[15] J. M. Ortega and W. C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*. Academic Press, 1970.

[16] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. Athena Scientific, 1997.

[17] R. Dykstra, "An Iterative Procedure for Obtaining I-Projections onto the Intersection of Convex Sets," *The Annals of Probability*, vol. 13, no. 3, pp. 975–984, Aug. 1985.

[18] H. Bauschke and A. Lewis, "Dykstra's algorithm with Bregman projections: a convergence proof," *Optimization*, no. 48, pp. 409–427, 2000.

[19] T. J. Richardson, M. A. Shokrollahi, and R. L. Urbanke, "Design of Capacity-Approaching Irregular Low-Density Parity-Check Codes," *IEEE Trans. Inform. Theory*, no. 2, pp. 619–639, Feb. 2001.

[20] L. D. Brown, *Fundamentals of Statistical Exponential Families with Applications in Statistical Decision Theory*. Institute of Mathematical Statistics, 1986.

[21] Gérard Letac, *Lectures on Natural Exponential Families and their Variance Functions*. Rio de Janeiro: Institudo De Matemática Pura e Aplicada, 1992.

[22] Søren Johansen, *Introduction to the Theory of Regular Exponential Families*. University of Copenhagen: Institute of Mathematical Statistics, 1979.

[23] M. J. Wainwright and M. I. Jordan, "Graphical models, exponential families, and variational inference," Department of Statistics, University of California, Berkeley, Tech. Rep., 2003.

[24] C. E. Shannon, "A Mathematical Theory of Communication," *The Bell System Technical Journal*, vol. 27, pp. 379–432, 623–656, July and October 1948.

[25] C. Douillard, M. Jezequel, C. Berrou, P. Picart, P. Didier, and A. Glavieux, "Iterative correction of intersymbol interference: Turbo equalization." *European Telecommunications Transactions*, vol. 6, pp. 507–512, May 1995.

[26] C. Laot, A. Glavieux, and J. Labat, "Turbo equalization: Adaptive equalization and channel decoding jointly optimized." *IEEE J. Select. Areas Commun.*, vol. 19, pp. 673–680, Sept. 2001.

[27] M. Tuechler, R. Kotter, and A. Singer., "Turbo equalization: Principles and new results." *IEEE Trans. Commun.*, vol. 50, pp. 754–767, May 2002.

[28] J. R. Barry, A. Kavčić, S. W. McLaughlin, A. Nayak, and W. Zeng, "Iterative timing recovery," *IEEE Signal Processing Mag.*, pp. 89–102, January 2004.

[29] N. Noels, C. Herzet, A. Dejonghe, V. Lottici, H. Steendam, M. Moeneclaey, M. Luise, and L. Vandendorpe, "Turbo synchronization: an EM algorithm interpretation," in *IEEE International Conference on Communications*, vol. 4, 2003, pp. 2933–2937.

[30] B. Mielczarek and A. Svensson, "Timing error recovery in turbo-coded systems on AWGN channels," *IEEE Trans. Commun.*, vol. 50, pp. 1584–1592, Oct. 2002.

[31] S. Benedetto and E. Biglieri, *Principles of digital transmission : with wireless applications*. Kluwer Academic/Plenum Press, 1999.

[32] P. A. Regalia, "Iterative decoding of concatenated codes: A tutorial," *EURASIP J. Applied Signal Processing*, pp. 762–774, June 2005.

[33] R. J. McEliece, D. J. C. MacKay, and J.-F. Cheng, "Turbo decoding as an instance of Pearl's "belief propagation" algorithm." *IEEE J. Select. Areas Commun.*, vol. 16, pp. 140–152, Feb. 1998.

[34] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara, "Analysis, design, and iterative decoding of double serial concatenated codes with interleavers," *IEEE J. Select. Areas Commun.*, vol. 16, pp. 231–244, Feb. 1998.

[35] J. Yedidia, W. Freeman, and Y. Weiss, "Constructing free-energy approximations and generalized belief propagation algorithms," *IEEE Trans. Inform. Theory*, no. 7, pp. 2282–2312, July 2005.

[36] S. Ikeda, T. Tanaka, and S. Amari, "Information Geometry of Turbo and Low-Density Parity-Check Codes," *IEEE Trans. Inform. Theory*, vol. 50, pp. 1097–1114, June 2004.

[37] A. J. Grant, "Information geometry and iterative decoding," in *IEEE Communications Theory Workshop*, may 1999.

[38] S. L. Lauritzen, *Graphical Models*. Oxford University Press, 1996.

[39] H. V. Poor, *An Introduction to Signal Detection and Estimation*, 2nd ed. Springer-Verlag, 1994.

[40] S. B. Wicker and S. Kim, *Fundamentals of codes, graphs, and iterative decoding*. Kluwer Academic Publishers, 2003.

[41] I. M. Gelfand and S. V. Fomin, *Calculus of Variations*. Dover, 2000, english Translation by Richard A. Silverman.

[42] A. Montanari and N. Sourlas, "The statistical mechanics of turbo codes," *Eur. Phys. J. B.*, no. 18, pp. 107–109, 2000.

[43] S. Ikeda, T. Tanaka, and S. Amari, "Stochastic reasoning, free energy and information geometry," *Neural Computation*, pp. 1779–1810, 2004.

[44] P. Pakzad and V. Anantharam, "Belief propagation and statistical physics." in *Conference on Information Sciences and Systems*, Princeton University, Mar. 2002.

[45] ——, "Estimation and marginalization using kikuchi approximation methods." *Neural Computation*, pp. 1836–1876, Aug. 2005.

[46] S. ten Brink, "Convergence behavior of iteratively decoded parallel concatenated codes." *IEEE Trans. Commun.*, vol. 49, pp. 1727–1737, Oct. 2001.

[47] H. E. Gamal and A. R. Hammons, Jr., "Analyzing the turbo decoder using the gaussian approximation," *IEEE Trans. Inform. Theory*, vol. 47, pp. 671–686, Feb. 2001.

[48] M. Moher and T. A. Gulliver, "Cross-entropy and interative decoding," *IEEE Trans. Inform. Theory*, vol. 44, pp. 3097–3104, Nov. 1998.

[49] T. J. Richardson, "The geometry of turbo-decoding dynamics," *IEEE Trans. Inform. Theory*, vol. 46, pp. 9–23, Jan. 2000.

[50] B. Muquet, P. Duhamel, and M. de Courville, "Geometrical Interpretations of Iterative 'Turbo' Decoding," in *IEEE International Symposium on Information Theory*, June 2002.

[51] J. M. Walsh, P. A. Regalia, and C. R. Johnson, Jr., "A refined information geometric interpretation of turbo decoding," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Philadelpha, PA, Mar. 2005.

[52] P. Pakzad and V. Anantharam, "Kikuchi approximation method for joint decoding of ldpc codes and partial-response channels." submitted for publication to "IEEE Transactions on Communications".

[53] R. T. Rockafellar, *Convex Analysis*. Princeton University Press, 1970.

[54] D. P. Bertsekas, *Nonlinear Programming: 2nd Edition*. Athena Scientific, 1999.

[55] F. S. Acton, *Numerical Methods that Work*. Harper & Row Publishers, 1970.

[56] J. L. Buchanan and P. R. Turner, *Numerical Methods and Analysis*. McGraw Hill, Inc., 1992.

[57] G. H. Golub and C. F. V. Loan, *Matrix Computations 3rd Edition*. The Johns Hopkins University Press, 1996.

[58] J. M. Walsh, P. A. Regalia, and C. R. Johnson, Jr., "Turbo decoding as constrained optimization," in *43rd Allerton Conference on Communication, Control, and Computing.*, Sept. 2005.

[59] ——, "A convergence proof for the turbo decoder as an instance of the Gauss-Seidel iteration," in *IEEE International Symposium on Information Theory*, Adelaide, Australia, Sept. 2005.

[60] P. H. Tan and L. K. Rasmussen, "The serial and parallel belief propagation algorithms," in *IEEE International Symposium on Information Theory*, Adelaide, Australia, September 2005, pp. 729–733.

[61] P. H. Tan, "Simplified graphical approaches for cdma multi-user detection, decoding and power control," Ph.D. dissertation, Chalmers University of Technology, 2005.

[62] W. C. Rheinboldt, "On M-functions and their application to nonlinear Gauss-Seidel iterations and network flows," *Journal Mathematical Analysis and Applications*, pp. 274–307, 1971.

[63] J. J. Moré, "Nonlinear Generalizations of Matrix Diagonal Dominance with Application to Gauss-Seidel Iterations," *SIAM Journal on Numerical Analysis*, vol. 9, pp. 357–378, June 1972.

[64] W. C. Rheinboldt, "On classes of n-dimensional nonlinear mappings generalizing several types of matrices," in *Proc. Symposium on the Numerical Solution of Partial Differential Equations. II.* Academic Press, 1970, pp. 501–546.

[65] O. Axelsson, *Iterative Solution Methods.* Cambridge University Press, 1994.

[66] W. Rudin, *Principles of Mathematical Analysis 3rd Edition.* McGraw Hill, 1976.

[67] L. Gubin, B. Polyak, and E. Raik, "The method of projections for finding the common point of convex sets," *USSR Journal on Computational Mathematics and Mathematical Physics*, vol. 7, no. 6, pp. 1211–1228, 1967.

[68] L. M. Bregman, "Proof of the Convergence of Sheleikhovskii's Method for a Problem with Transportation Constraints," *USSR Journal on Computational Mathematics and Mathematical Physics*, vol. 7, no. 1, pp. 147–156, 1967.

[69] H. Bauschke and J. Borwein, "Legendre functions and the method of random Bregman projections," *Journal of Convex Analysis*, no. 4(1), pp. 27–67, 1997.

[70] H. Bauschke and D. Noll, "The method of forward projections," *Journal of Nonlinear and Convex Analysis*, vol. 3, no. 2, pp. 191–205, 2002.

[71] L. Bregman, Y. Censor, and S. Reich, "Dykstra's Algorithm as the Nonlinear Extension of Bregman's Optimization Method," *Journal of Convex Analysis*, vol. 6, no. 2, pp. 319–333, 1999.

[72] I. Csiszár and G. Tusnády, "Information geometry and alterating minimization procedures," *Statistics and Decisions, Supplement Issue*, pp. 205–237, 1984.

[73] H. Bauschke and J. Borwein, "Dykstra's alternating projection algorithm for two sets," *Journal of Approximation Theory*, no. 79(3), pp. 418–443, 1994.

[74] H. Bauschke, P. L. Combettes, and D. Noll, "Joint minimization with alternating bregman proximity operators," available online. [Online]. Available: http://mip.ups-tlse.fr/~noll/PAPERS/heinz.pdf

[75] S. Amari, *Methods of Information Geometry*. AMS Translations of Mathematical Monographs, 2004, vol. 191.

[76] I. Csiszár and F. Matúš, "Information projections revisited," *IEEE Trans. Inform. Theory*, vol. 49, pp. 1474–1490, June 2003.

[77] S. ten Brink, G. Kramer, and A. Ashikhmin, "Design of low-density parity-check codes for modulation and detection," *IEEE Trans. Commun.*, pp. 670 – 678, Apr. 2004.

[78] D. Divsalar, S. Dolinar, and F. Pollara, "Iterative turbo decoder analysis based on density evolution." *IEEE J. Select. Areas Commun.*, vol. 19, pp. 891–907, May 2001.

[79] T. J. Richardson and R. L. Urbanke, "The Capacity of Low-Density Parity-Check Codes Under Message-Passing Decoding," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 599–618, Feb. 2001.

[80] D. P. Bertsekas, A. Nedic, and A. E. Ozdaglar, *Convex analysis and optimization*. Athena Scientific, 2003.

[81] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[82] F. Y. Wan, *Introduction to the Calculus of Variations and Its Applications*. Chapman and Hall Mathematics, 1995.

[83] I. Csiszár, "I-Divergence Geometry of Probability Distributions and Minimization Problems," *The Annals of Probability*, vol. 3, no. 1, pp. 146–158, Feb. 1975.

[84] ——, "Information type measures of difference of probability distributions and indirect observations," *Studia Scientiarum Mathematicarum Hungarica*, vol. 2, 1967.

[85] Y. Kou, S. Lin, and M. P. C. Fossorier, "Low-density parity-check codes based on finite geometries: A rediscovery and new results," *IEEE Trans. Inform. Theory*, vol. 47, pp. 2711–2736, Nov. 2001.

[86] J. Hagenauer, E. Offer, and L. Papke, "Iterative decoding of binary block and convolutional codes," *IEEE Trans. Inform. Theory*, vol. 42, pp. 429–445, Mar. 1996.

[87] A. Giulietti, B. Bougard, and L. V. der Perre, *Turbo Codes. Desireable and Designable.* Kluwer Academic Publishers, 2004.

[88] C. Heegard and S. B. Wicker, *Turbo coding.* Kluwer Academic Publishers, 1999.

[89] J. Fan, *Constrained coding and soft iterative decoding.* Kluwer Academic Publishers, 2001.

[90] N. Wiberg, "Codes and decoding on general graphs," Ph.D. dissertation, Linköping University, Sweden, 1996.

[91] J. Hagenauer and P. Hoeher, "A Viterbi algorithm with soft-decision outputs and its applications," in *IEEE Global Telecommunications Conference*, vol. 3, Nov. 1989, pp. 1680–1686.

[92] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara, "Soft-input soft-output building blocks for the construction and distributed iterative decoding of code networks." *European Transactions on Telecommunciations*, vol. 9, pp. 155–172, Apr. 1998.

[93] E. Liu and J. Moura, "Fusion in sensor networks: Convergence study," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Montreal, Quebec, May 2004, pp. III–865–III–868.

[94] P. A. Regalia, "Contractivity in turbo iterations," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Montreal, Quebec, May 2004.

[95] D. Agrawal and A. Vardy, "The turbo decoding algorithm and its phase trajectories," *IEEE Trans. Inform. Theory*, vol. 47, pp. 699–722, Feb. 2001.

[96] A. Sella and Y. Be'ery, "Convergence analysis of turbo decoding of product codes," *IEEE Trans. Inform. Theory*, vol. 47, pp. 723–735, Feb. 2001.

[97] B. Frey, R. Koetter, and A. Vardy, "Signal-space characterization of iterative decoding," *IEEE Trans. Inform. Theory*, vol. 47, pp. 766–781, Feb. 2001.

[98] S. Amari, "Information geometry of the EM and em algorithms for neural networks," *Neural Networks*, vol. 8, no. 9, pp. 1379–1408, 1996.

[99] J. M. Walsh, P. A. Regalia, and C. R. Johnson, Jr., "The turbo decoder as a least squares cost gradient descent," in *IEEE International Workshop on Signal Processing Advances for Wireless Commun.*, New York, NY, June 2005.

[100] S.-Y. Chung, T. Richardson, and R. Urbanke, "Analysis of sum-product decoding of low-density parity-check codes using a gaussian approximation," *IEEE Trans. Inform. Theory*, vol. 47, pp. 657–670, Feb. 2001.

[101] Y. Weiss and W. Freeman, "On the optimality of solutions of the max-product belief-propagation algorithm in arbitrary graphs," *IEEE Trans. Inform. Theory*, vol. 47, pp. 736–744, Feb. 2001.

[102] P. Rusmevichientong and B. V. Roy, "An analysis of belief propagation on the turbo decoding graph with gaussian densities," *IEEE Trans. Inform. Theory*, vol. 47, pp. 745–765, Feb. 2001.

[103] D. Burshtein and G. Miller, "Expander graph arguments for message-passing algorithms," *IEEE Trans. Inform. Theory*, vol. 47, pp. 782–790, Feb. 2001.

[104] X. Wang and H. V. Poor, "Iterative (turbo) soft interference cancellation and decoding for coded CDMA." *IEEE Trans. Commun.*, vol. 47, pp. 1046–1061, July 1999.

[105] ——, "Blind joint equalization and multiuser detection for DS-CDMA in unknown correlated noise." *IEEE Trans. Circuits Syst. II*, vol. 46, pp. 886–895, July 1999.

[106] R. L. Bidan, C. Laot, D. LeRoux, and A. Glavieux, "Analyse de convergence en turbo détection." in *Proceedings GRETSI2001*, Toulouse, France, Sept. 2001.

[107] A. Roumy, A. J. Grant, I. Fijalkow, P. D. Alexander, and D. Pirez., "Turbo-equalization: Convergence analysis." in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 4, Salt Lake City, May 2001, pp. 2654–2648.

[108] P. A. Regalia, "Blind turbo equalization using the constant modulus algorithm," in *13th IFAC Symp. System Identification (SYSID 2003)*, Rotterdam, The Netherlands, August 2003, pp. 584–589.

[109] N. Noels, H. Steendam, and M. Moeneclaey, "The Cramer-Rao bound for phase estimation from coded linearly modulated signals," *IEEE Commun. Lett.*, vol. 7, pp. 207–209, May 2003.

[110] ——, "On the Cramer-Rao lower Bound and the performance of synchronizers for (turbo) encoded systems," in *IEEE International Workshop on Signal Processing Advances for Wireless Commun.*, July 2004, pp. 1–5.

[111] ——, "Carrier and Clock recovery in (turbo) coded systems: Cramer-Rao Bound and Synchronizer Performance," *EURASIP Journal on Applied Signal Processing*, no. 6, pp. 972–980.

[112] B. Mielczarek and A. Svensson, "Post-decoding timing synchronization of turbo codes on awgn channels," in *Proceedings VTC 2000*, vol. 2, May 2000, pp. 1265–1270.

[113] ——, "Joint synchronization and decoding of turbo codes on awgn channels," in *Proceedings VTC 1999*, vol. 3, May 1999, pp. 1886–1890.

[114] L. Lu and S. G. Wilson, "Synchronization of turbo coded modulation systems at low snr," in *IEEE International Conference on Communications*, vol. 1, June 1998, pp. 428–432.

[115] B. Mielczarek and A. Svensson, "Improved map decoders for turbo codes with non-perfect timing and phase synchronization," in *Proceedings VTC 1999*, vol. 3, Sept. 1999, pp. 1590–1594.

[116] J. Sun and M. Valenti, "Synchronization of turbo codes based on online statistics," in *IEEE International Conference on Communications*, vol. 4, June 2003, pp. 2733–2737.

[117] J. M. Walsh, P. A. Regalia, and C. R. Johnson, Jr., "Joint synchronization and decoding exploiting the turbo principle," in *Conference on Information Sciences and Systems*, Mar. 2004.

[118] T. Cover and J. Thomas, *Elements of Information Theory.* John Wiley and Sons, 1991.

[119] C. J. Wu, "On the Convergence Properties of the EM Algorithm," *The Annals of Statistics*, vol. 11, pp. 95–103, 1984.

[120] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm," *Journal Royal Statistical Society, Serial B*, pp. 1–38, 1977.

[121] R. E. Blahut, "Computation of channel capacity and rate-distortion functions," *IEEE Trans. Inform. Theory*, vol. 18, pp. 460–473, 1972.

[122] S. Arimoto, "An algorithm for computing the capacity of arbitrary discrete memoryless channels," *IEEE Trans. Inform. Theory*, vol. 18, pp. 14–20, 1972.

[123] J. M. Walsh, P. A. Regalia, and C. R. Johnson, Jr., "Turbo decoding is iterative constrained maximum likelihood sequence detection," submitted to *IEEE Trans. Inform. Theory.*

[124] ——, "Belief propagation as iterative constrained network-wide maximum likelihood detection," submitted to *IEEE Trans. Inform. Theory*.

[125] J. M. Walsh and P. A. Regalia, "Connecting belief propagation with maximum likelihood detection," Apr. 2006, to appear at *Fourth International Symposium on Turbo Codes*.

[126] ——, "Iterative constrained maximum likelihood estimation via expectation propagation," May 2006, to appear at *IEEE International Conference on Acoustics, Speech, and Signal Processing*.

[127] A. N. Kolmogorov and S. V. Fomin, *Introductory Real Analysis*. Dover, 1975, english Translation by Richard A. Silverman.

[128] A. Friedman, *Foundations of Modern Analysis*. Dover, 1982.

[129] R. S. Strichartz, *The Way of Analysis*. Jones & Bartlett Publishers, 2000.

[130] A. Greenbaum, *Iterative Methods for Solving Linear Systems*. Society for Industrial and Applied Mathematics, 1997.

[131] L. Gripp and M. Sciandrone, "On the Convergence of the block nonlinear Gauss-Seidel method under convex constraints," *Operations Research Letters*, vol. 26, pp. 127–136, 2000.

[132] P. Moqvist and T. M. Aulin, "Turbo-decoding as a numerical analysis problem," in *Proc. IEEE Internation Symposium on Information Theory, 2000.*, June 2000, p. 485.

[133] D. G. Luenberger, *Linear and Nonlinear Programming: 2nd Edition*. Addison-Wesley, 1984.

[134] T. K. Moon and W. C. Stirling, *Mathematical Methods and Algorithms for Signal Processing*. Prentice Hall, 2000.

[135] H. Bauschke and P. Combettes, "Construction of best Bregman approximations in reflexive Banach spaces," *Proceedings of the American Mathematical Society*, no. 131(12), pp. 3757–3766, 2003.

[136] ——, "Iterating Bregman retractions," *SIAM Journal on Optimization*, no. 13(4), pp. 1159–1173, 2003.

[137] H. Bauschke, "Duality for Bregman projections onto translated cones and affine subspaces," *Journal of Approximation Theory*, no. 121, pp. 1–12, 2003.

[138] H. Bauschke, F. Deutsch, H. Hundal, and S.-H. Park, "Accelerating the convergence of the method of alternating projections," *Transactions of the American Mathematical Society*, no. 355(9), pp. 3433–3461, 2003.

[139] H. Bauschke, J. Borwein, and P. Combettes, "Bregman monotone optimization algorithms," *SIAM Journal on Control and Optimization*, no. 42(2), pp. 596–636, 2003.

[140] H. Bauschke, P. Combettes, and D. Luke, "Finding best approximation pairs relative to two closed convex sets in Hilbert spaces," *Journal of Approximation Theory*, no. 127, pp. 178–192, 2004.

[141] C. Byrne and Y. Censor, "Proximity function minimization using multiple Bregman projections, with applications to split feasibility and Kullback-Leibler distance minimization," *Annals of Operations Research*, vol. 105, pp. 77–98, 2001.

[142] W. Byrne, "Alternating Minimization and Boltzmann Machine Learning," *IEEE Trans. Neural Networks*, vol. 3, no. 4, July 1992.

[143] Y. Censor and S. Reich, "The Dykstra algorithm with Bregman projections," *Communications in Applied Analysis*, vol. 2, pp. 407–419, 1998.

[144] P. P. B. Eggermont, "Multiplicative iterative algorithms for convex programming," *Linear Algebra and its Applications*, no. 130, pp. 25–42, 1990.

[145] A. N. Iusem, "Convergence analysis for a multiplicatively relaxed em algorithm," *Mathematical Methods in the Applied Sciences*, vol. 14, pp. 573–593, 1991.

[146] W. E. Winkler, "On dykstra's iterative fitting procedure," *The Annals of Probability*, vol. 18, no. 3, pp. 1410–1415, July 1990.

[147] S. Kullback, *Information Theory and Statistics*. Dover, 1968.

[148] H. V. Poor, *An Introduction to Signal Detection and Estimation*, 2nd ed. Springer, 1994.

[149] D. J. C. MacKay, *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 2003.

[150] M. J. Wainwright, T. S. Jaakkola, and A. S. Willsky, "Tree-based reparameterization framework for analysis of sum-product and related algorithms," *IEEE Trans. Inform. Theory*, vol. 49, no. 5, pp. 1120–1146, May 2003.

[151] R. Vicente, D. Saad, and Y. Kabashima, "Statistical physics of irregular low-density parity-check codes," *J. Phys. A. Math. Gen*, no. 33, pp. 6527–6542, 2000.

[152] J. S. Yedidia, W. T. Freeman, and Y. Weiss, "Understanding belief propagation and its generalizations," Mitsubishi Electric Research Laboratory, Tech. Rep., Jan. 2002, mERL TR-2001-22.

[153] S. C. Tatikonda, "Convergence of the sum-product algorithm," in *Proceedings ITW2003*, Apr. 2003.

[154] T. Heskes, "On the uniqueness of loopy belief propagation fixed points," *Neural Computation*, no. 16, pp. 2379–2413, 2004.

[155] T. P. Minka, "The ep energy function and minimization schemes," Aug. 2001, unpublished Technical Report. [Online]. Available: http://research.microsoft.com/~minka/papers/

[156] M. J. Wainwright, T. S. Jaakkola, and A. S. Willsky, "A new class of upper bounds on the log partition function," *IEEE Trans. Inform. Theory*, vol. 51, no. 7, pp. 2313–2335, July 2005.

[157] ——, "Map estimation via agreement on trees: Message-passing and linear programming," *IEEE Trans. Inform. Theory*, vol. 51, no. 11, pp. 3697–3717, Nov. 2005.

[158] T. Tanaka and M. Okada, "Approximate Belief Propagation, Density Evolution, and Statistical Neurodynamics for CDMA Multiuser Detection," *IEEE Trans. Inform. Theory*, vol. 51, no. 2, Feb. 2005.

[159] Y. Weiss and W. T. Freeman, "Correctness of Belief Propagation in Gaussian Graphical Models of Arbitrary Topology," *Neural Computation*, no. 13, pp. 2173–2200, 2003.

[160] Y. Weiss, "Correctness of Local Probability Propagation in Graphical Models with Loops," *Neural Computation*, no. 12, pp. 1–41, 2000.

[161] A. Ashikhmin, G. Kramer, and S. ten Brink, "Extrinsic information transfer functions: Model and erasure channel properties," *IEEE Trans. Inform. Theory*, vol. 50, no. 11, pp. 2657–2673, Nov. 2004.

[162] M. Fu, "Stochastic analysis of turbo decoding," *IEEE Trans. Inform. Theory*, vol. 41, pp. 81–100, Jan. 2005.

[163] M. Lentmaier, D. V. Truchachev, K. S. Zigangirov, and D. J. Costello, "An analysis of the block error probability performance of iterative decoding," *IEEE Trans. Inform. Theory*, vol. 51, no. 11, pp. 3834–3855, Nov. 2005.

[164] Brännström and L. K. Rassmussen and A. J. Grant, "Convergence analysis and optimal scheduling for multiple concatenated codes," *IEEE Trans. Inform. Theory*, vol. 51, no. 9, pp. 3354–3364, Sept. 2005.

[165] N. Miladinovic and M. P. C. Fossorier, "Improved bit-flipping decoding of low-density parity-check codes," *IEEE Trans. Inform. Theory*, vol. 51, no. 4, pp. 1594–1606, Apr. 2005.