Low-Complexity Message-Passing Algorithms for Distributed Computation

Nima Noorshams

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2013-53
http://www.eecs.berkeley.edu/Pubs/TechRpts/2013/EECS-2013-53.html

May 10, 2013

Low-Complexity Message-Passing Algorithms for Distributed Computation

by

Nima Noorshams

A dissertation submitted in partial satisfaction of the
requirements for the degree of
Doctor of Philosophy

in

Engineering-Electrical Engineering & Computer Sciences

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Martin J. Wainwright, Chair
Professor Kannan Ramchandran
Professor David Aldous

Spring 2013

Low-Complexity Message-Passing Algorithms for Distributed Computation

Abstract

Low-Complexity Message-Passing Algorithms for Distributed Computation

by

Nima Noorshams

Doctor of Philosophy in Engineering-Electrical Engineering & Computer Sciences

University of California, Berkeley

Professor Martin J. Wainwright, Chair

Central to many statistical inference problems is the computation of some quantities defined over variables that can be fruitfully modeled in terms of graphs. Examples of such quantities include marginal distributions over graphical models and empirical average of observations over sensor networks. For practical purposes, distributed message-passing algorithms are well suited to deal with such problems. In particular, the computation is broken down into pieces and distributed among different nodes. Following some local computations, the intermediate results are shared among neighboring nodes via the so called messages. The process is repeated until the desired quantity is obtained. These distributed inference algorithms have two primary aspects: statistical properties, in which characterize how mathematically sound an algorithm is, and computational complexity that describes the efficiency of a particular algorithm. In this thesis, we propose low-complexity (efficient), message-passing algorithms as alternatives to some well known inference problems while providing rigorous mathematical analysis of their performances. These problems include the computation of the marginal distribution via belief propagation for discrete as well as continuous random variables, and the computation of the average of distributed observations in a noisy sensor network via gossip-type algorithms.

To my mother for her unconditional love and support, to my father for his keen interest in knowledge and education, and to my brother for being my best friend.

Contents

List of Figures

Acknowledgments

I am grateful to have the opportunity to thank those who have provided me help and support throughout my time in graduate school at Berkeley. First and for most, I should acknowledge my adviser Prof. Martin J. Wainwright. Working with Martin has been a tremendous opportunity for me. During my time at Berkeley wireless foundation lab, I learned a great deal from him, not only technical matters but also other aspects of research including technical writing and presentation skills. From our whiteboard discussions that lead to some of the ideas presented in this thesis to writing, Martin's role has been indispensable and I am grateful to him. I would like to thank Prof. Kannan Ramchandran, the chair of my qualification exam committee, for giving me constructive comments and helpful advice regarding research and work. I also would like to thank other members of my qual. committee, Prof. David Aldous, and Prof. Venkat Anantharam. Before joining Martin's group at the end of my second year, I had a chance to work with Prof. Ahmad Bahai on cognitive radio wireless communication and learned a lot from him.

In addition to faculty members at Berkeley, i have enjoyed occasional collaboration with others. I would like to thank Alekh Agarwal for helpful discussions on stochastic optimization techniques, Dr. Pascal Vontobel, the associate editor of the IEEE IT transaction, for his very careful reading of our SBP paper (Chapter 3), and Prof. Erik Sudderth for helpful comments on experimental results concerning the SOSMP work (Chapter 4). I also should thank my undergraduate mentors from Sharif University of Technology, Prof. Massoud Babaie-Zadeh and Prof. Bagher Shamsollahi. Finally, I would like to thank the members of the wireless foundation lab and Martin's research group, my friends, Sahand Negahban, John Duchi, Arash Amini, Garvesh Raskutti, Po-Ling Loh, Sameer Pawar, Venkatesan Ekambaram, Naveen Goela, Nebojsa Milosavljevic, and Amin Gohari for sitting through my practice presentations and giving me helpful and constructive feedback.

Chapter 1

Introduction

Many practical systems are affected by random variations such as observational error, communication noise, model uncertainty, and so on. Examples of such systems can be found in different fields including telecommunications, signal and image processing, computer vision, machine learning, finance, bioinformatics, among others. The purpose of *statistical inference* is to draw conclusions based on data arising from such systems. To characterize their behavior, the stochastic systems are first modeled, which means the relationship between random variables are formalized (typically via a set of parameters). Then, based on some realizations of the data, an estimate of the parameters that best describe the system are determined. Inference problems have two fundamental aspects: statistical properties that characterize the behavior of an algorithm (such as consistency, rate of convergence, etc.) as well as computational complexity that describes the efficiency of a particular algorithm. In this dissertation, our primary focus will be on the latter. We will provide efficient, low-complexity solutions to some algorithms with wide range of applications, while analyzing their statistical behavior.

To describe these concepts more clearly, we consider a concrete example in telecommunication and signal processing. Suppose we want to find the location of a moving access terminal (such as a smart phone) in an indoor environment where GPS fails [79, 91, 98, 54]. One approach would be to exploit the signal strength received from access points (WiFi routers). In that case, the problem can be modeled as a hidden Markov chain [21], where the location of the access terminal is the latent variable, evolving according to a Markov chain, and at each location the received signal strengths is the observable variable. Given the movement model and the communication channels, we can formalize the relationship between the latent random variables (i.e. access terminal locations along the route) via a joint distribution. In order to estimate a particular location, we need to compute the marginal distribution, from which maximization or averaging yields the MAP or Bayesian estimates respectively. Suppose we have n latent variables and also the state space (floor plane) is discretized so that there exists d possible outcomes for each variable. A naive approach to solve this problem, that is summing over all variables but one, requires nd^{n-1} operations.

This solution is exact and it always works; however from complexity point of view it is not feasible. It requires exponentially growing number of computations that can be prohibitive even for a moderate-size problem. The sum-product algorithm has been proposed [4, 119] for solving the marginalization problem. It is an iterative message-passing algorithm, where at each iteration an estimate of the marginals are computed. Moreover, it can be shown that the estimates on a hidden Markov model converge to the exact solution in a finite number of iterations, so that the computational complexity is much lower than that of the naive approach. More precisely, one requires of the order of nd^2 computations in order to calculate all the marginals. In Chapter 3, we show that the dependency of the computational complexity on the state dimension d can be further reduced to linear (as opposed to quadratic) while preserving the favorable statistical properties.

Central to the inference problems is the computation of some statistical quantities. Examples of such quantities include the marginal distributions of a joint distribution (in the case of the previous example), likelihoods, regression functions, averages over a collection of random variables obtained from a sensor network. Typically, the computation involves variables that can be well modeled with a *graph* [30, 16]. A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ consists of a set of nodes $\mathcal{V}$ and a set of edges $\mathcal{E}$ and provides an abstract representation of a set of objects some of which are connected. In more precise terms, every object is associated to a unique node or vertex $i \in \mathcal{V}$; moreover, two vertices i, and j are connected by an edge or link if and only if the pair (i, j) belongs to the set $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$. The links represent statistical dependencies among random variables or communication links among physically separated sensors when nodes represent random variables or sensors respectively.

Due to their nature, distributed computation is well suited for problems on graphs. The existence of the required infrastructure for the centralized computation is not necessarily guaranteed in the sensor network application. Also, as will become clear in the following chapter, a form of the divide and conquer algorithm on graphs can significantly increases the efficiency of the marginalization algorithm. The general idea behind distributed computation is to break down the calculation and distribute the pieces among different nodes. Then following some local computations, every node shares its information with other nodes by passing the so called *messages* to its neighbors along the edges of the graph. The received messages constitute the intermediate results and could be of the form of a d-dimensional vector, a real-valued function, or a noise-corrupted signal. Every node, uses the received messages in order to update an estimate of the desired quantity. Of interested to us are the statistical properties of these estimates, as well as the efficiency of the local computations. But first we need to formally introduce two popular mathematical models for distributed computations, graphical models and sensor networks.

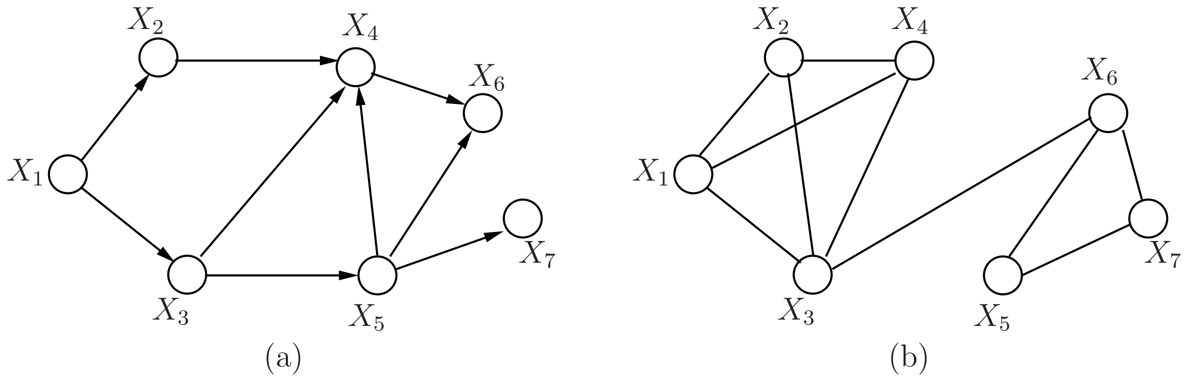


Figure 1.1: Examples of graphical models: (a) a Bayesian network, and (b) a Markov random field. There is a one-to-one mapping between random variables and nodes of the graph. Moreover, in a Baysian netowrk edges are directed (ordered pairs of nodes), whereas in a Markov ranodm field edges are undirected (unordered pairs of nodes).

1.1 Graphical Models

By bringing together graph theory and probability theory, graphical models provide a general and flexible framework for describing statistical interactions among random variables [92, 61, 119, 90, 118, 121, 44, 69, 102, 4, 62]. A broad range of fields—among them statistical signal processing, computer vision, coding theory, bioinformatics, natural language processing—involve problems with large number of random variables that can be fruitfully formulated in terms of graphs. A few of such eamples include, positioning and tracking problems that can be modeled by chains [79, 54, 35], low-density parity-check codes that can be described by factor graphs [99], some image processing as well as vision problems that can be formulated by two dimensional grids [112, 60, 15], and text processing that can be modeled by Bayesian networks [107].

The statistical dependencies among random variables are encoded by the structure of the graph. This is accomplished by first mapping the random variables to the nodes of the graph and then factorizing the joint probability distribution over local functions defined on the graph. There are two common families of graphical models, directed models also known as Bayesian networks and undirected models also known as Markov random fields. See Figure 1.1 for an illustration of these two models. Bayesian networks are defined on directed graphs that is every edge consists of an *ordered* pair of nodes. It can be shown that if the directed graphical model is acyclic (i.e. does not include any directed cycles), the joint probability density becomes equal to the product of a collection of local conditional probability densities [61]. In contrast to Bayesian networks, Markov random fields are defined on undirected graphs in which each edge is an *unordered* pair of nodes. As we will see in the next chapter, probability densities over such graphs also have local factorization, in particular over positive functions defined on the fully connected sub-graphs. Even though these two models are closely related, they make different assertions of conditional independence between random variables. Therefore, depending on the application one might be a better option than the other. For instance, a Markov random field could a better choice for problems that involve random variables with no clear causal relationships.

Generally in a graphical model application, we are interested in a conditional distribution of a set of random variables given another set of random variables. A popular special case of this problem is the *marginalization*, meaning the computation of the marginal distributions from the joint distribution by summing over all the variables but one. The marginal distributions can be used in a wide range of applications some of which include estimating the location of a moving target in the positioning application, or estimating the transmitted signal given the received noisy version in the error-correction application, or detecting an object's movement in a series of video frames in the optical flow application. There are different approaches to solve these problems efficiently, most notably sampling techniques [44, 69, 102] and variational algorithms [61, 119, 90, 118, 121].

There are several different sampling methods (such as importance sampling, Gibbs sampling, Metropolis Hastings, etc.) all of which attempt to draw samples from the joint probability distribution, defined on a graph. Typically, these techniques are a variation of Markov chain Monte Carlo [69, 102]. They are based on constructing a Markov chain that has the desired distribution as its stationary distribution and work as follows: the Markov chain is first initialized arbitrarily and then updated with random jumps. The state of the chain after many iterations are used as a sample of the desired distribution. This process is repeated until enough samples are gathered. Having drawn several joint samples, discarding the irrelevant components yields an estimate of the marginal distributions.

Parallel to the sampling techniques are the variational algorithms. The basic idea behind variational methods is to characterize the probability distribution as the solution to an optimization problem and solve the optimization problem (or its relaxed version) instead. The sum-product algorithm is an attempt to solve the Bethe free energy optimization [121]. Limiting the optimization to the so called "tractable" distributions leads to the mean field algorithm [90]. Convex relaxation of the optimization space yields the tree-reweighted belief propagation algorithm [77, 118]. In this thesis our primary focus is on the sum-product algorithm also known as belief propagation (BP). Belief propagation is an iterative algorithm consisting of a set of local message-passing rounds that provides a fast and efficient mechanism for computing either exact or approximate marginal distributions [92, 119, 61, 4]. As it turns out BP is a form of divide and conquer algorithm that exploits the particular form of the factorization induced by a graph. The fundamental idea behind the BP is nothing but the simple distributive law and can be easily applied to other semi-rings such as max-product, min-sum, etc. suitable for computing the MAP estimator [4].

Even though BP provides and efficient algorithm for the marginalization problem, its computational and communication costs could be prohibitive for large-scale problems. Therefore, in the first part of the thesis, we focus on ways to reduce these complexities. In doing so we will propose lower-complexity alternatives to the BP algorithm for both discrete and continuous-valued random variables and provide rigorous mathematical analysis of their behavior. In addition, we will confirm the mathematical guarantees by some experimental results.

1.2 Sensor Networks

A sensor network is a collection of specially separated autonomous sensors with limited memory, communication, computation and energy resources. Telecommunications and environment monitoring under harsh conditions such as forests fire detection, air quality, ocean level, and glacier temperature monitoring are among the primary applications of the sensor networks [120, 117, 122, 67]. Sensors are small, cheap, and easy to implement; therefore, a large number of them are typically scattered in an environment to be monitored. Lack of the necessary infrastructure, massive databases, and insufficient memory at each mote all preclude storing all the data at a central location. Therefore, sensor networks must be accompanied with distributed computation techniques. On the other hand, sensors are prone to failure and have a short life span, which makes designing efficient and robust algorithms even more essential for such systems.

Graphs provide a natural framework for modeling and understanding problems arising in sensor networks. More specifically, every sensor is associated to a node in a graph and two nodes are connected if and only if there is a communication link (normally a two-way channel) between them. A common model for wireless sensor networks is the random geometric graph (RGG) [94], where nodes are assumed to be distributed uniformly at random inside the unite square and two nodes are connected if and only if their euclidean distance is bellow some threshold. Moreover, there are several different models for communication channels. A simple model, used by various researchers [28, 116, 18, 31, 12, 11], in order to study sensor networks, is the noiseless model in which the transmitted signal by a sensor is received by its neighbors unaltered. A somewhat more realistic model is to consider the effect of noise, for instance additive white Gaussian noise (AWGN), packet dropping, quantization noise, etc. [95, 45, 7, 56]. It is also conceivable to consider more complicated and realistic stochastic models such as flat fading or frequency selective for wireless communication channels [115].

In a typical application of sensor networks, we are interested in reaching a global decision, based on local information sharing among individual sensors. Each mote, following some simple processing, transmits its information to its neighbors until they all reach a global consensus. A class of such problems, of interest to us in this thesis, is the *network-constrained averaging* [18, 31, 12, 95, 37]. In more details, we are interested in computing the average of a set of numbers distributed throughout a network, using an algorithm that is allowed to pass messages only along edges of the graph. In applications, the average might represent a statistical estimate of some physical quantity (e.g. temperature, pressure, etc.), or an intermediate quantity in a more complex algorithm (e.g. log-likelihood for estimation or gradient for distributed optimization).

A major bulk of the work in the literature focuses on the case of noise-less averaging [18, 31, 57, 8, 11, 22]. Moreover, in the noisy case the current algorithms are not optimal [95, 45, 37, 56, 9]. In the last part of the thesis, we propose an order optimal algorithm for the problem of network-constrained averaging with AWGN channels.

1.3 Contributions and Dissertation Overview

The reminder of the dissertation is organized as follows. We will begin by an overview of some necessary background, including materials on undirected graphical models, the sum-product or belief propagation algorithm, stochastic approximation techniques and gossip type algorithms in Chapter 2. In Chapter 3, we focus on pairwise Markov random fields with discrete random values and propose the stochastic belief propagation (SBP), a low-complexity alternative to the belief propagation algorithm. In more specific terms, for pairwise Markov random fields over d-state discrete random variables, the SBP algorithm reduces the per-iteration computational complexity to order d as opposed to d^2 for the case of BP. In addition, the SBP algorithm reduces the per-iteration communication complexity to $\log d$ bits in contrast to order d bits for the usual BP message updates. We also provide numerous mathematical guarantees for SBP including consistency, rate of convergence, and bounds with high probability for tree-structured as well as general graphical models. Moreover, we provide various experimental results to support theoretical guarantees. Chapter 4 is devoted to a similar problem but one involving continuous-valued random variables. We propose stochastic orthogonal series message-passing (SOSMP), an efficient message-passing algorithm for continuous state spaces. We also analyze SOSMP by providing rigorous mathematical guarantees for general graphical models (including almost sure convergence and rate of convergence), and characterizing the complexity-accuracy trade-off for the models involving positive semidefinite kernels. In Chapter 5, we turn to a somewhat different problem. We propose an order optimal algorithm for the network-constrained averaging with AWGN channels. We prove this claim by providing non-asymptotic bounds on the mean-squared error for several types of networks including regular cycles and two-dimensional grids as well as random geometric graphs. Some of the more technical aspects of each chapter will be deferred to Appendices A, B, and C.

Chapter 2

Background

In this chapter, we will review some of the mathematical concepts that will be employed throughout this thesis. These will include graphical models, the belief propagation algorithm, stochastic approximation, and gossip algorithms. We begin by reviewing the basic concepts of undirected graphical models as well as the belief propagation algorithm in Sections 2.1 and 2.2 respectively. Since stochastic approximation and optimization techniques are a crucial part of this work, we turn to this subject in Section 2.3. Finally, we provide some basics on gossip algorithms in Section 2.4.

2.1 Undirected Graphical Models

Consider a random vector $X := \{X_1, X_2, \ldots, X_n\}$, where for each $v = 1, 2, \ldots, n$, the variable X_v takes values[1] in some discrete or continuous space $\mathcal{X}$.[2] We assign these random variables to the vertex set of an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, indexed by $\mathcal{V} := \{1, 2, \ldots, n\}$. Although they are certainly different, we sometime ignore the distinction and refer to node v as random variable X_v and vice versa. In addition to the vertex set, the graph $\mathcal{G}$ consists of a collection of edges $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$, where an unordered pair $(u, v) \in \mathcal{E}$ if and only if nodes u and v are connected by an edge. Also self-edges are forbidden, meaning that $(v, v) \notin \mathcal{E}$ for all $v \in \mathcal{V}$.

An undirected graphical model, also known as a Markov random field, defines a family of joint probability distributions over the random vector X. These probability distributions are assumed to be absolutely continuous with respect to a given measure μ, typically the counting measure for the case of discrete random variables or the Lebesgue measure for continuous random variables. The structure of the graph describes the statistical dependencies among the different random variables—in particular via the notion of *graph separation*. For a set

[1] In this thesis, random variables are represented with capital letters such as $X, Y, \ldots$, whereas their realizations are represented with lowercase letters such as $x, y, \ldots$.

[2] Although our theory allows for distinct state spaces at each node, to streamline our presentation, we drop the node dependence and assume $\mathcal{X}_v = \mathcal{X}$ for all $v \in \mathcal{V}$.

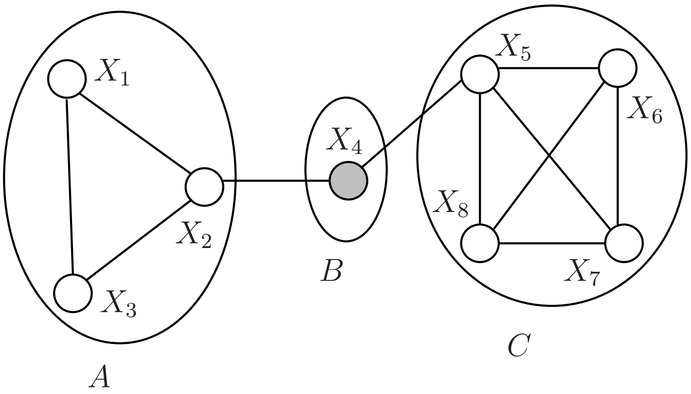


Figure 2.1: The notion of graph separation. Set B separates sets A and C. More precisely, every path from a node in set A to a node in set C goes through a node in set B. Accordingly, the set of random variables $\{X_1, X_2, X_3\}$ is independent of the set $\{X_5, X_6, X_7, X_8\}$ given X_4.

A, define the sub-vector $X_A := \{X_v | v \in A\}$, similarly defined for sets B and C. We say that

$$\text{“the random vector } X_A \text{ is independent of } X_C \text{ given } X_B\text{”} \tag{2.1}$$

if and only if set B separates sets A and C. More specifically, every path from a node in set A to a node in set C goes through the set B. See Figure 2.1 for an illustration of the graph separation notion. Every graph $\mathcal{G}$, induces a set of such conditional independence statements also known as Markov properties. Moreover, it should be noted that such Markov properties must hold for all members of the family of the probability distributions associated with $\mathcal{G}$. Therefore, the family of acceptable probability distributions is constrained by the set of all Markov properties and thus have a particular form of factorization. In order to make this statement precise, we need to define the notion of *cliques*. A clique I of a graph is a subset of vertices that are all joined by edges, and so form a fully connected sub-graph. For instance in Figure 2.1, the set of variables $\{X_1, X_2, X_3\}$ and $\{X_5, X_6, X_7, X_8\}$ form cliques of size three and four respectively. The close connection of the Markov properties induced by the graph (via graph separation) and cliques are captured by the next theorem. Indeed it can be shown that a distribution defined on $\mathcal{G}$ respects all such conditional independence statements if and only if it can be factorized over the cliques of $\mathcal{G}$ [40, 14].

Theorem 1 (Hammersley-Clifford). *Let $\mathcal{G}$ be an undirected graphical model with a set of cliques $\mathcal{C}$. Suppose that the probability density $\mathbb{P}$ over a discrete random vector X is factorized over the cliques of $\mathcal{G}$*

$$\mathbb{P}(x_1, x_2, \ldots, x_n) \propto \prod_{I \in \mathcal{C}} \psi_I(x_I), \tag{2.2}$$

where $\psi_I : \mathcal{X}^{|I|} \to [0, \infty)$ is the compatibility function. Then the underlying process respects all the Markov properties (2.1) induced by the graph $\mathcal{G}$. Conversely if $\mathbb{P}(x)$ is positive for all

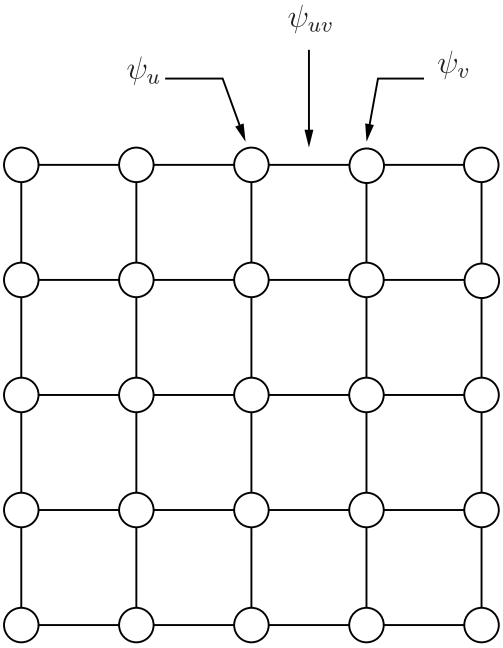


Figure 2.2: A two-dimensional grid, an example of pairwise Markov random fields. The potential functions ψ_u and ψ_v are associated with nodes u and v, respectively, whereas the potential function ψ_{uv} is associated with edge (u, v).

$x \in \mathcal{X}^n$, and respects all the Markov properties induced by $\mathcal{G}$, then it has a factorization of the form (2.2)*.*

2.1.1 Pairwise Markov Random Fields

Many applications involve pairwise interactions among nodes. In those instances, since cliques consist of the set of all vertices $\mathcal{V}$ together with the set of all edges $\mathcal{E}$, the general factorization (2.2) takes the special form

$$\mathbb{P}(x_1, x_2, \ldots, x_n) \propto \prod_{u \in \mathcal{V}} \psi_u(x_u) \prod_{(u,v) \in \mathcal{E}} \psi_{uv}(x_u, x_v), \tag{2.3}$$

where $\psi_u : \mathcal{X} \to (0, \infty)$ is the node potential function for node u, and $\psi_{uv} : \mathcal{X} \times \mathcal{X} \to (0, \infty)$ is the edge potential function for the edge (u, v). A factorization of this form (2.3) is known as a *pairwise Markov random field*. As a concrete example, consider the two-dimensional grid shown in Figure 2.2, used in image processing and computer vision applications. It is important to note that for discrete random variables, there is no loss of generality in assuming a pairwise factorization of this form; indeed, any graphical model with discrete random variables can be converted into a pairwise form by suitably augmenting the state space (e.g., see Yedidia et al. [121] or Wainwright and Jordan [119], Appendix E.3). For the remainder of this thesis, we focus on the case of a pairwise Markov random fields.

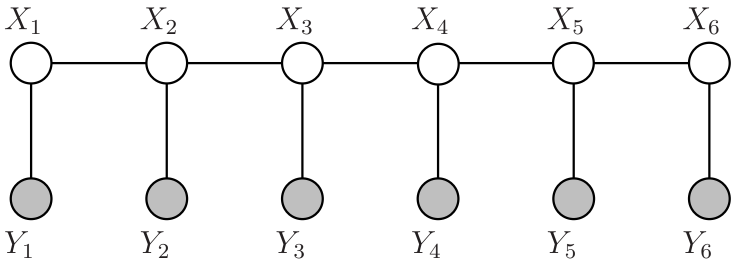


Figure 2.3: A hidden Markov model including both hidden variables $(X_1, X_2, X_3, X_4, X_5, X_6)$, represented as white nodes, and observed variables $(Y_1, Y_2, Y_3, Y_4, Y_5, Y_6)$, represented as shaded nodes.

2.1.2 Inference via Marginalization

In various application contexts, the random vector $(X_1, X_2, \ldots, X_n)$ is an unobserved or "hidden" quantity, and the goal is to draw inferences on the basis of a collection of observations $(Y_1, Y_2, \ldots, Y_n)$. (See Figure 2.3 for illustration.) The link between the observed and hidden variables is specified in terms of a conditional probability distribution, which in many cases can be written in the product form $\mathbb{P}(y \mid x) = \prod_{v=1}^{n} \mathbb{P}(y_v \mid x_v)$. For instance, in error-control coding using a low-density parity-check code, the vector X takes values in a linear subspace of $GF(2)^n$, corresponding to valid codewords, and the observation vector Y is obtained from some form of memoryless channel (e.g., binary symmetric, additive white Gaussian noise, etc.). In image denoising applications, the vector X represents a rasterized form of the image, and the observation Y corresponds to a corrupted form of the image. In terms of drawing conclusions about the hidden variables based on the observations, the central object is the posterior distribution $\mathbb{P}(x \mid y)$. From the definition of the conditional probability and the form of the prior and likelihoods, this posterior can also be factorized in pairwise form

$$\begin{aligned} \mathbb{P}(x \mid y) &\propto \mathbb{P}(x_1, x_2, \ldots, x_n) \prod_{v=1}^{n} \mathbb{P}(y_v \mid x_v) \\ &= \prod_{v \in \mathcal{V}} \widetilde{\psi}_v(x_v) \prod_{(u,v) \in \mathcal{E}} \psi_{uv}(x_u, x_v), \end{aligned} \tag{2.4}$$

where $\widetilde{\psi}_v(x_v) := \psi_v(x_v)\mathbb{P}(y_v \mid x_v)$ is the new node compatibility function. (Since the observation y_v is fixed, there is no need to track its functional dependence.) Thus, the problem of inferring data from a posterior distribution can be cast[3] as an instance of a pairwise Markov random field (2.3).

[3] For illustrative purposes, we have assumed here that the distribution $\mathbb{P}(y \mid x)$ has a product form, but a somewhat more involved reduction also applies to a general observation model.

A computational challenge central to the data inference problem is the *marginalization problem*, meaning the computation of the single-node marginal distributions

$$\mathbb{P}(x_v) \quad := \underbrace{\int_{\mathcal{X}} \cdots \int_{\mathcal{X}}}_{(n-1) \text{ times}} \mathbb{P}(x_1, x_2, \ldots, x_n) \prod_{u \in \mathcal{V} \backslash \{v\}} \mu(dx_u), \tag{2.5}$$

for each $v \in \mathcal{V}$ and more generally, higher-order marginal distributions on edges and cliques. For instance in the error-control coding application, computing the marginal distribution of a particular bit given the output of a channel, we can detect the most likely value of the bit that was transmitted. Similarly, in the image denoising problem, the marginal distribution of a pixel's value given the corrupted image yields the most likely value of that pixel. Naively approached, this problem suffers from the curse of dimensionality, since it requires computing a multi-dimensional integral (for continuous random variables) or summation (for discrete random variables) over an $(n-1)$-dimensional space. To clarify a bit, for the case of discrete random variables, where μ is the counting measure, we have

$$\mathbb{P}(x_v) \quad := \sum_{\{x' \mid x'_v = x_v\}} \mathbb{P}\left(x'_1, x'_2, \ldots, x'_n\right). \tag{2.6}$$

To calculate this summation, brute force is not tractable and requires d^{n-1} computations. For any graph without cycles—known as a tree—this computation can be carried far more efficiently using an algorithm known as the belief propagation, to which we now turn.

2.2 Belief Propagation Algorithm

Belief propagation is an iterative algorithm consisting of a set of local message-passing rounds, for computing either exact or approximate marginal distributions defined on a graph. As discussed in the previous section, if approached naively, computing marginal distributions is intractable. However, exploiting the particular form of the factorization induced by the graph, BP provides a fast and efficient method for dealing with this problem. In this section, we will provide an overview of the BP algorithm over the sum-product semi-ring.

In order to formally introduce the message-passing updates, we first need to define the notion of the factor graph. A factor graph is a graphical representation of factorizations (2.2). In precise terms, it is a bipartite graph $\mathcal{G}' := (\mathcal{V}_1 \cup \mathcal{V}_2, \mathcal{E}')$, consisting of variable nodes indexed by $\mathcal{V}_1 := \{1, 2, \ldots, n\}$ and factor nodes $\mathcal{V}_2 := \{I | I \in \mathcal{C}\}$. Moreover, an edge connects the variable node i to the factor I (i.e. $(i, I) \in \mathcal{E}'$) if and only if x_i belongs to the local function $\psi_I(\cdot)$. Figure 2.4 illustrates the factor graph representation of the graphical model depicted in Figure 2.1. To provide some intuition regarding the message-passing algorithm, we proceed with a simple example.

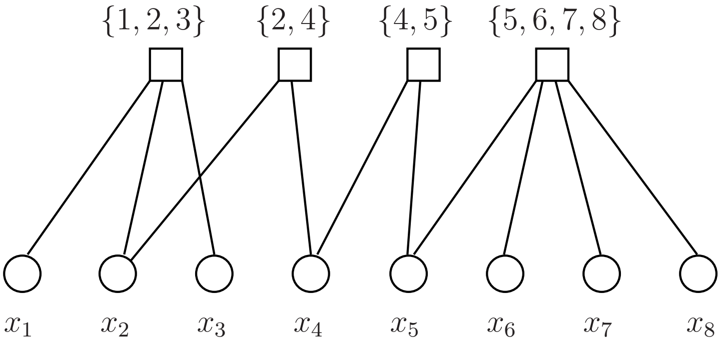


Figure 2.4: Factor graph representation of the graphical model in Figure 2.1. The bipartite graph has a node corresponding to each variable (circular nodes) and a node corresponding to each factor (square nodes). The variable node x_i is connected to the factor node I if and only if x_i belongs to the factor I.

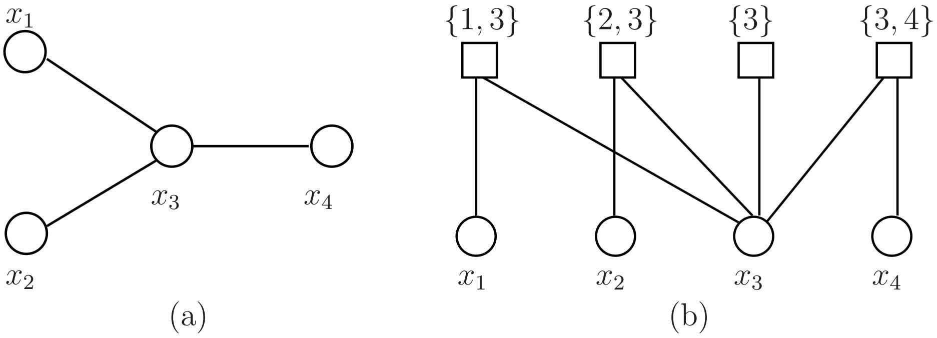


Figure 2.5: Graphical representation of Example 1. (a) Pairwise Markov random field, (b) Factor graph.

Example 1. Consider the probability distribution

$$\mathbb{P}(x_1, x_2, x_3, x_4) \;=\; \psi_{13}(x_1, x_3)\,\psi_{23}(x_2, x_3)\,\psi_3(x_3)\,\psi_{34}(x_3, x_4),$$

defined over the discrete random variables $\{X_1, X_2, X_3, X_4\}$ (see Figure 2.5 for the corresponding graphical representation). Our goal is to compute $\mathbb{P}(x_4)$, the marginal density over X_4. By definition and use of the distributive law we have

$$\begin{aligned}\mathbb{P}(x_4) \;&=\; \sum_{x_1,x_2,x_3} \psi_{13}(x_1, x_3)\,\psi_{23}(x_2, x_3)\,\psi_3(x_3)\,\psi_{34}(x_3, x_4)\\ &=\underbrace{\Big(\sum_{x_3} \underbrace{\underbrace{\Big(\sum_{x_1} \psi_{13}(x_1, x_3)\Big)}_{m_{\{1,3\}\to 3}} \underbrace{\Big(\sum_{x_2} \psi_{23}(x_2, x_3)\Big)}_{m_{\{2,3\}\to 3}} \underbrace{\psi_3(x_3)}_{m_{\{3\}\to 3}}}_{m_{3\to\{3,4\}}} \psi_{34}(x_3, x_4)\Big)}_{m_{\{3,4\}\to 4}}. \qquad (2.7)\end{aligned}$$

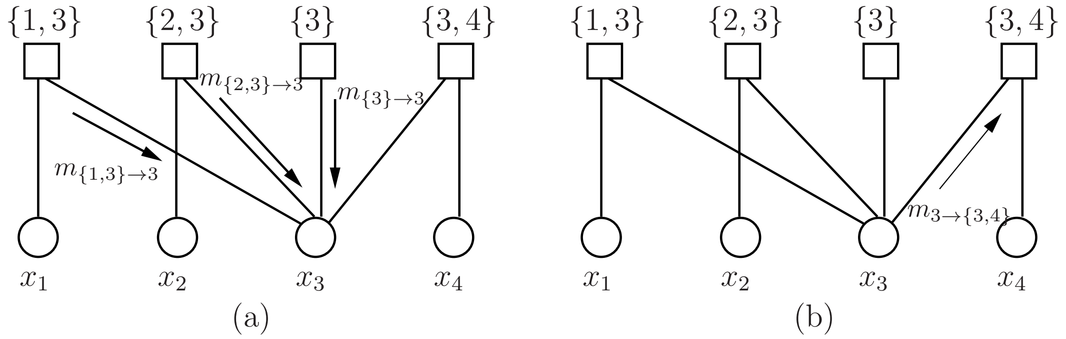


Figure 2.6: Message flow from (a) factor nodes to variable nodes, and (b) from variable nodes to factor nodes.

The previous equation can be expressed as a set of local message-passing between variable and factor nodes as depicted in equation (2.7). The message-passing rounds are also shown in Figure 2.6.

In order to setup the message-passing updates properly, we require some further notation. For every variable node $i \in \mathcal{V}_1$ let $\mathcal{N}(i) := \{I | i \in I\} \subset \mathcal{V}_2$ be the set of neighboring factor nodes and similarly for every $I \in \mathcal{V}_2$ define $\mathcal{N}(I) := \{i | i \in I\} \subset \mathcal{V}_1$, the set of neighboring variable nodes. In the BP algorithm, a pair of messages (μ-measurable functions) is assigned to every edge $(i, I) \in \mathcal{E}'$, one for each direction. In particular, let $m_{i\to I}$ be the message sent from the variable node i to the factor node I, and similarly let $m_{I\to i}$ be the message sent from the factor node I to the variable node i. Then, the message transmitted from a variable node to a factor node is the product of all incoming messages from neighboring factor nodes. (For instance, in our previous example, we have $m_{3\to\{3,4\}} = m_{\{1,3\}\to 3}\, m_{\{2,3\}\to 3}\, m_{\{3\}\to 3}$.) On the other hand, the message sent from a factor node to a variable node is obtained by the product of the local factor and the incoming messages summarized for the desired variable. (In the example discussed, we have $m_{\{3,4\}\to 4} = \sum_{x_3} m_{3\to\{3,4\}}\, \psi_{34}(x_3, x_4)$). Upon receiving all the messages from neighboring factor nodes, each variable node updates its estimate of the marginal distribution by the product of the incoming messages. The set of BP message-passing rounds are summarized in Figure 2.7.

For tree-structured (cycle-free) graphs, BP message updates can be derived as a divide-and-conquer algorithm: we solve a large problem by breaking it down to a set of smaller ones. The structure of trees provide a natural way of such decomposition. By solving the problem for the sub-trees emanating from the root first, it can be shown that BP message-based marginals converge to the exact marginals in a finite number of iterations [92, 62, 119]. More precisely, let diameter of the graph $\mathcal{G}$, denoted by $\mathrm{diam}(\mathcal{G})$, be the length of the longest path between any pair of nodes. Then, we have the following theorem:

Theorem 2 (**BP on trees**)**.** *Consider the sequence of marginals $\{\tau_i^{t+1}(x_i)\}_{t=0}^{\infty}$, for $i = 1, 2, \ldots, n$, generated by the BP algorithm on a tree-structured graphical model $\mathcal{G}$. Then, we have $\tau_i^{\mathrm{diam}(\mathcal{G})}(x_i) = \mathbb{P}(x_i)$.*

Belief Propagation Algorithm:

(I) For all factor nodes I and variable nodes $i \in I$, initialize the messages $m^0_{i\to I}(x_i) = 1$ at iteration $t = 0$.

(II) For iterations $t = 0, 1, 2, \ldots$ and $i \in I$, update the messages according to:

- factor to variable node:

$$m^{t+1}_{I\to i}(x_i) \;=\; \int \Big(\psi_I(x_I) \prod_{j\in\mathcal{N}(I)\backslash\{i\}} m^t_{j\to I}(x_j)\Big) \prod_{j\in\mathcal{N}(I)\backslash\{i\}} \mu(dx_j)$$

- variable to factor node:

$$m^{t+1}_{i\to I}(x_i) \;=\; \prod_{J\in\mathcal{N}(i)\backslash\{I\}} m^{t+1}_{J\to i}(x_i)$$

- node i update its marginal distribution

$$\tau^{t+1}_i(x_i) \;\propto\; \prod_{I\in\mathcal{N}(i)} m^{t+1}_{I\to i}(x_i)$$

Figure 2.7: Specification of the belief propagation algorithm on factor graphs.

Given the local form of the updates (Figure 2.7), the same message-passing updates can also be applied to more general graphs, which yields the "loopy" form of the belief propagation. (In this thesis we do not differentiate between the loopy and the normal form of the BP and we refer to both as BP.) The behavior of the ordinary BP algorithm to a graph with cycles—in contrast to the tree-structured case—is more complicated. On one hand, for strictly positive potential functions (as considered here), a version of Brouwer's fixed point theorem can be used to establish existence of fixed points [119]. However, in general, there may be multiple fixed points, and BP convergence is not guaranteed. Accordingly, various researchers have studied conditions that are sufficient to guarantee uniqueness of fixed points and/or convergence of the ordinary BP algorithm (e.g., [113, 49, 78, 103]). In addition, BP is known to be extremely effective for computing approximate marginals in numerous applications [62, 4, 121, 119].

2.3 Stochastic Approximation

In this section, we review some of the stochastic approximation techniques that are a crucial part of this thesis. Basically, stochastic approximation provides a general framework for analyzing the behavior of dynamical systems and associated algorithms in the presence of random variates. The range of scientific fields and applications that are affected by such techniques are astonishing. From a simple application of finding the roots of an unknown function from noise-corrupted observations to more complicated applications in adaptive signal processing, communication systems, artificial neural networks, and control theory, one can find footprints of stochastic approximation and optimization techniques. A common thread in all these applications is having a dynamical system in the presence of uncertainty (noise). In order to track the behavior of such systems, one requires adaptive algorithms with small increments that fits well within the framework of stochastic approximation. Since the seminal work of Robbins and Monro [101] in 1951, there has been a tremendous amount of research both in theory and in application. In this section we provide a brief overview of stochastic approximation referring the interested readers to numerous books and papers [65, 13, 23, 17, 71, 72, 59, 70, 63, 64].

2.3.1 General Framework and Motivating Examples

Stochastic approximation, in its abstract form, consists of a simple stochastic difference equation with small step size. Consider a dynamical system with parameter vector $\theta^t \in \mathbb{R}^n$ at time $t = 0, 1, \ldots$. In order to monitor and tune these parameters, we require to monitor the system. This task can be accomplished via the stochastic state vector $X^{t+1} \in \mathbb{R}^m$. Typically, the adaptive rule to update the system parameters from time t to $t+1$ will be of the form:[4]

$$\theta^{t+1} \;=\; \theta^t \,+\, \eta^t \, H\big(\theta^t, X^{t+1}\big). \tag{2.8}$$

Here η^t is a small positive step size and H is the observation function, which essentially determines how the parameters are updated. In order to motivate this formulation and demonstrate its widespread appeal, let us consider some simple examples.

Example 2 (Robbins-Monro). Our first example concerns finding a root of a function from noisy observations. Consider a real-valued function $f : \mathbb{R} \to \mathbb{R}$ with a unique root θ^* and suppose our goal is to estimate the root. If the function were known and differentiable,

[4] It is also possible to consider second order perturbation of the form

$$\theta^{t+1} \;=\; \theta^t \,+\, \eta^t \, H\big(\theta^t, X^{t+1}\big) \,+\, (\eta^t)^2 \, \epsilon^t\big(\theta^t, X^{t+1}\big).$$

For in depth analysis of such systems see the book [13].

then classic numerical techniques such as Newton's method could be applied. More precisely, one can form a sequence of estimators $\{\theta^t\}_{t=0}^{\infty}$ from the recursion

$$\theta^{t+1} \;=\; \theta^t \;-\; \frac{f(\theta^t)}{f'(\theta^t)},$$

where $f'(\cdot)$ denotes the derivative of the function $f(\cdot)$. It is known [19] that if $f'(\theta)$, is negative and bounded in the neighborhood of θ^*, and $|\theta^0 - \theta^*|$ is sufficiently small, then $\theta^t \to \theta^*$ geometrically fast as $t \to \infty$. An alternative method, which does not require differentiability, is to consider the following recursion for sufficiently small step size η,

$$\theta^{t+1} \;=\; \theta^t \;+\; \eta\, f(\theta^t).$$

Now suppose that the function f is not known and we only have access to it via noisy observations. More specifically, instead of $f(\theta^t)$ at time t, we observe its noise-corrupted version Y^t. To deal with this problem, Robbins and Monro [101] proposed the estimators $\{\theta^t\}_{t=0}^{\infty}$ derived from

$$\theta^{t+1} \;=\; \theta^t \;+\; \eta^t\, Y^t,$$

where the positive step sizes $\{\eta^t\}_{t=0}^{\infty}$ satisfy $\eta^t \to 0$ as $t \to \infty$, $\sum_{t=0}^{\infty} \eta^t = \infty$, and $\sum_{t=0}^{\infty} (\eta^t)^2 < \infty$. It should be noted that the observation error $Y^t - f(\theta^t)$ could be a complicated function of the past (i.e. $\{\theta^\tau\}_{\tau=0}^{t}$); however, the necessary condition for convergence is to have $\mathbb{E}\big[Y^t - f(\theta^t) \mid \{\theta^\tau\}_{\tau=0}^{t}\big] = 0$.

Example 3 (**Kiefer-Wolfowitz**)**.** Consider a concave function $f(\theta)$ ($f : \mathbb{R}^n \to \mathbb{R}$), and suppose we want to find its unique maximizer θ^*. If the function were known and differentiable, then standard algorithms such as gradient ascent [19] can be used to estimate θ^*. More specifically, we can form the estimators

$$\theta^{t+1} \;=\; \theta^t \;+\; \eta\, \nabla f(\theta^t), \quad \text{for } t = 0, 1, \ldots,$$

where $\eta > 0$ is a small step size and $\nabla f(\theta^t)$ denotes the gradient of the function f, computed at θ^t. Now suppose we don't have access to the function f and wish to maximize it only by observing a random function F with the correct mean (i.e. $\mathbb{E}[F(\theta)] = f(\theta)$). Intuitively, one could replace the gradient in the previous equation with its stochastic approximate and hope the estimators converge to the correct maximizer. Kiefer and Wolfowitz proposed and analyzed the following algorithm [59]. Let e_i denote the standard unite vector in the ith direction for $i = 1, 2, \ldots, n$ and let $\{a^t\}_{t=0}^{\infty}$ be a sequence of finite difference intervals. Composing the approximate derivative in the ith direction by

$$Y_i^t \;:=\; \frac{1}{2\,a^t}\,\big[F(\theta^t + a^t e_i) \;-\; F(\theta^t - a^t e_i)\big],$$

we define the approximate gradient

$$Y^t \;:=\; [Y_1^t, Y_2^t, \ldots, Y_n^t]^T \;\in\; \mathbb{R}^n,$$

at each iteration $t = 0, 1, \ldots$. Then, we update the parameter θ^t according to

$$\theta^{t+1} \;=\; \theta^t \,+\, \eta^t\, Y^t.$$

In this algorithm, the sequences $\{a^t\}_{t=0}^{\infty}$, and $\{\eta^t\}_{t=0}^{\infty}$ satisfy $a^t \to 0$, $\sum_{t=0}^{\infty} \eta^t = \infty$, $\sum_{t=0}^{\infty} a^t \eta^t < \infty$, and $\sum_{t=0}^{\infty} (\eta^t / a^t)^2 < \infty$ (e.g. $\eta^t = t^{-1}$, and $a^t = t^{-1/3}$).

Example 4 (**Adaptive equalization**)**.** Our final example deals with a more practical problem that is of great importance in adaptive signal processing and telecommunication systems [89]. Suppose we wish to transmit information through a linear time-invariant channel with impulse response $h = [h_0, h_1, \ldots, h_{\ell-1}]^T$. Denoting the data to be transmitted by $\{x_i\}_{i=0}^{\infty}$ and adopting the convention that $x_i = 0$ for $i < 0$, the output of the channel will be

$$y_k \;=\; \sum_{i=0}^{\ell-1} h_i\, x_{k-i} \,+\, \nu_k, \quad \text{for } k \geq 0,$$

where ν_k denotes the contaminating noise, normally modeled as an additive white Gaussian random variable. The receiver's objective is to estimate the transmitted signal from the output of the channel by maximizing the likelihood

$$\hat{x}_k \;=\; \arg\max_{x_k} \, \mathbb{P}(x_k \mid y_0, y_1, \ldots).$$

Typically in telecommunication systems, we have $\ell \geq 2$ that leads to inter-symbol interference and could seriously affect the communication's quality, even if the noise is negligible. Therefore, we need to devise a method to invert the effect of the channel. There are different approaches to deal with this issue. A popular method is to first learn the channel by transmitting a universally known training sequence and then reverse its effect. However, in many instances (specially in wireless communications) the channel is subject to significant temporal variations and requires an adaptive algorithm to keep track of the changes. An alternative approach is to consider an equalizer (a linear time-invariant filter) and gradually tune its parameters to match the inverse channel. Let $\theta := [\theta_0, \theta_1, \ldots, \theta_{n-1}]^T$ denote the equalizer's parameters. Passing the channel output through the equalizer, we obtain

$$z_k \;=\; \sum_{i=0}^{n-1} \theta_i\, y_{k-i}, \quad \text{for } k \geq 0,$$

from which quantization yields $\hat{x}_k = \Pi(z_k)$, the estimate of the input signal x_k. Here, $\Pi(\cdot)$ denotes a quantization scheme that maps the space of the output of the equalizer to the space of the input signal. In order to minimize the mean-squared error $\mathbb{E}[(x_k - \hat{x}_k)^2]$ and keep track of the changes in the channel, we need to tune the equalizer's parameters adaptively. This task can be accomplished as follows: we first initialize the equalizer at time $t = 0$. Then, for each iteration $t = 0, 1, \ldots$, we compute the output of the equalizer at instant t according to $z_t = \sum_{i=0}^{n-1} \theta_i^t y_{t-i}$. Finally, defining

$$Y^t \;:=\; (\Pi(z_t) - z_t)\,[y_t, y_{t-1}, \ldots, y_{t-n+1}]^T,$$

we update the equalizer via

$$\theta^{t+1} \;=\; \theta^t \,+\, \eta^t\, Y^t,$$

where η^t is a small positive step size. This procedure is repeated until convergence.

So far we have stated the general form of the stochastic approximation and provided motivating examples. As illustrated by these examples, typically, the purpose of an adaptive algorithm is to track an unknown parameter θ^*. Therefore, a few questions can be raised regarding the stochastic sequence $\{\theta^t\}_{t=0}^{\infty}$ generated by the update equation (2.8). The first question concerns the *asymptotic consistency* of the algorithm. Given the assumption that the desired quantity θ^* is unique, under what conditions do we have $\theta^t \to \theta^*$ almost surely as $t \to \infty$? The second question concerns the *asymptotic efficiency* of the algorithm. Assuming that θ^t converges to θ^*, how fast does it take place? Here, we are interested in the asymptotic distribution of a suitably normalized random sequence of the form $\{\sqrt{t}\,(\theta^t - \theta^*)\}_{t=0}^{\infty}$. The reminder of this section is devoted to addressing these questions. Later on the thesis, we develop more refined, *non-asymptotic* bounds (similar to the ones developed by other researchers in the optimization context [55, 46]) on the mean-squared error $\mathbb{E}\big[\|\theta^t - \theta^*\|_2^2\big]$ at each iteration $t = 0, 1, \ldots$.

2.3.2 Theoretical Guarantees

There are numerous asymptotic results regarding the stochastic approximation in the literature, results concerning the "finite vs. infinite horizon", "constant vs. decreasing step size", "bounded vs. unbounded state space", "Martingale difference vs. correlated noise", etc. Here, we only present some of the typical results relevant to our work. For an extensive treatment of the subject matter, see the excellent books by Kushner and Yin [65] and Benveniste, Metivier, and Priouret [13].

Much of the asymptotic analysis of stochastic approximation is based on exploiting the so-called ordinary differential equation (ODE) method [70, 63, 64, 13]. At a high level, the main idea is that the noise effect becomes averaged out in the long run (asymptotically); thus the behavior of the system can be explained by a mean ODE. The analysis is based on interpolating the discrete sequence $\{\theta^t\}_{t=0}^{\infty}$ into a continuous-time process with intervals equal to $\{\eta^t\}_{t=0}^t$. Denoting the continuous-time parameter by $\zeta \in \mathbb{R}^+$, one can define the interpolated process

$$\theta(\zeta) \;:=\; \theta^t, \quad \text{for} \quad \sum_{\tau=0}^{t} \eta^\tau \;\leq\; \zeta < \sum_{\tau=0}^{t+1} \eta^\tau .$$

The asymptotic of the discrete sequence ($t \to \infty$) is the same as the asymptotic of the interpolated process ($\zeta \to \infty$). It is the asymptotic behavior of the interpolated sequence that is modeled by that of a mean ODE. In order to make these ideas precise, we need to make some assumptions:

Assumption 1. The dynamic process of the state vector X^{t+1} can be represented by a Markov chain controlled by θ^t i.e.

$$\mathbb{P}(X^{t+1} \mid X^t, X^{t-1}, \ldots;\ \theta^t, \theta^{t-1}, \ldots) \;=\; \mathbb{P}(X^{t+1} \mid X^t;\ \theta^t).$$

Assumption 2. There exist a regular mean vector field defined by

$$h(\theta) \;:=\; \lim_{t\to\infty} \mathbb{E}_\theta[H(\theta, X^t)], \tag{2.9}$$

where the expectation is taken place with respect to X^t for a fixed value of θ. Moreover, the ODE

$$\frac{d\,\theta}{d\,\zeta} \;=\; h(\theta), \tag{2.10}$$

has an attractor θ^* with the domain of attraction D^*.[5]

Assumption 3. The sequence of step sizes $\{\eta^t\}_{t=0}^\infty$, satisfy $\eta^t \geq 0$, $\sum_{t=0}^\infty \eta^t = \infty$, and $\sum_{t=0}^\infty (\eta^t)^\alpha < \infty$, for some $\alpha > 1$.

With these assumptions we have the following theorem:

Theorem 3 (**Asymptotic consistency**)**.** *Consider the sequence $\{\theta^t\}_{t=0}^\infty$, generated by the update equation (2.8) with the initial point $\theta^0 \in Q$, where Q is a compact subset of the domain of attraction D^*. Also for a fixed compact set $Q' \in \mathbb{R}^m$, consider the set of trajectories (θ^t, X^{t+1}) that hit the compact set $Q \times Q'$ infinitely often. Then, under the Assumptions 1, 2, and 3, we have*

$$\theta^t \xrightarrow{a.s.} \theta^*, \quad \textit{almost surely as } t \to \infty.$$

[5] Normally the attractor is a single point but this may not always be the case. Also the domain of attraction is the set of starting points $\theta(0)$, such that $\theta(\zeta) \to \theta^*$ as $\zeta \to \infty$.

Theorem 3 is rather general. It does not address the verification of the boundedness condition that is the trajectory (θ^t, X^{t+1}) intersect a compact set infinitely often. However, it should be noted that often in practice, parameters $\{\theta^t\}_{t=0}^{\infty}$ belong to a closed and bounded space; therefore, the condition is automatically satisfied. In order to guarantee the validity of the boundedness condition, several authors [65, 72] have proposed a two-phase algorithm, in which after the update (2.8) the parameter θ^{t+1} gets projected onto some compact space. There are other results that does not require the verification of the boundedness condition. The next theorem, associated to the case of Robbins and Monro, provides the same result under somewhat more restrictive conditions. (We will make use of the this theorem in the following chapters). Suppose the problem of stochastic approximation satisfy:

Assumption 4. The dynamic process of the state vector X^{t+1} can be controlled by θ^t i.e.

$$\mathbb{P}(X^{t+1} \mid X^t, X^{t-1}, \ldots;\, \theta^t, \theta^{t-1}, \ldots) \;=\; \mathbb{P}(X^{t+1} \mid \theta^t).$$

Assumption 5. There exist a constant c such that

$$\mathbb{E}_\theta\left[\|H(\theta, X)\|_2^2\right] \;\leq\; c\,(1 + \|\theta\|_2^2),$$

where the expectation is taken place with respect to the random variable X for a fixed θ.

Assumption 6. The mean vector field defined in (2.9), satisfy the following stability condition: there exists θ^* such that

$$\sup_{\theta \neq \theta^*} (\theta - \theta^*)^T\, h(\theta) \;<\; 0.$$

With these assumptions, we have:

Theorem 4 (Robbins-Monro). *Consider the sequence $\{\theta^t\}_{t=0}^{\infty}$, generated by the update equation (2.8). Then, under the Assumptions 3 (for $\alpha = 2$), 4, 5, and 6, we have*

$$\theta^t \xrightarrow{a.s.} \theta^*, \quad \textit{almost surely as } t \to \infty.$$

In addition to consistency, we are also interested in the rate of convergence to which we now turn. But first, we need to define some further notation. Let h' denote the Hessian matrix of the mean vector field

$$h' \;:=\; \frac{d\,h}{d\,\theta}.$$

Moreover, let Σ be the covariance matrix

$$\Sigma(\theta) \;:=\; \sum_{t=0}^{\infty} \operatorname{cov}_\theta[H(\theta, X^t), H(\theta, X^0)],$$

where $\operatorname{cov}_\theta$ denotes the covariance operator when the parameter θ is fixed.

Assumption 7. Assume the covariance $\Sigma(\theta)$ exists for $\theta = \theta^*$, the attractor of the ODE (2.10). In addition, suppose eigenvalues of the Hessian matrix $h'(\theta^*)$, computed at $\theta = \theta^*$, have real values strictly less than $-1/2$.

The following theorem, address the issue of the rate of convergence.

Theorem 5 (**Asymptotic efficiency**)**.** *Suppose the step size η^t is of the order of $1/t$, i.e. there exist constants c_1 and c_2 such that $c_1/t \leq \eta^t \leq c_2/t$, for all $t = 1, 2, \ldots$. Then under the Assumptions 1, 2, and 7, we have*

$$\sqrt{t}\,(\theta^t \,-\, \theta^*) \;\xrightarrow{d}\; N(0, C), \quad \text{as } t \to \infty,$$

where the convergence is in distribution, and $N(0, C)$ is a multivariate Gaussian random variable with zero mean and covariance C. Moreover, C is the unique, symmetric, positive semi definite solution of the Lyapunov equation

$$C\left(\frac{I}{2} + h'(\theta^*)\right)^T \,+\, \left(\frac{I}{2} + h'(\theta^*)\right) C \,+\, \Sigma(\theta^*) \;=\; 0. \tag{2.11}$$

Theorem 5 provides a more refined result for the stochastic approximation algorithm. Under some stability condition (Assumption 7), the sequence of normalized errors, $\{\sqrt{t}(\theta^t - \theta^*)\}_{t=0}^{\infty}$, converges to a Gaussian distribution with zero mean and finite covariance. This result state that, roughly speaking, the squared error $\|\theta^t - \theta^*\|_2^2$ decays as c/t, for some constant c derived from the Lyapunov equation (2.11).

2.4 Gossip Algorithms

As discussed in the previous chapter, normally in sensor and peer-to-peer networks, we are interested in computing a global quantity based on local observations made by every mote. Obviously flooding the network—that is, having nodes act as relays and pass along whatever they receive until everybody obtains the whole information—is not feasible due to memory and energy constraints. Gossip type algorithms provide a distributed method to achieve that objective. As suggested by its name, it is inspired by the way a “gossip” gets distributed in a social network. In a nutshell, at every iteration a sensor wakes up, chooses one of its neighbors at random and pass along the “gossip” (the estimate of the desired quantity). Upon acquiring the new information, the receiving sensor updates its own “gossip”. This procedure is repeated till the entire network obtains the whole information and reaches a consensus.

Network-constrained averaging is a special but important instance of such problems. Consider a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ modeling a sensor network and suppose every sensor $i = 1, 2, \ldots, n$ has an observation θ_i^0 at time $t = 0$. By exchanging messages along edges of the graph, the network's objective is to reach a consensus on the global average $\bar{\theta} := (\sum_{i=1}^n \theta_i^0)/n$. In particular, denoting the vector of estimates $\theta^t = [\theta_1^t, \theta_2^t, \ldots, \theta_n^t]^T$ at iteration $t = 0, 1, \ldots$, we are interested in the convergence $\theta^t \to \bar{\theta}\vec{1}$ as $t \to \infty$, where $\vec{1}$ denotes the all one vector. There are various ways of achieving this objective. For illustrative purposes, we consider the one proposed by Boyd et al. [18] concerning the case of noiseless communication.

At iteration t, one node, say node i, wakes up and picks one of its neighbors,[6] say node j, with probability p_{ij}. Then, nodes i and j update their estimates at time $t+1$ with the average of their current estimates; more specifically, they set $\theta_i^{t+1} = \theta_j^{t+1} = (\theta_i^t + \theta_j^t)/2$. The rest of the nodes remain unchanged, i.e. $\theta_k^{t+1} = \theta_k^t$, for $k \neq i, j$. These updates can be written in the vector form

$$\theta^{t+1} \;=\; W^t\,\theta^t \quad \text{for } t = 0, 1, \ldots, \tag{2.12}$$

where W^t is a symmetric and stochastic averaging matrix. Boyd et al. proved that the algorithm is strongly consistent meaning that $\theta^t \xrightarrow{\text{a.s.}} \bar{\theta}\vec{1}$, almost surely as $t \to \infty$. Moreover, they showed that the rate of convergence is inversely proportional to the second smallest eigenvalue of the matrix $I - \mathbb{E}\big[W^t\big]$, also known as the spectral radius. This quantity that is closely related to the mixing time of a random walk with jump probabilities $[p_{ij}]$, provides a measure of information diffusion in the network.

As it turns out, even for the optimum set of jump probabilities, the previous algorithm diffuses very slowly. In particular, for popular sensor network models of grid and random geometric graph, obtaining an accurate solution requires order n^2 iterations. Changing the averaging matrix W^t, in order to obtain faster diffusion, researcher have proposed more efficient averaging algorithms [31, 12]. By averaging nodes that are not necessarily neighbors, Dimakis et al. [31] proposed geographic gossip that drops factors of $\sqrt{n}$ and $\sqrt{n/\log n}$ from the required number of communications for the cases of grid and RGG respectively. On the other hand, by establishing a stochastic route and averaging alongside it, Benezit et al. [12] reduced the number of communications to n (essentially the optimal scaling) for the cases of grid and RGG. Both of these algorithms consider perfect, noiseless communications. However, more realistic models should account for random variations such as noise. Simple updates of the form (2.12) will fail in the noisy environments. Therefore, we need to use stochastic approximation techniques (discussed in Section 2.3), suitable for dealing with noise effects. When the communication channels are modeled as AWGN, Rajagopal and Wainwright [95] analyzed a damped version of the usual consensus updates, and provided scaling of the iteration number as a function of the graph topology and size. We will discus the problem of network scaling for noisy averaging in more detail in Chapter 5.

[6] Here the set of neighbors include the node i itself. In the case of choosing i, the update will not change, i.e. $\theta_i^{t+1} = \theta_i^t$.

Chapter 3

Stochastic Belief Propagation

3.1 Introduction

In this chapter, we focus on the problem of implementing the belief propagation message-passing for high-dimensional discrete random variables. In many applications of BP, the messages themselves are high-dimensional in nature, either due to discrete random variables with a very large number of possible realizations d (which will be referred to as the number of states), due to factor nodes with high degree, or due to continuous random variables that are discretized. Examples of such problems include disparity estimation in computer vision, tracking problems in sensor networks, and error-control decoding. For such problems, it may be expensive to compute and/or store the messages, and as a consequence, BP may run slowly, and be limited to small-scale instances. Motivated by this challenge, researchers have studied a variety of techniques to reduce the complexity of BP in different applications (e.g., see the papers [38, 110, 76, 51, 53, 58, 27, 106, 50, 20, 66, 105, 114, 97] and references therein). At the core of the BP message-passing is a matrix-vector multiplication, with complexity scaling quadratically in the number of states d. Certain graphical models have special structures that can be exploited so as to reduce this complexity. For instance, in applications to the decoding of low-density parity-check codes in channel coding (e.g., [39, 62]), the complexity of message-passing, if performed naively, would scale exponentially in the factor degrees. However, a clever use of the fast Fourier transform over $\mathrm{GF}(2^r)$ reduces this complexity to linear in the factor degrees (e.g., see the paper [105] for details). Other problems arising in computer vision involve pairwise factors with a circulant structure for which the fast Fourier transform can also reduce complexity [38]. Similarly, computation can be accelerated by exploiting symmetry in factors [58], or additional factorization properties of the distribution [76]. In the absence of structure to exploit, other researchers have proposed different types of quantization strategies for BP message updates [27, 53], as well as stochastic methods based on particle filtering or non-parametric belief propagation (e.g., [5, 110, 32]) that approximate continuous messages by finite numbers of particles. For certain classes of these methods, it is possible to establish consistency as the number of particles tends to infinity [32], or to establish non-asymptotic results inversely proportional to the square root of the number of particles [51]. As the number of particles diverges, the approximation error becomes negligible, a property that underlies such consistency proofs. Researchers have also proposed stochastic techniques to improve the decoding efficiency of binary error-correcting codes [114, 97]. These techniques, which are based on encoding messages with sequences of Bernoulli random variables, lead to efficient decoding hardware architectures.

The main contribution of this chapter is to propose a novel low-complexity algorithm, which we refer to as *stochastic belief propagation* (SBP) and provide rigorous mathematical analysis of its performance. As suggested by its name, it is an adaptively randomized version of the BP algorithm, where each node only passes randomly selected partial information to its neighbors at each round. The SBP algorithm has two features that make it practically appealing. First, it reduces the computational cost of BP by an order of magnitude; in concrete terms, for arbitrary pairwise potentials over d states, it reduces the per iteration computational complexity from quadratic to linear—that is, from $\Theta\big(d^2\big)$ to $\Theta\big(d\big)$.[1] Second, it significantly reduces the message/communication complexity, requiring transmission of only $\log_2 d$ bits per edge as opposed to $(d-1)$ real numbers in the case of BP.

Even though SBP is based on low-complexity updates, we are able to establish conditions under which it converges (in a stochastic sense) to the exact BP fixed point, and moreover, to establish quantitative bounds on this rate of convergence. These bounds show that SBP can yield provable reductions in the complexity of computing a BP fixed point to a tolerance $\delta > 0$. In more precise terms, we first show that SBP is strongly consistent on any tree-structured graph, meaning that it converges almost surely to the unique BP fixed point; in addition, we provide non-asymptotic upper bounds on the ℓ_∞ norm (maximum value) of the error vector as a function of iteration number (Theorem 6). For general graphs with cycles, we show that when the ordinary BP message updates satisfy a type of contraction condition, then the SBP message updates are strongly consistent, and converge in normalized mean-squared error at the rate $\mathcal{O}(1/t)$ to the unique BP fixed point, where t is the number of iterations. We also show that the typical performance is sharply concentrated around its mean (Theorem 7). These theoretical results are supported by simulation studies, showing the convergence of the algorithm on various graphs, and the associated reduction in computational complexity that is possible.

The remainder of this chapter is organized as follows. We begin in Section 3.2 with background and problem statement. In Section 3.3, we provide a precise description of the SBP, before turning in Section 3.4 to statements of our main theoretical results, as well as discussion of some of their consequences. Section 3.5 is devoted to the proofs of our results, with more technical aspects of the proofs deferred to the Appendices. In Section 3.6,

[1] The notation $f(d) = \mathcal{O}(g(d))$ means that there exists a fixed constant c so that $f(d) \leq c\, g(d)$, whereas $f(d) = \Theta(g(d))$ means that there exists constants c and c' such that $c'\, g(d) \leq f(d) \leq c\, g(d)$, for sufficiently large d.

we demonstrate the correspondence between our theoretical predictions and the algorithm's practical behavior. Finally, we conclude the chapter in Section 3.7.

3.2 Background and Problem Statement

In this section, we set-up the notation and state the precise formulation of the problem. Consider a pair wise Markov random field $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ over the discrete random variables $\{X_1, X_2, \ldots, X_n\}$ each of which taking values in some space $\mathcal{X} := \{1, 2, \ldots, d\}$ with cardinality d. The probability densities associated with this graphical model is factorized accroding to

$$\mathbb{P}(x_1, x_2, \ldots, x_n) \;\propto\; \prod_{u \in \mathcal{V}} \psi_u(x_u) \prod_{(u,v) \in \mathcal{E}} \psi_{vu}(x_v, x_u), \tag{3.1}$$

where $\psi_u : \mathcal{X} \to (0, \infty)$ denotes the node potential for $u \in \mathcal{V}$, and $\psi_{uv} : \mathcal{X} \times \mathcal{X} \to (0, \infty)$ denotes the edge potential for $(u, v) \in \mathcal{E}$. As discussed in the previous chapter, a computational challenge important to many applications is the computation of the marginal distributions

$$\mathbb{P}(x_1) \quad := \sum_{x'_2} \cdots \sum_{x'_n} \mathbb{P}\left(x_1, x'_2, \ldots, x'_n\right), \tag{3.2}$$

similarly defined for other variables. A naive approach to this problem would incur d^{n-1} computation which becomes intractable even for small problems. However, this computational challenge can (to some extent) be resolved by the BP algorithm. Since all factor nodes on a pairwise Markov random field have degree less than or equal to two, BP message updates take a simple form on such graphical models.

In order to define the message-passing updates, we require some further notation. For each node $u \in \mathcal{V}$, let $\vec{\mathcal{E}}(u) := \{(u \to v) \mid v \in \mathcal{N}(u)\}$ denote the set of all directed edges emanating from u, where $\mathcal{N}(u) := \{v \mid (u, v) \in \mathcal{E}\}$ denote its set of neighbors. Moreover, we define $\vec{\mathcal{E}} := \cup_{u \in \mathcal{V}} \vec{\mathcal{E}}(u)$, the set of all directed edges in the graph; note that $\vec{\mathcal{E}}$ has cardinality $2|\mathcal{E}|$. In the BP algorithm, one message $m_{u \to v} \in \mathbb{R}^d$ is assigned to every directed edge $(u \to v) \in \vec{\mathcal{E}}$. By concatenating all of these d-dimensional vectors, one for each of the $2|\mathcal{E}|$ members of $\vec{\mathcal{E}}$, we obtain a D-dimensional vector of messages $m = \{m_{u \to v}\}_{(u \to v) \in \vec{\mathcal{E}}}$, where $D := 2|\mathcal{E}|d$.

At each round $t = 0, 1, 2, \ldots$, every node $u \in \mathcal{V}$ calculates a message $m^{t+1}_{u \to v} \in \mathbb{R}^d$ to be sent to its neighbor $v \in \mathcal{N}(u)$. In mathematical terms, this operation can be represented as an update of the form $m^{t+1}_{u \to v} = F_{u \to v}(m^t)$ where $F_{u \to v} : \mathbb{R}^D \to \mathbb{R}^d$ is the local update function of the directed edge $(u \to v)$. In more detail, for each $x_v \in \mathcal{X}$, we have

$$m^{t+1}_{u \to v}(x_v) = [F_{u \to v}(m^t)](x_v) \;=\; \kappa \sum_{x_u \in \mathcal{X}} \Bigg(\psi_{vu}(x_v, x_u)\, \psi_u(x_u) \prod_{w \in \mathcal{N}(u) \backslash \{v\}} m^t_{w \to u}(x_u) \Bigg), \tag{3.3}$$

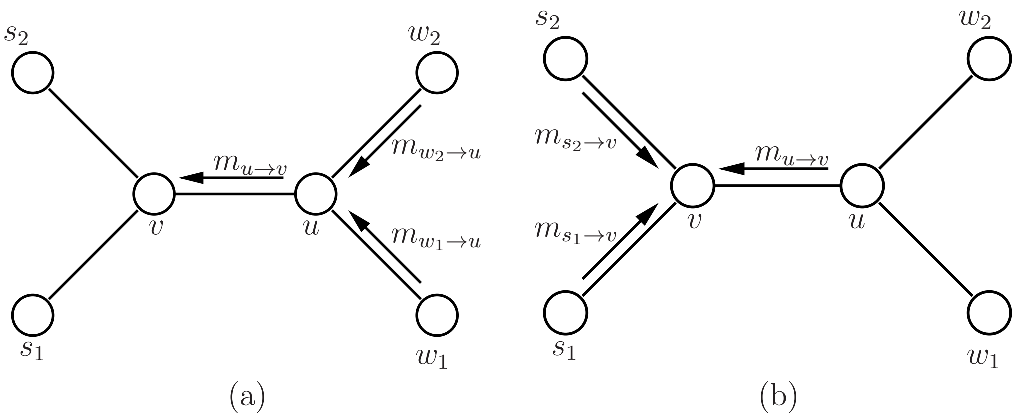


Figure 3.1: Graphical representation of message-passing algorithms. (a) Node u transmits the message $m_{u\to v} = F_{u\to v}(m)$, derived from (3.3), to its neighbor v. (b) Upon receiving all the messages, node v updates its marginal estimate.

where κ is a normalization constant chosen to ensure that $\sum_{x_v} m^{t+1}_{u\to v}(x_v) = 1$. Figure 3.1(a) provides a graphical representation of the flow of information in this local update. It is worth mentioning that $m^{t+1}_{u\to v}$ is only a function of the messages $m^t_{w\to u}$ for $w \in \mathcal{N}(u)\backslash\{v\}$. Therefore, we have $F_{u\to v} : \mathbb{R}^{(\rho_u-1)d} \to \mathbb{R}^d$, where ρ_u is the degree of the node u. Since it is clear from the context and for the purpose of reducing the notation overhead, we say $m^{t+1}_{u\to v} = F_{u\to v}(m^t)$ instead of $m^{t+1}_{u\to v} = F_{u\to v}(\{m^t_{w\to u}\}_{w\in\mathcal{N}(u)\backslash\{v\}})$.

By concatenating the local updates (3.3), we obtain a global update function $F : \mathbb{R}^D \to \mathbb{R}^D$ of the form

$$F(m) \;=\; \{F_{u\to v}(m)\}_{(u\to v)\in\vec{\mathcal{E}}}. \tag{3.4}$$

Typically, the goal of message-passing is to obtain a *fixed point*, meaning a vector $m^* \in \mathbb{R}^D$ such that $F(m^*) = m^*$ and (3.3) can be seen as an iterative way of solving this fixed-point equation. For any tree-structured graph, it is known that the update (3.4) has a unique fixed point. For a general graph (with some mild conditions on the potentials; see Yedidia et al. [121] for details), it is known that the global update (3.4) has at least one fixed point, but it is no longer unique in general. However, there are various types of contraction conditions that can be used to guarantee uniqueness on a general graph (e.g., [113, 49, 78, 103]). Given a fixed point m^*, node v computes its marginal (approximation) τ^*_v by combining the local potential function ψ_v with a product of all incoming messages as

$$\tau^*_v(x_v) \;=\; \kappa\, \psi_v(x_v) \prod_{u\in\mathcal{N}(v)} m^*_{u\to v}(x_v), \tag{3.5}$$

where κ is a normalization constant chosen so that $\sum_{x_v\in\mathcal{X}} \tau^*_v(x_v) = 1$. See Figure 3.1(b) for an illustration of this computation. For any tree-structured graph, the quantity $\tau^*_v(x_v)$ is equal to the single-node marginal $\mathbb{P}(x_v)$, as previously defined (3.2). For a graph with cycles, the vector τ_v^* represents an approximation to the single-node marginal, and is known to be a useful approximation for many classes of graphical models.

When applied to a pairwise graphical model with random variables taking d states, the number of summations and/or multiplications required by the original BP algorithm is $\Theta\big(d^2\big)$ per iteration and per edge as can be seen by inspection of the message update equation (3.3). This quadratic complexity—which is incurred on a per iteration, per edge basis—is prohibitive in many applications, where the state dimension may be on the order of thousands. As discussed earlier in Section 3.1, although certain graphical models have particular structures that can be exploited to reduce the complexity of the updates, not all problems have such special structures, so that a general-purpose approach is of interest. In addition to computational cost, a standard BP message update can also be expensive in terms of communication cost, since each update requires transmitting $(d-1)$ real numbers along each edge. For applications that involve power limitations, such as sensor networks, reducing this communication cost is also of interest.

3.3 Description of the SBP Algorithm

We now turn to a description of the SBP, a low-complexity message-passing algorithm on pairwise Markov random fields. Stochastic belief propagation is an adaptively randomized form of the usual BP message updates that yields savings in both computational and communication costs. It is motivated by a simple observation—namely, that the message-passing update along the directed edge $(u \to v)$ can be formulated as an expectation over suitably normalized columns of a compatibility matrix (see (3.6)). Here the probability distribution in question depends on the incoming messages, and changes from iteration to iteration. This perspective leads naturally to an *adaptively randomized variant* of BP: instead of computing and transmitting the full expectation at each round—which incurs $\Theta(d^2)$ computational cost and requires sending $\Theta(d)$ real numbers—the SBP algorithm simply picks a single normalized column with the appropriate (message-dependent) probability, and performs a randomized update. As we show, each such operation can be performed in $\Theta(d)$ time and requires transmitting only $\log_2 d$ bits, so that the SBP message updates are less costly by an order of magnitude.

With this intuition in hand, we are now ready for a precise description of the SBP algorithm. Let us view the edge potential function ψ_{uv} as a matrix of numbers $\psi_{uv}(i,j)$, for $i,j = 1,\ldots,d$. For the directed edge $(u \to v)$, define the collection of column vectors

$$\Gamma_{uv}(:,j) \;:=\; \frac{\psi_{vu}(:,j)}{\sum_{i=1}^{d} \psi_{vu}(i,j)}, \tag{3.6}$$

and marginal weights $\beta_{uv}(j) := \big(\sum_{i=1}^{d} \psi_{vu}(i,j)\big)\, \psi_u(j)$, for $j = 1,2,\ldots,d$. Note that, the columns of the compatibility matrix Γ_{uv} are normalized to sum to one: i.e., $\sum_{i=1}^{d} \Gamma_{uv}(i,j) = 1$

for all $j = 1, 2, \ldots, d$. We assume that the column vectors $\Gamma_{uv}(:, j)$ and normalization constants $\beta_{uv}(j)$ have been pre-computed and stored, which can be done in an off-line manner and requires $\Theta(d^2)$ operations. In addition, the algorithm makes use of a positive sequence of step sizes $\{\lambda^t\}_{t=0}^{\infty}$. In terms of these quantities, the SBP algorithm consists of the steps shown in Figure 3.2.

Stochastic Belief Propagation Algorithm:

(I) Initialize the message vector $m^0 \in \mathbb{R}_+^D$.

(II) For iterations $t = 0, 1, 2, 3, \ldots$, and for each directed edge $(u \to v) \in \vec{\mathcal{E}}$:

(a) Compute the product of incoming messages:

$$M_{u\to v}^t(j) \quad = \prod_{w\in\mathcal{N}(u)\backslash\{v\}} m_{w\to u}^t(j) \quad \text{for } j \in \{1, \ldots, d\}. \tag{3.7}$$

(b) Pick a random index $J_{u\to v}^{t+1} \in \{1, 2, \ldots, d\}$ according to the probability distribution

$$p_{u\to v}^t(j) \;\propto\; M_{u\to v}^t(j)\, \beta_{uv}(j) \quad \text{for } j \in \{1, \ldots, d\}. \tag{3.8}$$

(c) For a given step size $\lambda^t \in (0, 1)$, update the message $m_{u\to v}^{t+1} \in \mathbb{R}_+^d$ via

$$m_{u\to v}^{t+1} \;=\; (1 - \lambda^t)\, m_{u\to v}^t \;+\; \lambda^t\, \Gamma_{uv}(:, J_{u\to v}^{t+1}). \tag{3.9}$$

Figure 3.2: Specification of stochastic belief propagation.

The per iteration per edge computational complexity of the SBP algorithm lies in calculating the probability mass function $p_{u\to v}$, defined in (3.8); generating a random index $J_{u\to v}$ according to the mass function (3.8), and performing the weighted update (3.9). Denoting the maximum degree of the graph by $\rho_{\max}$, we require at most $(\rho_{\max} - 1)d$ multiplications to compute $M_{u\to v}$. Moreover, an additional $2d$ operations are needed to compute the probability mass function $p_{u\to v}$. On the other hand, generating a random index $J_{u\to v}$, can be done with less than d operations by picking a number U uniformly at random from $[0, 1]$ and setting[2] $J_{u\to v} \;:=\; \inf\big\{j : \sum_{\ell=1}^{j} p_{u\to v}(\ell) > U\big\}$. Finally the update (3.9) needs $3d + 3$ operations. Adding up these contributions, we find that the SBP algorithm requires at most

[2] It is known that for any distribution function $G(\cdot)$, the random variable $G^{-1}(U)$ has the distribution $G(\cdot)$.

$(\rho_{\max}+5)d+3$ multiplications and/or summations per iteration per edge to update the messages. As can be seen from (3.3), the regular BP complexity is $\Theta\big(d^2\big)$. Therefore, for graphs with bounded degree (of most interest in practical applications), the SBP message updates have reduced the per iteration computational complexity by a factor of d. In addition to computational efficiency, SBP provides us with a significant gain in message/communication complexity over BP. This can be observed from the fact that the normalized compatibility matrix Γ_{uv} is only a function of edge potentials ψ_{vu}, hence known to the node v. Therefore, node u has to transmit only the random column index $J_{u\to v}$ to node v, which can be done with $\log_2 d$ bits. This is a significant gain over BP that requires transmitting a $(d-1)$-dimensional vector of real numbers per edge at every round. Here we summarize the features of our algorithm that make it appealing for practical purposes.

- *Computational complexity*: SBP reduces the per iteration complexity by an order of magnitude from $\Theta(d^2)$ to $\Theta(d)$.
- *Communication complexity*: SBP requires transmitting only $\log_2 d$ bits per edge in contrast to transmitting a $(d-1)$-dimensional vector of real numbers in the case of BP.

The remainder of this chapter is devoted to understanding when, and if so, how quickly the SBP message updates converge to a BP fixed point. Let us provide some intuition as to why such a behavior might be expected. Recall that the update (3.9) is random, depending on the choice of index J chosen in step II(b). Suppose that we take expectations of the update (3.9) only over the distribution (3.8), in effect conditioning on all past randomness in the algorithm. (We make this idea precise via the notion of σ-fields in our analysis.) Doing so yields that the expectation of the update (3.9) is given by

$$\mathbb{E}\big[m^{t+1}_{u\to v} \mid m^t_{u\to v}\big] \;=\; (1-\lambda^t)\, m^t_{u\to v} \,+\, \lambda^t \sum_{j=1}^{d} \Gamma_{uv}(:,j)\, p^t_{u\to v}(j).$$

Recalling the definitions (3.6) and (3.8) of the matrix Γ_{uv} and mass function $p_{u\to v}$, respectively, and performing some algebra yields

$$\begin{aligned}\mathbb{E}\big[m^{t+1}_{u\to v} \mid m^t_{u\to v}\big] &= (1-\lambda^t)\, m^t_{u\to v}\\ &\quad + \lambda^t \sum_{j=1}^{d} \frac{\psi_{vu}(:,j)}{\sum_{i=1}^{d}\psi_{vu}(i,j)}\,\frac{\prod_{w\in\mathcal{N}(u)\backslash\{v\}} m^t_{w\to u}(j)\,\beta_{uv}(j)}{\sum_{\ell=1}^{d}\prod_{w\in\mathcal{N}(u)\backslash\{v\}} m^t_{w\to u}(\ell)\,\beta_{uv}(\ell)}\\ &= (1-\lambda^t)\, m^t_{u\to v} \,+\, \lambda^t\, F_{u\to v}(m^t).\end{aligned}$$

Therefore, in an average sense, the SBP message update is equivalent to (a damped version of the) usual BP message update. The technical difficulties lie in showing that despite the fluctuations around this average behavior, the SBP updates still converge to the BP fixed point when the stepsize or damping parameter λ^t is suitably chosen. We now turn to precisely this task.

3.4 Main Theoretical Results

Thus far, we have proposed a low-complexity stochastic variant of the usual belief propagation algorithm. In contrast to the usual deterministic updates, this algorithm generates a random sequence $\{m^t\}_{t=0}^{\infty}$ of message vectors. This randomness raises two natural questions:

- Is the SBP algorithm *strongly consistent*? More precisely, assuming that the ordinary BP algorithm has a unique fixed point m^*, under what conditions do we have $m^t \to m^*$ almost surely as $t \to \infty$?
- When convergence occurs, *how fast* does it take place? The computational complexity per iteration is significantly reduced, but what are the trade-offs incurred by the number of iterations required?

The goal of this section is to provide some precise answers to these questions, ones which show that under certain conditions, there are provable gains to be achieved by the SBP algorithm. We begin with the case of trees, for which the ordinary BP message updates are known to have a unique fixed point for any choice of potential functions. For any tree-structured problem, the upcoming Theorem 6 guarantees that the SBP message updates are strongly consistent, and moreover that in terms of the elementwise ℓ_∞ norm they converge in expectation at least as quickly as $\mathcal{O}(1/\sqrt{t})$, where t is the number of iterations. We then turn to the case of general graphs. Although the BP fixed point need not be unique in general, a number of contractivity conditions that guarantee uniqueness and convergence of ordinary BP have been developed (e.g., [113, 49, 78, 103]). Working under such conditions, we show in Theorem 7 that the SBP algorithm is strongly consistent, and we show that the normalized mean-squared error decays at least as quickly as $\mathcal{O}(1/t)$. In addition, we provide high probability bounds on the error at each iteration, showing that the typical performance is highly concentrated around its average. Finally, in Section 3.4.3, we provide a new set of sufficient conditions for contractivity in terms of node/edge potentials and the graph structure. As we discuss, our theoretical analysis shows not only that SBP is provably correct, but also that in various regimes, substantial gains in overall computational complexity can be obtained relative to the ordinary BP.

3.4.1 Guarantees for Tree-Structured Graphs

We begin with the case of a tree-structured graphical models. As a special case, the hidden Markov chain shown in Figure 2.3 is an instance of such graphs. Recall that for some integer $r \geq 1$, a square matrix A is said to be nilpotent of degree r if $A^r = 0$. (We refer the reader to Horn and Johnson [47] for further background on nilpotent matrices and their properties.) Also recall the definition of the diameter of a graph $\mathcal{G}$, denoted by $\operatorname{diam}(\mathcal{G})$, as the length (number of edges) of the longest path between any pair of nodes in the graph. For a tree, this diameter can be at most $n-1$, a bound achieved by the chain graph. In stating Theorem 6, we make use of the following definition: for vectors $x, y \in \mathbb{R}^D$, we write $x \preceq y$ if and only if $x(i) \leq y(i)$ for all $i = 1, 2, \ldots, D$. Moreover, for an arbitrary $x \in \mathbb{R}^D$, let $|x|$ denote the vector obtained from taking the absolute value of its elements. With this notation in hand, we are now ready to state our first result.

Theorem 6 (**Tree-structured graphs**). *For any tree-structured Markov random field, the sequence of messages $\{m^t\}_{t=0}^{\infty}$ generated by the SBP algorithm with step size $\lambda^t = 1/(t+1)$, has the following properties:*

(a) The message sequence $\{m^t\}_{t=0}^{\infty}$ converges almost surely to the unique BP fixed point m^* as $t \to \infty$.

(b) There exist a nilpotent matrix $A \in \mathbb{R}^{D \times D}$ of degree at most $r = \operatorname{diam}(\mathcal{G})$ such that the D-dimensional error vector $m^t - m^*$ satisfies the elementwise inequality

$$\mathbb{E}\big[|m^t - m^*|\big] \preceq \; 4\,(I - 2A)^{-1}\,\frac{\vec{1}}{\sqrt{t}} \tag{3.10}$$

for all iterations $t = 1, 2, \ldots$.

Remarks: The proof of this result is given in Section 3.5.1. Part (a) shows that the SBP algorithm is guaranteed to converge almost surely to the unique BP fixed point, regardless of the choice of node/edge potentials and the initial message vector. Part (b) refines this claim by providing a quantitative upper bound on the rate of convergence: in expectation, the ℓ_∞ norm of the error vector is guaranteed to decay at the rate $\mathcal{O}(1/\sqrt{t})$. It is worth noting that the upper bound in part (b) is likely to be conservative at times, since the inverse matrix $(I - 2A)^{-1}$ may have elements that grow exponentially in the graph diameter r. As shown by our experimental results, the theory is overly conservative in this way, as SBP still behaves well on trees with large diameters (such as chains). Indeed, in the following section, we provide results for general graphs under contractive conditions that are less conservative.

3.4.2 Guarantees for General Graphs

Our next theorem addresses the case of general graphs. In contrast to the case of tree-structured graphs, depending on the choice of potential functions, the BP message updates may have multiple fixed points, and need not converge in general. A sufficient condition for both uniqueness and convergence of the ordinary BP message updates, which we assume in our analysis of SBP, is that the update function F, defined in (3.4), is *contractive*. In particular, it suffices that there exist some $0 < \mu < 2$ such that

$$\|F(m) - F(m')\|_2 \leq \big(1 - \frac{\mu}{2}\big)\,\|m - m'\|_2. \tag{3.11}$$

Recalling the normalized compatibility matrix with columns $\Gamma_{uv}(:,j) := \psi_{vu}(:,j)\psi_u(j)/\beta_{uv}(j)$, we define its minimum and maximum values per row as follows:[3]

$$\underline{B}^0_{uv}(i) := \min_{j\in\mathcal{X}} \Gamma_{uv}(i,j) \;>\; 0, \quad \text{and} \quad \overline{B}^0_{uv}(i) := \max_{j\in\mathcal{X}} \Gamma_{uv}(i,j) \;<\; 1. \tag{3.12}$$

The pre-factor in our bounds involves the constant

$$K(\psi) := \; 4\,\frac{\sum_{(u\to v)\in\vec{\mathcal{E}}}\big(\max_{i\in\mathcal{X}} \overline{B}^0_{uv}(i)\big)}{\sum_{(u\to v)\in\vec{\mathcal{E}}}\big(\min_{i\in\mathcal{X}} \underline{B}^0_{uv}(i)\big)}. \tag{3.13}$$

With this notation, we have the following result:

Theorem 7 (**General graphs**)**.** *Suppose that the BP update function $F : \mathbb{R}^D \to \mathbb{R}^D$ satisfies the contraction condition (3.11).*

(a) Then BP has a unique fixed point m^*, and the SBP message sequence $\{m^t\}_{t=0}^{\infty}$, generated with the step size $\lambda^t = \mathcal{O}(1/t)$, converges almost surely to m^* as $t \to \infty$.

(b) With the step size $\lambda^t = \alpha/(\mu\cdot(t+2))$ for some fixed $1 < \alpha < 2$, we have

$$\frac{\mathbb{E}\big[\|m^t - m^*\|_2^2\big]}{\|m^*\|_2^2} \;\leq\; \frac{3^\alpha\, K(\psi)\,\alpha^2}{2^\alpha\,\mu^2(\alpha-1)}\left(\frac{1}{t}\right) + \frac{\|m^0 - m^*\|_2^2}{\|m^*\|_2^2}\left(\frac{2}{t}\right)^\alpha \tag{3.14}$$

for all iterations $t = 1, 2, \ldots$.

(c) With the step size $\lambda^t = 1/(\mu\cdot(t+1))$, we have

$$\frac{\mathbb{E}\big[\|m^t - m^*\|_2^2\big]}{\|m^*\|_2^2} \;\leq\; \frac{K(\psi)}{\mu^2}\left(\frac{1+\log t}{t}\right); \tag{3.15}$$

also for every $0 < \epsilon < 1$ and $t \geq 2$, we have

$$\frac{\|m^t - m^*\|_2^2}{\|m^*\|_2^2} \;\leq\; \frac{K(\psi)}{\mu^2}\left(1 + \frac{8}{\sqrt{\epsilon}}\right)\left(\frac{1+\log t}{t}\right) \tag{3.16}$$

with probability at least $1-\epsilon$.

[3] As will be discussed later, we can obtain a sequence of more refined (tighter) lower $\{\underline{B}^\ell_{uv}(i)\}_{\ell=0}^{\infty}$ and upper $\{\overline{B}^\ell_{uv}(i)\}_{\ell=0}^{\infty}$ bounds by confining the space of feasible messages.

Remarks: The proof of Theorem 7 is given in Section 3.5.2. Here we discuss some of the various guarantees that it provides. First, part (a) of the theorem shows that the SBP algorithm is strongly consistent, in that it converges almost surely to the unique BP fixed point. This claim is analogous to the almost sure convergence established in Theorem 6(a) for trees. Second, the bound (3.14) in Theorem 7(b) provides a non-asymptotic bound on the normalized mean-squared error $\mathbb{E}[\|m^t - m^*\|_2^2]/\|m^*\|_2^2$. For the specified choice of stepsize ($1 < \alpha < 2$), the first component of the bound (3.14) is dominant, hence the expected error (in squared ℓ_2-norm) is of the order $1/t$. Therefore, after $t = \Theta(1/\delta)$ iterations, the SBP algorithm returns a solution with MSE at most $\mathcal{O}(\delta)$. At least superficially, this rate might appear faster than the $1/\sqrt{t}$ rate established for trees in Theorem 6(b); however, the reader should be careful to note that Theorem 6 involves the element-wise ℓ_∞-norm, which is not squared, as opposed to the squared ℓ_2-norm studied in Theorem 7. Finally, part (c) provides bounds, both in expectation and with high probability, for a slightly different step size choice. On one hand, the bound in expectation (3.15) is of the order $\mathcal{O}(\log t/t)$, and so includes an additional logarithmic factor not present in the bounds from part (b). However, as shown in the high probability bound (3.16), the squared error is also guaranteed to satisfy a sample-wise version of the same bound with high probability. This theoretical claim is consistent with our later experimental results, showing that the error exhibits tight concentration around its expected behavior.

Let us now compare the guarantees of SBP to those of BP. Under the contraction condition of Theorem 7, the ordinary BP message updates are guaranteed to converge geometrically quickly, meaning that $\Theta(\log(1/\delta))$ iterations are sufficient to obtain δ-accurate solution. In contrast, under the same conditions, the SBP algorithm requires $\Theta(1/\delta)$ iterations to return a solution with MSE at most δ, so that its iteration complexity is larger. However, as noted earlier, the BP message updates require $\Theta(d^2)$ operations for each edge and iteration, whereas the SBP message updates require only $\Theta(d)$ operations. Putting the pieces together, we conclude that:

- on one hand, ordinary BP requires $\Theta\big(|\mathcal{E}|\, d^2\, \log(1/\delta)\big)$ operations to compute the fixed point to accuracy δ;
- in comparison, SBP requires $\Theta\big(|\mathcal{E}|\, d\, (1/\delta)\big)$ operations to compute the fixed point to expected accuracy δ.

Consequently, we see that as long the desired tolerance is not too small—in particular, if $\delta \geq 1/d$—then SBP leads to computational savings. In many practical applications, the state dimension is on the order of 10^3 to 10^5, so that the precision δ can be of the order 10^{-3} to 10^{-5} before the complexity of SBP becomes of comparable order to that of BP. Given that most graphical models represent approximations to reality, it is likely that larger tolerances δ are often of interest.

3.4.3 Sufficient Conditions for Contractivity

Theorem 7 is based on the assumption that the update function is contractive, meaning that its Lipschitz constant L is less than one. In past work, various authors have developed contractivity conditions, based on analyzing the log messages, that guarantee uniqueness and convergence of ordinary BP (e.g., [113, 49, 78, 103]). Our theorem requires contractivity on the messages (as opposed to log messages), which requires a related but slightly different argument. In this section, we show how to control L and thereby provide sufficient conditions for Theorem 7 to be applicable.

Our contractivity result applies when the messages under consideration belong to a set of the form

$$\mathcal{S} := \Big\{ m \in \mathbb{R}^D \;\Big|\; \sum_{i \in \mathcal{X}} m_{u\to v}(i) = 1,\ \underline{B}_{uv}(i) \le m_{u\to v}(i) \le \overline{B}_{uv}(i) \quad \forall (u \to v) \in \vec{\mathcal{E}},\ \forall i \in \mathcal{X} \Big\}, \tag{3.17}$$

for some choice of the upper and lower bounds—namely, $\overline{B}_{uv}(i)$ and $\underline{B}_{uv}(i)$ respectively. It turns out that the BP update function on the directed edge $(u \to v)$ is a convex combination of the normalized columns $\Gamma_{uv}(:, j)$ for $j = 1, \ldots, d$. Therefore, recalling the definition (3.12), we have $\underline{B}^0_{uv}(i) \le m_{uv}(i) \le \overline{B}^0_{uv}(i)$, for all $i = 1, \ldots, d$. Thus, for all iterations $t = 0, 1, \ldots$, the messages always belong to a set of the form (3.17) with $\underline{B}_{uv}(i) = \underline{B}^0_{uv}(i)$ and $\overline{B}_{uv}(i) = \overline{B}^0_{uv}(i)$. Since the bounds $(\underline{B}^0_{uv}(i), \overline{B}^0_{uv}(i))$ do not involve the node potentials, one suspects that they might be tightened at subsequent iterations, and indeed, there is a progressive refinement of upper and lower bounds of this form. Assuming that the messages belong to a set $\mathcal{S}$ at an initial iteration, then for any subsequent iterations, we are guaranteed the inclusion

$$m \in F(\mathcal{S}) := \big\{ F(m') \in \mathbb{R}^D \;\mid\; m' \in \mathcal{S} \big\}, \tag{3.18}$$

which then leads to the refined upper and lower bounds

$$\underline{B}^1_{uv}(i) := \inf_{m \in \mathcal{S}} \Big\{ \sum_{j=1}^{d} \Gamma_{uv}(i,j) \, \frac{\beta_{uv}(j)\, M_{u\to v}(j)}{\sum_{\ell=1}^{d} \beta_{uv}(\ell)\, M_{u\to v}(\ell)} \Big\}, \quad \text{and}$$

$$\overline{B}^1_{uv}(i) := \sup_{m \in \mathcal{S}} \Big\{ \sum_{j=1}^{d} \Gamma_{uv}(i,j) \, \frac{\beta_{uv}(j)\, M_{u\to v}(j)}{\sum_{\ell=1}^{d} \beta_{uv}(\ell)\, M_{u\to v}(\ell)} \Big\},$$

where we recall the quantity $M_{u\to v}(j) = \prod_{w \in \mathcal{N}(u)\backslash\{v\}} m_{w\to u}(j)$ previously defined (3.7). While such refinements are possible, in order to streamline our presentation, we focus primarily on the zero'th order bounds $\underline{B}_{uv}(i) = \underline{B}^0_{uv}(i)$ and $\overline{B}_{uv}(i) = \overline{B}^0_{uv}(i)$.

Given a set $\mathcal{S}$ of the form (3.17), we associate with the directed edges $(u \to v)$ and

$(w \to u)$ (where $w \in \mathcal{N}(u)\backslash\{v\}$) the non-negative numbers

$$\Phi_1(u \to v) \;:=\; \sum_{w\in\mathcal{N}(u)\backslash\{v\}} \Big(\phi_{u\to v,w\to u}\left(\phi_{u\to v,w\to u} + \chi_{u\to v,w\to u}\right)\Big)^{\frac{1}{2}}, \quad \text{and} \tag{3.19a}$$

$$\Phi_2(w \to u) \;:=\; \sum_{v\in\mathcal{N}(u)\backslash\{w\}} \Big(\phi_{u\to v,w\to u}\left(\phi_{u\to v,w\to u} + \chi_{u\to v,w\to u}\right)\Big)^{\frac{1}{2}}, \tag{3.19b}$$

where

$$\phi_{u\to v,w\to u} := \max_{j\in\mathcal{X}} \sup_{m\in\mathcal{S}} \left\{\frac{\beta_{uv}(j)\, M_{u\to v}(j)}{\sum_{k=1}^{d}\beta_{uv}(k)\, M_{u\to v}(k)}\, \frac{1}{m_{w\to u}(j)}\right\}, \quad \text{and} \tag{3.20a}$$

$$\chi_{u\to v,w\to u} := \max_{j\in\mathcal{X}} \sup_{m\in\mathcal{S}} \left\{\frac{\beta_{uv}(i)\, M_{u\to v}(i)}{\left(\sum_{k=1}^{d}\beta_{uv}(k)\, M_{u\to v}(k)\right)^2} \sum_{j=1}^{d} \frac{\beta_{uv}(j)\, M_{u\to v}(j)}{m_{w\to u}(j)}\right\}. \tag{3.20b}$$

Recall the normalized compatibility matrix $\Gamma_{uv} \in \mathbb{R}^{d\times d}$ on the edge (u, v), as previously defined in (3.6). Since Γ_{uv} is a stochastic matrix with positive entries, the Perron-Frobenius theorem [47] guarantees that the maximal eigenvalue is equal to one, and is associated with a pair of left and right eigenvectors (unique up to scaling) with positive entries. Since Γ_{uv} is column-stochastic, any multiple of the all-one vector $\vec{1}$ can be chosen as the left eigenvector. Letting $z_{uv} \in \mathbb{R}^d$ denote the right eigenvector with positive entries, we are guaranteed that $\vec{1}^T z_{uv} > 0$, and hence we may define the matrix $\Gamma_{uv} - z_{uv}\vec{1}^T/(\vec{1}^T z_{uv})$. By construction, this matrix has all of its eigenvalues strictly less than 1 in absolute value (Lemma 8.2.7, [47]).

Proposition 1. *The global update function $F : \mathbb{R}^D \to \mathbb{R}^D$ defined in (3.4) is Lipschitz with constant at most*

$$L \;:=\; 2 \max_{(u,v)\in\mathcal{E}} |\!|\!|\Gamma_{uv} - \frac{z_{uv}\vec{1}^T}{\vec{1}^T z_{uv}}|\!|\!|_2 \max_{(u\to v)\in\vec{\mathcal{E}}} \Phi_1(u \to v) \max_{(w\to u)\in\vec{\mathcal{E}}} \Phi_2(w \to u), \tag{3.21}$$

where $|\!|\!| \cdot |\!|\!|_2$ denotes the maximum singular value of a matrix.

In order to provide some intuition for Proposition 1, let us consider a simple but illuminating example.

Example 5 (Potts model). The Potts model [38, 112, 60] is often used for denoising, segmentation, and stereo computation in image processing and computer vision. It is a pairwise Markov random field that is based on edge potentials of the form

$$\psi_{vu}(i, j) = \begin{cases} 1 & \text{if } i = j, \text{ and} \\ \gamma & \text{if } i \neq j. \end{cases},$$

for all edges $(u, v) \in \mathcal{E}$ and $i, j \in \{1, 2, \ldots, d\}$. The parameter $\gamma \in (0, 1]$ can be tuned to enforce different degrees of smoothness: at one extreme, setting $\gamma = 1$ enforces no smoothness, whereas a choice close to zero enforces a very strong type of smoothness. (To be clear, the special structure of the potts model can be exploited to compute the BP message updates quickly; our motivation in considering it here is only to provide a simple illustration of our contractivity condition.)

For the Potts model, we have $\beta_{uv}(j) = \psi_u(j)\,(1+(d-1)\gamma)$, and hence Γ_{uv} is a symmetric matrix with

$$\Gamma_{uv}(i,j) \;=\; \begin{cases} \frac{1}{1+(d-1)\gamma} & \text{if } i=j \\ \frac{\gamma}{1+(d-1)\gamma} & \text{if } i\neq j. \end{cases}$$

Some straightforward algebra shows that the second largest singular value of Γ_{uv} is given by $(1-\gamma)/(1+(d-1)\gamma)$, whence

$$\max_{(u,v)\in\mathcal{E}} |\!|\!|\Gamma_{uv} - \frac{z_{uv}\vec{1}^T}{\vec{1}^T z_{uv}}|\!|\!|_2 \;=\; \frac{1-\gamma}{1+(d-1)\gamma}.$$

The next step is to find upper bounds on the terms $\Phi_1(u \to v)$ and $\Phi_2(w \to u)$, in particular by upper bounding the quantities $\phi_{u\to v, w\to u}$ and $\chi_{u\to v, w\to u}$, as defined in equations (3.20a) and (3.20b) respectively. In Appendix A.1, we show that the Lipschitz function of F is upper bounded as

$$L \leq 4\,(1-\gamma)\,(1+(d-1)\gamma)\;\max_{u\in\mathcal{V}}\left\{\frac{(\rho_u-1)^2}{\gamma^{2\rho_u}}\;\max_{j\in\mathcal{X}}\left\{\frac{\psi_u(j)}{\sum_{\ell=1}^{d}\psi_u(\ell)}\right\}^2\right\},$$

where ρ_u is the degree of node u. Therefore, a sufficient condition for contractivity in the case of the Potts model is

$$\max_{u\in\mathcal{V}}\left\{\frac{(\rho_u-1)}{\gamma^{\rho_u}}\;\max_{j\in\mathcal{X}}\left\{\frac{\psi_u(j)}{\sum_{\ell=1}^{d}\psi_u(\ell)}\right\}\right\} \;<\; \left(\frac{1}{4\,(1-\gamma)\,(1+(d-1)\gamma)}\right)^{\frac{1}{2}}. \tag{3.22}$$

To gain intuition, consider the special case in which the node potentials are uniform, so that $\psi_u(j)/(\sum_{\ell=1}^{d}\psi_u(\ell)) = 1/d$ for $j=1,2,\ldots,d$. In this case, for any graph with bounded node degrees, the bound (3.22) guarantees contraction for all γ in an interval $[\epsilon, 1]$. For non-uniform node potentials, the inequality (3.22) is weaker, but it can be improved via the refined sets (3.18) discussed previously.

3.5 Proof of the Main Results

We now turn to the proofs of our two main results, namely Theorems 6 and 7, as well as the auxiliary result, Proposition 1, on contractivity of the BP message updates. For our purposes, it is convenient to note that the ordinary BP update can be written as an expectation of the form

$$F_{u\to v}(m^t) \;=\; \mathbb{E}\,[\Gamma_{uv}(:, J^{t+1}_{u\to v})], \tag{3.23}$$

for all $t = 0, 1, \ldots$. Here the expectation is taken place over the randomness induced by $J^{t+1}_{u\to v}$ chosen randomly according to the probability mass function $p^t_{u\to v}$ (3.8).

3.5.1 Proof of Theorem 6

We begin by stating a lemma that plays a central role in the proof of Theorem 6.

Lemma 1. *For any tree-structured Markov random field, there exists a nilpotent matrix $A \in \mathbb{R}^{D\times D}$ of degree at most $r = \mathrm{diam}(\mathcal{G})$ such that*

$$|F(m) - F(m')| \;\preceq\; A\,|m - m'|, \tag{3.24}$$

for all $m, m' \in \mathcal{S}$.

The proof of this lemma is somewhat technical, so that we defer it to Appendix A.2. In interpreting this result, the reader should recall that for vectors $x, y \in \mathbb{R}^D$, the notation $x \preceq y$ denotes inequality in an elementwise sense—i.e., $x(i) \leq y(i)$ for $i = 1, \ldots, D$.

An immediate corollary of this lemma is the existence and uniqueness of the BP fixed point. Since we may iterate inequality (3.24), we find that

$$|F^{(\ell)}(m) - F^{(\ell)}(m')| \preceq \; A^{\ell}\,|m - m'|,$$

for all iterations $\ell = 1, 2, \ldots$, and arbitrary messages m, m', where $F^{(\ell)}$ denotes the composition of F with itself ℓ times. The nilpotence of A ensures that $A^r = 0$, and hence $F^{(r)}(m) = F^{(r)}(m')$ for all messages m, and m'. Let $m^* = F^{(r)}(m)$ denote the common value. The claim is that m^* is the unique fixed point of the BP update function F. This can be shown as follows: from Lemma 1 we have

$$|F(m^*) \;-\; m^*| \;=\; |F^{(r+1)}(m) \;-\; F^{(r)}(m)| \;\preceq\; A\,|F^{(r)}(m) \;-\; F^{(r-1)}(m)|.$$

Iterating the last inequality for the total of r times, we obtain

$$|F(m^*) \;-\; m^*| \;\preceq\; A^r\,|F(m) \;-\; m| \;=\; 0,$$

and hence $F(m^*) = m^*$. On the other hand, the uniqueness of the BP fixed point is a direct consequence of the facts that for any fixed point m^* we have $F^{(r)}(m^*) = m^*$, and for all arbitrary messages m, m' we have $F^{(r)}(m) = F^{(r)}(m')$. Accordingly, we see that Lemma 1 provides an alternative proof of the well-known fact that BP converges to a unique fixed point on trees after at most $r = \mathrm{diam}(\mathcal{G})$ iterations.

We now show how Lemma 1 can be used to establish the two claims of Theorem 6.

Part (a): Almost Sure Consistency

We begin with the almost sure consistency claim of part (a). By combining all the local updates, we form the global update rule

$$m^{t+1} \; = \; (1-\lambda^t)\, m^t + \lambda^t\, \nu^{t+1} \quad \text{for iterations } t = 0, 1, 2, \ldots, \tag{3.25}$$

where $\nu^{t+1} := \{\Gamma_{uv}(:, J^{t+1}_{u \to v})\}_{(u \to v) \in \vec{\mathcal{E}}}$ is the D-dimensional vector obtained from stacking up all the normalized columns $\Gamma_{uv}(:, J^{t+1}_{u \to v})$. Defining the vector $Y^{t+1} := \nu^{t+1} - F(m^t) \in \mathbb{R}^D$, we can rewrite the update equation (3.25) as

$$m^{t+1} \; = \; (1-\lambda^t)\, m^t + \lambda^t\, F(m^t) + \lambda^t\, Y^{t+1} \quad \text{for } t = 0, 1, 2, \ldots. \tag{3.26}$$

With our step size choice $\lambda^t = 1/(t+1)$, unwrapping the recursion (3.26) yields the representation

$$m^t \; = \; \frac{1}{t} \sum_{\ell=0}^{t-1} F(m^\ell) \; + \; \frac{1}{t} \sum_{\ell=1}^{t} Y^\ell.$$

Subtracting the unique fixed point m^* from both sides then leads to

$$m^t - m^* \; = \; \frac{1}{t} \sum_{\ell=1}^{t-1} (F(m^\ell) - F(m^*)) \; + \; \underbrace{\frac{1}{t} \sum_{\ell=1}^{t} Y^\ell + \frac{1}{t}\,(F(m^0) - F(m^*))}_{Z^t}, \tag{3.27}$$

where we have introduced the convenient shorthand Z^t. We may apply the triangle inequality to each element of this vector equation; doing so and using Lemma 1 to upper bound the terms $|F(m^\ell) - F(m^*)|$, we obtain the element-wise inequality

$$|m^t - m^*| \; \preceq \; \frac{1}{t} \sum_{\ell=1}^{t-1} A\, |m^\ell - m^*| \; + \; |Z^t| \quad \text{for } t = 1, 2, \ldots.$$

Since A^r is the all-zero matrix, unwrapping the last inequality $r = \operatorname{diam}(\mathcal{G})$ times yields the element-wise upper bound

$$|m^t - m^*| \; \preceq \; G^t_0 \; + \; A G^t_1 + A^2 G^t_2 + \cdots + A^{r-1} G^t_{r-1}, \tag{3.28}$$

where the terms G^t_ℓ are defined via the recursion $G^t_\ell := \big(\sum_{j=1}^{t-1} G^j_{\ell-1}\big)/t$ for $\ell = 1, \ldots, r-1$, with initial conditions $G^t_0 := |Z^t|$.

It remains to control the sequences $\{G^t_\ell\}_{t=1}^\infty$ for $\ell = 0, 1, \ldots, r-1$. In order to do so, we first establish a martingale difference property for the variables Y^t defined prior to (3.26).

For each $t = 0, 1, 2, \ldots$, define the σ-field $\mathcal{F}^t := \sigma(m^0, m^1, \ldots, m^t)$, as generated by the randomness in the messages up to time t. Based on the representation (3.23), we see that $\mathbb{E}\big[Y^{t+1} | \mathcal{F}^t\big] = \vec{0}$, showing that $\{Y^{t+1}\}_{t=0}^{\infty}$ forms martingale difference sequence with respect to the filtration $\{\mathcal{F}^t\}_{t=0}^{\infty}$. From the definition, it can be seen that the entries of Y^{t+1} are bounded; more precisely, we have $|Y^{t+1}(i)| \leq 1$ for all iterations $t = 0, 1, 2, \ldots$, and all states $i = 1, 2, \ldots D$. Consequently, the sequence $\{Y^\ell\}_{\ell=1}^{\infty}$ is a bounded martingale difference sequence.

We begin with the term G_0^t. Since Y^ℓ is a bounded martingale difference, standard convergence results [34] guarantee that $|\sum_{\ell=1}^{t} Y^\ell|/t \to \vec{0}$ almost surely. Moreover, we have the bound $|F(m^0) - F(m^*)|/t \preceq \vec{1}/t$. Recalling the definition of Z^t from (3.27), we conclude that $G_0^t = |Z^t|$ converges to the all-zero vector almost surely as $t \to \infty$. In order to extend our argument to the terms G_ℓ^t for $\ell = 1, \ldots, r-1$, we make use of the following fact: for any sequence of real numbers $\{x^t\}_{t=0}^{\infty}$ such that $x^t \to 0$, we also have $(\sum_{\ell=0}^{t-1} x^\ell)/t \to 0$ (e.g., see Royden [104]). Consequently, for any realization ω such that the deterministic sequence $\{G_0^t(\omega)\}_{t=0}^{\infty}$ converges to zero, we are also guaranteed that the sequence $\{G_1^t(\omega)\}_{t=0}^{\infty}$, with elements $G_1^t(\omega) = (\sum_{j=1}^{t-1} G_0^j(\omega))/t$, converges to zero. Since we have shown that $G_0^t \overset{\text{a.s.}}{\to} 0$, we conclude that $G_1^t \overset{\text{a.s.}}{\to} 0$ as well. This argument can be iterated, thereby establishing almost sure convergence for all of the terms G_ℓ^t. Putting the pieces together, we conclude that the vector $|m^t - m^*|$ converges almost surely to the all-zero vector as $t \to \infty$, thereby completing the proof of part (a).

Part (b): Bounds on Expected Absolute Error

We now turn to part (b) of Theorem 6, which provides upper bounds on the expected absolute error. We establish this claim by exploiting some martingale concentration inequalities [24]. From part (a), we know that $\{Y^t\}_{t=1}^{\infty}$ is a bounded martingale difference sequence, in particular with $|Y^t(i)| \leq 1$. Applying the Azuma-Hoeffding inequality [24] yields the tail bound

$$\mathbb{P}\Big(\frac{1}{t}\,|\sum_{\ell=1}^{t} Y^\ell(i)| > \gamma\Big) \;\leq\; 2\exp\Big(-\frac{t\,\gamma^2}{2}\Big),$$

for all $\gamma > 0$, and $i = 1, 2, \ldots, D$. By integrating this tail bound, we can upper bound the mean: in particular, we have

$$\mathbb{E}\Big[\frac{1}{t}|\sum_{\ell=1}^{t} Y^\ell(i)|\Big] \;=\; \int_0^\infty \mathbb{P}\Big(\frac{1}{t}|\sum_{\ell=1}^{t} Y^\ell(i)| > \gamma\Big)\, d\gamma \;\leq\; \sqrt{\frac{2\pi}{t}},$$

and hence

$$\mathbb{E}\big[G_0^t\big] \;=\; \mathbb{E}\big[|Z^t|\big] \;\preceq\; \sqrt{\frac{2\pi}{t}}\,\vec{1} + \frac{\vec{1}}{t} \;\preceq\; \frac{4}{\sqrt{t}}\,\vec{1}. \tag{3.29}$$

Turning to the term G_1^t, we have

$$\mathbb{E}[G_1^t] \;=\frac{1}{t}\sum_{\ell=1}^{t-1}\mathbb{E}[G_0^\ell] \;\overset{\text{(i)}}{\preceq}\; \frac{1}{t}\sum_{\ell=1}^{t-1}\frac{4}{\sqrt{\ell}}\,\vec{1} \;\overset{\text{(ii)}}{\preceq}\; \frac{2\cdot 4}{\sqrt{t}}\,\vec{1},$$

where step (i) uses the inequality (3.29), and step (ii) is based on the elementary upper bound $\sum_{\ell=1}^{t-1} 1/\sqrt{\ell} \;\le\; 1 + \int_1^{t-1} 1/\sqrt{x}\,dx \;<\; 2\sqrt{t}$. By repeating this same argument in a recursive manner, we conclude that $\mathbb{E}\big[G_\ell^t\big] \preceq (2^\ell \cdot 4/\sqrt{t})\,\vec{1}$ for $\ell = 2, 3, \ldots, r-1$. Taking the expectation on both sides of the the inequality (3.28) and substituting these upper bounds, we obtain

$$\mathbb{E}\big[|m^t - m^*|\big] \;\preceq\; 4\left(\sum_{\ell=0}^{r-1} 2^\ell A^\ell\right)\frac{\vec{1}}{\sqrt{t}} \;=\; 4\,(I-2A)^{-1}\,\frac{\vec{1}}{\sqrt{t}},$$

where we have used the fact that $A^r = 0$.

3.5.2 Proof of Theorem 7

We now turn to the proof of Theorem 7. Note that since the update function is contractive, the existence and uniqueness of the BP fixed point is an immediate consequence of the Banach fixed-point theorem [3].

Part (a): Almost Sure Consistency

We establish part (a) by applying the Robbins-Monro theorem, a classical result from stochastic approximation theory (see Theorem 4 from Section 2.3). In order to do so, we begin by writing the update (3.9) in the form

$$m_{u\to v}^{t+1} \;=\; m_{u\to v}^t \;-\; \lambda^t\,\underbrace{\big[m_{u\to v}^t \;-\; \Gamma_{uv}(:, J_{u\to v}^{t+1})\big]}_{H_{uv}(m_{u\to v}^t, J_{u\to v}^{t+1})},$$

where for any realization $\bar{J}_{u\to v} \in \{1, 2, \ldots, d\}$, the mapping $m_{u\to v} \mapsto H_{uv}(m_{u\to v}, \bar{J}_{u\to v})$ should be understood as a function from $\mathbb{R}^d$ to $\mathbb{R}^d$. By concatenating together all of these mappings, one for each directed edge $(u \to v)$, we obtain a family of mappings $H(\cdot, \bar{J})$ from $\mathbb{R}^D$ to $\mathbb{R}^D$, one for each realization $\bar{J} \in \{1, 2, \ldots, d\}^{2|\vec{\mathcal{E}}|}$ of column indices.

With this notation, we can write the message update of the SBP algorithm in the compact form

$$m^{t+1} \;=\; m^t \;-\; \lambda^t\, H(m^t, J^{t+1}), \tag{3.30}$$

valid for for $t = 1, 2, \ldots$, and suitable for application of the Robbins-Monro theorem. In order to apply this result, we need to verify its hypotheses. First of all, it is easy to see that we have a bound of the form

$$\mathbb{E}\big[\|H(m, J)\|_2^2\big] \;\leq\; c\,(1 + \|m\|_2^2),$$

for some constant c. Moreover, the conditional distribution of the vector J^{t+1}, given the past, depends only on m^t; more precisely we have

$$\mathbb{P}\big(J^{t+1} | J^t, J^{t-1}, \ldots, m^t, m^{t-1}, \ldots\big) \;=\; \mathbb{P}\big(J^{t+1} | m^t\big).$$

Lastly, defining the averaged function $h(m) := \mathbb{E}\big[H(m, J) | m\big] \;=\; m - F(m)$, the final requirement is to verify that the fixed point m^* satisfies the stability condition

$$\inf_{m \in \mathcal{S} \setminus \{m^*\}} \langle m - m^*,\, h(m) \rangle \;>\; 0, \tag{3.31}$$

where $\langle \cdot, \cdot \rangle$ denotes the Euclidean inner product, and $\mathcal{S}$ denotes the compact set in which the messages lie. Using the Cauchy-Schwartz inequality and the fact that F is Lipschitz with constant $L = 1 - \mu/2$, we obtain

$$\begin{aligned} \langle m - m^*,\, h(m) - h(m^*) \rangle \;&= \|m - m^*\|_2^2 \;-\; \langle m - m^*,\, F(m) - F(m^*) \rangle \\ &\geq \frac{\mu}{2}\, \|m - m^*\|_2^2 \;>\; 0, \end{aligned} \tag{3.32}$$

where the strict inequality holds for all $m \neq m^*$. Since m^* is a fixed point, we must have $h(m^*) = m^* - F(m^*) = 0$, which concludes the proof.

Part (b): Non-Asymptotic Bounds on Normalized Mean-Squared Error

Let $e^t := (m^t - m^*)/\|m^*\|_2$ denote the re-normalized error vector. In order to upper bound $\mathbb{E}\big[\|e^t\|_2^2\big]$ for all $t = 1, 2, \ldots$, we first control the quantity $\|e^{t+1}\|_2^2 - \|e^t\|_2^2$, corresponding to the increment in the squared error. Doing some simple algebra yields

$$\begin{aligned} \|e^{t+1}\|_2^2 - \|e^t\|_2^2 \;&=\; \frac{1}{\|m^*\|_2^2}\,\big(\|m^{t+1} - m^*\|_2^2 - \|m^t - m^*\|_2^2\big) \\ &=\; \frac{1}{\|m^*\|_2^2}\,\langle m^{t+1} - m^t,\, m^{t+1} + m^t - 2m^* \rangle. \end{aligned}$$

Recalling the update equation (3.30), we obtain

$$\begin{aligned} \|e^{t+1}\|_2^2 - \|e^t\|_2^2 \;&=\; \frac{1}{\|m^*\|_2^2}\,\langle -\lambda^t H(m^t, J^{t+1}),\, -\lambda^t H(m^t, J^{t+1}) + 2(m^t - m^*) \rangle \\ &=\; \frac{(\lambda^t)^2}{\|m^*\|_2^2}\,\|H(m^t, J^{t+1})\|_2^2 \;-\; \frac{2\lambda^t}{\|m^*\|_2^2}\,\langle H(m^t, J^{t+1}),\, m^t - m^* \rangle. \end{aligned} \tag{3.33}$$

Now taking the expectation on both sides of (3.33) yields

$$\begin{aligned}\mathbb{E}[\|e^{t+1}\|_2^2] - \mathbb{E}[\|e^t\|_2^2] &= \frac{(\lambda^t)^2}{\|m^*\|_2^2}\,\mathbb{E}\big[\|H(m^t, J^{t+1})\|_2^2\big] - \frac{2\lambda^t}{\|m^*\|_2^2}\,\mathbb{E}\big[\mathbb{E}\big[\langle H(m^t, J^{t+1}),\, m^t - m^*\rangle|\mathcal{F}^t\big]\big] \\ &= \frac{(\lambda^t)^2}{\|m^*\|_2^2}\,\mathbb{E}\big[\|H(m^t, J^{t+1})\|_2^2\big] - \frac{2\lambda^t}{\|m^*\|_2^2}\,\mathbb{E}\big[\langle h(m^t) - h(m^*),\, m^t - m^*\rangle\big],\end{aligned} \tag{3.34}$$

where we used the facts that $\mathbb{E}[H(m^t, J^{t+1})|\mathcal{F}^t] = h(m^t)$ and $h(m^*) = 0$. We continue by upper bounding the term $G_1 = \|H(m^t, J^{t+1})\|_2^2/\|m^*\|_2^2$ and lower bounding the term $G_2 = \langle h(m^t) - h(m^*),\, m^t - m^*\rangle/\|m^*\|_2^2$.

Lower bound on G_2: Recalling (3.32) from our proof of part (a), we see that

$$G_2 \;\geq\; \frac{\mu}{2}\,\|e^t\|_2^2. \tag{3.35}$$

Upper bound on G_1: From the definition of the update function, we have

$$\|H(m^t, J^{t+1})\|_2^2 \;=\; \sum_{(u\to v)\in\vec{\mathcal{E}}} \|m^t_{u\to v} - \Gamma_{uv}(:, J^t_{u\to v})\|_2^2 \;\leq\; 2\sum_{(u\to v)\in\vec{\mathcal{E}}} \big(\|m^t_{u\to v}\|_2^2 + \|\Gamma_{uv}(:, J^t_{u\to v})\|_2^2\big).$$

Recalling the bounds (3.12) and using the fact that vectors $m^t_{u\to v}$ and $\Gamma_{uv}(:, J^t_{u\to v})$ sum to one, we obtain

$$\begin{aligned}\|H(m^t, J^{t+1})\|_2^2 &\leq 2\sum_{(u\to v)\in\vec{\mathcal{E}}} \big(\max_{i\in\mathcal{X}} \overline{B}^0_{uv}(i)\big)\big(\|m^t_{u\to v}\|_1 + \|\Gamma_{uv}(:, J^t_{u\to v})\|_1\big) \\ &= 4\sum_{(u\to v)\in\vec{\mathcal{E}}} \big(\max_{i\in\mathcal{X}} \overline{B}^0_{uv}(i)\big).\end{aligned}$$

On the other hand, we also have

$$\|m^*\|_2^2 \;\geq\; \sum_{(u\to v)\in\vec{\mathcal{E}}} \big(\min_{i\in\mathcal{X}} \underline{B}^0_{uv}(i)\big)\,\|m^*_{uv}\|_1 \;=\; \sum_{(u\to v)\in\vec{\mathcal{E}}} \big(\min_{i\in\mathcal{X}} \underline{B}^0_{uv}(i)\big).$$

Combining the pieces, we conclude that the term G_1 is upper bounded as

$$G_1 \;\leq\; K(\psi) \;:=\; 4\,\frac{\sum_{(u\to v)\in\vec{\mathcal{E}}}\big(\max_{i\in\mathcal{X}} \overline{B}^0_{uv}(i)\big)}{\sum_{(u\to v)\in\vec{\mathcal{E}}}\big(\min_{i\in\mathcal{X}} \underline{B}^0_{uv}(i)\big)}. \tag{3.36}$$

Since both G_1 and G_2 are non-negative, the bounds (3.36) and (3.35) also hold in expectation. Combining these bounds with the representation (3.34), we obtain the upper bound

$$\mathbb{E}[\|e^{t+1}\|_2^2] - \mathbb{E}[\|e^t\|_2^2] \;\leq\; K(\psi)\,(\lambda^t)^2 - \lambda^t\mu\,\mathbb{E}[\|e^t\|_2^2],$$

or equivalently

$$\mathbb{E}[\|e^{t+1}\|_2^2] \;\leq\; K(\psi)\,(\lambda^t)^2 \;+\; (1-\lambda^t\mu)\,\mathbb{E}[\|e^t\|_2^2].$$

Setting $\lambda^t = \alpha/(\mu(t+2))$ and unwrapping this recursion yields

$$\mathbb{E}[\|e^{t+1}\|_2^2] \;\leq\; \frac{K(\psi)\,\alpha^2}{\mu^2} \sum_{i=2}^{t+2} \left(\frac{1}{i^2} \prod_{\ell=i+1}^{t+2} \Big(1-\frac{\alpha}{\ell}\Big)\right) \;+\; \prod_{\ell=2}^{t+2} \Big(1-\frac{\alpha}{\ell}\Big)\,\mathbb{E}[\|e^0\|_2^2], \tag{3.37}$$

where we have adopted the convention that the inside product is equal to one for $i = t+2$. The following lemma, proved in Appendix A.3, provides a useful upper bound on the products arising in this expression:

Lemma 2. *For all $i \in \{1, 2, \ldots, t+1\}$, we have*

$$\prod_{\ell=i+1}^{t+2} \Big(1-\frac{\alpha}{\ell}\Big) \;\leq\; \left(\frac{i+1}{t+3}\right)^{\alpha}.$$

Substituting this upper bound into the inequality (3.37) yields

$$\begin{aligned}\mathbb{E}[\|e^{t+1}\|_2^2] \;&\leq\; \frac{K(\psi)\,\alpha^2}{\mu^2(t+3)^\alpha} \sum_{i=2}^{t+2} \frac{(i+1)^\alpha}{i^2} \;+\; \left(\frac{2}{t+3}\right)^{\alpha} \mathbb{E}[\|e^0\|_2^2] \\ &\leq\; \frac{K(\psi)\,\alpha^2}{\mu^2(t+3)^\alpha} \big(\tfrac{3}{2}\big)^{\alpha} \sum_{i=2}^{t+2} \frac{1}{i^{2-\alpha}} \;+\; \left(\frac{2}{t+3}\right)^{\alpha} \mathbb{E}[\|e^0\|_2^2].\end{aligned}$$

It remains to upper bound the term $\sum_{i=2}^{t+2} 1/i^{2-\alpha}$. Since the function $1/x^{2-\alpha}$ is decreasing in x for $\alpha < 2$, we have the integral upper bound $\sum_{i=2}^{t+2} 1/i^{2-\alpha} \;\leq\; \int_1^{t+2} 1/x^{2-\alpha}\,dx$, which yields

$$\mathbb{E}[\|e^{t+1}\|_2^2] \;\leq\; \begin{cases} \big(\frac{3}{2}\big)^{\alpha} \frac{K(\psi)\,\alpha^2}{\mu^2(1-\alpha)}\, \frac{1}{(t+3)^\alpha} \;+\; \big(\frac{2}{t+3}\big)^{\alpha} \mathbb{E}[\|e^0\|_2^2] & \text{if } 0 < \alpha < 1 \\ \frac{3}{2}\, \frac{K(\psi)}{\mu^2}\, \frac{\log(t+2)}{t+3} \;+\; \frac{2}{t+3}\, \mathbb{E}[\|e^0\|_2^2] & \text{if } \alpha = 1 \\ \big(\frac{3}{2}\big)^{\alpha} \frac{K(\psi)\,\alpha^2}{\mu^2(\alpha-1)}\, \frac{(t+2)^{\alpha-1}}{(t+3)^\alpha} \;+\; \big(\frac{2}{t+3}\big)^{\alpha} \mathbb{E}[\|e^0\|_2^2] & \text{if } 1 < \alpha < 2 \end{cases}.$$

If we now focus on the range of $\alpha \in (1, 2)$, which yields the fastest convergence rate, some simple algebra yields the form of the claim given in the theorem statement.

Part (c): High Probability Bounds

Recall the algebra at the beginning of Section 3.5.2. Adding and subtracting the conditional mean of the second term of (3.33) yields

$$\|e^{t+1}\|_2^2 - \|e^t\|_2^2 \;=\; \frac{(\lambda^t)^2}{\|m^*\|_2^2}\,\|H(m^t, J^{t+1})\|_2^2 \;-\; \frac{2\lambda^t}{\|m^*\|_2^2}\,\langle h(m^t),\, m^t - m^*\rangle + 2\lambda^t\,\langle Y^{t+1},\, e^t\rangle,$$

where we have denoted the term

$$Y^{t+1} \;:=\; \frac{h(m^t) - H(m^t, J^{t+1})}{\|m^*\|_2}.$$

Recalling the bounds on $G_1 = \|H(m^t, J^{t+1})\|_2^2 \,/\, \|m^*\|_2^2$ and $G_2 = \langle h(m^t),\, m^t - m^* \rangle \,/\, \|m^*\|_2^2$ from part (b), we have

$$\|e^{t+1}\|_2^2 - \|e^t\|_2^2 \;\leq\; K(\psi)\,(\lambda^t)^2 \;-\; \mu\lambda^t\|e^t\|_2^2 \;+\; 2\lambda^t\,\langle Y^{t+1},\, e^t\rangle,$$

or equivalently

$$\|e^{t+1}\|_2^2 \;\leq\; K(\psi)\,(\lambda^t)^2 \;+\; (1-\mu\lambda^t)\|e^t\|_2^2 \;+\; 2\lambda^t\,\langle Y^{t+1},\, e^t\rangle.$$

Substituting the step size choice $\lambda^t = 1/(\mu(t+1))$ and then unwrapping this recursion yields

$$\begin{aligned} \|e^{t+1}\|_2^2 \;&\leq\; \frac{K(\psi)}{\mu^2(t+1)}\sum_{\tau=1}^{t+1}\frac{1}{\tau} \;+\; \frac{2}{\mu\,(t+1)}\sum_{\tau=0}^{t}\langle Y^{\tau+1},\, e^\tau\rangle \\ &\leq\; \frac{K(\psi)}{\mu^2}\frac{1+\log(t+1)}{t+1} \;+\; \frac{2}{\mu\,(t+1)}\sum_{\tau=0}^{t}\langle Y^{\tau+1},\, e^\tau\rangle. \end{aligned} \tag{3.38}$$

Note that by construction, the sequence $\{Y^\tau\}_{\tau=1}^\infty$ is a martingale difference sequence with respect to the filtration $\mathcal{F}^\tau = \sigma(m^0, m^1, \ldots, m^\tau)$ that is $\mathbb{E}\big[Y^{\tau+1} \mid \mathcal{F}^\tau\big] = \vec{0}$ and accordingly $\mathbb{E}\big[\langle Y^{\tau+1},\, e^\tau\rangle\big] = 0$ for $\tau = 0, 1, 2, \ldots$. We continue by controlling the stochastic term $(\sum_{\tau=0}^{t}\langle Y^{\tau+1},\, e^\tau\rangle)/(t+1)$—namely its variance,

$$\begin{aligned} \mathrm{var}\left(\frac{1}{t+1}\sum_{\tau=0}^{t}\langle Y^{\tau+1},\, e^\tau\rangle\right) \;&= \frac{1}{(t+1)^2}\,\mathbb{E}\Big[\big(\sum_{\tau=0}^{t}\langle Y^{\tau+1},\, e^\tau\rangle\big)^2\Big] \\ &= \underbrace{\frac{1}{(t+1)^2}\sum_{\tau=0}^{t}\mathbb{E}\big[\langle Y^{\tau+1},\, e^\tau\rangle^2\big]}_{T_1} \\ &\quad + \underbrace{\frac{2}{(t+1)^2}\sum_{0\leq\tau_2<\tau_1\leq t}\mathbb{E}\big[\langle Y^{\tau_1+1},\, e^{\tau_1}\rangle\langle Y^{\tau_2+1},\, e^{\tau_2}\rangle\big]}_{T_2}. \end{aligned}$$

Since we have

$$\begin{aligned} \mathbb{E}\big[\langle Y^{\tau_1+1},\, e^{\tau_1}\rangle\langle Y^{\tau_2+1},\, e^{\tau_2}\rangle\big] \;&= \mathbb{E}\big[\mathbb{E}\big[\langle Y^{\tau_1+1},\, e^{\tau_1}\rangle\langle Y^{\tau_2+1},\, e^{\tau_2}\rangle \mid \mathcal{F}^{\tau_1}\big]\big] \\ &= \mathbb{E}\big[\langle Y^{\tau_2+1},\, e^{\tau_2}\rangle\,\mathbb{E}\big[\langle Y^{\tau_1+1},\, e^{\tau_1}\rangle \mid \mathcal{F}^{\tau_1}\big]\big] \;=\; 0, \end{aligned}$$

for all $\tau_1 > \tau_2$, the cross product term T_2 vanishes. On the other hand, the martingale difference sequence is bounded. This can be shown as follows: from part (b) we know $\|H(m^\tau, J^{\tau+1})\|_2/\|m^*\|_2 \leq \sqrt{K(\psi)}$; also using the fact that $\|\cdot\|_2$ is convex, Jensen's inequality yields $\|h(m^\tau)\|_2/\|m^*\|_2 \leq \sqrt{K(\psi)}$; therefore, we have

$$\|Y^{\tau+1}\|_2 \;\leq\; \frac{\|H(m^\tau, J^{\tau+1})\|_2}{\|m^*\|_2} + \frac{\|h(m^\tau)\|_2}{\|m^*\|_2} \;\leq\; 2\sqrt{K(\psi)}.$$

Moving on to the first term T_1, we exploit the Cauchy Schwartz inequality in conjunction with the fact that the martingale difference sequence is bounded to obtain

$$\mathbb{E}\big[\langle Y^{\tau+1}, e^\tau\rangle^2\big] \;\leq\; \mathbb{E}\big[\|Y^{\tau+1}\|_2^2\,\|e^\tau\|_2^2\big] \;\leq\; 4\,K(\psi)\,\mathbb{E}\big[\|e^\tau\|_2^2\big].$$

Taking the expectation on both sides of the inequality (3.38) yields

$$\mathbb{E}\big[\|e^\tau\|_2^2\big] \;\leq\; \frac{K(\psi)}{\mu^2}\,\frac{1+\log\tau}{\tau};$$

and hence we have

$$\mathbb{E}\big[\langle Y^{\tau+1}, e^\tau\rangle^2\big] \;\leq\; \frac{4\,K(\psi)^2}{\mu^2}\,\frac{1+\log\tau}{\tau},$$

for all $\tau \geq 1$. Moreover, since

$$\frac{\|m^0\|_2}{\|m^*\|_2} \;\leq\; \left(\frac{\sum_{(v\leftarrow u)\in\vec{\mathcal{E}}}\big(\max_{i\in\mathcal{X}}\overline{B}^0_{uv}(i)\big)}{\sum_{(v\leftarrow u)\in\vec{\mathcal{E}}}\big(\min_{i\in\mathcal{X}}\underline{B}^0_{uv}(i)\big)}\right)^{\frac{1}{2}} \;=\; \sqrt{\frac{K(\psi)}{4}},$$

the initial term $\mathbb{E}\big[\langle Y^1, e^0\rangle^2\big] \leq 4\,K(\psi)\,\mathbb{E}\big[\|e^0\|_2^2\big]$ is upper bounded by $4\,K(\psi)^2$. Finally, putting all the pieces together, we obtain

$$\begin{aligned}\operatorname{var}\left(\frac{1}{t+1}\sum_{\tau=0}^{t}\langle Y^{\tau+1}, e^\tau\rangle\right) &\leq \frac{4\,K(\psi)^2}{\mu^2\,(t+1)^2}\sum_{\tau=1}^{t}\frac{1+\log\tau}{\tau} + \frac{4\,K(\psi)^2}{(t+1)^2}\\ &\overset{\text{(i)}}{\leq} \frac{4\,K(\psi)^2}{\mu^2}\,\frac{(1+\log(t+1))^2+4}{(t+1)^2},\end{aligned}$$

where inequality (i) follows from the facts $\sum_{\tau=1}^{t}(1+\log\tau)/\tau \leq (1+\log t)^2$, and $\mu < 2$. Consequently, we may apply Chebyshev's inequality [24] to control the stochastic deviation $\sum_{\tau=1}^{t+1}\langle Y^{\tau+1}, e^\tau\rangle/(t+1)$. More specifically, for $\gamma > 0$ (to be specified) we have

$$\mathbb{P}\Big(\Big|\frac{2}{\mu\,(t+1)}\sum_{\tau=0}^{t}\langle Y^{\tau+1}, e^\tau\rangle\Big| \;>\; \gamma\Big) \;\leq\; \frac{16\,K(\psi)^2}{\mu^4\,\gamma^2}\,\frac{(1+\log(t+1))^2+4}{(t+1)^2}. \tag{3.39}$$

We now combine our earlier bound (3.38) with the tail bound (3.39), making the specific choice

$$\gamma \;=\; \frac{4\,K(\psi)}{\mu^2\,\sqrt{\epsilon}}\;\frac{\sqrt{(1+\log(t+1))^2+4}}{t+1},$$

for a fixed $0<\epsilon<1$, thereby concluding that

$$\|e^{t+1}\|_2^2 \;\leq\; \frac{K(\psi)}{\mu^2}\,\frac{1+\log(t+1)}{t+1} \;+\; \frac{4\,K(\psi)}{\mu^2\,\sqrt{\epsilon}}\;\frac{\sqrt{(1+\log(t+1))^2+4}}{t+1},$$

with probability at least $1-\epsilon$. Simplifying the last bound, we obtain

$$\|e^{t+1}\|_2^2 \;\leq\; \frac{K(\psi)}{\mu^2}\Big(1+\frac{8}{\sqrt{\epsilon}}\Big)\,\frac{1+\log(t+1)}{t+1},$$

for all $t\geq 1$, with probability at least $1-\epsilon$.

3.5.3 Proof of Proposition 1

Recall the definition (3.8) of the probability mass function $\{p_{u\to v}(j)\}_{j\in\mathcal{X}}$ used in the update of directed edge $(u\to v)$. This probability depends on the current value of the message, so we can view it as being generated by a function $q_{u\to v}:\mathbb{R}^D\to\mathbb{R}^d$ that performs the mapping $m\mapsto\{p_{u\to v}(j)\}_{j\in\mathcal{X}}$. In terms of this function, we can rewrite the BP message update equation (3.3) on the directed edge $(u\to v)$ as $F_{u\to v}(m)=\Gamma_{uv}\,q_{u\to v}(m)$, where the renormalized compatibility matrix Γ_{uv} was defined previously (3.6). We now define the $D\times D$ block diagonal matrix $\Gamma:=\text{blkdiag}\{\Gamma_{uv}\}_{(u\to v)\in\vec{\mathcal{E}}}$, as well as the function $q:\mathbb{R}^D\to\mathbb{R}^D$ obtained by concatenating all of the functions $q_{u\to v}$, one for each directed edge. In terms of these quantities, we rewrite the global BP message update in the compact form $F(m)=\Gamma\,q(m)$.

With these preliminaries in place, we now bound the Lipschitz constant of the mapping $F:\mathbb{R}^D\to\mathbb{R}^D$. Given an arbitrary pair of messages $m,m'\in\mathcal{S}$, we have

$$\|F(m)-F(m')\|_2^2 \;=\; \|\Gamma\,\big(q(m)-q(m')\big)\|_2^2 \;=\sum_{(u\to v)\in\vec{\mathcal{E}}}\|\Gamma_{uv}\big(q_{u\to v}(m)-q_{u\to v}(m')\big)\|_2^2. \tag{3.40}$$

By the Perron-Frobenius theorem [47], we know that Γ_{uv} has a unique maximal eigenvalue of 1, achieved for the left eigenvector $\vec{1}\in\mathbb{R}^d$, where $\vec{1}$ denotes the vector of all ones. Since the d-dimensional vectors $q_{u\to v}(m)$ and $q_{u\to v}(m')$ are both probability distributions, we have $\langle\vec{1},\,q_{u\to v}(m)-q_{u\to v}(m')\rangle=0$. Therefore, we conclude that

$$\Gamma_{uv}\big(q_{u\to v}(m)-q_{u\to v}(m')\big) \;=\; \big(\Gamma_{uv}-\frac{z_{uv}\vec{1}^T}{\vec{1}^Tz_{uv}}\big)\big(q_{u\to v}(m)-q_{u\to v}(m')\big),$$

where z_{uv} denotes the right eigenvector of Γ_{uv} corresponding to the eigenvalue one. Combining this equality with the representation (3.40), we find that

$$\begin{aligned}\|F(m) - F(m')\|_2^2 &= \sum_{(u\to v)\in\vec{\mathcal{E}}} \|\big(\Gamma_{uv} - \frac{z_{uv}\vec{1}^T}{\vec{1}^T z_{uv}}\big)\big(q_{u\to v}(m) - q_{u\to v}(m')\big)\|_2^2 \\ &\leq \max_{(u,v)\in\mathcal{E}} \|\Gamma_{uv} - \frac{z_{uv}\vec{1}^T}{\vec{1}^T z_{uv}}\|_2^2 \;\; \|q(m) - q(m')\|_2^2. \qquad (3.41)\end{aligned}$$

It remains to upper bound the Lipschitz constant of the mapping $q : \mathbb{R}^D \to \mathbb{R}^D$ previously defined.

Lemma 3. *For all $m \neq m'$, we have*

$$\frac{\|q(m) - q(m')\|_2}{\|m - m'\|_2} \leq 2 \max_{(u\to v)\in\vec{\mathcal{E}}} \Phi_1(u \to v) \max_{(w\to u)\in\vec{\mathcal{E}}} \Phi_2(w \to u), \qquad (3.42)$$

where the quantities $\Phi_1(u \to v)$, and $\Phi_2(w \to u)$ were previously defined in (3.19a) and (3.19b).

As the proof of Lemma 3 is somewhat technical, we defer it to Appendix A.4. Combining the upper bound (3.42) with the earlier bound (3.41) completes the proof of the proposition.

3.6 Experimental Results

In this section, we present a variety of experimental results that confirm the theoretical predictions, and show that SBP is a practical algorithm. We provide results both for simulated graphical models, and real-world applications to image denoising and disparity computation.

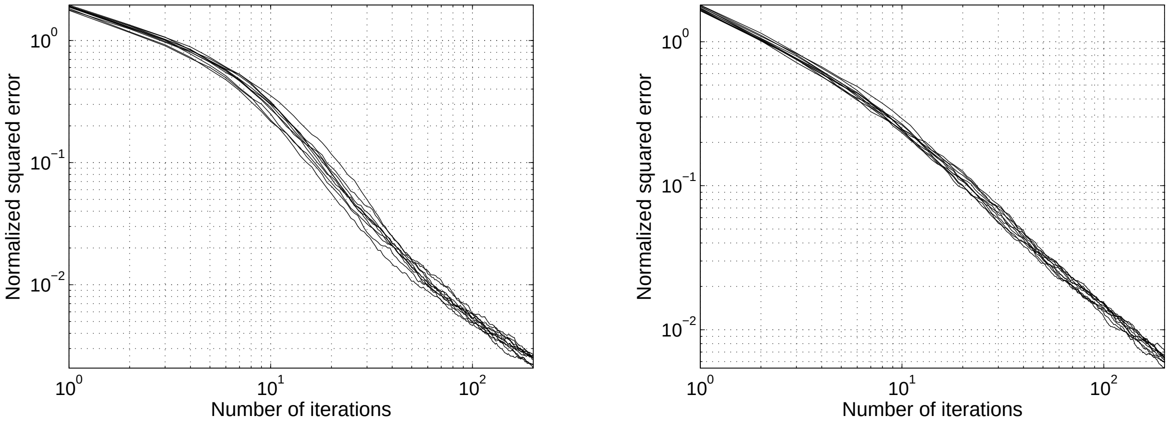
3.6.1 Simulations on Synthetic Problems

We start by performing some simulations for the potts model, in which the edge potentials are specified by a parameter $\gamma \in (0, 1]$, as discussed in Example 5. The node potentials are generated randomly, on the basis of fixed parameters $\mu \geq \sigma > 0$ satisfying $\mu + \sigma < 1$, as follows: for each $u \in \mathcal{V}$ and label $i \neq 1$, we generate an independent random variable $Z_{u;i}$ uniformly distributed on the interval $(-1, +1)$, and then set

$$\psi_u(i) = \begin{cases} 1 & i = 1 \\ \mu + \sigma Z_{u;i} & i \geq 2 \end{cases}.$$

For a fixed graph topology and collection of node/edge potentials, we first run BP to compute the fixed point m^*.[4] We then run the SBP algorithm to find the sequence of

[4] We stop the BP iterations when $\|m^{t+1} - m^t\|_2$ becomes less than 10^{-4}.



(a)

(b)

Figure 3.3: The panels illustrate the normalized squared error $\|m^t - m^*\|_2^2/\|m^*\|_2^2$ versus the number of iterations t for a chain of size $n = 100$ and state dimension $d = 64$. Each plot contains 10 different sample paths. Panel (a) corresponds to the coupling parameter $\gamma = 0.02$ whereas panel (b) corresponds to $\gamma = 0.05$. In all cases, the SBP algorithm was implemented with step size $\lambda^t = 2/(t+1)$, and the node potentials were generated with parameters $(\mu, \sigma) = (0.1, 0.1)$.

messages $\{m^t\}_{t=0}^{\infty}$ and compute the normalized squared error $\|m^t - m^*\|_2^2/\|m^*\|_2^2$. In cases where the normalized mean-squared error is reported, we computed it by averaging over 20 different runs of the algorithm. (Note that the runs are different, since the SBP algorithm is randomized.)

In our first set of experiments, we examine the consistency of the SBP on a chain-structured graph, as illustrated in Figure 2.3, representing a particular instance of a tree. We implemented the SBP algorithm with step size $\lambda^t = 2/(t+1)$, and performed simulations for a chain with $n = 100$ nodes, state dimension $d = 64$, node potential parameters $(\mu, \sigma) = (0.1, 0.1)$, and for two different choices of edge potential $\gamma \in \{0.02, 0.05\}$. The resulting traces of the normalized squared error versus iteration number are plotted in Figure 3.3; each panel contains 10 different sample paths. These plots confirm the prediction of strong consistency given in Theorem 6(a)—in particular, the error in each sample path converges to zero. We also observe that the typical performance is highly concentrated around its average, as can be observed from the small amount of variance in the sample paths.

Our next set of simulations are designed to study the effect of increasing of the state dimension d on convergence rates. We performed simulations both for the chain with $n = 100$ nodes, as well as a two-dimensional square grid with $n = 100$ nodes. In all cases, we implemented the SBP algorithm with step sizes $\lambda^t = 2/(t+1)$, and generated the node/edge potentials with parameters $(\mu, \sigma) = (0.1, 0.1)$ and $\gamma = 0.1$ respectively. In Figure 3.4, we plot the normalized mean-squared error (estimated by averaging over 20 trials) versus the number of iterations for the chain in panel (a), and the grid in panel (b). Each panel contains four different curves, each corresponding to a choice of state dimension $d \in \{128, 256, 512, 1024\}$.

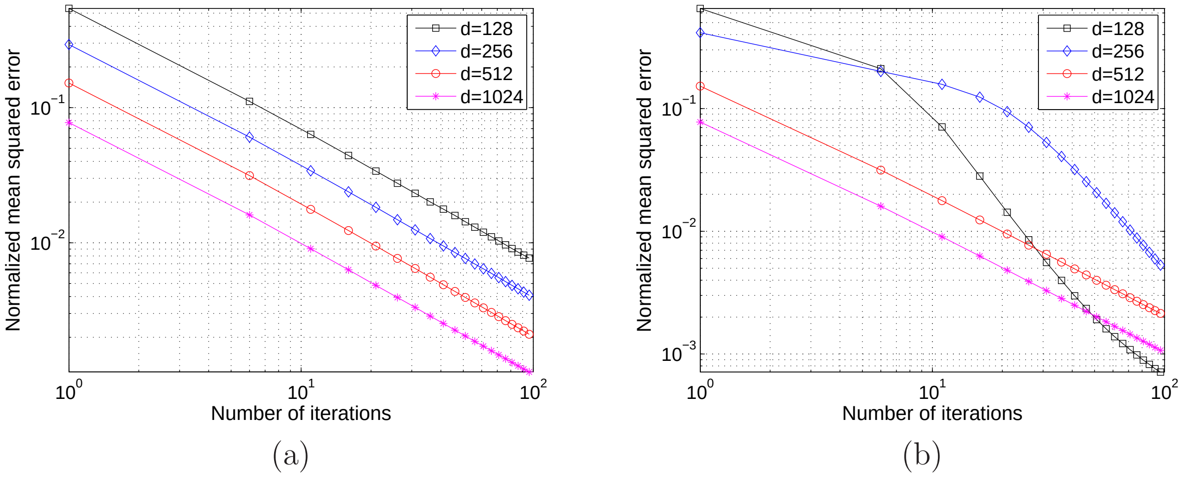


Figure 3.4: Effect of increasing state dimension on convergence rates. Plots of the normalized mean-squared error $\mathbb{E}\big[\|m^t - m^*\|_2^2\big]/\|m^*\|_2^2$ versus the number of iterations for two different graphs: (a) chain with $n = 100$ nodes, and (b) two-dimensional square grid with $n = 100$ nodes. In both panels, each curve corresponds different state dimension $d \in \{128, 256, 512, 1024\}$. All simulations were performed with step sizes $\lambda^t = 2/(t+1)$, and the node/edge parameters were generated with parameters $(\mu, \sigma) = (0.1, 0.1)$ and $\gamma = 0.1$ respectively.

For the given step size, Theorem 7 guarantees that the convergence rate should be upper bounded by $1/t^{\alpha}$ $(\alpha \leq 1)$ with the number of iterations t. In the log-log domain plot, this convergence rate manifests itself as a straight line with slope $-\alpha$. For the chain simulations shown in panel (a), all four curves exhibit exactly this behavior, with the only difference with increasing dimension being a vertical shift (no change in slope). For the grid simulations in panel (b), problems with smaller state dimension exhibit somewhat faster convergence rate than predicted by theory, whereas the larger problems ($d \in \{512, 1024\}$) exhibit linear convergence on the log-log scale.

As discussed previously, the SBP message updates are less expensive by a factor of d. The top two rows of Table 3.1 show the per iteration running time of both BP and SBP algorithms, for different state dimensions as indicated. As predicted by theory, the SBP running time per iteration is significantly lower than BP, scaling linearly in d in contrast to the quadratic scaling of BP. To be fair in our comparison, we also measured the total computation time required for either BP or SBP to converge to the fixed point up to a δ-tolerance, with $\delta = 0.01$. This comparison allows for the fact that BP may take many fewer iterations than SBP to converge to an approximate fixed point. Nonetheless, as shown in the bottom two rows of Table 3.1, in all cases except one (chain graph with dimension $d = 128$), we still see significant speed-ups from SBP in this overall running time. This gain becomes especially pronounced for larger dimensions, where these types of savings are more important.

		$d = 128$	$d = 256$	$d = 512$	$d = 1024$
Chain	BP (per iteration)	0.0700	0.2844	2.83	18.0774
	SBP (per iteration)	0.0036	0.0068	0.0145	0.0280
	BP (total)	0.14	0.57	5.66	36.15
	SBP (total)	0.26	0.27	0.29	0.28
Grid	BP (per iteration)	0.1300	0.5231	5.3125	32.5050
	SBP (per iteration)	0.0095	0.0172	0.0325	0.0620
	BP (total)	0.65	3.66	10.63	65.01
	SBP (total)	0.21	1.31	0.65	0.62

Table 3.1: Comparison of BP and SBP computational cost for two different graphs each with $n = 100$ nodes. For each graph type, the top two rows show per iteration running time (in seconds) of the BP and SBP algorithms for different state dimensions. The bottom two rows show total running time (in seconds) to compute the message fixed point to $\delta = 0.01$ accuracy.

3.6.2 Applications in Image Processing and Computer Vision

In our next set of experiments, we study the SBP on some larger scale graphs and more challenging problem instances, with applications to image processing and computer vision. Message-passing algorithms can be used for image denoising, in particular, on a two dimensional square grid where every node corresponds to a pixel. Running the BP algorithm on the graph, one can obtain (approximations to) the most likely value of every pixel based on the noisy observations. In this experiment, we consider a 200×200 image with $d = 256$ gray-scale levels, as showin in Figure 3.5(a). We then contaminate every pixel with an independent Gaussian random variable with standard deviation $\sigma = 0.1$, as shown in Figure 3.5(b). Enforcing the potts model with smoothness parameter $\gamma = 0.05$ as the edge potential, we run BP and SBP for the total of $t = 5$ and $t = 100$ iterations, respectively, to obtain the refined images (see panels (c) and (d), respectively, in Figure 3.5). Figure 3.6 illustrates the mean-squared error versus the running time for both BP and SBP denoising. As one can observe, despite smaller jumps in the error reduction, the per-iteration running time of SBP is substantially lower than BP. Overall, SBP has done a marginally better job than BP in a substantially shorter amount of time in this instance. Note that the purpose of this experiment is not to analyze the potential of SBP (or for that matter BP) in image denoising, but to rather observe their relative performances and computational complexities.

Finally, in our last experiment, we apply SBP to a computer vision problem. Graphical models and message-passing algorithms are popular in application to the stereo vision problem [112, 60], in which the goal is to estimate objects depth based on the pixel dissimilarities in two (left and right view) images. Adopting the original model in Sun et al. [112], we use a form of the prior to enforce smoothness, and also use the observation potentials given in the Sun et al. paper. We then run BP and SBP (with step size $3/(t + 2)$) for a total of $t = 10$ and $t = 50$ iterations respectively in order to estimate the pixel dissimilarities. The results for the test image "map" are presented in Figure 3.7. Here, the maximum pixel dissimilarity

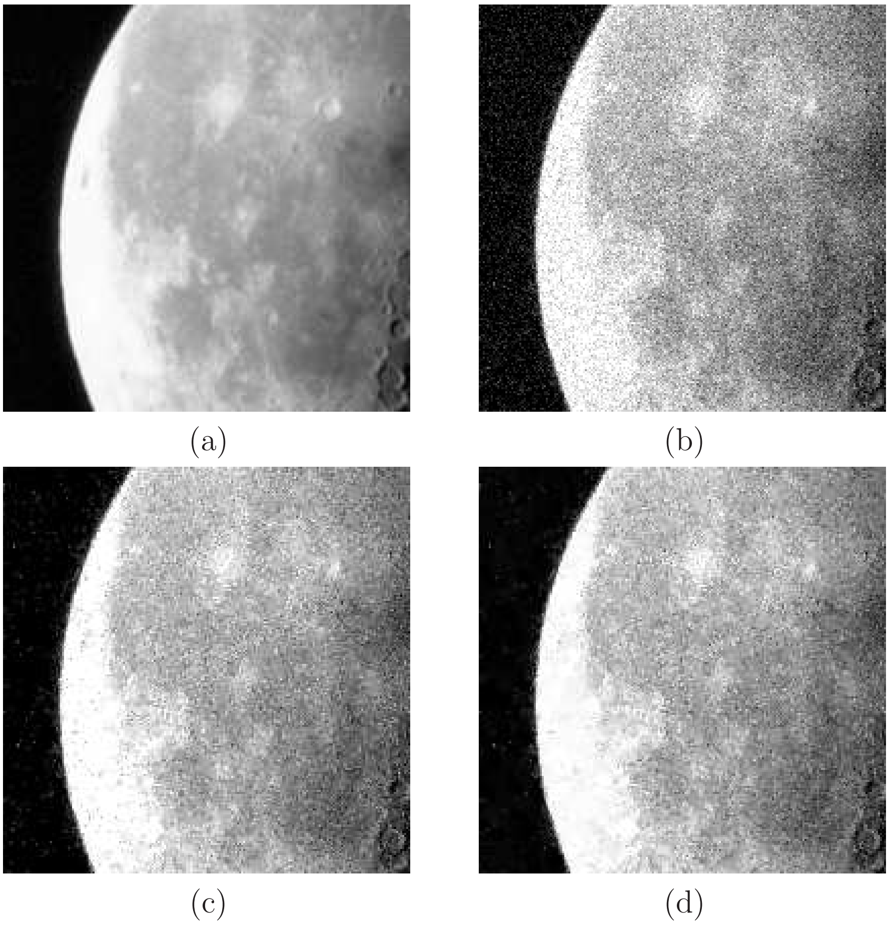


Figure 3.5: Image denoising application, (a) original image, (b) noisy image, (c) refined image obtained from BP after $t = 5$ iterations, and (d) refined image obtained from SBP after $t = 100$ iterations. The image is 200×200 with $d = 256$ gray-scale levels. The SBP step size, the potts model parameter, and noise standard deviation are set to $\lambda^t = 1/(t + 1)$, $\gamma = 0.05$, and $\sigma = 0.1$, respectively.

is $d = 32$, which makes stereo vision a relatively low-dimensional problem. In this particular application, the SBP is faster by about a factor of $3 - 4$ times per iteration; however, the need to run more iterations makes it comparable to BP. This is to be expected since the state dimension $d = 32$ is relatively small, and the relative advantage of SBP becomes more significant for larger state dimensions d.

3.7 Conclusion

In this chapter, we have developed and analyzed a new and low-complexity alternative to the BP message-passing. The SBP algorithm has per iteration computational complexity that

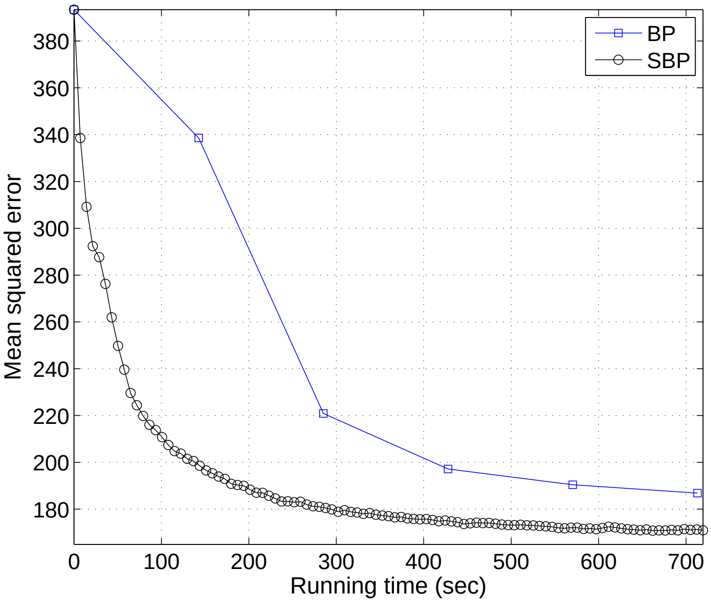


Figure 3.6: Mean-squared error versus the running time (in seconds) for both BP and SBP image denoising. The simulations are performed with the step size $\lambda^t = 1/(t+1)$, and the potts model parameter $\gamma = 0.05$ on a 200×200 image with $d = 256$ gray-scale levels. The noise is assumed to be additive, independent Gaussian random variables with standard deviation $\sigma = 0.1$.

scales linearly in the state dimension d, as opposed to the quadratic dependence of BP, and a communication cost of $\log_2 d$ bits per edge and iteration, as opposed to $d-1$ real numbers for standard BP message updates. Stochastic belief propagation is also easy to implement, requiring only random number generation and the usual distributed updates of a message-passing algorithm. Our main contribution was to prove a number of theoretical guarantees for the SBP message updates, including convergence for any tree-structured problem, as well as for general graphs for which the ordinary BP message update satisfies a suitable contraction condition. In addition, we provided non-asymptotic upper bounds on the SBP error, both in expectation and in high probability.

The results described here suggest a number of directions for future research. First, the ideas exploited here have natural generalizations to problems involving continuous random variables and also other algorithms that operate over the sum-product semi-ring, including the generalized belief propagation algorithm [121] as well as reweighted sum-product algorithms [118]. More generally, the BP algorithm can be seen as optimizing the dual of the Bethe free energy function [121], and it would be interesting to see if SBP can be interpreted as a stochastic version of this Bethe free energy minimization. It is also natural to consider whether similar ideas can be applied to analyze stochastic forms of message-passing over other semi-rings, such as the max-product algebra that underlies the computation of maximum a posteriori (MAP) configurations in graphical models. In this paper, we have developed

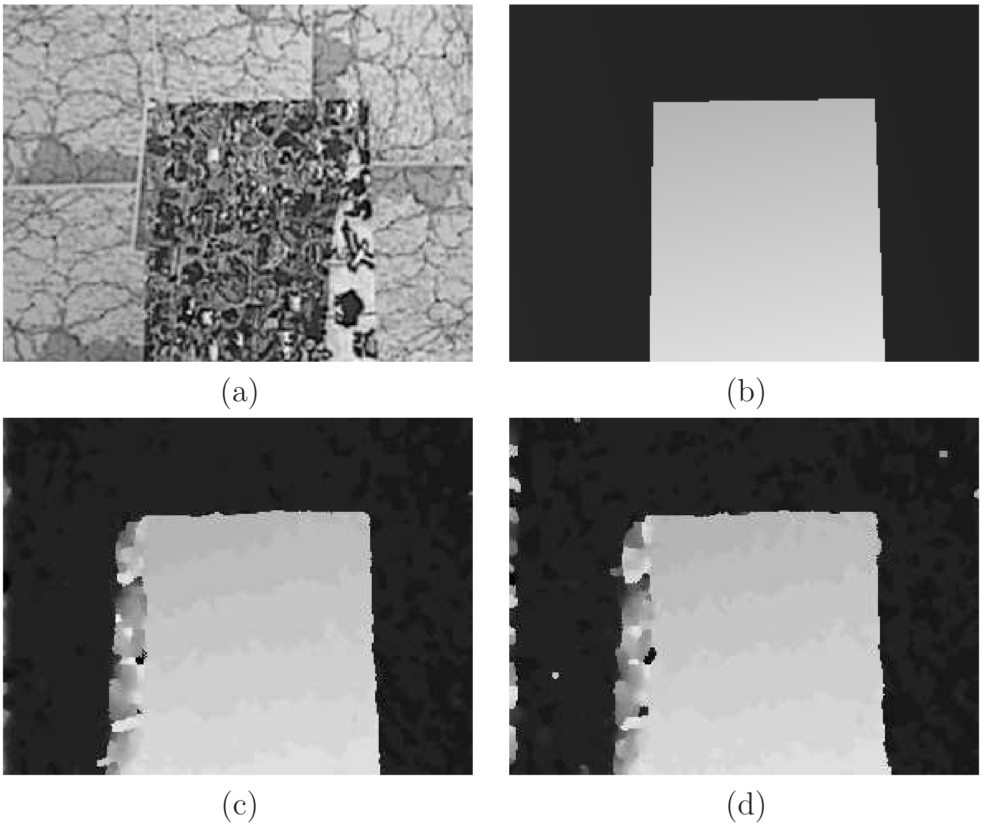


Figure 3.7: Stereo vision, depth recognition, application, (a) reference image, (b) ground truth, (c) BP estimate after $t = 10$ iterations, and (d) SBP estimate after $t = 50$ iterations. The algorithms are applied to the standard "map" image with maximum pixel dissimilarity $d = 32$. The SBP step size is set to $\lambda^t = 3/(t+2)$.

SBP for applications to Markov random fields with pairwise interactions. In principle, any undirected graphical model with discrete variables can be reduced to this form [121, 119]; however, in certain applications, such as decoding of LDPC codes over non-binary state spaces, this could be cumbersome. For such cases, it would be useful to derive a variant of SBP that applies directly to factor graphs with higher-order interactions. Moreover, the results derived in this paper are based on the assumption that the co-domain of the potential functions do not include zero. We suspect that these condition might be relaxed, and similar results could be obtained. Finally, our analysis for general graphs has been done under a contractivity condition, but it is likely that this requirement could be loosened. Indeed, the SBP algorithm works well for many problems where this condition need not be satisfied.[5]

[5] The materials of this chapter have been published in papers [84, 87].

Chapter 4

Stochastic Orthogonal Series Message-Passing

4.1 Introduction

In the previous chapter, we proposed a new low-complexity alternative to the belief propagation algorithm for the case of discrete random variables. However, in many applications of graphical models, we encounter random variables that take on continuous values (as opposed to discrete). For instance, in computer vision, the problem of optical flow calculation is most readily formulated in terms of estimating a vector field in $\mathbb{R}^2$. Other applications involving continuous random variables include tracking problems in sensor networks, vehicle localization, image geotagging, and protein folding in computational biology. With certain exceptions (such as multivariate Gaussian problems), the marginalization problem is very challenging for continuous random variables: in particular, the messages correspond to functions, so that they are expensive to compute and transmit, in which case BP may be limited to small-scale problems. Motivated by this challenge, researchers have proposed different techniques to reduce complexity of BP in different applications [5, 110, 111, 52, 32, 51, 27, 53, 20, 66, 106, 1]. For instance, various types of quantization schemes [27, 53] have used to reduce the effective state space and consequently the complexity. In another line of work, researchers have proposed stochastic methods inspired by particle filtering [5, 110, 111, 32, 51, 52]. These techniques are typically based on approximating the messages as weighted particles [32, 51], or mixture of Gaussians [111, 52]. Other researchers [106] have proposed the use of kernel methods to simultaneously estimate parameters and compute approximate marginals in a simultaneous manner.

In this chapter, we present a low-complexity alternative to BP with continous variables. Our method, which we refer to as stochastic orthogonal series message-passing (SOSMP), is applicable to general pairwise Markov random fields, and is equipped with various theoretical guarantees. As suggested by its name, the algorithm is based on combining two ingredients:

orthogonal series approximation of the messages, and the use of stochastic updates for efficiency. In this way, the SOSMP updates lead to a randomized algorithm with substantial reductions in communication and computational complexity. Our main contributions are to analyze the convergence properties of the SOSMP algorithm, and to provide rigorous bounds on the overall error as a function of the associated computational complexity. In particular, for tree-structured graphs, we estabish almost sure convergence, and provide an explicit inverse polynomial convergence rate (Theorem 8). For loopy graphical models on which the usual BP updates are contractive, we also establish similar convergence rates (Theorem 9). Our general theory provides quantitative upper bounds on the number of iterations required to compute a δ-accurate approximation to the BP message fixed point, as we illustrate in the case of kernel-based potential functions (Theorem 10).

The reminder of this chapter is organized as follows. We begin in Section 4.2, with the necessary background and the problem statement. Section 4.3 is devoted to a precise description of the SOSMP algorithm. In Section 4.4, we state our main theoretical results and develop some of their corollaries. In Section 4.5, we provide the proofs of our main results, with some of the technical aspects deferred to the appendices. In order to demonstrate the algorithm's effectiveness and confirm theoretical predictions, we provide some experimental results, on both synthetic and real data, in Section 4.6. Finally, we conclude the chapter in Section 4.7.

4.2 Background and Problem Statement

We begin by giving a precise description of the problem. Consider a pairwise Markov random field $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, consisting of a collection of vertices $\mathcal{V} = \{1, 2, \ldots, n\}$, along with a collection of edges $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$. For each $v \in \mathcal{V}$, let X_v be a random variable taking values in some continuous space $\mathcal{X}$. As discussed in the previous chapters, the pairwise Markov random field, defines a family of joint probability distributions over the random vector $X = \{X_v | v \in \mathcal{V}\}$, in which each density must factorize in terms of local potential functions associated with edges and nodes of the graph. More precisely, we consider the probability density p that respect the graph structure

$$p(x_1, x_2, \ldots, x_n) \;\propto\; \prod_{u \in \mathcal{V}} \psi_u(x_u) \prod_{(u,v) \in \mathcal{E}} \psi_{vu}(x_v, x_u). \tag{4.1}$$

Here $\psi_u : \mathcal{X} \to (0, \infty)$ is the node potential function, whereas $\psi_{vu} : \mathcal{X} \times \mathcal{X} \to (0, \infty)$ denotes the edge potential function.

The problem of marginalization, of utmost importance to many applications (see Section 2.1), suffers from the curse of dimensionality, since it requires computing a multi-dimensional integral over an $(n-1)$-dimensional space

$$p(x_v) \;:= \underbrace{\int_{\mathcal{X}} \dots \int_{\mathcal{X}}}_{(n-1)\text{ times}} p(x_1, x_2, \dots, x_n) \prod_{u \in \mathcal{V}\backslash\{v\}} dx_u \tag{4.2}$$

at each node $v \in \mathcal{V}$. Part of this exponential explosion can be circumvented by the use of the BP algorithm. In order to define the message-passing updates, we require some further notation. For each node $u \in \mathcal{V}$, recall the definition of the neighborhood $\mathcal{N}(u) = \{v \in \mathcal{V} \mid (u,v) \in \mathcal{E}\}$, the set of directed edges emanating from u, $\vec{\mathcal{E}}(u) = \{(u \to v) \mid v \in \mathcal{N}(u)\}$, as well as $\vec{\mathcal{E}} = \cup_{u \in \mathcal{V}} \vec{\mathcal{E}}(u)$, from Section 3.2. Let $\mathcal{M}$ denote the set of all probability densities defined on the space $\mathcal{X}$—that is

$$\mathcal{M} \;=\; \Big\{ m : \mathcal{X} \to [0, \infty) \,\Big|\, \int_{\mathcal{X}} m(x) dx = 1 \Big\}.$$

The messages passed by the BP algorithm are density functions, taking values in the space $\mathcal{M}$. More precisely, we assign one message $m_{u\to v} \in \mathcal{M}$ to every directed edge $(u \to v) \in \vec{\mathcal{E}}$, and we denote the collection of all messages by $m := \{m_{u\to v},\ (u \to v) \in \vec{\mathcal{E}}\}$. Note that the full collection of messages m takes values in the product space $\mathcal{M}^{|\vec{\mathcal{E}}|}$.

At an abstract level, the BP algorithm generates a sequence of message densities $\{m^t\}$ in the space $\mathcal{M}^{|\vec{\mathcal{E}}|}$, where $t = 0, 1, 2 \dots$ is the iteration number. The update of message m^t to message m^{t+1} can be written in the form $m^{t+1} = \mathcal{F}(m^t)$, where $\mathcal{F} : \mathcal{M}^{|\vec{\mathcal{E}}|} \to \mathcal{M}^{|\vec{\mathcal{E}}|}$ is a non-linear operator. This global operator is defined by the local update operators[1] $\mathcal{F}_{u\to v} : \mathcal{M}^{|\vec{\mathcal{E}}|} \to \mathcal{M}$, one for each directed edge of the graph, such that $m^{t+1}_{u\to v} = \mathcal{F}_{u\to v}(m^t)$. In more detail, the message update takes the form

$$\underbrace{[\mathcal{F}_{u\to v}(m^t)](\cdot)}_{m^{t+1}_{u\to v}(\cdot)} \;:=\; \kappa \int_{\mathcal{X}} \Big\{ \psi_{vu}(\cdot, x_u)\ \psi_u(x_u) \prod_{w \in \mathcal{N}(u)\backslash\{v\}} m^t_{w\to u}(x_u) \Big\}\, dx_u, \tag{4.3}$$

where κ is a normalization constant chosen to enforce the normalization condition

$$\int_{\mathcal{X}} m^{t+1}_{u\to v}(x_v)\, dx_v \;=\; 1.$$

Moreover, at each iteration $t = 0, 1, \dots$, each node $u \in \mathcal{V}$ transmits the message $m^{t+1}_{u\to v}$ (that is a real-valued function) to neighbor $v \in \mathcal{N}(u)$.

By concatenating the local updates (4.3), we obtain a global update operator $\mathcal{F} : \mathcal{M}^{|\vec{\mathcal{E}}|} \to \mathcal{M}^{|\vec{\mathcal{E}}|}$, as previously discussed. The goal of the BP message-passing is to obtain a fixed point, meaning an element $m^* \in \mathcal{M}^{|\vec{\mathcal{E}}|}$ such that $\mathcal{F}(m^*) = m^*$. Given a fixed point m^*, each node $v \in \mathcal{V}$

[1] It is worth mentioning, and important for the computational efficiency of BP , that $m_{u\to v}$ is only a function of the messages $m_{w\to u}$ for $w \in \mathcal{N}(u)\backslash\{v\}$. Therefore, we have $\mathcal{F}_{u\to v} : \mathcal{M}^{\rho_u - 1} \to \mathcal{M}$, where ρ_u is the degree of the node u. However, we suppress this local dependence so as to reduce notational clutter.

computes its marginal approximation $\tau_v^* \in \mathcal{M}$ by combining the local potential function ψ_v with a product of all incoming messages as

$$\tau_v^*(x_v) \;\propto\; \psi_v(x_v) \prod_{u \in \mathcal{N}(v)} m_{u \to v}^*(x_v). \tag{4.4}$$

Although the BP algorithm is considerably more efficient than the brute force approach to marginalization, the message update equation (4.3) still involves computing an integral and transmitting a real-valued function (message). With certain exceptions (such as multivariate Gaussians), these continuous-valued messages do not have finite representations, so that this approach is computationally very expensive. Although integrals can be computed by numerical methods, the BP algorithm requires performing many such integrals *at each iteration*, which becomes very expensive in practice. In this chapter, our goal is to develop low-complexity alternatives to BP for the case of continuous-valued random variables. Before doing so, we begin with some background on the main underlying ingredients: orthogonal series expansion, and stochastic message updates.

4.2.1 Orthogonal Series Expansion

As mentioned before, for continuous random variables, each message is a density function in the space $\mathcal{M} \subset L^2(\mathcal{X})$. We measure distances in this space using the usual L^2 norm $\|f - g\|_2^2 := \int_{\mathcal{X}} (f(x) - g(x))^2 \, dx$. A standard way in which to approximate functions is via orthogonal series expansion. In particular, let $\{\phi_j\}_{j=1}^{\infty}$ be an orthonormal basis of $L^2(\mathcal{X})$, meaning a collection of functions such that

$$\underbrace{\int_{\mathcal{X}} \phi_i(x)\phi_j(x) \, dx}_{:=\langle \phi_i, \, \phi_j \rangle} \;=\; \begin{cases} 1 & \text{when } i = j \\ 0 & \text{otherwise.} \end{cases} \tag{4.5}$$

Any function $f \in \mathcal{M} \subset L^2(\mathcal{X})$ then has an expansion of the form $f = \sum_{j=1}^{\infty} a_j \phi_j$, where $a_j = \langle f, \, \phi_j \rangle$ are the basis expansion coefficients.

Of course, maintaining the infinite sequence of basis coefficients $\{a_j\}_{j=1}^{\infty}$ is also computationally intractable, so that any practical algorithm will maintain only a finite number r of basis coefficients. For a given r, we let $\widehat{f}_r \propto \big[\sum_{j=1}^{r} a_j \phi_j\big]_+$ be the approximation based on the first r coefficients. (Applying the operator $[t]_+ = \max\{0, t\}$ amounts to projecting $\sum_{j=1}^{r} a_j \phi_j$ onto the space of non-negative functions, and we also normalize to ensure that it is a density function.) In using only r coefficients, we incur the *approximation error*

$$\|\widehat{f}_r - f\|_{L^2}^2 \;\overset{(i)}{\leq}\; \|\sum_{j=1}^{r} a_j \phi_j - f\|_{L^2}^2 \;\overset{(ii)}{=}\; \sum_{j=r+1}^{\infty} a_j^2 \tag{4.6}$$

where inequality (i) uses non-expansivity of the projection, and step (ii) follows from Parseval's theorem [88]. Consequently, the approximation error will depend both on

- how many coefficients r that we retain, and
- the decay rate of the expansion coefficients $\{a_j\}_{j=1}^{\infty}$.

For future reference, it is worth noting that the local message update (4.3) is defined in terms of an integral operator of the form

$$f(\cdot) \;\mapsto\; \int_{\mathcal{X}} \psi_{vu}(\cdot, x)\, f(x)\, dx. \tag{4.7}$$

Consequently, whenever the edge potential function ψ_{vu} has desirable properties—such as differentiability and/or higher order smoothness—then the messages also inherit these properties. With an appropriate choice of the basis $\{\phi_j\}_{j=1}^{\infty}$, such properties translate into decay conditions on the basis coefficients $\{a_j\}_{j=1}^{\infty}$. For instance, for α-times differentiable functions expanded into the Fourier basis, the Riemann-Lebesgue lemma [100] guarantees that the coefficients a_j decay faster than $(1/j)^{2\alpha}$. We develop these ideas at greater length in the sequel.

4.2.2 Stochastic Message Updates

In order to reduce the approximation error (4.6), the number of coefficients r needs to be increased (as a function of the ultimate desired error δ). Since increases in r lead to increases in computational complexity, we need to develop effective reduced-complexity methods. In this section, we describe (at a high-level) how this can be done via a stochastic version of the BP message-passing updates.

We begin by observing that message update (4.3), following some appropriate normalization, can be cast as an expectation operation. This equivalence is essential, because it allows us to obtain unbiased approximations of the message update using stochastic techniques. In particular, let us define the *normalized compatibility function*

$$\Gamma_{uv}(\cdot, x_u) := \psi_{vu}(\cdot, x_u)\frac{\psi_u(x_u)}{\beta_{uv}(x_u)}, \quad \text{where} \quad \beta_{uv}(x_u) := \psi_u(x_u)\int_{\mathcal{X}} \psi_{vu}(x_v, x_u)\, dx_v. \tag{4.8}$$

By construction, for each x_u, we have $\int_{\mathcal{X}} \Gamma_{uv}(x_v, x_u)\, dx_v = 1$.

Lemma 4. *Given an input collection of messages m, let Y be a random variable with density proportional to*

$$[p_{u\to v}(m)](y) \;\propto\; \beta_{uv}(y) \prod_{w\in\mathcal{N}(u)\backslash\{v\}} m_{w\to u}(y). \tag{4.9}$$

Then the message update equation (4.3) can be written as

$$[\mathcal{F}_{u\to v}(m)](\cdot) \;=\; \mathbb{E}_Y\big[\Gamma_{uv}(\cdot, Y)\big]. \tag{4.10}$$

Proof. Let us introduce the convenient shorthand $M(y) = \prod_{w \in \mathcal{N}(u) \setminus \{v\}} m_{w \to u}(y)$. By definition (4.3) of the message update, we have

$$[\mathcal{F}_{u \to v}(m)](\cdot) \;=\; \frac{\int_{\mathcal{X}} \left(\psi_{vu}(\cdot, y)\, \psi_u(y)\, M(y)\, dy \right.}{\int_{\mathcal{X}} \int_{\mathcal{X}} \left(\psi_{vu}(x, y)\, \psi_u(y) M(y) \right) dy\, dx}.$$

Since the integrand is positive, by Fubini's theorem [34], we can exchange the order of integrals in the denominator. Doing so and simplifying the expression yields

$$[\mathcal{F}_{u \to v}(m)](\cdot) \;=\; \int_{\mathcal{X}} \underbrace{\frac{\psi_{vu}(\cdot, y)}{\int_{\mathcal{X}} \psi_{vu}(x, y)\, dx}}_{\Gamma_{uv}(\cdot, y)} \; \underbrace{\frac{\beta_{uv}(y) M(y)}{\int_{\mathcal{X}} \beta_{uv}(z) M(z)\, dz}}_{[p_{u \to v}(m)](y)}\, dy, \tag{4.11}$$

which establishes the claim. □

Based on Lemma 4, we can obtain a stochastic approximation to the message update by drawing k i.i.d. samples Y_i from the density (4.9), and then computing $\sum_{i=1}^{k} \Gamma_{uv}(\cdot, Y_i) \,/\, k$. Given the non-negativity and chosen normalization of Γ_{uv}, note that this estimate belongs to $\mathcal{M}$ by construction. Moreover, it is an unbiased estimate of the correctly updated message, which plays an important role in our analysis. It is also worth mentioning that a similar idea is used in the stochastic belief propagation algorithm proposed in Chapter 3.

4.3 Description of the SOSMP Algorithm

In this section, we turn to the description of the SOSMP algorithm. The SOSMP algorithm involves a combination of the orthogonal series expansion techniques and stochastic methods previously described. Any particular version of the algorithm is specified by the choice of basis functions $\{\phi_j\}_{j=1}^{\infty}$ and two positive integers: the number of coefficients r that are maintained, and the number of samples k used in the stochastic update. Prior to running the algorithm, for each directed edge $(u \to v)$, we pre-compute the inner products

$$\gamma_{uv;j}(x_u) \;:=\; \underbrace{\int_{\mathcal{X}} \Gamma_{uv}(x_v, x_u)\, \phi_j(x_v)\, dx_v}_{\langle \Gamma_{uv}(\cdot, x_u), \phi_j(\cdot) \rangle}, \qquad \text{for } j = 1, \ldots, r. \tag{4.12}$$

When ψ_{vu} is a symmetric and positive semidefinite kernel function, these inner products have an explicit and simple representation in terms of its Mercer eigendecomposition (see Section 4.4.3). In the general setting, these r inner products can be computed via standard numerical integration techniques. Note that this is a fixed (one-time) cost prior to running the algorithm.

SOSMP algorithm for marginalization:

1. At time $t = 0$, initialize the message coefficients

$$a^0_{u\to v;j} = \frac{1}{r} \quad \text{for all } j = 1, \ldots, r, \text{ and } (u \to v) \in \vec{\mathcal{E}}.$$

2. For iterations $t = 0, 1, 2, \ldots$, and for each directed edge $(u \to v)$

   (a) Form the projected message approximation $\widehat{m}^t_{w\to u}(\cdot) = \Big[\sum_{j=1}^r a^t_{w\to u;j}\phi_j(\cdot)\Big]_+$, for all $w \in \mathcal{N}(u)\backslash\{v\}$.

   (b) Draw k i.i.d. samples Y_i from the probability density proportional to

$$\beta_{uv}(y) \prod_{w\in\mathcal{N}(u)\backslash\{v\}} \widehat{m}^t_{w\to u}(y), \tag{4.13}$$

   where β_{uv} was previously defined in equation (4.8).

   (c) Use the samples $\{Y_1, Y_2, \ldots, Y_k\}$ from step (b) to compute

$$\widetilde{b}^{t+1}_{u\to v;j} := \frac{1}{k}\sum_{i=1}^{k} \gamma_{uv;j}(Y_i) \qquad \text{for } j = 1, 2, \ldots, r, \tag{4.14}$$

   where the function $\gamma_{uv;j}$ is defined in equation (4.12).

   (a) For step size $\eta^t = 1/(t+1)$, update the r-dimensional message coefficient vectors $a^t_{u\to v} \mapsto a^{t+1}_{u\to v}$ via

$$a^{t+1}_{u\to v} = (1-\eta^t)\, a^t_{u\to v} + \eta^t\, \widetilde{b}^{t+1}_{u\to v}. \tag{4.15}$$

Figure 4.1: The *SOSMP* algorithm for continuous state spaces.

At each iteration $t = 0, 1, 2, \ldots$, the algorithm maintains an r-dimensional vector of basis expansion coefficients

$$a^t_{u\to v} = (a^t_{u\to v;1}, \ldots, a^t_{u\to v;r})^T \in \mathbb{R}^r, \quad \text{on directed edge } (u \to v) \in \vec{\mathcal{E}}.$$

This vector should be understood as defining the current message approximation $m^t_{u\to v}$ on the directed edge $(u \to v)$ via the expansion

$$m^t_{u\to v}(\cdot) := \sum_{j=1}^{r} a^t_{u\to v;j}\, \phi_j(\cdot) \tag{4.16}$$

We use $a^t = \{a^t_{u \to v} \mid (u \to v) \in \vec{\mathcal{E}}\}$ to denote the full set of $r\,|\vec{\mathcal{E}}|$ coefficients that are maintained by the algorithm at iteration t. With this notation, the algorithm consists of a sequence of steps, detailed in Figure 4.1, that perform the update $a^t \mapsto a^{t+1}$, and hence implicitly the update $m^t \mapsto m^{t+1}$.

As can be seen by inspection of the steps in Figure 4.1, each iteration requires $\mathcal{O}(rk)$ floating point operations per directed edge, which yields a total of $\mathcal{O}(rk\,|\vec{\mathcal{E}}|)$ operations per iteration.

4.4 Main Theoretical Results

We now turn to the theoretical analysis of the SOSMP algorithm, and guarantees relative to the fixed points of the true BP algorithm. For any tree-structured graph, the BP algorithm is guaranteed to have a unique message fixed point $m^* = \{m^*_{u \to v},\ (u \to v) \in \vec{\mathcal{E}}\}$. For graphs with cycles, uniqueness is no longer guaranteed, which would make it difficult to compare with the SOSMP algorithm. Accordingly, in our analysis of the loopy graph, we make a natural contractivity assumption, which guarantees uniqueness of the fixed point m^*.

The SOSMP algorithm generates a random sequence $\{a^t\}_{t=0}^{\infty}$, which define message approximations $\{m^t\}_{t=0}^{\infty}$ via the expansion (4.16). Of interest to us are the following questions:

- under what conditions do the message iterates approach a neighborhood of the BP fixed point m^* as $t \to +\infty$?
- when such convergence takes place, how fast is it?

In order to address these questions, we separate the error in our analysis into two terms: algorithmic error and approximation error. For a given r, let Π^r denote the projection operator onto the span of $\{\phi_1, \phi_2, \ldots, \phi_r\}$. In detail, given a function f represented in terms of the infinite series expansion $f = \sum_{j=1}^{\infty} a_j \phi_j$, we have

$$\Pi^r(f) := \sum_{j=1}^{r} a_j \phi_j.$$

For each directed edge $(u \to v) \in \vec{\mathcal{E}}$, define the functional error

$$\Delta^t_{u \to v} := m^t_{u \to v} - \Pi^r(m^*_{u \to v}) \tag{4.17}$$

between the message approximation at time t, and the BP fixed point projected onto the first r basis functions. Moreover, define the approximation error at the BP fixed point as

$$A^r_{u \to v} := m^*_{u \to v} - \Pi^r(m^*_{u \to v}). \tag{4.18}$$

Since $\Delta^t_{u\to v}$ belongs to the span of the first r basis functions, the Pythagorean theorem implies that the overall error can be decomposed as

$$\|m^t_{u\to v} - m^*_{u\to v}\|^2_{L^2} \quad = \quad \underbrace{\|\Delta^t_{u\to v}\|^2_{L^2}}_{\text{Estimation error}} \quad + \quad \underbrace{\|A^r_{u\to v}\|^2_{L^2}}_{\text{Approximation error}} \quad . \tag{4.19}$$

Note that the approximation error term is independent of the iteration number t, and can only be reduced by increasing the number r of coefficients used in the series expansion. Our analysis of the estimation error is based on controlling the $|\vec{\mathcal{E}}|$-dimensional error vector

$$\rho^2\big(\Delta^t\big) \;:=\; \big\{\|\Delta^t_{u\to v}\|^2_{L^2},\ (u\to v)\in\vec{\mathcal{E}}\big\} \;\in\; \mathbb{R}^{|\vec{\mathcal{E}}|}, \tag{4.20}$$

and in particular showing that it decreases as $\mathcal{O}(1/t)$ up to a lower floor imposed by the approximation error. In order to analyze the approximation error, we introduce the r-dimensional vector of approximation errors

$$\rho^2\big(A^r\big) := \big\{\|A^r_{u\to v}\|^2_{L^2},\ (u\to v)\in\vec{\mathcal{E}}\big\} \in \mathbb{R}^{|\vec{\mathcal{E}}|}. \tag{4.21}$$

By increasing r, we can reduce this approximation error term, but at the same time, we increase the computational complexity of each update. In Section 4.4.3, we discuss how to choose r so as to trade-off the estimation and approximation errors with computational complexity.

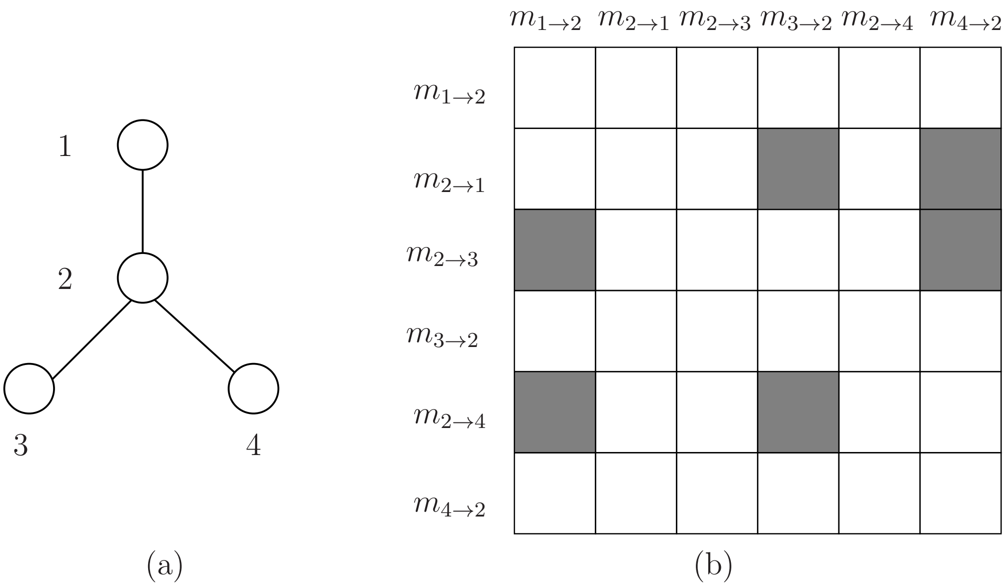
4.4.1 Bounds for Tree-Structured Graphs

With this set-up, we now turn to bounds for tree-structured graphs. Our analysis of the tree-structured case controls the vector of errors $\rho^2\big(\Delta^t\big)$ using a nilpotent matrix $N\in\mathbb{R}^{r\times r}$ determined by the tree structure (see the previous chapter). Recall that a matrix N is nilpotent with order ℓ if $N^\ell = 0$ and $N^{\ell-1}\neq 0$ for some ℓ. As illustrated in Figure 4.2, the rows and columns of N are indexed by directed edges. For the row indexed by $(u\to v)$, there can be non-zero entries only for edges in the set $\{(w\to u),\ w\in\mathcal{N}(u)\backslash\{v\}\}$. These directed edges are precisely those that pass messages relevant in updating the message from u to v, so that N tracks the propagation of message information in the graph. As shown in Chapter 3 (see Lemma 1), the matrix N with such structure is nilpotent with degree at most the diameter of the tree.

Moreover, our results on tree-structured graphs impose one condition on the vector of approximation errors A^r, namely that

$$\inf_{y\in\mathcal{X}} \Pi^r\big(\Gamma_{uv}(x,y)\big) \;>\; 0, \quad \text{and} \quad |A^r_{u\to v}(x)| \;\le\; \frac{1}{2}\inf_{y\in\mathcal{X}} \Pi^r\big(\Gamma_{uv}(x,y)\big) \tag{4.22}$$

for all $x\in\mathcal{X}$ and all directed edges $(u\to v)\in\vec{\mathcal{E}}$. This condition ensures that the L^2-norm of the approximation error is not too large relative to the compatibility functions. Since



	$m_{1\to2}$	$m_{2\to1}$	$m_{2\to3}$	$m_{3\to2}$	$m_{2\to4}$	$m_{4\to2}$
$m_{1\to2}$						
$m_{2\to1}$				■		■
$m_{2\to3}$	■					■
$m_{3\to2}$						
$m_{2\to4}$	■			■		
$m_{4\to2}$						

Figure 4.2: (a) A simple tree with $|\mathcal{E}| = 3$ edges and hence $|\vec{\mathcal{E}}| = 6$ directed edges. (b) Structure of nilpotent matrix $N \in \mathbb{R}^{|\vec{\mathcal{E}}|\times|\vec{\mathcal{E}}|}$ defined by the graph in (a). Rows and columns of the matrix are indexed by directed edges $(u \to v) \in \vec{\mathcal{E}}$; for the row indexed by $(u \to v)$, there can be non-zero entries only for edges in the set $\{(w \to u),\ w \in \mathcal{N}(u)\backslash\{v\}\}$.

$\sup_{x,y\in\mathcal{X}} |\Pi^r\big(\Gamma_{uv}(x,y)\big) - \Gamma_{uv}(x,y)| \to 0$ and $\sup_{x\in\mathcal{X}} |A^r_{u\to v}(x)| \to 0$ as $r \to +\infty$, assuming that the compatibility functions are uniformly bounded away from zero, condition (4.22) will hold once the number of basis expansion coefficients r is sufficiently large. Finally, our bounds involve the constants

$$B_j \;:=\; \max_{(u,v)\in\mathcal{E}} \sup_{y\in\mathcal{X}} \langle\Gamma_{uv}(\cdot,y),\, \phi_j\rangle. \tag{4.23}$$

With this set-up, we have the following guarantees:

Theorem 8. *Suppose that $\mathcal{X}$ is closed and bounded, the node and edge potential functions are continuous, and that condition (4.22) holds. Then for any tree-structured model, the sequence of messages $\{m^t\}_{t=0}^{\infty}$ generated by the SOSMP algorithm have the following properties:*

(a) There is a nilpotent matrix $N \in \mathbb{R}^{|\vec{\mathcal{E}}|\times|\vec{\mathcal{E}}|}$ such that the error vector $\rho^2\big(\Delta^t\big)$ converges almost surely to the set

$$\mathcal{B} := \big\{e \in \mathbb{R}^{|\vec{\mathcal{E}}|} \;\big|\; |e| \preceq N(I-N)^{-1}\rho^2\big(A^r\big)\big\}, \tag{4.24}$$

where $\preceq$ denotes elementwise inequality for vectors.

(b) Furthermore, for all iterations $t = 1, 2, \ldots,$ we have

$$\mathbb{E}\big[\rho^2\big(\Delta^t\big)\big] \;\preceq\; \Big(12\sum_{j=1}^{r} B_j^2\Big)\,\frac{(I-\log t\; N)^{-1}}{t}\,(N\,\vec{1}\,+\,8)\,+\,N(I-N)^{-1}\rho^2\big(A^r\big). \tag{4.25}$$

Remarks: To clarify the statement in part (a), it guarantees that the difference $\rho^2(\Delta^t) - \Pi_{\mathcal{B}}(\rho^2(\Delta^t))$ between the error vector and its projection onto the set $\mathcal{B}$ converges almost surely to zero. Part (b) provides a quantitative guarantee on how quickly the expected absolute value of this difference converges to zero. In particular, apart from logarithmic factors in t, the convergence rate guarantees is of the order $\mathcal{O}(1/t)$.

4.4.2 Bounds for General Graphs

Our next theorem addresses the case of general graphical models. To ensure the uniqueness of the fixed point and convergence of BP, a sufficient condition is contraction of the BP update operator. In our analysis of the SOSMP algorithm for a general graph, we impose the following form of contractivity: there exists a constant $0 < \gamma < 2$ such that

$$\|\mathcal{F}_{u\to v}(m) - \mathcal{F}_{u\to v}(m')\|_{L^2} \;\leq\; \big(1 - \frac{\gamma}{2}\big) \sqrt{\frac{1}{|\mathcal{N}(u)\backslash\{v\}|} \sum_{w\in\mathcal{N}(u)\backslash\{v\}} \|m_{w\to u} - m'_{w\to u}\|^2_{L^2}}, \tag{4.26}$$

for all directed edges $(u \to v) \in \vec{\mathcal{E}}$, and feasible messages m, and m'. We say that the ordinary BP algorithm is γ-contractive when condition (4.26) holds.

Theorem 9. *Suppose that the ordinary BP algorithm is γ-contractive, and consider the sequence of messages $\{m^t\}_{t=0}^{\infty}$ generated with step-size $\eta^t = 1/(\gamma\,(t+1))$. Then for all $t = 1, 2, \ldots$, the error sequence $\{\Delta^t_{u\to v}\}_{t=0}^{\infty}$ is bounded in mean-square as*

$$\mathbb{E}\big[\rho^2(\Delta^t)\big] \;\preceq\; \Bigg[\bigg(\frac{8\sum_{j=1}^{r} B_j^2}{\gamma^2}\bigg)\frac{\log t}{t} + \frac{1}{\gamma}\max_{(u\to v)\in\vec{\mathcal{E}}} \|A^r_{u\to v}\|^2_{L^2}\Bigg]\vec{1}. \tag{4.27}$$

where $A^r_{u\to v} = m^*_{u\to v} - \Pi^r(m^*_{u\to v})$ is the approximation error on edge $(u \to v)$.

Remarks: Theorem 9 guarantees that under the contractivity condition (4.26), the SOSMP iterates converge to a neighborhood of the BP fixed point. The error offset depends on the approximation error term that decays to zero as r is increased. Moreover, disregarding the logarithmic factor, the convergence rate is $\mathcal{O}(1/t)$, which is the best possible for a stochastic approximation scheme of this type [81, 2].

4.4.3 Explicit Rates for Kernel Classes

Theorems 8 and 9 are generic results that apply to any choices of the edge potential functions. In this section, we pursue a more refined analysis of the number of arithmetic operations that are required to compute a δ-*uniformly accurate* approximation to the BP fixed point

m^*, where $\delta > 0$ is a user-specified tolerance parameter. By a δ-uniformly accurate approximation, we mean a collection of messages m such that

$$\max_{(u \to v) \in \vec{\mathcal{E}}} \mathbb{E}\big[\|m_{u \to v} - m^*_{u \to v}\|^2_{L^2}\big] \;\leq\; \delta. \tag{4.28}$$

In order to obtain such an approximation, we need to specify both the number of coefficients r to be retained, and the number of iterations that we should perform. Based on these quantities, our goal is to the specify the *minimal number of basic arithmetic operations* $T(\delta)$ that are sufficient to compute a δ-accurate message approximation.

We study this issue in the context of kernel-based potential functions. In many applications, the edge potentials $\psi_{vu} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_+$ are symmetric and positive semidefinite (PSD) functions, frequently referred to as kernel functions. In detail, a PSD kernel function has the property that for all natural numbers n and $\{x_1, \ldots, x_n\} \subset \mathcal{X}$, the $n \times n$ kernel matrix with entries $\psi_{vu}(x_i, x_j)$ is symmetric and positive semidefinite. Commonly used examples include the Gaussian kernel $\psi_{vu}(x, y) = \exp(-\gamma\|x - y\|_2^2)$, the closely related Laplacian kernel, and other types of kernels that enforce smoothness priors. Any kernel function defines a positive semidefinite integral operator, namely via equation (4.7). When $\mathcal{X}$ is compact and the kernel function is continuous, then Mercer's theorem [100] guarantees that this integral operator has a countable set of eigenfunctions $\{\phi_j\}_{j=1}^{\infty}$ that form an orthonormal basis of $L^2(\mathcal{X})$. Moreover, the kernel function has the expansion

$$\psi_{vu}(x, y) \;=\; \sum_{j=1}^{\infty} \lambda_j \, \phi_j(x)\phi_j(y), \tag{4.29}$$

where $\lambda_1 \geq \lambda_2 \geq \cdots \geq 0$ are the eigenvalues, all guaranteed to be non-negative. In general, the eigenvalues might differ from edge to edge, but we suppress this dependence for simplicity in exposition. We study kernels that are trace class, meaning that the eigenvalues are absolutely summable (i.e., $\sum_{j=1}^{\infty} \lambda_j < \infty$).

For a given eigenvalue sequence $\{\lambda_j\}_{j=1}^{\infty}$ and some tolerance $\delta > 0$, we define the *critical dimension* $r^* = r^*(\delta; \{\lambda_j\})$ to be the smallest positive integer r such that

$$\lambda_r \leq \delta. \tag{4.30}$$

Since $\lambda_j \to 0$, the existence of $r^* < +\infty$ is guaranteed for any $\delta > 0$.

Theorem 10. *In addition to the conditions of Theorem 9, suppose that the compatibility functions are defined by a symmetric PSD trace-class kernel with eigenvalues $\{\lambda_j\}$. If we run the SOSMP algorithm with $r^* = r^*(\delta; \{\lambda_j\})$ basis coefficients, then it suffices to perform*

$$T(\delta; \{\lambda_j\}) \;=\; \mathcal{O}\Big(r^* \cdot \big(\sum_{j=1}^{r^*} \lambda_j^2\big) \; \big(1/\delta\big) \; \log(1/\delta)\Big) \tag{4.31}$$

arithmetic operations per edge in order to obtain a δ-accurate message vector m.

The proof of Theorem 10 is provided in Section 4.5.3. It is based on showing that the choice (4.30) suffices to reduce the approximation error to $\mathcal{O}(\delta)$, and then bounding the total operation complexity required to also reduce the estimation error.

Theorem 10 can be used to derive explicit estimates of the complexity for various types of kernel classes. We begin with the case of kernels in which the eigenvalues decay at an inverse polyomial rate: in particular, given some $\alpha > 1$, we say that they exhibit *α-polynomial decay* if there is a universal constant C such that

$$\lambda_j \;\leq\; \frac{C}{j^\alpha} \quad \text{for all } j = 1, 2, \ldots. \tag{4.32}$$

Examples of kernels in this class include Sobolov spline kernels [42], which are a widely used type of smoothness prior. For example, the spline class associated with functions that are s-times differentiable satisfies the decay condition (4.32) with $\alpha = 2s$.

Corollary 1. *In addition to the conditions of Theorem 9, suppose that the compatibility functions are symmetric kernels with α-polynomial decay (4.32). Then it suffices to perform*

$$T_{\text{poly}}(\delta) = \mathcal{O}\Big(\big(1/\delta\big)^{\frac{1+\alpha}{\alpha}} \log(1/\delta)\Big) \tag{4.33}$$

operations per edge in order to obtain a δ-accurate message vector m.

The proof of this corollary is immediate from Theorem 10: given the assumption $\lambda_j \leq C/j^\alpha$, we see that $r^* \leq (C/\delta)^{\frac{1}{\alpha}}$ and $\sum_{j=1}^{r^*} \lambda_j^2 = \mathcal{O}(1)$. Substituting into the bound (4.31) yields the claim. Corollary 1 confirms a natural intuition—namely, that it should be easier to compute an approximate BP fixed point for a graphical model with smooth potential functions. Disregarding the logarithmic factor (which is of lower-order), the operation complexity $T_{\text{poly}}(\delta)$ ranges ranges from $\mathcal{O}\big((1/\delta)^2\big)$, obtained as $\alpha \to 1^+$ all the way down to $\mathcal{O}\big(1/\delta\big)$, obtained as $\alpha \to +\infty$.

Another class of widely used kernels are those with exponentially decaying eigenvalues: in particular, for some $\alpha > 0$, we say that the kernel has *α-exponential decay* if there are universal constants (C, c) such that

$$\lambda_j \leq C \exp(-c j^\alpha) \qquad \text{for all } j = 1, 2, \ldots. \tag{4.34}$$

Examples of such kernels include the Gaussian kernel, which satisfies the decay condition (4.34) with $\alpha = 2$ (e.g., [108]).

Corollary 2. *In addition to the conditions of Theorem 9, suppose that the compatibility functions are symmetric kernels with α-exponential decay (4.34). Then it suffices to perform*

$$T_{\text{exp}}(\delta) = \mathcal{O}\Big((1/\delta)\, \big(\log(1/\delta)\big)^{\frac{1+\alpha}{\alpha}}\Big). \tag{4.35}$$

operations per edge in order to obtain a uniformly δ-accurate message vector m.

As with our earlier corollary, the proof of this claim is a straightforward consequence of Theorem 10. Corollary 2 demonstrates that kernel classes with exponentially decaying eigenvalues are not significantly different from parametric function classes, for which a stochastic algorithm would have operation complexity $\mathcal{O}(1/\delta)$. Apart from the lower order logarithmic terms, the complexity bound (4.35) matches this parametric rate.

4.5 Proof of the Main Results

We now turn to the proofs of our main results. They involve a collection of techniques from concentration of measure, stochastic approximation, and functional analysis.

4.5.1 Proof of Theorem 8

Our goal is to bound the error

$$\|\Delta^{t+1}_{u\to v}\|^2_{L^2} \;=\; \|m^{t+1}_{u\to v} - \Pi^r(m^*_{u\to v})\|^2_{L^2} \;=\; \sum_{j=1}^{r}\big[a^{t+1}_{u\to v;j} - a^*_{u\to v;j}\big]^2, \tag{4.36}$$

where the final equality follows by Parseval's theorem. Here $\{a^*_{u\to v;j}\}_{j=1}^r$ are the basis expansion coefficients that define the best r approximation to the BP fixed point m^*. The following lemma provides an upper bound on this error in terms of two related quantities. First, we let $\{b^t_{u\to v;j}\}_{j=1}^\infty$ denote the basis function expansion coefficients of the $\mathcal{F}_{u\to v}(\widehat{m}^t)$—that is,

$$\mathcal{F}_{u\to v}(\widehat{m}^t) \;=\; \sum_{j=1}^{\infty} b^t_{u\to v;j}\phi_j.$$

Second, for each $j = 1, 2, \ldots, r$, define the deviation

$$\zeta^{t+1}_{u\to v;j} \;:=\; \widetilde{b}^{t+1}_{u\to v;j} - b^t_{u\to v;j},$$

where the coefficients $\widetilde{b}^{t+1}_{u\to v;j}$ are updated in Step 2(c) Figure 4.1.

Lemma 5. *For each iteration $t = 0, 1, 2, \ldots,$ we have*

$$\|\Delta^{t+1}_{u\to v}\|^2_{L^2} \;\le\; \underbrace{\frac{2}{t+1}\sum_{j=1}^{r}\sum_{\tau=0}^{t}\big[b^\tau_{u\to v;j} - a^*_{u\to v;j}\big]^2}_{\textit{Deterministic term } D^{t+1}_{u\to v}} + \underbrace{\frac{2}{(t+1)^2}\sum_{j=1}^{r}\Big\{\sum_{\tau=0}^{t}\zeta^{\tau+1}_{u\to v;j}\Big\}^2}_{\textit{Stochastic term } S^{t+1}_{u\to v}} \tag{4.37}$$

The proof of this lemma is relatively straightforward; see Appendix B.1 for the details. Note that inequality (4.37) provides an upper bound on the error that involves two terms: the first term $D^{t+1}_{u\to v}$ depends only on the expansion coefficients $\{b^{\tau}_{u\to v;j}, \tau = 0, \ldots, t\}$ and the BP fixed point, and therefore is a deterministic quantity when we condition on all randomness in stages up to step t. The second term $S^{t+1}_{u\to v}$, even when conditioned on randomness through step t, remains stochastic, since the coefficients $\widetilde{b}^{t+1}_{u\to v}$ (involved in the error term $\zeta^{t+1}_{u\to v}$) are updated stochastically in moving from iteration t to $t+1$.

We split the remainder of our analysis into three parts: (a) control of the deterministic component; (b) control of the stochastic term; and (c) combining the pieces to provide a convergence bound.

Upper-bounding the deterministic term

By the Pythagorean theorem, we have

$$\sum_{\tau=0}^{t}\sum_{j=1}^{r}\big[b^{\tau}_{u\to v;j} - a^{*}_{u\to v;j}\big]^2 \;\le\; \sum_{\tau=0}^{t} \|\mathcal{F}_{u\to v}(\widehat{m}^t) - \mathcal{F}_{u\to v}(m^*)\|^2_{L^2} \tag{4.38}$$

In order to control this term, we make use of the following lemma, proved in Appendix B.2:

Lemma 6. *For all directed edges $(u \to v) \in \vec{\mathcal{E}}$, there exist constants $\{L_{u\to v, w\to u}, w \in \mathcal{N}(u)\backslash\{v\}\}$ such that*

$$\|\mathcal{F}_{u\to v}(\widehat{m}^t) \,-\, \mathcal{F}_{u\to v}(m^*)\|_{L^2} \;\le\; \sum_{w\in\mathcal{N}(u)\backslash\{v\}} L_{u\to v,w\to u}\, \|\widehat{m}^t_{w\to u} \,-\, m^*_{w\to u}\|_{L^2}$$

for all $t = 1, 2, \ldots$.

Substituting the result of Lemma 6 in equation (4.38) and performing some algebra, we find that

$$\begin{aligned}\sum_{\tau=0}^{t}\sum_{j=1}^{r}\big[b^{\tau}_{u\to v;j} - a^{*}_{u\to v;j}\big]^2 \;&\le\; \sum_{\tau=0}^{t}\Big(\sum_{w\in\mathcal{N}(u)\backslash\{v\}} L_{u\to v,w\to u}\,\|\widehat{m}^{\tau}_{w\to u} \,-\, m^*_{w\to u}\|_{L^2}\Big)^2 \\ &\le\; (\rho_u - 1)\sum_{\tau=0}^{t}\sum_{w\in\mathcal{N}(u)\backslash\{v\}} L^2_{u\to v,w\to u}\,\|\widehat{m}^{\tau}_{w\to u} \,-\, m^*_{w\to u}\|^2_{L^2},\end{aligned} \tag{4.39}$$

where ρ_u is the degree of node $u \in \mathcal{V}$. By definition, the message $\widehat{m}^{\tau}_{w\to u}$ is the L^2-projection of $m^{\tau}_{w\to u}$ onto $\mathcal{M}$. Since $m^*_{w\to u} \in \mathcal{M}$ and projection is non-expansive, we have

$$\begin{aligned}\|\widehat{m}^{\tau}_{w\to u} \,-\, m^*_{w\to u}\|^2_{L^2} \;&\le\; \|m^{\tau}_{w\to u} \,-\, m^*_{w\to u}\|^2_{L^2} \\ &=\; \|\Delta^{\tau}_{w\to u}\|^2_{L^2} \,+\, \|A^{r}_{w\to u}\|^2_{L^2},\end{aligned} \tag{4.40}$$

where in the second step we have used the Pythagorean identity and recalled the definitions of estimation error as well as approximation error from (4.17) and (4.18). Substituting the inequality (4.40) into the bound (4.39) yields

$$\sum_{\tau=0}^{t}\sum_{j=1}^{r}\left[b^{\tau}_{u\to v;j}-a^{*}_{u\to v;j}\right]^{2} \;\leq\; (\rho_u-1)\sum_{\tau=0}^{t}\sum_{w\in\mathcal{N}(u)\backslash\{v\}} L^{2}_{u\to v,w\to u}\left(\|\Delta^{\tau}_{w\to u}\|^{2}_{L^2}+\|A^{r}_{w\to u}\|^{2}_{L^2}\right).$$

Therefore, introducing the convenient shorthand $\widetilde{L}_{u\to v,w\to u} := 2\,(\rho_u-1)\,L^{2}_{u\to v,w\to u}$, we have shown that

$$D^{t+1}_{u\to v} \;\leq\; \frac{1}{t+1}\sum_{\tau=0}^{t}\sum_{w\in\mathcal{N}(u)\backslash\{v\}}\widetilde{L}_{u\to v,w\to u}\left(\|\Delta^{t}_{w\to u}\|^{2}_{L^2}\;+\;\|A^{r}_{w\to u}\|^{2}_{L^2}\right). \tag{4.41}$$

We make further use of this inequality shortly.

Controlling the stochastic term

We now turn to the stochastic part of the inequality (4.37). Our analysis is based on the following fact, proved in Appendix B.3:

Lemma 7. *For each $t\geq 0$, let $\mathcal{G}^t := \sigma(m^0,m^1,\ldots,m^t)$ be the σ-field generated by all messages through time t. Then for every fixed $j=1,2,\ldots,r$, the sequence $\zeta^{t+1}_{u\to v;j}=\widetilde{b}^{t+1}_{u\to v;j}-b^{t}_{u\to v;j}$ is a bounded martingale difference with respect to $\{\mathcal{G}^t\}_{t=0}^{\infty}$. In particular, we have $|\zeta^{t+1}_{u\to v;j}|\leq 2B_j$, where B_j was previously defined (4.23).*

Based on Lemma 7, standard martingale convergence results [34] guarantee that for each $j=1,2,\ldots,r$, we have $\sum_{\tau=0}^{t}\zeta^{\tau+1}_{u\to v;j}/(t+1)$ converges to 0 almost surely (a.s.) as $t\to\infty$, and hence

$$S^{t+1}_{u\to v} \;=\; \frac{2}{(t+1)^2}\sum_{j=1}^{r}\left\{\sum_{\tau=0}^{t}\zeta^{\tau+1}_{u\to v;j}\right\}^{2} \;=\; 2\sum_{j=1}^{r}\left\{\frac{1}{t+1}\sum_{\tau=0}^{t}\zeta^{\tau+1}_{u\to v;j}\right\}^{2} \;\xrightarrow{\text{a.s.}}\; 0. \tag{4.42}$$

Furthermore, we can apply the Azuma-Hoeffding inequality [24] in order to characterize the rate of convergence. For each $j=1,2,\ldots,r$, define the non-negative random variable $Z_j := \left\{\sum_{\tau=0}^{t}\zeta^{\tau+1}_{u\to v;j}\right\}^2/(t+1)^2$. Since $|\zeta^{\tau+1}_{u\to v;j}|\leq 2B_j$, for any $\delta\geq 0$, we have

$$\mathbb{P}\big(Z_j\geq\delta\big) \;=\; \mathbb{P}\big(\sqrt{Z_j}\;\geq\;\sqrt{\delta}\big) \;\leq\; 2\,\exp\left(-\,\frac{(t+1)\,\delta}{8\,B_j^2}\right),$$

for all $\delta>0$. Moreover, Z_j is non-negative; therefore, integrating its tail bound we can compute the expectation

$$\mathbb{E}[Z_j] \;=\; \int_0^\infty \mathbb{P}\big(Z_j\geq\delta\big)\,d\delta \;\leq\; 2\int_0^\infty \exp\left(-\,\frac{(t+1)\,\delta}{8\,B_j^2}\right)d\delta \;=\; \frac{16B_j^2}{t+1},$$

and consequently

$$\mathbb{E}[|S^{t+1}_{u\to v}|] \;\leq\; \frac{32\,\sum_{j=1}^{r} B_j^2}{t+1}. \tag{4.43}$$

Establishing convergence

We now make use of the results established so far to prove the claims. Substituting the upper bound (4.41) on $D^{t+1}_{u\to v}$ into the decomposition (4.37) from Lemma 5, we find that

$$\|\Delta^{t+1}_{u\to v}\|^2_{L^2} \;\leq\; \frac{1}{t+1}\sum_{\tau=0}^{t}\;\sum_{w\in\mathcal{N}(u)\backslash\{v\}} \widetilde{L}_{u\to v,w\to u}\left\{\|\Delta^{\tau}_{w\to u}\|^2_{L^2} \;+\; \|A^{r}_{w\to u}\|^2_{L^2}\right\} + S^{t+1}_{u\to v}. \tag{4.44}$$

For convenience, let us introduce the vector $T^{t+1} = \{T^{t+1}_{u\to v} \mid (u\to v)\in\vec{\mathcal{E}}\} \in \mathbb{R}^r$ with entries

$$T^{t+1}_{u\to v} \;:=\; \frac{1}{t+1}\left\{\sum_{w\in\mathcal{N}(u)\backslash\{v\}} \widetilde{L}_{u\to v,w\to u}\,\|\Delta^{0}_{w\to u}\|^2_{L^2}\right\} \;+\; S^{t+1}_{u\to v}. \tag{4.45}$$

Now define a matix $N\in\mathbb{R}^{r\times r}$ with entries indexed by the directed edges and set to

$$N_{u\to v,\;w\to s} := \begin{cases}\widetilde{L}_{u\to v,w\to u} & \text{if } s=u \text{ and } w\in\mathcal{N}(u)\backslash\{v\}\\ 0 & \text{otherwise.}\end{cases} \tag{4.46}$$

In terms of this matrix and the error terms $\rho^2(\,\cdot\,)$ previously defined in equations (4.20) and (4.21), the scalar inequalities (4.44) can be written in the matrix form

$$\rho^2\big(\Delta^{t+1}\big) \;\preceq\; N\left[\frac{1}{t+1}\sum_{\tau=1}^{t}\rho^2\big(\Delta^{\tau}\big)\right] + N\,\rho^2\big(A^r\big) \;+\; T^{t+1}, \tag{4.47}$$

where $\preceq$ denotes the element-wise inequality based on the orthant cone.

From Lemma 1 in Chapter 3, the matrix N is guaranteed to be nilpotent with degree ℓ equal to the graph diameter. Consequently, unwrapping the recursion (4.47) for a total of $\ell = \operatorname{diam}(\mathcal{G})$ times yields

$$\rho^2\big(\Delta^{t+1}\big) \;\preceq\; T^{t+1}_0 \;+\; N\,T^{t+1}_1 \;+\;\ldots\;+\; N^{\ell-1}\,T^{t+1}_{\ell-1} \;+\; (N \;+\; N^2 \;+\;\ldots\;+\; N^{\ell})\,\rho^2\big(A^r\big),$$

where we define $T^{t+1}_0 \equiv T^{t+1}$, and then recursively $T^{t+1}_s := (\sum_{\tau=1}^{t} T^{\tau}_{s-1})/(t+1)$ for $s = 1,2,\ldots,\ell-1$. By the nilpotency of N, we have the identity $I + N + \ldots + N^{\ell-1} = (I-N)^{-1}$; so we can further simplify the last inequality

$$\rho^2\big(\Delta^{t+1}\big) \;\preceq\; \sum_{s=0}^{\ell-1} N^s\,T^{t+1}_s \;+\; N\,(I-N)^{-1}\,\rho^2\big(A^r\big). \tag{4.48}$$

Recalling the definition $\mathcal{B} := \big\{e \in \mathbb{R}^r \mid |e| \preceq N(I-N)^{-1}\rho^2\big(A^r\big)\big\}$, inequality (4.48) implies that

$$\big|\rho^2\big(\Delta^{t+1}\big) \, - \, \Pi_{\mathcal{B}}\big(\rho^2\big(\Delta^{t+1}\big)\big)\big| \preceq \sum_{s=0}^{\ell-1} N^s \, T_s^{t+1}. \tag{4.49}$$

We now use the bound (4.49) to prove both parts of Theorem 8.

Proof of Theorem 8(a): To prove the almost sure convergence claim in part (a), it suffices to show that for each $s = 0, 1, \ldots, \ell-1$, we have $T_s^t \xrightarrow{\text{a.s.}} 0$ as $t \to +\infty$. From equation (4.42) we know $S_{u\to v}^{t+1} \to 0$ almost surely as $t \to \infty$. In addition, the first term in (4.45) is at most $\mathcal{O}(1/t)$, so that also converges to zero as $t \to \infty$. Therefore, we conclude that $T_0^t \xrightarrow{\text{a.s.}} 0$ as $t \to \infty$.

In order to extend this argument to higher-order terms, let us recall the following elementary fact from real analysis [104]: for any sequence of real numbers $\{x^t\}_{t=0}^{\infty}$ such that $x^t \to 0$, then we also have $(\sum_{\tau=0}^{t} x^\tau)/t \to 0$. In order to apply this fact, we observe that $T_0^t \xrightarrow{\text{a.s.}} 0$ means that for almost every sample point ω the deterministic sequence $\{T_0^{t+1}(\omega)\}_{t=0}^{\infty}$ converges to zero. Consequently, the above fact implies that $T_1^{t+1}(\omega) = (\sum_{\tau=1}^{t} T_0^\tau(\omega))/(t+1) \to 0$ as $t \to \infty$ for almost all sample points ω, which is equivalent to asserting that $T_1^t \xrightarrow{\text{a.s.}} \vec{0}$. Iterating the same argument, we establish $T_s^{t+1} \xrightarrow{\text{a.s.}} \vec{0}$ for all $s = 0, 1, \ldots, \ell-1$, thereby concluding the proof of Theorem 8(a).

Proof of Theorem 8(b): Taking the expectation on both sides of the inequality (4.48) yields

$$\mathbb{E}\big[|\rho^2\big(\Delta^{t+1}\big) - \Pi_{\mathcal{B}}\big(\rho^2\big(\Delta^{t+1}\big)\big)|\big] \;\preceq\; \sum_{s=0}^{\ell-1} N^s \, \mathbb{E}[T_s^{t+1}]. \tag{4.50}$$

so that it suffices to upper bound the expectations $\mathbb{E}[T_s^{t+1}]$ for $s = 0, 1, \ldots, \ell-1$. In Appendix B.4, we prove the following result:

Lemma 8. *Define the r-vector $\vec{v} := \big\{ \sum_{j=1}^{r} B_j^2\big\}(4N\vec{1} + 32)$. Then for all $s = 0, 1, \ldots, \ell-1$ and $t = 0, 1, 2, \ldots$,*

$$\mathbb{E}[T_s^{t+1}] \;\preceq\; \frac{\vec{v}}{t+1}\left(\sum_{u=0}^{s} \frac{(\log(t+1))^u}{u!}\right), \tag{4.51}$$

Using this lemma, the proof of part (b) follows easily. In particular, substituting the bounds (4.51) into equation (4.50) and doing some algebra yields

$$\begin{aligned}\mathbb{E}\big[|\rho^2\big(\Delta^{t+1}\big) \;-\; \Pi_{\mathcal{B}}\big(\rho^2\big(\Delta^{t+1}\big)\big)|\big] \;&\preceq\; \sum_{s=0}^{\ell-1} N^s \sum_{u=0}^{s} \frac{(\log(t+1))^u}{u!}\left(\frac{\vec{v}}{t+1}\right)\\ &\preceq 3\sum_{s=0}^{\ell-1}(\log\left(t+1\right))^s\, N^s\left(\frac{\vec{v}}{t+1}\right)\\ &\preceq 3\,(I \;-\; \log\left(t+1\right)N)^{-1}\left(\frac{\vec{v}}{t+1}\right),\end{aligned}$$

where again we used the fact that $N^\ell = 0$.

4.5.2 Proof of Theorem 9

Recall the definition of the estimation error $\Delta^t_{u\to v}$ from (4.17). By Parseval's identity we know that $\|\Delta^t_{u\to v}\|^2_{L^2} = \sum_{j=1}^r (a^t_{u\to v;j} - a^*_{u\to v;j})^2$. For convenience, we introduce the following shorthands for mean squared error on the directed edge $(u \to v)$

$$\bar{\rho}^2(\Delta^t_{u\to v}) \;:=\; \mathbb{E}[\|\Delta^t_{u\to v}\|^2_{L^2}] \;=\; \mathbb{E}\Big[\sum_{j=1}^{r}(a^t_{u\to v;j} - a^*_{u\to v;j})^2\Big],$$

as well as the ℓ_∞ error

$$\bar{\rho}^2_{\max}(\Delta^t) \;:=\; \max_{(u\to v)\in\vec{\mathcal{E}}} \mathbb{E}[\|\Delta^t_{u\to v}\|^2_{L^2}],$$

similarly defined for approximation error

$$\rho^2_{\max}(A^r) \;:=\; \max_{(u\to v)\in\vec{\mathcal{E}}} \|A^r_{u\to v}\|^2_{L^2}.$$

Using the definition of $\bar{\rho}^2(\Delta^t_{u\to v})$, some algebra yields

$$\begin{aligned}\bar{\rho}^2(\Delta^{t+1}_{u\to v}) - \bar{\rho}^2(\Delta^t_{u\to v}) \;&=\; \mathbb{E}\Big[\sum_{j=1}^{r}\big(a^{t+1}_{u\to v;j} - a^*_{u\to v;j}\big)^2 \;-\; \sum_{j=1}^{r}\big(a^t_{u\to v;j} - a^*_{u\to v;j}\big)^2\Big]\\ &=\; \mathbb{E}\Big[\sum_{j=1}^{r}\big\{a^{t+1}_{u\to v;j} - a^t_{u\to v;j}\big\}\,\big\{\big(a^{t+1}_{u\to v;j} - a^t_{u\to v;j}\big) \;+\; 2\big(a^t_{u\to v;j} - a^*_{u\to v;j}\big)\big\}\Big].\end{aligned}$$

From the update equation (4.15), we have

$$a^{t+1}_{u\to v;j} - a^t_{u\to v;j} \;=\; \eta^t\,\big(\widetilde{b}^{t+1}_{u\to v;j} - a^t_{u\to v;j}\big),$$

and hence

$$\bar{\rho}^2(\Delta^{t+1}_{u\to v}) - \bar{\rho}^2(\Delta^t_{u\to v}) \;=\; U^t_{u\to v} + V^t_{u\to v}, \tag{4.52}$$

where

$$U^t_{u\to v} \;:=\; (\eta^t)^2 \sum_{j=1}^{r} \mathbb{E}\big[\big(\widetilde{b}^{t+1}_{u\to v;j} - a^t_{u\to v;j}\big)^2\big], \quad \text{and} \tag{4.53a}$$

$$V^t_{u\to v} \;:=\; 2\eta^t \sum_{j=1}^{r} \mathbb{E}\big[\big(\widetilde{b}^{t+1}_{u\to v;j} - a^t_{u\to v;j}\big)\,\big(a^t_{u\to v;j} - a^*_{u\to v;j}\big)\big]. \tag{4.53b}$$

The following lemma, proved in Appendix B.5, provides upper bounds on these two terms.

Lemma 9. *For all iterations $t = 0, 1, 2, \ldots$, we have*

$$U^t_{u\to v} \;\leq\; 4\,(\eta^t)^2 \sum_{j=1}^{r} B_j^2, \quad \textit{and} \tag{4.54a}$$

$$V^t_{u\to v} \;\leq\; \eta^t\big(1-\frac{\gamma}{2}\big)\rho^2_{max}(A^r) \,+\, \eta^t\big(1-\frac{\gamma}{2}\big)\bar{\rho}^2_{max}(\Delta^t) \,-\, \eta^t\big(1+\frac{\gamma}{2}\big)\bar{\rho}^2(\Delta^t_{u\to v}). \tag{4.54b}$$

We continue upper-bounding $\bar{\rho}^2(\Delta^{t+1}_{u\to v})$ by substituting the results of Lemma 9 into equation (4.52), thereby obtaining

$$\begin{aligned}\bar{\rho}^2(\Delta^{t+1}_{u\to v}) \;&\leq 4\,(\eta^t)^2 \sum_{j=1}^{r} B_j^2 \,+\, \eta^t\big(1-\frac{\gamma}{2}\big)\,\rho^2_{\max}(A^r) \\ &\quad +\, \eta^t\big(1-\frac{\gamma}{2}\big)\,\bar{\rho}^2_{\max}(\Delta^t) \,+\, \Big\{1-\eta^t\big(1+\frac{\gamma}{2}\big)\Big\}\,\bar{\rho}^2(\Delta^t_{u\to v}) \\ &\leq 4\,(\eta^t)^2 \sum_{j=1}^{r} B_j^2 \,+\, \eta^t\big(1-\frac{\gamma}{2}\big)\,\rho^2_{\max}(A^r) \,+\, \big(1-\eta^t\gamma\big)\,\bar{\rho}^2_{\max}(\Delta^t).\end{aligned}$$

Since this equation holds for all directed edges $(u \to v)$, taking the maximum over the left-hand side yields the recursion

$$\bar{\rho}^2_{\max}(\Delta^{t+1}) \;\leq\; (\eta^t)^2\,\overline{B}^2 \,+\, \eta^t\big(1-\frac{\gamma}{2}\big)\,\rho^2_{\max}(A^r) \,+\, \big(1-\eta^t\gamma\big)\,\bar{\rho}^2_{\max}(\Delta^t), \tag{4.55}$$

where we have introduced the shorthand $\overline{B}^2 = 4\sum_{j=1}^r B_j^2$. Setting $\eta^t = 1/(\gamma\,(t+1))$ and unwrapping this recursion, we find that

$$\begin{aligned}\bar{\rho}^2_{\max}(\Delta^{t+1}) \;&\leq\; \frac{\overline{B}^2}{\gamma^2}\sum_{\tau=1}^{t+1}\frac{1}{\tau\,(t+1)} \,+\, \frac{2-\gamma}{2\gamma}\,\rho^2_{\max}(A^r) \\ &\leq\; \frac{2\,\overline{B}^2}{\gamma^2}\,\frac{\log(t+1)}{t+1} \,+\, \frac{1}{\gamma}\,\rho^2_{\max}(A^r),\end{aligned}$$

which establishes the claim.

4.5.3 Proof of Theorem 10

As discussed earlier, each iteration of the SOSMP algorithm requires $\mathcal{O}(r)$ operations per edge. Consequently, it suffices to show that running the algorithm with $r = r^*$ coefficients for $(\sum_{j=1}^{r} \lambda_j^2)(1/\delta)\log(1/\delta)$ iterations suffices to achieve mean-squared error less than δ.

The bound (4.27) consists of two terms. In order to characterize the first term (estimation error), we need to bound B_j defined in (4.23). Using the orthonormality of the basis functions and the fact that the supremum is attainable over the compact space $\mathcal{X}$, we obtain

$$B_j \;=\; \max_{(u,v)\in\mathcal{E}} \sup_{y\in\mathcal{X}} \frac{\lambda_j\,\phi_j(y)}{\int_{\mathcal{X}} \psi_{vu}(x,y)\,dx} \;=\; \mathcal{O}(\lambda_j).$$

Therefore, the estimation error decays at the rate $\mathcal{O}\big((\sum_{j=1}^{r}\lambda_j^2)\;(\log t/t)\big)$, so that $t = \mathcal{O}\big((\sum_{j=1}^{r}\lambda_j^2)(1/\delta)\log(1/\delta)\big)$ iterations are sufficient to reduce it to $\mathcal{O}(\delta)$.

The second term (approximation error) in the bound (4.27) depends only on the choice of r, and in particular on the r-term approximation error $\|A^r_{u\to v}\|_{L^2}^2 = \|m^*_{u\to v} - \Pi^r(m^*_{u\to v})\|^2_{L^2}$. To bound this term, we begin by representing $m^*_{u\to v}$ in terms of the basis expansion $\sum_{j=1}^{\infty} a_j^*\phi_j$. By the Pythagorean theorem, we have

$$\|m^*_{u\to v} - \Pi^r(m^*_{u\to v})\|^2_{L^2} = \sum_{j=r+1}^{\infty} (a_j^*)^2. \tag{4.56}$$

Our first claim is that $\sum_{j=1}^{\infty}(a_j^*)^2/\lambda_j < \infty$. Indeed, since m^* is a fixed point of the message update equation, we have

$$m^*_{u\to v}(\cdot) \;\propto\; \int_{\mathcal{X}} \psi_{vu}(\cdot, y)\,M(y)\,dy,$$

where $M(\cdot) := \psi_u(\cdot)\prod_{w\in\mathcal{N}(u)\backslash\{v\}} m^*_{w\to u}(\cdot)$. Exchanging the order of integrations using Fubini's theorem, we obtain

$$a_j^* \;=\; \langle m^*_{u\to v},\, \phi_j\rangle \;\propto\; \int_{\mathcal{X}} \langle \phi_j(\cdot),\, \psi_{vu}(\cdot, y)\rangle\; M(y)\,dy. \tag{4.57}$$

By the eigenexpansion of ψ_{vu}, we have

$$\langle \phi_j(\cdot),\, \psi_{vu}(\cdot,y)\rangle \;=\; \sum_{k=1}^{\infty} \lambda_k\,\langle \phi_j,\,\phi_k\rangle\;\phi_k(y) \;=\; \lambda_j\,\phi_j(y).$$

Substituting back into our initial equation (4.57), we find that

$$a_j^* \;\propto\; \lambda_j \int_{\mathcal{X}} \phi_j(y)\;M(y)\,dy \;=\; \lambda_j\,\widetilde{a}_j,$$

where $\widetilde{a}_j$ are the basis expansion coefficients of M. Since the space $\mathcal{X}$ is compact, one can see that $M \in L^2(\mathcal{X})$, and hence $\sum_{j=1}^{\infty} \widetilde{a}_j^2 < \infty$. Therefore, we have

$$\sum_{j=1}^{\infty} \frac{(a_j^*)^2}{\lambda_j} \;\propto\; \sum_{j=1}^{\infty} \lambda_j \, \widetilde{a}_j^2 \;<\; +\infty,$$

where we used the fact that $\sum_{j=1}^{\infty} \lambda_j < \infty$.

We now use this bound to control the approximation error (4.56). For any $r = 1, 2, \ldots$, we have

$$\sum_{j=r+1}^{\infty} (a_j^*)^2 \;=\; \sum_{j=r+1}^{\infty} \lambda_j \, \frac{(a_j^*)^2}{\lambda_j} \;\leq\; \lambda_r \sum_{j=r+1}^{\infty} \frac{(a_j^*)^2}{\lambda_j} \;=\; \mathcal{O}(\lambda_r),$$

using the non-increasing nature of the sequence $\{\lambda_j\}_{j=1}^{\infty}$. Consequently, by definition of r^* (4.30), we have

$$\|m_{u\to v}^* - \Pi^{r^*}(m_{u\to v}^*)\|_{L^2}^2 \;=\; \mathcal{O}(\delta),$$

as claimed.

4.6 Experimental Results

In this section, we describe some experimental results that help to illustrate the theoretical predictions discussed in Section 4.4.

4.6.1 Synthetic Data

We begin by running some experiments for a simple model, in which both the node and edge potentials are mixtures of Gaussians. More specifically, we form a graphical model with potential functions of the form

$$\psi_u(y) \;=\; \sum_{i=1}^{3} \pi_{u;i} \, \exp\left(-\frac{(y-\mu_{u;i})^2}{2\sigma_{u;i}^2}\right), \quad \text{for all } u \in \mathcal{V}, \text{ and} \tag{4.58a}$$

$$\psi_{vu}(x,y) \;=\; \sum_{i=1}^{3} \pi_{vu;i} \, \exp\left(-\frac{(x-y)^2}{2\sigma_{vu;i}^2}\right) \quad \text{for all } (u,v) \in \mathcal{E}, \tag{4.58b}$$

where the non-negative mixture weights are normalized (i.e., $\sum_{i=1}^{3} \pi_{vu;i} = \sum_{i=1}^{3} \pi_{u;i} = 1$). For each vertex and edge and for all $i = 1, 2, 3$, the mixture parameters are chosen randomly from uniform distributions over the range $\sigma_{u;i}^2, \sigma_{vu;i}^2 \in (0, 0.5]$ and $\mu_{u;i} \in [-3, 3]$.

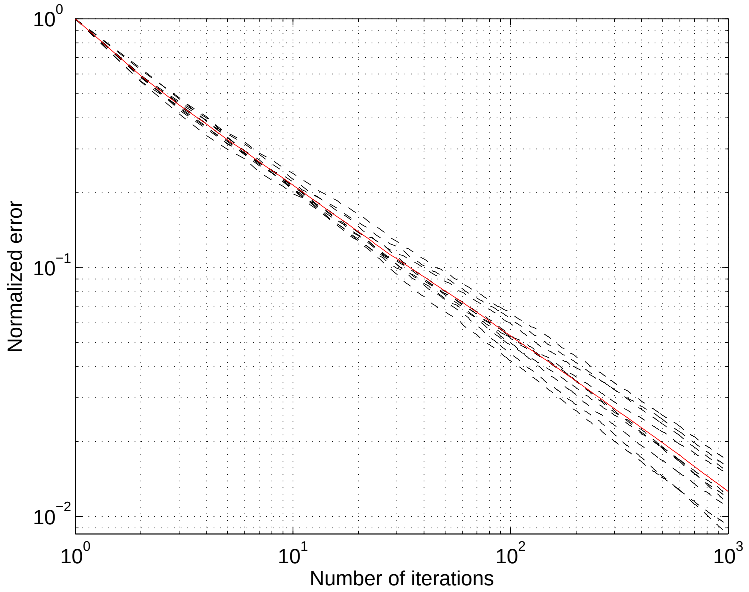


Figure 4.3: Plot of normalized error e^t/e^0 vs. the number of iterations t for 10 different sample paths on a chain of size $n = 100$. The dashed lines are sample paths whereas the solid line is the mean square error. In this experiment node and edge potentials are mixtures of three Gaussians (4.58) and we implemented SOSMP using the first $r = 10$ Fourier coefficients with $k = 5$ samples.

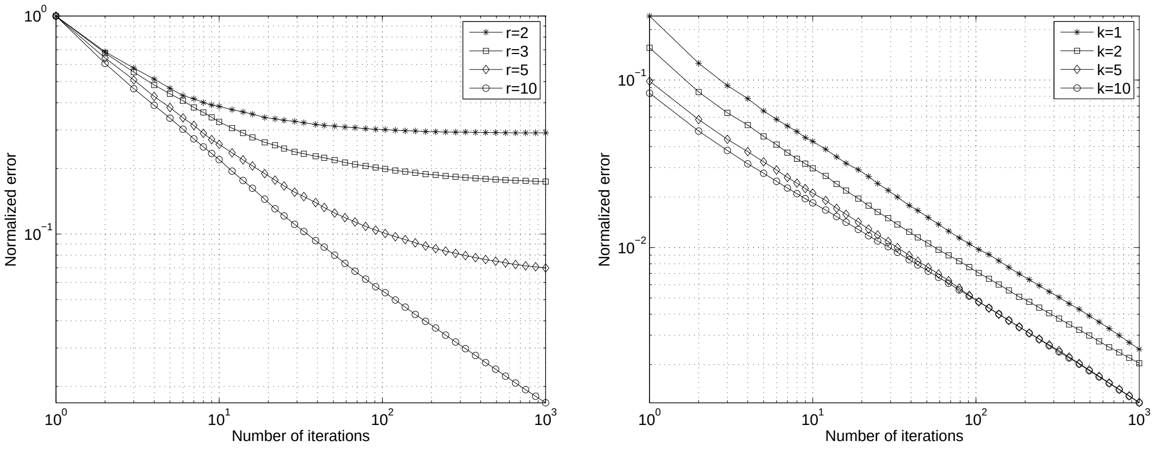
For a chain-structured graph with $n = 100$ nodes, we first compute the fixed point of standard BP, using direct numerical integration to compute the integrals, so to compute (an extremely accurate approximation of) the fixed point m^*. In particular, we approximate the integral update (4.3) with its Riemann sum over the range $\mathcal{X} = [-5, 5]$ and with 100 samples per unit time. We compare this "exact" answer to the approximation obtained by running the SOSMP algorithm using the first $r = 10$ Fourier basis coefficients and $k = 5$ samples. Having run the SOSMP algorithm, we compute the average squared error

$$e^t := \frac{1}{2|\vec{\mathcal{E}}|r} \sum_{(u \to v) \in \vec{\mathcal{E}}} \sum_{j=1}^{r} [a^t_{u \to v;j} - a^*_{u \to v;j}]^2 \tag{4.59}$$

at each time $t = 1, 2, \ldots$.

Figure 4.3 provides plots of the error (4.59) versus the number of iterations for 10 different trials of the SOSMP algorithm. (Since the algorithm is randomized, each path is slightly different.) The plots support our claim of of almost sure convergence, and moreover, the straight lines seen in the log-log plots confirm that convergence takes place at a rate inverse polynomial in t.

In the next few simulations, we test the algorithm's behavior with respect to the number of expansion coefficients r, and number of samples k. In particular, Figure 4.4(a) illustrates the expected error, averaged over several sample paths, vs. the number of iterations for different



(a) (b)

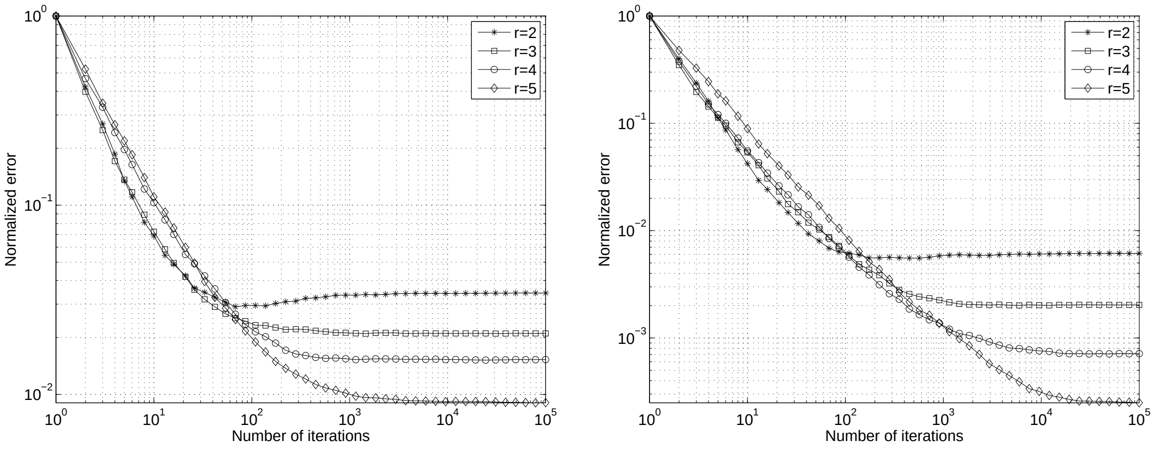
Figure 4.4: Normalized mean squared error $\mathbb{E}[e^t/e^0]$ verses the number of iterations for a Markov chain with $n = 100$ nodes, using potential functions specified by the mixture of Gaussians model (4.58). (a) Behavior as the number of expansion coefficients is varied over the range $r \in \{2, 3, 5, 10\}$ with $k = 5$ samples in all cases. As predicted by the theory, the error drops monotonically with the number of iterations until it hits a floor. The error floor, which corresponds to the approximation error incurred by message expansion truncation, decreases as the number of coefficients r is increased. (b) Mean squared error $\mathbb{E}[e^t]$ verses the number of iterations t for different number of samples $k \in \{1, 2, 5, 10\}$, in all cases using $r = 10$ coefficients. Increasing the number of samples k results in a downward shift in the error.

number of expansion coefficients $r \in \{2, 3, 5, 10\}$ when $k = 5$ fixed; whereas Figure 4.4(b) depicts the expected error vs. the number of iterations for different number of samples $k \in \{1, 2, 5, 10\}$ when $r = 10$ is fixed. As expected, in Figure 4.4(a), the error decreases monotonically, with the rate of $1/t$, till it hits a floor corresponding the offset incurred by the approximation error. Moreover, the error floor decreases with the number of expansion coefficients. On the other hand, in Figure 4.4(b), increasing the number of samples causes a downward shift in the error. This behavior is also expected since increasing the number of samples reduces the variance of the empirical expectation in equation (4.14).

In our next set of experiments, still on a chain with $n = 100$ vertices, we test the behavior of the SOSMP algorithm on graphs with edge potentials of varying degrees of smoothness. In all cases, we use node potentials from the Gaussian mixture ensemble (4.58) previously discussed, but form the edge potentials in terms of a family of kernel functions. More specifically, consider the basis functions

$$\phi_j(x) = \sin\left(\frac{(2j-1)\pi(x+5)}{10}\right) \qquad \text{for } j = 1, 2, \ldots.$$

each defined on the interval $[-5, 5]$. It is straightforward that the family $\{\phi_j\}_{j=1}^{\infty}$ forms an



(a) (b)

Figure 4.5: Plot of the estimation error e^t/e^0 verses the number of iterations t for the cases of (a) $\alpha = 0.1$ and (b) $\alpha = 1$. The BP messages are smoother when $\alpha = 1$, and accordingly the SOSMP estimates are more accurate with the same number of expansion coefficients. Moreover, the error decays with the rate of $1/t$ till it hits a floor corresponding to the approximation error incurred by truncating the message expansion coefficients.

orthonormal basis of $L^2[-5, 5]$. We use this basis to form the edge potential functions

$$\psi_{vu}(x, y) = \sum_{j=1}^{1000} \left(\frac{1}{j}\right)^{\alpha} \phi_j(x)\, \phi_j(y), \tag{4.60}$$

where $\alpha > 0$ is a parameter to be specified. By construction, each edge potential is a positive semidefinite kernel function satisfying the α-polynomial decay condition (4.32).

Figure 4.5 illustrate the error curves for two different choices of the smoothness parameter: panel (a) shows $\alpha = 0.1$, whereas panel (b) shows $\alpha = 1$. For the larger value of α shown in panel (b), the messages in the BP algorithm are smoother, so that the SOSMP estimates are more accurate with the same number of expansion coefficients. Moreover, similar to what we have observed previously, the error decays with the rate of $1/t$ till it hits the error floor. Note that this error floor is lower for the smoother kernel ($\alpha = 1$) compared to the rougher case ($\alpha = 0.1$); note the difference in axis scaling between panels (a) and (b). Moreover, as predicted by our theory, the approximation error decays faster for the smoother kernel, as shown by the plots in Figure 4.6, in which we plot the final error, due purely to approximation effects, versus the number of expansion coefficients r. The semilog plot of Figure 4.6 shows that the resulting lines have different slopes, as would be expected.

4.6.2 Computer Vision Application

Moving beyond simulated problems, we conclude by showing the SOSMP algorithm in application to a larger scale problem that arises in computer vision—namely, that of optical

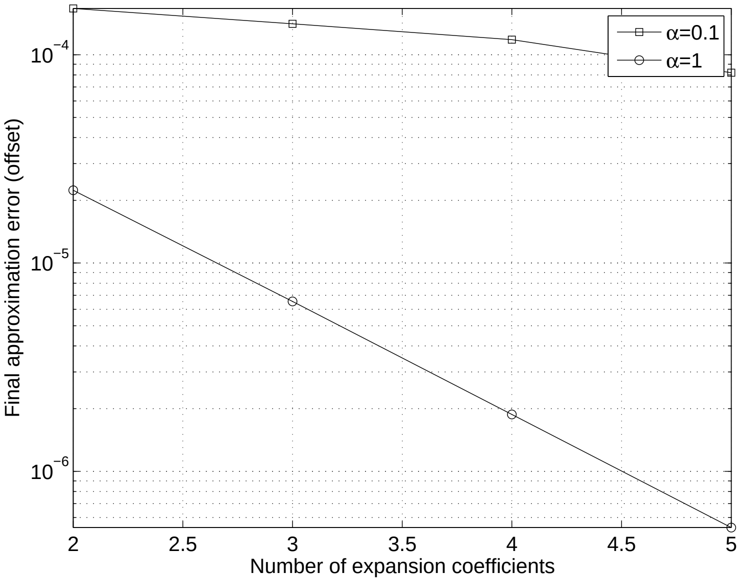


Figure 4.6: Final approximation error vs. the number of expansion coefficients for the cases of $\alpha = 0.1$ and $\alpha = 1$. As predicted by the theory, the error floor decays with a faster pace for the smoother edge potential.

flow estimation [10]. In this problem, the input data are two successive frames of a video sequence. We model each frame as a collection of pixels arrayed over a $\sqrt{n} \times \sqrt{n}$ grid, and measured intensity values at each pixel location of the form $\{I(i,j), I'(i,j)\}_{i,j=1}^{\sqrt{n}}$. Our goal is to estimate a 2-dimensional motion vector $x_u = (x_{u,1}, x_{u,2})$ that captures the local motion at each pixel $u = (i,j)$, $i,j = 1,2,\ldots,\sqrt{n}$ of the image sequence.

In order to cast this optical flow problem in terms of message-passing on a graph, we adopt the model used by Boccignone et al. [15]. We model the local motion X_u as a 2-dimensional random vector taking values in the space $\mathcal{X} = [-d,d] \times [-d,d]$, and associate the random vector X_u with vertex u, in a 2-dimensional grid (see Figure 2.2). At node $u = (i,j)$, we use the change between the two image frames to specify the node potential

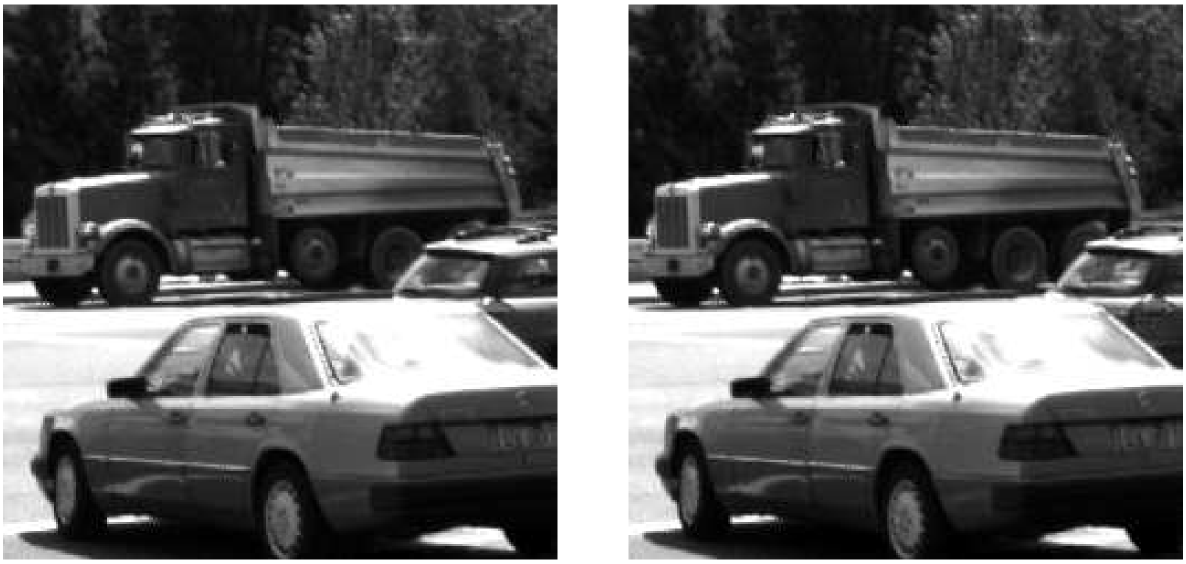
$$\psi_u(x_{u,1}, x_{u,2}) \;\propto\; \exp\left(-\frac{(I(i,j) - I'(i+x_{u,1},\ j+x_{u,2}))^2}{2\sigma_u^2}\right).$$

On each edge (v,u), we introduce the potential function

$$\psi_{vu}(x_v, x_u) \;\propto\; \exp\left(-\frac{\|x_v - x_u\|^2}{2\sigma_{vu}^2}\right),$$

which enforces a type of smoothness prior over the image.

To estimate the motion of a truck, we applied the SOSMP algorithm using the 2-dimensional Fourier expansion as our orthonormal basis to two 250×250 frames from a



(a) (b)

Figure 4.7: Two frames, each of dimension 250 × 250 pixels, taken from a video sequence of moving cars.

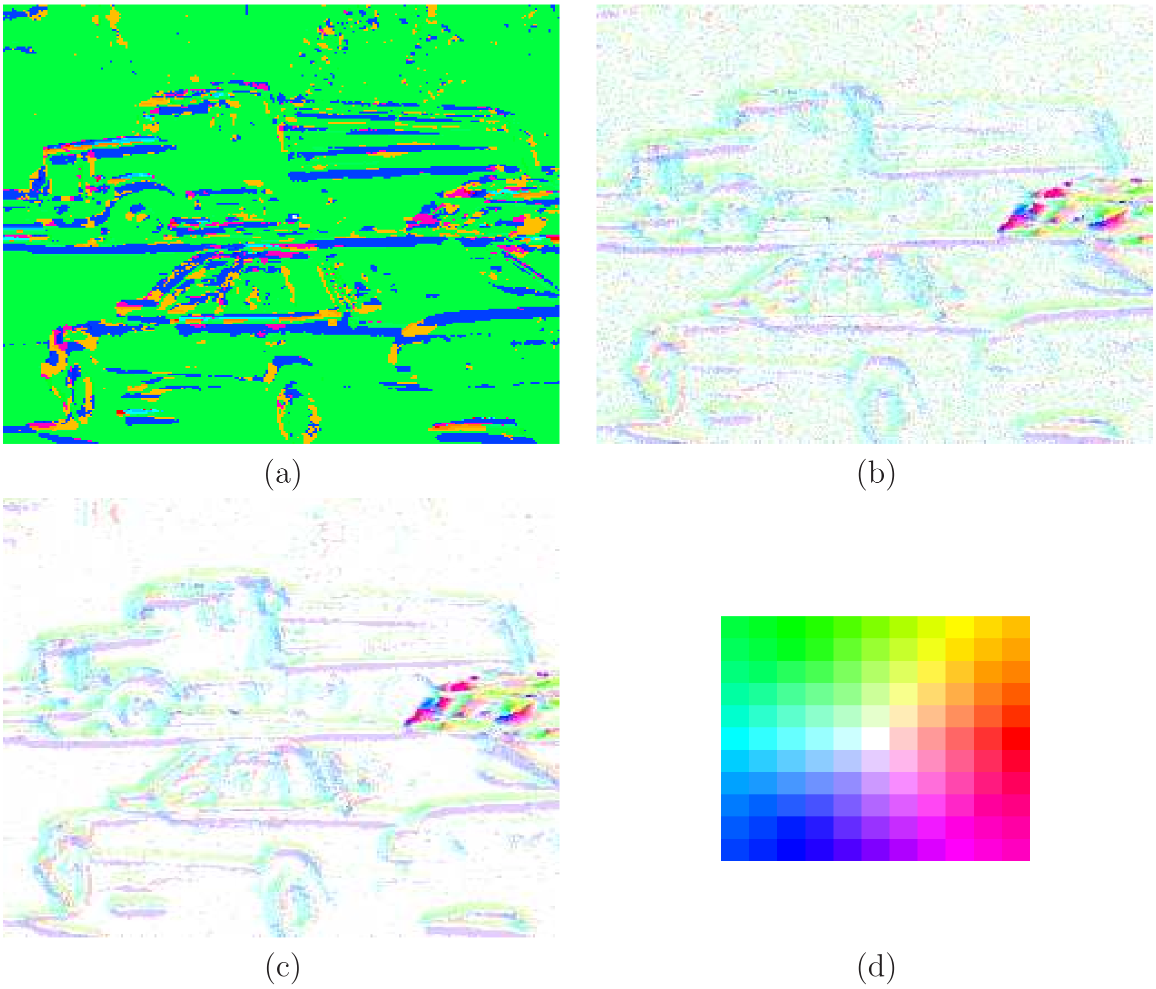
truck video sequence (see Figure 4.7). We apply the SOSMP algorithm using the first $r = 9$ coefficients and $k = 3$ samples. Figure 4.8 shows the HSV (hue, saturation, value) codings of the estimated motions after $t = 1, 10, 40$ iterations, in panels (a), (b) and (c) respectively. Panel (d) provides an illustration of the HSV encoding: hue is used to represent the angular direction of the motion whereas the speed (magnitude of the motion) is encoded by the saturation (darker colors meaning higher speeds). The initial estimates of the motion vectors are noisy, but it fairly rapidly converges to a reasonable optical flow field. (To be clear, the purpose of this experiment is not to show the effectiveness of SOSMP or BP as a particular method for optical flow, but rather to demonstrate its correctness and feasibility of the SOSMP in an applied setting.)

4.7 Conclusion

In this chapter, we have presented and analyzed the SOSMP algorithm for running BP in models with continuous variables. It is based on two forms of approximation: a *deterministic approximation* that involves projecting messages onto the span of r basis functions, and a *stochastic approximation* that involves approximating integrals by Monte Carlo estimates. These approximations, while leading to an algorithm with substantially reduced complexity, are also controlled: we provide upper bounds on the convergence of the stochastic error, showing that it goes to zero as $\mathcal{O}(\log t/t)$ with the number of iterations, and also control on the deterministic error. For graphs with relatively smooth potential functions, as reflected in the decay rate of their basis coefficients, we provided a quantitative bound on the total number of basic arithmetic operations required to compute the BP fixed point to within δ-accuracy. We illustrated our theoretical predictions using experiments on simulated graphical models, as well as in a real-world instance of optical flow estimation.

Our work leaves open a number of interesting questions. First, although we have focused exclusively on models with pairwise interactions, it should be possible to develop forms of SOSMP for higher-order factor graphs. Second, the bulk of our analysis was performed under a type of contractivity condition, as has been used in past work [113, 49, 78, 103] on convergence of the standard BP updates. However, we suspect that this condition might be somewhat relaxed, and doing so would demonstrate applicability of the SOSMP algorithm to a larger class of graphical models.[2]

[2] The materials of this chapter have been published in [86] and submitted to [85].



(a)

(b)

(c)

(d)

Figure 4.8: Color coded images of the estimated motion vectors after (a) $t = 1$, (b) $t = 10$, (c) $t = 40$ iterations. Panel (d) illustrates the HSV color coding of the flow. The color hue is used to encode the angular dimension of the motion, whereas the saturation level corresponds to the speed (length of motion vector). We implemented the SOSMP algorithm by expanding in the two-dimensional Fourier basis, using $r = 9$ coefficients and $k = 3$ samples. Although the initial estimates are noisy, it converges to a reasonable optical flow estimate after around 40 iterations.

Chapter 5

Efficient Distributed Averaging

5.1 Introduction

As disscused in Chappter 1, the problem of network-constrained averaging is to compute the average of a set of numbers distributed throughout a graph, using an algorithm that is allowed to pass messages only along edges of the graph. The focus of this chapter is a noisy version of this problem, in which inter-node communication is modeled by an additive white Gaussian noise (AWGN) channel. There is now an extensive literature on network-averaging, consensus problems, as well as distributed optimization and estimation (e.g., see the papers [18, 31, 28, 116, 57, 8, 11, 12, 22, 74, 73]). The bulk of the earlier work has focused on the noiseless variant, in which communication between nodes in the graph is assumed to be noiseless. A more recent line of work has studied versions of the problem with noisy communication links (e.g., see the papers [45, 37, 95, 7, 109, 56, 9] and references therein).

Given the communication randomness, any algorithm is necessarily stochastic, and the corresponding sequence of random variables can be analyzed in various ways. The simplest question to ask is whether the algorithm is consistent—that is, does it compute an approximate average or achieve consensus in an asymptotic sense for a given fixed graph? A more refined analysis seeks to provide information about this convergence rate. In this chapter, we do so by posing the following question: for a given algorithm, how does number of iterations required to compute the average to within δ-accuracy scale as a function of the graph topology and number of nodes n? For obvious reasons, we refer to this as the *network scaling* of an algorithm, and we are interested in finding an algorithm that has optimal scaling law.

The issue of network scaling has been studied by a number of authors in the noiseless setting, in which the communication between nodes is perfect. Of particular relevance here is the work of Benezit et al. [12], who in the case of perfect communication, provided a scheme that has essentially optimal message scaling law for random geometric graphs. A portion of the method proposed in this chapter is inspired by their scheme, albeit with suitable extensions to multiple paths that are essential in the noisy setting. The issue of network scaling has also been studied in the noisy setting; in particular, past work by Rajagopal and Wainwright [95] analyzed a damped version of the usual consensus updates, and provided scalings of the iteration number as a function of the graph topology and size. However, our new algorithm has much better scaling than the method [95].

The main contributions of this work are the development of a novel two-phase (outer and inner phase) algorithm for network-constrained averaging with noise, and establishing the near-optimality of its network scaling. At a high level, the outer phase of our algorithm produces a sequence of iterates $\{\theta^\tau\}_{\tau=0}^\infty$ based on a recursive linear update with decaying step size, as in stochastic approximation methods. The system matrix in this update is a time-varying and random quantity, whose structure is determined by the updates within the inner phase. These inner rounds are based on establishing multiple paths between pairs of nodes, and averaging along them simultaneously. By combining a careful analysis of the spectral properties of this random matrix with stochastic approximation theory, we prove that this two-phase algorithm computes a δ-accurate version of the average using a number of iterations that grows with the graph diameter (up to logarithmic factors).[1] As we discuss in more detail following the statement of our main result, this result is optimal up to logarithmic factors, meaning that no algorithm can be substantially better in terms of network scaling.

The remainder of this chapter is organized as follows. We begin in Section 5.2 with background and formulation of the problem. In Section 5.3, we describe our algorithm, and state various theoretical guarantees on its performance. We then provide the proof of our main result in Section 5.4. Section 5.5 is devoted to some simulation results that confirm the sharpness of our theoretical predictions.

5.2 Background and Problem Statement

We begin in this section by introducing necessary background and setting-up the problem more precisely.

5.2.1 Network-Constrained Averaging

Consider a collection $\{\theta_i^0\}_{i=1}^n$ of n numbers. In statistical settings, these numbers would be modeled as independent identically distributed (i.i.d.) draws from an unknown distribution $\mathbb{Q}$ with mean μ. In a centralized setting, a standard estimator for the mean is the sample average $\bar{\theta} := \big[\sum_{i=1}^n \theta_i^0\big]/n$. When all of the data can be aggregated at a central location, then computation of $\bar{\theta}$ is straightforward. In this paper, we consider the network-constrained version of this estimation problem, modeled by an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ that consists of a vertex set $\mathcal{V} = \{1, 2, \ldots, n\}$, and a collection of edges $\mathcal{E}$, where $(i, j) \in \mathcal{E}$ if and only if vertices i and j are connected. For $i \in \mathcal{V}$, we view each measurement θ_i^0 as associated with

[1] Recall that the graph diameter is the minimal number of edges needed to connect any two pairs of nodes in the graph.

vertex i. (For instance, in the context of sensor networks, each vertex would contain a mote and collect observations of the environment.) The edge structure of the graph enforces communication constraints on the processing: in particular, the presence of edge (i, j) indicates that it is possible for sensors i and j to exchange information via a noisy communication channel. Conversely, sensor pairs that are *not* joined by an edge are not permitted to communicate directly.[2] Every node has a synchronized internal clock, and acts at discrete times $t = 1, 2, \ldots$. For any given pair of sensors $(i, j) \in \mathcal{E}$, we assume that the message sent from i to j is perturbed by an independent identically distributed $N(0, \sigma^2)$ variate. Although this additive white Gaussian noise (AWGN) model is more realistic than a noiseless model, it is conceivable that other stochastic channel models might be more suitable for certain types of sensor networks, and we leave this exploration for future research.

Given this set-up, of interest to us are stochastic algorithms that generate sequences $\{\theta^t\}_{t=0}^{\infty}$ of iterates contained within $\mathbb{R}^n$, and we require that the algorithm be *graph-respecting*, meaning that in each iteration, it is allowed to send at most one message for each direction of every edge $(i, j) \in \mathcal{E}$. At time t, we measure the distance between θ^t and the desired average $\bar{\theta}$ via the average (per node) mean-squared error, given by

$$\text{MSE}(\theta^t) \;:=\; \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}\big[(\theta_i^t - \bar{\theta})^2\big]. \tag{5.1}$$

In this chapter, our goal is for every node to compute the average $\bar{\theta}$ up to an error tolerance δ. In addition, we require almost sure consensus among nodes, meaning

$$\mathbb{P}\big(\theta_i^t = \theta_j^t \quad \forall\, i, j = 1, 2, \ldots, n\big) \to 1 \quad \text{as } t \to \infty.$$

Our primary goal is in characterizing the rate of convergence as a function of the graph topology and the number of nodes, to which we refer as the *network-scaling function* of the algorithm. More precisely, in order to study this network scaling, we consider sequences of graphs $\{\mathcal{G}_n\}$ indexed by the number of nodes n. For any given algorithm (defined for each graph $\mathcal{G}_n$) and a fixed tolerance parameter $\delta > 0$, our goal is to determine bounds on the quantity

$$T_{\mathcal{G}}(n; \delta) \;:=\; \inf\big\{t = 1, 2, \ldots \mid \text{MSE}(\theta^t) \;\leq\; \delta\big\}. \tag{5.2}$$

Note that $T_{\mathcal{G}}(n; \delta)$ is a stopping time, given by the smallest number of iterations required to obtain mean-squared error less than δ on a graph of type $\mathcal{G}$ with n nodes.

5.2.2 Graph topologies

Of course, the question that we have posed will depend on the graph type, and this paper analyzes three types of graphs, as shown in Figure 5.1. The first two graphs have regular

[2] Moreover, since the edges are undirected, there is no difference between edge (i, j) and (j, i); moreover, we exclude self-edges, meaning that $(i, i) \notin \mathcal{E}$ for all $i \in \mathcal{V}$.

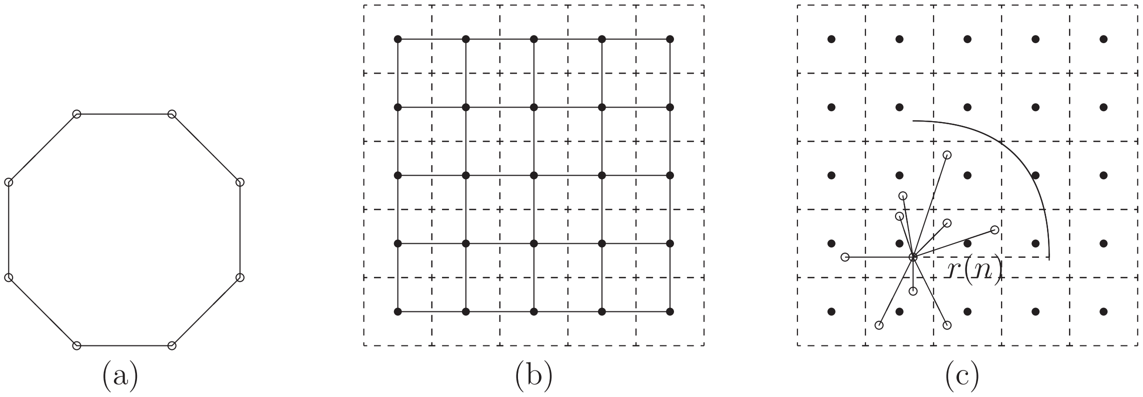


Figure 5.1: Illustration of graph topologies. (a) A single cycle graph. (b) Two-dimensional grid with four-nearest-neighbor connectivity. (c) Illustration of a random geometric graph (RGG). Two nodes are connected if their distance is less than $r(n)$. The solid circles represent the center of squares.

topologies: the single cycle graph in panel (a) is degree two-regular, and the two-dimensional grid graph in panel (b) is degree four-regular. In addition, we also analyze an important class of random graphs with irregular topology, namely the class of random geometric graphs. As illustrated in Figure 5.1(c), a random geometric graph (RGG) in the plane is formed according by placing n nodes uniformly at random in the unit square $[0,1] \times [0,1]$, and the connecting two nodes if their Euclidean distance is less than some radius $r(n)$. It is known that an RGG will be connected with high probability as long as $r(n) = \Omega(\sqrt{\log n/n})$ that is there exists a constant c such that $r(n) \geq c\sqrt{\log n/n}$. See Penrose [94] for discussion of this and other properties of random geometric graphs.

A key graph-theoretic parameter relevant to our analysis is the *graph diameter*, denoted by $D_n = \text{diam}(\mathcal{G}_n)$. The path distance between any pair of nodes is the length of the shortest path joining them in the graph, and by definition, the graph diameter is the maximum path distance taken over all node pairs in the graph. It is straightforward to see that $D_n = \Theta(n)$ for the single cycle graph, and that $D_n = \Theta(\sqrt{n})$ for the two-dimensional grid. For a random geometric graph with radius chosen to ensure connectivity, it is known that $D_n = \Theta(\sqrt{n/\log n})$.[3]

Finally, in order to simplify the routing problem explained later, we divide the unit square into m^2 sub-regions (squares) of side length $\sqrt{1/n}$ in case of grid, and for some constant $c > 0$, of side length $\sqrt{c\,\log n/n}$ in case of RGG. Here we have $m = \sqrt{n}$ for the regular grid, and $m = \sqrt{n/(c\log n)}$ for the RGG. We also assume that each node knows its location and is aware of the center of these m^2 sub-regions namely (x_i, y_j) $i, j = 1, 2, \ldots, m$. As a convention, we assume that (x_1, y_1) is the left bottom square, to which we refer to as the

[3] The notation $f(n) = \mathcal{O}(g(n))$ means that there exists some constant $c \in (0, \infty)$ and $n_0 \in \mathbb{N}$ such $f(n) \leq cg(n)$ for all $n \geq n_0$, whereas $f(n) = \Omega(g(n))$ means that $f(n) \geq c'g(n)$ for all $n \geq n_0$. The notation $f(n) = \Theta(g(n))$ means that $f(n) = \mathcal{O}(g(n))$ and $f(n) = \Omega(g(n))$.

first square.

By construction, in a regular grid, each square will contain one and only one node which is located at the center of the square. Also, from known properties of RGGs [94, 43], each of the given subregions will contain at least one node with high probability (w.h.p.). Moreover, an RGG is regular w.h.p, meaning that each square contains $\Theta(\log n)$ nodes (see Lemma 1 in the paper [31]). Accordingly, in the remainder of the paper, we assume without loss of generality that any given RGG is regular. Note that by construction, the transmission radius $r(n)$ is selected so that each node in each square is connected to every other node in four adjacent squares.

5.3 Proposed Algorithm and its Properties

In this section we state our main result which is followed by a detailed description of the proposed algorithm.

5.3.1 Theoretical Guarantees

Our main result guarantees the existence of a graph-respecting algorithm with desirable properties. Recall the definition of the graph respecting scheme, as well as the definition of our AWGN channel model given in Section 5.2. In the following statement, the quantity c_0 denotes a universal constant, independent of n, δ, and σ^2.

Theorem 11. *For the communication model in which each link is an AWGN channel with variance σ^2, there is a graph-respecting algorithm such that:*

a) Nodes almost surely reach a consensus. More precisely, we have

$$\theta^t \xrightarrow{a.s.} \widetilde{\theta}\,\vec{1} \quad \text{as } t \to \infty, \tag{5.3}$$

for some $\widetilde{\theta} \in \mathbb{R}$.

b) After $T = T_{\mathcal{G}}(n;\delta)$ iterations, the algorithm satisfy the following bounds on the $\mathrm{MSE}(\theta^T)$:

i) For fixed tolerance $\delta > 0$ sufficiently small, we have $\mathrm{MSE}(\theta^T) = \mathcal{O}(\sigma^2\delta)$ after

$$T_{\text{cyc}}(n;\delta) \;\leq\; c_0\, n\, \max\left\{\frac{1}{\delta}\log\frac{1}{\delta}\,,\, \frac{\mathrm{MSE}(\theta^0)}{\sigma^2\delta^2}\right\}$$

iterations for a single cycle graph.

ii) For fixed tolerance $\delta > 0$ sufficiently small, we have $\mathrm{MSE}(\theta^T) = \mathcal{O}(\sigma^2\delta)$ after

$$T_{\text{grid}}(n;\delta) \;\leq\; c_0\, \sqrt{n}\, \max\left\{\frac{1}{\delta}\log\frac{1}{\delta}\,,\, \frac{\mathrm{MSE}(\theta^0)}{\sigma^2\delta^2}\right\}$$

iterations for the regular grid in two dimensions.

iii) Assume that $\delta = \widetilde{\delta}/(\log n)^2$*, for some fixed* $\widetilde{\delta}$ *sufficiently small. Then we have* $\mathrm{MSE}(\theta^T) = \mathcal{O}\big(\sigma^2\widetilde{\delta}\big)$ *after*

$$T_{\mathrm{RGG}}(n;\delta) \;\leq\; c_0\,\sqrt{n(\log n)^3}\,\max\left\{\frac{1}{\widetilde{\delta}}\log\frac{(\log n)^2}{\widetilde{\delta}}\,,\;\frac{\mathrm{MSE}(\theta^0)}{\sigma^2\widetilde{\delta}^2}\right\}$$

iterations for a regular random geometric graph.

Here c_0 *is some constant independent of* n*,* δ*, and* σ^2*, whose value may change from line to line.*

Remarks: A few comments are in order regarding the interpretation of this result. First, it is worth mentioning that the quality of the different links does not have to be the same. Similar arguments apply to the case where noises have different variances. Second, although nodes almost surely reach a consensus, as guaranteed in part (a), this consensus value is not necessarily the same as the sample mean $\bar{\theta}$. The choice of $\widetilde{\theta}$ is intentional to emphasize this point. However, as guaranteed by part (b), this consensus value is within $\sigma^2\delta$ distance of the actual sample mean. Since the sample mean itself represents a noisy estimate of some underlying population quantity, there is little point to computing it to arbitrary accuracy. Third, it is worthwhile comparing part (b) with previous results on network scaling in the noisy setting. Rajagopal and Wainwright [95] analyzed a simple set of damped updates, and showed that $T_{\mathrm{cyc}}(n;\delta) = \mathcal{O}\big(n^2\big)$ for the single cycle, and that $T_{\mathrm{grid}}(n) = \mathcal{O}\big(n\big)$ for the two-dimensional grid. By comparison, the algorithm proposed here and our analysis thereof has removed factors of n and $\sqrt{n}$ from this scaling.

5.3.2 Optimality of the Results

As we now discuss, the scalings in Theorem 11 are optimal for the cases of cycle and grid and near-optimal (up to logarithmic factor) for the case of RGG. In an adversarial setting, any algorithm needs at least $\Omega(D_n)$ iterations, where D_n denotes the graph diameter, in order to approximate the average; otherwise, some node will fail to have any information from some subset of other nodes (and their values can be set in a worst-case manner). Theorem 11 provides upper bounds on the number of iterations that, at most, are within logarithmic factors of the diameter, and hence are also within logarithmic factors of the optimal latency scaling law. For the graphs given here, the scalings are also optimal in a non-adversarial setting, in which $\{\theta_i^0\}_{i=1}^n$ are modeled as chosen i.i.d. from some distribution. Indeed, for a given node $j \in \mathcal{V}$, and positive integer t, we let $\mathcal{N}(j;t)$ denote the depth t neighborhood of j, meaning the set of nodes that are connected to j by a path of length at most t. We then define the graph spreading function $\psi_{\mathcal{G}}(t) = \min_{j\in\mathcal{V}} |\mathcal{N}(j;t)|$. Note that the function $\psi_{\mathcal{G}}$ is non-decreasing, so that we may define its inverse function $\psi_{\mathcal{G}}^{-1}(s) = \inf\{t \mid \psi_{\mathcal{G}}(t) \leq s\}$. As some examples:

- for a cycle on n nodes, we have $\psi_{\mathcal{G}}(t) = 2t$, and hence $\psi_{\mathcal{G}}^{-1}(s) = s/2$.

- for a n-grid in two dimensions, we have the upper bound $\psi_{\mathcal{G}}(t) \leq 2t^2$, and hence the lower bound $\psi_{\mathcal{G}}^{-1}(s) \geq \sqrt{s/2}$.
- for a random geometric graph (RGG), we have the upper bound $\psi_{\mathcal{G}}(t) = \Theta(t^2 \log n)$, which implies the lower bound $\psi_{\mathcal{G}}^{-1}(s) = \Theta(\sqrt{s/\log n})$

After t steps, a given node can gather the information of at most $\psi_{\mathcal{G}}(t)$ nodes. For the average based on $\psi_{\mathcal{G}}(t)$ nodes to be comparable to $\bar{\theta}$, we require that $\psi_{\mathcal{G}}(t) = \Omega(n)$, and hence the iteration number t should be at least $\Omega(\psi_{\mathcal{G}}^{-1}(n))$. For the three graphs considered here, this leads to the same conclusion, namely that $\Omega(D_n)$ iterations are required. We note also that using information-theoretic techniques, Ayaso et al. [6] proved a lower bound on the number of iterations for a general graph in terms of the Cheeger constant [25]. For the graphs considered here, the Cheeger constant is of the order of the diameter.

5.3.3 Description of the Algorithm

We now describe the algorithm that achieves the bounds stated in Theorem 11. At the highest level, the algorithm can be divided into two types of phases: an inner phase, and an outer phase. The outer phase produces a sequence of iterates $\{\theta^\tau\}$, where $\tau = 0, 1, 2, \ldots$ is the outer time scale parameter. By design of the algorithm, each update of the outer parameters requires a total of M message-passing rounds (these rounds corresponding to the inner phase), where in each round the algorithm can pass at most two messages per edge (one for each direction). To put everything in a nutshell, the algorithm is based on establishing multiple routes, averaging along them in an inner phase and updating the estimates based on the noisy version of averages along routes in an outer phase. Consequently, if we use the estimate θ^τ, then in the language of Theorem 11, it corresponds to $T = M\tau$ rounds of message-passing. Our goal is to establish upper bounds on T that guarantee the MSE is $\mathcal{O}(\sigma^2\delta)$. Figure 5.2 illustrates the basic operations of the algorithm.

Outer phase

In the outer phase, we produce a sequence of iterates $\{\theta^\tau\}_{\tau=1}^\infty$ according to the recursive update

$$\theta^{\tau+1} = \theta^\tau - \epsilon^\tau \big\{ L^\tau\, \theta^\tau + v^\tau \big\}. \tag{5.4}$$

Here $\{\epsilon^\tau\}_{\tau=1}^\infty$ is a sequence of positive decreasing step sizes. For a given precision, δ, we set $\epsilon^\tau = 1/(\tau + 1/\delta)$. For each τ, the quantity $L^\tau \in \mathbb{R}^{n\times n}$ is a random matrix, whose structure is determined by the inner phase, and $v^\tau \in \mathbb{R}^n$ is an additive Gaussian term, whose structure is also determined in the inner phase. As will become clear in the sequel, even though L and v are dependent, they are both independent of θ. Moreover, given L, the random vector v is Gaussian with bounded variance.

Two-phase algorithm for distributed consensus:

- Inner phase:
  - Deciding the averaging direction
  - Choosing the head nodes
  - Establishing the routes
  - Averaging along the routes
- Outer phase:
  - Based on the averages along the routes, update the estimates according to

$$\theta^{\tau+1} \;=\; \theta^{\tau} \;-\; \epsilon^{\tau}\big\{L^{\tau}\,\theta^{\tau} \;+\; v^{\tau}\big\}$$

Figure 5.2: Basic operations of a two-phase algorithm for distributed consensus.

Inner phase

The inner phase is the core of the algorithm and it involves a number of steps, as we describe here. We use $s = 1, 2, \ldots, M$ to index the iterations within any inner phase, and use $\{\gamma^s\}_{s=1}^{M}$ to denote the sequence of inner iterates within $\mathbb{R}^n$. For the inner phase corresponding to outer update from $\theta^{\tau} \rightarrow \theta^{\tau+1}$, the inner phase takes the initialization $\gamma^1 \leftarrow \theta^{\tau}$, and then reduces as output $\gamma^M \rightarrow \theta^{\tau+1}$ to the outer iteration. In more detail, the inner phase can be broken down into three steps, which we now describe in detail.

Step 1, deciding the averaging direction: The first step is to choose a direction in which to perform averaging. In a single cycle graph, since left and right are viewed as the same, there is only one choice, and hence nothing to be decided. In contrast, the grid or RGG graphs require a decision-making phase, which proceeds as follows. One node in the first (bottom left) square, wakes up and chooses uniformly at random to send in the horizontal or vertical direction. We code this decision using the random variable $\zeta \in \{-1, 1\}$, where $\zeta = -1$ (respectively $\zeta = +1$) represents the horizontal (respectively vertical) direction. To simplify matters, we assume in the remainder of this description that the averaging direction is horizontal, with the modifications required for vertical averaging being standard.

Step 2, choosing the head nodes: This step applies only to the grid and RGG graphs. Given our assumption that the node in the first square has chosen the horizontal direction, it then passes a token message to a randomly selected node in the above adjacent square.

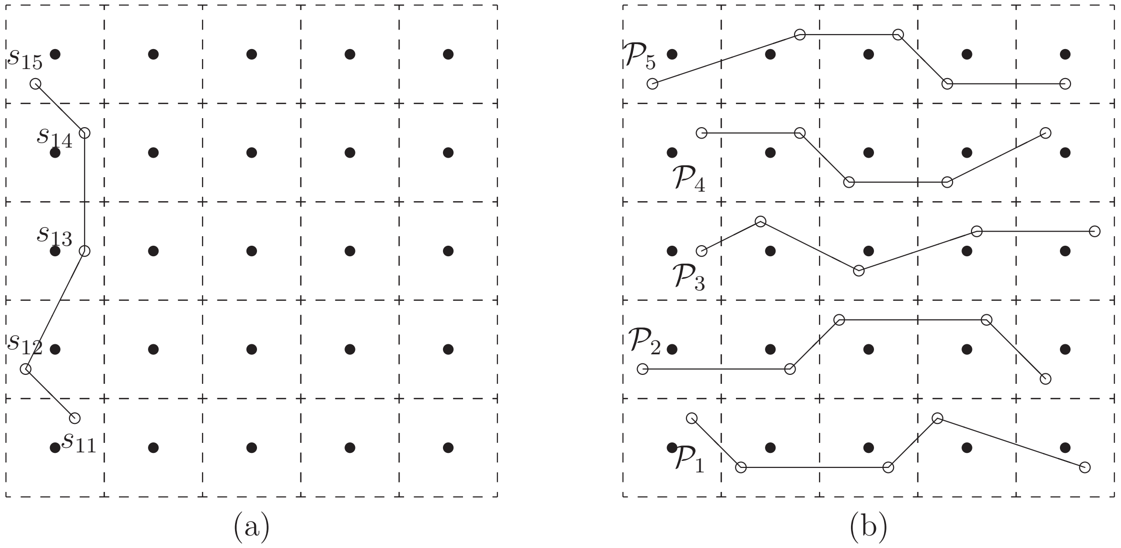


Figure 5.3: (a) The node labeled s_{11} in the first square, chooses the horizontal direction for averaging ($\zeta = -1$); it passes the token vertically to inform other nodes to average horizontally. Nodes who receive the token pass it to another node in the above adjacent square. (b) The head nodes s_{1j}, $j = 1, 2, \ldots, m$, as determined in the first step, establish routes horizontally ($\mathcal{P}_j$, $j = 1, 2, \ldots, m$) and then average along these paths.

The purpose of this token is to determine which node (referred to as the head node) should be involved in establishing the route passing through the given square. After receiving the token, the receiving node passes it to another randomly selected node in the above adjacent square and so on. Note that in the special case of grid, there is only one node in each square, and so no choices are required within squares. After m rounds, one node in each square $(x_1, y_j), j = 1, 2, \ldots, m$ $((x_i, y_1), i = 1, 2, \ldots, m)$ receives the token, as illustrated in Figure 5.3. Note that again in a single cycle graph, there is nothing to be decided, since the direction and head nodes are all determined.

Step 3, establishing routes and averaging: In this phase, each of the head nodes establishes a horizontal path, and then perform averaging along the path, as illustrated in Figure 5.3(b). This part of algorithm involves three substeps, which we now describe in detail.

- For $j = 1, 2, \ldots, m$, each head node s_{1j} selects a node s_{2j} uniformly at random (u.a.r.) from within the right adjacent square, and passes to it the quantity γ_{1j}^1. Given the Gaussian noise model, node s_{2j} then receives the quantity

$$\widetilde{\gamma}_{1j}^1 = \gamma_{1j}^1 + v_{1j}, \quad \text{where } v_{1j} \sim N(0, \sigma^2),$$

and then updates its own local variable as $\gamma_{2j}^2 = \gamma_{2j}^1 + \widetilde{\gamma}_{1j}^1$. We then iterate this same procedure—that is, node s_{2j} selects another s_{3j} u.a.r. from its right adjacent square, and passes the message γ^2_{2j}. Overall, at round i of this update procedure, we have

$$\gamma^{i+1}_{(i+1)j} \;=\; \gamma^{i}_{(i+1)j} + \widetilde{\gamma}^{i}_{ij},$$

where $\widetilde{\gamma}^i_{ij} = \gamma^i_{ij} + v_{ij}$, and $v_{ij} \sim N(0, \sigma^2)$. At the end of round m, node s_{mj} can compute a noisy version of the average along the path $\mathcal{P}_j : s_{1j} \to s_{2j} \to \cdots \to s_{mj}$, in particular via the rescaled quantity

$$\eta_j := \frac{\gamma^m_{mj}}{m} \;=\; \frac{1}{m}\sum_{l=1}^{m} \theta^{\tau}_{s_{lj}} \;+\; v_j \quad \text{for } j = 1, 2, \ldots, m.$$

Here the variable $v_j \sim N(0, \sigma^2/m)$, since the noise variables associated with different edges are independent.

- At this point, for each $j = 1, 2, \ldots, m$, each node s_{mj} which has the noisy version, η_j, of the path average along route $\mathcal{P}_j$; can share this information with other nodes in the path by sending η_j back to the head node. A naive way to do this is as follows: node s_{mj} makes m copies of η_j—namely, $\eta^{(l)}_j = \eta_j$, $l = 1, 2, \ldots, m$—and starts transmitting one copy at a time back to the head node. Nodes along the path simply forward what they receive, so that after $m - i + m - 1$ time steps, node s_{ij} receives m noisy copies of the average, $\widetilde{\eta}^{(l)}_{ij} = \eta^{(l)}_j + v^{(l)}_{ij}$ where $v^{(l)}_{ij} \sim N(0, (m - i)\sigma^2)$. Averaging the m copies, node s_{ij} can compute the quantity

$$\gamma^{3m-i-1}_{ij} \;:=\; \frac{1}{m}\sum_{l=1}^{m} \widetilde{\eta}^{(l)}_{ij} \;=\; \frac{1}{m}\sum_{l=1}^{m} \theta^{\tau}_{s_{lj}} \;+\; w_{ij},$$

where $w_{ij} = v_j + \big(\sum_{l=1}^m v^{(l)}_{ij}\big)/m$. Since the noise on different links and different time steps are independent Gaussian random variables, we have $w_{ij} \sim N(0, \sigma^2_i)$, with

$$\sigma^2_i \;=\; \frac{1}{m}\,\sigma^2 \;+\; \Big(1 - \frac{i}{m}\Big)\,\sigma^2 \;=\; \Big(1 - \frac{(i-1)}{m}\Big)\,\sigma^2 \;\le\; \sigma^2.$$

Therefore, at the end of $M = \Theta(m)$ rounds, for each $j = 1, 2, \ldots, m$, all nodes have the average of the estimates in the path $\mathcal{P}_j$ that is perturbed by Gaussian noise with variance at most σ^2. Since $m = \Theta(D_n)$, we have $M = \Theta(D_n)$.

- At the end of the inner phase τ, nodes that were involved in a path use their estimate of the average along the path to update θ^τ, while estimate of the nodes that were not involved in any route remain the same. A given node s_{ij} on a path updates its estimate via

$$\theta^{\tau+1}_{s_{ij}} \;=\; \big(1 - \epsilon^{\tau}_1\big)\,\theta^{\tau}_{s_{ij}} \;+\; \epsilon^{\tau}_1\,\gamma^{3m}_{ij}, \tag{5.5}$$

where $\epsilon^\tau_1 = \mathcal{O}\big(1/(\tau + 1/\delta)\big)$. On the other hand, using $\langle\cdot,\,\cdot\rangle$ to denote the Euclidean inner product, we have $\gamma^{3m}_{ij} = \langle w,\, \theta^\tau\rangle + v_{s_{ij}}$, where w is the averaging vector of the route

$\mathcal{P}_j$ with the entries $w(s_{lj}) = 1/m$ for $l = 1, 2, \ldots, m$, and zero otherwise. Combining the scalar updates (5.5) yields the matrix-form update

$$\theta^{\tau+1} \; = \; \theta^{\tau} \; - \; \epsilon_1^{\tau} \left[(I - W^{\tau})\, \theta^{\tau} \; + \; v_1^{\tau} \right], \tag{5.6}$$

where the matrix $W^{\tau} = W(\tau; \mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_m, \zeta)$ is a random averaging matrix induced by the choice of routes $\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_m$ and the random directions ζ. The noise vector $v_1^{\tau} \sim N(0, C')$ is additive noise. Note that for any given time, the noise at different nodes are correlated via the matrix C', but for different time instants $\tau \neq \tau'$, the noise vectors v_1^{τ} and $v_1^{\tau'}$ are independent. Moreover, from our earlier arguments, we have the upper bound $\max\limits_{i=1,\ldots,n} C'_{ii} \leq \sigma^2$.

5.4 Proof of Theorem 11

We now turn to the proof of Theorem 11. At a high-level, the structure of the argument consists of decomposing the vector $\theta^{\tau} \in \mathbb{R}^n$ into a sum of two terms: a component within the consensus subspace (meaning all values of the vector are identical), and a component in the orthogonal complement. Using this decomposition, the mean-squared error splits into a sum of two terms and we use standard techniques to bound them. As will be shown, these bounds depend on the parameter δ, noise variance, the initial MSE, and finally the (inverse) spectral gap of the update matrix. The final step is to lower bound the spectral gap of our update matrix.

5.4.1 Setting-Up the Proof

Recalling the averaging matrix W^{τ} from the update (5.6), we define the Laplacian matrix $S^{\tau} := I - W^{\tau}$. We then define the average matrix $\overline{W} := \mathbb{E}\big[W^{\tau}\big]$, where the expectation is taken place over the randomness due to the choice of routes;[4] in a similar way, we define the associated (average) Laplacian $\bar{S} := I - \overline{W}$. Finally, we define the rescaled quantities

$$\epsilon^{\tau} \; := \; \lambda_2(\bar{S})\, \epsilon_1^{\tau}, \quad L^{\tau} \; := \; \frac{1}{\lambda_2(\bar{S})}\, S^{\tau}, \quad \text{and} \quad v^{\tau} \; := \; \frac{1}{\lambda_2(\bar{S})}\, v_1^{\tau}, \tag{5.7}$$

where we recall that $\lambda_2(\cdot)$ denotes the second smallest eigenvalue of a symmetric matrix. In terms of these rescaled quantities, our algorithm has the form

$$\theta^{\tau+1} \; = \; \theta^{\tau} \; - \; \epsilon^{\tau} \left[L^{\tau}\, \theta^{\tau} \; + \; v^{\tau} \right], \tag{5.8}$$

[4] For the single cycle graph, there is only one route that involves all the nodes at each round, so W^{τ} is deterministic in this case.

as stated previously in the update equation (5.4). Moreover, by construction, we have $v^\tau \sim N(0, C)$ where $C = C'/[\lambda_2(\bar{S})]^2$. We also, for theoretical convenience, set

$$\epsilon_1^\tau \;=\; \frac{1}{\lambda_2(\bar{S})\,(\tau + 1/\delta)}, \tag{5.9}$$

or equivalently $\epsilon^\tau = 1/(\tau + 1/\delta)$ for $\tau = 1, 2, \ldots$.

We first claim that the matrix $\overline{W}$ is symmetric and (doubly) stochastic. The symmetry follows from the fact that different routes do not collide, whereas the matrix is stochastic because every row of W (depending on whether the node corresponding to that row participates in a route or not) either represents an averaging along a route or is the corresponding row of the identity matrix. Consequently, we can interpret $\overline{W}$ as the transition matrix of a reversible Markov chain. It is an irreducible Markov chain, because within any updating round, there is a positive chance of averaging nodes that are in the same column or row, which implies that the associated Markov chain can transition from one state to any other in at most two steps. Moreover, the stationary distribution of the chain is uniform (i.e., $\pi = \vec{1}/n$).

We now use these properties to simplify our study of the sequence $\{\theta^\tau\}_{\tau=1}^\infty$ generated by the update equation (5.8). Since $\bar{S}$ is real and symmetric, it has the eigenvalue decomposition $\bar{S} = U\Lambda U^*$, where $U = [u_1\ u_2\ \ldots\ u_n]$ is a unitary matrix (that is, $U^*U = I_n$).[5] Moreover, we have $\Lambda = \mathrm{diag}\{\lambda_1(\bar{S}),\, \lambda_2(\bar{S}),\, \ldots,\, \lambda_n(\bar{S})\}$, where $\lambda_i(\bar{S})$ is the eigenvalue corresponding to the eigenvector u_i, for $i = 1, 2, \ldots, n$. Since $\bar{L} = (I - \overline{W})/\lambda_2(\bar{S})$, the eigenvalues of $\bar{L}$ and $\overline{W}$ are related via

$$\begin{aligned}\lambda_i(\bar{L}) \;&=\; \frac{1}{\lambda_2(\bar{S})}\,(1 - \lambda_{n+1-i}(\overline{W}))\\ &=\; \frac{1}{1 - \lambda_{n-1}(\overline{W})}\,(1 - \lambda_{n+1-i}(\overline{W})).\end{aligned}$$

Since the largest eigenvalue of an irreducible Markov chain is one (with multiplicity one) [41], we have $1 = \lambda_n(\overline{W}) > \lambda_{n-1}(\overline{W}) \geq \ldots \geq \lambda_1(\overline{W})$, or equivalently

$$0 \;=\; \lambda_1(\bar{L}) \;<\; \lambda_2(\bar{L}) \;\leq\; \ldots \;\leq\; \lambda_n(\bar{L}),$$

with $\lambda_2(\bar{L}) = 1$. Moreover, we have $\bar{S}\vec{1} = \bar{L}\vec{1} = \vec{0}$, so that the first eigenvector $u_1 = \vec{1}/\sqrt{n}$ corresponds to the eigenvalue $\lambda_1(\bar{L}) = 0$. Let $\widetilde{U}$ denote the matrix obtained from U by deleting its first column, u_1. Since the smallest eigenvalue of $\bar{L}$ is zero, we may write $\bar{L} = \widetilde{U}\widetilde{\Lambda}\widetilde{U}^*$, where $\widetilde{\Lambda} = \mathrm{diag}\{\lambda_2(\bar{L}), \ldots \lambda_n(\bar{L})\}$, $\widetilde{U}^*\widetilde{U} = I_{n-1}$, and $\widetilde{U}\widetilde{U}^* = I_n - \vec{1}\vec{1}^*/n$. With this notation, our analysis is based on the decomposition

$$\theta^\tau \;=\; \alpha^\tau\,\frac{\vec{1}}{\sqrt{n}} + \widetilde{U}\,\beta^\tau, \tag{5.10}$$

[5] In this chapter, we denote the transpose of a matri or a vector by $(\cdot)^*$.

where we have defined $\alpha^\tau := \langle \vec{1}/\sqrt{n},\, \theta^\tau\rangle \in \mathbb{R}$ and $\beta^\tau := \widetilde{U}^* \theta^\tau \in \mathbb{R}^{n-1}$. Since $\vec{1}^* L^\tau = \vec{0}^*$ for all $\tau = 1, 2, \ldots$, from the decomposition (5.10) and the form of the updates (5.8), we have the following recursions,

$$\alpha^{\tau+1} \;=\; \alpha^\tau \;-\; \epsilon^\tau\, \frac{\vec{1}^*}{\sqrt{n}}\, v^\tau, \quad \text{and} \tag{5.11}$$

$$\beta^{\tau+1} \;=\; \beta^\tau \;-\; \epsilon^\tau \left[\underline{L}^\tau\, \beta^\tau \;+\; \widetilde{U}^* v^\tau\right]. \tag{5.12}$$

Here $\underline{L}$ is an $(n-1)\times(n-1)$ matrix defined by the relation

$$U^* L^\tau U = \begin{bmatrix} 0 & \vec{0}^* \\ \vec{0} & \underline{L}^\tau \end{bmatrix}_{n\times n}.$$

5.4.2 Main Steps

As we show, part (a) of the theorem requires some intermediate results of the proof of part (b); accordingly, we defer it to Appendix C.3. With this set-up, we now state the two main technical lemmas that form the core of Theorem 11. Our first lemma, proved in Appendix C.1, concerns the behavior of the component sequences $\{\alpha^\tau\}_{\tau=0}^\infty$ and $\{\beta^\tau\}_{\tau=0}^\infty$ which evolve according to equations (5.11) and (5.12) respectively.

Lemma 10. *Given the random sequence $\{\theta^\tau\}_{\tau=0}^\infty$ generated by the update equation (5.4), we have*

$$\text{MSE}(\theta^\tau) \;=\; \underbrace{\frac{1}{n}\,\text{var}\left(\alpha^\tau\right)}_{e_1^\tau} + \underbrace{\frac{1}{n}\,\mathbb{E}[\|\beta^\tau\|_2^2]}_{e_2^\tau}. \tag{5.13}$$

Furthermore, e_1^τ and e_2^τ satisfy the following bounds:

(a) For each iteration $\tau = 1, 2, \ldots$, we have

$$e_1^\tau \;\le\; \frac{2\,\sigma^2\,\delta}{[\lambda_2(\bar{S})]^2}. \tag{5.14}$$

(b) Moreover, for each iteration $\tau = 1, 2, \ldots$ we have

$$e_2^\tau \;\le\; \frac{\sigma^2}{[\lambda_2(\bar{S})]^2}\,\frac{\log(\tau + 1/\delta - 1)}{\tau + 1/\delta - 1} + e_2^0\,\frac{1/\delta - 1}{\tau + 1/\delta - 1}, \tag{5.15}$$

From Lemma 10, we conclude that in order to guarantee an $\mathcal{O}\big(\sigma^2\delta/[\lambda_2(\bar{S})]^2\big)$ bound on the MSE, it suffices to take τ such that

$$\frac{1/\delta - 1}{\tau + 1/\delta - 1} \;\leq\; \frac{\sigma^2\,\delta}{e_2^0\,[\lambda_2(\bar{S})]^2}, \quad \text{and} \quad \frac{\log(\tau + 1/\delta - 1)}{\tau + 1/\delta - 1} \;\leq\; \delta.$$

Note that the first inequality is satisfied when $\tau \geq e_2^0[\lambda_2(\bar{S})]^2/(\sigma^2\delta^2)$. Moreover, doing a little bit of algebra, one can see that $\tau = (2/\delta)\log(1/\delta) - (1/\delta - 1)$ is sufficient to satisfy the second inequality. Accordingly, we take

$$\tau \;=\; \max\left\{\frac{1}{\delta}\,\log\frac{1}{\delta}\,,\; \frac{e_2^0\,[\lambda_2(\bar{S})]^2}{\sigma^2\,\delta^2}\right\}$$

outer iterations.

The last part of the proof is to bound the second smallest eigenvalue of the Laplacian matrix $\bar{S}$. The following lemma, which we prove in Appendix C.2, addresses this issue. Recall that $\lambda_2(\cdot)$ denotes the second smallest eigenvalue of a matrix.

Lemma 11. *The averaged matrix $\bar{S}$ that arises from our protocol has the following properties:*

(a) For a cycle and a regular grid we have $\lambda_2(\bar{S}) = \Omega(1)$, and

(b) for a random geometric graph, we have $\lambda_2(\bar{S}) = \Omega(1/\log n)$, with high probability.

It is important to note that the averaged matrix $\bar{S}$ is *not the same* as the graph Laplacian that would arise from standard averaging on these graphs. Rather, as a consequence of establishing many paths and averaging along them in each inner phase, our protocol ensures that the matrix behaves essentially like the graph Laplacian for the fully connected graph.

As established previously, each outer step requires $M = \mathcal{O}(D_n)$ iterations. Therefore, we have shown that it is sufficient to take a total of

$$T \;=\; \mathcal{O}\Big(D_n\;\max\left\{\frac{1}{\delta}\log\frac{1}{\delta}\,,\;\frac{e_2^0\,[\lambda_2(\bar{S})]^2}{\sigma^2\,\delta^2}\right\}\Big)$$

transmissions per edge in order to guarantee a $\mathcal{O}(\sigma^2\delta/[\lambda_2(\bar{S})]^2)$ bound on the MSE. As we will see in the next section, assuming that the initial values are fixed, we have $e_1^0 = 0$, hence $\mathrm{MSE}(\theta^0) = e_2^0$. The claims in Theorem 11 then follow by standard calculations of the diameters of the various graphs and the result of the Lemma 11.

5.5 Simulation Results

In order to demonstrate the effectiveness of the proposed algorithm, we conducted a set of simulations. More specifically, we apply the proposed algorithm to four nearest-neighbor square grids of different sizes. We initially generate the data θ_i^0, $i = 1, 2, \ldots, n$ as random

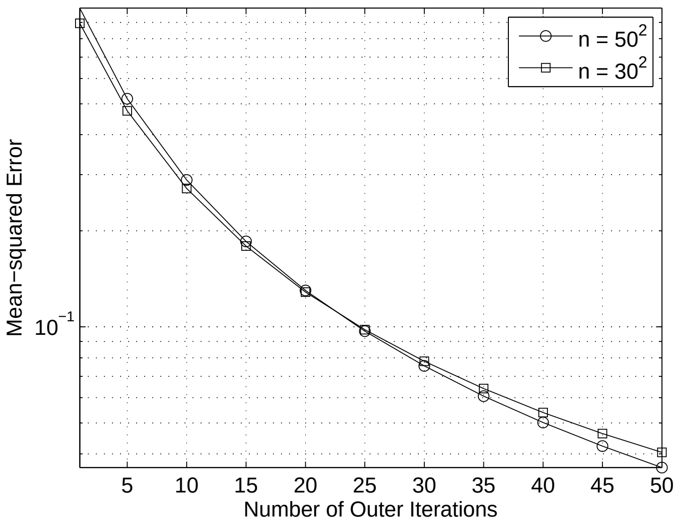


Figure 5.4: Mean-squared error versus the number of outer loop iterations for grids with $n \in \{30^2, 50^2\}$ nodes. As expected the MSE monotonically decreases, which supports the convergence claim.

$N(1, 1)$ variables and fix them throughout the simulation. So for each run of the algorithm the initial data is fixed. In implementing the algorithm, we adopt $\sigma^2 = 1$ as the channel noise variance, and we set the tolerance parameter $\delta = 0.1$, leading to the step size $\epsilon^\tau = 1/(10+\tau)$. We estimated the mean-squared error, defined in equation (5.1), by taking the average over 50 sample paths. As discussed in Section 5.3, every outer phase update requires $M = \mathcal{O}\big(\sqrt{n}\big)$ time steps.

Figure 5.4 shows the mean-squared error versus the number of outer loop iterations; the panel contains two different curves, one for a graph with $n = 30^2$ nodes, and the other for $n = 50^2$ nodes. As expected, the MSE monotonically decreases as the number of iterations increases, showing convergence of the algorithm. More importantly, the gap between the two plots is negligible. This phenomenon, which is predicted by our theory, is explored further in our next set of experiments.

In order to study the network scaling of the grid more precisely, for a given set of graph sizes, we compute the number of the *outer iterations* $\tau = \tau(n, \delta)$, such that $\mathrm{MSE}(\theta^{\tau M}) \leq \sigma^2 \delta$. Recall that this stopping time is the focus of Theorem 11(b). Figure 5.5 provides a box plot of this stopping time τ versus the graph size n. Theorem 11(b) predicts that this stopping time should be inversely proportional to the spectral gap of the Laplacian matrix $\bar{S}$, which for the grid scales as $\Omega(1)$ (in particular, see Lemma 11). As shown in Figure 5.5, over a range of graphs of size varying from $n = 1000$ to $n = 10000$, the stopping time is roughly constant ($\tau \approx 25$), which is consistent with the theory.

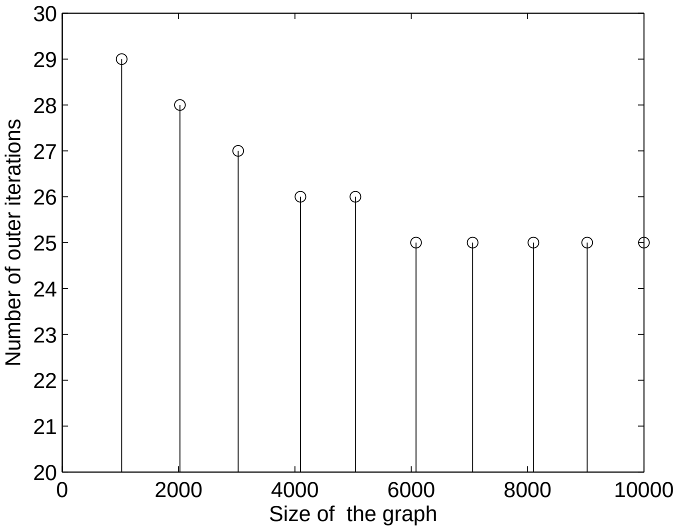


Figure 5.5: Stopping time $\tau = \tau(n, \delta)$ vs. the graph size n. For different graph sizes, we compute the first outer phase time instance $\tau(n, \delta)$, such that $\mathrm{MSE}(\theta^{\tau M}) \leq \sigma^2\delta$. Here we have fixed the parameters to $\sigma^2 = 1$, and $\delta = 0.1$. As you can see, over a range of graphs of size varying from 1000 to 10000, this stopping time is roughly constant (≈ 25), which is consistent with the theory (Theorem 11(b) and Lemma 11).

5.6 Conclusion

In this paper, we proposed and analyzed a two-phase graph-respecting algorithm for computing averages in a network, where communication is modeled as an additive white Gaussian noise channel. We showed that it achieves consensus, and we characterized the rate of convergence as a function of the graph topology and graph size. For our algorithm, this network scaling is within logarithmic factors of the graph diameter, showing that it is near-optimal, since the graph diameter provides a lower bound for any algorithm.

There are various issues left open in this work. First, while the AWGN model is more realistic than noiseless communication, many channels in wireless networks may be more complicated, for instance involving fading, interference and other types of memory. In principle, our algorithm could be applied to such channels and networks, but its behavior and associated convergence rates remain to be analyzed. In a separate direction, it is also worth noting that gossip-type algorithms can be used to solve other problems, such as distributed optimization problems (e.g., [80, 96, 33]) and kernel density estimation (e.g., [48]). Complexity reduction and studying the issue of optimal network scaling for such problems is also of interest.[6]

[6] The materials of this chapter have been published in papers [82, 83].

Appendix A

Proofs for Chapter 3

A.1 Details of Example 5

In this appendix, we verify the sufficient condition for contractivity (3.22). Recall the definition (3.12) of the zero'th order bounds. By construction, we have the relations

$$\begin{aligned}\underline{B}_{uv}(i) &= \underline{B}^0_{uv}(i) = \frac{\gamma}{1+(d-1)\gamma}, \quad \text{and} \\ \overline{B}_{uv}(i) &= \overline{B}^0_{uv}(i) = \frac{1}{1+(d-1)\gamma} \quad \text{for all } i \in \mathcal{X} \text{ and } (u \to v) \in \vec{\mathcal{E}}.\end{aligned}$$

Substituting these bounds into the definitions (3.20a) and (3.20b) and doing some simple algebra yields the upper bounds

$$\begin{aligned}\phi_{u\to v, w\to u} &\leq \max_{j\in\mathcal{X}} \left\{ \frac{\beta_{uv}(j) \prod_{s\in\mathcal{N}(u)\backslash\{v,w\}} \overline{B}_{us}(j)}{\sum_{\ell=1}^d \beta_{uv}(\ell) \prod_{s\in\mathcal{N}(u)\backslash v} \underline{B}_{us}(\ell)} \right\} \\ &= \frac{1+(d-1)\gamma}{\gamma^{\rho_u - 1}} \max_{j\in\mathcal{X}} \left\{ \frac{\psi_u(j)}{\sum_{\ell=1}^d \psi_u(\ell)} \right\},\end{aligned}$$

and

$$\begin{aligned}\chi_{u\to v, w\to u} &\leq \max_{j\in\mathcal{X}} \left\{ \frac{\beta_{uv}(j) \prod_{s\in\mathcal{N}(u)\backslash v} \overline{B}_{us}(j)}{\sum_{\ell=1}^d \beta_{uv}(\ell) \prod_{s\in\mathcal{N}(u)\backslash v} \underline{B}_{us}(\ell)} \right\} \max_{j\in\mathcal{X}} \left\{ \frac{1}{\underline{B}_{wu}(j)} \right\} \\ &= \frac{1+(d-1)\gamma}{\gamma^{\rho_u}} \max_{j\in\mathcal{X}} \left\{ \frac{\psi_u(j)}{\sum_{\ell=1}^d \psi_u(\ell)} \right\},\end{aligned}$$

where we have denoted the degree of the node u by ρ_u. Substituting these inequalities into expression (3.21) and noting that $\gamma \leq 1$, we find that the global update function has

Lipschitz constant at most

$$L \;\leq\; 4\,(1-\gamma)(1+(d-1)\gamma)\;\max_{u\in\mathcal{V}}\left\{\frac{(\rho_u-1)^2}{\gamma^{2\rho_u}}\;\max_{j\in\mathcal{X}}\left\{\frac{\psi_u(j)}{\sum_\ell \psi_u(\ell)}\right\}^2\right\},$$

as claimed.

A.2 Proof of Lemma 1

By construction, for each directed edge $(u \to v)$, the message vector $m_{u\to v}$ belongs to the probability simplex—that is, $\sum_{i\in\mathcal{X}} m_{u\to v}(i) = 1$, and $m_{u\to v} \succeq \vec{0}$. From equation (3.23), the vector $m_{u\to v}$ is a convex combination of the columns of the matrix Γ_{uv}. Recalling bounds (3.12), we conclude that the message vector must belong to the set $\mathcal{S}$, as defined in (3.17), in particular with $\underline{B}_{uv}(i) = \underline{B}^0_{uv}(i)$ and $\overline{B}_{uv}(i) = \overline{B}^0_{uv}(i)$. Note that the set $\mathcal{S}$ is compact, and any member of it has strictly positive elements under our assumptions.

For directed edges $(u \to v)$ and $(w \to s)$, let $\frac{\partial F_{u\to v}}{\partial m_{w\to s}} \in \mathbb{R}^{d\times d}$ denote the Jacobian matrix obtained from taking the partial derivative of the update function $F_{u\to v}$ with respect to the message vector $m_{w\to s}$. By inspection, the function $F_{u\to v}$ is continuously differentiable; consequently, the function $\frac{\partial F_{u\to v}(i;m)}{\partial m_{w\to s}(j)}$ is continuous, and hence must achieve its supremum over the compact set $\mathcal{S}$. We may use these Jacobian matrices to define a matrix $A_{u\to v,w\to s} \in \mathbb{R}^{d\times d}$ with entries

$$A_{u\to v,w\to s}(i,j) := \max_{m\in\mathcal{S}}\left|\frac{\partial F_{u\to v}(i;m)}{\partial m_{w\to s}(j)}\right|, \quad \text{for } i,j = 1,\ldots,d.$$

We then use these matrices to define a larger matrix $A \in \mathbb{R}^{D\times D}$, consisting of $2|\mathcal{E}|\times 2|\mathcal{E}|$ sub-blocks each of size $d\times d$, with the sub-blocks indexed by pairs of directed edges $(u \to v) \in \vec{\mathcal{E}}$. In particular, the matrix $A_{u\to v,w\to s}$ occupies the sub-block indexed by the edge pair $(u \to v)$ and $(w \to s)$. Note that by the structure of the update function F, the matrix $A_{u\to v,w\to s}$ can be non-zero only if $s = u$ and $w \in \mathcal{N}(u)\backslash\{v\}$.

Now let $\nabla F \in \mathbb{R}^{D\times D}$ denote the Jacobian matrix of the update function F. By the integral form of the mean value theorem, we have the representation

$$F(m) - F(m') \;=\; \left[\int_0^1 \nabla F(m' + \tau(m-m'))\,d\tau\right](m-m').$$

Applying triangle inequality separately to each component of this D-dimensional vector and then using the definition of A, we obtain the elementwise upper bound

$$|F(m) - F(m')| \;\preceq\; A\,|m-m'|.$$

It remains to show that A is nilpotent: more precisely, we show that A^r is the all-zero matrix, where $r = \operatorname{diam}(\mathcal{G})$ denotes the diameter of the graph $\mathcal{G}$. In order to do so, we first let $B \in \mathbb{R}^{2|\mathcal{E}| \times 2|\mathcal{E}|}$ be the "block indicator" matrix—that is, its entries are given by

$$B(u \to v, w \to s) \;=\; \begin{cases} 1 & \text{if } A_{u\to v, w\to s} \neq 0 \\ 0 & \text{otherwise.} \end{cases}$$

Based on this definition, it is straightforward to verify that if $B^r = 0$ for some positive integer r, then we also have $A^r = 0$. Consequently, it suffices to show that $B^r = 0$ for $r = \operatorname{diam}(\mathcal{G})$.

Fix a pair of directed edges $(u \to v)$ and $(w \to s)$, and some integer $\ell \geq 1$. We first claim that the matrix entry $B^\ell(u \to v, w \to s)$ is non-zero only if there exists a backtrackless *directed path* of length $\ell + 1$ from w to v that includes both s and u, meaning that there exist nodes $s_1, s_2, \ldots, s_{\ell-2}$ such that

$$w \in \mathcal{N}(s) \setminus s_1, \quad s_1 \in \mathcal{N}(s_2) \setminus s_3, \ldots, \quad \text{and} \quad s_{\ell-2} \in \mathcal{N}(u) \setminus v.$$

We prove this claim via induction. The base case $\ell = 1$ is true by construction. Now supposing that the claim holds at order ℓ, we show that it must hold at order $\ell + 1$. By definition of matrix multiplication, we have

$$B^{\ell+1}(u \to v, w \to s) \;=\; \sum_{(y \to x) \in \vec{\mathcal{E}}} B^\ell(u \to v, y \to x)\, B(y \to x, w \to s).$$

In order for this entry to be non-zero, there must exist a directed edge $(y \to x)$ that forms a $(\ell+1)$-directed path to $(u \to v)$, and moreover, we must have $s = y$, and $w \in \mathcal{N}(y) \setminus x$. These conditions are equivalent of having a backtrackless directed path of length $\ell + 2$ from w to v, with s and u as intermediate nodes, thereby completing the proof of our intermediate claim.

Finally, we observe that in a tree-structured graph, there can be no directed path of length greater than $r = \operatorname{diam}(\mathcal{G})$. Consequently, our intermediate claim implies that $B^r = 0$ for any tree-structured graph, which completes the proof.

A.3 Proof of Lemma 2

Noting that it is equivalent to bound the logarithm, we have

$$\log \prod_{\ell=i+1}^{t+2} \left(1 - \frac{\alpha}{\ell}\right) \;=\; \sum_{\ell=i+1}^{t+2} \log\left(1 - \frac{\alpha}{\ell}\right) \;\leq\; -\alpha \sum_{\ell=i+1}^{t+2} \frac{1}{\ell}, \tag{A.1}$$

where we used the fact that $\log(1-x) \leq -x$ for $x \in (0,1)$. Since the function $1/x$ is decreasing, we have

$$\sum_{\ell=i+1}^{t+2} \frac{1}{\ell} \;\geq\; \int_{i+1}^{t+3} \frac{1}{x}\, dx \;=\; \log(t+3) \;-\; \log(i+1). \tag{A.2}$$

Substituting inequality (A.2) into (A.1) yields

$$\log \prod_{\ell=i+1}^{t+2} \Big(1 - \frac{\alpha}{\ell}\Big) \;\leq\; \alpha \left(\log(i+1) \;-\; \log(t+3)\right),$$

from which the claim stated in the lemma follows.

A.4 Proof of Lemma 3

Let $\nabla q(m) \in \mathbb{R}^{D \times D}$ denote the Jacobian matrix of the function $q : \mathbb{R}^D \to \mathbb{R}^D$ evaluated at m. Since q is differentiable, we can apply the integral form of the mean value theorem to write

$$q(m) - q(m') \;=\; \Big[\int_0^1 \nabla q(m' + \tau(m - m'))\, d\tau \Big]\, (m - m').$$

From this representation, we obtain the upper bound

$$\begin{aligned} \|q(m) - q(m')\|_2 \;&\leq \Big[\int_0^1 |\!|\!|\nabla q(m' + \lambda(m - m'))|\!|\!|_2 \, d\lambda \Big] \|m - m'\|_2 \\ &\leq \sup_{m \in \mathcal{S}} |\!|\!|\nabla q(m)|\!|\!|_2 \, \|m - m'\|_2, \end{aligned}$$

showing that it suffices to control the quantity $\sup_{m \in \mathcal{S}} |\!|\!|\nabla q(m)|\!|\!|_2$.

Let $\frac{\partial q_{u \to v}(m)}{\partial m_{w \to s}}$ be the $d \times d$ matrix of partial derivatives of the function $q_{u \to v} : \mathbb{R}^D \to \mathbb{R}^d$ obtained from taking the partial derivatives with respect to the message vector $m_{w \to s} \in \mathbb{R}^d$. We then define a $2|\mathcal{E}| \times 2|\mathcal{E}|$-dimensional matrix A with the entries

$$A(u \to v, w \to s) \;:=\; \begin{cases} \sup_{m \in \mathcal{S}} |\!|\!| \frac{\partial q_{u \to v}(m)}{\partial m_{w \to s}} |\!|\!|_2 & \text{if } s = u, \text{ and } w \in \mathcal{N}(u) \backslash \{v\} \\ 0 & \text{otherwise.} \end{cases} \tag{A.3}$$

Our next step is to show that $\sup_{m \in \mathcal{S}} |\!|\!|\nabla q(m)|\!|\!|_2 \;\leq\; |\!|\!|A|\!|\!|_2$. Let $y = \{y_{u \to v}\}_{(u \to v) \in \vec{\mathcal{E}}}$ be an arbitrary D-dimensional vector, where each sub-vector $y_{u \to v}$ is an element of $\mathbb{R}^d$. By exploiting the structure of $\nabla q(m)$ and y, we have

$$\begin{aligned}
\|\nabla q(m)\, y\|_2^2 &= \sum_{(u\to v)\in\vec{\mathcal{E}}} \| \sum_{w\in\mathcal{N}(u)\backslash\{v\}} \frac{\partial q_{u\to v}(m)}{\partial m_{w\to u}}\, y_{w\to u}\|_2^2 \\
&\overset{\text{(i)}}{\leq} \sum_{(u\to v)\in\vec{\mathcal{E}}} \Big(\sum_{w\in\mathcal{N}(u)\backslash\{v\}} \|\frac{\partial q_{u\to v}(m)}{\partial m_{w\to u}}\, y_{w\to u}\|_2\Big)^2 \\
&\overset{\text{(ii)}}{\leq} \sum_{(u\to v)\in\vec{\mathcal{E}}} \Big(\sum_{w\in\mathcal{N}(u)\backslash\{v\}} |\!|\!|\frac{\partial q_{u\to v}(m)}{\partial m_{w\to u}}|\!|\!|_2 \|y_{w\to u}\|_2\Big)^2 \\
&\overset{\text{(iii)}}{\leq} \sum_{(u\to v)\in\vec{\mathcal{E}}} \Big(\sum_{w\in\mathcal{N}(u)\backslash\{v\}} A(u\to v, w\to u)\|y_{w\to u}\|_2\Big)^2,
\end{aligned}$$

where the bound (i) follows by triangle inequality; the bound (ii) follows from definition of the operator norm; and the final inequality (iii) follows by definition of A.

Defining the vector $z \in \mathbb{R}^{2|\mathcal{E}|}$ with the entries $z_{w\to u} = \|y_{w\to u}\|_2$, we have established the upper bound $\|\nabla q(m)\, y\|_2^2 \leq \|Az\|_2^2$, and hence that

$$\|\nabla q(m)\, y\|_2^2 \;\leq\; |\!|\!|A|\!|\!|_2^2\, \|z\|_2^2 \;=\; |\!|\!|A|\!|\!|_2^2\, \|y\|_2^2,$$

where the final equality uses the fact that $\|y\|_2^2 = \|z\|_2^2$ by construction. Since both the message m and vector y were arbitrary, we have shown that $\sup_{m\in\mathcal{S}} |\!|\!|\nabla q(m)|\!|\!|_2 \leq |\!|\!|A|\!|\!|_2$, as claimed.

Our final step is to control the quantities $\sup_{m\in\mathcal{S}} |\!|\!|\frac{\partial q_{u\to v}(m)}{\partial m_{w\to s}}|\!|\!|_2$ that define the entries of A. In this argument, we make repeated use of the elementary matrix inequality [47]

$$|\!|\!|B|\!|\!|_2^2 \leq \underbrace{\Big(\max_{i=1,\ldots,n}\sum_{j=1}^n |B_{ij}|\Big)}_{|\!|\!|B|\!|\!|_\infty} \underbrace{\Big(\max_{j=1,\ldots,n}\sum_{i=1}^n |B_{ij}|\Big)}_{|\!|\!|B|\!|\!|_1}, \tag{A.4}$$

valid for any $n \times n$ matrix.

Recall the definition of the probability distribution (3.8) that defines the function $q_{u\to v} : \mathbb{R}^D \to \mathbb{R}^d$, as well as our shorthand notation $M_{u\to v}(k) = \prod_{w\in\mathcal{N}(u)\backslash\{v\}} m_{w\to u}(k)$. Taking the derivatives and performing some algebra yields

$$\begin{aligned}
\frac{\partial q_{u\to v}(i\,;\,m)}{\partial m_{w\to u}(j)} &= \sum_{k=1}^d \frac{\partial q_{u\to v}(i\,;\,m)}{\partial M_{u\to v}(k)}\,\frac{\partial M_{u\to v}(k)}{\partial m_{w\to u}(j)} \\
&= \frac{\partial q_{u\to v}(i\,;\,m)}{\partial M_{u\to v}(j)}\,\frac{M_{u\to v}(j)}{m_{w\to u}(j)} \\
&= \frac{-\beta_{uv}(i)\, M_{u\to v}(i)\, \beta_{uv}(j)}{\big(\sum_{k=1}^d \beta_{uv}(k) M_{u\to v}(k)\big)^2}\,\frac{M_{u\to v}(j)}{m_{w\to u}(j)},
\end{aligned}$$

for $i \neq j$, and $w \in \mathcal{N}(u)\backslash\{v\}$. For $i = j$, we have

$$\begin{aligned}\frac{\partial q_{u\to v}(i\,;\,m)}{\partial m_{w\to u}(i)} &= \frac{\partial q_{u\to v}(i\,;\,m)}{\partial M_{u\to v}(i)}\,\frac{M_{u\to v}(i)}{m_{w\to u}(i)}\\ &= \left[\frac{\beta_{uv}(i)}{\sum_{k=1}^d \beta_{uv}(k) M_{u\to v}(k)} - \frac{\beta_{uv}(i)^2\, M_{u\to v}(i)}{\left(\sum_{k=1}^d \beta_{uv}(k) M_{u\to v}(k)\right)^2}\right]\frac{M_{u\to v}(i)}{m_{w\to u}(i)}.\end{aligned}$$

Putting together the pieces leads to the upper bounds

$$\begin{aligned}|\!|\!|\frac{\partial q_{u\to v}(m)}{\partial m_{w\to u}}|\!|\!|_1 &\leq 2\max_{j\in\mathcal{X}}\left\{\frac{\beta_{uv}(j)\, M_{u\to v}(j)}{\sum_{k=1}^d \beta_{uv}(k)\, M_{u\to v}(k)}\,\frac{1}{m_{w\to u}(j)}\right\}, \quad \text{and}\\ |\!|\!|\frac{\partial q_{u\to v}(m)}{\partial m_{w\to u}}|\!|\!|_\infty &\leq \max_{i\in\mathcal{X}}\left\{\frac{\beta_{uv}(i)\, M_{u\to v}(i)}{\sum_{k=1}^d \beta_{uv}(k)\, M_{u\to v}(k)}\,\frac{1}{m_{w\to u}(i)}\right.\\ &\qquad\left. + \frac{\beta_{uv}(i)\, M_{u\to v}(i)}{\left(\sum_{k=1}^d \beta_{uv}(k)\, M_{u\to v}(k)\right)^2}\sum_{j=1}^d \frac{\beta_{uv}(j)\, M_{u\to v}(j)}{m_{w\to u(j)}}\right\}.\end{aligned}$$

Recalling the definitions (3.20a) and (3.20b) of $\phi_{u\to v,w\to u}$ and $\chi_{u\to v,w\to u}$ respectively, we find that

$$|\!|\!|\frac{\partial q_{u\to v}(m)}{\partial m_{w\to u}}|\!|\!|_1 \leq 2\,\phi_{u\to v,w\to u}, \quad \text{and} \quad |\!|\!|\frac{\partial q_{u\to v}(m)}{\partial m_{w\to u}}|\!|\!|_\infty \leq \phi_{u\to v,w\to u} + \chi_{u\to v,w\to u}.$$

Thus, by applying inequality (A.4) with $B = \frac{\partial q_{u\to v}(m)}{\partial m_{w\to u}}$, we conclude that

$$|\!|\!|\frac{\partial q_{u\to v}(m)}{\partial m_{w\to u}}|\!|\!|_2^2 \leq 2\,\phi_{u\to v,w\to u}\,(\phi_{u\to v,w\to u} + \chi_{u\to v,w\to u}).$$

Since this bound holds for any message $m \in \mathcal{S}$, we conclude that each of the matrix entries $A(u \to v, w \to u)$ satisfies the same inequality. Again applying the basic matrix inequality (A.4), this time with $B = A$, we conclude that $|\!|\!|A|\!|\!|_2$ is upper bounded by

$$\begin{aligned}&2\left(\max_{(u\to v)\in\vec{\mathcal{E}}}\sum_{w\in\mathcal{N}(u)\backslash\{v\}}\left(\phi_{u\to v,w\to u}\,(\phi_{u\to v,w\to u} + \chi_{u\to v,w\to u})\right)^{\frac{1}{2}}\right)\\ &\left(\max_{(w\to u)\in\vec{\mathcal{E}}}\sum_{v\in\mathcal{N}(u)\backslash w}\left(\phi_{u\to v,w\to u}\,(\phi_{u\to v,w\to u} + \chi_{u\to v,w\to u})\right)^{\frac{1}{2}}\right),\end{aligned}$$

which concludes the proof.

Appendix B

Proofs for Chapter 4

B.1 Proof of Lemma 5

Subtracting $a^*_{u\to v;j}$ from both sides of the update (4.15) in Step 2(c), we obtain

$$a^{t+1}_{u\to v;j} - a^*_{u\to v;j} \;=\; (1-\eta^t)\left[a^t_{u\to v;j} - a^*_{u\to v;j}\right] \;+\; \eta^t\left[b^t_{u\to v;j} - a^*_{u\to v;j}\right] \;+\; \eta^t\,\zeta^{t+1}_{u\to v;j}. \tag{B.1}$$

Setting $\eta^t = 1/(t+1)$ and unwrapping the recursion (B.1) then yields

$$a^{t+1}_{u\to v;j} - a^*_{u\to v;j} \;=\; \frac{1}{t+1}\sum_{\tau=0}^{t}\left[b^\tau_{u\to v;j} - a^*_{u\to v;j}\right] \;+\; \frac{1}{t+1}\sum_{\tau=0}^{t}\zeta^{\tau+1}_{u\to v;j}.$$

Squaring both sides of this equality and using the upper bound $(a+b)^2 \le 2a^2 + 2b^2$, we obtain

$$\left(a^{t+1}_{u\to v;j} - a^*_{u\to v;j}\right)^2 \;\le\; \frac{2}{(t+1)^2}\left\{\sum_{\tau=0}^{t}\left[b^\tau_{u\to v;j} - a^*_{u\to v;j}\right]\right\}^2 \;+\; \frac{2}{(t+1)^2}\left\{\sum_{\tau=0}^{t}\zeta^{\tau+1}_{u\to v;j}\right\}^2.$$

Summing over indices $j = 1, 2, \ldots, r$ and recalling the expansion (4.36), we find that

$$\begin{aligned}
\|\Delta^t_{u\to v}\|^2_{L^2} \;&\le\; \sum_{j=1}^{r}\left\{\frac{2}{(t+1)^2}\left\{\sum_{\tau=0}^{t}\left[b^\tau_{u\to v;j} - a^*_{u\to v;j}\right]\right\}^2 \;+\; \frac{2}{(t+1)^2}\left\{\sum_{\tau=0}^{t}\zeta^{\tau+1}_{u\to v;j}\right\}^2\right\} \\
&\overset{\text{(i)}}{\le}\; \underbrace{\frac{2}{(t+1)}\sum_{j=1}^{r}\sum_{\tau=0}^{t}\left[b^\tau_{u\to v;j} - a^*_{u\to v;j}\right]^2}_{\text{Deterministic term } D^{t+1}_{u\to v}} \;+\; \underbrace{\frac{2}{(t+1)^2}\sum_{j=1}^{r}\left\{\sum_{\tau=0}^{t}\zeta^{\tau+1}_{u\to v;j}\right\}^2.}_{\text{Stochastic term } S^{t+1}_{u\to v}}
\end{aligned}$$

Here step (i) follows from the elementary inequality

$$\left\{\sum_{\tau=0}^{t}\left[b^\tau_{u\to v;j} - a^*_{u\to v;j}\right]\right\}^2 \;\le\; (t+1)\sum_{\tau=0}^{t}\left[b^\tau_{u\to v;j} - a^*_{u\to v;j}\right]^2.$$

B.2 Proof of Lemma 6

Recall the probability density

$$[p_{u\to v}(m)](\cdot) \;\propto\; \beta_{uv}(\cdot) \prod_{w\in\mathcal{N}(u)\backslash\{v\}} m_{w\to u}(\cdot)$$

defined in Step 2 of the SOSMP algorithm. Using this shorthand notation, the claim of Lemma 4 can be re-written as $[\mathcal{F}_{u\to v}(m)](x) = \langle \Gamma_{uv}(x,\cdot),\, [p_{u\to v}(m)](\cdot)\rangle$. Therefore, applying the Cauchy-Schwartz inequality yields

$$|[\mathcal{F}_{u\to v}(m)](x) - [\mathcal{F}_{u\to v}(m')](x)|^2 \;\leq\; \|\Gamma_{vu}(x,\cdot)\|_{L^2}^2 \;\|p_{u\to v}(m) \;-\; p_{u\to v}(m')\|_{L^2}^2.$$

Integrating both sides of the previous inequality over $\mathcal{X}$ and taking square roots yields

$$\|\mathcal{F}_{u\to v}(m) \;-\; \mathcal{F}_{u\to v}(m')\|_{L^2} \;\leq\; C_{uv}\,\|p_{u\to v}(m) \;-\; p_{u\to v}(m')\|_{L^2},$$

where we have denoted the constant $C_{uv} := \big(\int_{\mathcal{X}} |\Gamma_{uv}(x,y)|^2 dydx\big)^{1/2}$.

Next step would be to upper bound the term $\|p_{u\to v}(m) - p_{u\to v}(m')\|_{L^2}$. In order to do so, we first show that $p_{u\to v}(m)$ is a Frechet differentiable operator [36, 68, 26] on the space $\mathcal{M}' := \text{convhull}\{m^*, \oplus_{(u\to v)\in\vec{\mathcal{E}}}\mathcal{M}'_{u\to v}\}$, where

$$\mathcal{M}'_{u\to v} \;:=\; \Big\{\widehat{m}_{u\to v} \;\Big|\; \widehat{m}_{u\to v} = \Big[\mathbb{E}_{Y\sim f}\big[\Pi^r\big(\Gamma_{uv}(\cdot,Y)\big)\big]\Big]_+,\ \text{for some probability density } f\Big\},$$

denotes the space of all feasible SOSMP messages on the directed edge $(u\to v)$. Doing some calculus using the chain rule, we calculate the partial directional (Gateaux) derivative [26, 93] of the operator $p_{u\to v}(m)$ with respect to the function $m_{w\to u}$. More specifically, for an arbitrary function $h_{w\to u}$, we have

$$\begin{aligned}[][\mathcal{D}_{w\to u}\, p_{u\to v}(m)](h_{w\to u}) \;&=\; \frac{\beta_{uv}\prod_{s\in\mathcal{N}(u)\backslash\{v,w\}} m_{s\to u}}{\langle M_{uv},\, \beta_{uv}\rangle}\; h_{w\to u}\\ &\quad -\; \frac{\beta_{uv}M_{uv}}{\langle M_{uv},\,\beta_{uv}\rangle^2}\;\langle h_{w\to u},\, \beta_{uv}\prod_{s\in\mathcal{N}(u)\backslash\{v,w\}} m_{s\to u}\rangle,\end{aligned}$$

where $M_{uv} = \prod_{w\in\mathcal{N}(u)\backslash\{v\}} m_{w\to u}$. Clearly the Gateaux derivative is linear and continuous. It is also bounded as will be shown now. Massaging the operator norm's definition, we obtain

$$\begin{aligned}\sup_{m\in\mathcal{M}'}\|\mathcal{D}_{w\to u}\, p_{u\to v}(m)\|_2 \;&=\; \sup_{m\in\mathcal{M}'}\ \sup_{h_{w\to u}\in\mathcal{M}'_{w\to u}} \frac{\|[\mathcal{D}_{w\to u}\, p_{u\to v}(m)](h_{w\to u})\|_{L^2}}{\|h_{w\to u}\|_{L^2}}\\ &\leq\; \sup_{m\in\mathcal{M}'} \frac{\sup_{x\in\mathcal{X}}\ \beta_{uv}(x)\prod_{s\in\mathcal{N}(u)\backslash\{v,w\}} m_{s\to u}(x)}{\langle M_{uv},\,\beta_{uv}\rangle}\\ &\quad +\; \sup_{m\in\mathcal{M}'} \frac{\|\beta_{uv}M_{uv}\|_{L^2}\,\|\beta_{uv}\prod_{s\in\mathcal{N}(u)\backslash\{v,w\}} m_{s\to u}\|_{L^2}}{\langle M_{uv},\,\beta_{uv}\rangle^2}. \qquad (B.2)\end{aligned}$$

Since the space $\mathcal{X}$ is compact, the continuous functions β_{uv} and $m_{s\to u}$ achieve their maximum over $\mathcal{X}$. Therefore, the numerator of (B.2) is bounded and we only need to show that the denominator is bounded away from zero.

For an arbitrary message $m_{u\to v} \in \mathcal{M}'_{u\to v}$ there exist $0 < \alpha < 1$ and a bounded probability density f so that

$$m_{u\to v}(x) \;=\; \alpha\, m^*_{u\to v}(x) \;+\; (1-\alpha)\Big[\mathbb{E}_{Y\sim f}\big[\widetilde{\Gamma}_{uv}(x,Y)\big]\Big]_+,$$

where we have introduced the shorthand $\widetilde{\Gamma}_{uv}(\cdot,y) := \Pi^r(\Gamma_{uv}(\cdot,y))$. According to Lemma 4, we know $m^*_{u\to v} = \mathbb{E}_Y[\Gamma_{uv}(\cdot,Y)]$, where $Y \sim p_{u\to v}(m^*)$. Therefore, denoting $p^* = p_{u\to v}(m^*)$, we have

$$\begin{aligned} m_{u\to v}(x) \;&\geq\; \alpha\,\mathbb{E}_{Y\sim p^*}[\Gamma_{uv}(x,Y)] \;+\; (1-\alpha)\,\mathbb{E}_{Y\sim f}[\widetilde{\Gamma}_{uv}(x,Y)] \\ &=\; \mathbb{E}_{Y\sim(\alpha p^*+(1-\alpha)f)}[\widetilde{\Gamma}_{uv}(x,Y)] \;+\; \alpha\,\mathbb{E}_{Y\sim p^*}[\Gamma_{uv}(x,Y) - \widetilde{\Gamma}_{uv}(x,Y)]. \end{aligned} \tag{B.3}$$

On the other hand, since $\mathcal{X}$ is compact, we can exchange the order of expectation and projection using Fubini's theorem to obtain

$$\mathbb{E}_{Y\sim p^*}[\Gamma_{uv}(\cdot,Y) - \widetilde{\Gamma}_{uv}(\cdot,Y)] \;=\; m^*_{u\to v} - \Pi^r(m^*_{u\to v}) \;=\; A^r_{u\to v}.$$

Substituting the last equality into the bound (B.3) yields

$$m_{u\to v}(x) \;\geq\; \inf_{y\in\mathcal{X}} \widetilde{\Gamma}_{uv}(x,y) \;-\; |A^r_{u\to v}(x)|.$$

Recalling the assumption (4.22), one can conclude that the right hand side of the above inequality is positive for all directed edges $(u \to v)$. Therefore, the denominator of the expression (B.2) is bounded away from zero and more importantly $\sup_{m\in\mathcal{M}} |\!|\!|\mathcal{D}_{w\to u} p_{u\to v}(m)|\!|\!|_2$ is attainable.

Since the derivative is a bounded, linear, and continuous operator, the Gateaux and Frechet derivatives coincides [26, 93] and we can use the mean value theorem (Luenberger [75], page 176) to obtain the following upper bound

$$\|p_{u\to v}(m) \,-\, p_{u\to v}(m')\|_{L^2} \;\leq\; \sum_{w\in\mathcal{N}(u)\backslash\{v\}} \sup_{0\leq\alpha\leq 1} |\!|\!|\mathcal{D}_{w\to u}\, p_{u\to v}(m' + \alpha\,(m - m'))|\!|\!|_2\, \|m_{w\to u} \,-\, m'_{w\to u}\|_{L^2}.$$

Setting $L_{u\to v, w\to u} \;:=\; C_{uv}\, \sup_{m\in\mathcal{M}'} |\!|\!|\mathcal{D}_{w\to u}\, p_{u\to v}(m)|\!|\!|_2$ and putting the pieces together yields

$$\|\mathcal{F}_{u\to v}(m) \,-\, \mathcal{F}_{u\to v}(m')\|_{L^2} \;\leq\; \sum_{w\in\mathcal{N}(u)\backslash\{v\}} L_{u\to v,w\to u}\, \|m_{w\to u} \,-\, m'_{w\to u}\|_{L^2},$$

for all $m, m' \in \mathcal{M}'$.

The last step of the proof is to verify that $m^* \in \mathcal{M}'$, and $\widehat{m}^t \in \mathcal{M}'$ for all $t = 1, 2, \ldots$. By definition we have $m^* \in \mathcal{M}'$. On the other hand, unwrapping the update (4.15) we obtain

$$\begin{aligned} a^t_{u\to v;j} &= \frac{1}{t} \sum_{\tau=0}^{t-1} \widetilde{b}^{\tau+1}_{u\to v;j} \\ &= \frac{1}{t} \sum_{\tau=0}^{t-1} \frac{1}{k} \sum_{\ell=1}^{k} \int_{\mathcal{X}} \Gamma_{uv}(x, Y_\ell)\, \phi_j(x)\, dx \\ &= \int_{\mathcal{X}} \mathbb{E}_{Y\sim\hat{p}}[\Gamma_{uv}(x, Y)]\, \phi_j(x)\, dx, \end{aligned}$$

where $\hat{p}$ denotes the empirical probability density. Therefore, $m^t_{u\to v} = \sum_{j=1}^r a^t_{u\to v;j}\, \phi_j$ is equal to $\Pi^r(\mathbb{E}_{Y\sim\hat{p}}[\Gamma_{uv}(\cdot, Y)])$, thereby completing the proof.

B.3 Proof of Lemma 7

We begin by taking the conditional expectation of $\widetilde{b}^{t+1}_{u\to v;j}$, previously defined (4.14), given the filteration $\mathcal{G}^t$ and with respect to the random samples $\{Y_1, \ldots, Y_k\} \overset{\text{i.i.d.}}{\sim} [p_{u\to v}(\widehat{m})](\cdot)$. Exchanging the order of expectation and integral[1] and exploiting the result of Lemma 4, we obtain

$$\mathbb{E}[\widetilde{b}^{t+1}_{u\to v;j} \mid \mathcal{G}^t] \;=\; \int_{\mathcal{X}} [\mathcal{F}_{u\to v}(\widehat{m}^t)](x)\, \phi_j(x)\, dx \;=\; b^t_{u\to v;j}, \tag{B.4}$$

and hence $\mathbb{E}[\zeta^{t+1}_{u\to v;j} \mid \mathcal{G}^t] = 0$, for all $j = 1, 2, \ldots, r$ and all directed edges $(u \to v) \in \vec{\mathcal{E}}$. Also it is clear that $\zeta^{t+1}_{u\to v;j}$ is $\mathcal{G}^t$-measurable. Therefore, $\{\zeta^{\tau+1}_{u\to v;j}\}_{\tau=0}^{\infty}$ forms a martingale difference sequence with respect to the filtration $\{\mathcal{G}^\tau\}_{\tau=0}^{\infty}$. On the other hand, recalling the bound (4.23), we have

$$|\widetilde{b}^{t+1}_{u\to v;j}| \;\le\; \frac{1}{k} \sum_{\ell=1}^{k} |\langle \Gamma_{uv}(\cdot, Y_\ell),\, \phi_j \rangle| \;\le\; B_j. \tag{B.5}$$

Moreover, exploiting the result of Lemma 4 and exchanging the order of the integration and expectation once more yields

$$|b^t_{u\to v;j}| \;=\; |\langle \mathbb{E}_Y[\Gamma_{uv}(\cdot, Y)],\, \phi_j \rangle| \;=\; |\mathbb{E}_Y[\langle \Gamma_{uv}(\cdot, Y),\, \phi_j \rangle]| \;\le\; B_j, \tag{B.6}$$

where we have $Y \sim [p_{u\to v}(\widehat{m}^t)](y)$. Therefore, the martingale difference sequence is bounded, in particular with

$$|\zeta^{t+1}_{u\to v;j}| \;\le |\widetilde{b}^{t+1}_{u\to v;j}| + |\, b^t_{u\to v;j}| \;\le\; 2\, B_j.$$

[1] Since $\Gamma_{uv}(x, y)\phi_i(x)[p_{u\to v}(\widehat{m}^t)](y)$ is absolutely integrable, we can exchange the order of the integrals using Fubini's theorem.

B.4 Proof of Lemma 8

We start by uniformly upper-bounding the terms $\mathbb{E}[|T^{t+1}_{u\to v}|]$. To do so we first need to bound $\|\Delta^t_{u\to v}\|_{L^2}$. By definition we know $\|\Delta^t_{u\to v}\|^2_{L^2} = \sum_{j=1}^r [a^t_{u\to v;j} - a^*_{u\to v;j}]^2$; therefore we only need to control the terms $a^t_{u\to v;j}$ and $a^*_{u\to v;j}$ for $j = 1, 2, \ldots, r$.

By construction, we always have $|\widetilde{b}^{t+1}_{u\to v;j}| \le B_j$ for all iterations $t = 0, 1, \ldots$. Also, assuming that $|a^0_{u\to v;j}| \le B_j$, without loss of generality, a simple induction using the update equation (4.15) shows that $|a^t_{u\to v;j}| \le B_j$ for all t. Moreover, using a similar argument leading to (B.6), we obtain

$$|a^*_{u\to v;j}| \;=\; |\langle \mathbb{E}_Y[\Gamma_{uv}(\cdot, Y)],\, \phi_j\rangle| \;=\; |\mathbb{E}_Y[\langle \Gamma_{uv}(\cdot, Y),\, \phi_j\rangle]| \;\le\; B_j,$$

where we have $Y \sim [p_{u\to v}(m^*)](y)$. Therefore, putting the pieces together, recalling the definition (4.45) of $T^{t+1}_{u\to v}$ yields

$$\mathbb{E}[|T^{t+1}_{u\to v}|] \;\le\; \frac{4}{t+1} \sum_{w\in\mathcal{N}(u)\backslash\{v\}} \widetilde{L}_{u\to v, w\to u} \sum_{j=1}^r B_j^2 + \frac{32}{t+1} \sum_{j=1}^r B_j^2.$$

Concatenating the previous scalar inequalities yields $\mathbb{E}[T^{t+1}_0] \preceq \vec{v}/(t+1)$, for all $t \ge 0$, where we have defined the r-vector $\vec{v} := \big\{\sum_{j=1}^r B_j^2\big\}(4N\vec{1} + 32)$.

We now show, using an inductive argument, that

$$\mathbb{E}[T^{t+1}_s] \;\preceq\; \frac{\vec{v}}{t+1} \sum_{u=0}^s \frac{(\log(t+1))^u}{u!}, \tag{B.7}$$

for all $s = 0, 1, 2, \ldots$ and $t = 0, 1, 2, \ldots$. We have already established the base case $s = 0$. For some $s > 0$, assume that the claim holds for $s-1$. By the definition of T^{t+1}_s, we have

$$\begin{aligned}\mathbb{E}[T^{t+1}_s] &= \frac{1}{t+1} \sum_{\tau=1}^t \mathbb{E}[T^\tau_{s-1}] \\ &\preceq \frac{\vec{v}}{t+1} \sum_{\tau=1}^t \Big\{\frac{1}{\tau} + \sum_{u=1}^{s-1} \frac{(\log\tau)^u}{u!\,\tau}\Big\},\end{aligned}$$

where the inequality follows from the induction hypothesis. We now make note of the elementary inequalities $\sum_{\tau=1}^t 1/\tau \le 1 + \log t$, and

$$\sum_{\tau=1}^t \frac{(\log\tau)^u}{u!\,\tau} \;\le\; \int_1^t \frac{(\log x)^u}{u!\,x}\,dx \;=\; \frac{(\log t)^{(u+1)}}{(u+1)!}, \qquad \text{for all } u \ge 1$$

from which the claim follows.

B.5 Proof of Lemma 9

Upper-bounding the term $U^t_{u\to v}$: By construction, we always have $|\widetilde{b}^{t+1}_{u\to v;j}| \le B_j$ for all iterations $t = 0, 1, 2, \ldots$. Moreover, assuming $|a^0_{u\to v;j}| \le B_j$, without loss of generality, a simple induction on the update equation shows that $|a^t_{u\to v;j}| \le B_j$ for all iterations $t = 0, 1, \ldots$. On this basis, we find that

$$U^t_{u\to v} \;=\; (\eta^t)^2 \sum_{j=1}^r \mathbb{E}\left[\left(\widetilde{b}^{t+1}_{u\to v;j} - a^t_{u\to v;j}\right)^2\right] \;\le\; 4\,(\eta^t)^2 \sum_{j=1}^r B_j^2,$$

which establishes the bound (4.54a).

Upper-bounding the term $V^t_{u\to v}$: It remains to establish the bound (4.54b) on $V^t_{u\to v}$. We first condition on the σ-field $\mathcal{G}^t = \sigma(m^0, \ldots, m^t)$ and take expectations over the remaining randomness, thereby obtaining

$$\begin{aligned} V^t_{u\to v} \;&=\; 2\eta^t\,\mathbb{E}\Big[\mathbb{E}\big[\sum_{j=1}^r \big(\widetilde{b}^{t+1}_{u\to v;j} - a^t_{u\to v;j}\big)\,\big(a^t_{u\to v;j} - a^*_{u\to v;j}\big)\,\big|\,\mathcal{G}^t\big]\Big] \\ &=\; 2\eta^t\,\mathbb{E}\Big[\sum_{j=1}^r \big(b^t_{u\to v;j} - a^t_{u\to v;j}\big)\,\big(a^t_{u\to v;j} - a^*_{u\to v;j}\big)\Big], \end{aligned}$$

where $\{b^t_{u\to v;j}\}_{j=1}^\infty$ are the expansion coefficients of the function $\mathcal{F}_{u\to v}(\widehat{m}^t)$ (i.e. $b^t_{u\to v;j} = \langle \mathcal{F}_{u\to v}(\widehat{m}^t),\, \phi_j\rangle$), and we have recalled the result $\mathbb{E}[\widetilde{b}^{t+1}_{u\to v;j}|\mathcal{G}^t] = b^t_{u\to v;j}$ from (B.4). By Parseval's identity, we have

$$\begin{aligned} T \;&:=\; \sum_{j=1}^r \big(b^t_{u\to v;j} - a^t_{u\to v;j}\big)\,\big(a^t_{u\to v;j} - a^*_{u\to v;j}\big) \\ &=\; \langle \Pi^r(\mathcal{F}_{u\to v}(\widehat{m}^t)) - m^t_{u\to v},\, m^t_{u\to v} - \Pi^r(m^*_{u\to v})\rangle. \end{aligned}$$

Here we have used the basis expansions

$$m^t_{u\to v} = \sum_{j=1}^r a^t_{u\to v;j}\phi_j, \quad \text{and} \quad \Pi^r(m^*_{u\to v}) = \sum_{j=1}^r a^*_{u\to v;j}\phi_j.$$

Since $\Pi^r(m^t_{u\to v}) = m^t_{u\to v}$ and $\mathcal{F}_{u\to v}(m^*) = m^*_{u\to v}$, we have

$$\begin{aligned} T \;&=\; \langle \Pi^r\big(\mathcal{F}_{u\to v}(\widehat{m}^t) - \mathcal{F}_{u\to v}(m^*)\big),\, m^t_{u\to v} - \Pi^r(m^*_{u\to v})\rangle \;-\; \|m^t_{u\to v} - \Pi^r(m^*_{u\to v})\|^2_{L^2} \\ &\overset{(i)}{\le}\; \|\Pi^r\big(\mathcal{F}_{u\to v}(\widehat{m}^t) - \mathcal{F}_{u\to v}(m^*)\big)\|_{L^2}\;\|m^t_{u\to v} - \Pi^r(m^*_{u\to v})\|_{L^2} \;-\; \|m^t_{u\to v} - \Pi^r(m^*_{u\to v})\|^2_{L^2} \\ &\overset{(ii)}{\le}\; \|\mathcal{F}_{u\to v}(\widehat{m}^t) - \mathcal{F}_{u\to v}(m^*)\|_{L^2}\,\|m^t_{u\to v} - \Pi^r(m^*_{u\to v})\|_{L^2} \;-\; \|m^t_{u\to v} - \Pi^r(m^*_{u\to v})\|^2_{L^2}. \end{aligned}$$

where step (i) uses the Cauchy-Schwarz inequality, and step (ii) uses the non-expansivity of projection. Applying the contraction condition (4.26), we obtain

$$\begin{aligned}
T \;&\leq \big(1-\frac{\gamma}{2}\big)\sqrt{\frac{\sum\limits_{w\in\mathcal{N}(u)\backslash\{v\}}\|\widehat{m}^t_{w\to u}-m^*_{w\to u}\|^2_{L^2}}{|\mathcal{N}(u)|-1}}\;\|m^t_{u\to v}-\Pi^r(m^*_{u\to v})\|_{L^2} \\
&\quad-\;\|m^t_{u\to v}-\Pi^r(m^*_{u\to v})\|^2_{L^2} \\
&\leq \big(1-\frac{\gamma}{2}\big)\Big\{\frac{1}{2}\,\frac{\sum_{w\in\mathcal{N}(u)\backslash\{v\}}\|m^t_{w\to u}-m^*_{w\to u}\|^2_{L^2}}{|\mathcal{N}(u)|-1}\;+\;\frac{1}{2}\,\|m^t_{u\to v}-\Pi^r(m^*_{u\to v})\|^2_{L^2}\Big\} \\
&\quad-\;\|m^t_{u\to v}-\Pi^r(m^*_{u\to v})\|^2_{L^2},
\end{aligned}$$

where the second step follows from the elementary inequality $ab \leq a^2/2 + b^2/2$ and the non-expansivity of projection onto the space of non-negative functions. By the Pythagorean theorem, we have

$$\begin{aligned}
\|m^t_{w\to u}-m^*_{w\to u}\|^2_{L^2} \;&=\; \|m^t_{w\to u}-\Pi^r(m^*_{w\to u})\|^2_{L^2} \;+\; \|\Pi^r(m^*_{w\to u})-m^*_{w\to u}\|^2_{L^2} \\
&=\; \|\Delta^t_{w\to u}\|^2_{L^2}+\|A^r_{w\to u}\|^2_{L^2}.
\end{aligned}$$

Using this equality and taking expectations, we obtain

$$\begin{aligned}
\mathbb{E}[T] \;&\leq\; \big(1-\frac{\gamma}{2}\big)\Big\{\frac{1}{2}\,\frac{\sum_{w\in\mathcal{N}(u)\backslash\{v\}}[\bar{\rho}^2(\Delta^t_{w\to u})+\|A^r_{w\to u}\|^2_{L^2}]}{|\mathcal{N}(u)|-1}+\frac{1}{2}\,\bar{\rho}^2(\Delta^t_{u\to v})\Big\}\;-\;\bar{\rho}^2(\Delta^t_{u\to v}) \\
&\leq\; \big(\frac{1}{2}-\frac{\gamma}{4}\big)\,\rho^2_{\max}(A^r)\;+\;\big(\frac{1}{2}-\frac{\gamma}{4}\big)\,\bar{\rho}^2_{\max}(\Delta^t)\;-\;\big(\frac{1}{2}+\frac{\gamma}{4}\big)\,\bar{\rho}^2(\Delta^t_{u\to v}).
\end{aligned}$$

Since $V^t_{u\to v} = 2\eta^t\,\mathbb{E}[T]$, the claim follows.

Appendix C

Proofs for Chapter 5

C.1 Proof of Lemma 10

We begin by observing that

$$\mathbb{E}\big[(\theta^\tau - \bar{\theta}\vec{1})\,(\theta^\tau - \bar{\theta}\vec{1})^*\big] \;=\; F_1 \,+\, F_2 \,+\, F_3,$$

where $F_1 := \mathbb{E}\big[(\alpha^\tau - \sqrt{n}\bar{\theta})^2\big]\,\vec{1}\vec{1}^*/n$, the second term is given by $F_2 := \mathbb{E}\big[\widetilde{U}\beta^\tau(\beta^\tau)^*\widetilde{U}^*\big]$, and

$$F_3 \;:=\; \mathbb{E}\Big[(\alpha^\tau - \sqrt{n}\bar{\theta})\,\frac{\vec{1}}{\sqrt{n}}\,(\beta^\tau)^*\widetilde{U}^*\Big] \,+\, \mathbb{E}\Big[(\alpha^\tau - \sqrt{n}\bar{\theta})\,\widetilde{U}\beta^\tau\,\frac{\vec{1}^*}{\sqrt{n}}\Big].$$

Since $\widetilde{U}$ has orthonormal columns, all orthogonal to the all one vector ($\vec{1}^*\widetilde{U} = \vec{0}$), it follows that $\operatorname{trace}(F_2) = \mathbb{E}\big[\|\beta^\tau\|_2^2\big]$, and $\operatorname{trace}(F_3) = 0$.

It remains to compute $\operatorname{trace}(F_1)$. Unwrapping the recursion (5.11) and using the fact that initialization θ^0 implies $\alpha^0 = \sqrt{n}\bar{\theta}$ yields

$$\alpha^\tau \;=\; \sqrt{n}\bar{\theta} \,-\, \sum_{l=0}^{\tau-1}\epsilon^l\,\langle\frac{\vec{1}}{\sqrt{n}},\,v^l\rangle, \tag{C.1}$$

for all $\tau = 1, 2, \ldots$. Since v^l, $l = 0, 1, \ldots, \tau-1$, are zero mean random vectors, from equation (C.1) we conclude that $\mathbb{E}[\alpha^\tau] = \sqrt{n}\bar{\theta}$ [1] and accordingly, $\operatorname{trace}(F_1) = \operatorname{var}\big(\alpha^\tau\big)$. Recalling the definition of the MSE (5.1) and combining the pieces yields the claim (5.13).

Part (a): From equation (C.1), it is clear that each α^τ is Gaussian with mean $\sqrt{n}\bar{\theta}$. It remains to bound the variance. Using the i.i.d. nature of the sequence $v^l \sim N(0, C)$, we

[1] Here we have assumed that the initial values, θ_i^0, $i = 1, 2, \ldots, n$, are given (fixed).

have

$$\mathrm{var}\,\big(\alpha^{\tau}\big) \;=\; \mathbb{E}\Big[\big(\sum_{l=0}^{\tau-1} \epsilon^{l}\,\langle\frac{\vec{1}}{\sqrt{n}},\, v^{l}\rangle\big)^{2}\Big] \;=\; \sum_{l=0}^{\tau-1}\frac{(\epsilon^{l})^{2}}{n}\langle\vec{1},\, C\vec{1}\rangle \;=\; \sum_{l=0}^{\tau-1}(\epsilon_{1}^{l})^{2}\frac{\langle\vec{1},\, C'\vec{1}\rangle}{n},$$

where we have recalled the rescaled quantities (5.7). Recalling the fact that $C'_{ii} \leq \sigma^2$ and using the Cauchy-Schwarz inequality, we have $C'_{ij} \leq \sqrt{C'_{ii}C'_{jj}} \;\leq\; \sigma^2$. Hence, for $\delta \in (0,1)$, we obtain

$$\begin{aligned}\mathrm{var}\,\big(\alpha^{\tau}\big) \;&\leq\; n\,\sigma^{2}\sum_{l=0}^{\tau-1}(\epsilon_{1}^{l})^{2} \;=\; \frac{n\,\sigma^{2}}{[\lambda_{2}(\bar{S})]^{2}}\sum_{l=0}^{\tau-1}\frac{1}{(1/\delta+l)^{2}}\\ &\leq\; \frac{n\,\sigma^{2}}{[\lambda_{2}(\bar{S})]^{2}}\Big(\delta^{2} \,+\, \int_{1/\delta}^{\infty}\frac{1}{x^{2}}\,dx\Big) \;=\; \frac{n\,\sigma^{2}\,\delta\,(1+\delta)}{[\lambda_{2}(\bar{S})]^{2}} \;\leq\; \frac{2\,n\,\sigma^{2}\,\delta}{[\lambda_{2}(\bar{S})]^{2}};\end{aligned}$$

from which rescaling by $1/n$ establishes the bound (5.14).

Part (b): Defining $H(\beta^{\tau}, v^{\tau}) := \underline{L}^{\tau}\beta^{\tau} + \widetilde{U}^{*}v^{\tau}$, the update equation (5.12) can be written as

$$\beta^{\tau+1} \;=\; \beta^{\tau} \,-\, \epsilon^{\,\tau} H(\beta^{\tau}, v^{\tau}),$$

for $\tau = 1, 2, \ldots$. In order to upper bound $e_2^{\tau+1}$, defined in (5.13), we need to control $e_2^{\tau+1} - e_2^{\tau}$. Doing some algebra yields

$$\begin{aligned}e_{2}^{\tau+1} - e_{2}^{\tau} \;&=\; \frac{1}{n}\,\mathbb{E}\big[\langle\beta^{\tau+1} - \beta^{\tau},\, \beta^{\tau+1} + \beta^{\tau}\rangle\big]\\ &=\; \frac{1}{n}\,\mathbb{E}\big[\langle-\epsilon^{\,\tau}H(\beta^{\tau}, v^{\tau}),\, -\epsilon^{\,\tau}H(\beta^{\tau}, v^{\tau}) + 2\,\beta^{\tau}\rangle\big],\end{aligned}$$

and hence

$$e_{2}^{\tau+1} - e_{2}^{\tau} \;=\; \frac{(\epsilon^{\tau})^{2}}{n}\,\mathbb{E}\big[\|H(\beta^{\tau}, v^{\tau})\|_{2}^{2}\big] \,-\, \frac{2\epsilon^{\tau}}{n}\,\mathbb{E}\big[\langle H(\beta^{\tau}, v^{\tau}),\, \beta^{\tau}\rangle\big].$$

Since β^{τ} is independent of both L^{τ} and v^{τ}, by conditioning on the β^{τ} and using the tower property of expectation, we obtain

$$\mathbb{E}\big[\langle H(\beta^{\tau}, v^{\tau}),\, \beta^{\tau}\rangle\big] \;=\; \mathbb{E}\big[\langle\mathbb{E}\big[\underline{L}\big]\beta^{\tau},\, \beta^{\tau}\rangle\big].$$

By construction all the eigenvalues of $\mathbb{E}\big[\underline{L}\big]$ are greater than one, hence

$$\langle\mathbb{E}\big[\underline{L}\big]\,\beta^{\tau},\, \beta^{\tau}\rangle \;\geq\; \|\beta^{\tau}\|_{2}^{2}.$$

Putting the pieces together, we obtain

$$\begin{aligned} e_2^{\tau+1} &\leq \frac{1}{n}\,(\epsilon^\tau)^2\,\mathbb{E}\big[\|H(\beta^\tau,v^\tau)\|_2^2\big] \;+\;(1-2\epsilon^\tau)\,e_2^\tau \\ &= \frac{1}{n}\,(\epsilon^\tau)^2\,\underbrace{\mathbb{E}\big[\|\underline{L}^\tau\beta^\tau\|_2^2\big]}_{F_1} + \frac{1}{n}\,(\epsilon^\tau)^2\,\underbrace{\mathbb{E}\big[\|\widetilde{U}^* v^\tau\|_2^2\big]}_{F_2} + (1-2\epsilon^\tau)\,e_2^\tau, \end{aligned} \tag{C.2}$$

where we used the fact that $\mathbb{E}\big[\langle\underline{L}^\tau\beta^\tau,\,\widetilde{U}^*v^\tau\rangle\big]=0$. We continue by upper bounding the terms $F_1:=\mathbb{E}\big[\|\underline{L}^\tau\beta^\tau\|_2^2\big]$, and $F_2:=\mathbb{E}\big[\|\widetilde{U}^*v^\tau\|_2^2\big]$. First, we bound the former. By definition of the l_2-operator norm, we have

$$\mathbb{E}\big[\|\underline{L}^\tau\beta^\tau\|_2^2\big] \;\leq\; \mathbb{E}\big[|\!|\!|\underline{L}^\tau|\!|\!|_2^2\,\|\beta^\tau\|_2^2\big],$$

where $|\!|\!|\cdot|\!|\!|_2$ denotes the operator norm [47]. On the other hand, using the fact that $\underline{L}^\tau=\widetilde{U}^*(I-W^\tau)\widetilde{U}/\lambda_2(\bar{S})$ (recall the identities of the Section 5.4.1) yields[2]

$$|\!|\!|\underline{L}^\tau|\!|\!|_2 \;\leq\; \frac{1}{\lambda_2(\bar{S})}\,(1+|\!|\!|W^\tau|\!|\!|_2) \;=\; \frac{2}{\lambda_2(\bar{S})}.$$

Therefore, we have the following bound on F_1

$$F_1 \;\leq\; \frac{4}{[\lambda_2(\bar{S})]^2}\,\mathbb{E}\big[\|\beta^\tau\|_2^2\big]. \tag{C.3}$$

Turning to term F_2, we have

$$F_2 \;=\; \mathbb{E}\Big[(v^\tau)^*\big(I-\frac{\vec{1}\vec{1}^*}{n}\big)\,v^\tau\Big] \;\leq\; \operatorname{trace}\big(\operatorname{cov}(v^\tau)\big) \;\leq\; \frac{n\,\sigma^2}{[\lambda_2(\bar{S})]^2}. \tag{C.4}$$

Substituting the inequalities (C.3) and (C.4) into (C.2), we obtain the following recursive bound on $e_2^{\tau+1}$

$$e_2^{\tau+1} \;\leq\; \frac{\sigma^2\,(\epsilon^\tau)^2}{[\lambda_2(\bar{S})]^2} \;+\; \Big(1\;-\;2\epsilon^\tau\;+\;\frac{4\,(\epsilon^\tau)^2}{[\lambda_2(\bar{S})]^2}\Big)\,e_2^\tau.$$

Recall the definitions (5.7) and (5.9). If $\delta\leq[\lambda_2(\bar{S})]^2/4$, then we have

$$1-2\epsilon^\tau+\frac{4(\epsilon^\tau)^2}{[\lambda_2(\bar{S})]^2} \;\leq\; 1-\epsilon^\tau,$$

[2] Let v be an eigenvector of the matrix W^τ corresponding to the eigenvalue $\lambda\neq 1$. Since $\vec{1}^*v=0$, there exist an $(n-1)$-dimensional vector u such that $v=\widetilde{U}u$. Therefore we have,

$$\widetilde{U}^*(I-W^\tau)\widetilde{U}u \;=\; \widetilde{U}^*(I-W^\tau)v \;=\; (1-\lambda)\widetilde{U}^*v \;=\; (1-\lambda)u.$$

So by subtracting one from the eigenvalues of $\widetilde{U}^*(I-W^\tau)\widetilde{U}$, we obtain the non-one eigenvalues of W^τ.

and hence

$$e_2^{\tau+1} \;\leq\; \frac{\sigma^2\,(\epsilon^\tau)^2}{[\lambda_2(\bar{S})]^2} \;+\; (1-\epsilon^\tau)\, e_2^\tau, \tag{C.5}$$

for all $\tau = 1, 2, \ldots$. Unwrapping the inequality (C.5) yields

$$e_2^{\tau+1} \;\leq\; \frac{\sigma^2}{[\lambda_2(\bar{S})]^2} \sum_{k=0}^{\tau} (\epsilon^k)^2 \prod_{l=k+1}^{\tau} (1-\epsilon^l) \;+\; \prod_{l=0}^{\tau} (1-\epsilon^l)\, e_2^0. \tag{C.6}$$

On the other hand, the product $\prod_{l=k+1}^{\tau}(1-\epsilon^l)$ forms a telescopic series and is equal to $(k+1/\delta)/(\tau+1/\delta)$. Substituting this fact into the equation (C.6) yields

$$\begin{aligned} e_2^{\tau+1} \;&\leq\; \frac{\sigma^2}{[\lambda_2(\bar{S})]^2} \sum_{k=0}^{\tau} \frac{1}{(k+1/\delta)\,(\tau+1/\delta)} \;+\; e_2^0\, \frac{1/\delta - 1}{\tau + 1/\delta} \\ &\overset{(i)}{\leq}\; \frac{\sigma^2}{[\lambda_2(\bar{S})]^2}\, \frac{\log(\tau+1/\delta)}{\tau+1/\delta} \;+\; e_2^0\, \frac{1/\delta-1}{\tau+1/\delta}, \end{aligned}$$

where step (i) uses the following inequality

$$\sum_{k=0}^{\tau} \frac{1}{k+1/\delta} \;\leq\; \int_{1/\delta-1}^{\tau+1/\delta} \frac{1}{x}\, dx \;\leq\; \log(\tau+1/\delta),$$

valid for $\delta \in (0, 1/2)$.

C.2 Proof of Lemma 11

In the case of cycle there is only one averaging path and all the nodes are involved in that at each round so the averaging matrix, W, is fixed. More precisely, we have $\overline{W} = W = \vec{1}\vec{1}^*/n$. Therefore, $\overline{W}$ is a rank 1 matrix with $\lambda_{n-1}(\overline{W}) = 0$ and accordingly we have $\lambda_2(\bar{S}) = 1 - \lambda_{n-1}(\overline{W}) = 1$.

For the case of grid or random geometric graphs, we use the Poincare inequality [29]. A version of this theorem can be stated as follows: Let $A = [a_{ij}]$ denote the transition matrix of an irreducible aperiodic time reversible Markov chain with stationary distribution π. For each ordered pair of nodes (s, u) in the transition diagram, choose one and only one path $\eta_{su} = (s, s_1, s_2, \ldots, s_l, u)$ between s and u and define

$$|\eta_{su}| \;:=\; \frac{1}{\pi_s\, a_{ss_1}} \;+\; \frac{1}{\pi_{s_1}\, a_{s_1 s_2}} + \ldots + \frac{1}{\pi_{s_l}\, a_{s_l u}}. \tag{C.7}$$

Then the Poincare coefficient is

$$\kappa \;:=\; \max_{e \in E'} \sum_{\eta_{su} \ni e} |\eta_{su}|\, \pi_s\, \pi_u, \tag{C.8}$$

where E' is the set of directed edges formed in the previous step. Defining this quantity, the theorem states that $\lambda_{n-1}(A) \leq 1 - 1/\kappa$ or equivalently,

$$1 - \lambda_{n-1}(A) \;\geq\; \frac{1}{\kappa}. \tag{C.9}$$

We apply this theorem to the Markov chain formed by $\overline{W}$; the idea is to upper bound its Poincare coefficient.

C.2.1 Two Dimensional Grid

We first define a path η_{su} for every pair of nodes $\{s, u\}$. Two different cases can be distinguished here. For an illustration of the path η_{su} see Figure C.1.

Case 1: Nodes s and u do not belong to the same column or row. In this case, we consider a two-hop path $\eta_{su} = (s \to w \to u)$, where $w = (x_u, y_s)$ is the vertex of the rectangle constructed by s and u. Note that x_u is the x-coordinate of u and y_s is the y-coordinate of s. Since nodes $\{s, w\}$ and $\{w, u\}$ are averaged $1/2$ of the time, we have $\overline{W}_{sw} = \overline{W}_{wu} = 1/(2m)$. Substituting this into (C.7) and using the fact that $\pi = \vec{1}/n$ yields

$$|\eta_{su}| \;=\; \frac{1}{\overline{W}_{sw}\,\pi_s} + \frac{1}{\overline{W}_{wu}\,\pi_w} \;=\; 4mn.$$

Case 2: Nodes s and u belong to the same row or column. In this case, we set $\eta_{su} = (s \to u)$ which leads to

$$|\eta_{su}| \;=\; \frac{1}{\overline{W}_{su}\,\pi_s} \;=\; 2mn.$$

Moreover, a given edge $e = (s \to w)$ is involved in at most m paths. As node u varies in the corresponding column or row, we obtain $m - 1$ paths in case 1, and one path in case 2. Combining the pieces, we compute the Poincare coefficient

$$\kappa \;=\; \max_{e \in E'} \sum_{\eta_{su} \ni e} |\eta_{su}|\,\pi_s\,\pi_u \;\leq\; m\frac{4mn}{n^2} = 4.$$

Finally, from equation (C.9), we have

$$\lambda_2(\bar{S}) \;=\; 1 - \lambda_{n-1}(\overline{W}) \;\geq\; \frac{1}{\kappa} \;\geq\; \frac{1}{4},$$

which concludes the proof for the case of a grid-structured graph.

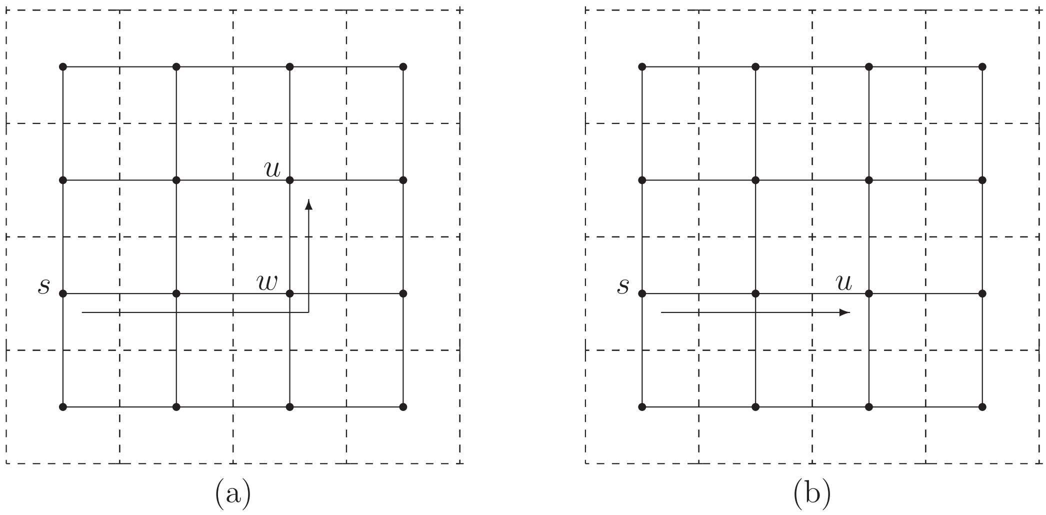


Figure C.1: Illustration of the path η_{su} for a grid-structured graph. (a) Case 1, where nodes s and u do not belong to the same column or row. (b) Case 2, where nodes s and u belong to the same column or row. This choice of η_{su} yield a tight upper bound on the Poincare coefficient.

C.2.2 Random Geometric Graph

For the RGG, we follow the same proof structure: namely, we first find a path for each pair of nodes $\{s, u\}$, and then upper bound the Poincare coefficient for the Markov chain $\overline{W}$. We first introduce some useful notation. Let $\mathcal{C} : \mathcal{V} \to \{1, 2, \ldots, m\}^2$ be the mapping that takes a node as its input and returns the sub-square of that node. More precisely, for some $s \in \mathcal{V}$ we have

$$\mathcal{C}(s) \;=\; (i, j) \quad \text{if}\, s \in (i, j)\text{-th square } i, j = 1, 2, \ldots, m.$$

Furthermore, we enumerate the nodes in square $\mathcal{C}(s) = (i, j)$ from 1 to n_{ij} where n_{ij} denotes the total number of nodes in $\mathcal{C}(s)$. Recall that by regularity assumption of the RGG we have $b \log n \geq n_{ij} \geq a \log n$ for some constants $b > a$. We refer to the label of node s as $\mathcal{N}_{\mathcal{C}(s)}(s)$ where $\mathcal{N}_{\mathcal{C}(s)}(\cdot)$ is the enumeration operator for the square $\mathcal{C}(s)$. Also let $n^* = \min_{i,j} n_{ij}$ denote the minimum number of nodes in one sub-square which by assumption is greater than $a \log n$. We split the problem into three different cases. Figure C.2 illustrates these there different cases.

Case 1: Nodes s and u do not belong to the the same column or row. In this case, a two hop path $\eta_{su} = (s \to w \to u)$ is considered. First, we pick $\mathcal{C}(w)$, the vertex of the rectangle constructed by $\mathcal{C}(s)$ and $\mathcal{C}(u)$ with the same x-coordinate as $\mathcal{C}(u)$ and the same y-coordinate as $\mathcal{C}(s)$. Now choose a node, w, inside $\mathcal{C}(w)$ such that

$$\mathcal{N}_{\mathcal{C}(w)}(w) \;=\; \mathcal{N}_{\mathcal{C}(s)}(s) + \mathcal{N}_{\mathcal{C}(u)}(u) \mod n^*. \tag{C.10}$$

Since each square has at least n^* nodes, such a choice can be made. On the other hand, since nodes in each square is picked uniformly at random in the averaging phase and there are at most $b \log n$ nodes in each square (for some constant b) we have $\overline{W}_{sw}, \overline{W}_{wu} \geq 1/(2m(b\log n)^2)$, where the factor of 2 is due to the choice of ζ, the averaging direction. Substituting this inequality into (C.7), we obtain

$$|\eta_{su}| \;=\; \frac{1}{\overline{W}_{sw}\,\pi_s} + \frac{1}{\overline{W}_{wu}\,\pi_w} \;\leq\; 4b^2 mn\,(\log n)^2.$$

Furthermore, from equation (C.10), we see that for a fixed s there are at most b/a nodes in the square $\mathcal{C}(u)$ that result in choosing w. Therefore, edge $e : (s \to w)$ is involved in at most $(m-1)\,b/a$ such paths.

Case 2: Nodes s and u belong to the same row or column. In this case, by setting $\eta_{su} = (s \to u)$, we obtain

$$|\eta_{su}| \;=\; \frac{1}{\overline{W}_{su}\,\pi_s} \;\leq\; 2b^2 mn\,(\log n)^2.$$

Note that there is only one path containing e of this type.

Case 3: Nodes s and u belong to the same square, meaning $\mathcal{C}(s) = \mathcal{C}(u)$. In this case a node w is chosen in a square adjacent to $\mathcal{C}(s)$ according to (C.10) such that $\mathcal{C}(w)$ is to the right of $\mathcal{C}(s)$; unless $\mathcal{C}(s)$ is in the last column, in which case $\mathcal{C}(w)$ is to the left of $\mathcal{C}(s)$. The same argument as case 1 would give us a bound on $|\eta_{su}|$. As for the upper bound on the number of paths: the edge $e : (s \to w)$ is involved in at most b/a such paths.

Combining all the pieces, we obtain

$$|\eta_{su}| \;\leq\; 4b^2 mn\,(\log n)^2 \quad \forall\, s, u \in \mathcal{V},$$

and

$$\max_{e \in E'} \sum_{s,u} \mathbb{I}\{\eta_{su} \ni e\} \;\leq\; m\,\frac{b}{a} + 1.$$

Substituting these two inequalities into (C.8) yields

$$\kappa \;\leq\; \left(m\,\frac{b}{a} + 1\right) \frac{4b^2\, mn\,(\log n)^2}{n^2} \;\leq\; \frac{2mb}{a}\,\frac{4b^2\, mn\,(\log n)^2}{n^2} \;=\; c_1 \log n$$

for some constant c_1. Therefore, from Poincare Theorem, we have

$$\lambda_2(\bar{S}) \;=\; 1 - \lambda_{n-1}(\overline{W}) \;\geq\; \frac{1}{\kappa} \;\geq\; \frac{1}{c_1 \log n},$$

which concludes the second part of Lemma 11.

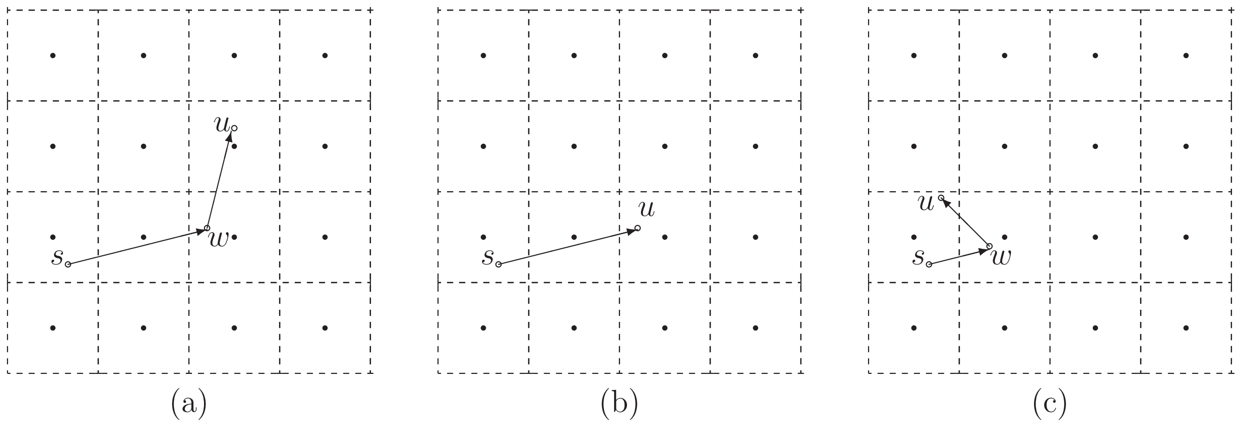


Figure C.2: Illustration of the path η_{su} for the case of RGG. (a) Case 1, where nodes s and u belong to the sub-squares in different row and columns (b) Case 2, where nodes s and u belong to the sub-squares in the same row or column. (c) Case 3, nodes s and u belong to the same square.

C.3 Proof of Part (a) of Theorem 11

We now return to the proof of part (a) of Theorem 11. Combining equations (5.10) and (C.1) yields

$$\theta^{\tau} = (\bar{\theta} - w^{\tau})\,\vec{1} + \widetilde{U}\beta^{\tau}, \tag{C.11}$$

where $w^{\tau} = \left[\sum_{l=0}^{\tau-1} \epsilon^{l}\,\langle \vec{1}/\sqrt{n},\, v^{l}\rangle\right]/\sqrt{n}$. As previously established, we know that $\mathbb{E}[w^{\tau}] = 0$ and $\text{var}(w^{\tau}) \leq 2\sigma^2\delta/[\lambda_2(\bar{S})]^2$ for all $\tau = 1, 2, \ldots$ and $\delta \in (0,1)$. Therefore, invoking a result on convergence of series with bounded variance (Theorem 8.3 from Chapter 1 of [34]), we have

$$w^{\tau} \xrightarrow{\text{a.s.}} w \quad \text{as } \tau \to \infty, \tag{C.12}$$

for some random variable w. Since w^{τ} is a sum of independent Gaussian random variables (and hence Gaussian), it is absolutely integrable [34]. Therefore, we have $\mathbb{E}[w] = \lim_{\tau\to\infty} \mathbb{E}[w^{\tau}] = 0$ and also $\text{var}(w) = \lim_{\tau\to\infty} \text{var}(w^{\tau}) \leq 2\sigma^2\delta/[\lambda_2(\bar{S})]^2$.

Now we move on to the next part of the proof, analyzing the sequence $\{\beta^{\tau}\}_{\tau=1}^{\infty}$. Recalling the update equation (5.12), our problem can be cast within the framework of the stochastic approximation theory, discussed in Chapter 2. In particular, the state sequence is $\{\beta^{\tau}\}_{\tau=1}^{\infty}$, the noise sequence is formed by zero-mean i.i.d. random vectors, the decreasing sequence is $\epsilon^{\tau} = 1/(\tau + 1/\delta)$, and finally $H(\beta, v) = -(\underline{L}\beta + \widetilde{U}^{*}v)$ is a linear function with the mean vector field $h(\beta) = \mathbb{E}[H(\beta, v) \mid \beta] = -\mathbb{E}[\underline{L}]\beta$. Note because we removed the zero eigenvalue from the average Laplacian matrix, the matrix $\mathbb{E}[\underline{L}]$ has all positive eigenvalues, and so $\gamma^{*} = 0$ is the unique stable point of the linear differential equation $d\gamma(\zeta)/d\zeta = -\mathbb{E}[\underline{L}]\gamma$. Therefore, an application of the Robbins Monro theorem 4 guarantees that

$$\beta^{\tau} \xrightarrow{a.s.} 0 \quad \text{as } \tau \to \infty. \tag{C.13}$$

Substituting the results (C.12) and (C.13) into equation (C.11), we obtain

$$\theta^{\tau} \xrightarrow{\text{a.s.}} (\bar{\theta} - w)\vec{1} \quad \text{as } \tau \to \infty.$$

In other words, nodes will almost surely reach a consensus; moreover, the consensus value, $\widetilde{\theta} = \bar{\theta} - w$, is within $2\sigma^2\delta/[\lambda_2(\bar{S})]^2$ distance of the true sample mean.

Bibliography

[1] D. Achilioptas and F. McSherry. On spectral learning of mixtures of distributions. In *18th Annual Conference on Learning Theory (COLT)*, July 2005.

[2] A. Agarwal, P. L. Bartlett, P. Ravikumar, and M. J. Wainwright. Information-theoretic lower bounds on the oracle complexity of stochastic convex optimization. *IEEE Transactions on Information Theory*, 58(5):3235–3249, May 2012.

[3] R. P. Agarwal, M. Meehan, and D. O'Regan. *Fixed Point Theory and Applications.* Cambridge University Press, 2004.

[4] S. M. Aji and R. J. McEliece. The generalized distributive law. *IEEE Transactions on Information Theory*, 46(2):325–343, March 2000.

[5] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Transaction on Signal Processing*, 50(2):174–188, 2002.

[6] O. Ayaso, D. Shah, and M. Dahleh. Information theoretic bounds for distributed computation over networks of point-to-point channels. *IEEE Transactions on Information Theory*, 56(12):6020–6039, 2010.

[7] T. C. Aysal, M. J. Coates, and M. G. Rabbat. Distributed average consensus with dithered quantization. *IEEE Transactions on Signal Processing*, 56:4905–4918, 2008.

[8] T. C. Aysal, M. E. Yildiz, A. D. Sarwate, and A. Scaglione. Broadcast gossip algorithms for consensus. *IEEE Transactions on Signal Processing*, 57:2748–2761, 2009.

[9] A. G. Dimakis B. Nazer and M. Gastpar. Neighborhood gossip: Concurrent averaging through local interference. In *Proc. IEEE ICASSP*, 2009.

[10] S. Baker, D. Scharstein, J. P. Lewis, S. Roth, M. Black, and R. Szeliski. A database and evaluation methodology for optical flow. *International Journal of Computer Vision*, 92(1):1–31, March 2011.

[11] F. Benezit, V. Blondel, P. Thiran, J. Tsitsiklis, and M. Vetterli. Weighted gossip: Distributed averaging using non-doubly stochastic matrices. In *Proc. IEEE International Symposium on Information Theory*, 2010.

[12] F. Benezit, A. G. Dimakis, P. Thiran, and M. Vetterli. Order-optimal consensus through randomized path averaging. *IEEE Transaction on Information Theory*, 56(10):5150–5167, 2010.

[13] A. Benveniste, M. Metivier, and P. Priouret. *Stochastic Approximations and Adaptive Algorithms*. Springer-Verlag New York, Inc., New York, 1990.

[14] J. Besag. Spatial interaction and the statistical analysis of lattice systems. *J. Roy. Stat. Soc. Series B*, 36:192–236, 1974.

[15] G. Boccignone, A. Marcelli, P. Napoletano, and M. Ferraro. Motion estimation via belief propagation. In *Proceedings of the International Conference on Image Analysis and Processing*, 2007.

[16] B. Bollobas. *Modern Graph Theory*. Springer-Verlag, New York, 1998.

[17] V. S. Borkar. *Stochastic Approximation: A Dynamical System Viewpoint*. Cambridge University Press, Cambridge, UK, 2008.

[18] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah. Randomized gossip algorithms. *IEEE Transactions on Information Theory*, 52:2508–2530, 2006.

[19] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, Cambridge, UK, 2004.

[20] M. Briers, A. Doucet, and S. S. Singh. Sequential auxiliary particle belief propagation. In *Proceedings of the 8th International Conference on Information Fusion*, pages 826–834, 2005.

[21] O. Cappe, E. Moulines, and T. Ryden. *Inference in Hidden Markov Models*. Springer, New York, 2010.

[22] F. Cattivelli and A. H. Sayed. Diffusion lms strategies for distributed estimation. *IEEE Transactions on Signal Processing*, 58(3):1035–1048, March 2010.

[23] H. Chen. *Stochastic Approximation and its Applications*. Kluwer Academic Publishers, Dordrecht, Netherlands, 2002.

[24] F. Chung and L. Lu. Concentration inequalities and martingale inequalities: A survey. *Internet Mathematics*, 3(1):79–127, 2006.

[25] Fan R. K. Chung. *Spectral Graph Theory.* American Mathematical Society, 1997.

[26] F. Clarke. *Functional Analysis, Calculus of Variations and Optimal Control.* Springer-Verlag, London, 2013.

[27] J. Coughlan and H. Shen. Dynamic quantization for belief propagation in sparse spaces. *Computer Vision and Image Understanding*, 106(1):47–58, 2007.

[28] M. H. deGroot. Reaching a consensus. *Journal of the American Statistical Association*, 69(345):118–121, March 1974.

[29] P. Diaconis and D. Stroock. Geometric bounds for eigenvalues of Markov chains. *Ann. Applied Probability*, 1:36–61, 1991.

[30] R. Diestel. *Graph Theory.* Springer-Verlag, New York, 2000.

[31] A. G. Dimakis, A. Sarwate, and M. J. Wainwright. Geographic gossip: Efficient averaging for sensor networks. *IEEE Trans. Signal Processing*, 53:1205–1216, March 2008.

[32] A. Doucet, N. de Freitas, and N. Gordon. *Sequential Monte Carlo Methods in Practice.* Springer, New York, 2001.

[33] J. Duchi, A. Agawarl, and M. J. Wainwright. Dual averaging for distributed optimization: Convergence analysis and network scaling. Technical Report arXiv:1005.2012, UC Berkeley, May 2010.

[34] R. Durrett. *Probability: Theory and Examples.* Duxbury Press, New York, NY, 2005.

[35] V. N. Ekambaram and K. Ramchandran. Distributed high accuracy peer-to-peer localization in mobile multipath enviroments. In *IEEE Global Communications Conference*, pages 1–5, 2010.

[36] M. Fabian, P. Habala, P. Hajek, V. Montesinos, and V. Zizler. *Banach Space Theory: The Basis for Linear and Nonlinear Analysis.* Springer, New York, 2011.

[37] F. Fagnani and S. Zampieri. Average consensus with packet drop communication. *SIAM J. on Control and Optimization*, 2007. To appear.

[38] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient belief propagation for early vision. *International Journal of Computer Vision*, 70(1):41–54, 2006.

[39] R. G. Gallager. *Low-Density Parity-Check Codes.* PhD thesis, Cambridge, MA, 1963.

[40] G. R. Grimmett. A theorem about random fields. *Bulletin of the London Mathematical Society*, 5:81–84, 1973.

[41] G.R. Grimmett and D.R. Stirzaker. *Probability and Random Processes.* Oxford Science Publications, Clarendon Press, Oxford, 1992.

[42] C. Gu. *Smoothing spline ANOVA models.* Springer Series in Statistics. Springer, New York, NY, 2002.

[43] P. Gupta and P. Kumar. The capacity of wireless networks. *IEEE Trans. on Inf. Theory*, 46(2):388–404, Mar 2000.

[44] J. M. Hammersley and D. C. Handscomb. *Monte Carlo Methods.* Barnes and Noble, New York, 1964.

[45] Y. Hatano, A. K. Das, and M. Mesbahi. Agreement in presence of noise: pseudogradients on random geometric networks. In *Proceedings of the 44th IEEE Conference on Decision and Control*, December 2005.

[46] E. Hazan, A. Agarwal, and S. Kale. Logarithmic regret algorithms for online convex optimization. *Journal of Machine Learning*, 69(2-3):169–192, 2007.

[47] R. A. Horn and C. R. Johnson. *Matrix Analysis.* Cambridge University Press, Cambridge, 1985.

[48] Y. Hu, H. Chen, J. Lou, and J. Li. Distributed density estimation using nonparameteric statistics. In *Proceedings of the 27th International Conference on Distributed Computing Systems*, pages 28–36, 2007.

[49] A. T. Ihler, J. W. Fisher, and A. S. Willsky. Loopy belief propagation: convergence and effects of message errors. *Journal of Machine Learning Research*, 6:905–936, May 2005.

[50] A. T. Ihler, A. J. Frank, and P. Smyth. Particle-based variational inference for continuous systems. In *Proceedings Advances in Neural Information Processing Systems (NIPS)*, pages 826–834, 2009.

[51] A. T. Ihler and D. McAllester. Particle belief propagation. In *Proceedings Conference on Artificial Intelligence and Statistics, Clearwater, Florida, USA*, pages 256–263, 2009.

[52] M. Isard. PAMPAS: Real-valued graphical models for computer vision. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 613–620, 2003.

[53] M. Isard, J. MacCormick, and K. Achan. Continuously-adaptive discretization for message-passing algorithms. In *Proceedings Advances in Neural Information Processing Systems, Vancouver, Canada*, pages 737–744, 2009.

[54] A. Jovicic, I. Klimek, C. Measson, T. Richardson, and L. Zhang. Mobile device positioning using learning and cooperation. In *46th Annual Conference on Information Sciences and Systems*, pages 1–6, 2012.

[55] A. Juditsky, G. Lan, A. Nemirovski, and A. Shapiro. Stochastic approximation approach to stochastic programming. *SIAM Journal on Optimization*, 19(4):1574–1609, 2009.

[56] S. Kar and J. M. F. Moura. Distributed consensus algorithm in sensor networks with imperfect communication: link failures and channel noise. *IEEE Transactions on Signal Processing*, 57(5):355–369, Jan 2009.

[57] D. Kempe, A. Dobra, and J. Gehrke. Gossip-based computation of aggregate information. In *Proc. IEEE Conf. Foundation of Computer Science (FOCS)*, 2003.

[58] K. Kersting, B. Ahmadi, and S. Natarajan. Counting belief propagation. In *Proceedings Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, Montreal, Canada*, 2009.

[59] J. Kiefer and J. Wolfowitz. Stochastic estimation of the maximum of a regression function. *Annals of Mathematical Statistics*, 23:462–466, 1952.

[60] A. Klaus, M. Sormann, and K. Karner. Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure. In *Proceedings 18th International Conference on Pattern Recognition, Hong Kong*, pages 15–18, 2006.

[61] D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, Cambridge, 2009.

[62] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger. Factor graphs and the sum-product algorithm. *IEEE Transaction on Information Theory*, 47(2):498–519, 2001.

[63] H. J. Kushner. General convergence results for stochastic approximations via weak convergence theory. *Journal of mathematical analysis and applications*, 61:490–503, 1977.

[64] H. J. Kushner and D. S. Clark. *Stochastic Approximation for Constrained and Unconstrained Systems*. Springer-Verlag, Berlin and New York, 1978.

[65] H. J. Kushner and G. G. Yin. *Stochastic Approximation and Recursive Algorithms and Applications*. Springer-Verlag New York, Inc., New York, 2003.

[66] X. Lan, S. Roth, D. Huttenlocher, and M. J. Black. Efficient belief propagation with learned higher-order markov random fields. *Lecture Notes in Computer Science*, 3952:269–282, 2006.

[67] N. E. Leonard, D. A. Paley, F. Lekien, R. Sepulcher, D. M. Fratantoni, and R. E. Davis. Colective motion, sensor networks, and ocean sampling. *Proceedings of the IEEE*, 95(1):48–74, 2007.

[68] J. Lindenstrauss, D. Preiss, and J. Tiser. *Frechet Differentiability of Lipschitz Functions and Porous Sets in Banach Spaces.* Princeton University Press, New Jersey, 2012.

[69] J. S. Liu. *Monte Carlo Strategies in Scientific Computing.* Springer-Verlag, New York, NY, 2001.

[70] L. Ljung. Analysis of recursive stochastic algorithms. *IEEE Transactions on Automatic Control*, 22:551–575, 1977.

[71] L. Ljung, G. Pflug, and H. Walk. *Stochastic Approximation and Optimization of Random Systems.* Birkhauser Verlag Basel, Berlin, Germany, 1992.

[72] L. Ljung and T. Soderstorm. *Theory and Practice of Recursive Identification.* MIT Press, Cambridge, USA, 1983.

[73] C. G. Lopes and A. H. Sayed. Incremental adaptive strategies over distributed networks. *IEEE Transactions on Signal Processing*, 55(8):4064–4077, August 2007.

[74] C. G. Lopes and A. H. Sayed. Diffusion least-mean squares over adaptive networks: Formulation and performance analysis. *IEEE Transactions on Signal Processing*, 56(7):3122–3136, July 2008.

[75] D. G. Luenberger. *Optimization by Vector Space Methods.* Wiley, New York, 1969.

[76] J. J. McAuley and T. S. Caetano. Faster algorithms for max-product message passing. *Journal of Machine Learning Research*, 12:1349–1388, 2011.

[77] T. Meltzer, C. Yanover, and Y. Weiss. Globally optimal solutions for energy minimization in stereo vision using reweighted belief propagation. In *International Conference on Computer Vision*, June 2005.

[78] J. M. Mooij and H. J. Kappen. Sufficient conditions for convergence of the sum-product algorithm. *IEEE Transactions on Information Theory*, 53(12):4422–4437, December 2007.

[79] C. Morelli, M. Nicoli, V. Rampa, and U. Spagnolini. Hidden Markov models for radio localization in mixed LOS/NLOS conditions. *IEEE Transaction on Signal Processing*, 55:1525–1542, 2007.

[80] A. Nedic and A. Ozdaglar. Distributed subgradient methods for multi-agent optimization. *IEEE Transactions on Automatic Control*, 54:48–61, 2009.

[81] A. S. Nemirovsky and D. B. Yudin. *Problem Complexity and Method Efficiency in Optimization.* New York, 1983.

[82] N. Noorshams and M. J. Wainwright. A near-optimal algorithm for network-constrained averaging with noisy links. In *Proceedings of the IEEE International Symposium on Information Theory*, 2010.

[83] N. Noorshams and M. J. Wainwright. Non-asymptotic analysis of an optimal algorithm for network-constrained averaging with noisy links. *IEEE Journal of Selected Topics in Signal Processing*, 5(4):833–844, August 2011.

[84] N. Noorshams and M. J. Wainwright. Stochastic belief propagation: Low-complexity message-passing with guarantees. In *Proceedings of the 49th Annual Allerton Conference on Communication, Control, and Computing*, 2011.

[85] N. Noorshams and M. J. Wainwright. Belief propagation for continuous state spaces: Stochastic message-passing with quantitative guarantees. Submitted to the Journal of Machine Learning Research, December 2012.

[86] N. Noorshams and M. J. Wainwright. Quantized stochastic belief propagation: Efficient message-passing for continuous state spaces. In *Proceedings of the IEEE International Symposium on Information Theory*, 2012.

[87] N. Noorshams and M. J. Wainwright. Stochastic belief propagation: A low-complexity alternative to the sum-product algorithm. *IEEE Transaction on Information Theory*, 59(4):1981–2000, April 2013.

[88] A. V. Oppenheim and R. W. Schafer. *Digital Signal Processing.* Prentice-Hall, Englewood Cliffs, NJ, 1975.

[89] A. V. Oppenheim, S. Willsky, and H. Nawab. *Signals and Systmes.* Prentice-Hall, Englewood Cliffs, NJ, 1997.

[90] G. Parisi. *Statistical Field Theory.* Addison-Wesley, 1988.

[91] A. S. Paul and E. A. Wan. RSSI-Based indoor localization and tracking using sigma-point kalman smoothers. *IEEE Journal of Selected Topics in Signal Processing*, 3(5):860–873, 2009.

[92] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan Kaufman, San Mateo, 1988.

[93] J. P. Penot. *Calculus Without Derivatives.* Springer, New York, 2013.

[94] M. Penrose. *Oxford studies in probability, Random Geometric Graphs.* Oxford Univ. Press, Oxford U.K., 2003.

[95] R. Rajagopal and M. J. Wainwright. Network-based consensus averaging with general noisy channels. *IEEE Transaction on Signal Processing*, 59(1):373–385, 2011.

[96] S. Sundhar Ram, A. Nedic, and V. V. Veeravalli. Distributed subgradient projection algorithm for convex optimization. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 3653–3656, 2009.

[97] A. C. Rapley, C. Winstead, V. C. Gaudet, and C. Schlegel. Stochastic iterative decoding on factor graphs. In *Proceedings 3rd International Symposium on Turbo Codes and Related Topics, Brest, France*, pages 507–510, 2003.

[98] F. Reichenbach and D. Timmermann. Indoor localization with low complexity in wireless sensor networks. In *IEEE Internaitonal Conference on Industrial Informatics*, pages 1018–1023, 2006.

[99] T. Richardson and R. Urbanke. *Modern Coding Theory.* Cambridge University Press, 2008.

[100] F. Riesz and B.S. Nagy. *Functional Analysis.* Dover Publications Inc., New York, 1990.

[101] H. Robbins and S. Monro. A stochastic approximation method. *Annals of Mathematical Statistics*, 22:400–407, 1951.

[102] C. P. Robert and G. Casella. *Monte Carlo Statistical Methods.* Springer-Verlag, New York, 1999.

[103] T. G. Roosta, M. J. Wainwright, and S. S. Sastry. Convergence analysis of reweighted sum-product algorithms. *IEEE Transactions on Signal Processing*, 56(9):4293–4305, September 2008.

[104] H. L. Royden. *Real Analysis.* Prentice-Hall, New Jersey, 1988.

[105] H. Song and J. R. Cruz. Reduced-complexity decoding of q-ary LDPC codes for magnetic recording. *IEEE Transaction on Magnetics*, 39(2):1081–1087, 2003.

[106] L. Song, A. Gretton, D. Bickson, Y. Low, and C. Guestrin. Kernel belief propagation. In *Proceedings Artificial Intelligence and Statistics, Ft. Lauderdale, Florida, USA*, 2011.

[107] A. N. Srivastava and M. Sahami. *Text Mining: Classification, Clustering, and Applications.* Chapman-Hall, Boca Raton, 2009.

[108] I. Steinwart and A. Christmann. *Support Vector Machines.* Springer, New York, 2008.

[109] Han I Su and A. El Gamal. Distributed lossy averaging. In *Proc. IEEE International Symposium on Information Theory*, 2009.

[110] E. B. Sudderth, A. T. Ihler, W. T. Freeman, and A. S. Willsky. Nonparametric belief propagation. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition, Madison, Wisconsin, USA*, volume 1, pages 605–612, 2003.

[111] E. B. Sudderth, A. T. Ihler, M. Israd, W. T. Freeman, and A. S. Willsky. Nonparametric belief propagation. *Communications of the ACM Magazine*, 53(10):95–103, 2010.

[112] J. Sun, H. Y. Shum, and N. N. Zheng. Stereo matching using belief propagation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(7):787–800, 2003.

[113] S. Tatikonda and M. I. Jordan. Loopy belief propagation and Gibbs measures. In *Proceedings 18th confernce on Uncertainty in Artificial Intelligence, Alberta, Canada*, volume 18, pages 493–500, August 2002.

[114] S. S. Tehrani, W. J. Gross, and S. Mannor. Stochastic decoding of LDPC codes. *IEEE Communications Letters*, 10(10):716–718, 2006.

[115] D. Tse and P. Viswanath. *Fundamentals of Wireless Communication.* Cambridge University Press, Cambridge, UK, 2005.

[116] J. Tsitsiklis. *Problems in decentralized decision-making and computation.* PhD thesis, Department of EECS, MIT, 1984.

[117] W. Tsujita, A. Yoshino, H. Ishida, and T. Moriizumi. Gas sensor network for air-pollution monitoring. *Elsevier Journal of Sensors and Actuators B: Chemical*, 110(2):304–311, 2005.

[118] M. J. Wainwright, T. S. Jaakkola, and A. S. Willsky. A new class of upper bounds on the log partition function. *IEEE Transaction on Information Theory*, 51(7):2313–2335, July 2005.

[119] M. J. Wainwright and M. I. Jordan. *Graphical Models, Exponential Families, and Variational Inference.* Now Publishers Inc, Hanover, MA 02339, USA, 2008.

[120] G. Werner-Allen, K. Lorincz, M. Welsh, O. Marcillo, J. Johnson, M. Ruiz, and J. Lees. Deploying a wireless sensor network on an active volcano. *IEEE Journal of Internet Computing*, 10(2):18–25, 2006.

[121] J. S. Yedidia, W. T. Freeman, and Y. Weiss. Constructing free energy approximations and generalized belief propagation algorithms. *IEEE Transaction on Information Theory*, 51(7):2282–2312, July 2005.

[122] L. Yu, N. Wang, and X. Meng. Real-time forest fire detection with wireless sensor networks. In *International Conference on Wireless Communications, Networking and Mobile Computing*, pages 1214–1217, 2005.