Approximate Inference in Gaussian Graphical Models

by

Dmitry M. Malioutov

Submitted to the Department of Electrical Engineering and Computer Science in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Electrical Engineering and Computer Science
at the Massachusetts Institute of Technology

June, 2008

Signature of Author: _____

Department of Electrical Engineering and Computer Science
May 23, 2008

Certified by: _____

Alan S. Willsky, Professor of EECS
Thesis Supervisor

Accepted by: _____

Terry P. Orlando, Professor of Electrical Engineering
Chair, Committee for Graduate Students

Approximate Inference in Gaussian Graphical Models

by Dmitry M. Malioutov

Submitted to the Department of Electrical Engineering
and Computer Science on May 23, 2008
in Partial Fulfillment of the Requirements for the Degree
of Doctor of Philosophy in Electrical Engineering and Computer Science

Abstract

The focus of this thesis is approximate inference in Gaussian graphical models. A graphical model is a family of probability distributions in which the structure of interactions among the random variables is captured by a graph. Graphical models have become a powerful tool to describe complex high-dimensional systems specified through local interactions. While such models are extremely rich and can represent a diverse range of phenomena, inference in general graphical models is a hard problem.

In this thesis we study Gaussian graphical models, in which the joint distribution of all the random variables is Gaussian, and the graphical structure is exposed in the inverse of the covariance matrix. Such models are commonly used in a variety of fields, including remote sensing, computer vision, biology and sensor networks. Inference in Gaussian models reduces to matrix inversion, but for very large-scale models and for models requiring distributed inference, matrix inversion is not feasible.

We first study a representation of inference in Gaussian graphical models in terms of computing sums of weights of walks in the graph – where means, variances and correlations can be represented as such walk-sums. This representation holds in a wide class of Gaussian models that we call walk-summable. We develop a walk-sum interpretation for a popular distributed approximate inference algorithm called loopy belief propagation (LBP), and establish conditions for its convergence. We also extend the walk-sum framework to analyze more powerful versions of LBP that trade off convergence and accuracy for computational complexity, and establish conditions for their convergence.

Next we consider an efficient approach to find approximate variances in large scale Gaussian graphical models. Our approach relies on constructing a low-rank aliasing matrix with respect to the Markov graph of the model which can be used to compute an approximation to the inverse of the information matrix for the model. By designing this matrix such that only the weakly correlated terms are aliased, we are able to give provably accurate variance approximations. We describe a construction of such a low-rank aliasing matrix for models with short-range correlations, and a wavelet-based construction for models with smooth long-range correlations. We also establish accuracy guarantees for the resulting variance approximations.

Thesis Supervisor: Alan S. Willsky
Title: Professor of Electrical Engineering and Computer Science

Notational Conventions

Symbol	Definition
General Notation	
$\lvert \cdot \rvert$	absolute value
x_i	the ith component of the vector x
X_{ij}	element in the ith row and jth column of matrix X
$(\cdot)^T$	matrix or vector transpose
$(\cdot)^{-1}$	matrix inverse
$\det(\cdot)$	determinant of a matrix
$\mathrm{tr}(\cdot)$	trace of a matrix
$\mathbb{R}$	real numbers
$\mathbb{R}^N$	vector space of real-valued N-dimensional vectors
I	identity matrix
$p(x)$	probability distribution of a random vector x
$p_i(x_i)$	marginal probability distribution of x_i
$p(x \mid y)$	conditional probability distribution of x given y
$\mathbb{E}[\cdot]$	expected value
$\mathbb{E}_x[\cdot]$	expected value, expectation is over $p(x)$
$\varrho(\cdot)$	spectral radius of a matrix
Graph theory	
G	undirected graph
V	vertex or node set of a graph
$\mathcal{E}$	edge set of a graph
$\lvert V \rvert$	number of nodes, i.e. cardinality of the set V
2^V	set of all subsets of V
$A \backslash B$	set difference
$V \backslash i$	all vertices except i, shorthand for $V \backslash \{i\}$
$\{i, j\}$	an undirected edge in a graph (unordered pair)
(i, j)	a directed edge (ordered pair)
$\mathcal{N}(i)$	set of neighbors of node i
$\mathcal{H}$	hypergraph
$\mathcal{F}$	the collection of hyperedges in a hypergraph

Symbol	Definition
Graphical models	
ψ_i	single-node potential
ψ_{ij}	edge potential
ψ_F	factor potential
Z	normalization constant
F	factor, a subset of nodes
x_F	subvector of x index by elements of F
$m_{i\to j}$	message from i to j in BP
$m_{A\to i}, m_{i\to A}$	messages in factor graph version of BP
$\Delta J_{i\to j}$, $\Delta h_{i\to j}$	messages in Gaussian BP
$\Delta J_{A\to i}$, $\Delta h_{A\to i}$	messages in Gaussian FG-LBP
$T_i^{(n)}$	n-step LBP computation tree rooted at node i
$T_{i\to j}^{(n)}$	n-step LBP computation tree for message $m_{i\to j}$
$\mathcal{N}(\mu, P)$	Gaussian distribution with mean μ and covariance P
J	Information matrix for a Gaussian distribution
h	potential vector for a Gaussian distribution
J_F	a submatrix of J indexed by F
$[J_F]$	J_F zero-padded to have size $N \times N$
$\lambda_{\min}(J)$	smallest eigenvalue of J
r_{ij}	partial correlation coefficient
Walk-sums	
R	partial correlation matrix
$\bar{R}$	matrix of elementwise absolute values of R
w	a walk
$w : i \to j$	set of walks from i to j
$w : * \to j$	set of walks that start anywhere and end at j
$w : i \xrightarrow{l} j$	set of walks from i to j of length l
$\phi(w)$	weight of a walk
$\mathcal{W}$	a collection of walks
$\mathcal{W}(i \to i)$	self-return walks
$\mathcal{W}(i \xrightarrow{\setminus i} i)$	single-revisit self-return walks
$\mathcal{W}(* \xrightarrow{\setminus i} i)$	single-visit walks

Symbol	Definition
walk-sums (continued)	
$\phi(\mathcal{W})$	walk-sum
$\phi(i \to i)$	self-return walk-sum
$\phi_h(\mathcal{W})$	input reweighted walk-sum
$R_i^{(n)}$	partial correlation matrix for computation tree $T_i^{(n)}$
ϱ_∞	limit of the spectral radius of $R_i^{(n)}$
$\mathcal{Q}_G$	set of block-orthogonal matrices on G
$\mathcal{S}_G$	set of block-invertible matrices on G
$\bar{\phi}_k$	a matrix of absolute walk-sums for walks of length k
Low-rank variance approximation	
$\hat{P}$	approximation of P
P_i	ith column of P
v_i	ith standard basis vector
BB^T	low-rank aliasing matrix
B	a spliced basis
b_i	ith row of B corresponding to node i
B_k	kth column of B
R_k	solution to the system $JR_k = B_k$
σ_i	random sign for node i
E	error in covariance, $\hat{P} - P$
$\mathcal{C}(i)$	set of nodes of the same color as i
$Var(\cdot)$	variance of a random variable
$\phi_{s,k}(t)$	kth wavelet function at scale s
$\psi_{s,k}(t)$	kth scaling function at scale s
W	a wavelet basis

Acknowledgments

I have been very fortunate to work under the supervision of Professor Alan Willsky in the wonderful research environment that he has created in the Stochastic Systems Group. Alan's deep and extensive knowledge and insight, energy, enthusiasm, and the ability to clearly explain even the most obscure concepts in only a few words are remarkable and have been very inspiring throughout the course of my studies. There are too many reasons to thank Alan – from providing invaluable guidance on research, and allowing the freedom to explore topics that interest me the most, to financial support, to giving me an opportunity to meet and interact with world-class researchers through SSG seminars and sending me to conferences[1], and finally for the extremely prompt and yet very careful review process of my thesis. I would like to thank my committee members, Professor Pablo Parrilo, and Professor William Freeman, for suggesting interesting research directions, and for detailed reading of my thesis.

It has been a very rewarding experience interacting with fellow students at SSG – from discussing and collaborating on research ideas, and providing thoughtful criticism and improvements of papers and presentations, to relaxing after work. In particular I'd like to thank Jason Johnson who has been my office-mate throughout my graduate studies. Much of the work in this thesis has greatly benefited from discussions with Jason – he has always been willing to spend time and explain topics from graphical models, and he has introduced me to the walk-sum expansion of the inverse of the information matrix, which has lead to exciting collaboration the product of which now forms the bulk of my thesis. In addition to research collaboration, Jason has become a good friend, and among a variety of other things he introduced me to the idea of sparsifying some of the ill-formed neural connections by going to the Muddy on Wednesdays, and taught me the deadly skill of spinning cards. I would also like to thank Sujay Sanghavi for generously suggesting and collaborating on a broad range of interesting ideas, some of which have matured into papers (albeit, on topics not directly related to the title of this thesis). I also acknowledge interactions with a number of researchers both at MIT and outside. I would like to thank Devavrat Shah, David Gamarnik, Vivek Goyal, Venkatesh Saligrama, Alexander Postnikov, Dimitri Bertsekas, Mauro Maggioni, Alfred Hero, Mujdat Cetin, John Fisher, Justin Dauwels, Eric Feron, Mardavij Roozbehani, Raj Rao, Ashish Khisti, Shashi Borade, Emin Martinian, and Dmitry Vasiliev. I especially would like to thank Hanoch Lev-Ari for interesting discussions on relation of walk-sums with electrical circuits. I learned a lot about seismic signal processing from

[1] Although for especially exotic locations, such as Hawaii, it sometimes took quite a bit of convincing why the submitted work was innovative and interesting.

Jonathan Kane and others at Shell. Also I greatly enjoyed spending a summer working on distributed video coding at MERL under the supervision of Jonathan Yedidia and Anthony Vetro. I would like to thank my close friends Gevorg Grigoryan and Nikolai Slavov for ongoing discussions on relations of graphical models and biology.

My years at MIT have been very enjoyable thanks in large part to the great fellow students at SSG. I would like to thank Jason Johnson, Ayres Fan and Lei Chen for making our office such a pleasant place to work, or rather a second home, for a number of years. Special thanks to Ayres for help with the many practical sides of life including the job search process and (in collaboration with Walter Sun) for teaching me not to keep my rubles under the pillow. I very much enjoyed research discussions and playing Hold'em with Venkat Chandrasekaran, Jin Choi, and Vincent Tan. Vincent – a day will come when I will, with enough practice, defeat you in ping-pong. Thanks to Kush Varshney for eloquently broadcasting the news from SSG to the outside world, to Pat Kreidl and Michael Chen for their sage, but very contrasting, philosophical advice about life. Thanks to our dynamite-lady Emily Fox for infusing the group with lively energy, and for proving that it is possible to TA, do research, cycle competitively, play hockey, and serve on numerous GSC committees all at the same time. Thanks to the student-forever Dr. Andy Tsai for gruesome medical stories during his brief vacations away from medical school which he chose to spend doing research at SSG. I also enjoyed interacting with former SSG students – Walter Sun, Junmo Kim, Eric Sudderth, Alex Ihler, Jason Williams, Lei Chen, Martin Wainwright and Dewey Tucker. Many thanks to Brian Jones for timely expert help with computer and network problems, and to Rachel Cohen for administrative help.

During my brief encounters outside of SSG I have enjoyed the company of many students at LIDS, CSAIL, and greater MIT. Many thanks for the great memories to the 5-th floor residents over the years, in particular the french, the italians (my tortellini-cooking friends Riccardo, Gianbattista, and Enzo), the exotic singaporeans (big thanks to Abby for many occasions), and the (near)-russians, Tolya, Maksim, Ilya, Evgeny, Nikolai, Michael, Grisha. Thanks to Masha and Gevorg for being great friends, and making sure that I do not forget to pass by the gym or the swimming pool once every few weeks. Many thanks for all my other russian friends for good times and for keeping my accent always strong. I was thrilled to learn brazilian rhythms, and even occasionally perform, with Deraldo Ferreira, and his earth-shaking samba tremeterra, and Marcus Santos. I would especially like to thank Wei for her sense of humour, her fashion advice, and in general for being wonderful.

Finally I thank all of my family, my MIT-brother Igor, and my parents for their constant support and encouragement. I dedicate this thesis to my parents.

Contents

Chapter 1

Introduction

Analysis and modeling of complex high-dimensional data has become a critical research problem in the fields of machine learning, statistics, and many of their applications. A significant ongoing effort has been to develop rich classes of statistical models that can represent the data faithfully, and at the same time allow tractable learning, estimation and sampling. *Graphical models* [13, 35, 78, 83] constitute a powerful framework for statistical modeling that is based on exploiting the structure of conditional independence among the variables encoded by a sparse graph. Certain examples of graphical models have been in use for quite a long time, but recently the field has been gaining momentum and reaching to an ever increasing and diverse range of applications.

A graphical model represents how a complex joint probability distribution decomposes into products of simple local functions (or factors) that only depend on small subsets of variables. This decomposition is represented by a graph: a random variable is associated with each vertex, and the edges or cliques represent the local functions. An important fact that makes the framework of graphical models very powerful is that the graph captures the conditional independence structure among the random variables. It is the presence of this structure that enables the compact representation of rich classes of probability models and efficient algorithms for estimation and learning. The applications of graphical models range from computer vision [51,100,121], speech and language processing [14, 15, 114], communications and error control coding [29, 52, 55, 91], sensor networks [24, 66, 96], to biology and medicine [54, 84, 137], statistical physics [93, 104], and combinatorial optimization [94, 112]. The use of graphical models has led to revolutionary advances in many of these fields.

The graph in the model is often specified by the application: in a genomic application the variables may represent expression levels of certain genes and the edges may represent real biological interactions; in computer vision the nodes may correspond to pixels or patches of an image and the edges may represent the fact that nearby nodes are likely to be similar for natural images. The graph may also be constructed for the purpose of efficiency: e.g., tree-structured and multiscale models allow particularly efficient estimation and learning [33,135]. In this case nodes and edges may or may not have direct physical meaning. The use of graphical models involves a variety of tasks – from defining or learning the graph structure, optimizing model parameters given data, finding tractable approximations if the model is too complex, sampling configurations

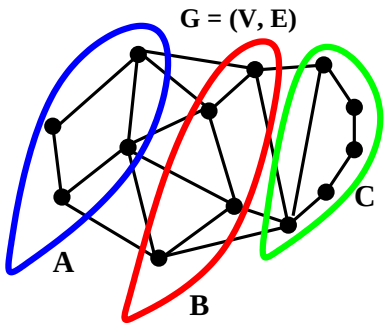


Figure 1.1. Markov property of a graphical model: graph separation implies conditional independence.

of the model, and finally doing inference – estimating the states of certain variables given possibly sparse and noisy observations. In this thesis we focus mostly on the last problem – doing inference when the model is already fully defined. This is an important task in and of itself, but in addition inference can also be an essential part of learning and sampling.

Graphical models encompass constructions on various types of graphs (directed and undirected, chain-graphs and factor-graphs), and in principle have no restrictions on the state-space of the random variables – the random variables can be discrete, continuous, and even non-parametric. Of course, with such freedom comes responsibility – the most general form of a graphical model is utterly intractable. Hence, only certain special cases of the general graphical models formalism have been able to make the transition from theory into application.

■ 1.1 Gaussian Graphical Models

In this thesis we focus on Gaussian graphical models, where the variables in the model are jointly Gaussian. Also, for the most part we restrict ourselves to *Markov random fields* (MRF), i.e. models defined on undirected graphs [110]. We will use acronyms Gaussian graphical model (GGM) and Gaussian Markov random field (GMRF) interchangeably. These models have been first used in the statistics literature under the name *covariance selection* models [43, 115]. A well-known special case of a GGM is a linear state-space model – it can be represented as a graphical model defined on a chain.

As for any jointly Gaussian random variables, it is possible to write the probability density in the conventional form:

$$p(x) = \frac{1}{\sqrt{(2\pi)^N \det(P)^{-1}}} \exp(-\frac{1}{2}(x-\mu)^T P^{-1}(x-\mu)) \tag{1.1}$$

where the mean is $\mu = \mathbb{E}[x]$, and the covariance matrix is $P = \mathbb{E}[xx^T]$. What makes the GGM special is a structure of conditional independence among certain sets of variables,

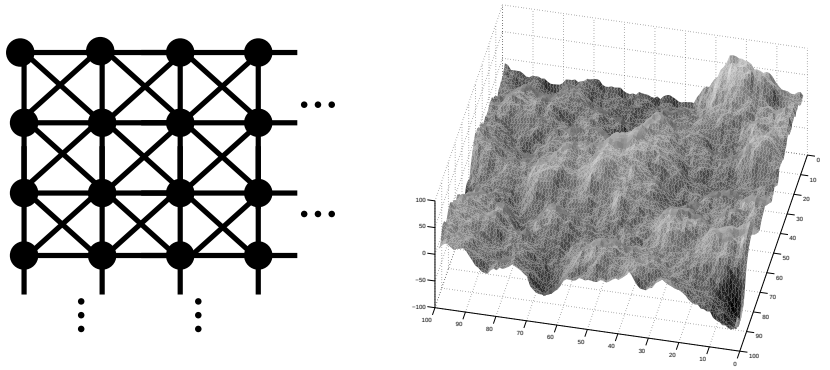


Figure 1.2. (a) A GMRF on a grid graph. (b) A sample from the GMRF.

also called the *Markov structure*, which is captured by a graph. Given a Markov graph of the model the conditional independence relationships can be immediately obtained. Consider Figure 1.1. Suppose that removing a set of nodes B separates the graph into two disconnected components, A and C. Then the variables x_A and x_C (corresponding to nodes in A and C) are conditionally independent given x_B. This generalizes the well-known property for Markov chains: the past is independent of the future given the current state (removing the current state separates the chain into two disconnected components).

For GGM the Markov structure can be seamlessly obtained from the inverse covariance matrix $J \triangleq P^{-1}$, also called the *information matrix*. In fact, the sparsity of J exactly matches the Markov graph of the model: if an edge $\{i, j\}$ is missing from the graph then $J_{ij} = 0$. This has the interpretation that x_i and x_j are conditionally independent given all the other variables in the model. Instead of (1.1) we will extensively use an alternative representation of a Gaussian probability density which reveals the Markov structure, which is parameterized by $J = P^{-1}$ and $h = J\mu$. This representation is called the *information form* of a Gaussian density with J and h being the *information parameters*:

$$p(x) \propto \exp(-\frac{1}{2}x^T Jx + h^T x) \tag{1.2}$$

GMRF models are used in a wide variety of fields – from geostatistics, sensor networks, and computer vision to genomics and epidemiological studies [27, 28, 36, 46, 96, 110]. In addition, many quadratic problems arising in machine learning and optimization may be represented as Gaussian graphical models, thus allowing to apply methods developed in the graphical modeling literature [10, 11, 128]. To give an example of how a GMRF might be used we briefly mention the problem of image interpolation from sparse noisy measurements. Suppose the random variables are the gray-levels at each

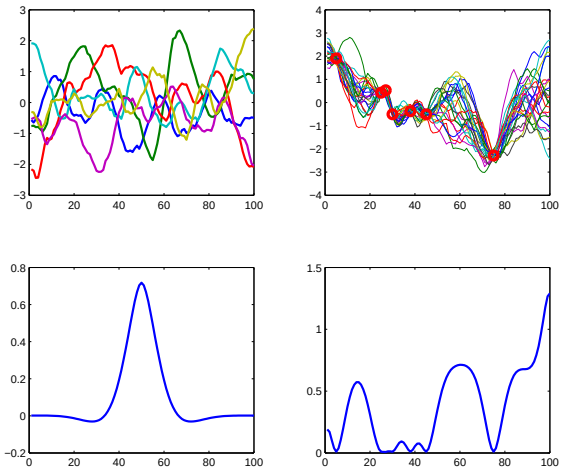


Figure 1.3. Samples from a chain GMRF, and the correlations between the center node and the other nodes.

of the pixels in the image, and we use the *thin-plate* model which captures smoothness properties of natural images (we discuss such models in more detail in Chapter 2). The Markov graph for the thin-plate model is a grid with connections up to two steps away, see Figure 1.2 (a). In Figure 1.2 (b) we show a random sample from this model, which looks like a plausible geological surface. A typical application in geostatistics would try to fit model parameters such that the model captures a class of surfaces of interest, and then given a sparse set of noisy observations interpolate the surface and provide error variances. For clarity we show a one-dimensional chain example in Figure 1.3. In the top left plot we show a number of samples from the prior, and in the top right plot we show several conditional samples given a few sparse noisy measurements (shown in circles). The bottom left plot displays the long correlation in the prior model (between the center node and the other nodes), and the bottom right plot shows that the posterior variances are smallest near the measurements.

As we discuss in more detail in Chapter 2 the prior model $p(x)$ specifies a sparse J matrix. Adding local measurements of the form $p(y|x) = \prod p(y_i|x_i)$, the posterior becomes $p(x|y) \propto p(y|x)p(x)$. The local nature of the measurements does not change the graph structure for the posterior – it only changes the diagonal of J and the h-vector.

■ 1.2 Inference in Gaussian Models

Given a GGM model in information form, we consider the problem of inference (or estimation) – i.e. determining the marginal densities of the variables given some ob-

servations. This requires computing marginal means and variances at each node. In principle, both means and variances can be obtained by inverting the information matrix: $P = J^{-1}$ and $\mu = Ph$. The complexity of matrix inversion is cubic in the number of variables, so it is appropriate for models of moderate size. More efficient recursive calculations are possible in graphs with very sparse structure—e.g., in chains, trees and in graphs with "thin" junction trees [83] (see Chapter 2). For these models, *belief propagation* (BP) or its junction tree variants [35,103] efficiently compute the marginals in time linear in the number of variables[1]. In large-scale models with more complex graphs, e.g. for models arising in oceanography, 3D-tomography, and seismology, even the junction tree approach becomes computationally prohibitive. Junction-tree versions of belief propagation reduce the complexity of exact inference from cubic in the number of variables to cubic in the "tree-width" of the graph [83]. For square and cubic lattice models with N nodes this leads to complexity $O(N^{3/2})$ and $O(N^2)$ respectively. Despite being a great improvement from brute-force matrix inversion, this is still not scalable for large models. In addition, junction-tree algorithms are quite involved to implement. A recent method, recursive cavity modeling (RCM) [76], provides tractable computation of approximate means and variances using a combination of junction-tree ideas with recursive model-thinning. This is a very appealing approach, but analytical guarantees of accuracy have not yet been established, and the implementation of the method is technically challenging. We also mention a recently developed Lagrangian relaxation (LR) method which decomposes loopy graphs into tractable subgraphs and uses Lagrange dual formulation to enforce consistency constraints among them [73,75]. LR can be applied to both Gaussian and discrete models, and for the Gaussian case it computes the exact means and provides upper bounds on the variances.

Iterative and multigrid methods from numerical linear algebra [126, 127] can be used to compute the marginal means in a sparse GMRF to any desired accuracy, but these methods do not provide the variances. In order to also efficiently compute variances in large-scale models, approximate methods have to be used. A wide variety of approximate inference methods exists which can be roughly divided into variational inference [95,131,136] and Monte Carlo sampling methods [57,108]. In the first part of the thesis we focus on an approach called *loopy belief propagation* (LBP) [103,111,133,138], which iteratively applies the same local updates as tree-structured belief propagation to graphs with loops. It falls within the realm of variational inference. We now briefly motivate LBP and tree-structured exact BP. Chapter 2 contains a more detailed presentation.

[1]For Gaussian models these methods correspond to direct methods for sparse matrix inversion.

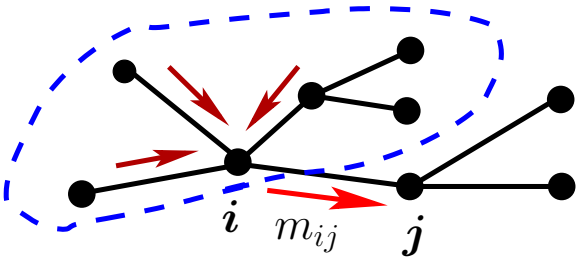


Figure 1.4. BP figure.

■ 1.3 Belief Propagation: Exact and Loopy

In tree-structured models belief propagation (or the *sum-product algorithm*) is an exact message-passing algorithm to compute the marginals[2]. It can be viewed as a form of variable elimination – integrating out the variables one by one until just the variable of interest remains. To naively compute all the marginals, simple variable elimination would have to be applied for each variable, producing a lot of redundant repeated computations. Belief propagation eliminates this redundancy by processing all variables together and storing the intermediate computations as messages. Consider Figure 1.4: a message m_{ij} from i to j captures the effect of eliminating the whole subtree that extends from i in the direction away from j – this message will be used to compute all the marginals to the right of node i. By passing these messages sequentially from the leaves to some designated root and back to the leaves, all marginals can be computed in $O(N)$ message updates. Message updates can also be done in parallel: all messages are first initialized to an uninformative value, and are repeatedly updated until they reach a fixed point. In tree-structured models parallel form of updates is also guaranteed to converge and provide the correct marginals after a fixed number of iterations.

Variable elimination corresponds to simple message updates only in tree-structured graphs. In presence of loops it modifies the graph by introducing new interactions (edges) among the neighbors of the eliminated variables. This can be resolved by merging variables together until the graph becomes a tree (form a junction tree), but, as we mentioned, for grids and denser graphs this quickly becomes computationally intractable.

Alternatively, one could ignore the loops in the graph and still carry out local BP message updates in parallel until they (hopefully) converge. This approach is called loopy belief propagation (LBP). LBP has been shown to often provide excellent approximate solutions for many hard problems, it is tractable (has a low cost per iteration) and allows distributed implementation, which is crucial in applications such as sensor networks [96]. However, in general it 'double counts' messages that travel multiple

[2] A version of belief propagation called *max-product* also addresses MAP estimation, but for Gaussian models the two algorithms are essentially the same.

times around loops which may in certain cases give very poor approximations, and it is not even guaranteed to converge [101].

There has been a significant effort to explain or predict the success of LBP for both discrete and Gaussian models: in graphs with long loops and weak pairwise interactions, errors due to loops will be small; the binary MRF max-product version of loopy belief propagation is shown to be locally optimal with respect to a large set of local changes [134] and for the weighted matching problem the performance of LBP has been related to that of linear programming relaxation [112]; in GMRFs it has been shown that upon convergence the means are correct [129, 133]; sufficient conditions for LBP convergence are given in [69, 99, 124]; and there is an interpretation of loopy belief propagation fixed points as being stationary points of the Bethe-free energy [138]. However despite this progress, the understanding of LBP convergence and accuracy is very limited, and further analysis is an ongoing research effort. Analysis of LBP using the *walk-sum* framework for Gaussian inference [74, 86] is the subject of Chapters 3 and 4 of this thesis. This analysis provides much new insight into the operation of Gaussian LBP, gives the tightest sufficient conditions for its convergence, and suggests when LBP may be a suitable algorithm for a particular application.

There are some scenarios where the use of LBP to compute the variances is less than ideal (e.g. for models with long-range correlations): either LBP fails to converge or converges excruciatingly slowly or gives very inaccurate approximations for the variances. In Chapter 5 we propose an efficient method for computing accurate approximate variances in very large scale Gaussian models based on low-rank approximations.

■ 1.4 Thesis Contributions

This thesis makes two main contributions: a graph-theoretic framework for interpreting Gaussian loopy belief propagation in terms of computing walk-sums and new results on LBP convergence and accuracy, and a low-rank approach to compute accurate approximate variances in large-scale GMRF models. We now introduce these two contributions in more detail.

■ 1.4.1 Walk-sum Analysis of Loopy Belief Propagation

We first describe an intuitive graphical framework for the analysis of inference in Gaussian models. It is based on the representation of the means, variances and correlations in terms of weights of certain sets of walks in the graph. This 'walk-sum' formulation of Gaussian inference originated from a course project in [72] and is based on the Neumann series (power-series) for the matrix inverse:

$$P = J^{-1} = (I - R)^{-1} = \sum_{k=0}^{\infty} R^k, \text{ if } \rho(R) < 1. \tag{1.3}$$

Suppose that J is the normalized (unit-diagonal) information matrix of a GMRF, then R is the sparse matrix of partial correlation coefficients which has zero-diagonal, but

the same off-diagonal sparsity structure as J. As we discuss in Chapter 3, taking k-th power of R corresponds to computing sums of weights of walks of length k. And we show that means, variances, and correlations are walk-sums (sums of weights of the walks) over certain infinite sets of walks.

This walk-sum formulation applies to a wide class of GMRFs for which the expansion in (1.3) converges (if the spectral radius satisfies $\rho(R) < 1$). However, we are interested in a stricter condition where the result of the summation is independent of its order – i.e. the sum over walks converges absolutely. We call models with this property *walk-summable*. We characterize the class of walk-summable models and show that it contains (and extends well beyond) some "easy" classes of models, including models on trees, attractive, non-frustrated, and diagonally dominant models. We also show that walk-summability is equivalent to the fundamental notion of pairwise-normalizability.

We use the walk-sum formulation to develop a new interpretation of BP in trees and of LBP in general. Based on this interpretation we are able to extend the previously known sufficient conditions for convergence of LBP to the class of walk-summable models. Our sufficient condition is tighter than that based on diagonal dominance in [133] as walk-summable models are a strict superset of the class of diagonally dominant models, and as far as we know is the tightest sufficient condition for convergence of Gaussian LBP[3].

We also give a new explanation, in terms of walk-sums, of why LBP converges to the correct means but not to the correct variances. The reason is that LBP captures all of the walks needed to compute the means but only computes a subset of the walks needed for the variances. This difference between means and variances comes up because of the mapping that assigns walks from the loopy graph to the so-called LBP computation tree: *non-backtracking walks* (see Chapter 3) in the loopy graph get mapped to walks that are not 'seen' by LBP variances in the computation tree.

In general, walk-summability is sufficient but not necessary for LBP convergence. Hence, we also provide a tighter (essentially necessary) condition for convergence of LBP *variances* based on a weaker form of walk-summability defined on the LBP computation tree. This provides deeper insight into why LBP can fail to converge—because the LBP computation tree is not always well-posed.

In addition to scalar walk-summability we also consider the notions of vector and factor-graph walk-summability, and vector and factor-graph normalizability. Any Gaussian model can be perfectly represented as a scalar pairwise MRF, but using LBP on equivalent scalar and factor-graph models gives very different approximations. Using factor-graph models with larger factors provides the flexibility of being able to trade off complexity versus accuracy of approximation. While many of our scalar results do

[3]In related work [97] (concurrent with [86]) the authors make use of our walk-sum analysis of LBP, assuming pairwise-normalizability, to consider other initializations of the algorithm. Here, we choose one particular initialization of LBP. However, fixing this initialization does not in any way restrict the class of models or applications for which our results apply. For instance, the application considered by [96] can also be handled in our framework by a simple reparameterization. However, the critical condition is still walk-summability, which is presented in [86].

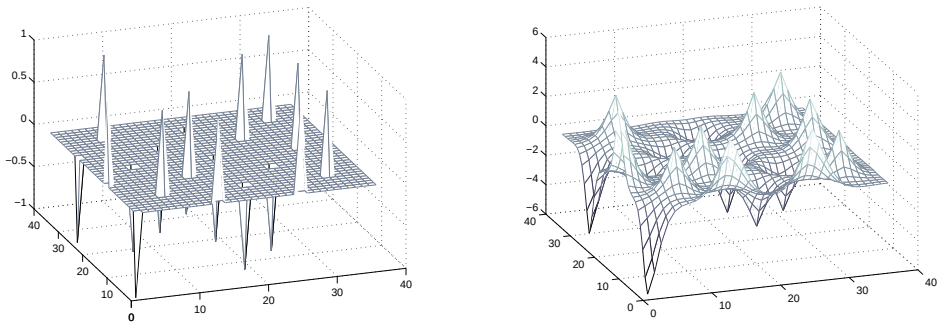


Figure 1.5. Aliasing of the covariance matrix in a 2D GMRF model. For large-scale GMRF models this allows tractable computation of approximate variances.

carry over to these more general conditions, some of the results become more involved and many interesting open questions remain.

The intuitive interpretation of correlations as walk-sums for Gaussian models begs the question of whether related walk-sum interpretation exists for other graphical models. While the power-series origin of the walk-sum expansion (1.3) is limited to Gaussian models, related expansions (over paths, self-avoiding walks, loops, or subgraphs) have been developed for other types of models [23,30,49,77,122], and exploring possible connections to Gaussian walk-sums is an exciting direction for further work. In addition, Gaussian walk-sums have potentials to develop new algorithms which go beyond LBP and capture more of the variance-walks in loopy graphs. We suggest these and other directions for further research in Chapter 6 of the thesis.

■ 1.4.2 Variance Approximation

Error variances are a crucial component of estimation, providing the reliability information for the means. They are also useful in other respects: regions of the field where residuals exceed error variances may be used to detect and correct model-mismatch (for example when smoothness models are applied to fields that contain abrupt edges). Also, as inference is an essential component of learning a model (for both parameter and structure estimation), accurate variance computation is needed when designing and fitting models to data. Another use of variances is to assist in selecting the location of new measurements to maximally reduce uncertainty.

We have already discussed the difficulties of computing the variances in large-scale models: unless the model is 'thin', exact computations are intractable. The method of loopy belief propagation can be a viable solution for certain classes of models, but in models with long-range correlations it either does not converge at all, or gives poor approximations.

We propose a simple framework for variance approximations that provides theo-

retical guarantees of accuracy. In our approach we use a low-rank aliasing matrix to compute an approximation to the inverse $J^{-1} = P$. By designing this matrix such that only the weakly correlated terms are aliased (see Figure 1.5), we are able to give provably accurate variance approximations. We propose a few different constructions for the low-rank matrix. We start with a design for single-scale models with short correlation length, and then extend it to single-scale models with long correlation length using a wavelet-based aliasing matrix construction. GMRFs with long correlation lengths, e.g. fractional Gaussian noise, are often better modeled using multiple scales. Thus we also extend our wavelet based construction to multi-scale models, in essence making both the modeling and the processing multi-scale.

■ 1.5 Thesis Outline

We start by providing a more detailed introduction to graphical models and GMRF models in Chapter 2. We discuss directed and undirected models and factor-graph formulations, and provide a detailed discussion of LBP and the computation-tree interpretation of LBP. In Chapter 3 we describe the walk-sum framework for Gaussian inference, and use it to analyze the LBP algorithm for Gaussian models, providing the best known sufficient conditions for its convergence. In Chapter 4 we generalize the walk-sum framework to vector and factor-graph models, and extend some of the results from the scalar ones. We also outline certain combinatorial ideas for computing walk-sums. In Chapter 5 we move on to describe the low-rank approach to compute approximate variances in large-scale GMRF models. We first describe the time-domain short-correlation approach, and then describe the wavelet-based long-range correlation version. In Chapter 6 we discuss open problems and suggestions for further work.

Bibliographic notes Parts of this thesis are based on our publications [74, 85–88] reflecting research done in collaboration with J. Johnson.

Chapter 2

Background

In this chapter we give a brief self-contained introduction to graphical models, including factorizations of probability distributions, their representations by graphs, and the Markov (conditional independence) properties. We start with general graphical models in Section 2.1, and then specialize to the Gaussian case in Section 2.3. We outline approaches to inference in graphical models, both exact and approximate. We summarize exact belief propagation on trees and the junction tree algorithm in Section 2.2, and the approximate loopy belief propagation on general graphs in Section 2.2.2.

■ 2.1 Preliminaries: Graphical Models

In this Section we formalize the concept of a graphical model, describe several types of graphical model such as MRFs, factor graphs and Bayesian networks, their graphical representation, and the implied conditional independence properties. First we briefly review some basic notions from graph theory [8, 12, 16], mainly to fix notation.

■ 2.1.1 Graph Theory

A *graph* $G = (V, \mathcal{E})$ is specified as a collection of vertices (or nodes) V together with a collection of edges $\mathcal{E} \subset V \times V$, i.e. $\mathcal{E}$ is a subset of all pairs of vertices. In this thesis we mostly deal with simple undirected graphs, which have no self-loops, and at most one edge between any pair of vertices. For undirected edges we use the set notation $\{i, j\}$ as the ordering of the two vertices does not matter. Unless we state otherwise, we will assume by default that all edges are undirected. In case we need to refer to directed edges we use the ordered pair notation (i, j).

The *neighborhood* of a vertex i in a graph is the set $\mathcal{N}(i) = \{j \in V \mid \{i, j\} \in \mathcal{E}\}$. The *degree* of a vertex i is its number of neighbors $|\mathcal{N}(i)|$. A graph is called *k-regular* if the degree of every vertex is k. A *subgraph* of G is a graph $G_s = (V_s, \mathcal{E}_s)$, where $V_s \subset V$, and $\mathcal{E}_s \subset V_s \times V_s$. We also say that G is a *supergraph* of G_s and that G_s is *embedded* in G. A *clique* of G is a fully connected subgraph of G, i.e. $\mathcal{C} = (V_s, \mathcal{E}_s)$ with every pair of vertices connected: $i, j \in V_s \Rightarrow \{i, j\} \in \mathcal{E}_s$. A clique is *maximal* if it is not contained within another clique. A *walk* w in a graph G is a sequence of vertices $w = (w_0, w_1, ..., w_l)$, $w_i \in V$, where each pair of consequent vertices is connected by an edge, $\{w_i, w_{i+1}\} \in \mathcal{E}$. The length of the walk is the number of edges that it traverses;

the walk w in our definition has length l. A *path* is a walk where all the edges and all the vertices are distinct. A graph is called *connected* if there is a path between any two vertices. The diameter of a graph $\text{diam}(G)$ is the maximum distance between any pair of vertices, where distance is defined as the length of the shortest path between the pair of vertices.

A *chain* is a connected graph where two of the vertices have one neighbor each, and all other vertices have two neighbors. A *cycle* is a connected graph, where each vertex has exactly two neighbors. A *tree* is a connected graph which contains no cycles as subgraphs. A graph is called *chordal* if every cycle of the graph which has length 4 or more contains a chord (an edge between two non-adjacent vertices of the cycle). The *treewidth* of a graph G is the minimum over all chordal graphs containing G of the size of the largest clique in the chordal graph minus one. As we explain later, treewidth of a graph is a measure of complexity of exact inference for graphical models.

A *hypergraph* $\mathcal{H} = (V, \mathcal{F})$ is a generalization of an undirected graph which allows hyper-edges $F \in \mathcal{F}$ connecting arbitrary subsets of vertices, rather than just pairs of vertices. Here $\mathcal{F} \subset 2^V$ is a collection of hyper-edges, i.e. arbitrary subsets of V.

■ 2.1.2 Graphical Representations of Factorizations of Probability

Graphical models are multivariate statistical models defined with respect to a graph. The main premise is that the joint density of a collection of random variables can be expressed as a product of several factors, each depending only on a small subset of the variables. Such factorization induces a structure of conditional independence among the variables. The graph encodes the structure of these local factors, and, importantly, it gives a very convenient representation of the conditional independence properties, thus enabling efficient algorithms which have made graphical models so popular.

There are various ways to use graphs to represent factorizations of a joint density into factors: *Bayesian networks* are based on directed graphs [35, 71], *Markov random fields* (MRF) are based on undirected graphs [9, 28, 110], and *factor graphs* [82] use hypergraphs (encoded as bipartite graphs with variable and factor nodes). We now describe these graphical representation of factorization, and how they relate to conditional independence properties of the graphical model. We focus on factor graph and MRF representation first, and later comment on their relation to the directed (Bayesian network) representation.

Suppose that we have a vector of random variables, $x = (x_1, x_2, \ ..., \ x_N)$ with discrete or continuous state space $x_i \in \mathcal{X}$. Suppose further that their joint density $p(x)$ can be expressed as a product of several positive functions (also called factors or potentials[1]) $\psi_F \geq 0$, indexed by subsets $F \subset \{1, .., N\}$ over some collection $F \in \mathcal{F}$. Each of the functions ψ_F only depends on the subset of random variables in F, i.e.

[1]To be precise, it is actually the negative logarithms of ψ_F that are usually referred to as potentials in the statistical mechanics literature. We abuse the terminology slightly for convenience.

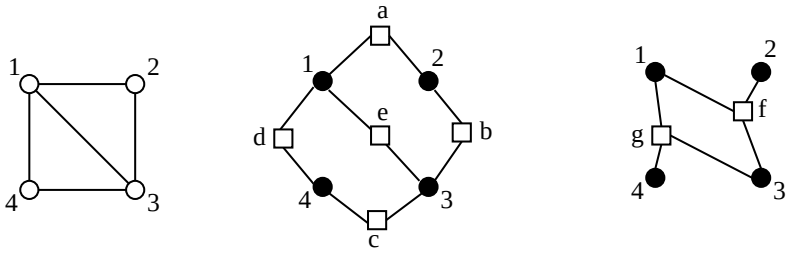


Figure 2.1. An undirected graph representation and two possible factor graphs corresponding to it. See Exaple 1 for an explanation.

$\psi_F = \psi_F(x_F)$, where we use x_F to denote the variables in F, i.e. $x_F = \{x_i,\ i \in F\}$:

$$p(x) = \frac{1}{Z} \prod_{F \in \mathcal{F}} \psi_F(x_F), \quad F \in \mathcal{F}. \tag{2.1}$$

Z is a normalizing constant, also called the *partition function*, which makes $p(x)$ integrate (or sum) to 1, $Z = \sum_x \prod_{F \in \mathcal{F}} \psi_F(x_F), \quad F \in \mathcal{F}$. Typically the factors ψ_F depend only on a small subset of variables, $|F| \ll |V|$, and a complicated probability distribution over many variables can be represented simply by specifying these local factors.

A factor graph summarizes the factorization structure of $p(x)$ by having two sets of vertices: variable-nodes $V_v = \{1, ..., N\}$ and factor-nodes $V_f = \{1, ..., |\mathcal{F}|\}$. The graph has an edge between a variable-node $i \in V_v$ and a factor-node $F \in V_f$ if $i \in F$, i.e. if ψ_F does depend on x_i. The factor graph has no other edges[2]. Two examples with 4 variables are displayed in Figure 2.1, middle and right plots, with circles representing variables and squares representing the factors.

Another way to encode the structure of $p(x)$ is using undirected graphs, $G = (V, \mathcal{E})$, which is referred to as the Markov random field (MRF) representation. Each vertex i corresponds to a random variable x_i, and an edge $\{i, j\}$ appears between nodes i and j if some factor ψ_F depends on both x_i and x_j. It is clear that each subset $F \in \mathcal{F}$ of nodes is a clique in G. Thus instead of using special factor-nodes, an MRF representation encodes factors by cliques. However, the representation is somewhat ambiguous as some cliques may not correspond to a single factor but to several smaller factors, which together cover the clique. Hence, the fine details of the factorization of $p(x)$ in (2.1) may not be exposed just from the undirected graph and only become apparent using a factor graph representation. We explain these ideas in Example 1 below. The undirected graph representation is however very convenient in providing the Markov (conditional independence) properties of the model.

The mapping from the conditional independence properties of an MRF model to the structure of the graph comes from the concept of graph separation. Suppose that the

[2]A factor graph is bipartite: the vertices are partitioned into two sets V_v and V_f, and every edge connects some vertex in V_v to a vertex in V_f.

set of nodes is partitioned into three disjoint sets $V = A \cup B \cup C$. Then B *separates* A from C if any path from a vertex in A to a vertex in C has to go through some vertex in B. A distribution $p(x)$ is called *Markov* with respect to G if for any such partition, x_A is independent of x_C given x_B, i.e. $p(x_A, x_C|x_B) = p(x_A|x_B)p(x_C|x_B)$. The connection between factorization and the Markov graph is formalized in the theorem of Hammersley and Clifford (see [21, 59, 83] for a proof):

Theorem 2.1.1 (Hammersley-Clifford Theorem). *If* $p(x) = \frac{1}{Z}\prod_{F\in\mathcal{F}} \psi_F(x_F)$ *with* $\psi_F(x_F) \geq 0$*, then* $p(x)$ *is Markov with respect to the corresponding graph* G*. Conversely, if* $p(x) > 0$ *for all* x*, and* $p(x)$ *is Markov with respect to* G*, then* $p(x)$ *can be expressed as a product of factors corresponding to cliques of* G*.*

Example 1 To illustrate the interplay of density factorization, Markov properties and factor graph and MRF representation, consider the undirected graph in Figure 2.1 on the left. The graph is a 4-node cycle with a chord. The absence of the edge $\{2, 4\}$ implies that for any distribution that is Markov with respect to G, x_2 and x_4 are independent given x_1 and x_3. However, x_2 and x_4 are not independent given x_1 alone, since there is a path $(2, 3, 4)$ which connects them, and does not go through x_1.

The graph has two maximal cliques of size 3: $\{1, 2, 3\}$, and $\{1, 3, 4\}$, and five cliques of size 2: one for each of the edges. By Hammersley-Clifford theorem, any distribution that is Markov over this graph is a product of factors over all the cliques. However, for a particular distribution some of these factors may be trivially equal to 1 (and can be ignored). Hence, there may be a few different factor graphs associated with this graph. Two possibilities are illustrated in Figure 2.1, center plot, with $p(x) = \psi_a(x_1, x_2)\psi_b(x_2, x_3)\psi_c(x_3, x_4)\psi_d(x_1, x_4)\psi_e(x_1, x_3)$, and right plot with $p(x) = \psi_f(x_1, x_2, x_3)\psi_g(x_1, x_3, x_4)$. The variable-nodes are denoted by circles, and the factor-nodes are denoted by squares. The example shows that the undirected representation is useful in obtaining the Markov properties, but a factor graph can serve as a more accurate (more restrictive) representation of the factorization. □

It is convenient to restrict attention to models with pairwise interactions – i.e. models where all the factors depend on at most two variables (i.e. all ψ_F satisfy $|F| \leq 2$, for example see Figure 2.1 middle plot). By merging some of the variables together (thereby increasing the state space) any MRF can be converted into a pairwise MRF. In the sequel, unless stated otherwise, we use pairwise MRFs. When dealing only with pairwise MRFs there is little benefit in using the factor graph representation, since it does nothing except adding factor-nodes in the middle of every edge. An MRF representation carries exactly the same information for the pairwise case. The factorization of a density for a pairwise MRF has the following form:

$$p(x) = \frac{1}{Z}\prod_{i\in V} \psi_i(x_i) \prod_{\{i,j\}\in\mathcal{E}} \psi_{i,j}(x_i, x_j) \tag{2.2}$$

where $\psi_i(x_i)$ are the self-potentials, and $\psi_{i,j}(x_i, x_j)$ are the pairwise or edge potentials.

Tree-structured MRF models A very important subclass of MRF models is based on tree-structured graphs which have no loops (we include chains and forests, i.e. collection of disjoint trees, into this category). Many of the computational tasks including inference, learning, and sampling are extremely efficient on trees. Thus trees are both popular models themselves, and also are used in various ways as approximations or as embedded structures to ease the computational burden for models defined on more general graphs [33, 120, 129, 130, 135].

In general MRFs the potentials need not have any connection to edge or clique marginals. However for trees a specification of potentials is possible which correspond to probabilities:

$$p(x) = \prod_{i \in V} p_i(x_i) \prod_{\{i,j\} \in \mathcal{E}} \frac{p_{ij}(x_i, x_j)}{p_i(x_i) p_j(x_j)} \tag{2.3}$$

This corresponds to a pairwise MRF in (2.2) with $\psi_i(x_i) = p_i(x_i)$, and $\psi_{ij}(x_i, x_j) = \frac{p_{ij}(x_i,x_j)}{p_i(x_i)p_j(x_j)}$. Another representation is obtained by picking a designated root, and an ordering of the variables (based on distance from the root), such that a parent-child relationship can be established between any pair of vertices connected by an edge. The following factorization then holds[3]:

$$p(x) = p_1(x_1) \prod_{\{i,j\} \in \mathcal{E},\ i<j} p(x_i \mid x_j) \tag{2.4}$$

Here we arbitrarily pick node 1 to be the root, and the notation $i < j$ represents that j is a parent of i. A more general directed representation is the base for Bayesian networks.

Bayesian networks: models on directed graphs In this thesis we use MRF and factor graph models, which are closely related to another graphical representation of probability factorization based on directed acyclic graphs, called Bayesian networks [71]. These models are particularly useful when there are causal relationships among the variables.

Bayesian networks specify for each vertex j a (possibly empty) set of parents $\pi(j) = \{i \mid (i,j) \in \mathcal{E}\}$, i.e. vertices corresponding to tails of all the directed edges that point to j. The acyclic property forbids the existence of directed cycles, and hence there exists a partial order on the vertices. The joint density $p(x)$ factorizes into conditional probabilities of variables given their parents:

$$p(x) = \prod_i p(i \mid \pi(i)). \tag{2.5}$$

This is in contrast to MRFs, where the factors are arbitrary positive functions, and in general do not correspond to probabilities. The absence of directed cycles ensures that $p(x)$ is a valid probability consistent with the conditional probabilities $p(i \mid \pi(i))$.

[3]This is essentially the chain rule for probabilities, which uses the Markov properties of the graph to simplify the conditioning. The factorization in (2.3) can be obtained from it by simple algebra.

Another important distinction from MRFs is the absence of the normalization constant in (2.5), as the density $p(x)$ integrates to 1 as specified.

The Markov properties of Bayesian networks are related to a notion of D-separation [13,83], which is markedly different from graph separation for undirected graphs that we have described earlier. The classes of conditional independence properties that directed and undirected representations capture are not the same (there is an intersection, but in general neither class is contained in the other). However, at the cost of losing some structure it is easy to convert from directed graphs to undirected by interconnecting each set $\{i, \pi(i)\}$ into a clique, and replacing all directed edges with undirected ones [78].

Exponential families The formalism of graphical models applies to models with arbitrary state spaces, both discrete and continuous. However, in order to be amenable to computations – these models need to have a finite representation, and allow efficient numerical operations such as conditioning and marginalization. Predominantly graphical models are chosen from the *exponential family* [6], a family of parameterized probability densities, which has the following form:

$$p(x) = \frac{1}{Z(\theta)} \exp(\sum_k \theta_k f_k(x_{F_k})). \tag{2.6}$$

Each $f_k(x_{F_k})$ (for $k \in \{1, .., K\}$) is a *feature function* that depends only on the subset of variables x_{F_k}, $F_k \subset V$. The function f_k maps each possible state of x_{F_k} to a real value. To each feature f_k there is an associated weight θ_k, also called a canonical or exponential parameter. The model is parameterized by $\theta = (\theta_1, \ ..., \ \theta_K)$. $Z(\theta)$ normalizes the density, and the valid set of θ is such that $Z(\theta) < \infty$, i.e. the model is *normalizable*. Note that by using $\psi_{F_k}(x_{F_k}) = \exp(\theta_k f_k(x_{F_k}))$ we recover the probability factorization representation in (2.1) which shows how exponential families may be described in the language of graphical models.

The exponential family includes very many common parametric probability distributions, both continuous and discrete, including Gaussian, multinomial, Poisson, geometric, exponential, among many others. There is a rich theory describing the exponential family with connections to diverse fields ranging from convex analysis to information geometry [1, 2, 131]. Some of the appealing properties of the exponential family include the maximum-entropy interpretation, moment-matching conditions for maximizing the likelihood, and the fact that features $f_F(x_F)$ are sufficient statistics. We refer the interested reader to [6] for a thorough presentation of the exponential family.

Note that from (2.6), conditioning on one of the variables can be easily done in any exponential family model, and the resulting conditional distribution belongs to a lower-order family of the same form. However, in general this does not hold for marginalization, apart from two exceptions: discrete multinomial and Gaussian densities. As computing marginals is one of the key tasks in graphical models, it comes as no surprise that these two models are the most convenient for computations – at least

in principle computing marginals does not require approximations[4].

In this thesis we mostly use Gaussian graphical models (GGM), where the random variables are jointly Gaussian, and have a finite parameterization. We introduce Gaussian graphical models in Section 2.3. Note that both conditionals and marginals remain Gaussian, so GGM are very attractive computationally.

■ 2.1.3 Using Graphical Models

Applying graphical models to model natural phenomena and to make predictions involves a number of steps. First one needs to specify the structure of the graph – this may come either directly from an application (e.g. grid graphs for images in computer vision), from expert knowledge – Bayesian networks for expert systems [35], or this structure must be learned from the data – as in genetic regulatory networks [41, 46, 92, 132]. In addition, we may choose the model structure to balance how well it models the data versus the ease of computation that it provides. If computational cost is critical and the model contains many variables then we may be forced to restrict the class of structures to tree-structured [33], thin graphs [5, 117], or multi-scale approximations [31, 32, 135].

After deciding on the graphical structure of the model, one must learn the parameters of the model to best fit the observed data. When all the variables are observed the maximum likelihood (ML) estimates of the parameters can be obtained by various optimization methods, or by iterative updates such as iterative proportional fitting (IPF) [70] and generalized iterative scaling [38, 40]. In case of unobserved variables, the EM algorithm and its variants have to be used [44]. Alternatively, one may chose to work in the Bayesian setting, with the parameters themselves being treated as random variables, and assigning a prior for them.

Finally, once the model is fully specified then it can be used for inference – making predictions of certain variables in the model based on observations of some other ones, and to draw samples from the model. In this thesis we focus on the problem of inference – we assume the model has been already fully specified, both the graph structure and the parameters. Typically inference in the field of graphical models refers to computing marginal densities or to finding the MAP (max *a-posteriori*) assignment of a subset of variables given observations of another subset. Inference is an important task in and of itself, but it can also appear as an essential component of parameter learning: in the exponential family the gradient of the log-likelihood with respect to the parameters θ in (2.6) depends on the difference of the observed moments from the data and the moments under θ, $\mathbb{E}_\theta[f_k(x_{F_k})]$. Hence learning model parameters also involves inference.

[4] Other classes of MRF models with continuous state-spaces and non-Gaussian interactions are often used for MAP estimation (e.g. Laplacian priors in the context of edge-preserving image restoration), but finding exact marginals is intractable in such models (and requires various approximations). Graphical models with mixed state-spaces [83] are also common, and even graphical models with non-parametric density representation are also starting to be subjected to practical use [119]. Again, these models only allow approximate inference.

■ 2.2 Inference Problems in Graphical Models

Inference (or estimation) refers to making predictions about the state of unobserved random variables x given the values of some other random variables y in the model. In the field of graphical models inference has become synonymous with either finding the marginals $p(x_i \mid y) = \int p(x \mid y) dx_{V\setminus i}$, or with finding the MAP assignment, $\hat{x} = \arg\max_x p(x \mid y)$, (here '\' represents set difference, and we write $V\setminus i$ as a shorthand for $V\setminus\{i\}$, so $x_{V\setminus i}$ stands for all the variables except x_i).

In graphical models observations are often introduced by combining a prior model $p(x)$ for the hidden variables with observations y whose likelihood is given by $p(y|x)$. This gives the following posterior:

$$p(x|y) \propto p(x)p(y|x). \tag{2.7}$$

It is most convenient when the observations are *local*[5], i.e. that given the state of x_i, each y_i is independent of the other variables x_j and observations y_j for $j \neq i$. In this case, the likelihood of the observations can be factorized: $p(y|x) \propto \prod_{i\in V} p(y_i|x_i)$. If we now modify the self-potentials (factors depending only on one node) as follows: $\psi_i(x_i, y_i) = \psi(x_i)p(y_i|x_i)$, then the graph structure of the model does not change upon incorporating local observations. Now the posterior density for a pairwise MRF can be written as:

$$p(x|y) \propto p(x)p(y|x) = \prod_{i\in V} \psi_i(x_i, y_i) \prod_{\{i,j\}\in\mathcal{E}} \psi_{i,j}(x_i, x_j) \tag{2.8}$$

A notational simplification comes from the fact that once y is observed, it no longer varies, so we can redefine $\tilde{p}(x) \triangleq p(x|y)$, and compute the unconditional marginals or MAP estimates in the model $\tilde{p}(x)$. The self-potentials for this model can be defined as $\tilde{\psi}(x_i) = \psi(x_i, y_i)$, and their dependence on y_i does not need to be present in the notation. Hence the problems of computing conditional and unconditional marginals (or MAP estimates) are essentially equivalent, and for simplicity of notation we will use the latter from now on.

An MRF is specified by giving a list of potentials, e.g., ψ_i for $i \in V$ and ψ_{ij} $\{i, j\} \in \mathcal{E}$ in the pairwise case. The normalization constant Z is typically not available, and is only defined implicitly. To compute the marginal densities the knowledge of Z is not necessary – if one can obtain an unnormalized marginal (a function of just one variable) then its normalization constant can be found by one-dimensional integration. Likewise, Z is not needed to find the MAP estimate.

The complexity of brute-force inference increases rapidly with the number of variables in a graphical model. In the discrete case, to compute either the marginal density or the MAP estimate for a model with $|V|$ variables each having S states requires examining every one of the possible $S^{|V|}$ states (to either compute the sum or to find

[5]Non-local observations may induce 'fill' and produce a posterior which has a more dense Markov graph than the prior.

the maximum). Clearly, brute-force inference is infeasible for discrete graphical models with even a moderate number of variables. For the Gaussian case, exact inference involves computing an inverse of a $|V| \times |V|$ matrix, which scales as a cubic in $|V|$.

This seems trivial when compared with the exponential complexity in the discrete case, but for models involving lattices, or volumes, with the number of nodes exceeding millions, exact calculation also becomes intractable. Brute-force calculation is agnostic of the structure of the graph – it does not take advantage of the main asset of a graphical model. Next we discuss how graph-structure can be used for possibly dramatic reductions in computational complexity.

■ 2.2.1 Exact Inference: BP and JT

Suppose that $p(x)$ is given by an MRF with pairwise interactions and $|V| = N$ nodes. The marginal at node i can be computed as $p_i(x_i) = \sum_{x_j, j \neq i} p(x)$ (for continuous variables the summation is replaced by an integral). In this section we focus on computing the marginals, but by replacing the summation by maximization one obtains algorithms for MAP estimation[6]. Take $i = 1$, then we need to compute:

$$p_1(x_1) = \sum_{x_2,...,x_N} p(x) = \sum_{x_2} \sum_{x_3} ... \left[\sum_{x_N} p(x_1, ..., x_N) \right] = \sum_{x_2,..,N-1} p_{V \backslash N}(x_{V \backslash N}) \tag{2.9}$$

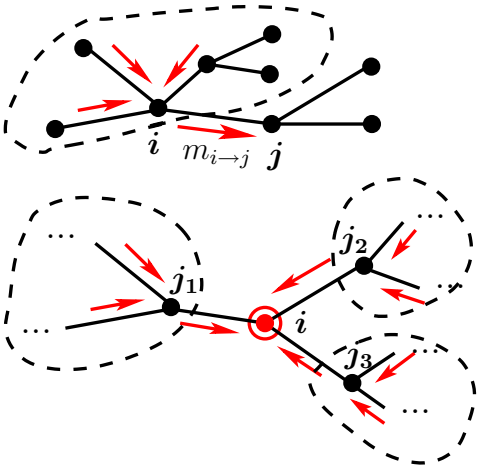
Recall that $V \backslash N$ represents all nodes except N. The action of summing over x_N (marginalizing out x_N) is equivalent to *variable elimination*. This reduces the problem of computing the marginal of $p(x)$ in an N-node graph to computing the marginal of $p_{V \backslash N}(x_{V \backslash N})$ in a $(N-1)$-node graph. Let us take a closer look at variable elimination for a pairwise MRF:

$$\sum_{x_N} p(x_1, ..., x_N) = \frac{1}{Z} \sum_{x_N} \prod_{i \in V} \psi_i(x_i) \prod_{\{i,j\} \in \mathcal{E}} \psi_{i,j}(x_i, x_j) = \tag{2.10}$$

$$\frac{1}{Z} \prod_{i \in V, i \neq N} \psi_i(x_i) \prod_{\{i,j\} \in \mathcal{E}, i,j \neq N} \psi_{i,j}(x_i, x_j) \sum_{x_N} \left[\psi_N(x_N) \prod_{i:\{i,N\} \in \mathcal{E}} \psi_{i,N}(x_i, x_N) \right]$$

The complexity of eliminating one variable depends on the number of neighbors that the variable has in G. Computing the sum $\sum_{x_N} \left[\psi_N(x_N) \prod_{i:\{i,N\} \in \mathcal{E}} \psi_{i,N}(x_i, x_N) \right]$ induces a new potential in the subgraph $V \backslash N$, which depends on all the neighbors $\mathcal{N}(N)$ of node N. This adds new edges to the graph, between each pair of neighbors of N. The same occurs for $N-1$, $N-2$, and so forth. However, for the case of singly-connected graphs (chains and trees), by eliminating variables one by one starting from the leaves, each eliminated node has exactly one neighbor, so no new edges are induced.

[6] Instead of marginals such algorithms compute *max-marginals* $M_i(x_i) = \max_{x_{V \backslash i}} p(x)$, which provide the MAP estimate.



A message $m_{i\to j}$ passed from node i to node $j \in \mathcal{N}(i)$ captures the effect of eliminating the subtree rooted at i.

Once all the messages are received at node i, the marginal can be computed as $p_i(x_i) = \psi_i(x_i) \prod_{j\in\mathcal{N}(i)} m_{j\to i}$. This can be seen as fusing the information from each subtree of i with the local information $\psi_i(x_i)$.

Figure 2.2. An illustration of BP message-passing on trees.

This makes the computation extremely efficient: for discrete models with S states at every node, variable elimination requires N calculations of complexity S^2 each, whereas brute force calculation involves S^N terms. Well-known examples of algorithms defined on chains which take advantage of this structure include the Kalman filter [80], and the forward-backward algorithm for hidden Markov models [106]. We now present this computation as sequential message-passing which will allow us to seamlessly introduce BP on trees.

Suppose x_N is a leaf-node which is connected to x_{N-1}. The newly induced potential in (2.10) is

$$m_{N\to N-1}(x_{N-1}) \triangleq \sum_{x_N} \psi_N(x_N)\psi_{N-1,N}(x_{N-1}, x_N) \tag{2.11}$$

This can be viewed as a message that the variable x_N sends to x_{N-1} reflecting its belief about the state of x_{N-1}. Now the self-potential of x_{N-1} in $p_{V\setminus N}(x_{V\setminus N})$ gets modified to $\psi_{N-1}(x_{N-1})m_{N\to N-1}(x_{N-1})$. Suppose the time comes to eliminate a variable i, a leaf in the current reduced graph, which has already had all its neighbors eliminated except neighbor j. The self-potential for node i has already been modified to $\psi_i(x_i) \prod_{k\in\mathcal{N}(i)\setminus j} m_{k\to i}(x_i)$. Now when we eliminate the variable x_i, we pass a message from i to j as follows:

$$m_{i\to j}(x_j) \triangleq \sum_{x_i} \psi_{i,j}(x_i, x_j)\psi_i(x_i) \prod_{k\in\mathcal{N}(i)\setminus j} m_{k\to i}(x_i) \tag{2.12}$$

The message $\mu_{i\to j}$ passed from node i to node $j \in N(i)$ captures the effect of eliminating the whole subtree rooted at i which extends in the direction opposite of j, see Figure 2.2, top plot. Variable elimination terminates once only the desired node

remains (see Figure 2.2, bottom plot), at which point we can obtain the marginals:

$$p_i(x_i) = \psi_i(x_i) \prod_{k \in \mathcal{N}(i)} m_{k \to i}(x_i). \tag{2.13}$$

Equations (2.12) and (2.13) summarize the steps of sequential variable elimination to obtain the marginal at one node. However, if we are interested in the marginals at all the nodes, then blindly applying this sequential variable elimination procedure for each node separately repeats many of the computations thus being very redundant.

BP on trees Belief propagation (BP) on trees is a message-passing algorithm that computes the marginals at all the nodes in the tree simultaneously. It can be interpreted as a sequential or iterative solution of the fixed point equations in (2.12).

The sequential version of BP on trees is equivalent to an efficient implementation of variable elimination done for all the nodes in parallel, but avoiding the redundant computations. BP does this by storing the results of these intermediate computations (the messages). Consider Figure 2.2, top plot. The message $m_{i \to j}$ is needed to compute the marginals at all the nodes to the left of j. Instead of computing it for each such node separately, we can compute it once and store it. A message is passed from i to j once all the messages from other neighbors of i, $k \in \mathcal{N}(i) \backslash j$ have been received. BP starts from the leaves, passes messages towards some designated root, and back to the leaves, thus computing all the messages (two for each edge – one for each direction). It is easy to check that all the necessary messages are computed after $2|\mathcal{E}|$ steps, and all the marginals can then be computed by a local operation at each node.

Summary: pairwise MRF BP on trees

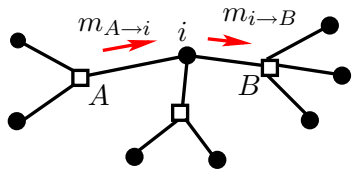
1. **(Message update)** Pass message $m_{i \to j}$ from i to j once i receives messages from all of its other neighbors, $k \in N(i) \backslash j$:

$$m_{i \to j}(x_j) \triangleq \sum_{x_i} \psi_i(x_i) \psi_{i,j}(x_i, x_j) \prod_{k \in \mathcal{N}(i) \backslash j} m_{k \to i}(x_i) \tag{2.14}$$

2. **(Compute marginals)** For any node i that has received all the messages compute the marginals:

$$p_i(x_i) = \psi_i(x_i) \prod_{k \in \mathcal{N}(i)} m_{k \to i}(x_i) \tag{2.15}$$

In addition to the sequential version of the algorithm, it is also possible to use an iterative version. Instead of viewing BP message updates (2.14) as a sequence of steps needed to compute a marginal, we can view them as a set of fixed point equations (one for each message) that we would like to satisfy. To solve them we arbitrarily initialize the messages (e.g. to 1) and iteratively apply the message updates in parallel, or according to some other message schedule, until convergence. In tree-structured graphs



Two types of messages in factor graph BP: factor-to-variable $m_{A\to i}$ and variable-to-factor $m_{i\to B}$. Message $m_{A\to i}$ captures the effect of eliminating the subtree rooted at i extending in the direction of A.

Figure 2.3. Illustration: factor graph version of BP.

it is easy to show that the parallel version of these updates converges to the correct answers (same as sequential) after diam(G) steps. The parallel version of BP is less efficient than the serial one, but it opens the door to a whole new world of approximate inference in graphs with loops via loopy belief propagation (LBP), which we describe in Section 2.2.2.

BP on tree-structured factor graphs A simple extension of the above algorithm can also be used to compute the marginals in a graphical model represented by a tree-structured factor graph. Suppose $p(x) \propto \prod_i \psi_i(x_i) \prod_F \psi_F(x_F)$, and the associated factor graph is tree-structured, see Figure 2.3 for an example. We explicitly separate the single-node factors ψ_i from higher-order factors ψ_F (here all $|F| > 1$) for convenience, and omit the single-node factors in the figure. Recall that a factor graph has two type of nodes: variable-nodes denoted by circles and factor-nodes denoted by squares. We use symbols i, j, k for variable-nodes and A, B for factor-nodes. For tree factor graphs we can do sequential variable elimination akin to the one in (2.10). We call a variable-node a leaf if it has only one neighboring higher-order factor-node. Sequential variable elimination can be done analogous to the pairwise case by repeatedly eliminating the leaves until the variable of interest remains.

For factor graph models it is convenient to view variable elimination as consisting of groups of two steps: eliminating a variable-node, and eliminating a factor-node. By eliminating a leaf variable-node i which is connected to A, we are fusing the factor ψ_A with the local self-potential ψ_i for i, and the incoming messages into i, $\prod m_{B\to i}(x_i)$.

By eliminating a factor-node A (when all but one of the variable-neighbors of A have already been eliminated) we compute the sum $\sum_{x_{A\setminus i}} \psi_A(x_A) \prod m_{k\to A}(x_k)$ thus reducing ψ_A into a function of a single variable x_i and fusing it with the self-potential ψ_i of i. The procedure starts by eliminating all leaf-variable-nodes, and then all leaf-factor-nodes repeatedly until just the variable of interest remains. Of course, to compute all the marginals efficiently we compute all them in parallel – analogous to the previous section for a pairwise MRF: first we compute all the messages, and then combine them to obtain the marginals. This is the factor graph version of BP.

Summary: BP on tree-structured factor graphs

1. **(Factor-variable message update)** Pass message $m_{A\to i}$ once factor A has re-

ceived messages from all of its other variable-neighbors, $j \in \mathcal{N}(A)\backslash i$:

$$m_{A\to i}(x_i) = \sum_{x_{A\backslash i}} \psi_A(x_A) \prod_{j\in\mathcal{N}(A)\backslash i} m_{j\to A}(x_j) \tag{2.16}$$

2. **(Variable-factor message update)** Pass message $m_{i\to A}$ once variable i has received messages from all of its other factor-neighbors, $B \in \mathcal{N}(i)\backslash A$:

$$m_{i\to A}(x_i) = \psi_i(x_i) \prod_{B\in\mathcal{N}(i)\backslash A} m_{B\to i}(x_i) \tag{2.17}$$

3. **(Compute the marginals)** For any variable-node i or for any $A \in \mathcal{F}$ compute the marginals:

$$p_i(x_i) = \psi_i(x_i) \prod_{A\in\mathcal{N}(i)} m_{A\to i}(x_i) \qquad p_A(x_A) = \psi_A(x_A) \prod_{i\in A} m_{i\to A}(x_i) \tag{2.18}$$

Exact inference via junction trees For graphs (or factor graphs) with loops, variable elimination necessarily induces new edges in the graph G. For each eliminated variable, all its remaining neighbors become fully connected (a clique is formed) and a new factor is induced which depends on all the variables in the clique. If we decide on an ordering of the variables, and eliminate variables one by one according to this ordering, then the complexity will depend on the size of the largest clique encountered (exponential in the discrete case, and cubic in the Gaussian case). The size of the largest such clique in general depends on the ordering of the nodes. The minimum over all possible elimination orders of the size of the largest clique, minus one, is called the treewidth of the graph. Treewidth serves as a lower bound on the complexity of exact inference. For tree-structured models, by always eliminating one of the remaining leaves, the largest clique is always of size 2 (and treewidth is 1). For more general graphs picking an order to minimize the treewidth is NP hard [3]; however, approximate procedures exist which pick a good suboptimal ordering [62].

To find all the marginals in parallel, there exists an efficient analogue of BP which operates on the *junction tree* of the graph – a tree whose nodes are the maximal cliques of the graph, connected in a way to satisfy certain consistency requirements. In order to construct the junction tree we first add all the edges that would have been introduced by variable elimination following some elimination order. This creates a chordal graph[7].

Next, a clique-tree is constructed which links maximal cliques sharing common nodes into a tree. The clique-tree must be chosen to satisfy a consistency requirement called the running intersection property: for any two cliques $\mathcal{C}_1$ and $\mathcal{C}_2$ in the clique-tree,

[7] See the definition in Section 2.1.1. A chordal graph has a *perfect elimination ordering* – a variable elimination following this ordering does not add new edges to the graph. Clearly, from our construction the graph is chordal, as we have added all the missing edges for some particular elimination order.

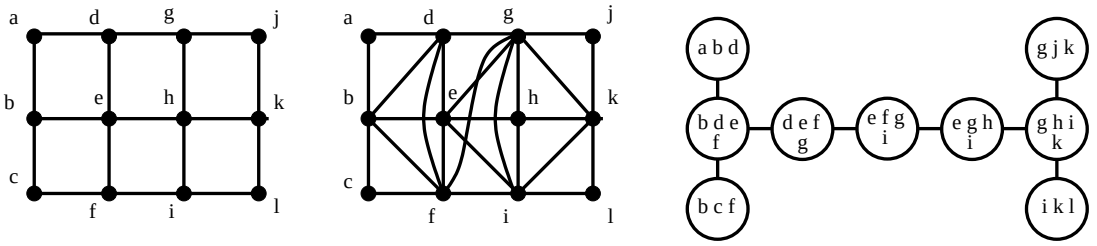


Figure 2.4. (left) A grid graph. (middle) A chordal supergraph corresponding to elimination order $(a, c, b, d, e, f, j, l, k, g, i)$. (right) A junction tree.

all the cliques $\mathcal{C}_i$ on the unique path connecting the two contain all the nodes in the intersection $\mathcal{C}_1 \cap \mathcal{C}_2 \in \mathcal{C}_i$. A clique-tree with the running-intersection property is called a junction tree. It is guaranteed to exist in chordal graphs. See an illustration in Figure 2.4.

Once a junction-tree is defined, a message passing generalization of BP which passes messages between cliques of the junction tree (corresponding to eliminating a subtree of the junction tree) can be defined [83,131]. This algorithm computes the exact marginals on all the cliques of the junction tree after passing all the messages. Note that the complexity of the junction tree algorithm is lower bounded by the treewidth of the graph (the bound is tight if an optimal elimination order is found). Hence, the algorithm is only attractive for thin graphs – chains, trees, and graphs with small treewidth.

For general discrete problems and large-scale Gaussian problems there are no tractable approaches for exact inference, and approximations have to be used. We focus here on a method that has received much attention recently, loopy belief propagation (LBP)[8].

■ 2.2.2 Loopy Belief Propagation

Loopy belief propagation was described in [103] as a heuristic approach to tractable approximate inference in loopy graphs. The BP update equations (2.14) and (2.15) derived for trees are purely local and only depend on the immediate neighborhood of a node. Hence, they are completely agnostic to the presence of cycles in the graph, and can also be applied to models defined on graphs with cycles, even though this no longer corresponds exactly to variable elimination in the graph.

Of course in this case, since there are cycles in the graph, only iterative message-scheduling forms can be defined. To be precise, a message schedule $\{\mathcal{M}^{(n)}\}$ specifies which messages $m_{i\to j}^{(n)}$, corresponding to directed[9] edges $(i, j) \in \mathcal{M}^{(n)}$, are updated at

[8] For alternative approaches to approximate inference in graphical models we refer the reader to [95, 131, 136] (variational methods) and [57, 108] (Monte Carlo sampling methods).

[9] For each undirected edge $\{i, j\} \in \mathcal{E}$ there are two messages: $m_{i\to j}$ for direction (i, j), and $m_{j\to i}$ for (j, i).

step n. The messages in $\mathcal{M}^{(n)}$ are updated using

$$m_{i\to j}^{(n)}(x_j) = \int \psi_{ij}(x_i, x_j)\psi_i(x_i) \prod_{k\in\mathcal{N}(i)\setminus j} m_{k\to i}^{(n-1)}(x_i) \; dx_i \tag{2.19}$$

and $m_{i\to j}^{(n)} = m_{i\to j}^{(n-1)}$ for the other messages. For example, in the fully-parallel case *all* messages are updated at each iteration whereas, in serial versions, only one message is updated at each iteration. We initialize LBP messages with non-informative values $m_{i\to j} = 1$. As is well-known, LBP may or may not converge, and if it does, in general, will not yield the correct values for the marginal distributions.

Theoretical understanding of LBP performance has been lagging well behind its success in practical applications. The ultimate goal for such analysis is being able to predict for a given problem whether LBP is a suitable approach – this involves developing necessary and sufficient conditions for its convergence, and understanding the quality of approximations.

For graphs which are tree-like, i.e. where the loops are long, one can argue that the effect of the loops should be negligible if there is sufficient mixing[10] (or decay of correlation for far away vertices). This approach is taken in the coding literature, where various tools have been developed to analyze the performance of iterative BP decoding for LDPC codes with large girth (long minimum cycles) [107, 125]. However LBP has been successfully applied even to models which contain many short loops [51, 53, 101]. The existing theoretical analysis of loopy BP does not explain this aspect of BP fully.

Much insight into the structure of LBP (and max-product) comes from considering its *computation tree* that captures the history of message updates since the first iteration. It has been used to develop sufficient conditions for LBP convergence in the discrete [124] and Gaussian case [133], to develop accuracy guarantees for max-product [134], and in the analysis of max-product for matching and independent-set problems [112, 113]. We give a detailed description of the computation tree in Section 2.2.3, as it will serve an important role in our analysis of LBP based on walk-sums in Chapter 3.

Another powerful interpretation of LBP comes from the statistical physics literature [138, 139]. We only give a gist of this connection here, and refer to [138] for details. The interpretation comes from the variational formulation of inference, where the correct marginals can be computed by minimizing the so-called *Gibbs free energy*, which is in general intractable for models on graphs with loops. By using simpler approximate free energies or constraining the feasible set to have tractable structure one can get lower bounds and approximations. Mean-field, Bethe and Kikuchi approximations and the recent tree-reweighted approaches all fall into this general variational framework [131]. Belief propagation has been shown to be related to Bethe free energy approximation: fixed points of LBP are stationary points of the Bethe free energy [63, 138, 139]. This is a powerful connection, but of course it depends on how close Bethe free energy approximates the correct free energy, which is not in general known.

[10] Also, [69] and [99] develop sufficient conditions for LBP convergence based on mixing which use bounds on the dynamic ranges of the pairwise potentials and node degrees, rather than girth.

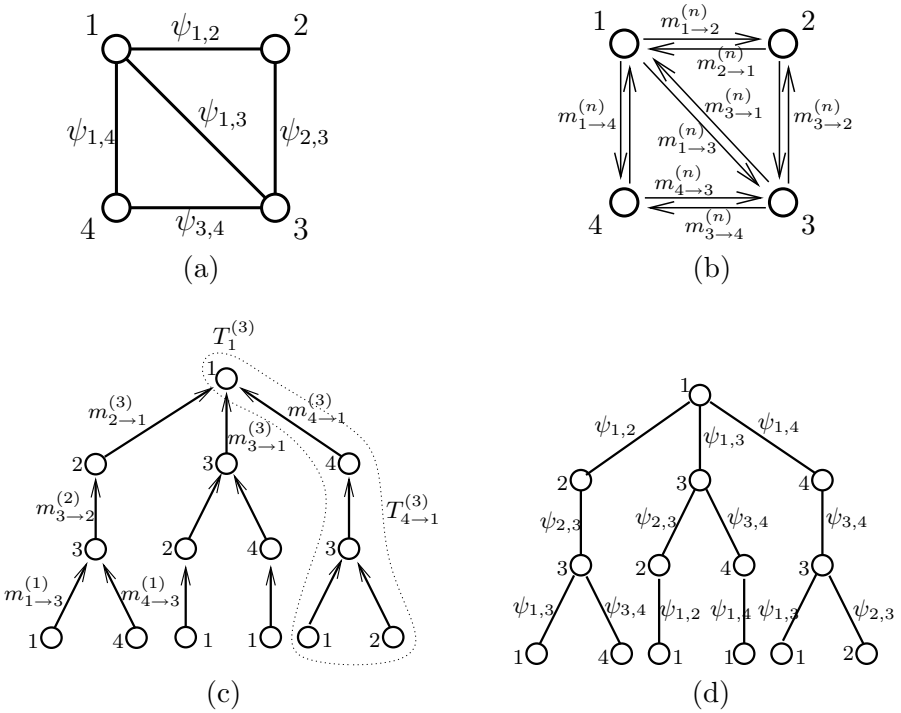


Figure 2.5. (a) Graph of a Gauss-Markov model with nodes $\{1, 2, 3, 4\}$ and with edge-potentials as shown. (b) The parallel LBP message passing scheme. In (c), we show how, after 3 iterations, messages link up to form the computation tree $T_1^{(3)}$ of node 1 (the subtree $T_{4\to1}^{(3)}$, associated with message $m_{4\to1}^{(3)}$, is also indicated within the dotted outline). In (d), we illustrate an equivalent Gauss-Markov tree model, with edge-potentials copied from (a), which has the same marginal at the root node as computed by LBP after 3 iterations.

For the particular case of Gaussian LBP it has been established that upon convergence the means are correct (although the variances are not), and sufficient conditions for convergence have been established [98, 111, 129, 133]. While clearly there has been considerable work on analyzing the convergence of LBP in general and for GMRFs in particular, the story is still far from being complete. A major contribution of Chapters 3 and 4 of this thesis is analysis that both provides new insights into LBP for Gaussian models and also brings the Gaussian story several steps closer to completion.

■ 2.2.3 Computation Tree Interpretation of LBP

A key component of our analysis is the insightful interpretation of LBP in terms of the computation tree [124, 133], which captures the structure of LBP computations. The basic idea is that a computation tree $T_i^{(n)}$ is constructed by "unwinding" the loopy graph intro a tree starting from node i. First node i is selected to be the root of the tree, and its neighbors as leaves in the tree – this creates $T_i^{(1)}$. Then for each leaf of the tree we add their neighbors from G as new nodes, except for their immediate parent in

the tree. This is repeated n times. Importantly, nodes and edges of the original graph may be replicated many times in the computation tree, but in a manner which preserves the local neighborhood structure. By assigning potential functions to the nodes and edges of $T_i^{(n)}$, copying these from the corresponding nodes and edges of the original loopy graphical model, we obtain a Markov tree model in which the marginal at the root node is precisely $p_i^{(n)}$ as computed by LBP. We illustrate the computation tree $T_1^{(3)}$ for a 4-cycle with a chord in Figure 2.5(d), and the graph itself in plot (a).

For our analysis in Chapter 3 we will need a more detailed description of how the computation trees grow with the LBP message updates. In addition to $T_i^{(n)}$, we also introduce $T_{i\to j}^{(n)}$ which summarizes the pedigree of message $m_{i\to j}^{(n)}$. Initially, the trees $T_i^{(n)}$ and $T_{i\to j}^{(n)}$ are just single nodes. When message $m_{i\to j}^{(n)}$ is computed, its computation tree $T_{i\to j}^{(n)}$ is constructed by joining the trees $T_{k\to i}^{(n-1)}$, for all neighbors k of i except j, at their common root node i and then adding an additional edge (i,j) to form $T_{i\to j}^{(n)}$ rooted at j. When marginal estimate $p_i^{(n)}$ is computed, its computation tree $T_i^{(n)}$ is formed by joining the trees $T_{k\to i}^{(n-1)}$, for all neighbors k of i, at their common root. An illustration appears in Figure 2.5.

In the case of the fully-parallel form of LBP, this leads to a collection of "balanced" computation trees $T_i^{(n)}$ (assuming there are no leaves in G) having uniform depth n, as the one in Figure 2.5. The same construction applies for other message schedules with the only difference being that the resulting computation trees may grow in a non-uniform manner. Our walk-sum analysis of LBP in Chapter 3, which relies on computation trees, applies for general message passing schedules.

■ 2.3 Gaussian Graphical Models

In this section we give a brief background on Gaussian graphical models and inference, describe a Gaussian version of both exact and loopy belief propagation (Section 2.3.1) and relate them to Gaussian elimination.

A Gaussian graphical model (GGM), which we also refer to as a Gaussian MRF (GMRF), is defined by an undirected graph $G = (V, \mathcal{E})$, and a collection of jointly Gaussian random variables $x = (x_i, i \in V)$. The probability density is given by[11]

$$p(x) \propto \exp\{-\tfrac{1}{2}x^T Jx + h^T x\} \tag{2.20}$$

where J is a symmetric, positive definite matrix ($J \succ 0$), which is sparse so as to respect the graph G: if $\{i,j\} \notin \mathcal{E}$ then $J_{ij} = 0$. The condition $J \succ 0$ is necessary so that (2.20) defines a *valid* (i.e., normalizable) probability density. This is the *information form* of the Gaussian density. We call J the *information matrix* and h the *potential vector*. They

[11]The constant of proportionality is $\exp\left(-\frac{1}{2}(|V|\log(2\pi) - \log|J| + h^T J^{-1}h)\right)$. Note that it does depend on the parameter h, while in the parameterization by P and μ the normalization constant is independent of μ: $p(x) = \frac{1}{\sqrt{(2\pi)^{|V|}\det(P)}}\exp(-\frac{1}{2}(x-\mu)^T P^{-1}(x-\mu))$.

are related to the standard Gaussian parameterization in terms of the mean $\mu \triangleq \mathbb{E}\{x\}$ and covariance $P \triangleq \mathbb{E}\{(x-\mu)(x-\mu)^T\}$ as follows:

$$\mu = J^{-1}h \quad \text{and} \quad P = J^{-1}$$

The class of densities in (2.20) is precisely the family of non-degenerate Gaussian distributions which are Markov with respect to the graph G [115]: since J is sparse, one can decompose $x^T Jx$ into terms that only depend on nodes and edges of G, e.g. $x^T Jx = \sum_i J_{ii}x_i^2 + 2\sum_{\{i,j\}\in\mathcal{E}} J_{ij}x_i x_j$, and appeal to the Hammersley-Clifford theorem, as we explain in more detail shortly.

The information parameterization reveals the conditional structure: if the joint density for x and y has information matrix $J = \left[\begin{smallmatrix} J_x & J_{x,y} \\ J_{y,x} & J_y \end{smallmatrix}\right]$, then the conditional density of x given $y = \tilde{y}$ has information matrix $J_{x|y} = J_x$, i.e. it is simply a submatrix of J, and $h_{x|y} = h_x - J_{x,y}\tilde{y}$. However, working with marginals in information parameterization requires more work: the marginal information matrix $\hat{J}_x$ of x is given by $\hat{J}_x = J_x - J_{x,y}J_y^{-1}J_{y,x}$ (a Schur complement computation [64] which follows from the matrix inversion lemma) and $\hat{h}_x = h_x - J_{x,y}J_y^{-1}h_y$. This is the opposite from the parameterization by P where the marginal covariance for x is a submatrix of P, while computing the conditionals requires Schur complements[12].

Recall the local Markov property: conditioned on $\mathcal{N}(i)$, the variable x_i is independent of the rest of the variables in the graph. The conditional variance of x_i given $x_{\mathcal{N}(i)}$ is given by the inverse of the i-th diagonal entry of J:

$$\text{var}(x_i \mid x_{\mathcal{N}(i)}) = (J_{i,i})^{-1}. \tag{2.21}$$

The *partial correlation coefficient* between variables x_i and x_j measures their conditional correlation given the values of the other variables $x_{V\setminus ij} \triangleq (x_k, k \in V \setminus \{i,j\})$. These are computed by normalizing the off-diagonal entries of the information matrix [83]:

$$r_{ij} \triangleq \frac{\text{cov}(x_i; x_j | x_{V\setminus ij})}{\sqrt{\text{var}(x_i|x_{V\setminus ij})\text{var}(x_j|x_{V\setminus ij})}} = -\frac{J_{ij}}{\sqrt{J_{ii}J_{jj}}}. \tag{2.22}$$

We observe that i and j are conditionally independent given $V\setminus\{i,j\}$ if $J_{ij} = 0$, illustrating the relation between the sparsity of J and conditional independence properties of the model. We now relate the information form of a Gaussian model in (2.20) with the pairwise factorization of $p(x)$. In agreement with the Hammersley-Clifford theorem[13] we can use a decomposition

$$p(x) \propto \prod_{i\in V} \psi_i(x_i) \prod_{\{i,j\}\in\mathcal{E}} \psi_{ij}(x_i, x_j)$$

[12]Compare: if $P = \left[\begin{smallmatrix} P_x & P_{x,y} \\ P_{y,x} & P_y \end{smallmatrix}\right]$, then $\hat{P}_x = P_x$, and $P_{x|y} = P_x - P_{x,y}P_y^{-1}P_{y,x}$. For the means, $\hat{\mu}_x = \mu_x$, and $\mu_{x|y} = \mu_x - P_{x,y}P_x^{-1}(\tilde{y} - \mu_y)$.

[13]Since a Gaussian model is inherently pairwise (the quadratic form in the exponent can be decomposed into pairwise interactions $\sum J_{ij}x_i x_j$), a stronger form of the Hammersley-Clifford holds that only requires pairwise factors rather than factors on maximal cliques.

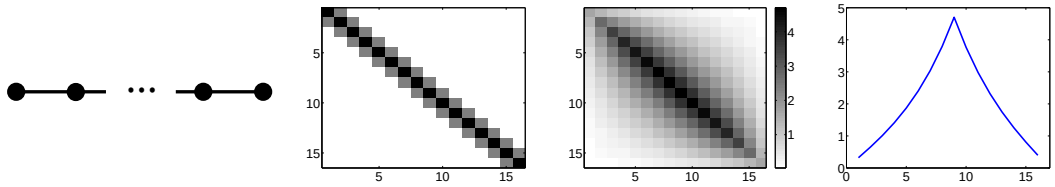


Figure 2.6. (a) A chain with $N = 16$ nodes. (b) The matrix J is tri-diagonal reflecting the chain structure of G. (c) The covariance matrix is *not* sparse – correlations decay away from the diagonal, but are never exactly zero. (d) A plot of a correlation of the central node with the other nodes (a column of P).

in terms of node and edge potential functions:

$$\psi_i(x_i) = \exp\{-\tfrac{1}{2}A_i x_i^2 + h_i x_i\} \quad \text{and} \quad \psi_{ij}(x_i, x_j) = \exp\{-\tfrac{1}{2}\left[\begin{smallmatrix} x_i & x_j \end{smallmatrix}\right] B_{ij} \left[\begin{smallmatrix} x_i \\ x_j \end{smallmatrix}\right]\} \quad (2.23)$$

Here, A_i and B_{ij} must add up to J such that:

$$x^T J x = \sum_i A_i x_i^2 + \sum_{\{i,j\}\in\mathcal{E}} \left(\begin{smallmatrix} x_i & x_j \end{smallmatrix}\right) B_{ij} \left(\begin{smallmatrix} x_i \\ x_j \end{smallmatrix}\right)$$

The choice of a decomposition of J into such A_i and B_{ij} is not unique: the diagonal elements J_{ii} can be split in various ways between A_i and B_{ij}, but the off-diagonal elements of J are copied directly into the corresponding B_{ij}. It is *not* always possible to find a decomposition of J such that both $A_i > 0$ and $B_{ij} \succ 0$.[14] We call models where such a decomposition exists *pairwise-normalizable*.

Our analysis in Chapter 3 is not limited to pairwise-normalizable models. Instead we use the decomposition $A_i = J_{ii}$ and $B_{ij} = \left[\begin{smallmatrix} 0 & J_{ij} \\ J_{ij} & 0 \end{smallmatrix}\right]$, which always exists, and leads to the following node and edge potentials:

$$\psi_i(x_i) = \exp\{-\tfrac{1}{2}J_{ii}x_i^2 + h_i x_i\} \quad \text{and} \quad \psi_{ij}(x_i, x_j) = \exp\{-x_i J_{ij} x_j\} \quad (2.24)$$

Note that *any* decomposition in (2.23) can easily be converted to the decomposition in (2.24). Now to build intuition we consider several examples of GMRFs.

Example GMRFs First consider a model defined on a simple chain, with $J_{ii} = 1$, $J_{i,j} = -\rho$ for $|i - j| = 1$ and $J_{i,j} = 0$ otherwise. We set $\rho = 0.49$. Figure 2.6(a) shows the chain, and plot (b) shows the corresponding sparse matrix J. The covariance $P = J^{-1}$ is not sparse: correlations decay away from the diagonal, but are never exactly

[14] For example the model with $J = \left[\begin{smallmatrix} 1 & 0.6 & 0.6 \\ 0.6 & 1 & 0.6 \\ 0.6 & 0.6 & 1 \end{smallmatrix}\right]$ is a valid model with $J \succ 0$, but no decomposition into single and pairwise positive definite factors exists. This can be verified by posing an appropriate semidefinite feasibility problem [20], or as we discuss in Chapter 3 through walk-summability.

zero, see plot (c) and (d). Such chain models arise in auto-regressive modeling[15], with $x_i = \alpha x_{i-1} + n_i$, and white Gaussian noise n_i.

Next, we illustrate the GMRF framework with a prototypical estimation problem. Consider the *thin-membrane prior*, commonly used for data interpolation:

$$p(x) \propto \exp\left(-\frac{\alpha}{2}\sum_{\{i,j\}\in\mathcal{E}}(x_i - x_j)^2\right). \tag{2.25}$$

This prior enforces leveled fields, i.e. it favors neighbors having similar values. As written in (2.25), this prior is degenerate (non-integrable), because any constant field x has the same probability. This degeneracy disappears once we condition on observations, or if small regularization $-\gamma\sum_i x_i^2$ is added in the exponent. The J matrix can be readily deduced from (2.25): $J_{ij} = 0$ for $i \neq j$ with $\{i,j\} \notin \mathcal{E}$, $J_{ij} = -\alpha$ for $\{i,j\} \in \mathcal{E}$, and $J_{ii} = \alpha d_i$. Here d_i is the degree of node i, $d_i = |N(i)|$. Another common prior in image-processing is the *thin-plate* prior:

$$p(x) \propto \exp\left(-\frac{\alpha}{2}\sum_{i\in V}(x_i - \frac{1}{d_i}\sum_{j\in N(i)} x_j)^2\right). \tag{2.26}$$

The thin-plate prior enforces that each node is close to the average of its neighbors[16], and penalizes curvature.

We can easily incorporate local observations y_i, with Gaussian $p(y_i|x_i)$. Assume that y_i is independent of x_j and other y_j for $j \neq i$: $p(y|x) = \prod_{i\in V} p(y_i|x_i)$. The posterior is now $p(x|y) \propto p(y|x)p(x)$, which is a GMRF that is Markov on the same graph (recall that we do not add new nodes for y's because they are observed and do not change). Hence, adding local observations only modifies the diagonal of J and the vector h.

For a concrete example, consider the linear Gaussian problem, with observations $y = Hx + n$, where x is zero-mean with covariance P, and independent noise n is zero-mean and with covariance Q. Then the Bayes least-squares estimate $\hat{x} = \mathbb{E}[x|y]$ and its error covariance $\hat{P} = \mathrm{cov}[x - \hat{x}]$ are given by:

$$\begin{aligned}(P^{-1} + H^T Q^{-1} H)\ \hat{x} &= H^T Q^{-1} y,\\ \hat{P} &= (P^{-1} + H^T Q^{-1} H)^{-1}.\end{aligned} \tag{2.27}$$

If $J_{prior} = P^{-1}$ is a sparse GMRF prior on x, and y are local conditionally independent observations, then $J = (P^{-1} + H^T Q^{-1} H)$ has the same sparsity as J_{prior}, with only the diagonal terms being modified. Now J and $h = H^T Q^{-1} y$ are the information parameters specifying the conditional model given the observations.

[15] A more general class of models used in the GMRF context are the conditionally auto-regressive models (CAR) [9] that specify that $Ax \propto \mathcal{N}(0, \sigma^2 I)$, where x are the random variables of interest, and A is some linear transformation.

[16] A thin plate model on a square grid has a more dense Markov graph: neighbors up to two steps away are connected by an edge.

Given a model in information form specified by (J, h), it is of interest to estimate the (conditional) means μ and the variances P_{ii} for all x_i. As we have discussed in Section 1.2, efficient direct matrix inversion methods [48,110] are possible for moderate-size examples, but for large-scale GMRFs exact computation of P and μ using matrix inversion is intractable. We now take a closer look at BP and LBP in the context of Gaussian models.

■ 2.3.1 Belief Propagation and Gaussian Elimination

We now consider BP in the context of Gaussian graphical models. We describe how BP message updates reduce to simple algebraic manipulations of the information parameters of the messages. After presenting the connection of BP on tree-structured models to Gaussian elimination, we make some remarks on loopy BP and its computation tree in this context.

Belief Propagation on Trees For GMRFs, as we have discussed, there are a variety of ways in which the information matrix can be decomposed into edge and node potential functions, and each such decomposition leads to BP iterations that are different in detail.[17] In our development we will use the simple decomposition in (2.24), directly in terms of the elements of J.

For Gaussian models in information form variable elimination/marginalization corresponds to *Gaussian elimination*. For example, if we wish to eliminate a single variable, i, from a GMRF to obtain the marginal over $U = V \backslash i$, the formulas yielding the information parameterization for the marginal on U are:

$$\hat{J}_U = J_{U,U} - J_{U,i} J_{ii}^{-1} J_{i,U} \quad \text{and} \quad \hat{h}_U = h_U - J_{U,i} J_{ii}^{-1} h_i$$

Here $\hat{J}_U$ and $\hat{h}_U$ specify the marginal density on x_U, whereas $J_{U,U}$ and h_U are a submatrix and a subvector of the information parameters on the full graph. The messages in Gaussian models can be parameterized in information form

$$m_{i \to j}(x_j) \triangleq \exp\{-\tfrac{1}{2} \Delta J_{i \to j} x_j^2 + \Delta h_{i \to j} x_j\}, \tag{2.28}$$

so that the fixed-point equations (2.19) can be stated in terms of these information parameters. We do this in two steps. The first step corresponds to preparing the message to be sent from node i to node j by collecting information from all of the other neighbors of i:

$$\hat{J}_{i \backslash j} = J_{ii} + \sum_{k \in \mathcal{N}(i) \backslash j} \Delta J_{k \to i} \quad \text{and} \quad \hat{h}_{i \backslash j} = h_i + \sum_{k \in \mathcal{N}(i) \backslash j} \Delta h_{k \to i} \tag{2.29}$$

The second step produces the information quantities to be propagated to node j:

$$\Delta J_{i \to j} = -J_{ji} \hat{J}_{i \backslash j}^{-1} J_{ij} \quad \text{and} \quad \Delta h_{i \to j} = -J_{ji} \hat{J}_{i \backslash j}^{-1} \hat{h}_{i \backslash j} \tag{2.30}$$

[17] One common decomposition for pairwise-normalizable models selects $A_i > 0$ and $B_{ij} \succ 0$ in (2.23) [96, 105, 133].

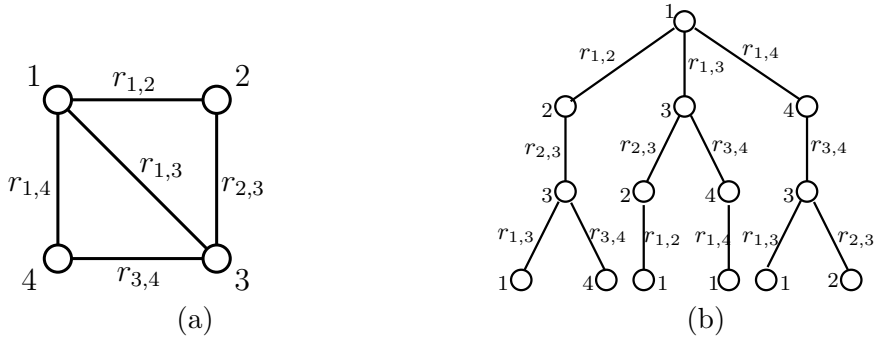


Figure 2.7. (a) Graph of a Gauss-Markov model with nodes $\{1, 2, 3, 4\}$ and with edge weights (partial correlations) as shown. (b) LBP computation tree for node 1, after 3 iterations.

As before, these equations can be solved by various message schedules, ranging from leaf-root-leaf Gaussian elimination and back-substitution to fully parallel iteration starting from the non-informative messages in which all $\Delta J_{i\to j}$ and $\Delta h_{i\to j}$ are set to zero. When the fixed point solution is obtained, the computation of the marginal at each node is obtained by combining messages and local information:

$$\hat{J}_i = J_{ii} + \sum_{k\in\mathcal{N}(i)} \Delta J_{k\to i} \quad \text{and} \quad \hat{h}_i = h_i + \sum_{k\in\mathcal{N}(i)} \Delta h_{k\to i} \tag{2.31}$$

which can be easily inverted to recover the marginal mean and variance:

$$\mu_i = \hat{J}_i^{-1}\hat{h}_i \quad \text{and} \quad P_{ii} = \hat{J}_i^{-1}$$

In general, performing Gaussian elimination corresponds, up to a permutation, to computing an LDL^T factorization of the information matrix – i.e., $QJQ^T = LDL^T$ where L is lower-triangular, D is diagonal and Q is a permutation matrix corresponding to a particular choice of elimination order. The factorization exists if J is non-singular. As we have discussed, in trees the elimination order can be chosen such that at each step of the procedure, the next node eliminated is a leaf node of the remaining subtree. Each node elimination step then corresponds to a message in the "upward" pass of the leaf-root-leaf form of Gaussian BP. In particular, $D_{ii} = \hat{J}_{i\setminus j}$ at all nodes i except the last (here, j is the parent of node i when i is eliminated) and $D_{ii} = \hat{J}_i$ for that last variable corresponding to the root of the tree. It is clear that $D_{ii} > 0$ for all i if and only if J is positive definite. We conclude that for models on trees, J being positive-definite is equivalent to all of the quantities $\hat{J}_{i\setminus j}$ and $\hat{J}_i$ in (2.29),(2.31) being positive, a condition we indicate by saying that BP on this tree is *well-posed*. Thus, performing Gaussian BP on trees serves as a simple test for validity of the model. The importance of this notion will become apparent in Chapter 3.

Loopy Belief Propagation and Gaussian models For GMRFs the application of LBP updates in (2.19) reduces to iterative application of equations (2.29) and (2.30). We

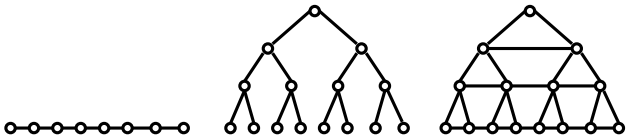


Figure 2.8. (left) Single-scale model. (center) Tree-structured multi-scale model. (right) Loopy multi-scale model on a pyramidal graph.

parameterize the messages $m_{i\to j}^{(n)}$ as in (2.28), and we denote the information parameters at step n by $\Delta J_{i\to j}^{(n)}$ and $\Delta h_{i\to j}^{(n)}$. We initialize LBP with non-informative zero values for all of the information parameters in these messages.

Gaussian LBP has received some attention in the literature. Sufficient conditions for its convergence in a turbo-decoding graph are given in [111] and for a multi-user detection problem in [98]. Sufficient conditions for LBP convergence for arbitrary graphs based on diagonal dominance of J are obtained in [133]. It is known [111, 129, 133] that if LBP converges, it yields the correct mean values but, in general, incorrect values for the variances. In Chapter 3 we use the walk-sum framework for Gaussian inference in conjunction with the computation tree construction to derive tighter results for Gaussian LBP.

As we have mentioned, BP on trees, which corresponds to performing Gaussian elimination, is well-posed if and only if J is positive-definite. LBP on Gaussian models corresponds to Gaussian elimination in the computation tree, which has its own information matrix composed by adding local terms for each node and edge in the tree. It corresponds to the unfolding illustrated in Figure 2.7 that involves replication of information parameters of the original loopy graphical model. Consequently, LBP is well-posed, yielding non-negative variances at each stage of the iteration, if and only if the model on the computation tree is *valid*, i.e., if and only if the information matrix for the computation tree is positive-definite. Very importantly, this is *not* always the case (even though the matrix J on the original graph is positive-definite). The analysis in Chapter 3 (in particular in Section 3.3) makes this point clear by considering situations in which LBP converges and when it fails to converge.

■ 2.3.2 Multi-scale GMRF Models

Single-scale models with only local interactions, such as thin membrane and thin plate models have limitations on the kind of fields that they represent. In particular, the tails of the correlation for such models fall-off exponentially fast. To represent long-range correlations with slower decay other models are needed. One can certainly accomplish this by using far denser single-scale graphs, with long-range interactions, but this defeats the sparsity needed for efficient algorithms. An alternative is to make use of multi-scale models which represent the phenomenon of interest at multiple scales or resolutions. Coarser scales correspond to local aggregates of finer scales: coarse-scale variables cap-

ture summaries of local regions at the finer scale. The multiple scales may represent physically meaningful quantities with measurements acquired at different scales. Alternatively, coarser scale may be artificially introduced hidden variables without measurements, which facilitate more efficient estimation. The scales may be disjoint, with estimates in coarser scales used to simplify estimation in the finer scales [56, 126], or they may be linked together into a coherent statistical model, with either deterministic or stochastic interactions between scales [19,31,32,135]. A significant effort has been devoted to the development of extremely efficient tree-structured (see Figure 2.8, center) multiscale models [135]. The main draw-back of tree-structured models is that certain neighbors in the fine-scale model may become quite distant in the tree-structured model, which leads to blocky artifacts in the estimates. To avoid these artifacts, multi-scale models which allow loops have also received attention, e.g. [19, 120].

In Chapter 5, Section 5.2.3, we consider a class of multi-scale models on pyramidal graphs with loops described in [31, 32]. The different scales in this model constitute a coherent statistical model with non-deterministic inter-scale interactions. The Markov graph for the model is illustrated in Figure 2.8 (right). In the picture we show each scale to be one-dimensional, but they can also be two- and three-dimensional. The model has a pyramidal structure including interactions within the scale, and between neighboring scales. The model has many short loops, so exact methods for tree-structured graphs do not apply, but the model is much richer representationally than tree-structured ones. The motivation for this multi-scale model is to represent or approximate a single-scale model with long correlation length. The correlations in the single-scale model get distributed among scales in the multi-scale model, and the long correlations are mostly accounted for through coarse-scale interactions. Conditioned on the coarse-scale variables the conditional correlations among the fine-scale variables are more local. In Section 5.2.3 we describe an extension of our low-rank variance approximation framework to find variances when such a model is specified.

Chapter 3

Walksum analysis of Gaussian Belief Propagation

In this chapter we present a new framework for analysis of inference in Gaussian graphical models based on walks in a graph. We decompose correlations between variables as a sum of weights over all walks between those variables in the graph, with the weight of each walk being given by the product of partial correlations on its edges. In Section 3.1 we set the stage by defining walk-sums, and characterizing the class of models where the decomposition holds – we call such models *walk-summable*. In Section 3.2 we provide an interpretation of Gaussian LBP in terms of computing certain walk-sums, and derive powerful sufficient conditions for LBP convergence. Finally in Section 3.3 we consider LBP outside the class of walk-summable models, and derive (almost) necessary and sufficient conditions for LBP convergence based on the validity (positive-definiteness) of the LBP computation tree.

■ 3.1 Walk-Summable Gaussian Models

We now describe our walk-sum framework for Gaussian inference. It is convenient to assume that we have normalized our model (by rescaling variables) so that $J_{ii} = 1$ for all i. Then, $J = I - R$ where R has zero diagonal and the off-diagonal elements are equal to the partial correlation coefficients r_{ij} in (2.22). Note that R inherits the sparsity from J: an off-diagonal element of R is non-zero only if there is a corresponding edge in G. We label each edge $\{i, j\}$ of the graph G with partial correlations r_{ij} as edge weights (e.g., see Figures 3.1 and 3.3). In this chapter we often refer to valid models – we call a model *valid* if its information matrix J is positive-definite, $J \succ 0$.

■ 3.1.1 Walk-Summability

Recall from Chapter 2 that a *walk* of length $l \geq 0$ in a graph G is a sequence $w = (w_0, w_1, \ldots, w_l)$ of nodes $w_k \in V$ such that each step of the walk (w_k, w_{k+1}) corresponds to an edge of the graph $\{w_k, w_{k+1}\} \in \mathcal{E}$. Walks may visit nodes and cross edges multiple times. We let $l(w)$ denote the length of walk w. We define the *weight* of a walk to be

the product of edge weights along the walk:

$$\phi(w) = \prod_{k=1}^{l(w)} r_{w_{k-1},w_k}$$

We also allow zero-length "self" walks $w = (v)$ at each node v for which we define $\phi(w) = 1$. To make a connection between these walks and Gaussian inference, we decompose the covariance matrix using the Neumann power series for the matrix inverse:[1]

$$P = J^{-1} = (I - R)^{-1} = \sum_{k=0}^{\infty} R^k, \quad \text{for} \quad \varrho(R) < 1$$

Here $\varrho(R)$ is the *spectral radius* of R, the maximum absolute value of eigenvalues of R. The power series converges if $\varrho(R) < 1$.[2] The (i,j)-th element of R^l can be expressed as a sum of weights of walks w that go from i to j and have length l (we denote this set of walks by $w : i \xrightarrow{l} j$):

$$(R^l)_{ij} = \sum_{w_1,\ldots,w_{l-1}} r_{i,w_1} r_{w_1,w_2} \ldots r_{w_{l-1},j} = \sum_{w: i \xrightarrow{l} j} \phi(w)$$

The last equality holds because only the terms that correspond to walks in the graph have non-zero contributions: for all other terms at least one of the partial correlation coefficients $r_{w_k,w_{k+1}}$ is zero. The set of walks from i to j of length l is finite, and the sum of weights of these walks (the walk-sum) is well-defined. We would also like to define walk-sums over arbitrary countable sets of walks. However, care must be taken, as walk-sums over countably many walks may or may not converge, and convergence may depend on the order of summation. This motivates the following definition:

We say that a Gaussian distribution is *walk-summable* (WS) if for all $i, j \in V$ the unordered sum over all walks w from i to j (denoted $w : i \to j$)

$$\sum_{w: i \to j} \phi(w)$$

is well-defined (i.e., converges to the same value for every possible summation order). Appealing to basic results of analysis [58, 109], the unordered sum is well-defined if and only if it *converges absolutely*, i.e., iff $\sum_{w: i \to j} |\phi(w)|$ converges.

Before we take a closer look at walk-summability, we introduce additional notation. For a matrix A, let $\bar{A}$ be the element-wise absolute value of A, i.e., $\bar{A}_{ij} = |A_{ij}|$. We

[1] The Neumann series holds for the unnormalized case as well: $J = D - K$, where D is the diagonal part of J. With the weight of a walk defined as $\phi(w) = \prod_{k=1}^{l(w)} K_{w_{k-1},w_k} / \prod_{k=0}^{l(w)} D_{w_k,w_k}$, all our analysis extends to the unnormalized case. We will say more about this in Section A.2.1 of Chapter 4.

[2] Note that $\varrho(R)$ can be greater than 1 while $I - R \succ 0$. This occurs if R has an eigenvalue less than -1. Such models are not walk-summable, so the analysis in Section 3.3 (rather than Section 3.2.2) applies.

use the notation $A \geq B$ for element-wise comparisons, and $A \succeq B$ for comparisons in positive-definite ordering. The following version of the Perron-Frobenius theorem [64, 127] for non-negative matrices (here $\bar{R} \geq 0$) is used on several occasions in our analysis:

Perron-Frobenius theorem There exists a non-negative eigenvector $x \geq 0$ of $\bar{R}$ with eigenvalue $\varrho(\bar{R})$. If the graph G is connected (where $r_{ij} \neq 0$ for all edges of G) then $\varrho(\bar{R})$ and x are strictly positive and, apart from γx with $\gamma > 0$, there are no other non-negative eigenvectors of $\bar{R}$.

In addition, we often use the following monotonicity properties of the spectral radius:

$$\text{(i)} \quad \varrho(R) \leq \varrho(\bar{R}) \qquad \text{(ii) If } \bar{R}_1 \leq \bar{R}_2 \text{ then } \varrho(\bar{R}_1) \leq \varrho(\bar{R}_2) \tag{3.1}$$

We now present several equivalent conditions for walk-summability:

Proposition 3.1.1 (Walk-Summability). *Each of the following conditions are equivalent to walk-summability:*

(i) $\sum_{w:i\to j} |\phi(w)|$ *converges for all* $i, j \in V$.

(ii) $\sum_l \bar{R}^l$ *converges.*

(iii) $\varrho(\bar{R}) < 1$.

(iv) $I - \bar{R} \succ 0$.

The proof appears in Appendix A.1. It uses absolute convergence to rearrange walks in order of increasing length, and the Perron-Frobenius theorem for part (iv). The condition $\varrho(\bar{R}) < 1$ is stronger than $\varrho(R) < 1$. The latter is sufficient for the convergence of the walks ordered by increasing length, whereas walk-summability enables convergence to the same answer in arbitrary order of summation. Note that (iv) implies that the model is walk-summable if and only if we can replace all negative partial correlation coefficients by their absolute values and still have a well-defined model (i.e., with information matrix $I - \bar{R} \succ 0$). We also note that condition (iv) relates walk-summability to the so-called H-matrices in linear algebra [65, 127].[3] As an immediate corollary, we identify the following important subclass of walk-summable models:

Corollary 3.1.1 (Attractive Models). *Let $J = I - R$ be a valid model ($J \succ 0$) with non-negative partial correlations $R \geq 0$. Then, $J = I - R$ is walk-summable.*

A superclass of attractive models is the set of *non-frustrated models*. A model is non-frustrated if it does not contain any frustrated cycles, i.e. cycles with an odd number of negative edge-weights. We show in Appendix A.1 (in the proof of Corollary 3.1.2)

[3] A (possibly non-symmetric) matrix A is an *H-matrix* if all eigenvalues of the matrix $M(A)$, where $M_{ii} = |A_{ii}|$, and $M_{ij} = -|A_{ij}|$ for $i \neq j$, have positive real parts. For symmetric matrices this is equivalent to M being positive definite. In (iv) J is an H-matrix since $M(J) = I - \bar{R} \succ 0$.

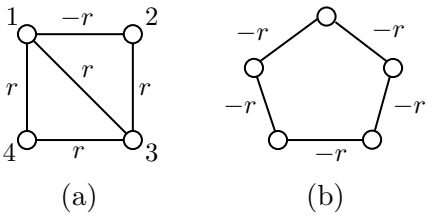


Figure 3.1. Example graphs: (a) 4-cycle with a chord. (b) 5-cycle.

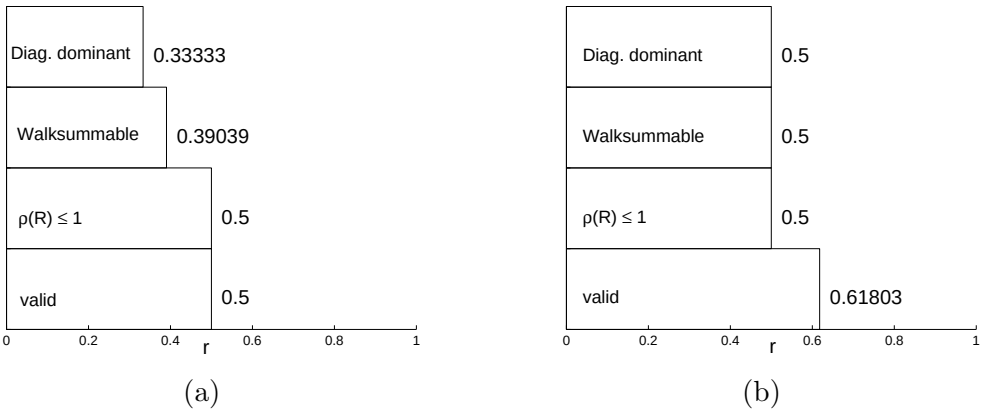


Figure 3.2. Critical regions for example models from Figure 3.1. (a) 4-cycle with a chord. (b) 5-cycle.

that if the model is non-frustrated, then one can negate some of the variables to make the model attractive[4]. Hence, we have another subclass of walk-summable models (the inclusion is strict as some frustrated models are walk-summable, see Example 1):

Corollary 3.1.2 (Non-frustrated models). *Let $J = I - R$ be valid. If R is non-frustrated then J is walk-summable.*

Example 1. In Figure 3.1 we illustrate two small Gaussian graphical models, which we use throughout this chapter. In both models the information matrix J is normalized to have unit diagonal and to have partial correlations as indicated in the figure. Consider the 4-cycle with a chord in Figure 3.1(a). The model is frustrated (due to the opposing sign of one of the partial correlations), and increasing r worsens the frustration. For $0 \leq r \leq 0.39039$, the model is valid and walk-summable: e.g., for $r = 0.39$, $\lambda_{\min}(J) = 0.22 > 0$, and $\varrho(\bar{R}) \approx 0.9990 < 1$. In the interval $0.39039 \leq r \leq 0.5$ the model is valid, but not walk-summable: e.g., for $r = 0.4$, $\lambda_{\min} = 0.2 > 0$, and $\varrho(\bar{R}) \approx 1.0246 > 1$. Also, note that for R (as opposed to $\bar{R}$), $\varrho(R) \leq 1$ for $r \leq 0.5$

[4]This result is referred to in [81]. However, our proof, in addition to proving that there exists such a sign similarity, also gives an algorithm which checks whether or not the model is frustrated, and determines which subset of variables to negate if the model is non-frustrated.

and $\varrho(R) > 1$ for $r > 0.5$. Finally, the model stops being diagonally dominant[5] above $r = \frac{1}{3}$, but walk-summability is a strictly larger set and extends until $r \approx 0.39039$. We summarize various critical points for this model and for the model in Figure 3.1(b) in the diagram in Figure 3.2.

Here are additional useful implications of walk-summability, with proof in Appendix A.1:

Proposition 3.1.2 (WS Necessary Conditions). *All of the following are implied by walk-summability:*

(i) $\varrho(R) < 1$.

(ii) $J = I - R \succ 0$.

(iii) $\sum_k R^k = (I - R)^{-1}$.

Implication (ii) shows that walk-summability is a sufficient condition for validity of the model. Also, (iii) shows the relevance of walk-sums for inference since $P = J^{-1} = (I - R)^{-1} = \sum_k R^k$ and $\mu = J^{-1}h = \sum_k R^k h$.

■ 3.1.2 Walk-Sums for Inference

Next we show that, in walk-summable models, means and variances correspond to walk-sums over certain sets of walks.

Proposition 3.1.3 (WS Inference). *If $J = I - R$ is walk-summable, then the covariance $P = J^{-1}$ is given by the walk-sums:*

$$P_{ij} = \sum_{w:i\to j} \phi(w)$$

Also, the means are walk-sums reweighted by the value of h at the start of each walk:

$$\mu_i = \sum_{w:*\to i} h_* \phi(w)$$

*where the sum is over **all** walks which end at node i (with arbitrary starting node), and where $*$ denotes the starting node of the walk w.*

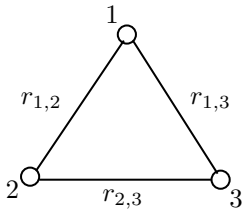
Proof. We use the fact that $(R^l)_{ij} = \sum_{w:i\overset{l}{\to}j} \phi(w)$. Then,

$$P_{ij} = \sum_l (R^l)_{ij} = \sum_l \sum_{w:i\overset{l}{\to}j} \phi(w) = \sum_{w:i\to j} \phi(w)$$

and

$$\mu_i = \sum_j h_j P_{ji} = \sum_j \sum_{w:j\overset{l}{\to}i} h_j \phi(w) = \sum_{w:*\to i} h_* \phi(w) \quad \square$$

[5]The definition of diagonal dominance appears before Proposition 3.1.9.



Single walk: $w = (1, 2, 3)$. Weight: $\phi(w) = r_{1,2}r_{2,3}$, $\phi_h(w) = h_1 r_{1,2}r_{2,3}$.

Self-return walks, $\mathcal{W}(1 \to 1)$: $\{(1), (1, 2, 1), (1, 3, 1), (1, 2, 3, 1), (1, 3, 2, 1), (1, 2, 1, 2, 1), ...\}$
$P_{1,1} = \phi(1 \to 1) = 1 + r_{1,2}r_{2,1} + r_{1,3}r_{3,1} + r_{1,2}r_{2,3}r_{3,1} + ...$

Set of walks $\mathcal{W}(* \to 1)$: $\{(1), (2, 1), (3, 1), (2, 3, 1), (3, 2, 1), (1, 2, 1)(1, 3, 1), ...\}$
$\mu_1 = \phi_h(* \to 1) = h_1 + h_2 r_{2,1} + h_3 r_{3,1} + h_2 r_{2,3}r_{3,1} + ...$

Figure 3.3. Illustration of walk-sums for means and variances.

Walk-Sum Notation We now provide a more compact notation for walk-sets and walk-sums. In general, given a set of walks $\mathcal{W}$ we define the walk-sum:

$$\phi(\mathcal{W}) = \sum_{w \in \mathcal{W}} \phi(w)$$

and the reweighted walk-sum:

$$\phi_h(\mathcal{W}) = \sum_{w \in \mathcal{W}} h_{w_0} \phi(w)$$

where w_0 denotes the initial node in the walk w. Also, we adopt the convention that $\mathcal{W}(\dots)$ denotes the set of all walks having some property $\dots$ and denote the associated walk-sums simply as $\phi(\dots)$ or $\phi_h(\dots)$. For instance, $\mathcal{W}(i \to j)$ denotes the set of all walks from i to j and $\phi(i \to j)$ is the corresponding walk-sum. Also, $\mathcal{W}(* \to i)$ denotes the set all walks that end at node i and $\phi_h(* \to i)$ is the corresponding reweighted walk-sum. In this notation, $P_{ij} = \phi(i \to j)$ and $\mu_i = \phi_h(* \to i)$. An illustration of walk-sums and their connection to inference appears in Figure 3.3 where we list some walks and walk-sums for a 3-cycle graph.

Walk-Sum Algebra We now show that the walk-sums required for inference in walk-summable models can be significantly simplified by exploiting the recursive structure of walks. To do so, we make use of some simple algebraic properties of walk-sums. The following lemmas all assume that the model is walk-summable.

Lemma 3.1.1. *Let $\mathcal{W} = \cup_{k=1}^{\infty} \mathcal{W}_k$ where the subsets $\mathcal{W}_k$ are disjoint. Then, $\phi(\mathcal{W}) = \sum_{k=1}^{\infty} \phi(\mathcal{W}_k)$.*

Proof. By the sum-partition theorem for absolutely convergent series [58]: $\sum_{w \in \mathcal{W}} \phi(w) = \sum_k \sum_{w \in \mathcal{W}_k} \phi(w)$. □

Lemma 3.1.2. *Let $\mathcal{W} = \cup_{k=1}^{\infty} \mathcal{W}_k$ where $\mathcal{W}_k \subset \mathcal{W}_{k+1}$ for all k. Then, $\phi(\mathcal{W}) = \lim_{k\to\infty} \phi(\mathcal{W}_k)$.*

Proof. Let $\mathcal{W}_0$ be the empty set. Then, $\mathcal{W} = \cup_{k=1}^{\infty}(\mathcal{W}_k \setminus \mathcal{W}_{k-1})$. By Lemma 3.1.1,

$$\phi(\mathcal{W}) = \sum_{k=1}^{\infty} \phi(\mathcal{W}_k \setminus \mathcal{W}_{k-1}) = \lim_{N\to\infty} \sum_{k=1}^{N} (\phi(\mathcal{W}_k) - \phi(\mathcal{W}_{k-1})) = \lim_{N\to\infty} (\phi(\mathcal{W}_N) - \phi(\mathcal{W}_0))$$

where we use $\phi(\mathcal{W}_0) = 0$ in the last step to obtain the result. □

Given two walks $u = (u_0, \ldots, u_n)$ and $v = (v_0, \ldots, v_m)$ with $u_n = v_0$ (walk v begins where walk u ends) we define the product of walks $uv = (u_0, \ldots, u_n, v_1, \ldots, v_m)$. Let $\mathcal{U}$ and $\mathcal{V}$ be two countable sets of walks such that every walk in $\mathcal{U}$ ends at a given node i and every walk in $\mathcal{V}$ begin at this node. Then we define the product set $\mathcal{UV} = \{uv \mid u \in \mathcal{U}, v \in \mathcal{V}\}$. We say that $(\mathcal{U}, \mathcal{V})$ is a *valid decomposition* if for every $w \in \mathcal{UV}$ there is a unique pair $(u, v) \in \mathcal{U} \times \mathcal{V}$ such that $uv = w$.

Lemma 3.1.3. *Let $(\mathcal{U}, \mathcal{V})$ be a valid decomposition. Then, $\phi(\mathcal{UV}) = \phi(\mathcal{U})\phi(\mathcal{V})$.*

Proof. For individual walks it is evident that $\phi(uv) = \phi(u)\phi(v)$. Note that $\mathcal{UV} = \cup_{u\in\mathcal{U}} u\mathcal{V}$, where the sets $u\mathcal{V} \triangleq \{uv | v \in \mathcal{V}\}$ are mutually disjoint. By Lemma 3.1.1,

$$\phi(\mathcal{UV}) = \sum_{u\in\mathcal{U}} \phi(u\mathcal{V}) = \sum_{u\in\mathcal{U}} \sum_{v\in\mathcal{V}} \phi(uv) = \sum_{u\in\mathcal{U}} \sum_{v\in\mathcal{V}} \phi(u)\phi(v) = \left(\sum_{u\in\mathcal{U}} \phi(u)\right)\left(\sum_{v\in\mathcal{V}} \phi(v)\right)$$

where we have used $\phi(u\mathcal{V}) = \sum_{v\in\mathcal{V}} \phi(uv)$ because $u\mathcal{V}$ is one-to-one with $\mathcal{V}$. □

Note that $\mathcal{W}(i \to i)$ is the set of *self-return walks* at node i, i.e., walks which begin and end at node i. These self-return walks include walks which return to i many times. Let $\mathcal{W}(i \xrightarrow{\setminus i} i)$ be the set of all walks with non-zero length which begin and end at i but do not visit i in between. We call these the *single-revisit* self-return walks at node i. The set of self-return walks that return exactly k times is generated by taking the product of k copies of $\mathcal{W}(i \xrightarrow{\setminus i} i)$ denoted by $\mathcal{W}^k(i \xrightarrow{\setminus i} i)$. Thus, we obtain all self-return walks as:

$$\mathcal{W}(i \to i) = \cup_{k\geq 0} \mathcal{W}^k(i \xrightarrow{\setminus i} i) \tag{3.2}$$

where $\mathcal{W}^0(i \xrightarrow{\setminus i} i) \triangleq \{(i)\}$.

Similarly, recall that $\mathcal{W}(* \to i)$ denotes the set of all walks which end at node i. Let $\mathcal{W}(* \xrightarrow{\setminus i} i)$ denote the set of walks with non-zero length which end at node i and do not visit i previously (we call them *single-visit walks*). Thus, all walks which end at i are obtained as:

$$\mathcal{W}(* \to i) = \left(\{(i)\} \cup \mathcal{W}(* \xrightarrow{\setminus i} i)\right) \mathcal{W}(i \to i) \tag{3.3}$$

which is a valid decomposition.

Now we can decompose means and variances in terms of single-visit and single-revisit walk-sums, which we will use in Section 3.2.1 to analyze BP:

Proposition 3.1.4. *Let* $\alpha_i = \phi(i \overset{\setminus i}{\to} i)$ *and* $\beta_i = \phi_h(* \overset{\setminus i}{\to} i)$. *Then,*

$$P_{ii} = \frac{1}{1-\alpha_i} \quad \text{and} \quad \mu_i = \frac{h_i + \beta_i}{1-\alpha_i}$$

Proof. First note that the decomposition of $\mathcal{W}^k(i \overset{\setminus i}{\to} i)$ into products of k single-revisit self-return walks is a valid decomposition. By Lemma 3.1.3, $\phi(\mathcal{W}^k(i \overset{\setminus i}{\to} i)) = \phi^k(i \overset{\setminus i}{\to} i) = \alpha_i^k$. Then, by (3.2) and Lemma 3.1.1:

$$P_{ii} = \phi(i \to i) = \sum_k \alpha_i^k = \frac{1}{1-\alpha_i}$$

Walk-summability of the model implies convergence of the geometric series (i.e., $|\alpha_i| < 1$). Lastly, the decomposition in (3.3) implies

$$\mu_i = \phi_h(* \to i) = (h_i + \phi_h(* \overset{\setminus i}{\to} i))\phi(i \to i) = \frac{h_i + \beta_i}{1-\alpha_i} \quad \square$$

■ 3.1.3 Correspondence to Attractive Models

We have already shown that attractive models are walk-summable. Interestingly, it turns out that inference in any walk-summable model can be reduced to inference in a corresponding attractive model defined on a graph with twice as many nodes. The basic idea here is to separate out the walks with positive and negative weights.

Specifically, let $\hat{G} = (\hat{V}, \hat{\mathcal{E}})$ be defined as follows. For each node $i \in V$ we define two corresponding nodes $i_+ \in V_+$ and $i_- \in V_-$, and set $\hat{V} = V_+ \cup V_-$. For each edge $\{i,j\} \in \mathcal{E}$ with $r_{ij} > 0$ we define two edges $\{i_+, j_+\}, \{i_-, j_-\} \in \hat{\mathcal{E}}$, and set the partial correlations on these edges to be equal to r_{ij}. For each edge $\{i,j\} \in \mathcal{E}$ with $r_{ij} < 0$ we define two edges $\{i_+, j_-\}, \{i_-, j_+\} \in \hat{G}$, and set the partial correlations to be $-r_{ij}$. See Figure 3.4 for an illustration.

Let $(R_+)_{ij} = \max\{R_{ij}, 0\}$ and $(R_-)_{ij} = \max\{-R_{ij}, 0\}$, then, R can be expressed as the difference of these non-negative matrices: $R = R_+ - R_-$. Based on our construction, we have that $\hat{R} = \left(\begin{smallmatrix} R_+ & R_- \\ R_- & R_+ \end{smallmatrix}\right)$ and $\hat{J} = I - \hat{R}$. This defines a unit-diagonal information matrix $\hat{J}$ on $\hat{G}$. Note that if $\hat{J} \succ 0$ then this defines a valid attractive model.

Proposition 3.1.5. $\hat{J} = I - \hat{R} \succ 0$ *if and only if* $J = I - R$ *is walk-summable.*

The proof relies on the Perron-Frobenius theorem and is given in Appendix A.1. Now, let $h = h_+ - h_-$ with $(h_+)_i = \max\{h_i, 0\}$ and $(h_-)_i = \max\{-h_i, 0\}$. Define $\hat{h} = \left(\begin{smallmatrix} h_+ \\ h_- \end{smallmatrix}\right)$. Now we have the information form model $(\hat{h}, \hat{J})$ which is a valid, attractive

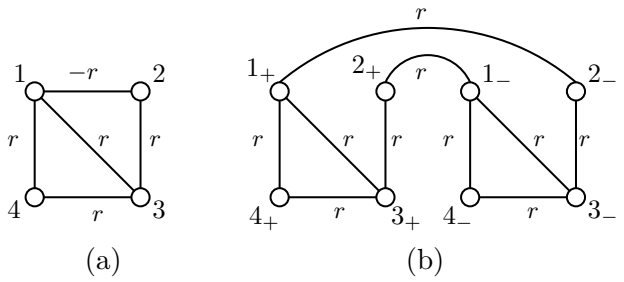


Figure 3.4. (a) A frustrated model defined on G with one negative edge ($r > 0$). (b) The corresponding attractive model defined on $\hat{G}$.

model and also has non-negative node potentials. Performing inference with respect to this augmented model, we obtain the mean vector $\hat{\mu} = \begin{pmatrix} \hat{\mu}_+ \\ \hat{\mu}_- \end{pmatrix} \triangleq \hat{J}^{-1}\hat{h}$ and covariance matrix $\hat{P} = \begin{pmatrix} \hat{P}_{++} & \hat{P}_{+-} \\ \hat{P}_{-+} & \hat{P}_{--} \end{pmatrix} \triangleq \hat{J}^{-1}$. From these calculations, we can obtain the moments (μ, P) of the original walk-summable model (h, J):

Proposition 3.1.6. *$P = \hat{P}_{++} - \hat{P}_{+-}$ and $\mu = \hat{\mu}_+ - \hat{\mu}_-$.*

The proof appears in Appendix A.1. This proposition shows that estimation of walk-summable models may be reduced to inference in an attractive model in which all walk-sums are sums of positive weights. In essence, this is accomplished by summing walks with positive and negative weights separately and then taking the difference, which is only possible for walk-summable models.

■ 3.1.4 Pairwise-Normalizability

To simplify presentation we assume that the graph does not contain any isolated nodes (a node without any incident edges). Then, we say that the information matrix J is *pairwise-normalizable* (PN) if we can represent J in the form

$$J = \sum_{e \in \mathcal{E}} [J_e]$$

where each J_e is a 2×2 symmetric, positive definite matrix.[6] The notation $[J_e]$ means that J_e is zero-padded to a $|V| \times |V|$ matrix with its principal submatrix for $\{i, j\}$ being J_e (with $e = \{i, j\}$). Thus, $x^T[J_e]x = x_e^T J_e x_e$. Pairwise-normalizability implies that $J \succ 0$ because each node is covered by at least one positive definite submatrix J_e. Let $\mathcal{J}_{PN}$ denote the set of $n \times n$ pairwise-normalizable information matrices J (not requiring unit-diagonal normalization). This set has nice convexity properties. Recall that a set $\mathcal{X}$ is *convex* if $x, y \in \mathcal{X}$ implies $\lambda x + (1 - \lambda)y \in \mathcal{X}$ for all $0 \leq \lambda \leq 1$ and is a *cone* if $x \in \mathcal{X}$ implies $\alpha x \in \mathcal{X}$ for all $\alpha > 0$. A cone $\mathcal{X}$ is *pointed* if $\mathcal{X} \cap -\mathcal{X} = \{0\}$.

[6] An alternative definition of pairwise-normalizability is the existence of a decomposition $J = cI + \sum_{e \in \mathcal{E}} [J_e]$, where $c > 0$, and $J_e \succeq 0$. For graphs without isolated nodes, both definitions are equivalent.

Proposition 3.1.7 (Convexity of PN models). *The set $\mathcal{J}_{PN}$ is a convex pointed cone.*

The proof is in Appendix A.1. We now establish the following fundamental result:

Proposition 3.1.8 (WS $\Leftrightarrow$ PN). *$J = I - R$ is walk-summable if and only if it is pairwise-normalizable.*

Our proof appears in in Appendix A.1. An equivalent result has been derived independently in the linear algebra literature: [18] establishes that symmetric H-matrices with positive diagonals (which is equivalent to WS by part (iv) of Proposition 3.1.1) are equivalent to matrices with factor width at most two (PN models). However, the result $PN \Rightarrow WS$ was established earlier by [72]. Our proof for $WS \Rightarrow PN$ uses the Perron-Frobenius theorem, whereas [18] use the generalized diagonal dominance property of H-matrices. Also our proof reveals the following connection between the two notions: define the strength of walk-summability as $\epsilon_{WS} = 1 - \varrho(\bar{R})$ and the strength of pairwise-normalizability as $\epsilon_{PN} = \max_c$ such that $J = cI + \sum_{e \in \mathcal{E}} [J_e]$ with $J_e \succeq 0$. Strength of WS and PN measure how much the model can be perturbed while still being WS or PN, respectively. Then:

Corollary 3.1.3. *For normalized (unit-diagonal) J we have $\epsilon_{WS} = \epsilon_{PN}$.*

Equivalence to pairwise-normalizability gives much insight into the set of walk-summable models. For example, the set of unit-diagonal J matrices that are walk-summable is convex, because it is the intersection of $\mathcal{J}_{PN}$ with an affine space. Also, the set of walk-summable J matrices that are sparse with respect to a particular graph G (with some entries of J are restricted to 0) is convex.

Another important class of models are those that have a *diagonally dominant* information matrix, i.e., where for each i it holds that $\sum_{j \neq i} |J_{ij}| < J_{ii}$.

Proposition 3.1.9. *Diagonally dominant models are pairwise-normalizable (walk-summable).*

A constructive proof is given in Appendix A.1. The converse does not hold: not all pairwise-normalizable models are diagonally dominant. For instance, in our 4-cycle with a chord example, Figure 3.1(a), with $r = .38$ the model is not diagonally dominant but is walk-summable and hence pairwise-normalizable.

■ 3.2 Walk-sum Interpretation of Belief Propagation

In this section we use the concepts and machinery of walk-sums to analyze belief propagation. We begin with models on trees, for which, as we show, all valid models are walk-summable. Moreover, for these models we show that exact walk-sums over infinite sets of walks for means and variances can be computed efficiently in a recursive fashion. We show that these walk-sum computations map exactly to belief propagation updates. These results (and the computation tree interpretation of LBP recursions) then provide the foundation for our analysis of loopy belief propagation in Section 3.2.2.

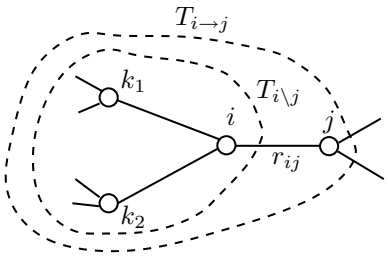


Figure 3.5. Illustration of the subtree notation, $T_{i\to j}$ and $T_{i\setminus j}$.

■ 3.2.1 Walk-Sums and BP on Trees

Our analysis of BP makes use of the following property:

Proposition 3.2.1 (Trees are walk-summable). *For tree structured models $J \succ 0 \Leftrightarrow \varrho(\bar{R}) \leq 1$ (i.e., all valid trees are walk-summable). Also, for trees $\varrho(\bar{R}) = \varrho(R) = \lambda_{\max}(R)$.*

Proof. The proof is a special case of the proof of Corollary 3.1.2. Trees are non-frustrated (as they contain no cycles, and hence no frustrated cycles) so they are walk-summable. Negating some variables makes the model attractive and does not change the eigenvalues. □

The proposition shows that walk-sums for means and variances are always defined on tree-structured models, and can be reordered in arbitrary ways without affecting convergence. We rely on this fact heavily in subsequent sections. The next two results identify walk-sum variance and mean computations with the BP update equations. The ingredients for these results are decompositions of the variance and mean walk-sums in terms of sums over walks on subtrees, together with the decomposition in terms of single-revisit and single-visit walks provided in Proposition 3.1.4.

Walk-Sum Variance Calculation Let us look first at the computation of the variance at node j, which is equal to the self-return walk-sum $\phi(j \to j)$, and which from Proposition 3.1.4 can be computed directly from the single-revisit walk sums $\alpha_j = \phi(j \overset{\setminus j}{\to} j)$. This latter walk-sum can be further decomposed into sums over disjoint subsets of walks each of which corresponds to single-revisit self-return walks that exit node j via a specific one of its neighbors, say i. In particular, as illustrated in Figure 3.5, the single-revisit self-return walks that do this correspond to walks that live in the subtree $T_{i\to j}$. Using the notation $\alpha_{i\to j} \triangleq \phi(j \overset{\setminus j}{\to} j \mid T_{i\to j})$ for the walk-sum over the set of all single-revisit walks which are restricted to stay in subtree $T_{i\to j}$ we see that

$$\alpha_j = \phi(j \overset{\setminus j}{\to} j) = \sum_{i\in\mathcal{N}(j)} \phi(j \overset{\setminus j}{\to} j \mid T_{i\to j}) = \sum_{i\in\mathcal{N}(j)} \alpha_{i\to j}$$

Moreover, every single-revisit self-return walk that lives in $T_{i\to j}$ must leave *and* return to node j through the single edge (i,j), and between these first and last steps must execute a (possibly multiple-revisit) self-return walk at node i that is constrained *not* to pass through node j, i.e., to live in the subtree $T_{i\backslash j}$ indicated in Figure 3.5. Thus

$$\alpha_{i\to j} = \phi(j \overset{\backslash j}{\to} j \mid T_{i\to j}) = r_{ij}^2 \phi(i \to i \mid T_{i\backslash j}) \triangleq r_{ij}^2 \gamma_{i\backslash j} \tag{3.4}$$

We next show that the walk-sums α_j and $\alpha_{i\to j}$ (hence variances P_j) can be efficiently calculated by a walk-sum analog of belief propagation. We have the following result:

Proposition 3.2.2. *Consider a valid tree model $J = I - R$. Then $\alpha_{i\to j} = -\Delta J_{i\to j}$ and $\gamma_{i\backslash j} = \hat{J}_{i\backslash j}^{-1}$, where $\Delta J_{i\to j}$ and $\hat{J}_{i\backslash j}^{-1}$ are the quantities defined in the Gaussian BP equations (2.29) and (2.30).*

See Appendix A.1 for the proof. This gives us a walk-sum interpretation of LBP message updates for variances.

Walk-Sum Mean Calculation We extend the above analysis to calculate means in trees. Mean μ_j is the reweighted walk-sum over walks that start anywhere and end at node j, $\mu_j = \phi_h(* \to j)$. Any walk that ends at node j can be expressed as a single-visit walk to node j followed by a multiple-revisit self-return walk from node j: $\phi_h(* \to j) = \left(h_j + \phi_h(* \overset{\backslash j}{\to} j)\right) \phi(j \to j)$, where the term h_j corresponds to the length-0 walk that starts and ends at node j.

As we have done for the variances, the single-visit walks to node j can be partitioned into the single-visit walks that reach node j from each of its neighbors, say node i and thus prior to this last step across the edge (i,j), reside in the subtree $T_{i\backslash j}$, so that

$$\beta_{i\to j} \triangleq \phi_h(* \overset{\backslash j}{\to} j \mid T_{i\to j}) = r_{ij} \phi_h(* \to i \mid T_{i\backslash j})$$

Proposition 3.2.3. *Consider a valid tree model $J = I - R$. Then $\beta_{i\to j} = \Delta h_{i\to j}$, where $\Delta h_{i\to j}$ is the quantity defined in the Gaussian BP equation (2.30).*

The proof appears in Appendix A.1. Now we have a complete walk-sum interpretation of Gaussian LBP message updates.

■ 3.2.2 LBP in Walk-Summable Models

In this subsection we use the LBP computation tree to show that LBP includes all the walks for the means, but only a subset of the walks for the variances. This allows us to prove LBP convergence for all walk-summable models. In contrast, for non-walksummable models LBP may or may not converge (and in fact the variances may converge while the means do not). As we will see in Section 3.3, this can be analyzed by examining not the walk-summability of the original model but the walk-summability (and hence the validity) of the computation tree.

As we have discussed in Section 2.2.3, running LBP for some number of iterations yields identical calculations at any particular node i to the exact inference calculations on the corresponding computation tree rooted at node i. We use the notation $T_i^{(n)}$ for the depth-n computation tree at node i, T_i for the full computation tree (as $n \to \infty$) and we assign the label 0 to the root node. $P_0(T_i^{(n)})$ is the variance at the root node of the nth computation tree rooted at node i in G. The LBP variance estimate at node i after n steps is equal to:

$$\hat{P}_i^{(n)} = P_0(T_i^{(n)}) = \phi(0 \to 0 \mid T_i^{(n)})$$

Similarly, the LBP estimate of the mean μ_i after n steps of LBP is:

$$\hat{\mu}_i^{(n)} = \mu_0(T_i^{(n)}) = \phi_h(* \to 0 \mid T_i^{(n)})$$

As we have mentioned in Section 2.2.3, the definition of the computation trees $T_i^{(n)}$ depends upon the message schedule $\{\mathcal{M}^{(n)}\}$ of LBP, which specifies which subset of messages are updated at iteration n. We say that a message schedule is *proper* if every message is updated infinitely often, i.e., if for every $m > 0$ and every directed edge (i, j) in the graph there exists $n > m$ such that $(i, j) \in \mathcal{M}^{(n)}$. Clearly, the fully-parallel form is proper since every message is updated at every iteration. Serial forms which iteratively cycle through the directed edges of the graph are also proper. All of our convergence analysis in this section presumes a proper message schedule. We remark that as walk-summability ensures convergence of walk-sums independent of the order of summation, it makes the choice of a particular message schedule unimportant in our convergence analysis. The following result relating walks in the loopy graph G and walks in the computation tree T_i is proven in Appendix A.1.

Lemma 3.2.1 (Walks in G and in T_i). *There is a one-to one correspondence between finite-length walks in G that end at i, and walks in T_i that end at the root node. In particular, for each such walk in G there is a corresponding walk in $T_i^{(n)}$ for n large enough.*

Now, recall that to compute the mean μ_i we need to gather walk-sums over all walks that start anywhere and end at i. We have just shown that LBP gathers all of these walks as the computation tree grows to infinity. The story for the variances is different. The true variance P_{ii} is a walk-sum over all self-return walks that start and end at i in G. However, walks in G that start and end at i may map to walks that start at the root node of $T_i^{(n)}$, but end at a *replica* of the root node instead of the root. These walks are not captured by the LBP variance estimate.[7] The walks for the variance estimate $P_0(T_i^{(n)})$ are self-return walks $\mathcal{W}(0 \to 0 \mid T_i^{(n)})$ that start and end at

[7]Recall that the computation tree is a representation of the computations seen at the root node of the tree, and it is *only* the computation at *this node*—i.e., at *this replica* of node i that corresponds to the LBP computation at node i in G.

the root node in the computation tree. Consider Figure 2.7. The walk $(1, 2, 3, 1)$ is a self-return walk in the original graph G but is *not* a self-return walk in the computation tree shown in Figure 2.7(d). LBP variances capture only those self-return walks of the original graph G which are also self-return walks in the computation tree—e.g., the walk $(1, 3, 2, 3, 4, 3, 1)$ is a self-return walk in both Figures 2.7(a) and (d). We call such walks *backtracking*. Hence,

Lemma 3.2.2 (Self-return walks in G and in T_i). *The LBP variance estimate at each node is a sum over the backtracking self-return walks in G, a subset of all self-return walks needed to calculate the correct variance.*

Note that back-tracking walks for the variances have positive weights, since each edge in the walk is traversed an even number of times. With each LBP step the computation tree grows and new back-tracking walks are included, hence variance estimates grow monotonically.[8]

We have shown which walks LBP gathers based on the computation tree. It remains to analyze the convergence of the walk-sums for these walks. In walk-summable models the answer is simple:

Lemma 3.2.3 (Computation trees of WS models are WS). *For a walk-summable model all its computation trees $T_i^{(n)}$ (for all n and i) are walk-summable and hence valid (positive-definite).*

Intuitively, walks in the computation tree $T_i^{(n)}$ are subsets of the walks in G, and hence they converge. That means that the computation trees are walk-summable, and hence valid. This argument can be made precise, but a shorter formal proof using monotonicity of the spectral radius (3.1) appears in Appendix A.1. Next, we use these observations to show convergence of LBP for walk-summable models.

Proposition 3.2.4 (Convergence of LBP for walk-summable models). *If a model on a graph G is walk-summable, then LBP is well-posed, the means converge to the true means and the LBP variances converge to walk-sums over the backtracking self-return walks at each node.*

Proof. Let $\mathcal{W}(i \overset{BT}{\to} i)$ denote the back-tracking self-return walks at node i. By Lemmas 3.2.1 and 3.2.2, we have:

$$\begin{aligned} \mathcal{W}(* \to i) &= \cup_n \mathcal{W}(* \to 0 | T_i^{(n)}) \\ \mathcal{W}(i \overset{BT}{\to} i) &= \cup_n \mathcal{W}(0 \to 0 | T_i^{(n)}) \end{aligned}$$

[8] Monotonically increasing variance estimates is a characteristic of the particular initialization of LBP that we use, i.e., the potential decomposition (2.24) together with uninformative initial messages. If one instead uses a pairwise-normalized potential decomposition, the variances are then monotonically *decreasing*.

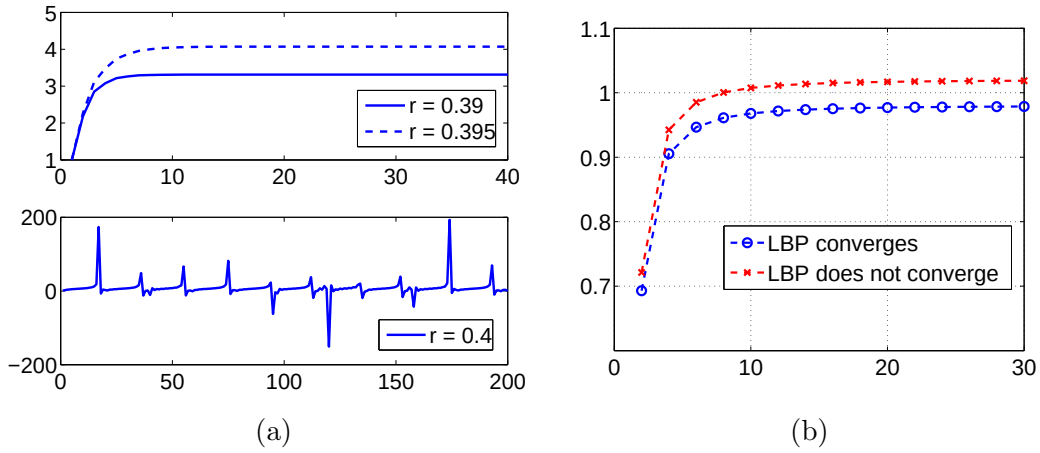


Figure 3.6. (a) LBP variances vs. iteration. (b) $\varrho(R_n)$ vs. iteration.

We note that the computation trees $T_i^{(n)}$ at node i are nested, $T_i^{(n)} \subset T_i^{(n+1)}$ for all n. Hence, $\mathcal{W}(* \to 0|T_i^{(n)}) \subset \mathcal{W}(* \to 0|T_i^{(n+1)})$ and $\mathcal{W}(0 \to 0|T_i^{(n)}) \subset \mathcal{W}(0 \to 0|T_i^{(n+1)})$. Then, by Lemma 3.1.2, we obtain the result:

$$\begin{aligned} \mu_i = \phi_h(* \to i) &= \lim_{n\to\infty} \phi_h(* \to 0|T_i^{(n)}) = \lim_{n\to\infty} \hat{\mu}_i^{(n)} \\ P_i^{(BT)} \triangleq \phi(i \overset{BT}{\to} i) &= \lim_{n\to\infty} \phi(0 \to 0|T_i^{(n)}) = \lim_{n\to\infty} \hat{P}_i^{(n)}. \end{aligned} \qquad \square$$

Corollary 3.2.1. *LBP converges for attractive, non-frustrated, and diagonally dominant models. In attractive and non-frustrated models LBP variance estimates are less than or equal to the true variances (the missing non-backtracking walks all have positive weights).*

In [133] Gaussian LBP is analyzed for pairwise-normalizable models. They show convergence for the case of diagonally dominant models, and correctness of the means in case of convergence. The class of walk-summable models is strictly larger than the class of diagonally dominant models, so our sufficient condition is stronger. They also show that LBP variances omit some terms needed for the correct variances. These terms correspond to correlations between the root and its replicas in the computation tree. In our framework, each such correlation is a walk-sum over the subset of non-backtracking self-return walks in G which, in the computation tree, begin at a particular replica of the root.

Example 2. Consider the model in Figure 3.1(a). We summarize various critical points for this model in Figure 3.7. For $0 \le r \le .39039$ the model is walk-summable and LBP converges; then for a small interval $.39039 \le r \le .39865$ the model is not walk-summable but LBP still converges, and for larger r LBP does not converge. We

apply LBP to this model with $r = 0.39, 0.395$ and 0.4, and plot the LBP variance estimates for node 1 vs. the iteration number in Figure 3.6(a). LBP converges in the walk-summable case for $r = .39$, with $\varrho(\bar{R}) \approx .9990$. It also converges for $r = 0.395$ with $\varrho(\bar{R}) \approx 1.0118$, but soon fails to converge as we increase r to 0.4 with $\varrho(\bar{R}) \approx 1.0246$.

Also, for $r = .4$, we note that $\varrho(R) = .8 < 1$ and the series $\sum_l R^l$ converges (but $\sum_l \bar{R}^l$ does not) and LBP does not converge. Hence, $\varrho(R) < 1$ is *not* sufficient for LBP convergence showing the importance of the stricter walk-summability condition $\varrho(\bar{R}) < 1$.

■ 3.3 LBP in Non-Walksummable Models

While the condition in Proposition 3.2.4 is necessary and sufficient for certain special classes of models—e.g., for trees and single cycles—it is only sufficient more generally, and, as in Example 2, LBP may converge for some non-walksummable models. We extend our analysis to develop a tighter condition for convergence of LBP variances based on a weaker form of walk-summability defined with respect to the computation trees (instead of G). We have shown in Proposition 3.2.1 that for trees walk-summability and validity are equivalent, and $\varrho(\bar{R}) < 1 \Leftrightarrow \varrho(R) < 1 \Leftrightarrow J \succ 0$. Hence our condition essentially corresponds to validity of the computation tree.

First, we note that when a model on G is valid (J is positive-definite) but not walk-summable, then some finite computation trees may be invalid (indefinite). This turns out to be the primary reason why belief propagation can fail to converge. Walk-summability on the original graph implies walk-summability (and hence validity) on all of its computation trees. But if the model is not walk-summable, then its computation tree may or may not be valid.

We characterize walk-summability of the computation trees as follows. Let $T_i^{(n)}$ be the nth computation tree rooted at some node i. We define $R_i^{(n)} \triangleq I - J_i^{(n)}$ where $J_i^{(n)}$ is the normalized information matrix for $T_i^{(n)}$ and I is an identity matrix. The nth computation tree $T_i^{(n)}$ is walk-summable (valid) if and only if $\varrho(R_i^{(n)}) < 1$ due to the fact that $\varrho(\bar{R}_i^{(n)}) = \varrho(R_i^{(n)})$ for trees. We are interested in the validity of all finite computation trees, so we consider the quantity $\lim_{n\to\infty} \varrho(R_i^{(n)})$. Lemma 3.3.1 guarantees the existence of this limit:

Lemma 3.3.1. *The sequence* $\{\varrho(R_i^{(n)})\}$ *is monotonically increasing and bounded above by* $\varrho(\bar{R})$. *Thus,* $\lim_{n\to\infty} \varrho(R_i^{(n)})$ *exists, and is equal to* $\sup_n \varrho(R_i^{(n)})$.

In the proof we use *k-fold graphs*, which we introduce in Appendix A.1.1. The proof appears in Appendix A.1. The limit in Lemma 3.3.1 is defined with respect to a particular root node and message schedule. The next lemma shows that for connected graphs, as long as the message schedule is proper, they do not matter.

Lemma 3.3.2. *For connected graphs and with a proper message schedule, the limit* $\varrho_\infty \triangleq \lim_{n\to\infty} \varrho(R_i^{(n)})$ *is independent of* i *and the choice of proper message schedule.*

This independence results from the fact that for large n the computation trees rooted at different nodes overlap significantly. Technical details of the proof appear in Appendix A.1. Using this lemma we suppress the dependence on the root-node i from the notation to simplify matters. The limit ϱ_∞ turns out to be critical for convergence of LBP variances:

Proposition 3.3.1 (LBP validity/variance convergence). *(i) If* $\varrho_\infty < 1$*, then all finite computation trees are valid and the LBP variances converge to walk-sums over the back-tracking self-return walks. (ii) If* $\varrho_\infty > 1$*, then the computation tree eventually becomes invalid and LBP fails (produces negative variances).*

Proof. (i) Since $\varrho_\infty = \lim_{n\to\infty} \varrho(R^{(n)}) < 1$ and the sequence $\{\varrho(R^{(n)})\}$ is monotonically increasing, then there exists $\delta > 0$ such that $\varrho(R^{(n)}) \le 1 - \delta$ for all n. This implies that all the computation trees $T^{(n)}$ are walk-summable and that LBP variances monotonically increase (since weights of backtracking walks are positive; see the discussion after Lemma 3.2.2). We have that $\lambda_{\max}(R^{(n)}) \le 1 - \delta$, so $\lambda_{\min}(J^{(n)}) \ge \delta$ and $\lambda_{\max}(P^{(n)}) \le \frac{1}{\delta}$. The maximum eigenvalue of a matrix is a bound on the maximum entry of the matrix, so $(P^{(n)})_{ii} \le \lambda_{\max}(P^{(n)}) \le \frac{1}{\delta}$. The variances are monotonically increasing and bounded above, hence they converge.

(ii) If $\lim_{n\to\infty} \varrho(R^{(n)}) > 1$, then there exists an m such that $\varrho(R^{(n)}) > 1$ for all $n \ge m$. This means that these computation trees $T^{(n)}$ are invalid, and that the variance estimates at some of the nodes are negative. □

As discussed in Section 2.3.1, the LBP computation tree is valid if and only if the information parameters $\hat{J}_{i\backslash j}^{(n)}$ and $\hat{J}_i^{(n)}$ in (2.29), (2.31) computed during LBP iterations are strictly positive for all n. Hence, it is easily detected if the LBP computation tree becomes invalid. In this case, continuing to run LBP is not meaningful and will lead to division by zero (if the computation tree is singular) or to negative variances (if it is not positive definite).

Recall that the limit ϱ_∞ is invariant to message order by Lemma 3.3.2. Hence, by Proposition 3.3.1, convergence of LBP variances is likewise invariant to message order (except possibly when $\varrho_\infty = 1$). The limit ϱ_∞ is bounded above by $\varrho(\bar{R})$, hence walk-summability in G is a sufficient condition for well-posedness of the computation tree: $\varrho_\infty \le \varrho(\bar{R}) < 1$. However, the bound is not tight in general (except for trees and single cycles). This is related to the phenomenon that the limit of the spectral radius of the finite computation trees can be less than the spectral radius of the infinite computation tree (which has no leafs). See [61] for analysis of a related discrepancy.

Means in non-WS models For the case in which $\rho_\infty < 1 < \varrho(\bar{R})$, the walk-sums for LBP variances converge absolutely (see proof of Proposition 3.3.1), but the walk-sums for the means do not converge absolutely. The reason is that LBP only computes

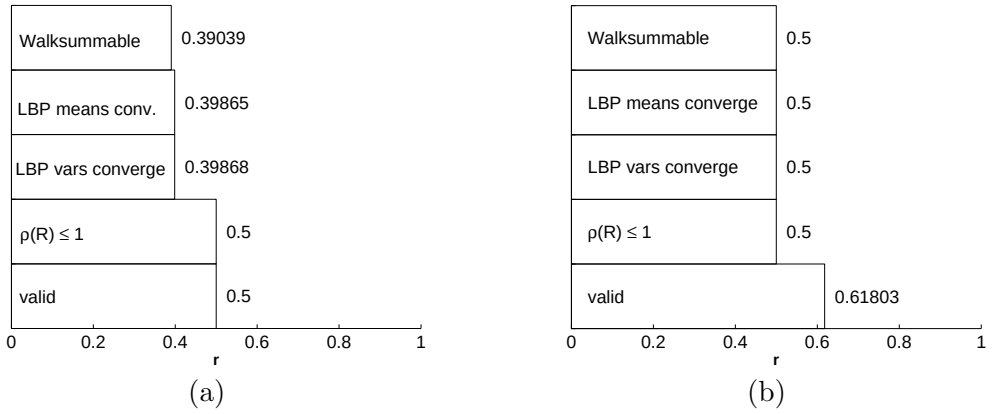


Figure 3.7. Critical regions for example models from Figure 3.1. (a) 4-cycle with a chord. (b) 5-cycle.

a subset of the self-return walks for the variances but captures all the walks for the means. However, the series LBP computes for the means, corresponding to a particular ordering of walks, may or may not still converge.

It is well-known [111] that once variances converge, the updates for the means follow a linear system. Consider (2.29) and (2.30) with $\hat{J}_{i\backslash j}$ fixed, then the LBP messages for the means $\Delta h = (\Delta h_{i\to j} \mid \{i,j\} \in \mathcal{E})$ clearly follow a linear system update. For the parallel message schedule we can express this as:

$$\Delta h^{(n+1)} = L\ \Delta h^{(n)} + b \tag{3.5}$$

for some matrix L and some vector b. Convergence of this system depends on the spectral radius $\varrho(L)$. However, it is difficult to analyze $\varrho(L)$ since the matrix L depends on the converged values of the LBP variances. To improve convergence of the means, one can damp the message updates by modifying (2.30) as follows:

$$\Delta h_{i\to j}^{(n+1)} = (1-\alpha)\Delta h_{i\to j}^{(n)} + \alpha(-J_{ij}(\hat{J}_{i\backslash j}^{(n)})^{-1}\hat{h}_{i\backslash j}^{(n)}) \quad \text{with} \quad 0 < \alpha \leq 1 \tag{3.6}$$

We have observed in experiments that for all the cases in which variances converge we also obtain convergence of the means with enough damping of BP messages. We have also tried damping the updates for the ΔJ messages, but whether or not variances converge appears to be independent of damping. Apparently, it is the validity of the computation tree ($\rho_\infty < 1$) that is essential for convergence of both means and variances in damped versions of Gaussian LBP.

Example 3. We illustrate Proposition 3.3.1 on a simple example. Consider the 5-node cycle model from Figure 3.1(b). In Figure 3.6(b), for $\rho = .49$ we plot $\varrho(R_n)$ vs. n (lower curve) and observe that $\lim_{n\to\infty} \varrho(R_n) \approx .98 < 1$, and LBP converges. For $\rho = .51$ (upper curve), the model defined on the 5-node cycle is still valid but $\lim_{n\to\infty} \varrho(R_n) \approx 1.02 > 1$ and LBP fails: it does not converge and eventually starts producing negative variances.

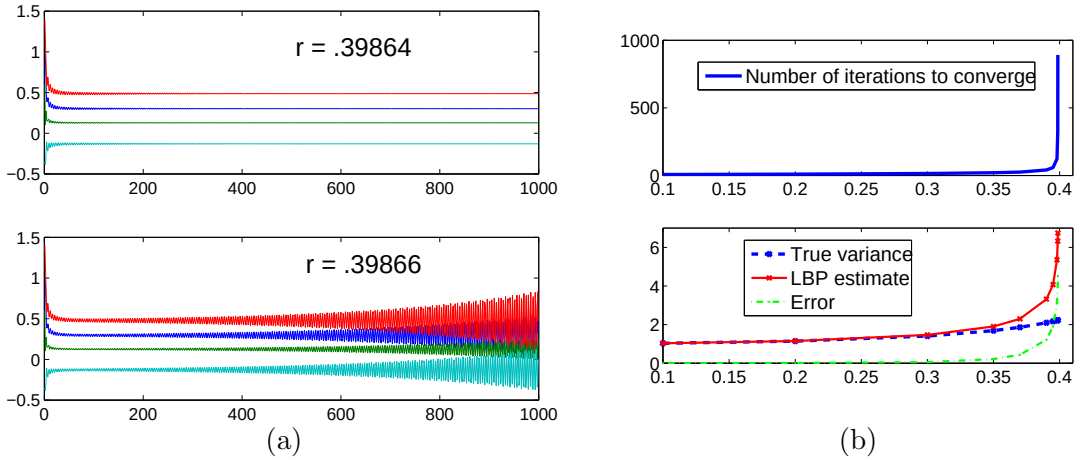


Figure 3.8. The 4-cycle with a chord example. (a) Convergence and divergence of the means near the LBP mean critical point. (b) Variance near the LBP variance critical point: (top) number of iterations for variances to converge, (bottom) true variance, LBP estimate and the error at node 1.

As we mentioned, in non-walksummable models the series that LBP computes for the means is not absolutely convergent and may diverge even when variances converge. For our 4-cycle with a chord example in Figure 3.1(a), the region in which variances converge but means diverge is very narrow, $r \approx .39865$ to $r \approx .39867$ (we use the parallel message schedule here; the critical point for the means is slightly higher using a serial schedule). In Figure 3.8(a) we show mean estimates vs. the iteration number on both sides of the LBP mean critical point for $r = 0.39864$ and for $r = 0.39866$. In the first case the means converge, while in the latter they slowly but very definitely diverge. The spectral radius of the linear system for mean updates in (3.5) for the two cases is $\varrho(L) = 0.99717 < 1$ and $\varrho(L) = 1.00157 > 1$, respectively. In the divergent example, all the eigenvalues of L have real components less than 1 (the maximum such real component is $0.8063 < 1$). Thus by damping we can force all the eigenvalues of L to enter the unit circle: the damped linear system is $(1-\alpha)I + \alpha L$. Using $\alpha = 0.9$ in (3.6) the means converge.

In Figure 3.8(b) we illustrate that near the LBP variance critical point, the LBP estimates become more difficult to obtain and their quality deteriorates dramatically. We consider the graph in Figure 3.1(a) again as r approaches 0.39867, the critical point for the convergence of the variances. The picture shows that the number of iterations as well as the error in LBP variance estimates explode near the critical point. In the figure we show the variance at node 1, but similar behavior occurs at every node. In Figure 3.7, we summarize the critical points of both models from Figure 3.1.

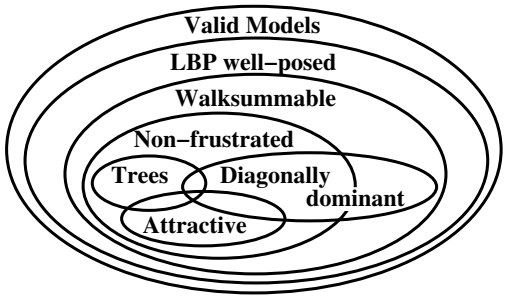


Figure 3.9. Venn diagram summarizing various subclasses of Gaussian models.

■ 3.4 Chapter Summary

We have presented a walk-sum interpretation of inference in Gaussian graphical models, which holds for a wide class of models that we call walk-summable. We have shown that walk-summability encompasses many classes of models which are considered "easy" for inference—trees, attractive, non-frustrated and diagonally dominant models—but also includes many models outside of these classes. A Venn diagram summarizing relations between these sets appears in Figure 3.9. We have also shown the equivalence of walk-summability to pairwise-normalizability.

We have established that in walk-summable models LBP is guaranteed to converge, for both means and variances, and that upon convergence the means are correct, whereas the variances only capture walk-sums over back-tracking walks. We have also used the walk-summability of valid (positive-definite) models on trees to develop a more complete picture of LBP for non-walksummable models, relating variance convergence to validity of the LBP computation tree.

In the next chapter we use combinatorial ideas to compute walk-sums in regular graphs, and develop vector and factor-graph extensions of walk-summability and their connection to LBP. We describe possible directions for further work in Chapter 6, including ideas based on walks or paths for inference in discrete models, and improvements over LBP which attempt to capture more walks for variances. We also note that walk-sum analysis has been applied to the analysis of the embedded trees algorithm for Gaussian inference in [26].

Chapter 4

Extensions: Combinatorial, Vector and Factor Graph Walk-sums

In this chapter we continue our study of the walk-sum framework presented in Chapter 3. First we consider combinatorial ideas for calculating walk-sums in regular graphs, which shed light on the stability of the computation trees in the difficult case with $\varrho_\infty = 1$, and give insight into the accuracy of LBP variances. Next, we consider generalized walk-sums in vector and factor graph Gaussian graphical models. The corresponding vector and factor graph versions of LBP provide a rich class of algorithms with trade-off between accuracy and convergence versus computational complexity. We develop sufficient conditions for vector and factor graph walk-summability, and convergence of LBP, and also show that these settings are inherently more complex than the scalar walk-sum framework in Chapter 3. Finally we talk about a rich class of factor graph normalizable models, and show that this condition is sufficient to guarantee convergence of the variances. We also relate this factor graph normalizable condition to a recently proposed complex-valued version of Gaussian LBP.

■ 4.1 Combinatorial Walk-sum Analysis

In this section we apply some basic counting ideas to analyze Gaussian walk-sums in regular graphs. Recall that a graph is k-regular if the degree of every vertex is k. In addition, we call the edge-weights *homogeneous* if they are all the same. First we use counting ideas to give insight into LBP variance behaviour in the critical case with $\varrho_\infty = 1$. Then, we analyze the accuracy of LBP variances in regular graphs with homogeneous weights (we may also negate some or all of the edge-weights on occasion). And finally we derive an approximate expression for expected walk-sums using a stochastic model for edge-weights in regular graphs with non-homogeneous weights.

■ 4.1.1 LBP Variance Estimates for $\varrho_\infty = 1$

Recall Proposition 3.3.1 in Chapter 3 regarding convergence of LBP variances in non-walk-summable models. It states that LBP variances converge if $\varrho_\infty < 1$, and that they

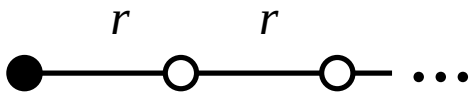


Figure 4.1. (top) One sided infinite chain, with homogeneous weights r. The root node is filled.

diverge if $\varrho_\infty > 1$, but leaves the case $\varrho_\infty = 1$ open. A reasonable conjecture may be that in this case LBP variances stay positive but diverge to infinity as LBP progresses. Below we analyze this conjecture from a combinatorial perspective, and show that for single cycles with homogeneous weights the conjecture holds, but that in general for more complex graphs it does not, and LBP variances may approach a finite limit when $\varrho_\infty = 1$.

The LBP variance walk-sum is a multivariate power-series in the edge-weights. Suppose there are M edges in the graph, with weights r_e. Let α be a multi-index, $\alpha = (\alpha_1, ..., \alpha_M)$, with $\alpha_e \in \mathcal{N}$. Then $\phi(i \to i) = \sum_\alpha w_\alpha r^\alpha$, where $r^\alpha = \prod r_e^{\alpha_e}$, and w_α is the number of walks which go α_e times across each edge e. The power series converges for r in the interior of its region of convergence[1], but it may or may not converge on its boundary: for example consider the univariate power series $\sum_k \frac{1}{k} r^k$, and $\sum_k \frac{1}{k^2} r^k$. The radius of convergence is $r = 1$ for both. However, for $r = 1$ the series $\sum_k \frac{1}{k^2}$ is finite, whereas $\sum_k \frac{1}{k}$ diverges. We are interested if these two cases are possible for convergence of walk-sums.

Next we use combinatorics to analytically compute self-return walk-sums at the root node of some simple graphs: a single-sided chain, a k-tree, a two-sided chain, and a computation tree of a k-regular graph. We will pay particular attention to the critical case where the models are near-singular, and this will give insight into the behavior of variances for LBP. We assume that all the edge-weights are set to r. We also note that in tree-structured models (including computation trees) the self-return walk-sums will not get affected by negating weights on some of the edges: every edge appears an even number of times in each walk, and hence, the signs cancel out. Thus, our calculations apply not only to regular chains and trees with homogeneous weights r, but also to the case where edge-weights are $\pm r$. However, in loopy graphs negating some of the edge-weights may change the self-return walk-sums, and may even affect the validity (positive-definiteness) of $J = I - R$. We will see the implications of this fact for loopy graphs and their computation trees.

One-sided infinite chain. First, we consider an infinite homogeneous chain rooted at node 0 and extending in one direction, with all the edge-weights set to r, see Figure 4.1. The infinite single-sided chain is not a computation tree for any finite graph, but

[1]For univariate power series expansion around 0, the region of convergence is a disk of some radius r_0 around 0, with its interior being the open disk $|r| < r_0$. For multivariate power series, the region of convergence is a more complicated object – a log-convex set.

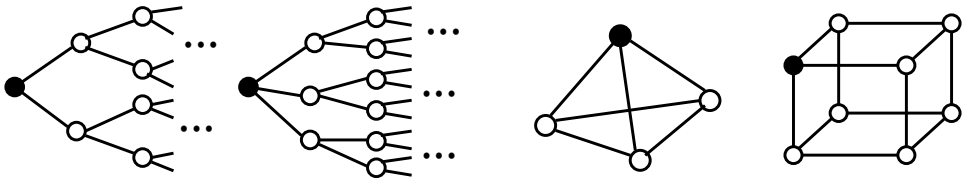


Figure 4.2. (a) A 2-tree, i.e. binary tree, (b) the computation tree for a 3-regular graph and (c) example 3-regular graphs which both generate the computation tree in (b).

we will shortly use it to find walk-sums for computation trees of regular graphs. The number of walks of length $2n$ that start and end at node 0, i.e. the self-return walks, can be calculated analytically (note there are no self-return walks of odd length). It is described by Catalan numbers [118][2]: $C_n = \frac{1}{n+1}\binom{2n}{n}$. The weight of a walk of length $2n$ in a homogeneous chain is equal to r^{2n}. Hence the walk-sum is equal to

$$\phi_a(0 \to 0) = \sum_n C_n r^{2n} = \sum \frac{r^{2n}}{n+1}\binom{2n}{n} \tag{4.1}$$

Using the Stirling approximation for the factorial [22], $n! \approx \sqrt{2\pi n}\left(\frac{n}{e}\right)^n$, which is asymptotically accurate, we have:

$$\phi_a(0 \to 0) \approx \sum_n \frac{(4r^2)^n}{n^{3/2}\sqrt{\pi}} \tag{4.2}$$

The critical value is $r = \frac{1}{2}$, at which the sum $\sum_n \frac{1}{n^{3/2}\sqrt{\pi}}$ is convergent. Hence, at the critical value of r, as the single-sided chain grows, the self-return walk-sum at the root (the variance) does not keep growing without bound, but instead converges to a finite value. In fact, the expression for $\phi_a(0 \to 0)$ is precisely the generating function[3] for Catalan numbers [118], and can be computed analytically. In the region of convergence, for $r < \frac{1}{2}$ we have

$$\phi_a(0 \to 0) = \sum \frac{r^{2n}}{n+1}\binom{2n}{n} = \frac{1-\sqrt{1-4r^2}}{2r^2} \tag{4.3}$$

Infinite k-tree. Next we analyze a k-tree, where each node has k children, shown in Figure 4.2 (a). A k-tree is also not a computation tree for any finite graph (the neighborhood structure at the root-node is different from the rest of the nodes), but it will be used in the analysis of computation trees for $(k+1)$-regular graphs. To compute

[2]One interpretation of Catalan numbers is the number of ways to arrange n left parenthesis and n right parenthesis, such that the sequence is proper – i.e. '()' is proper ')(' is not. In our problem we have steps to the right and to the left instead of the parentheses.

[3]The generating function for a sequence a_n is the formal power series $G_a(r) = \sum a_n r^n$.

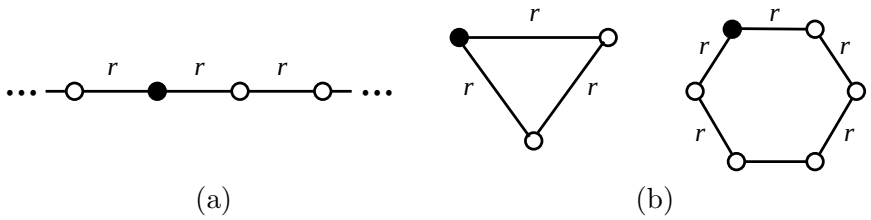


Figure 4.3. (a) Two sided infinite chain, with homogeneous weights r. The root node is filled. (b) example homogeneous cycles which generate the computation tree.

the number of self-return walks of length $2n$ starting from the root node in a k-tree we reduce the problem to the single-sided chain. We encode every step down into the tree, i.e. away from the root, as $+1$, and every step up, or towards the root, as -1. The number of such encodings is $\frac{1}{n+1}\binom{2n}{n}$ from our earlier calculation with one-sided chain. For every such encoding of ± 1 there are k^n possible walks in the k-tree: each step down can go along one of the k branches, whereas each step up is uniquely determined. Hence, the number of self-return walks of length $2n$ in a k-tree is $\frac{k^n}{n+1}\binom{2n}{n}$. Using a homogeneous weight r on all the edges, and the Stirling approximation for factorials this translates into the following approximate walk-sum:

$$\phi_b(0 \to 0) \approx \sum_n \frac{(4kr^2)^n}{n^{3/2}\sqrt{\pi}} \tag{4.4}$$

The critical value in comparison to the one-sided chain changes to $\frac{1}{2\sqrt{k}}$, but for this critical value the sum reduces to $\sum_n \frac{1}{n^{3/2}\sqrt{\pi}}$, similar to the one-sided chain, and the self-return walk-sums stay bounded. The generating function valid in the region of convergence is $\phi_b(0 \to 0) = \frac{1-\sqrt{1-4kr^2}}{2kr^2}$. At the critical value the k-tree behaves similarly to the one-sided chain.

Two-sided infinite chain. Next, we consider a two-sided infinite homogeneous chain, which *is* a computation tree for a homogeneous single-cycle with any number of nodes, see Figure 4.3. Thus, the self-return walk-sum at the root node in the two-sided chain is also the LBP variance estimate in the single-cycle model. The behavior at the critical value of r is different from the previous two examples. The number of self-return walks of length n is now $\binom{2n}{n}$, this is the number of ways to pick n pluses out of $2n$ pluses and minuses, with pluses and minuses corresponding to forward and reverse steps in the chain. This makes the walk-sum grow as

$$\phi_c(0 \to 0) \approx \sum_n \frac{(4r^2)^n}{n^{1/2}\sqrt{\pi}} \tag{4.5}$$

So, the critical value is $r = \frac{1}{2}$, and as $r \to \frac{1}{2}$, the LBP variance does increase without bound, because $\sum_n \frac{1}{n^{1/2}\sqrt{\pi}}$ diverges. So, for single homogeneous cycles at the critical

value of r, as the computation tree grows, the variances *do* increase without bound. The corresponding generating function for self-return walks in a two-sided infinite chain, which applies for $r < \frac{1}{2}$ is:

$$\phi_c(0 \to 0 \mid BT) = \frac{1}{\sqrt{1-4r^2}} \tag{4.6}$$

In agreement with our analysis, as $r \to \frac{1}{2}$, the expression increases without bound.

Computation trees for regular graphs. Finally, we consider $(k+1)$-regular graphs, where each node has $k+1$ neighbors. The computation tree for a $(k+1)$-regular graph looks almost like a k-tree, except at the root node, which has no parent, it has $k+1$ branches instead of k. See Figure 4.2 for an illustration with $k=2$. We can use the expression for walk-sums in k-trees to find the walk-sums in the computation tree for a $(k+1)$-regular graph. The root node has $k+1$ neighbors, each of which are the roots of k-trees. Using our recursive walk-sum update equations for trees from Chapter 3, the self-return walk-sum at the root node is equal to

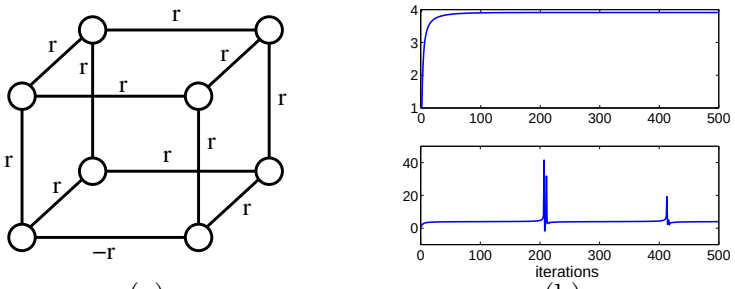
$$\phi_d(0 \to 0) = \frac{1}{1-(k+1)r^2\phi_d(1 \to 1\backslash 0)} = \frac{1}{1-(k+1)r^2\frac{1-\sqrt{1-4kr^2}}{2kr^2}} \tag{4.7}$$

where our expression for $\phi_d(1 \to 1\backslash 0)$, the walk-sum in the subtree rooted at the neighbor of 0, simply equals $\phi_b(0 \to 0)$ for k-trees. Note that all the three neighbors of the root node have the same subtrees, so we just use the expression for node 1 and add it $k+1$ times. Upon simplification, we have

$$\phi_d(0 \to 0) = \frac{2k}{k-1+(k+1)\sqrt{1-4kr^2}} \tag{4.8}$$

Apart from the case $k=1$ which is equivalent to the 2-sided chain, for critical values of $r = \frac{1}{2\sqrt{k}}$ the above expression is finite. Hence, for $k+1$ regular graphs with $k > 1$, as the computation tree at the critical value of r grows, the variances stay bounded.

We need to point out a potential pitfall in applying this analysis. Recall our remark about negating some of the edge-weights at the beginning of the section. In computation trees both the self-return walk-sums and the validity of the tree do not get affected by changing the signs of some or all of the edge-weights (see the proof of Proposition 3.2.1). However, for loopy graphs in general both are affected by changing signs. Thus the attractive k-regular graph (with all edge-weights positive) and the frustrated k-regular graph (in which some or all of the edge-weights may be negative) produce equivalent computation trees. It turns out that at the critical value of r the frustrated k-regular model with all but one edge-weight positive and equal to r, and one equal to $-r$, is valid. However, for $k > 2$, at the same value of r the attractive k-regular model with all edge-weights equal to r corresponds to an invalid information matrix J with negative eigenvalues. The attractive model is valid for smaller values of r, but, as r



(a) (b)

Figure 4.4. (a) 3-regular graph (a cube). Note that it is frustrated (one of the edges has weight $-r$). (b) LBP variances as a function of iteration. For $r = \frac{1}{2\sqrt{2}} - \epsilon$ variances converge (top), but for $r = \frac{1}{2\sqrt{2}} + \epsilon$, LBP fails (variances become negative and oscillate). Here, $\epsilon = 10^{-5}$.

increases, becomes invalid earlier than its computation tree. Hence, when investigating the critical value of the computation tree we must assume that the underlying k-regular graph is frustrated.

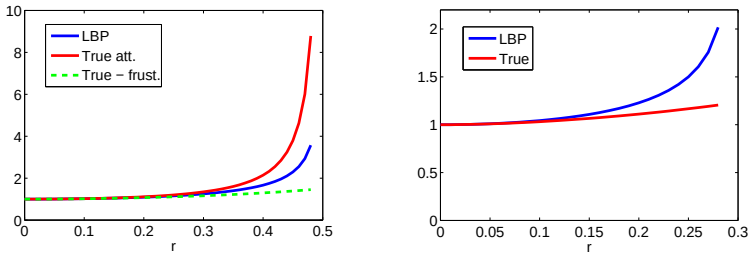
Example 1. In Figure 4.4 we show a frustrated 3-regular graph, a cube (note that $k = 2$, and $k+1 = 3$). The information matrix at the critical value $r = \frac{1}{2\sqrt{2}}$ is positive-definite with $\lambda_{\min} \approx 0.0421$. We apply LBP for $r = \frac{1}{2\sqrt{2}} - \epsilon$, just below our computed threshold, and for $r = \frac{1}{2\sqrt{2}} + \epsilon$, just above the threshold. In the first case LBP variances converge, but in the second case LBP fails – variances eventually become negative and start to oscillate. These observations agree with our computed critical value of r. Also note that as r approaches the critical value, the LBP variance estimate approaches a finite value 4, in agreement with (4.8), and does not explode to infinity.

In conclusion, the behaviour of LBP variances for critical computation trees with $\varrho_\infty = 1$ depends on the model: for some models the variances keep growing without bound, while for other models they converge to a finite value. Thus we leave the statement of Proposition 3.3.1 as is, and $\varrho_\infty = 1$ case has to be considered for each specific model.

■ 4.1.2 Assessing the Accuracy of LBP Variances

There is experimental evidence that far from the boundary of computation-tree validity, for $\varrho_\infty \ll 1$, LBP variances tend to approximate the true variances well. We show this for small regular graphs with homogeneous weights where both the LBP walk-sums and the complete variance walk-sums can be calculated analytically.

Consider an attractive 3-node cycle with all edge-weights set to $r > 0$. The corresponding computation tree is a two-sided infinite chain, see Figure 4.3. The expression



(a) (b)

Figure 4.5. LBP variances and true variances vs. r. (a) 3-node cycle. (b) Frustrated 4-clique.

we derived for $\phi_c(0 \to 0)$ in (4.6) is also the LBP variance estimate in the 3-node cycle, $\phi(0 \to 0 \mid BT)$, i.e. the walk-sum over backtracking walks in the 3-cycle. The expression for the correct variance, i.e. the walk-sum over *all* self-return walks, not just backtracking walks, can be obtained by inverting the 3×3 matrix $J = I - R = \begin{bmatrix} 1 & -r & -r \\ -r & 1 & -r \\ -r & -r & 1 \end{bmatrix}$. We have:

$$P_{ii} = \frac{1-r}{1-r-2r^2} \tag{4.9}$$

The same two-sided infinite chain is also equivalent to the computation tree of a frustrated 3-node cycle, with weights r, r and $-r$, as the edges in the computation tree can be negated without affecting the variances. The correct variance in this 3-node frustrated cycle is $P_i^f = \frac{1+r}{1+r-2r^2}$.

In Figure 4.5 (a) we plot the LBP variance estimate $\phi(0 \to 0 \mid BT)$ and the true variances P_{ii} and P_i^f for both the attractive and the frustrated model. We see that LBP underestimates the variances in the attractive model, and overestimates the variances in the frustrated model. For small values of r LBP gives a good approximation, while for values closer to $\frac{1}{2}$, LBP is very inaccurate. We expect to observe a similar behaviour for all models: when the model has $\varrho_\infty \ll 1$, short walks matter the most, and backtracking walks constitute a large fraction of short walks. However, when ϱ_∞ approaches 1, long walks start to have non-negligible weights, and since the fraction of back-tracking walks among long walks is much smaller, this leads to very inaccurate answers.

Accuracy of variances for fully-connected graphs. Next, we consider a frustrated[4] fully-connected graph with homogeneous weights: all edge-weights are set to $-r$. A fully-connected graph with $(k+2)$ vertices is the simplest $(k+1)$-regular graph. All $(k+1)$-regular graphs share the same computation tree, and in (4.8) we derived the expression for LBP variances, or equivalently, the self-return walk-sum at the root node in the

[4]Recall that the attractive fully-connected models is not valid at the critical value of r, as we discussed in Section 4.1.1.

computation tree:

$$\phi_d(0 \to 0) = \frac{2k}{k - 1 + (k+1)\sqrt{1 - 4kr^2}} \tag{4.10}$$

Now we compute the correct variances. The frustrated fully-connected homogeneous graph has information matrix $J = I - R$ with 1 on the diagonal, and r elsewhere ($R_{ij} = -r$ and $J_{ij} = r$). We can analytically calculate the variances[5]:

$$P_{ii} = \frac{1}{k+2}\left(\frac{k+1}{1-r} + \frac{1}{(k+1)r+1}\right) \tag{4.11}$$

In Figure 4.5 (b) we plot the LBP variances and the true variances for a 4-clique, i.e. $k = 2$. For small values of r the LBP answer is a good approximation, while for values closer to the critical, $r = \frac{1}{2\sqrt{2}}$, LBP (in the walk-sum form) noticeably overestimates the variance in this model.

■ 4.1.3 Finding the Expected Walk-sum with Stochastic Edge-weights

We next consider an application of combinatorics to approximately computing walk-sums in a probabilistic setting. While it may or may not result in a practical approach, it does bring up interesting ideas.

First we remark that combinatorial ideas can be used to compute not only the back-tracking LBP walk-sums, but also the correct walk-sums for the variance in certain regular loopy graphs. For example, in an infinite 2D lattice, the number of walks of length $2n$ that start and end at the origin equals $\binom{2n}{n}^2$. Hence, the correct variance in a homogeneous attractive lattice with edge-weight r is equal to $P_{ii} = \sum_n \binom{2n}{n}^2 r^{2n}$. When r is small, this expression can serve as a good approximation for nodes that are far from the boundary in a finite lattice.

Now suppose that we know the numbers of self-return walks of length n, but the edge-weights are non-homogeneous. Furthermore, suppose that the edge-weights can be modeled as i.i.d. random variables. Then we can make an approximation that each step in a walk is i.i.d.,[6] which allows us to calculate the expected walk-sum for self-return walks at node i. This approximation can be used to gauge the rough scale of the true self-return walk-sum in models where LBP variances are very inaccurate.

Suppose for simplicity that all edge-weights are positive. We model the log-weights as Gaussian with mean μ and variance σ^2, i.e. $\log(r_{ij}) \sim \mathcal{N}(\mu, \sigma^2)$. Since partial correlations r_{ij} are all bounded by 1 in magnitude, μ has to be negative. For a walk of

[5] The eigenvalues of this J matrix are $(k+1)r + 1$ and $k+1$ repeated copies of $1 - r$. Hence the eigenvalues of P are $\frac{1}{(k+1)r+1}$ and $k+1$ repeated copies of $\frac{1}{1-r}$. The trace of P is equal to $(k+2)P_{ii}$, since all the variances are the same. It is also the sum of the eigenvalues of P, i.e. $\mathrm{tr}(P) = \frac{k+1}{1-r} + \frac{1}{(k+1)r+1}$.

[6] Clearly, this is an approximation: the same edge may be traversed many times by a walk, while the probability that a real value appears twice in a finite sample of a continuous distribution is 0.

length n, the weight of the walk is the product of edge-weights. Hence

$$\log \mathbb{E}[\phi(w)] = \log \mathbb{E}[\ \prod_{e \in w} r_e\ ] = \sum_{e \in w} \log \mathbb{E}[r_e\ ] = \sum_{e \in w} \log \mathbb{E}[\exp(\log(r_e))] = \sum_{e \in w} \log \exp(\mu + \frac{\sigma^2}{2}) = n(\mu + \frac{\sigma^2}{2}) \quad (4.12)$$

Here we have used the moment-generating function of a normal random variable: if $x \sim \mathcal{N}(\mu, \sigma^2)$ then $\mathbb{E}[\exp(x)] = \exp(\mu + \frac{\sigma^2}{2})$.

Thus the expected weight of a walk of length n is $\mathbb{E}[\phi_n(w)] = \exp(n(\mu + \frac{\sigma^2}{2}))$. Given the number of self-return walks of length n and the expected weight of a walk of length n, we can calculate the expected walk-sum. For example, for our infinite 2D lattice $P_{ii} \approx \sum_n \binom{2n}{n}^2 \exp(n(\mu + \frac{\sigma^2}{2}))$. Recall that μ is negative, so for μ negative enough and σ small enough, the expression converges. Also, by setting $\sigma = 0$ we recover the expression for walk-sums in the homogeneous deterministic setting, with $r = \exp(\mu)$.

■ 4.2 Vector-LBP and Vector Walk-summability

We now consider Gaussian graphical models with vector variables and later, in Section 4.3, models defined with respect to a factor graph. It is true that the scalar pairwise form of Gaussian models is sufficient to describe all possible Gaussian models, so no richness in representation is lost by limiting to that case. The main reason to consider vector and factor graph representations of Gaussian models is that the corresponding versions of LBP can have much stronger convergence properties and provide much more accurate variances compared to scalar LBP, at the cost of more computational complexity. For example, in the extreme case of grouping all the variables into one node, or grouping enough variables together into blocks such that they form a tree, there are no convergence issues as BP on trees is exact and terminates in a finite number of steps. This is essentially the same idea as the junction tree algorithm. Of course, when the number of variables is large, this is not feasible because the computational complexity will be too high. Hence, a compromise which involves loopy graphs with blocks of variables may be necessary. We will see that there are many parallels between the vector and scalar walk-sum analysis, but there are also significant differences, and the story is much more involved. We now consider vector-LBP, and later in Section 4.3 move on to factor graph LBP and walk-summability.

■ 4.2.1 Defining Vector Walk-summability

In the vector representation we have a graph $G = (V, \mathcal{E})$ where each node $i \in V$ corresponds to a vector variable x_{α_i} for some $\alpha_i \subset \{1, .., N\}$. The sets α_i form a partition of the scalar variables, and J is a block-matrix. The graph G can be thought of as a coarsening of the graph in the scalar case of Chapter 3, which has a node for every scalar variable. For $i, j \in V$ we use the notation $J_{ij} \triangleq J_{\alpha_i, \alpha_j}$ and $h_i \triangleq h_{\alpha_i}$. An example

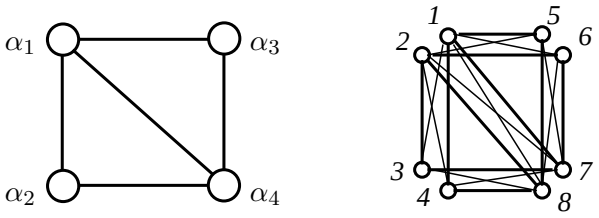


Figure 4.6. (left) Graph of a model with vector-variables, and (right) the graph of the corresponding scalar model.

appears in Figure 4.6 – by blocking the variables of the complicated scalar model (right plot) into pairs $\alpha_1 = \{1, 2\}$, $\alpha_2 = \{3, 4\}$, $\alpha_3 = \{5, 6\}$, and $\alpha_4 = \{7, 8\}$, the corresponding vector-model is a 4-cycle with a chord (left plot).

Roadmap for vector-WS. For a vector model there is a rich class of possible walk-sum decompositions. We first assume that J is normalized to have identities on the diagonal blocks, the matrix[7] $R = I - J$ has zero diagonal blocks, and the corresponding graph G has no self loops. We show that such normalization is not unique, and also that, while vector walk-summability does not depend on the choice of normalization, sufficient conditions for it do. We also consider transformations of the variables in the model (or not normalizing the model) to give other decompositions $J = I - R$ where the matrix R has non-zero diagonal blocks, and the graph G has self-loops. We define a notion of generalized vector walk-summability that takes such transformations into account. We study the relationship between vector and generalized vector walk-summability, and their implications for the vector version of LBP – while we make significant progress in this regard, these notions are inherently more complex than the scalar ones, and several questions remain open.

Let $R = I - J$, and define $R_{ij} \triangleq R_{\alpha_i,\alpha_j}$. For this section we assume that J has been normalized to have its block-diagonal elements corresponding to blocks $\{\alpha_i\}$ equal to identity, and diagonal blocks R_{ii} are zero. This is easily accomplished via the Cholesky decomposition of block-diagonal elements[8], but unlike in the scalar case, the normalization is *not* unique, and any $\tilde{J} = QJQ^T$ with block-orthogonal $Q = \text{blockdiag}(Q_i)$, $Q_i^T Q_i = I$, also has its diagonal blocks equal to identity.

[7] Without loss of generality, we only consider decompositions of the form $J = I - R$, instead of the more general ones based on $J = D - K$, with D block-diagonal. Please see a discussion in Section A.2.1 of Appendix A.

[8] The block-diagonal entries J_{ii} are positive-definite, hence we can use Cholesky decomposition: $J_{ii} = L_i L_i^T$ where $L_i \succ 0$. Let $L = \text{blockdiag}(L_i)$ for $i \in V$. Applying L^{-1} whitens the block-diagonal elements of $\tilde{J} = L^{-1} J L^{-T}$ as $L_i^{-1} J_{ii} L_i^{-T} = I$. Now $\tilde{J}$ has identities on the diagonal and $\tilde{R} = I - \tilde{J}$ has zero-blocks on the diagonal.

We assign matrix edge-weights R_{ij} (possibly non-symmetric and in general non-square) to each edge $\{i, j\}$ of G. With our normalization the block-diagonal entries of R are zero, and there are no self-loops (i, i). To each walk w we assign a weight $\phi(w)$ to be the product of the edge-weight matrices in the order of traversal of the walk (the product is formed from left to right):

$$\phi(w) = \prod_{k=1}^{l(w)} R_{w_{k-1},w_k} \tag{4.13}$$

Hence, $\phi(w)$ is itself a matrix. For a walk w of length 0, the weight is $\phi(w) = I$, an identity matrix (instead of 1 in the scalar case). A walk-sum $\phi(\mathcal{W})$ for a set of walks is defined exactly as in Chapter 3; it is the sum of walk weights over the walks in the set, $\phi(\mathcal{W}) = \sum_{w\in\mathcal{W}} \phi(w)$. It is also a matrix. Provided that these walk-sums are well-defined, as we describe next, we also have the walk-sum interpretation for covariances, and the means:

$$P_{ii} = \sum_{w:i\to i} \phi(w), \quad \text{and} \quad \mu_i^T = \sum_{w:*\to i} h_*^T \phi(w) \tag{4.14}$$

where P_{ii} is a block of the covariance matrix corresponding to α_i, and $\mu_i \triangleq \mu_{\alpha_i}$.

Definition: Vector-WS. We call a vector model with blocks $\{\alpha_i\}$ and with information matrix $J = I - R$ *vector walk-summable* if for all pairs of vertices i and j the absolute walk-sum $\bar{\phi}(i \to j) \triangleq \sum_{w\in\mathcal{W}(i\to j)} |\phi(w)|$ converges, and hence $\phi(i \to j)$ is independent of the order of summation of the walks. The absolute value $|\phi(w)|$ is taken element-wise. We stress that $\bar{\phi}(i \to j) \geq \left|\sum_{w\in\mathcal{W}(i\to j)} \phi(w)\right|$, and equality *does not* hold in general. This is an abstract definition of walk-summability, and we are interested in relating it to an easily testable condition (recall that in the scalar case walk-summability exactly corresponds to $\varrho(\bar{R}) \leq 1$ by Proposition 3.1.1). We will see in this chapter that the story for vector walk-summability is more complicated, and there may not be a simple characterization. Instead we develop a family of sufficient conditions.

■ 4.2.2 Sufficient Conditions for Vector Walk-summability

Our first sufficient condition comes from considering the relationship between vector and scalar walk-summability. To any vector model on $G = (V, \mathcal{E})$ with blocks α_i there corresponds a scalar model on $G^s = (V^s, \mathcal{E}^s)$ which has a node for every scalar variable. A node i in the vector model corresponds to a group of nodes $i^s \in \alpha_i$ in the scalar model. Suppose that the vector model is specified by $J = I - R$. Then the same matrix $J = I - R$ also defines a scalar model on G^s. If this scalar model is scalar walk-summable, i.e. $\varrho(\bar{R}) < 1$, then the vector model is vector walk-summable – i.e. scalar walk-summability is a sufficient (but not in general necessary) condition for vector walk-summability:

Lemma 4.2.1 (Scalar and vector WS). *Suppose that $J = I - R$ is scalar walk-summable on G^s. Group the variables into blocks $\{\alpha_i\}$ and define the vector model with the same matrix R on G. Then the vector model is vector walk-summable.*

Proof. The walk-sum $\phi(i \to j \mid G)$ in the vector model is a matrix. Its entries correspond to walk-sums in the scalar model $\phi(i^s \to j^s \mid G^s)$ where $i^s \in \alpha_i$, and $j^s \in \alpha_j$. Vector walk-summability requires convergence of $\sum_{w\in\mathcal{W}(i\to j)} \bar{\phi}(w) = \sum_{w\in\mathcal{W}(i\to j)} |\prod_{e\in w} R_e|$, with absolute values taken elementwise. Scalar walk-summability condition is equivalent to convergence of $\sum_{w\in\mathcal{W}(i^s\to j^s)} \prod_{e\in w} \bar{R}_e$, because the matrix sum converges if and only each of its entries converge. Now, for any walk w, we have $\prod_{e\in w} \bar{R}_e \geq |\prod_{e\in w} R_e|$, so vector walk-summability is implied by scalar walk-summability. □

In the vector case an important role will be played by transformations of the variables: now we are not limited to mere scaling the variables but also can consider block-diagonal transformation conforming to the sizes of the blocks α_i. Such transformations leave the Markov structure of the vector model the same. We now consider block-orthogonal transformations $Q = \text{blockdiag}(Q_i)$, with $Q_i^T Q_i = I$, and show that they do not affect vector walk-summability:

Lemma 4.2.2. *Take any block-orthogonal matrix $Q = blockdiag(Q_1, ..., Q_{|V|})$, where Q_i are compatible with the blocks α_i (i.e. have size $|\alpha_i|$), and $Q_i^T Q_i = I$. Then $\tilde{J} = QJQ^T$ is vector walk-summable if and only if J is.*

Proof. Since $\tilde{J} = QJQ^T$, we also have $\tilde{R} = I - \tilde{J} = QRQ^T$. Take any walk w and consider $\phi(w) = R_{i_1,i_2} R_{i_2,i_3} R_{i_3,i_4} ... R_{i_{n-1},i_n}$. With an orthogonal transformation of the blocks we have $\tilde{R}_{ij} = Q_i R_{ij} Q_j^T$. Hence the corresponding walks in the transformed model has weight $\tilde{\phi}(w) = \tilde{R}_{i_1,i_2} ... \tilde{R}_{i_{n-1},i_n} = Q_{i_1} R_{i_1,i_2} Q_{i_2}^T Q_{i_2} R_{i_2,i_3} ... Q_{i_{n-1}}^T Q_{i_{n-1}} R_{i_{n-1},i_n} Q_{i_n}^T = Q_{i_1} \phi(w) Q_{i_n}^T$. Now $\bar{\tilde{\phi}}(i \to j) = \sum_{w\in\mathcal{W}(i\to j)} \overline{Q_i \phi(w) Q_j^T} \leq \bar{Q}_i \left(\sum_{w\in\mathcal{W}(i\to j)} \bar{\phi}(w) \right) \bar{Q}_j^T$. Hence, the series $\bar{\tilde{\phi}}(i \to j)$ converges if $\bar{\phi}(i \to j) = \sum_{w\in\mathcal{W}(i\to j)} \bar{\phi}(w)$ converges. The converse follows because we also have $R = Q^T \tilde{R} Q$. □

Since arbitrary block-orthogonal transformations do not change vector walk-summability, we get a tighter sufficient condition extending Lemma 4.2.1:

Lemma 4.2.3 (Sufficient condition for vector-WS). *Let $\mathcal{Q}_G$ be the set of block-orthogonal matrices on G with respect to blocks $\{\alpha_i\}$, i.e. $Q = blockdiag(Q_i)$ for $i \in V$, $Q_i Q_i^T = I$. Then $\min_{Q\in\mathcal{Q}_G} \varrho(\overline{QRQ^T}) < 1$ is a sufficient condition for vector walk-summability.*

Proof. Block-orthogonal change of variables does not affect vector walk-summability. For each $Q \in \mathcal{Q}_G$ the condition $\varrho(\overline{QRQ^T}) < 1$ is sufficient for vector walk-summability of QRQ^T and of R. Taking the minimum over all $\mathcal{Q}_G$ our condition follows. □

An interesting but hard question is whether $\min_Q \varrho(\overline{QRQ^T}) < 1$ is also necessary for vector walk-summability. We have not yet answered it. We describe the difficulties with characterizing the notion of vector walk-summability next.

K-step conditions So far, we have developed conditions based on scalar walk-summability which in turn implied vector walk-summability. The novelty beyond the scalar case has been the freedom to optimize over various transformations. Now we consider conditions which go beyond scalar walk-summability.

Vector walk-summability is an inherently more complicated condition to check. The convenient property of scalar walk-sums $\bar{\phi}(w) = |\prod_{e\in w} r_e| = \prod_{e\in w} |r_e|$ does not carry over to the vector case as $|\prod_{e\in w} R_e| \neq \prod_{e\in w} |R_e|$. Absolute convergence of vector walk-sums does not correspond to the spectral radius of a matrix: neither $\bar{R}^k$ nor $\overline{R^k}$ correspond to absolute walk-sums $\sum_{w\in\mathcal{W}(i\xrightarrow{k}j)} |\phi(w)| = \bar{\phi}(i \xrightarrow{k} j)$ over length-k walks[9]. Denote the matrix of absolute walk-sums of length k by $\bar{\phi}_k$: the (i,j)-th block of $\bar{\phi}_k$ contains $\bar{\phi}(i \xrightarrow{k} j)$. We have $\overline{R^k} \leq \bar{\phi}_k \leq \bar{R}^k$. The scalar walk-summability condition $\varrho(\bar{R}) < 1$ is sufficient for vector walk-summability, but it can be loose. However, we can take advantage of the following k-step expansion:

$$\sum_{l=0}^{\infty} R^l = (I + R + ... + R^{k-1}) \sum_{l=0}^{\infty} (R^k)^l \tag{4.15}$$

which converges if $\varrho(R) < 1$. Now if $k = l+m$, then $R^k = R^l R^m$, and $\bar{\phi}_k \leq \bar{\phi}_l \bar{\phi}_m$. Apply the k-step expansion to vector walk-summability: $\sum_l \bar{\phi}_l \leq (I+\bar{\phi}_1+..+\bar{\phi}_{k-1}) \sum_l (\bar{\phi}_{kl}) \leq (I+\bar{\phi}_1+..+\bar{\phi}_{k-1}) \sum_l (\bar{\phi}_k)^l$. Convergence of the first series is implied by the convergence of the second, which in turn depends on $\varrho(\bar{\phi}_k) < 1$. This gives a family of sufficient conditions for vector walk-summability based on $\varrho(\bar{\phi}_k)$. For $k = 1$ it recovers our first sufficient condition for vector walk-summability[10] based on $\varrho(\bar{R})$ because $\varrho(\bar{\phi}_1) = \varrho(\bar{R})$. As $k \to \infty$ the condition becomes tight – in essence it reduces to calculating absolute vector walk-sums:

Lemma 4.2.4 (K-step sufficient conditions for vector WS). *If $\varrho(\bar{\phi}_k) < 1$ for some k then the model is vector walk-summable.*

We could also consider K-step conditions after a block-orthogonal transformation Q, which, for a good choice of Q, may give tighter sufficient condition. Next, we shift gears, and define generalized walk-summability, and consider its implications for scalar and vector models.

[9]Note the difference from $\left|\sum_{w\in\mathcal{W}(i\xrightarrow{k}j)} \phi(w)\right| \leq \bar{\phi}(i \xrightarrow{k} j)$. Equality does not hold in general. To calculate $\bar{\phi}(i \xrightarrow{k} j)$ in the vector case we need to find the weight of *each individual* walk, take its absolute value, and add to the sum. Thus the absolute walk-sum is only computable for small values of k, as the computation (the number of walks) grows exponentially with k. We stress that neither $\overline{R^k}$ nor $(\bar{R})^k$ compute this absolute walk-sum.

[10]In the scalar case all k-step conditions simply reduce to $\varrho(\bar{R}) < 1$, because there it does hold that $\bar{\phi}_k = \bar{R}^k$.

Generalized Vector Walk-summability. Let us consider a class of transformations that is richer than the set of block-orthogonal transformations: let S be a block-invertible matrix $S = \text{blockdiag}(S_i)$, with blocks $\{\alpha_i\}$, $\det(S_i) \neq 0$. Define $\tilde{x} = S^{-T}x$, then $\tilde{J} = SJS^T$. Here J may or may not be normalized to have diagonal blocks equal to identity matrices – this does not matter because the class of block-invertible matrices contains the normalizing transformations as special cases. Consider the power-series expansion of $\tilde{J}$, by defining $\tilde{R} = I - \tilde{J} = I - SJS^T$:

$$\tilde{J}^{-1} = \sum_k \tilde{R}^k = \sum_k (I - SJS^T)^k. \tag{4.16}$$

Here the matrix $\tilde{R}$ does not have the interpretation of partial correlation coefficients, $\tilde{J}$ need not be unit-diagonal, and the matrix $\tilde{R}$ may have non-zero entries in its diagonal blocks. We need to slightly extend our walk-sum representation to accommodate these non-zero diagonal blocks: we now introduce self-loops with weights $\tilde{R}_{ii}$, and allow walks to make steps (i, i) which follow the self-loop. With this provision, the definitions of weights of the walks and walk-sums remain the same as in Section 4.2.1, and the walk-sum interpretation of BP in trees and of LBP both carry through with self loops. We discuss this in Section A.2.1 of Appendix A. Note that the convergence of vector walk-sums based on the expansion in (4.16) will in general depend on S – they may converge for some choices of S and diverge for others. This leads us to define generalized vector walk-summability:

Definition: Generalized WS. If there exists an S such that the expansion based on $\tilde{R} = I - SJS^T$ is vector walk-summable, then we call J *generalized vector walk-summable*.

We could in principle also consider generalized scalar walk-summability, by allowing non-unit diagonal scaling DJD and matrices $\tilde{R} = I - DJD$ with non-zero diagonals. The necessary and sufficient condition for absolute convergence of the corresponding scalar walk-sums is $\varrho(\overline{I - DJD}) < 1$. In Section A.2.1 of Appendix A we show that the canonical zero-diagonal decomposition (where D is simply identity) gives the tightest sufficient condition in the scalar case, obviating the need for generalized scalar walk-summability. However, in the vector case this does not appear to be true: in Section 4.2.5 we present numerical evidence that a model which does not satisfy $\varrho(\overline{QRQ^T}) < 1$ for any choice of block-orthogonal Q may still be generalized walk-summable. Another numerical study in Section 4.2.5 shows that for a block-tree model (the vector-variables form a tree) $\varrho(\overline{QRQ^T}) < 1$ may not be satisfied, while generalized walk-summability holds; also vector pairwise-normalizability (to be defined shortly in Section 4.2.4) is more closely related to generalized vector walk-summability, rather than ordinary vector walk-summability. Hence, generalized vector walk-summability appears to be a more fundamental notion. Next, we relate ordinary and generalized vector walk-summability to convergence of vector-LBP.

■ 4.2.3 Vector-LBP.

Vector LBP involves passing messages between nodes that represent vector variables. These messages can be parameterized in information form similar to the scalar case (2.28), but with $\Delta h_{i\to j}$ being a vector and $\Delta J_{i\to j}$ a matrix. The message update equations in (2.29-2.30) apply to the vector case without any changes. The computation tree interpretation in Section 2.2.3 still applies for the vector case, with the understanding that its nodes correspond to vector variables and the edges have vector weights. We now apply vector walk-sum analysis to have more insight into the behavior of vector-LBP on graphs with loops.

As scalar walk-summability is a sufficient condition for the scalar version of LBP to converge, similarly, we now show that vector walk-summability is a sufficient condition for the convergence of vector-LBP. Throughout this section we assume the walk-sum potential decomposition for LBP in (2.24) with the understanding that now $J_{ii} = J_{\alpha(i),\alpha(i)}$, and $J_{ij} = J_{\alpha(i),\alpha(j)}$. For the most part our proofs for the scalar case generalize to the vector case, but some of the details are quite different.

In the scalar case the weight of a back-tracking self-return walk is positive, so one may conjecture that in the vector case the weight of a back-tracking self-return walk has to be positive semi-definite. This is not true. In fact the weight does not even have to be symmetric. Take a self-return walk w and the reverse walk $-w$, which traverses the same steps as w but in the opposite direction. Then, even $\phi(w) + \phi(-w)$ does not have to be positive-semi-definite. As a simple example, the walk $w = (i, j, i, k, i)$ has weight $\phi(w) = R_{ij}R_{ji}R_{ik}R_{ki}$ and $\phi(w) + \phi(-w)$ is not in general positive-semi-definite, and can have negative eigenvalues. However, when a new node is added to the computation tree, the sum over all new self-return walks at the root node does in fact have a positive-semi-definite weight (proof is in Appendix A):

Lemma 4.2.5 (Monotonicity of LBP covariances). *As the LBP computation tree grows, the self-return walk-sums at the root node increase monotonically in the positive-definite sense.*

This allows us to prove that vector-WS is sufficient for convergence of vector-LBP for both the covariances and the means (the proof appears in Appendix A):

Proposition 4.2.1 (LBP in vector-WS models). *Vector-LBP converges in vector-WS models: LBP means converge to correct mean estimates, and LBP covariances converge to a walk-sum over back-tracking self-return walks.*

A non-orthogonal transformation of the variables may change the convergence of the walk-sum decomposition: a non-walk-summable decomposition may become walk-summable, and vice versa. We now show that arbitrary block-invertible transformations do not affect convergence and estimates of vector-LBP: if we rescale the variables $\tilde{x} = S^{-T}x$ and perform vector-LBP, this is equivalent to first performing vector-LBP and then rescaling the estimates by S^{-T}.

Lemma 4.2.6 (Invariance of LBP to block transformations). *Consider a block-invertible transformation $S = \text{blockdiag}(S_i)$, with $i \in V$, and S_i invertible. Then, vector-LBP converges for the model SJS^T if and only if it converges for J, and the LBP covariance estimates for SJS^T are scaled versions of the ones for vector-LBP on J.*

Proof. Consider an n-step LBP computation trees for models J and $\tilde{J} = SJS^T$, with the information matrices $J^{(n)}$ and $\tilde{J}^{(n)}$, respectively. One is obtained from the other by $\tilde{J}^{(n)} = S^{(n)}J^{(n)}(S^{(n)})^T$, where $S^{(n)} = \text{blockdiag}(S_i)$ for $i \in V^{(n)}$ (the vertices of the computation tree). Now the covariance estimate at the root node of the computation tree $\tilde{T}^{(n)}$ is equal to $\tilde{P}_0 = [S^{-1}J^{-1}S^{-T}]_0 = S_0^{-1}[J^{-1}]_0 S_0^{-T} = S_0^{-1}P_0 S_0^{-T}$. Here P_0 is the root-node covariance estimate in the computation tree for the model J. Thus as $n \to \infty$, $\tilde{P}_0$ converges if and only if P_0 converges, and the covariance estimate in the scaled model is a scaled version of the covariance estimate in the original model. □

Using this invariance we obtain a tighter sufficient condition for convergence of LBP:

Lemma 4.2.7 (Sufficient condition for vector-LBP convergence). *Let $\mathcal{S}_G$ be the set of block-invertible transformations G. Then $\min_{S\in\mathcal{S}_G} \varrho(\overline{I - SJS^T}) < 1$ is a sufficient condition for vector LBP convergence.*

Proof. To each invertible transformation S there is a corresponding walk-sum decomposition: $R = I - SJS^T$. If this R matrix is scalar walk-summable then it is vector walk-summable and hence vector-LBP converges. We take the tightest sufficient condition over all such walk-sum decompositions. □

■ 4.2.4 Connection to Vector Pairwise-normalizability.

We have seen the importance of the notion of pairwise-normalizability (PN) in the scalar case in Chapter 3. The definition of pairwise-normalizability extends seamlessly to the vector case: a model is vector-PN if

$$J = \sum_{e\in\mathcal{E}} [J_e], \quad \text{with } J_e \succ 0 \tag{4.17}$$

Here the edges $\mathcal{E}$ correspond to the vector model with blocks $\{\alpha_i\}$, and J_e are the corresponding off-diagonal blocks of J. Recall that $[J_e]$ represents J_e zero-padded to match the size of J. The strength of vector-PN, which measures how much the model can be perturbed while still satisfying vector-PN, is defined as

$$\epsilon_{\max}^{vec} \triangleq \max\{\epsilon \mid J = \epsilon I + \sum_{e\in\mathcal{E}} [J_e], \quad J_e \succeq 0\} \tag{4.18}$$

The superscript *vec* stands for 'vector', and distinguishes vector-PN strength $\epsilon_{\max}^{vec}$ from the scalar one $\epsilon_{\max}$.

In the scalar case we have seen the equivalence of PN and walk-summability (WS) in Proposition 3.1.8, and it even holds that the strength of PN is equivalent to the strength of WS: $\epsilon_{\max} = 1 - \varrho(\bar{R})$, see Corollary 3.1.3. For the vector case we have not related vector-WS and vector-PN directly, but we can show that sufficient conditions for vector-WS imply vector-PN:

Lemma 4.2.8 (Vector walk-summability and vector-PN). *Let $\mathcal{Q}_G$ and $\mathcal{S}_G$ be the sets of block-orthogonal, and block-invertible matrices with respect to blocks α_i. If $\min_{Q\in\mathcal{Q}_G} \varrho(\overline{QRQ^T}) < 1$ then the model is vector-PN. If $\min_{S\in\mathcal{S}_G} \varrho(\overline{I - SJS^T}) < 1$, then the model is vector-PN. Also, $\epsilon^{vec}_{\max} \leq 1 - \min_{Q\in\mathcal{Q}_G} \varrho(\overline{QRQ^T})$.*

The proof appears in Appendix A, it is based on the fact that if a model is vector-PN then it is vector-PN after any block-invertible transformations, and also that if J is scalar-PN, then it is also vector-PN. We do not know whether the converse holds, i.e. that vector-PN implies vector-WS, but in Section 4.3.3 we will see that vector-PN by itself guarantees convergence of vector-LBP covariances. Next, we show that block-trees (trees where each variable is a vector) are guaranteed to be vector-PN:

Lemma 4.2.9 (Block-trees are vector-PN). *A valid block-tree model is vector pairwise-normalizable.*

Proof. We use the directed factorization which holds for tree-structured models: $p(x) = p(x_1)\prod_i p(x_i \mid x_{\pi(i)})$. Consider any pair $(x_{\pi(i)}, x_i)$, and rename these variables x_A and x_B for convenience. Then the joint marginal $p(x_A, x_B)$ is given by

$$p(x_A, x_B) \propto \exp\left\{-\frac{1}{2}\begin{pmatrix} x_A & x_B \end{pmatrix}\begin{pmatrix} J_A & J_{A,B} \\ J_{B,A} & J_B \end{pmatrix}\begin{pmatrix} x_A \\ x_B \end{pmatrix}\right\} = \tag{4.19}$$

$$\exp\left\{-\frac{1}{2}x_A^T\hat{J}_A x_a\right\}\exp\left\{-\frac{1}{2}\begin{pmatrix} x_A & x_B \end{pmatrix}\begin{pmatrix} J_{A,B}J_B^{-1}J_{B,A} & J_{A,B} \\ J_{B,A} & J_B \end{pmatrix}\begin{pmatrix} x_A \\ x_B \end{pmatrix}\right\} \tag{4.20}$$

where the two terms in the second line are $p(x_A)$ and $p(x_B \mid x_A)$. The quadratic form in the second term in the second line is positive-semi-definite (p.s.d.), as $J_B \succ 0$ and the Schur complement $J_{A,B}J_B^{-1}J_{B,A} - J_{A,B}J_B^{-1}J_{B,A} = 0$ is trivially p.s.d. Hence $p(x_B \mid x_A)$ is based on a quadratic form with a p.s.d. matrix. Since this holds for all $p(x_i|x_{\pi(i)})$, and $\hat{J}_1 \succ 0$, we have a pairwise-normalizable decomposition. □

This lemma generalizes the scalar case, where we have seen that valid trees are both PN and walk-summable. Interestingly, based on numerical evidence in Section 4.2.5, the block-orthogonal sufficient condition $\min_{Q\in\mathcal{Q}_G} \varrho(\overline{QRQ^T}) < 1$ may not be satisfied for tree-structured models. However, again through numerical experiments, it appears that for valid trees it holds that $\min_{S\in\mathcal{S}_G} \varrho(\overline{I - SJS^T}) < 1$ with S block-invertible, thus illustrating the importance of generalized vector walk-summability. Based on simultaneous diagonalization of a pair of matrices it is easy to establish a much simpler statement that any valid model consisting of *a pair* of vector nodes is vector walk-summable (proof appears in Appendix A):

Lemma 4.2.10 (Walk-summability of two-node vector models). *Any vector model with $J \succ 0$ with 2 vector nodes is vector walk-summable.*

Unfortunately, in general it is impossible to jointly diagonalize more than two variables simultaneously, so the question of whether block-trees are vector-WS or generalized vector-WS, and their exact relation to vector-PN, remains unresolved. Next we investigate this and related questions through numerical experiments.

■ 4.2.5 Numerical Studies.

In this section we study vector-LBP and vector walk-summability on numerical experiments. We first show that using vectorization can greatly improve LBP performance. Then we conduct numerical studies to get intuition for the relationship among the different conditions for vector walk-summability, and to provide numerical evidence for various conjectures made in Section 4.2.

Example 1. (Vector LBP convergence and accuracy). We consider a thin-plate model on a 120×120 grid, with small regularization $0.2I$ added to make the model non-singular. We have $\lambda_{\min}(J) = 0.2$. However, the model is severely non-walk-summable with $\varrho(\bar{R}) \approx 1.7507$. Scalar LBP fails - it quickly starts to oscillate and produce negative variances. We use the vector version of LBP, grouping the variables into $L \times L$ blocks, for $L \in \{2, 3, 4, 6, 8, 12\}$. Vector LBP converges for each of these block sizes. The norm of the errors in variances and the number of LBP iterations to converge to some specified tolerance is displayed in Figure 4.7 (a) and (b) as a function of L. We see that accuracy in variances improves consistently with larger blocks, while the number of iterations improves at first and then levels out. In Figure 4.7(c) we display the errors in variances spatially: these are mostly negligible except where groups of 4 blocks meet. In these locations some of the non-backtracking walks not captured by vector LBP are not negligible. To get uniformly accurate variances we could offset the blocks by $L/2$ such that we get accurate variances in the troublesome regions, and use them in place of the first set of estimates. At any rate, it is clear that vector-LBP provides significant advantages over scalar LBP which simply fails for this example.

Example 2 (a). Next, we study the relationship between the block-orthogonal sufficient condition $\min_Q \varrho(\overline{QRQ^T}) < 1$, the block-invertible condition $\min_S \varrho(\overline{I - SJS^T}) < 1$ and pairwise-normalizability. We use the 4-cycle with a chord shown in Figure 4.6. Each vertex in the graph corresponds to a vector of size 2; the blocks are $\{1, 2\}$, $\{3, 4\}$, $\{5, 6\}, \{7, 8\}$. The graph over the blocks along with its original scalar variable version appear in the figure. We pick a positive-definite unit-diagonal information matrix $\tilde{J}$ (the pairwise potentials are picked randomly and the matrix is then normalized). The sufficient condition for scalar walk-summability is not satisfied for $\tilde{J} = I - \tilde{R}$: $\varrho(\overline{\tilde{R}}) \approx 1.2117 > 1$. The model is far from being walk-summable, and scalar LBP quickly starts to oscillate producing negative variances, and diverges for the means.

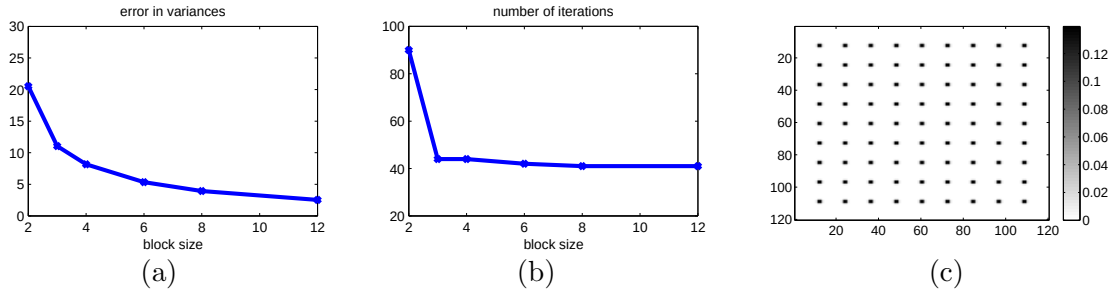


Figure 4.7. Vector-LBP performance on a 120×120 thin-plate model. (a) Norm of error in variances and (b) number of LBP iterations as a function of block-size. (c) Errors in variances on a 2D grid for block-size 12: errors are mostly negligible except at the junction of 4 blocks.

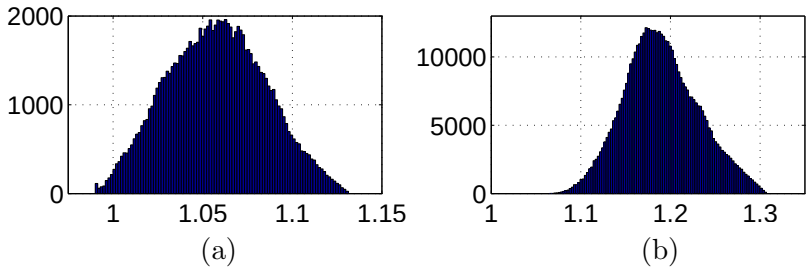


Figure 4.8. Histograms of $\varrho(\overline{QRQ^T})$ for example 2(a) and 2(b). In (a), the histogram extends below 1, showing that the model is vector-WS. In (b) the minimum value is well above 1, hence the sufficient condition is not informative. However, both models are vector-PN with $\epsilon^{vec}_{\max} = 0.0100$.

We apply block-whitening transformations so that $J = T\tilde{J}T^T$ not only has unit diagonal, but also has identity matrices in each diagonal block. The matrix $R = I - J$ has zeros on the block diagonal. As the whitening transformation is not unique (it is invariant to block-orthogonal transformations) we investigate the dependence of $\varrho(\overline{QRQ^T})$ on the choice of Q. In Figure 4.8, left plot, we show a histogram of values of $\varrho(\overline{QRQ^T})$, with a uniform random sampling of block-orthogonal matrices[11]. The distribution of $\varrho(\overline{QRQ^T})$ over 100000 random samples of block-orthogonal Q is very wide, with some rare choices of Q making the model walk-summable. The minimum value of over the samples is $\varrho(\overline{QRQ^T}) \approx 0.990$, which is in fact the optimum for the problem, as the vector pairwise-normalizability index in (4.18) is $\epsilon^{vec}_{\max} = 0.0100$ for this model. From Lemma 4.2.8 we have seen that $\epsilon^{vec}_{\max} \geq 1 - \min_Q \varrho(\overline{QRQ^T})$. Thus the model is vector walk-summable, and vector LBP converges. We also use local optimization[12] over block-invertible matrices S with 100 random starting points (to combat non-convexity) to find a matrix S with $\varrho(\overline{I - SJS^T}) \approx 0.990$. Hence, there

[11] To get uniform sample from the set of orthogonal matrices (for each block) we use the QR decomposition of a random Gaussian matrix, and adjust the signs. This can be done by the following matlab code: $[\mathtt{Q},\mathtt{R}] = \mathtt{qr}(\mathtt{randn}(\mathtt{L}))$; $\mathtt{Q_s} = \mathtt{Q} * \mathtt{diag}(\mathtt{sign}(\mathtt{diag}(\mathtt{R})))$;.

[12] Sampling over block-invertible matrices is not realistic as the set is high-dimensional and non-compact.

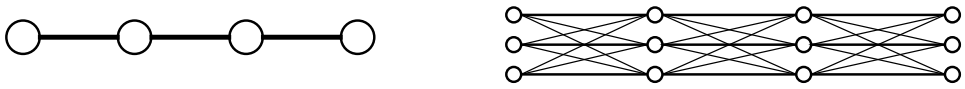


Figure 4.9. (left) A chain with vector variables and (right) the graph of the corresponding scalar model.

is numerical evidence that for this problem $\min_Q \varrho(\overline{QRQ^T}) = \min_S \varrho(\overline{I - SJS^T}) = 1 - \epsilon^{vec}_{\max}$. Unfortunately, as we see next, this may not hold in general.

Example 2 (b). We next repeat this example with the same graph but with another choice of a unit-diagonal positive-definite matrix $\tilde{J}$, which has $\varrho(\overline{\tilde{R}}) \approx 1.4244 > 1$. We again block-whiten this matrix to get R with zero diagonal blocks, and plot a histogram of $\varrho(\overline{QRQ^T})$ over 500000 uniform samples of block-orthogonal Q in Figure 4.8, right plot. None of the samples fall below 1 – in fact the minimum value is 1.06523. However, this model is also vector-PN with $\epsilon^{vec}_{\max} = 0.0100$. As the number of samples is very large, and the set of block-orthogonal matrices is compact, this strongly suggests that in contrast to the scalar case, in general $\epsilon^{vec}_{\max} \geq 1 - \min_Q \varrho(\overline{QRQ^T})$ is *not tight*, and the two notions are *not equivalent*. In example 2 (a) we have seen that in some cases the equality may hold, but example 2 (b) shows that it does not hold in general. Continuing with example 2 (b), we next perform local numerical optimization over block-orthogonal S with 5000 random initial conditions to find a choice of S with $\varrho(\overline{I - SJS^T}) \approx 0.9901$. This is a difficult global optimization problem, and finding such S is challenging, requiring many random restarts. However, the existence of such S shows that the model is generalized walk-summable. We conjecture that the set of block-orthogonal transformations is not enough to capture generalized walk-summability, but that block-invertible transformations may be sufficient. Proving these results appears challenging.

Example 3. We now apply similar analysis to the chain-graph in Figure 4.9, with vector variables. Of course, vector-LBP in this model will terminate finitely and give exact answers. We are interested to see if vector walk-summability is powerful enough to guarantee this fact (vector walk-summability is sufficient but not necessary for vector-LBP convergence). We pick a unit-diagonal positive-definite information matrix (again the edge-weights are chosen randomly, and then the matrix is normalized). From Lemma 4.2.9 tree-structured models are vector-PN. The index of vector-PN is $\epsilon^{vec}_{\max} = 0.9900$. We took 500000 uniform samples of block-orthogonal Q and observed the minimum to have $\varrho(\overline{QRQ^T}) \approx 1.05206$ which is well above 1. Although the sufficient condition for vector walk-summability does not hold, we can not conclude whether or not the model is vector walk-summable. However, using local optimization over block-invertible matrices with 100 random starting points, we obtain $\varrho(\overline{I - SRS^T}) \approx 0.9902$, which shows that J *is* generalized vector walk-summable. This leads us to a conjecture

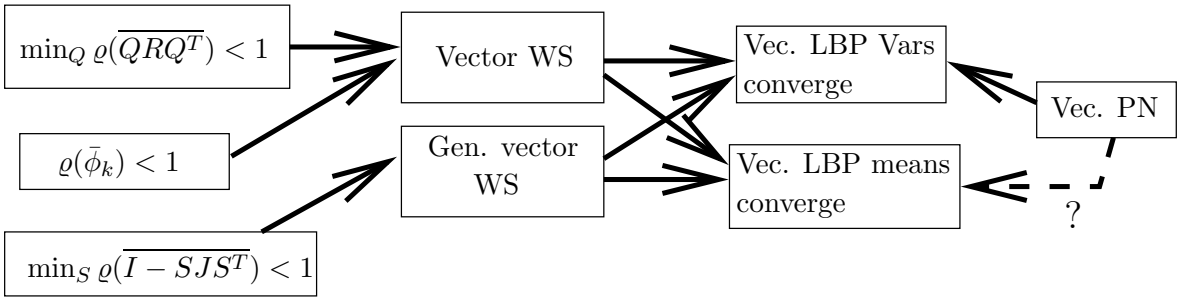


Figure 4.10. Inter-relation of various conditions for vector LBP convergence.

that the sufficient condition $\varrho(\overline{I - SRS^T}) < 1$ is always satisfied for tree-structured models, and such models are not only vector-PN but also generalized vector-WS.

■ 4.2.6 Remarks on Vector-WS

We have seen from our sufficient conditions and from the numerical experiments that the vector case is much more involved than the scalar case. In the scalar case walk-summability is equivalent to $\varrho(\bar{R}) < 1$, which is in turn equivalent to pairwise-normalizability. Also, in the scalar case there is never any need to consider generalized walk-summability (that leads to R matrices with non-zero diagonals) as the best sufficient conditions are obtained from 0-diagonal matrices. In the vector case we do not have a simple characterization of vector walk-summability, only a family of sufficient conditions. In addition, in the vector case allowing R matrices with non-zero block-diagonal (generalized walk-summability) is more powerful than ordinary vector walk-sum decompositions. Finally the precise relationship between vector-PN and vector-WS is not entirely clear. We summarize various sufficient conditions for vector walk-summability and vector LBP convergence in Figure 4.10. We also have a family of interesting conjectures which appear challenging to prove:

1. Is generalized vector-WS equivalent to vector-PN?
2. Are tree-structured models generalized vector-WS?
3. Is $\min_Q \varrho(\overline{QRQ^T}) < 1$ equivalent to vector-WS? (we established sufficiency, but not necessity.)
4. Is $\min_S \varrho(\overline{I - SJS^T}) < 1$ equivalent to generalized vector-WS?

Despite the list of unsolved conjectures, it is clear from this section that vector walk-summability is a useful notion, and it justifies a powerful family of vector-LBP algorithms which allows one to balance convergence and accuracy versus computational complexity.

■ 4.3 Factor Graph LBP and FG Walk-summability

Next, we investigate a factor graph (FG) representation of Gaussian models, where the information matrix J admits a decomposition into a set of matrices J_F over factors F. This representation is important as in some applications Gaussian models may be conveniently specified in a factor graph form – this includes the thin-plate model, and the multi-user detection problem that we mention in Section 4.3.4. In addition, similar to using vector-variables, factor graph representation provides a more general version of LBP, which allows one to trade-off computational complexity of inference for accuracy of the results. We discussed factor graphs in Section 2.1.2, but to remind the reader we briefly review our notation. Let V be a set of vertices[13], and $\mathcal{F} \subset 2^V$ be a collection of subsets of V. The pair, $(V, \mathcal{F})$ defines a hypergraph, a generalization of undirected graphs which allows 'hyper-edges' $F \in \mathcal{F}$ (edges which join vertex-sets of size greater than two). A convenient representation of a hypergraph is a factor graph – a bipartite graph with factor and variable nodes, and edges indicating which vertices belong to which factor. A probability density factorizes over a factor graph if it can be represented as a product of local potential functions ψ_F which depend only on x_F, the variables corresponding to the factor:

$$p(x) \propto \prod_{F \in \mathcal{F}} \psi_F(x_F) \tag{4.21}$$

A Gaussian density in a factor graph form has potentials of the form

$$\psi_F(x_F) = \exp\{-\frac{1}{2} x_F^T J_F x_F + h_F^T x_F\}.$$

The information parameters associated with these potentials must satisfy:

$$x^T J x = \sum_{F \in \mathcal{F}} x_F^T J_F x_F, \qquad \text{and} \qquad h^T x = \sum_{F \in \mathcal{F}} h_F^T x_F, \tag{4.22}$$

i.e. local information parameters must add up to the global ones.

■ 4.3.1 Factor Graph LBP (FG-LBP) Specification

Recall the factor graph version of belief propagation from Chapter 2. For convenience, we separate the factors into single-node factors ψ_i and higher-order factors ψ_F (with $|F| > 1$): $p(x) \propto \prod_{i \in V} \psi_i(x_i) \prod_{F \in \mathcal{F}} \psi_F(x_F)$. We summarize FG-LBP message updates, for both factor-to-variable messages, and variable-to-factor messages, described

[13]The random variable x_i at a vertex i is in general allowed to be vector-valued, although here we only consider the scalar case for simplicity.

in Chapter 2. We use i and j to refer to variables and A and B to refer to factors:

$$m_{i\to A}(x_i) = \psi_i(x_i) \prod_{B\in\mathcal{N}(i)\setminus A} m_{B\to i}(x_i) \tag{4.23}$$

$$m_{A\to i}(x_i) = \int_{x_A\setminus x_i} \psi_A(x_A) \prod_{j\in\mathcal{N}(A)\setminus i} m_{j\to A}(x_j) \tag{4.24}$$

The beliefs are calculated by fusing all the incoming messages:

$$b_i(x_i) \propto \psi_i(x_i) \prod_{A\in\mathcal{N}(i)} m_{A\to i}(x_i) \quad \text{and} \quad b_A(x_A) \propto \psi_A(x_A) \prod_{i\in\mathcal{N}(A)} m_{i\to A}(x_i) \tag{4.25}$$

Let us specialize the above message updates to the Gaussian case. We parameterize both types of messages in information form:

$$m_{A\to i}(x_i) \propto \exp(-\frac{1}{2}x_i^T\ \Delta J_{A\to i}\ x_i + \Delta h_{A\to i}^T x_i) \tag{4.26}$$

$$m_{i\to A}(x_i) \propto \exp(-\frac{1}{2}x_i^T\ \Delta J_{i\to A}\ x_i + \Delta h_{i\to A}^T x_i) \tag{4.27}$$

Marginalization corresponds to taking Schur complements of the information parameters, so belief propagation equations reduce to the following:

$$\begin{aligned}
\hat{J}_{A\setminus i} &= [J_A]_{A\setminus i} + \sum_{j\in\mathcal{N}(A)\setminus i} \Delta J_{j\to A} \\
\hat{h}_{A\setminus i} &= [h_A]_{A\setminus i} + \sum_{j\in\mathcal{N}(A)\setminus i} \Delta h_{j\to A} \\
\Delta J_{A\to i} &= [J_A]_i - [J_A]_{i,A\setminus i}\ \hat{J}_{A\setminus i}^{-1}\ [J_A]_{A\setminus i,i} \\
\Delta h_{A\to i} &= [h_A]_i - [J_A]_{i,A\setminus i}\ \hat{J}_{A\setminus i}^{-1}\ [h_A]_{A\setminus i} \\
\Delta J_{i\to A} &= J_i + \sum_{B\in\mathcal{N}(i)\setminus A} \Delta J_{B\to i} \\
\Delta h_{i\to A} &= h_i + \sum_{B\in\mathcal{N}(i)\setminus A} \Delta h_{B\to i}
\end{aligned} \tag{4.28}$$

The notation $[J_A]$ represents the local size $|A|$ square matrix J_A zero-padded to have size $|V|$, and $[J_A]_{A\setminus i}$ represents the submatrix of J_A corresponding to the variables $A\setminus i$. FG-LBP starts with non-informative messages (all information parameters being set to zero). Upon convergence the marginals at each variable and each factor can be evaluated by

$$\hat{J}_i = J_i + \sum_{A\in\mathcal{N}(i)} \Delta J_{A\to i} \quad \text{and} \quad \hat{h}_i = h_i + \sum_{A\in\mathcal{N}(i)} \Delta h_{A\to i} \tag{4.29}$$

$$\hat{J}_A = J_A + \sum_{j\in\mathcal{N}(A)} \Delta J_{j\to A} \quad \text{and} \quad \hat{h}_A = h_A + \sum_{j\in\mathcal{N}(A)} \Delta h_{j\to A} \tag{4.30}$$

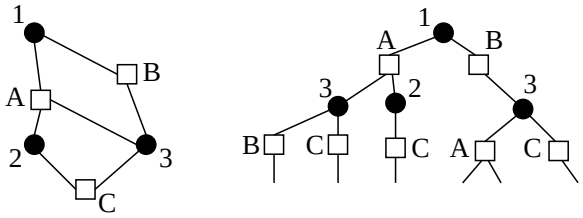


Figure 4.11. A simple factor graph, and its computation tree rooted at variable-node 1.

The marginal covariance and means can then be evaluated by:

$$\hat{P}_i = \hat{J}_i^{-1}, \quad \hat{P}_A = \hat{J}_A^{-1}, \qquad \text{and} \qquad \hat{\mu}_i = \hat{J}_i^{-1}\hat{h}_i, \quad \hat{\mu}_A = \hat{J}_A^{-1}\hat{h}_A \tag{4.31}$$

The history of message updates in FG-LBP can be captured by the computation tree, which is constructed by "unwinding" the factor graph, analogous to the pairwise-MRF computation tree construction. An illustration of a computation tree rooted at variable-node 1 appears in Figure 4.11. The root-node estimates in an N-step computation tree rooted at variable-node i correspond to FG-LBP estimates at variable-node i after N steps.

■ 4.3.2 Factor Graph Walk-summability

We now describe a walk-sum representation of inference in factor graph representations of Gaussian models. We decompose the information matrix: $J = I - R$, with $R = \sum_F [R_F]$. This decomposition is not unique, and finding a good decomposition will play an important role in our analysis. For simplicity we set $h_F = 0$, and only use self potentials h_i. We define a walk in a factor graph as a sequence of connected steps (i, F, j), where $i, j \in F$, and the weight of a step to be $[R_F]_{i,j}$. The matrices R_F may in general have entries on the diagonal, so a step (i, F, i), i.e. a self-loop, may also carry non-zero weight. Define the weight of a walk to be $\phi(w) = \prod_{(i,F,j)\in w}[R_F]_{i,j}$, and for a set of walks $\mathcal{W}$, $\phi(\mathcal{W}) = \sum_{w\in\mathcal{W}} \phi(w)$. Also we define the reweighted walk-sum $\phi_h(\mathcal{W}) = \sum_{w\in\mathcal{W}} h_{w_0}\phi(W)$, where w_0 is the starting variable-node of the walk w. Consider the power series expression for the inverse of J:

$$P = J^{-1} = (I - \sum_F [R_F])^{-1} = \sum_k (\sum_F [R_F])^k$$

Entry (i, j) of the matrix $(\sum_F [R_F])^k$ is a walk-sum over walks that start at variable-node i, finish at variable-node j and make k steps in between, switching from one factor to another (from one matrix R_F to another) in between. In parallel to the scalar case we have a walk-sum formulation for inference: $P_{i,j} = \phi(\mathcal{W}_{i\to j})$, and $\mu_i = \phi_h(\mathcal{W}_{*\to j})$ provided that these walk-sums are well-defined, i.e. converge independent of the order of summation.

Definition: FG-WS. We call a model *factor graph walk-summable* if for all pairs of vertices i and j the walk-sum $\phi(i \to j) \triangleq \sum_{w \in \mathcal{W}(i \to j)} \phi(w)$ converges, and is independent of the order of summation of the walks, or alternatively if $\bar{\phi}(i \to j) \triangleq \sum_{w \in \mathcal{W}(i \to j)} |\phi(w)|$ converges.

We have a simple characterization for FG walk-summability with scalar variables:

Proposition 4.3.1 (Sufficient condition for FG walk-summability). *A decomposition $R = \sum_F [R_F]$ is factor graph walk-summable if and only if $\varrho(\sum_F [\bar{R}_F]) < 1$.*

Proof. We have $\bar{\phi}(i \overset{k}{\to} j) = \sum_{w \in \mathcal{W}(i \overset{k}{\to} j)} |\phi(w)| = \left((\sum_F [\bar{R}_F])^k\right)_{i,j}$, and $\bar{\phi}(i \to j) = \sum_k \bar{\phi}(i \overset{k}{\to} j) = \left(\sum_k (\sum_F [\bar{R}_F])^k\right)_{i,j}$. The convergence of the latter is determined by $\varrho(\sum_F [\bar{R}_F]) < 1$. □

The scalar sufficient condition $\varrho(\bar{R}) < 1$ is in general not sufficient for factor graph walk-summability, and a stronger condition is required: $\varrho(\sum_F [\bar{R}_F]) < 1$. Note that by triangle inequality $\varrho(\sum_F [\bar{R}_F]) \geq \varrho(\bar{R})$, so the factor graph condition appears more restrictive.

However, the advantage of using the factor graph representation over the scalar one comes from allowing various splitting $R = \sum_F R_F$, and transformations over certain blocks of variables[14]. Thus, one is interested to find the best decomposition which leads to the most general sufficient condition. Finding the best walk-sum decomposition is even more challenging than in the vector-LBP case. We do not pursue this analysis further here, but we will see from examples in Section 4.3.4 that the factor graph version of LBP can be much more powerful than scalar LBP. We now state the relation between factor graph walk-summability and factor graph LBP, with details described in the appendix.

Walk-sum interpretation of FG-BP in trees The message updates of the factor graph version of BP can be related to computing walk-sums in subtrees of the model illustrated in Figure 4.12. We only state the general result here, and refer to Section A.2.3 in Appendix A for a detailed statement and the proof.

Lemma 4.3.1 (Walk-sum interpretation of FG-BP). *Messages in the factor graph version of BP in tree-structured models correspond to computing walk-sums in certain subtrees of the model.*

The convergence of FG-LBP for both means and variances is guaranteed by factor graph walk-summability. The proof is provided in Section A.2.3 of Appendix A.

[14] In order not to introduce additional fill in the model, the allowed transformations are over blocks which are fully contained in each of the factors that they intersect. Consider the factor graph in Figure 4.11. Block $\{2, 3\}$ is fully contained in factors A and C, but it is not contained in B even though it intersects B. The same happens for blocks $\{1, 2\}$ and $\{1, 3\}$, so only scalar transformations are allowed for this factor graph.

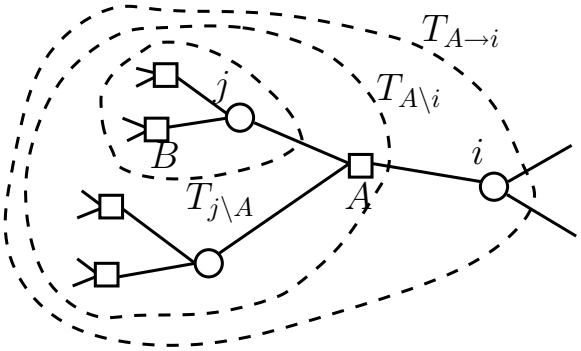


Figure 4.12. Subtrees used in the analysis of FG-BP: $T_{A\to i}$ and $T_{A\setminus i}$. See Section A.2.3 in Appendix A.1 for definitions.

Proposition 4.3.2 (Convergence of FG-LBP). *FG-LBP converges in factor graph walk-summable models: the means are correct, and the variances correspond to the walk-sum over back-tracking walks in the factor graph computation tree.*

Note that back-tracking walks in a factor graph computation tree are different from the back-tracking walks in the scalar one: if we construct an equivalent scalar version of the factor graph by interconnecting the variables belonging to each factor, then walks that were back-tracking in the factor graph may no longer be back-tracking in this scalar graph. Next, we consider factor graph normalizability and its relation to FG-LBP. Instead of attempting to establish connections between FG-normalizability and FG-WS, we will see that FG-normalizability by itself can guarantee convergence of FG-LBP variances, although damping may be needed to also obtain convergence of the means.

■ 4.3.3 Factor Graph Normalizability and its Relation to LBP

We have seen the importance of pairwise-normalizability for the scalar and vector versions of LBP. We now investigate the related notion of factor graph normalizability, and show that it is sufficient for FG-LBP variance convergence, but may not guarantee convergence of the means for some models.

We say that a Gaussian model is *factor graph normalizable* if

$$J = \sum_{F\in\mathcal{F}} [J_F], \quad \text{where} \quad J_F \succ 0 \tag{4.32}$$

The matrix J_F is a local $|F| \times |F|$ matrix. We use the notation $[J_F]$ to denote a zero-padded matrix of size $|V| \times |V|$, whose principle submatrix for x_F is J_F.

For the analysis of factor graph normalizable models we use a decomposition of J into the following factor and node potentials: $J = \sum_i J_i + \sum_F J_F$, where $J_f \succ 0$ and $J_i > 0$. In this section we also use these potentials to specify FG-LBP, instead of our

usual walk-sum potentials in (2.24) with zero-diagonal J_f, and $J_i = 1$. We start by establishing convergence of LBP variances in FG-normalizable models.

Proposition 4.3.3 (LBP variances in FG-normalizable models). *FG-LBP variances converge in FG-normalizable models.*

The proof appears in Section A.2.4 in Appendix A. The key idea is that as the computation tree grows, its information matrix gets appended with positive-definite (p.d.) terms, so it monotonically increases in the p.d. sense. Hence, the corresponding computation tree covariance matrix monotonically decreases in the p.d. sense, and the LBP variance estimate at the root node monotonically decreases. Since the computation trees for FG-normalizable models are always valid, LBP variances are always positive, hence bounded below by zero, and must converge. We note that in the walk-sum decomposition the variances are monotonically increasing, whereas in an FG-normalizable decomposition of the same model the variances are monotonically decreasing![15]

Unfortunately, this simple proof does not extend to the convergence of the means, which does not hold in general. We now provide an example of an FG-normalizable model where FG-LBP means do not converge.

Example 4. We provide a simple model with two factors over variables $\{1, 2, 3\}$, $J = J_1 + J_2$, where $J_1 \succ 0$ and $J_2 \succ 0$. They are

$$J_1 = \begin{pmatrix} 510.9423 & -549.4371 & 341.4739 \\ -549.4371 & 604.3130 & -384.2904 \\ 341.4739 & -384.2904 & 253.6471 \end{pmatrix}, \quad \text{and} \quad J_2 = \begin{pmatrix} 283.1945 & 161.4936 & -196.3614 \\ 161.4936 & 377.9960 & 88.1683 \\ -196.3614 & 88.1683 & 316.1255 \end{pmatrix}. \quad (4.33)$$

The model is FG-normalizable, variances converge, but means diverge. The splitting is a valid FG-normalizable splitting, but it is not a wise splitting – combining the two factors into a single one, J, makes the model trivial, containing just one factor, and inference is exact at no additional computational cost. Also note that the model is in fact scalar walk-summable, with $\varrho(\bar{R}) \approx 0.7085$, so even scalar LBP converges for this model. Thus the splitting into factors is indeed very unfortunate, but it illustrates the point that FG-normalizability does not guarantee convergence of the means in FG-LBP.

Recall that once the variances converge, the mean updates follow a linear system, see (4.28). We have observed that we can force the means to converge by sufficient damping of this linear system, similar to what we have seen for scalar LBP. Since damping does not change the LBP fixed points, this still provides the correct means. In the next section we describe an application where FG-LBP representation dramatically outperforms scalar LBP.

[15] The difference between these two versions of FG-LBP stems from the choice of decomposition of the matrix J into terms J_F and J_i. In the interior of the computation tree the potentials for all the neighbors are added, and the choice of a decomposition is unimportant. However, at the leaves some of the terms are missing, and hence the behaviour of FG-LBP depends on the potential decomposition. This can also be thought of as changing the initial conditions.

■ 4.3.4 Relation of Factor Graph and Complex-valued Version of LBP

Montanari, Prabhakar and Tse [98] described a generalized complex-valued version of Gaussian LBP and used it for inference in hard Gaussian models for multi-user detection problems where regular scalar pairwise LBP fails. We describe their approach and show that it is closely related to a real-valued factor graph version of LBP.

Suppose $y = Hx + n$, where x and n are 0-mean Gaussian random vectors with information matrices J and Q, respectively. Then, the posterior distribution of x conditioned on y has information parameters $\hat{h} = H^T Q y$ and $\hat{J} = H^T Q H + J$. When the noise is small, the matrix $\hat{J} = H^T Q H + J$ may be highly ill-conditioned, and scalar LBP fails to compute the marginals of the posterior of x given y.

Montanari et al. [98] instead use a complex-valued form of LBP. The paper does not provide the details of the derivation. We provide our derivation here for completeness. The joint density of x and n conditioned on y is degenerate: $p(x, n \mid y) \propto e^{-\frac{1}{2}x^T Jx} e^{-\frac{1}{2}n^T Qn} \delta(y - Hx - n)$, where $\delta(x)$ is the Dirac delta function[16]. Thus $p(x, n \mid y)$ is non-zero only on the subspace $y = Hx + n$. We use the identity $\delta(n - n_0) = \int_w e^{jw(n-n_0)} dw$, with $j = \sqrt{-1}$, to do the following:

$$p(x \mid y) \propto \int e^{-\frac{1}{2}x^T Jx} e^{-\frac{1}{2}n^T Qn} \delta(y - Hx - n) dn = \tag{4.34}$$

$$\int e^{-\frac{1}{2}x^T Jx} e^{-\frac{1}{2}n^T Qn} \int_w e^{j\omega^T (y - Hx - n)} dw \; dn = \tag{4.35}$$

$$\int_w e^{-\frac{1}{2}x^T Jx} e^{j\omega^T (y - Hx)} \int_n e^{-\frac{1}{2}n^T Qn} e^{-j\omega^T n} dn \; dw \tag{4.36}$$

The integral $\int_n e^{-\frac{1}{2}n^T Qn + j\omega^T n} dn$ is equal[17] to $\sqrt{(2\pi)^n} |J|^{-1/2} e^{-\frac{1}{2}w^T Q^{-1} w}$. Now we have:

$$p(x \mid y) \propto \int_w e^{-\frac{1}{2}x^T Jx} e^{j\omega^T (y - Hx)} e^{-\frac{1}{2}w^T Q^{-1} w} \; dw \tag{4.37}$$

Note that the quantity inside the integral is complex, and is not a probability density in the standard sense. However, by integrating out w, the corresponding marginal $p(x \mid y)$ is an ordinary real-valued probability density. This can be written in information form as follows:

$$p(x \mid y) \propto \int_w e^{-\frac{1}{2}\left[\begin{smallmatrix} x \\ w \end{smallmatrix}\right]^T \left(\begin{smallmatrix} J & jH^T \\ jH & Q^{-1} \end{smallmatrix}\right) \left[\begin{smallmatrix} x \\ w \end{smallmatrix}\right] + \left[\begin{smallmatrix} 0 \\ jy \end{smallmatrix}\right]^T \left[\begin{smallmatrix} x \\ w \end{smallmatrix}\right]} dw \tag{4.38}$$

The corresponding information matrix is $J_x = J - j^2 H^T Q H = J + H^T Q H$, and $h_x = 0 - j^2 H^T Q y = H^T Q y$ – both are real-valued, and J_x is positive-definite.

[16] Here, $\delta(x)$ is a generalized function, with the defining property $\int_{-\infty}^{\infty} f(x)\delta(x - x_0)dx = f(x_0)$ for any continuous $f(x)$. Refer to [102, 140] for details.

[17] Using $\int e^{-\frac{1}{2}x^T Jx + h^T x} dx = \sqrt{(2\pi)^n} |J|^{-1/2} e^{\frac{1}{2}h^T J^{-1} h}$.

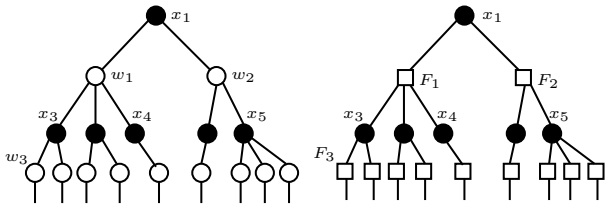


Figure 4.13. (a) The computation tree for complex-valued LBP, with filled nodes corresponding to x and unfilled nodes corresponding to w. (b) The corresponding real-valued factor graph.

We do not consider such models in full generality, but rather make a restriction that both J and Q are diagonal. This restriction still represents a very wide class of models: for example the conditionally auto-regressive (CAR) models [110] fall into this class. A CAR models for x is specified by listing a consistent set of conditional probabilities $p(x_i \mid x_{V\setminus i})$ for all i. The thin-plate model described in Chapter 2 belongs to the class of CAR models, and admits the representation in (4.38) with diagonal blocks, as we describe in Section A.2.5 of Appendix A. The multi-user detection problem described in [98] is also in the block-diagonal form with both x and n i.i.d. We use $J = I$ and $Q = \sigma^{-2}I$. The information parameters for the pair (x, w) is

$$\tilde{J} = \begin{pmatrix} I & jH^T \\ jH & \sigma^2 I \end{pmatrix} \quad \text{and} \quad \tilde{h} = \begin{bmatrix} 0 \\ jy \end{bmatrix} \tag{4.39}$$

Given these information parameters $\tilde{J}$ and $\tilde{h}$ one can apply Gaussian LBP equations in (2.29, 2.30, 2.31) to find the marginal means and variances of x. The graph corresponding to $\tilde{J}$ is bipartite, with two components x and w, and the edges connecting variables in x to variables in w, and no edges within each component.

We note that although some of the entries of $\tilde{J}$ and $\tilde{h}$ are complex-valued, the message updates for variances only involve passing real-valued quantities. For example, every quantity $\hat{J}_{a\setminus b}$ is real-valued at the start (here we use a and b to denote variable nodes, because i, j would interfere with the complex number notation). The messages $\Delta J_{a\to b} = -J_{b,a}\hat{J}^{-1}_{a\setminus b}J_{ab}$ are also real-valued: these messages are only passed between the two bipartite components, hence $\Delta J_{a\to b} = -(jH_{b,a})\hat{J}^{-1}_{a\setminus b}(jH_{a,b}) = H_{b,a}\hat{J}^{-1}_{a\setminus b}H_{a,b}$ is also real-valued. Thus the subsequent quantities $\hat{J}_{a\setminus b} = J_{ii} + \sum_{k\in\mathcal{N}(a)\setminus b}\Delta J_{k\to a}$ and the subsequent messages $\Delta J_{a\to b}$ are also real-valued. Similarly all the messages $\Delta h_{b\to a}$ coming to nodes a that correspond to the x component, and all $\hat{h}_{a\setminus b}$ are real-valued. Thus the means and variances of x obtained by the complex-valued version of LBP are guaranteed to be real-valued. However, for the nodes that are in the w component the quantities $\Delta h_{b\to a}$ and $\hat{h}_{a\setminus b}$ are purely imaginary.

Equivalence of the complex-valued and factor graph forms of LBP Now consider the computation tree for the complex version of LBP displayed in Figure 4.13(a). The computation tree is bipartite with alternating levels corresponding to variables in x

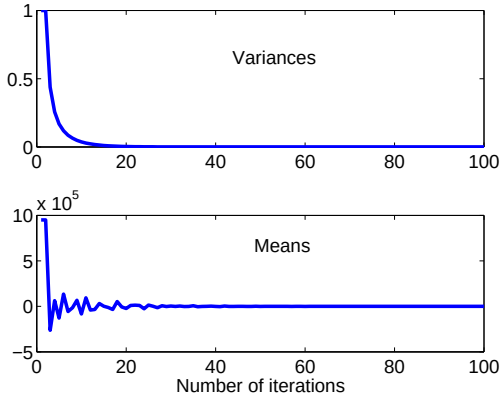


Figure 4.14. Convergence of FG-LBP variances (top) and means (bottom) for the multi-user detection example.

and in w. If we marginalize out variable w_k we induce a factor F depending on all of its neighbors in x. The induced factor will have information parameters $J_F = -(jH_k)\sigma^{-2}(jH_k)^T = \sigma^{-2}H_kH_k^T$ and $h_F = -(jH_k)\sigma^{-2}(jy_k) = \sigma^{-2}H_ky_k$.

Hence, the computation tree from the point of view of the root node is equivalent to the factor graph computation tree depicted in Figure 4.13(b), with factors F replacing variables w. This computation tree corresponds to the factor graph LBP applied to the factorization of $\hat{J}$ into rank-1 potentials $J_F = \sigma^{-2}H_kH_k^T$ and self-potentials $J_i = 1$. We note that convergence of the variances for this factor-graph LBP is guaranteed by factor graph normalizability, as the rank-1 matrices $H_kH_k^T$ are positive-semi-definite. The paper [98] proves convergence of the variances directly from LBP updates, without making a connection to factor graphs. They also are able to establish convergence of LBP means with high probability as the number of variables tends to infinity.

Example 5. (Factor-graph LBP for multi-user detection). We consider the Gaussian multi-user detection problem described in [98]. Suppose each of K users would like to transmit a symbol x_i, and encodes it with a signature vector $H_i \in \mathbb{R}^N$. The received signal is a linear combination

$$y = \sum_{i=1}^{K} H_i x_i + n = Hx + n, \tag{4.40}$$

where we define $H = [H_1, ..., H_K]$, and n is additive Gaussian noise. We assume that $x \sim \mathcal{N}(0, I)$ and $n \sim \mathcal{N}(0, \sigma^2 I)$. This fits exactly in the form described in (4.39). Following [98], we pick the matrix H to have i.i.d. ± 1 entries with equal probability. We use $N = 50$ receivers, and $K = 40$ users.

First we apply scalar LBP to compute marginals of the posterior of x given y with information parameters $J = I + \frac{1}{\sigma^2} H^T H$, and $h = \frac{1}{\sigma^2} H^T y$. By normalizing J to have unit diagonal (and 0-diagonal R), we have $\varrho(\bar{R}) \approx 4.4173$. This model is severely non-walk-summable, and scalar LBP quickly fails.

Next, we apply the factor graph factorization to the problem, with factor potentials $J_F = \sigma^{-2} H_k H_k^T$, self-potentials $J_i = 1$, and $h_i = (\frac{1}{\sigma^2} H^T y)_i$. In agreement with our sufficient condition, FG-LBP variances converge. The LBP means also converge in this example (and, of course, converge to the correct values). This is in line with the high-probability convergence results for the means in [98], which use random matrix theory. The plot of convergence of FG-LBP variances and means appears in Figure 4.14: after about 50 iterations both means and variances reach fixed points[18].

This demonstrates that a judicious choice of a factor graph representation may lead to much better performance of FG-LBP in comparison with scalar LBP.

■ 4.4 Chapter Summary

In this chapter we continued our study of the walk-sum interpretation of Gaussian inference. We started with a combinatorial analysis of walk-sums in regular graphs with homogeneous weights, and used it to give insight into the behaviour of LBP variances. We then explored more general notions of walk-summability geared towards models with vector variables and models defined with respect to factor graphs. Walk-summability can be successfully extended to these scenarios, and we have shown that it guarantees convergence of the corresponding versions of LBP. However, walk-summability itself becomes harder to characterize, and we only developed sufficient conditions. Finally we have considered factor graph normalizability, established that it guarantees convergence of the variances for FG-LBP, and related FG-LBP to an intriguing complex-valued version of LBP.

[18] FG-LBP fixed points for means and variances are all non-zero. This is hard to see from the plot because of its dynamic range.

Chapter 5

Low-rank Variance Approximation in Large-scale GMRFs

We now make a departure from the walk-sum framework described in Chapters 3 and 4, and consider the problem of computing accurate approximate variances in very large-scale GMRFs. Such large-scale problems with 2D or 3D fields with millions of variables appear in image-processing and remote sensing. Since exact approaches are not tractable for this setting, and no guarantees are available for the accuracy of LBP variances, we propose a new approach, which, in addition to its efficiency and simplicity, also provides accuracy guarantees.

Our approach relies on constructing a low-rank aliasing matrix with respect to the Markov graph of the model which can be used to compute an approximation to the inverse $J^{-1} = P$. By designing this matrix such that only the weakly correlated terms are aliased, we are able to give provably accurate variance approximations. The method uses iterative solutions of sparse linear systems, and it is scalable.

We introduce our low-rank variance approximation approach, apply it to short-correlation models, and establish accuracy guarantees in Section 5.1. We then describe the spliced-wavelet extension for models with long correlation lengths in Section 5.2, and in Section 5.2.3 we apply the construction to multi-scale models. In Section 5.3 we test our approach with experiments, including estimation problems from oceanography and gravity inversion.

■ 5.1 Low-rank Variance Approximation

We devote an entire chapter to computing the variances in large-scale GMRFs firstly because they serve a crucial role in estimation and learning, and secondly because computing variances is a far harder problem than computing the means. In essence, to find the means we need to solve a sparse linear system, while to obtain the variances we have a much harder task of computing the diagonal of the inverse of a sparse positive-definite matrix. Owing to the sparsity of the graph, approximate means can be computed with linear complexity in the number of nodes using iterative solvers such as preconditioned conjugate gradients, or multigrid [126]. Such methods do not provide the variances of the estimates. LBP can be also used to obtain exact means, but has no accuracy

guarantees on the variances.

The primary reason why variances are a crucial component of estimation, and their computation can not simply be sidestepped, is that they give the reliability information for the means. They are also useful in other respects: regions of the field where residuals exceed error variances may be used to detect and correct model-mismatch (for example when smoothness models are applied to fields that contain abrupt edges). Also, as inference is an essential component of learning a model (for both parameter and structure estimation), accurate variance computation is needed when designing and fitting models to data. Yet another use of variances is to assist in selecting the location of new measurements to maximally reduce uncertainty.

■ 5.1.1 Introducing the Low-rank Framework

Finding the means of a GMRF corresponds to solving the linear equations $J\mu = h$. For sparse graphs, a variety of efficient, iterative algorithms exist for solving such equations with complexity that grows roughly linearly with the number, N, of nodes in the graph (see Section 5.4 for details) [116]. However, except for models on trees, such linear complexity is not readily available for the computation of the covariance matrix. One way in which one might imagine performing this computation is to embed it in a set of N linear equation solvers. Let $v_i \in \mathbb{R}^N$ be the i-th standard basis vector, then the i-th column of P can be obtained by solving

$$JP_i = v_i \tag{5.1}$$

To get all N columns of P, this would have to be done N times, once at each node in the graph: $JP = [v_1, ..., v_N] = I$ with complexity $O(N^2)$. This is still intractable for large-scale models with millions of variables. Note that the full P matrix has N^2 elements, so quadratic complexity is a lower-bound to compute all of P.

However, in many cases we are most interested only in the diagonal elements, P_{ii}, of P (i.e., the individual variances)[1], and this raises the question as to whether we can compute or approximate these elements with procedures with only linear complexity. Of course the direct computation $\mathrm{diag}(P) = \mathrm{diag}(J^{-1}I)$ is costly. Instead we propose to design a low-rank matrix BB^T, with $B \in \mathbb{R}^{N\times M}$ and $M \ll N$, and use it instead of I. The system $J\hat{P} = BB^T$ can be solved with $O(MN)$ complexity in two steps: first we solve $JR_B = B$ using iterative solvers[2]. Then, we post-multiply R_B by B^T, i.e. $\hat{P}_{ii} = [R_B B^T]_{ii}$ (which requires MN operations, as we only need the diagonal).

To get accurate variance approximations, B must be designed appropriately, taking the graph and the correlation structure of the model into consideration. Let all rows b_i of B have unit norm: $b_i^T b_i = 1$. Consider the diagonal of $\hat{P} = J^{-1}(BB^T)$:

$$\hat{P}_{ii} \triangleq [J^{-1}(BB^T)]_{ii} = P_{ii} + \sum_{i\neq j} P_{ij}\ b_i^T b_j. \tag{5.2}$$

[1] It is also possible to use our approach to find accurate approximations of the elements of the covariance which correspond to nearby nodes. For sparse models there are $O(N)$ of such elements.

[2] We note that the matrix R_B is *not related* to the decomposition $J = I - R$ in Chapters 3 and 4.

To force $\hat{P}_{ii}$ to be accurate approximations of the variances we need the aliased terms $P_{ij}\ b_i^T b_j$ to be nearly zero for all pairs of nodes. We analyze two different cases. For models with short-range correlations P_{ij} decays fast and is nearly zero for most pairs, so we only have to take care of the nearby nodes. In the long-range correlation case we use a wavelet decomposition to decompose the correlation across several scales, thus producing several problems with short correlation length. Moreover, by adding randomness to the choice of B (and perhaps computing approximations with several such random choices), we can obtain unbiased approximations of the true covariances. We describe the short-range correlation construction next, and a wavelet-based extension for models with long correlations in Section 5.2.

■ 5.1.2 Constructing B for Models with Short Correlation

The key idea here is that to make $P_{ij}b_i^T b_j$ small, we need either P_{ij} or $b_i^T b_j$ to be small. Suppose that P_{ij} decays fast with distance from node i to j. Then, for nodes that are far apart in the graph (further than the correlation length[3]), the correlation P_{ij} and the corresponding error-terms in (5.2) are negligible. For pairs of nodes i and j that are nearby, we have to design B such that b_i and b_j are orthogonal: this is a problem of designing an overcomplete basis $\{b_i \in \mathbb{R}^M\}$ that is nearly orthogonal with respect to a graph G. We describe such a construction for chains and rectangular lattices, and suggest an approach for arbitrary sparse graphs.

For the sake of clarity we start with a simple 1D chain example[4]. We assume that the correlation between nodes decays rapidly with distance (e.g., in many models correlation decays exponentially with distance $d(i,j)$ between i and j: $|P_{ij}| \leq A\ \beta^{d(i,j)}$ with $0 \leq \beta < 1$). Consider Figure 5.1(a) and (b). We plot the i-th standard basis vector v_i with $i = 50$ in plot (a), and the i-th column P_i of P, the solution to the system $JP_i = v_i$ in plot (b). There is a spike of P_{ij} at $j = i$, a fast decaying response for j near i, and most of other entries are nearly zero. Now let $z = v_{i_1} + v_{i_2} + ...v_{i_K}$, where all indices i_k's are mutually well-separated. In Figure 5.1(c) and (d) we show z and the solution w to $Jw = z$. We also show P_{ii} (dashed). At each i_k we have a spike and a fast-decaying response. This operation can be seen as a convolution of a spike-train with a time-varying kernel. If the spikes are well-separated, then the interference from other spikes is small, and $w_{i_k} \approx P_{i_k,i_k}$ for each k. This is the basic idea behind the construction of our B matrix for the short-range correlation case.

Now to find such groups of well-separated nodes, we partition the nodes into classes, which we call *colors*, such that nodes of the same color are a distance M apart. For chains this can be done simply by periodically cycling through the M colors. We will

[3]We define the correlation length to be a distance in the graph beyond which the correlation coefficient between any two nodes becomes negligible (smaller than some specified threshold). For models with exponential decay this is consistent with the conventional definition, but it also applies to models with other modes of correlation decay.

[4]Here we consider a chain in a generalized sense, meaning that the nodes have an inherent 1D ordering, but the Markov graph does not have to be a chain and may have links a few steps away in the ordering.

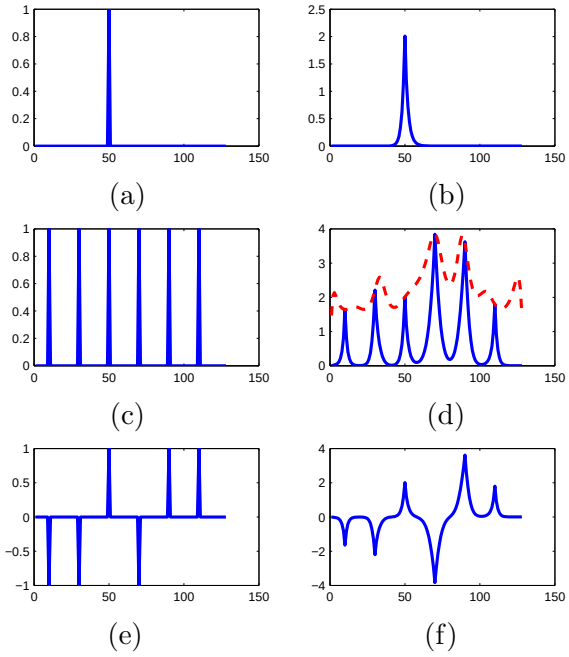


Figure 5.1. An illustration of the low-rank matrix construction for a 1D chain: (a) Single spike v_i results in (b) fast-decaying response $J^{-1}v_i$. Next, in (c) we add together several well-separated spikes, $z = \sum_{i \in c} v_i$ and in (d) show the resulting response $J^{-1}z$, which at the peaks is close to the correct variances P_{ii} (dashed). Next, we introduce random sign-flips σ_i. In (e) we plot $B_c = \sum_{i \in c} \sigma_i v_i$, and in (f) we show the response $R_c = J^{-1}B_c$.

have a column B_c of B for each color c. We assign $B_c(i) = \sigma_i = \pm 1$ i.i.d. random signs for each node i of color c, and $B_c(j) = 0$ for other nodes. An illustration appears in Figure 5.1(e), (and Figure 5.3). We assign random signs to entries of B_c in order to have destructive interference between the error terms, and we later show that it leads to unbiased variance approximations. In Figure 5.1(e) we plot a column B_c of B, and in plot (f) $R_c = J^{-1}B_c$. Next we apply B_c^T thus selecting the entries for nodes of color c. After repeating these steps for all the colors, and adding them together we get our approximation $\hat{P}$.

For rectangular-grid models the idea is very similar, we partition the nodes into several color classes such that nodes of the same color have a certain minimum distance between them. One such construction with 8 colors appears in Figure 5.2. By off-setting the blocks in a checker-board pattern the minimum distance can be increased to twice the dimension of each square. The rest of the procedure is the same as in 1D case: we assign $B_c(i)$ to be ± 1 randomly (i.i.d. flips of a fair coin) for each node i of color c, and solve $JR_c = B_c$ for all c.

For chains and lattices the nodes are easy to color by inspection. For arbitrary sparse graphs we suggest to use approximate graph-coloring to define B. To get a minimum distance l, one could augment the graph by connecting nodes up to l steps

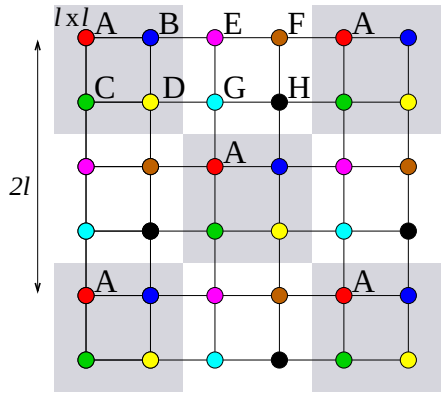


Figure 5.2. Local 2x2 regions for square lattice. Colors: $\{A, .., H\}$ with first 4 colors in shaded blocks and last 4 colors in transparent blocks. The blocks appear in a checkerboard pattern.

away, and solve the graph-coloring problem on it (assigning colors such that nodes of the same color do not share an edge). Finding an optimal coloring is very hard, but approximate solutions (allowing for some violations, and using more than the minimum number of colors) can be approached using spectral methods [4], or the max-product form of belief propagation. Upon defining the colors, we can follow the same steps as we have described for chains and grids.

Next we analyze the diagonal elements of $\hat{P}$, and show that they are unbiased and that the errors can be made arbitrarily small by increasing the minimum separation.

■ 5.1.3 Properties of the Approximation $\hat{P}$

Our construction of B can be viewed as aliasing of the columns of the standard basis I: groups of columns that correspond to nodes of the same color are added together. We refer to this process as *splicing*. It can be represented as $B = IC$. Here the c-th column C_c contains non-zero entries only for nodes of color c. The exact covariance P is the solution to linear system $JP = I$. We approximate it by solving $J\hat{P} = BB^T = ICC^TI$, i.e. $\hat{P} = J^{-1}CC^T$, and the error is

$$E = \hat{P} - P = J^{-1}(CC^T - I). \tag{5.3}$$

The matrix $(CC^T - I)$ serves the role of a signed adjacency matrix, showing which pairs of columns of I are aliased together. Let $\mathcal{C}(i)$ be the set of nodes of the same color as i, then:

$$(CC^T - I)_{i,j} = \begin{cases} \sigma_i\sigma_j, & \text{if } i \in C(j), j \neq i \\ 0, & \text{otherwise.} \end{cases} \tag{5.4}$$

We are interested in the diagonal entries of E:

$$\begin{aligned} E_{ii} = \left(P(CC^T - I)\right)_{ii} &= \sum_j P_{ij}(CC^T - I)_{ji} \\ &= \sum_{j \in C(i)\backslash i} \sigma_i \sigma_j P_{ij} = \; P_i^T \delta_{C(i)\backslash i}. \end{aligned} \tag{5.5}$$

The term $\delta_{C(i)\backslash i}$ is a signed indicator of the components aliased to i: i.e. $\delta_{C(i)\backslash i}(j) = \sigma_i\sigma_j = \pm 1$ if $j \in C(i)\backslash i$, and 0 otherwise.

Unbiased. The approximations $\hat{P}_{ii}$ are unbiased. The expectation of $\hat{P}$ over $\{\sigma_i\}$ is $\mathbb{E}_\sigma[\hat{P}_{ii}] = P_{ii} + \mathbb{E}_\sigma[E_{ii}]$. We have $\mathbb{E}_\sigma[E_{ii}] = \sum_{j \in C(i)\backslash i} P_{ij}\mathbb{E}_\sigma[\sigma_i\sigma_j] = 0$, as σ_i and σ_j are independent, and zero-mean. Hence $\mathbb{E}_\sigma[\hat{P}_{ii}] = P_{ii}$. We stress that unbiasedness involves averaging over choices of σ. However, if the variance of $\hat{P}$ is small then even one sample σ provides accurate approximations $\hat{P}$.

Variance of the approximations. Suppose that the correlations P_{ij} fall off exponentially with the distance $d(i,j)$ between i and j, i.e. $|P_{ij}| \leq A\ \beta^{d(i,j)}$, with $0 \leq \beta < 1$. This is true for a wide class of models including Markov models on bipartite graphs. Now, $\mathrm{Var}(\hat{P}_{ii}) = \mathbb{E}_\sigma[(\hat{P}_{ii} - P_{ii})^2] = \mathbb{E}_\sigma[E_{ii}^2] = \mathbb{E}_\sigma[(\sum_{j \in C(i)\backslash i} \sigma_i\sigma_j P_{ij})^2]$. We have

$$\begin{aligned} \mathrm{Var}(\hat{P}_{ii}) &= \mathbb{E}_\sigma \left\{ \left(\sum_{j \in C(i)\backslash i} \sigma_i \sigma_j P_{ij} \right)^2 \right\} \\ &= \sum_{j,j' \in C(i)\backslash i} \mathbb{E}\left\{\sigma_i^2 \sigma_j \sigma_{j'}\right\} P_{ij}P_{ij'} = \sum_{j \in \mathcal{C}(i)\backslash i} P_{ij}^2. \end{aligned} \tag{5.6}$$

In the second line we use the fact that $\sigma_i^2 = 1$, and that $\mathbb{E}\left\{\sigma_j\sigma_{j'}\right\} = 1$ if $j = j'$ and 0 otherwise.

In a 2D lattice model with our construction, the number of nodes of a given color that are $(2l)n$ steps away is $8n$ (all the distances between nodes of the same color are integer multiples of $2l$). Using the exponential decay bound, for nodes j with $d(i,j) = 2nl$, $P_{ij} = A\ \beta^{2nl}$. Hence,

$$\sum_{j \in \mathcal{C}(i)\backslash i} P_{ij}^2 \leq \sum_{n=1}^{\infty} 8nA^2\ \beta^{4nl} = 8A^2 \frac{\beta^{4l}}{(1-\beta^{4l})^2}. \tag{5.7}$$

We have used the following series: $\sum_{n=1}^{\infty} n\beta^n = \frac{\beta}{(1-\beta)^2}$. Thus, $\mathrm{Var}(\hat{P}_{ii}) \leq 8A^2\frac{\beta^{4l}}{(1-\beta^{4l})^2}$. Since, $|\beta| < 1$, we can choose l large enough such that the variance of the approximation is below any desired threshold.

Now let us repeat the analysis for 2D lattices with a slower, power-law rate of decay: i.e. $P_{ij} \leq A\ d(i,j)^{-p}$, where $p > 0$. Then the sum in (5.7) changes to:

$$\sum_{j \in \mathcal{C}(i) \setminus i} P_{ij}^2 \leq A^2 \sum_{n=1}^{\infty} \frac{8n}{(4nl)^{2p}} = \frac{8A^2}{(4l)^{2p}} \sum_n n^{1-2p}. \tag{5.8}$$

If $p > 1$, then the sum $\sum_n n^{1-2p}$ converges (and is equal to $\zeta(2p-1)$, the Riemann zeta function), and the errors can be made arbitrarily small by increasing l. However, if $p \leq 1$, then for any l the sum diverges[5]. In Section 5.2 we show that the wavelet-based construction can dramatically reduce the errors for such power-law decay, and can go beyond these limitations.

Remarks. In general there is a tradeoff concerning the size of the local region – one could pick a small local region, leading to high variance of $\hat{P}$, and average over many choices of σ_i. Alternatively, one could pick a larger local region, leading to small variance of $\hat{P}$ and average over few choices of σ_i (or not average at all). The second approach is more effective, as the variance decreases exponentially with separation length, while only as $\frac{1}{T}$ with T repeated experiments. Hence, we suggest that in practice l should be chosen to be comparable to the correlation length of the model. However, in case the correlation length is not known exactly, repeated trials over choices of random signs can also be useful – they can be used to obtain empirical variances of $\hat{P}$.

We can also bound the absolute error itself (rather than its variance): $|E_{ii}| \leq \sum_{j \in \mathcal{C}(i) \setminus i} |P_{ij}|$. For example, with exponential decay of P_{ij}, we have $|E_{ii}| \leq 8A \frac{\beta^{2l}}{(1-\beta^{2l})^2}$. The stochastic bound on $\mathbb{E}_\sigma[E_{ii}^2]$ in (5.6) is tighter, but the deterministic one does not involve expectation over the random signs σ. Hence, the two bounds are not redundant.

■ 5.2 Constructing Wavelet-based B for Models with Long Correlation

In our construction of matrix B in the last section we set the separation length between nodes of the same color to be comparable to the correlation length in the model. When the correlation length is short, the approach is very efficient. However, when the correlation length is long the approach is no longer attractive: making the separation length long will make the computational complexity high. Alternatively, if we violate the correlation length and use a short separation, then the method still gives unbiased variance approximations, but the variance of these variance approximations becomes very high (see examples in Section 5.3).

To address long-range correlations we propose using wavelets to decompose the aliasing matrix B across several scales, so that the correlation length in each scale is short. Note that in this section the GMRF model has just *one scale*. Multiple scales

[5]Here we are focusing on 2D models. More generally, the required p depends on the dimension of the lattice. In d dimensions, there are $O(n^{d-1})$ aliased terms at distance n, and the sum in (5.8) becomes $\propto \sum n^{(d-1)-2p}$. Thus, we need $p > d/2$ for convergence.

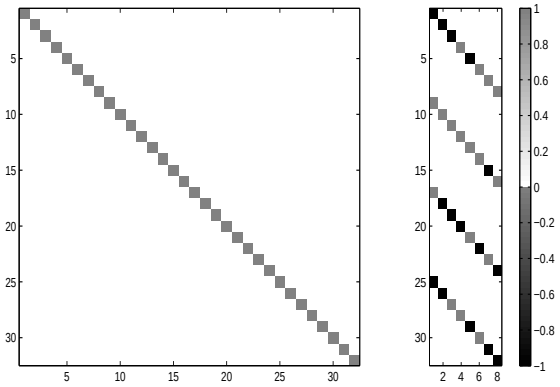


Figure 5.3. (left) identity and (right) locally orthogonal B matrix formed by adding certain columns of I together and changing signs randomly.

come from the wavelet decomposition. In Section 5.2.3 we apply the method to a *multi-scale model*, where the GMRF has hidden variables representing coarser scales and allows sparse representation of processes with slow correlation fall-off.

We start with one-dimensional wavelets in continuous time to simplify discussion and analysis. We briefly review the basics of wavelet decompositions mainly to set notation. A wavelet decomposition is specified by a scaling function $\phi(t)$ and a wavelet function $\psi(t)$, which generate a family of dilations and translations [89]:

$$\begin{aligned} \phi_{s,k}(t) &= \frac{1}{2^{s/2}}\phi(2^{-s}t - k), \\ \psi_{s,k}(t) &= \frac{1}{2^{s/2}}\psi(2^{-s}t - k). \end{aligned} \tag{5.9}$$

For a fixed scale s, the set $\{\phi_{s,k}(t)\}_k$ generates the approximation space $\mathcal{V}_s$. These spaces $\mathcal{V}_s$ are nested: $\mathcal{V}_1 \supset \mathcal{V}_2 \supset \mathcal{V}_3$ ..., with higher s corresponding to coarser scales. The span of the wavelets $\{\psi_{s,k}(t)\}_k$ at a given scale s gives the detail space $\mathcal{W}_s = \mathcal{V}_{s-1} \ominus \mathcal{V}_s$ (we use $\ominus$ to denote the orthogonal complement of $\mathcal{V}_s$ in $\mathcal{V}_{s-1}$). We can decompose the fine scale $\mathcal{V}_1$ over N_{sc} scales:

$$\mathcal{V}_1 = \mathcal{W}_1 \oplus \mathcal{W}_2 \oplus ... \oplus \mathcal{W}_{N_{sc}} \oplus \mathcal{V}_{N_{sc}}. \tag{5.10}$$

We focus on orthogonal[6] wavelet families with compact support where $\psi_{s,k}(t)$ is orthogonal to all other translations and dilations of $\psi(t)$, and to scaling functions at scale s and coarser.

To deal with discrete-time signals, we make the standard assumption that discrete samples f_k are the scaling coefficients $\langle \phi_{s_1,k}, f(t) \rangle$ of a continuous wavelet transform of

[6]One could also use biorthogonal wavelets [89] in our approach: instead of having an orthogonal wavelet basis W, we would have an analysis basis W_a and a synthesis basis W_s, such that $W_a W_s^T = I$.

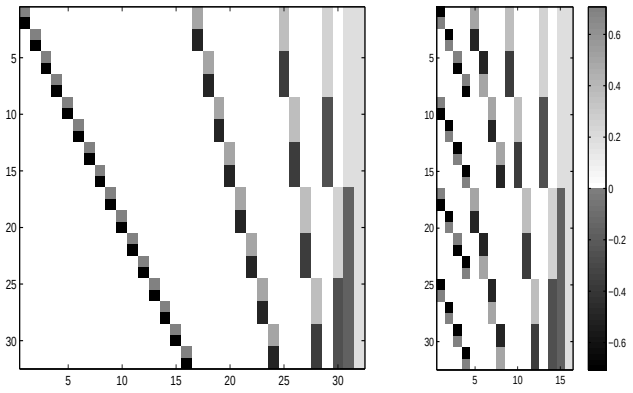


Figure 5.4. (left) A discrete wavelet basis, with columns corresponding to wavelets at different scales and translations, and (right) B matrix obtained by aliasing certain columns of W within each scale. In the wavelet basis W the number of columns doubles with each finer scale, but in B it stays constant.

some smooth function $f(t)$ at scale s_1 [89]. Let $s_1 = 1$ without loss of generality. Now, a discrete wavelet basis for the space $\mathcal{V}_1$ is constructed by collecting the scaling functions at the coarsest scale, and the wavelet functions at all finer scales as columns of a matrix W. Let S^s and W^s contain the scaling and wavelet functions, respectively, at scale s. In general we do not need to go all the way to the coarsest scale $N_{sc} = \log_2(N)$. Stopping the decomposition earlier with $N_{sc} < \log_2(N)$ also provides an orthogonal basis for the space $\mathcal{V}_1$. Our orthogonal basis is[7]:

$$W = [W^1 \; W^2 \; .... \; W^{N_{sc}-1} \; S^{N_{sc}}]. \tag{5.11}$$

An illustration of a Haar wavelet basis for $N = 32$ is given in Figure 5.4 (left). Columns (wavelets) are grouped by scale, and horizontal axis corresponds to translation. At scale s we have $2^{N_{sc}-s}$ possible translations, hence that many columns in W^s.

■ 5.2.1 Wavelet-based Construction of B

There is now a well-established literature [45,47,50,90] describing that for many classes of random processes their wavelet coefficients have faster decaying correlation than the original process itself. In our approach we do not transform the random process – instead, we consider solutions R_k to $JR_k = W_k$ (W_k is a column of W), and show that R_k exhibits fast decay (we also say correlation decay), which will allow compression of B and computational efficiency. Roughly speaking, we create a scale-dependent B, with

[7]Ideally one would use boundary wavelets at the edges of the signal [89]. We do not pursue this: we use $N_{sc} < \log_2(N)$, and assume that the support of the wavelets at the coarsest scale N_{sc} is small compared to the size of the field, and hence edge-effects have negligible impact in our approach.

a construction similar to Section 5.1 at each scale. We now present our wavelet-based construction, and then analyze it.

In the original single-scale construction we find an approximation $\hat{P}$ to P by solving $J\hat{P} = BB^T$ instead of $JP = II^T = I$. The matrix B is an aliased version of I, with $B = IC$. For the multi-scale construction we start by expressing the exact covariance as the solution to the system $JP = WW^T = I$. We approximate it by applying the aliasing operation at each scale, $B^s = W^s C^s$ (note, we do not alias wavelets across scales). We call this aliasing operation *wavelet splicing*. The k-th column of W^s contains $\psi_{s,k}(t)$, and corresponds to the k-th wavelet at scale s. We group these coefficients, and hence, the columns, into M^s groups (colors) such that any two coefficients of the same color are well separated with respect to the correlation length at scale s (i.e. correlation length for R_k at scale s). Each column of C^s contains non-zero entries only for nodes of a particular color. Similar to Section 5.1, we set $C^s_c(k) = \sigma^s_k = \pm 1$, for $k \in c$, and 0 otherwise. The signs σ^s_k are equiprobable and i.i.d. Combining all the scales together, this gives:

$$B = WC, \tag{5.12}$$

where $B = [B^1, ...B^{N_S}]$, $W = [W^1, ..., W^{N_{sc}-1}, S^{N_{sc}}]$, and $C = \text{blockdiag}([C^1, ..., C^{N_{sc}}])$. We illustrate matrices W and B in Figure 5.4 (left) and (right) respectively. The rest of the procedure follows the one for the short correlation length: we solve for the diagonal of $\hat{P}$ using $J\hat{P} = BB^T$, as described in Section 5.1.

In the wavelet decomposition, the majority of the coefficients are at fine scales. In the next section we describe that for well-behaved GMRFs R_k decays faster at finer scales[8]. While at the finer scales in W there are more coefficients (and columns), they can be aliased together more aggressively, see Figure 5.4 (right). We show that under certain assumptions the correlation length can be assumed to decrease two-fold with each finer scale, so the resulting number of columns of B^s stays the same for all scales. In this manner, the number of columns of B is $O(\log_2(N))$ instead of N for the wavelet basis W, giving significant computational savings in our approach.

Construction of B for 2D. We use the separable wavelet construction, which takes products of 1D functions to create a family of two-dimensional triplets [89][9]:

$$\begin{aligned}
\psi^{(1)}_{s;k_1,k_2}(x,y) &= \phi_{s,k_1}(x)\psi_{s,k_2}(y), \\
\psi^{(2)}_{s;k_1,k_2}(x,y) &= \psi_{s,k_1}(x)\phi_{s,k_2}(y), \\
\psi^{(3)}_{s;k_1,k_2}(x,y) &= \psi_{s,k_1}(x)\psi_{s,k_2}(y).
\end{aligned} \tag{5.13}$$

[8] We measure distance and separation relative to scale: separation of K at scale s corresponds to separation of $K2^{s-1}$ at scale 1.

[9] This is different from taking outer products between each pair of columns of W in (5.11). It would also give an orthogonal basis, but has the undesirable effect of mixing wavelets from different scales.

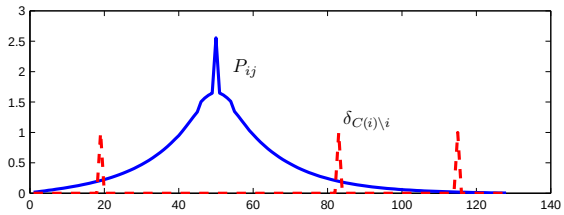


Figure 5.5. Errors with the aliased standard basis: the error is obtained by an inner product between P_i and $\delta_{C(i)\setminus i}$ (both signals are a function of j). Here $i = 50$. We set all the signs $\sigma_j = 1$ for simplicity.

Stacking $\psi^{(i)}_{s;k_1,k_2}$ as columns of a matrix creates an orthogonal basis $\bar{W}$ for two-dimensional fields. To produce the corresponding aliasing matrix $\bar{B}$ as in (5.12), we first create one-dimensional spliced matrices $B^s = W^s C^s$ and $\tilde{B}^s = S^s C^s$ containing linear combinations of wavelet and scaling functions at each scale. Then we create triplets using columns of B^s and $\tilde{B}^s$ in the same manner as in (5.13).

■ 5.2.2 Error Analysis.

In Section 5.1.2 we have analyzed the errors in the single scale construction, $E_{ii} = P_i^T \delta_{C(i)\setminus i}$. When the separation between nodes of the same color is smaller than the correlation length, the errors are significant (see Figure 5.5). We will now justify why the wavelet construction can dramatically reduce the errors for models with long-range correlations.

The variance approximation in the wavelet-based construction of B is $\hat{P} = J^{-1}BB^T = J^{-1}WCC^TW^T$. The aliasing matrix C is block diagonal with a block for each scale. Let $R_W = J^{-1}W$. Its k-th column $R_k = J^{-1}W_k$ is the response of the linear system $JR_k = W_k$ to the wavelet W_k. An illustration appears in Figure 5.6. We show the response $R_k = PW_k$, for wavelets W_k at two different scales, $s = 6$ and $s = 5$. We also show the wavelets W_l that are aliased to k with dashed lines. It is clear that R_k decays much faster than P_i in Figure 5.5. We discuss this decay in more detail later in this section. The regions where R_k and W_l overlap contribute to the errors in $\hat{P}_i$ for i falling in the support of W_l. The error is:

$$E = \hat{P} - P = J^{-1}W(CC^T - I)W^T = R_W(CC^T - I)W^T. \tag{5.14}$$

We have $(CC^T - I)_{k,l} = \sigma_k\sigma_l = \pm 1$ only if $k \neq l$ and the wavelets W_k and W_l are aliased together. In particular, if k and l belong to different scales, then $(CC^T - I)_{k,l} = 0$. Now the errors in variances are:

$$\begin{aligned} E_{ii} &= \sum_k \sum_l R_{ik}(CC^T - I)_{kl}W_{il} \\ &= \sum_k \sum_{l \in C^s(k)\setminus k} \sigma_k\sigma_l R_{ik}W_{il}. \end{aligned} \tag{5.15}$$

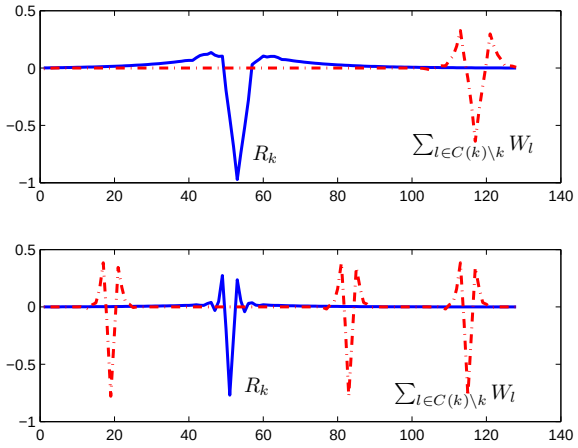


Figure 5.6. R_k, and W_l for $l \in C(k)\backslash k$. (top) Scale 6. (bottom) Scale 5. Regions where R_k and W_l overlap contribute to the errors in $\hat{P}_i$ for i in the support of W_l. The original P_i is shown in Figure 5.5

We will analyze $\mathbb{E}_\sigma[E_{ii}^2]$ in Proposition 5.2.1, and show that the interference of R_k and W_l decays fast with separation. We will show that for large fields, as $N \to \infty$ the error is stable (i.e. bounded), the approximation can be made accurate to any desired level by controlling aliasing, and that the multi-scale construction is much more accurate than the single-scale one for GMRFs which have substantial energy over multiple scales.

We can also bound $|E_{ii}|$ (and hence the ℓ_∞-norm of $e \triangleq \text{diag}(E)$, $\|e\|_\infty = \max_i |E_{ii}|$):

$$\begin{aligned} |E_{ii}| &= \left| \sum_k \sum_{l \in C^s(k)\backslash k} \sigma_k \sigma_l R_{ik} W_{il} \right| \\ &\leq \sum_k \sum_{l \in C^s(k)\backslash k} |R_{ik}||W_{il}| \end{aligned} \tag{5.16}$$

This bound is not as tight as the stochastic one we will get in (5.19), but on the other hand, it does not require taking expectations over the random signs σ.

Correlation decay. We now analyze the decay of $R_k(i)$. Note that, while there are similarities of spirit in this analysis and other work involving wavelets and covariance matrices, our objectives and indeed our analysis differ in significant ways. In particular, conventional analysis focuses on the covariance matrix of the wavelet coefficients, i.e., $P_W = W^T P W$. In contrast, our analysis is based on viewing the rows of P as deterministic signals and considering their transforms – i.e., on the matrix $R_W = PW$. That said, we will comment on possible ties to more conventional wavelet analysis at the end of this section.

We first recall some relevant facts from wavelet analysis [89]. Suppose a continuous-time function $f(t)$ is α-Lipschitz[10] (this is related to how many times $f(t)$ is continuously differentiable). Also suppose that the wavelet family $\psi_{s,k}(t)$ has m vanishing moments[11], with $m > \alpha$. Then the wavelet coefficients $Wf(s,k) = \langle \psi_{s,k}(t), f(t) \rangle$ satisfy $|Wf(s,k)| = O(2^{s(m+1/2)})$. If $m \geq 1$ then the magnitude of the wavelet coefficients in smooth regions drops fast for each finer scale.

However, this fast decay does not happen near a point of singularity of $f(t)$, say t_0. Suppose that the wavelet at scale 1 has support K. At a coarser scale, s, the support is $K2^{s-1}$. To avoid the point of singularity, the wavelet at scale s has to be outside the interval $t_0 \pm K2^{s-1}$, which gets twice wider with each coarser scale. This set over all scales is called the 'cone of influence', and it contains unusually high values of wavelet coefficients, a region of disturbance caused by the singular point [89].

For our analysis, we view P as samples of a continuous-time function, and assume that the correlation function P_{ij} may have a singularity at $i = j$, and that it is smooth otherwise. Consider scale s, $R^s = PW^s$. The i-th row of R^s contains the scale-s wavelet coefficients of the i-th row of P. The singularity of P_{ij} at $i = j$ will produce a disturbance region with high wavelet coefficients near that value of k for which W_k peaks at row i. Recall that the rows of R^s are indexed by nodes, and the columns correspond to wavelet coefficients. The disturbance region at node i in R^s will be roughly $K2^s$ rows wide, and K columns wide (since wavelet coefficients involve downsampling by 2^s). When columns of W_s are aliased together, we have to make sure that the cones of influence do not overlap. The region of disturbance is twice narrower (in terms of the number of rows) at each finer scale, so roughly twice as many wavelets can be aliased with each finer scale.

As an illustration consider Figure 5.6. The region of disturbance of $R_k(i)$ near $i = 50$ can be seen in Figure 5.6 for scales 6 and 5. The original P_i is shown in Figure 5.5 and has a singularity at $i = 50$. It is evident that by going to a finer scale, from $s = 6$ to $s = 5$, $R_k(i)$ decays faster, and more columns of W can be aliased without sacrificing the accuracy.

Properties of the wavelet-based approximation $\hat{P}$. In the single-scale case we showed that $\hat{P}$ is unbiased, and bounded the variance of the errors. We extend these results to our wavelet-based approximation. The total error is equal to

$$E = P - \hat{P} = P\left(WW^T - BB^T\right). \tag{5.17}$$

Unbiased. Let $\mathcal{C}(k)$ be the set of columns that get merged with column k. Then taking an expectation over $\{\sigma_k\}$, $\mathbb{E}_\sigma[BB^T] = WW^T + \sum_k \sum_{l \in \mathcal{C}(k) \setminus k} W_k W_l^T \mathbb{E}_\sigma[\sigma_k \sigma_l] = WW^T$.

[10] A function is pointwise α-Lipschitz [89] at t_0 if there exists $\gamma > 0$, and a polynomial p_v of degree $m = \lfloor \alpha \rfloor$ such that, $\forall t \in \mathbb{R}$, $|f(t) - p_v(t)| \leq \gamma |t - t_0|^\alpha$, $(\alpha > 0)$. It is uniformly Lipschitz over an interval if it is pointwise Lipschitz with γ not dependent of t.

[11] A wavelet with n vanishing moments is orthogonal to polynomials of degree $n - 1$, i.e. $\int_{-\infty}^{\infty} t^k \psi(t) dt = 0$ for $0 \leq k < n$.

The error terms cancel out because $\mathbb{E}_\sigma[\sigma_k\sigma_l] = 0$ for $k \neq l$. Thus, the approximation $\hat{P}$ is *unbiased*.

Variance of the approximations. We now obtain a bound based on the expression in (5.15). Since $\hat{P}$ is unbiased, we have $\mathrm{Var}(\hat{P}_i) = \mathbb{E}_\sigma[E_{ii}^2]$. Using (5.15) it follows:

$$\mathbb{E}_\sigma[E_{ii}^2] = \mathbb{E}\left[\left(\sum_l \sum_{k\in C(l)\setminus l} \sigma_k\sigma_l R_{ik}W_{il}\right)^2\right]. \tag{5.18}$$

The terms $\sigma_k\sigma_l$ and $\sigma_{k'}\sigma_{l'}$ are uncorrelated unless $(k,l) = (k',l')$ or $(k,l) = (l',k')$, so this expectation reduces to $\mathbb{E}_\sigma[E_{ii}^2] = \sum_l \sum_{k\in C(l)\setminus l} R_{ik}^2 W_{il}^2 + \sum_l \sum_{k\in C(l)\setminus l} R_{ik}W_{il}R_{il}W_{ik}$. Also, the second term is zero, as we require that the supports of the aliased terms W_l and W_k do not overlap, i.e. $W_{il}W_{ik} = 0$ for $k \in C(l)\setminus l$. Hence,

$$\mathbb{E}_\sigma[E_{ii}^2] = \sum_l \sum_{k\in C(l)\setminus l} R_{ik}^2 W_{il}^2. \tag{5.19}$$

To bound this sum we consider a model with exponential and power-law decay of correlations, and assume that the wavelet has m vanishing moments. Also, we *do not* use $N_{sc} = \log_2(N)$ scales in the decomposition, but rather set $N_{sc} \propto \log_2(L)$, where L is the correlation length of the model. Once the size of the field exceeds L, there is no advantage in including coarser scales that contain negligible energy.

Proposition 5.2.1 (Bounded errors). *Suppose for a 1D GMRF, $P_{ij} \sim \beta^{d(i,j)}$ or $P_{ij} \sim d(i,j)^{-p}$. Then, as the size of the field tends to infinity, the errors in (5.19) stay bounded, provided that the number of vanishing moments of the wavelet function satisfies $m \geq 1$. Also, by increasing the separation length, i.e. the distance between nearmost aliased terms, the errors can be made arbitrarily small.*

We establish this stability property in Appendix A.3. We avoid the issue of boundary effects as we fix the number of scales of the wavelet decomposition when the field size tends to infinity. In the appendix we show that the errors in the wavelet-based construction can be much smaller than in the single-scale one, if the GMRF has power distributed over multiple scales. For higher-dimensional lattices with power-law rate of decay, the required number of vanishing moments also has to satisfy $m + p > \frac{d}{2}$, where d is the dimension.

Alternative variance analysis. We also consider another line of analysis which makes ties to covariances of wavelet coefficients $P_W^s \triangleq (W^s)^T P W^s$ (rather than $R^s = PW^s$). It is important to emphasize that this analysis is approximate and does not lead to

bounds. Consider $\mathrm{tr}(E)$, and decompose it by scale. We have:

$$\begin{aligned}\mathrm{tr}(E_s) = \mathrm{tr}\left[P(W^s(W^s)^T - B^s(B^s)^T)\right] = \\ \mathrm{tr}\left[(W^s)^T P W^s - (B^s)^T P B^s\right] = \\ \mathrm{tr}\left[(W^s)^T P W^s - (C^s)^T (W^s)^T P W^s C^s\right] = \\ \mathrm{tr}\left[P_W^s (I - C^s (C^s)^T)\right]. \quad (5.20)\end{aligned}$$

Then, via the same analysis as in Section 5.1.3 we have:

$$\mathrm{Var}\left(\mathrm{tr}\left[P_W^s (I - C^s (C^s)^T)\right]\right) = \sum_k \sum_{l \in \mathcal{C}(k) \setminus k} ((P_W^s)_{k,l})^2. \quad (5.21)$$

Putting all the scales together, $\mathrm{Var}(\mathrm{tr}(E)) = \sum_s \sum_k \sum_{l \in \mathcal{C}(k) \setminus k} ((P_W^s)_{k,l})^2$. This equality holds since the signs σ_k at different scales are independent. Now, assuming that the errors at different nodes are only weakly correlated, which we justify with experiments in Section 5.3, we have $\sum_i \mathrm{Var}(E_{ii}) \approx \mathrm{Var}(\sum E_{ii}) = \mathrm{Var}(\mathrm{tr}(E)) = \sum_s \sum_k \sum_{l \in \mathcal{C}(k) \setminus k} ((P_W^s)_{k,l})^2$. We obtain an estimate of the variance of our approximation that explains how the errors are decomposed across scale. The accuracy of this approach relies on more detailed knowledge of the structure of the covariance than the bound we have presented earlier. That said, since the statistics of wavelet coefficients of various random processes have been analyzed in prior work [45, 47, 50, 90], there are certainly classes of processes in which this alternate variance approximation can be quite accurate. Moreover, taking advantage of such additional knowledge of covariance structure may suggest alternative bases to W, and in turn to B, that are adapted to the process structure and yield tighter bounds[12].

■ 5.2.3 Multi-scale Models for Processes with Long-range Correlations

In our analysis the errors in variance approximations mainly depend on the covariance structure of P, and the information matrix J does not play a direct role. However, J plays a crucial role during estimation – the model has to be Markov with respect to a sparse graph to be able to store it efficiently, and to solve the linear system $J\mu = h$ efficiently. Some processes with slow correlation fall-off do not have a sparse information matrix in a one-scale representation, so they do not fit well into our approach. However, slow correlation fall-off can be modeled using sparse multi-scale representations with hidden variables, as we discussed in Section 2.3.2. A pyramidal model with a stochastic relationship between scales was proposed in [31,32]. We consider the problem of finding approximate variances in such a model.

A representative structure for the model is illustrated in Figure 5.7 for both 1D and 2D fields. The variables in the bottom (fine) scale correspond to some physical phenomenon that is being modeled. The variables at coarser scales represent aggregates

[12]For example, one could use partial wavelet decompositions that stop at intermediate scales, and more generally wavelet packets [89] adapted to the statistics of wavelet coefficients at different scales.

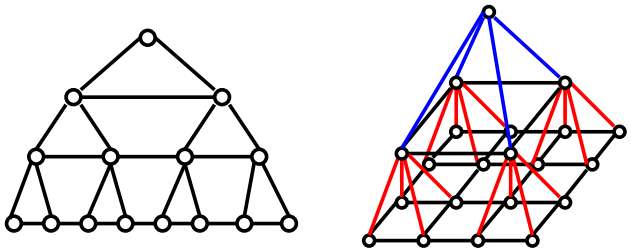


Figure 5.7. (left) 1D multi-scale model (right) 2D multi-scale model (colors serve to make the plot better interpretable by distinguishing inter and intra scale edges).

over local regions. They may or may not be of interest in the estimation, but they serve to induce a sparse graph structure (once they are integrated out, the fine-scale model in general has a complete, non-sparse, information matrix). Aggregation can mean that the coarser scale variable represents an average, or some weighted combination of the variables in the finer scale over a small region. However, the relationship across scale is non-deterministic, allowing for uncertainty. The graph is sparse, but has many loops.

The structure of the information matrix is such that variables in one scale are only connected to nearby scales. Hence the J matrix for a multi-scale model with 4 scales has the following chain structure (with scale 1 being the finest, and scale 4 – the coarsest):

$$J = \begin{pmatrix} J_1 & J_{12} & & \\ J_{21} & J_2 & J_{23} & \\ & J_{32} & J_3 & J_{34} \\ & & J_{43} & J_4 \end{pmatrix}. \tag{5.22}$$

Suppose that we are mainly interested in computing the variances of the variables at the finest scale (the other ones are auxiliary), i.e. in the block of J^{-1} corresponding to scale 1. Hence in our approach we only need to approximate blockdiag$(I, 0, 0, 0)$, and not the full I matrix. We use the matrix $B = \left(\begin{smallmatrix} B_1 \\ 0 \end{smallmatrix}\right)$, with 0 for all coarser scales[13]. Here B_1 is a spliced wavelet basis corresponding to variables at scale 1 (the same construction as in Section 5.2).

Our error analysis takes into account only the covariance structure of the fine scale variables, $P_1 = [J^{-1}]_1$. Hence, it is oblivious to the hidden variables representation, and only depends on the properties of the marginal covariance block P_1. Experimental results with this multi-scale model for processes with long-range correlations are presented in Section 5.3.

[13] Alternatively, if the variances at coarser scales are of interest, we use the matrix blockdiag(B_1, B_2, B_3, B_4), where B_i is a spliced wavelet bases corresponding to scale i. The errors are decoupled: errors from B_i at scale i are not propagated to other scales.

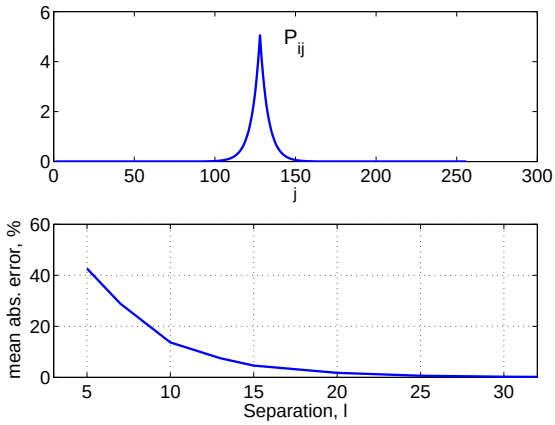


Figure 5.8. (top) Correlation P_{ij} from the center node. (bottom) Errors in variances (mean absolute error, in percent) vs. separation length l.

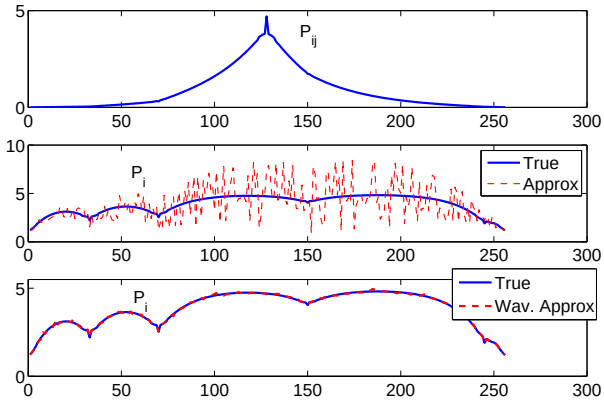


Figure 5.9. 1D example with long-correlation. (top) Correlation P_{ij} from the center node. (center) True variance, and low-rank approximate variance using one scale. (bottom) True variance, and low-rank wavelet-based approximate variance.

■ 5.3 Computational Experiments

Our first experiment involves a 1D thin-membrane model with length $N = 256$, with nearest neighbor connections. Noisy observations are added at a few randomly selected nodes. This model has a short correlation length, see Figure 5.8 (top). We apply the single-scale low-rank method from Section 5.1.2, and we plot the errors in variances (absolute error in percent, averaged over all nodes) versus the separation length in Figure 5.8 (bottom). The errors decay fast with separation length, in line with our analysis in Section 5.1.3.

Next we consider a 1D thin-membrane model with connections from each node to nodes up to 4 steps away. The J matrix is close to singular, and the correlation

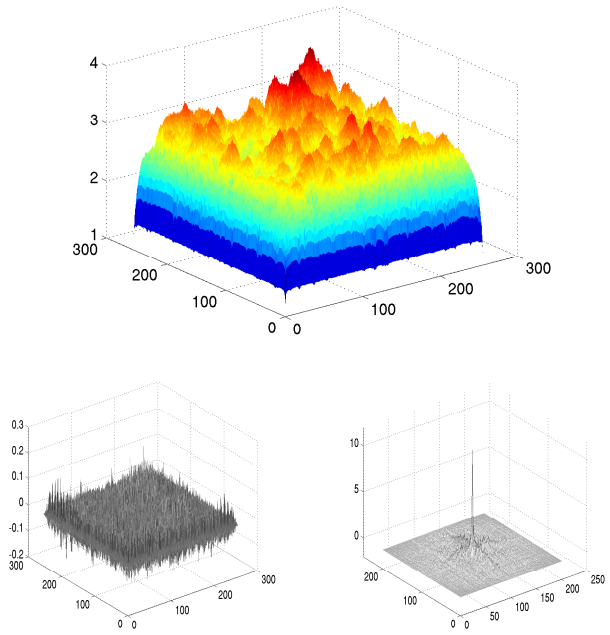


Figure 5.10. 2D thin-membrane example: (top) approximate variances, (bottom left) errors, and (bottom right) 2D auto-correlation of errors. The approximations are accurate (errors are much smaller than the variances), and the errors are weakly correlated.

length in the model is long, see Figure 5.9 (top). We illustrate the results using both the single-scale (middle plot) and the wavelet-based (bottom plot) low-rank methods. We use $M = 32$ for the single-scale approach, which is too small compared to the correlation length. While the approximation is unbiased, its high variance makes it practically useless. For the wavelet-based case, using a smaller matrix B with $M = 28$, constructed by splicing a Coifman wavelet basis (coiflet basis) [42], we are able to find very accurate variance approximations as seen in Figure 5.9 (bottom).

Next we apply the approach to a 2D thin-membrane model of size 256×256, with correlation length about 100 pixels, and with sparse noisy measurements taken at randomly selected locations. The underlying true field is flat. We use separable Coifman wavelets, and the resulting sparse B matrix has size 65536×304. This is a very significant reduction in the number of columns, compared to W. The results appear in Figure 5.10: the errors (bottom left) are small compared to the variances (top). Our approximate solution is a close match to the exact solution, which can still be computed for models of this size. The 2D auto-correlation of the errors appears in Figure 5.10 (bottom right): the errors are weakly correlated, supporting our alternative error analysis based on P_W in Section 5.2.2.

Next, we apply our low-rank variance approximation method to ocean surface height

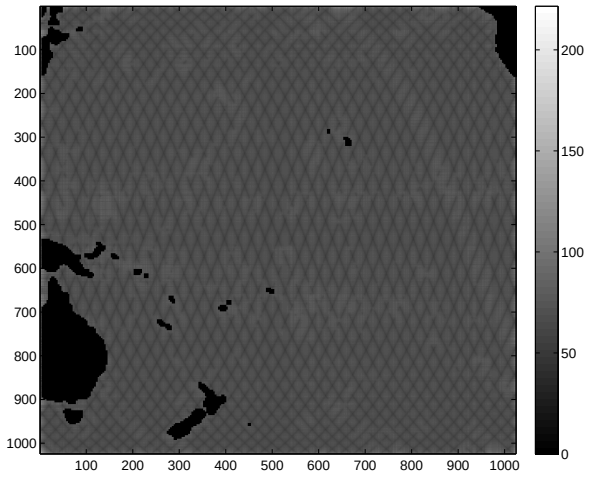


Figure 5.11. Approximate uncertainty (mm) of Pacific ocean surface height based on measurements along satellite tracks, 1024×1024 grid.

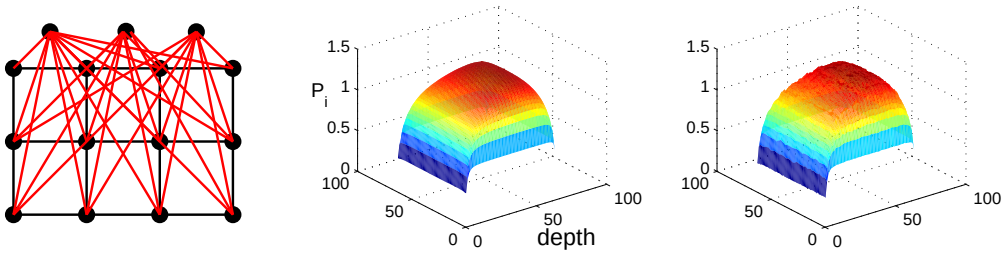
data collected along the tracks of Jason-1 satellite[14] over the Pacific ocean region. The data are sparse and highly irregular. We use the thin-plate model for ocean surface height[15]. The measurements in general fall between the grid points, and they are modeled as bilinear interpolation $y_k = h_k x + n_k$, of the nearest 4 nodes in the grid (h_k has 4 non-zero entries) with added white Gaussian noise $n_k \sim \mathcal{N}(0, \gamma)$. The posterior information matrix combining the thin-plate prior J_{tp} with the measurements is $J = J_{tp} + \frac{1}{\gamma} H^T H$, and it is still sparse because the measurements only induce local connections within each cell in the grid.

The size of the field is 1024×1024, i.e. over a million variables. Computing the variance in a model of this size is beyond what is practical with exact methods on a single workstation. We use our approximate variance calculation method. The correlation length is moderate, so using just 2 wavelet scales suffices, and the B matrix has only 448 columns. The resulting approximate variances using a version of the embedded trees (ET) iterative solver described in Section 5.4, appear in Figure 5.11. The regions over land are ignored (in black). The variances are lowest near the measurements (along the tracks) as expected.

Next, we consider a gravity inversion problem, where one is interested in estimating the underground geological structure of a 3D-volume based on gravity measurements

[14]This altimetry dataset is available from the Jet Propulsion Laboratory http://www.jpl.nasa.gov. It is over a ten day period beginning 12/1/2004. The altimetry data are normalized to remove seasonal spatially varying average sea levels.

[15]We refer to [76] for how to chose the parameters balancing the prior and the likelihood in this model.



(a) (b) (c)

Figure 5.12. Gravity inversion example: (a) Markov graph of the posterior density has edges from the thin-plate model and edges induced by observations (b) exact variances (c) accurate approximate variances using the wavelet-based low-rank approach. The variances increase with depth.

on its surface. As we now describe, this problem is more general than what we have considered so far, due to sparse but non-local measurements. For simplicity we consider a 2D version of this problem. We divide the 2D subsurface region into small blocks, and model the mass x in the blocks as a thin-plate GMRF. The gravity measurements y on the surface come from a discretization of Newton's law of universal gravitation. They are linear in the unknowns x, but non-local – they couple all the nodes in the GMRF:

$$y_i = G \sum_j u_{ij}\ x_j / d_{ij}^2 + n_i. \tag{5.23}$$

Here y_i is the 2-component (horizontal and vertical) gravity measurement at point i on the surface, x_j is the unknown mass at the j-th subsurface node, see Figure 5.12(a). Also, G is the gravitational constant, d_{ij} and u_{ij} are respectively the distance and the unit vector from the location of the j-th node to i-th measurement point, and n_i is Gaussian noise with diagonal covariance Q. Combining the linear measurement model $y = Hx + n$ with the thin-plate prior J_{tp} for the unknown field x, the posterior variance that we would like to approximate is:

$$P = (J_{tp} + H^T Q^{-1} H)^{-1}. \tag{5.24}$$

Note that this problem does not simply correspond to a sparse matrix J: in addition to the sparse component J_{tp} there is also a low-rank non-sparse component $H^T Q^{-1} H$ due to the non-local measurements. However, using a version of block Gauss-Seidel described in Section 5.4, we still obtain fast solution of the resulting linear system. We consider a square region with 64×64 nodes, with gravity measurements at 64 locations at the top[16]. We plot the true variances, and the ones obtained using a wavelet-based low-rank approach with 4 scales and 206 columns of B (instead of 4096). Despite the

[16] We assume that the density is approximately known outside the square region (this is not required, but it simplifies the problem).

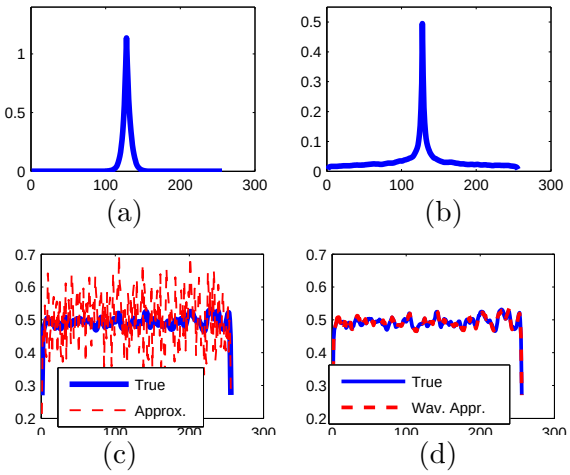


Figure 5.13. Multi-scale example: (a) conditional and (b) marginal correlation at the fine scale. (c) approximate variances using the low-rank approach: spliced standard basis. (d) accurate approximate variances using the wavelet-based low-rank approach.

long-range correlation induced by the observation model, and the addition of the non-sparse $H^T Q^{-1} H$ term, the method still gives accurate variances, as we show in Figure 5.12.

Finally, we apply our reduced-rank approach to a multi-scale model on a pyramidal graph as described in Section 2.3.2. The model has 256 variables in the finest scale, and 5 coarser levels, with the number of variables decreasing two-fold for each coarser level. The total number of variables is 496. In Figure 5.13 (a) we show the fast-decaying conditional correlation at the fine scale (conditioned on the coarser scales), and in plot (b) the slow-decaying marginal correlation at the fine scale. The fast decay of conditional correlations allows efficient solutions of the linear systems in our approach. However, the errors in our low-rank variance approximations depend on the long-range marginal correlations, requiring the use of the wavelet-based approach. In Figure 5.13 we show the results of computing approximate variance using the single-scale approach in plot (c), and the wavelet-based approach in plot (d). The sizes of the resulting aliasing matrices B are 496×32 and 496×28 respectively. It can be seen that the single-scale approach is inadequate, while the wavelet-based B yields very accurate variances, even though it uses an aliasing matrix B with fewer columns. This is as expected – the model has a long marginal correlation length at the fine scale, which only the wavelet-based approach is able to handle.

■ 5.4 Efficient Solution of Linear Systems

In our approach we compute the variances by solving a small number, $M \ll N$, of linear systems $JR_i = B_i$ all sharing the same matrix J. Whenever a fast solver for J is available, the overall variance approximation scheme is also fast.

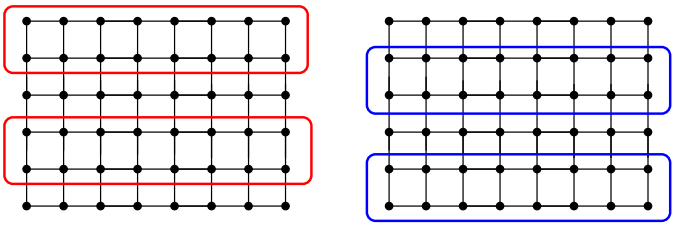


Figure 5.14. Block Gauss-Seidel with thin induced trees. Exact estimation is linear in the length of the strip using block-LBP.

Iterative approaches such as Richardson iterations and conjugate gradient methods are very appropriate for our approach, as multiplication by a sparse J is very efficient, so the cost per iteration is low. The number of iterations can be controlled by using a good preconditioner for J, one which is easy to evaluate, and which serves as an approximation of J^{-1}.

An efficient set of preconditioners based on embedded trees (ET) has been developed in [120] for the lattice GMRF model. The idea is that for models with a tree-structured graph G, solving the system $J\mu = h$ (i.e. applying J^{-1} to a vector) is highly efficient – it can be done in $O(N)$ operations. Hence, for general graphs G, [120] uses spanning trees $T \subset G$ with preconditioner J_T^{-1}. We use a similar strategy based on block Gauss-Seidel iterations that uses thin induced[17] subgraphs as blocks. We partition the lattice into narrow overlapping horizontal and vertical strips. Estimation in the strip (conditioned on other variables being fixed) can be done efficiently with the cost linear in the length and cubic in the width of the strip. By iterating over the strips convergence to the correct means is guaranteed[18]. An illustration of this procedure appears in Figure 5.14, where we divided all horizontal strips into two groups, such that strips do not overlap within the group. This way estimation can be performed for all the strips in a group in parallel. We use this approach in the experiments in Section 5.3.

There are several directions for designing potentially even more efficient preconditioners. Recently, [26] proposed an adaptive scheme based on ET that picks the spanning trees adaptively to have the most impact in reducing the error. This should be beneficial within the context of block Gauss-Seidel as well. Also, for single-scale models with long-range correlations, using multi-scale solvers such as [126] can dramatically improve convergence. Alternatively, when the MRF model itself has multiple scales (as in Section 5.2.3), then estimation approaches in [31,32] can be used. There the model is decomposed into a tractable tree-structured component, and disjoint horizontal components (one for each scale), which, conditioned on the coarser scale variables, have short conditional correlations and are also tractable. By iterating between these two tractable

[17] A subgraph of G is induced if it has a subset of vertices of G and *all* the edges of G that connect them. A spanning tree is not induced: it has all the vertices of G but only a subset of the edges.

[18] We note that in general the convergence of ET iterations is not guaranteed. By also requiring the subtrees to be induced, we force ET to be equivalent to Gauss-Seidel, guaranteeing its convergence.

subproblems, estimation in the whole multi-scale model can be done efficiently.

■ 5.5 Chapter Summary

We have presented a simple computationally efficient scheme to compute accurate variance approximations in large-scale GMRF models. The scheme involves designing a low-rank aliasing matrix which is used during matrix inversion. By a judicious choice of the aliasing matrix the errors in the approximation can be made unbiased and with small variances. We have designed aliasing matrices for both the short-range and smooth long-range correlation cases, and applied them to single and multi-scale GMRF models.

There are many interesting directions for further research: using wavelet packets to better adapt to the statistics of the GMRF; using diffusion wavelets [34] to extend the wavelet-based construction of B to arbitrary (non-regular) graphs; finding an interpretation of this approach in the walk-sum framework presented in Chapters 3 and 4. In addition, for multi-scale GMRF models we are interested to find ways to design a low-rank aliasing matrix that exploits the short correlation length of the conditional model within each scale, rather than using wavelet-based constructions.

Chapter 6

Conclusion

This thesis makes contributions in the area of approximate inference in Gaussian graphical models. In our first contribution we presented a walk-sum framework for Gaussian inference, and applied it to analyze Gaussian loopy belief propagation, establishing new results on its convergence. We also considered more general vector and factor graph versions of LBP, and extended the walk-sum framework to give insight into their behavior. In our second contribution we described an efficient approach to compute approximate variances in large-scale Gauss-Markov random fields and analyzed its accuracy. We started by considering models with short-range correlations, and then used wavelet analysis to extend the approach to handle models with smooth long-range correlations. Next we highlight the main contributions in each chapter, and then in Section 6.2 we discuss interesting questions raised in the thesis, and suggest directions for further research.

■ 6.1 Contributions

We now outline the main contributions of each chapter.

Chapter 3: Walk-sum analysis of Gaussian BP

• *Walk-summable models.* We have shown that in a large class of Gaussian graphical models, which we call walk-summable, inference can be interpreted in the language of walks and sums of weights over walks: means, variances and correlations correspond to walk-sums over certain sets of walks. We established that attractive models, tree-structured models, non-frustrated models, diagonally-dominant and pairwise-normalizable models are all walk-summable. We also showed that the class of pairwise-normalizable models is in fact equivalent to walk-summable models.

• *Walk-sum interpretation of LBP.* We have shown that BP message updates in a tree-structured model can be viewed as calculating walk-sums in subtrees of the model. Using the LBP computation tree we also presented a walk-sum interpretation of LBP.

• *Convergence of LBP.* We have shown that LBP converges in walk-summable mod-

els, and captures all the walks for the means, but only a subset of walks for the variances, the so-called back-tracking walks.

• *LBP in non-walk-summable models.* We also established an almost[1] necessary and sufficient condition for convergence of LBP variances based on the validity of the computation tree. While convergence of the variances in this setting does not guarantee convergence of the means, we observed empirically that means can always be forced to converge with sufficient damping.

Chapter 4: Extensions of walk-sum analysis

• *Combinatorial ideas for walk-sums.* We applied simple ideas from combinatorics to compute exact root-node self-return walk-sums in the computation trees for regular graphs, giving insight into the behavior of LBP variances.

• *Walk-sums for models with vector variables.* We extended the walk-sum framework for models with vector variables, and related it to convergence of vector-LBP. Along the way we formulated a multitude of conjectures.

• *Walk-sums for factor graph models.* We also extended the walk-sum framework for Gaussian models defined on factor graphs, and applied it to develop a sufficient condition for convergence of factor graph LBP.

• *Factor graph normalizable models.* As an alternative to factor graph walk-summability, we also considered the notion of factor graph normalizability, which guarantees convergence of FG-LBP variances. We also related FG-LBP with the recently proposed complex-valued version of LBP.

Chapter 5: Low-rank Variance approximation

• *Low-rank approximation.* We presented an approach that uses low-rank aliasing matrices to enable efficient computation of approximate variances in large-scale GMRF models by reducing the problem to a sequence of solutions of sparse linear problems.

• *Aliasing matrix for models with short-range correlation.* We described how to construct a low-rank aliasing matrix for models with short-range correlations, and established accuracy guarantees: the resulting variance approximations are unbiased, and their error variances can be made small.

• *Wavelet-based aliasing matrix for models with smooth long-range correlations.* By a non-trivial use of wavelet bases we were able to construct an aliasing matrix that is

[1] Our result here did not address the exact behavior of the variances for the special case with $\varrho_\infty = 1$.

geared towards models with smooth long-range correlations. We also derived accuracy guarantees for this wavelet-based approach – again the resulting variance approximations are unbiased, and the variance of the errors can be made small.

■ 6.2 Recommendations

In this section we highlight some of the questions raised in the thesis, and suggest directions for further research. The discussion ranges from concrete problems that were encountered in the thesis, to more general open-ended topics.

■ 6.2.1 Open Questions Concerning Walk-sums

LBP Means in Non-Walk-summable Models. We discussed non-walk-summable models in Section 3.3 and established that while variances converge when $\varrho_\infty < 1$, the means may or may not converge. A related phenomenon was observed in Section 4.3.3 concerning factor graph normalizable models: in such models FG-LBP variances are guaranteed to converge, but in some cases (which are rather difficult to find) means may still fail to converge. Empirically, in both of these scenarios we have seen that one can always force the means to converge by sufficient damping of the message updates. We mention some ideas that may be useful in formally proving this observation.

Recall that once variances converge, the updates for the means follow a linear system. This happens both for scalar LBP (2.29 - 2.30) and for factor graph LBP (4.28). Consider the scalar case, and define $\Delta h = (\Delta h_{i\to j} \mid \{i,j\} \in \mathcal{E})$. Then with all quantities $\hat{J}_{i\backslash j}$ fixed, the vector Δh follows a linear system:

$$\Delta h^{(n+1)} = L\ \Delta h^{(n)} + b \tag{6.1}$$

for some matrix L and some vector b. This system converges if its spectral radius satisfies $\varrho(L) < 1$. Characterizing $\varrho(L)$ is quite difficult because the matrix L depends on the converged values of the LBP variances.

One can attempt damping the message updates by using the following update:

$$\Delta h^{(n+1)} = (1-\alpha)\Delta h^{(n)} + \alpha(L\ \Delta h^{(n)} + b) \ \text{ with } \ 0 < \alpha \leq 1 \tag{6.2}$$

This system converges if $\varrho\left((1-\alpha)I + \alpha L\right) < 1$. Consider Figure 6.1. We display the eigenvalues of the undamped matrix L, and of the damped matrix $L_\alpha \triangleq (1-\alpha)I + \alpha L$ as α ranges from 0 to 1. Since L is not symmetric, the eigenvalues are in general complex-valued. If some eigenvalues of L are outside the unit circle, but there exists $0 < \alpha < 1$ such that the eigenvalues of L_α are inside the unit circle then the damped linear system will converge. From the figure it is clear that if $Re(\lambda_i) < 1$ for all the eigenvalues, then there exists a sufficiently small α such that all the eigenvalues of L_α enter the unit circle. This condition is equivalent to *Lyapunov stability* condition for a continuous dynamical system $\dot{x}(t) = (L-I)x(t)$, namely $\mathrm{Re}(\lambda_i(L-I)) < 0$ (which

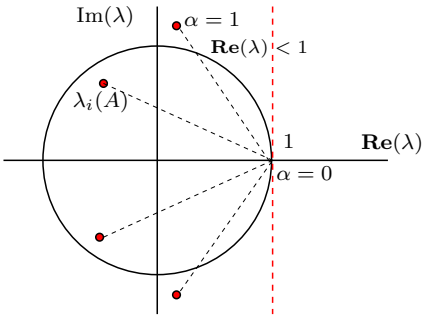


Figure 6.1. The effect of damping on convergence of the means: if $\mathrm{Re}(\lambda_i) < 1$, then means will converge with sufficient damping.

reduces to $\mathrm{Re}(\lambda_i(L)) < 1$) [39]. An equivalent condition is the existence of a positive-definite matrix Q such that

$$(I - L)^T Q + Q(I - L) \succ 0. \tag{6.3}$$

If one can design $Q \succ 0$ which depends on L (and hence on the converged variances), and show that (6.3) is satisfied whenever $\varrho_\infty < 1$ in the scalar case, or for FG-normalizable models in the factor graph case, then the question of mean convergence with damping would be resolved.

A more challenging question is whether it is possible to modify LBP to obtain means and variances when $\varrho_\infty > 1$. As we have mentioned in Chapter 3, damping of the variance updates does not help. However, one may consider methods in the field of divergent series [60], such as Aitken sequence transformation, to attempt to to accelerate convergence when LBP has slow convergence, and perhaps to extract the answers from divergent series.

Yet another question is to develop a walk-sum interpretation for LBP with damping – for both means and variances. The computation tree interpretation of LBP does not easily accommodate damping.

Characterizing Vector Walk-summability Recall that in Section 4.2 we introduced several sufficient conditions for vector walk-summability and generalized walk-summability, and established that these sufficient conditions also imply vector pairwise-normalizability. Reconciling all these notions and showing whether these sufficient conditions are also necessary, or finding counterexamples, will provide a much better understanding of vector walk-summability. We have listed a number of conjectures in Section 4.2.6. To add to the difficulty, numerical studies of these conjectures require either extensive sampling or challenging optimization over sets of block-orthogonal and block-invertible matrices.

6.2.2 Extending the Walk-sum Framework

Walk-sum interpretation of GBP In Chapter 4 we have considered pairwise MRF LBP with vector variables, and factor graph LBP, as examples of approaches which can get increasingly more accurate approximations at the cost of more computation. They are both special cases of the more general class of LBP-like message-passing schemes that are referred to as *generalized belief propagation* (GBP). GBP methods include as special cases the cluster-variational method, the junction graph method, and the more general region graph method [139]. The idea of the region graph method is to create a hierarchy of regions (each region containing some subset of factors and variables), subregions, and their subregions, connect regions to their subregions by directed links, and associate counting numbers with each region, such that the total number of appearances of each variable and each factor, including the counting numbers, is equal to 1. Then GBP is specified as a set of message-passing updates between regions and their subregions, iterated until convergence. These message-passing updates can be motivated by minimizing a certain region graph free energy (a generalization of the Bethe free energy) subject to consistency constraints on the beliefs of overlapping regions [139].

All the versions of LBP considered in Chapters 3 and 4 are special case of the more flexible and more powerful GBP formalism. They can be viewed as GBP with two levels of hierarchy, with large regions roughly corresponding to edges (or hyper-edges) and small regions to variables. It is of interest to develop a walk-sum interpretation of the more general versions of GBP with more than two levels of hierarchy, e.g. the cluster variational method, or the more general region graph method. The walk-sum perspective may provide insight into convergence and accuracy of these methods, and suggest solutions to the open question of how to design region graphs for better GBP performance.

Improving LBP – adding memory, alternate expansions In this thesis we have used walk-sums to analyze LBP and its variants. However, the walk-sum interpretation may also suggest ways to improve upon LBP – to either speed up convergence by gathering more of the walks faster, to find more accurate variances by taking into account some of the walks that LBP misses, or even to suggest entirely different inference approaches.

For example, we have seen in Chapter 3 that LBP variance estimates do not include walk-sums over non-backtracking walks. By explicitly adding walk-sums over the shortest non-backtracking walks we can get more accurate LBP variances. This suggests investigating LBP with more *memory*, where the nodes are not simply combining messages from the neighbors and passing them on, but are also able to take advantage of the recent history of message updates, or perhaps the knowledge of local graph structure to make more accurate assessments of the marginals, or to improve the speed of convergence. Vector, factor-graph and generalized LBP can be thought of as versions of LBP with more memory, but there may also be radically different approaches, for example ones based on *cycle bases* [16, 123], which expose the structure of the cycles of

the graph, and perhaps use walks on cycles as the basic building blocks to generate all walks.

We also note that the walk-sum framework has already been applied to analyze other inference algorithms for Gaussian graphical models – in particular the embedded trees (ET) algorithm, and to suggest their improvements [25, 26].

Walk-sum framework for log-determinants In the thesis we have described a walk-sum representation for means, variances and covariances. There also exists a walk-sum expansion for the the log-partition of a Gaussian graphical model:

$$\log Z = \log \int \exp(-\frac{1}{2}x^T Jx + h^T x)dx \tag{6.4}$$

The log-partition is given by:

$$\log Z = -\frac{1}{2}(|V|\log(2\pi) - \log\det(J) + h^T J^{-1}h). \tag{6.5}$$

The main challenge in evaluating the log-partition for large-scale models is the log-determinant of J; all the remaining terms are easily computable. Suppose $J = I - R$. Then the following expansion can be used to evaluate $\log\det(J)$:[2]

$$\log\det(J) = -\text{tr}\left(\sum_{l=1}^{\infty} R^l/l\right) \tag{6.6}$$

We see the familiar terms R^l inside the summation, but they are now weighted by $\frac{1}{l}$. What this means is that we have a walk-sum interpretation with a different definition of weight:

$$\phi_d(w) = \frac{1}{l(w)}\prod_{k=1}^{l(w)} r_{w_{k-1},w_k}$$

The weight of each walk is scaled by the length of the walk, $l(w)$. With these definition,

$$\log\det(J) = \sum_{i\in V}\sum_{w:i\to i}\phi_d(w) \tag{6.7}$$

The addition of scaling by the length of the walk seems like an innocuous enough change, but unfortunately it significantly complicates the walk-sum algebra. Take walks $u = (u_0, \ldots, u_n)$ and $v = (v_0, \ldots, v_m)$ with $u_n = v_0$ (walk v begins where walk u ends) and consider their concatenation, i.e. the walk $uv = (u_0, \ldots, u_n, v_1, \ldots, v_m)$. Its length is $l(uv) = l(u) + l(v) = n + m$. Now

$$\phi_d(uv) = \frac{n\phi_d(u) + m\phi_d(v)}{n+m}, \tag{6.8}$$

[2] Consider $\log\det(I-R) = -\text{tr}\left(\sum_{l=1}^{\infty} R^l/l\right)$. Differentiating both sides with respect to R, we obtain $-(I-R)^{-1} = -\sum_{l=1}^{\infty} lR^{l-1}/l = -\sum_{l=0}^{\infty} R^l$, which recovers the walk-sum expansion of $J = I - R$.

which is a more complicated operation compared to the one for ordinary walk-sums: there we simply have $\phi(uv) = \phi(u)\phi(v)$. Many of the walk-sum manipulations done in Chapter 3 become much more difficult, for instance the operation of going from single-revisit to multiple-revisit walk-sums, which for regular walks is a simple scalar mapping, is no longer tractable for the length-scaled walk-sums.

Thus, an interesting but nontrivial problem is developing recursive walk-sum calculations for log-determinants in tree structured models, and in computation trees of loopy graphs, and perhaps relating the latter to the Bethe approximation of the free energy.

Walk-summability of random graphs Random graphs arise in many important applications such as sensor networks, where the nodes may be deployed randomly over a field, or in the analysis of the Internet, or social networks. In these applications, in particular in sensor networks, one may be interested in applying distributed inference algorithms, such as LBP. Assuming a model for the random graph, e.g. Erdos-Renyi graphs or unit-disk graphs [17], and a model for the edge-weights, an interesting question is whether the resulting graphical model is walk-summable (with high probability, or in the large system limit), which would guarantee convergence of LBP. Another interesting question is whether a random or a regular structure of the graph may allow to approximately correct LBP variances to give closer approximations to the true variances. For example, in grid models it is often the case that LBP variances are roughly similar to the true variances scaled by some unknown constant. This suggests a simple way to improve LBP variances by finding this constant and rescaling them.

■ 6.2.3 Relation with Path-sums in Discrete Models

Our Gaussian walk-sum framework is built upon the power-series expansion of the matrix inverse, which is the basis for inference in Gaussian models. It does not directly apply to graphical models with discrete state spaces, in which inference is not described by matrix inversion. However, in the literature there have been a variety of graph-based ideas involving sums over paths, self-avoiding walks, and loops, to analyze inference in discrete models, which we now briefly mention.

For binary Ising models the paper [49] proposes an expansion of correlations in terms of so-called polygons, which is then approximated by sums of weights of self-avoiding walks in the graph. The weight of each walk is the product of certain scalar edge-weights. Another use of self-avoiding walks appears in [67, 79], where exact MAP estimates for binary models can be found by building a self-avoiding walk tree, and doing exact estimation in the tree. A recent work [30] proposes improving LBP in discrete models by so-called loop-series corrections[3]. It is interesting to compare these ideas with our walk-sum framework, perhaps find parallels or explore possibilities for cross-fertilization. Also, these graph-based ideas may inspire new inference algorithms

[3]These are very different from our simple idea of adding the weights of missing non-backtracking loops to LBP variances described in Section 6.2.2.

in discrete models, and may also be used to give new insight into LBP or max-product algorithms, e.g. [122].

We mention another possible connection to discrete models. The papers [69, 99] developed convergence conditions for discrete LBP, which depend on the spectral radius of a certain weighted sparse message-to-message adjacency matrix, which has resemblance to our scalar walk-summability condition. This resemblance may not be purely superficial, and may give a deeper insight into discrete LBP [68].

■ 6.2.4 Extensions for Low-rank Variance Approximation

There are many important extensions to the low-rank variance approximation framework that wait to be developed. For example, we have considered only two classes of GMRFs: ones with short-range correlations, and ones with smooth long-range correlations. In some remote sensing applications the GMRF models may not fall into either of these classes. For example, the data collected for seismic inversion involves convolutions of the unknown seismic field with the so-called seismic wavelet, an oscillatory acoustic test signal. This convolution induces rather long oscillatory correlations, which do not fall into either of our short-range or smooth long-range cases. Extending our approach so that it can take advantage of this and other sensing modalities is an important problem.

Some more broad problems include developing very fast and accurate coarse-resolution variance approximations: instead of finding an approximation of the individual variances at every node of the GMRF, one could attempt to find accurate approximations of a coarse summary of these variances (e.g. of low-pass filtered and subsampled variances). The hope is that such a rough evaluation of the variances could be obtained rapidly even for very large fields. Another important question is how to adapt our low-rank approximation approach to efficiently generate samples from large-scale non-homogeneous fields with irregular measurements. Standard MCMC approaches such as Gibbs sampling may be too slow for such scenarios.

Diffusion wavelets We mention a very interesting idea that comes from the diffusion wavelet literature [34]. Suppose that H is a diffusion operator (mapping from $R^{|V|} \to R^{|V|}$ on a graph). Consider the following expansion of the Green's function for this operator:

$$P_g = (I - H)^{-1} = \prod_{l=0}^{\infty} (I + H^{2^l}) \tag{6.9}$$

This multiplicative decomposition converges fast, so a very accurate approximation of P_g can be obtained by keeping only a few terms H^{2^l}. A key observation made in [34] is that for certain classes of H, as l increases, the terms H^{2^l} have progressively smaller effective rank (i.e. many eigenvalues fall below a small threshold and can be discarded), so these terms can be represented compactly. Using this insight the authors proposed a multi-resolution recursive decomposition of H^{2^l} into the so-called diffusion wavelet basis, that allows the computation of the matrix-vector product $P_g h$ in linear time.

We note that the expansion in (6.9) has intimate parallels to our walk-sum expansion of covariance, $P = J^{-1} = (I - R)^{-1} = \sum_{l=0}^{\infty} R^l$. Reorganizing the terms we have

$$P = (I - R)^{-1} = \sum_{l=0}^{\infty} R^l = \prod_{l=0}^{\infty} (I + R^{2^l}) \tag{6.10}$$

Now if we consider R to be the diffusion operator, then the covariance P is the corresponding Green's function. This suggests an alternative method to evaluate $\mu = Ph$ for discrete graphs through diffusion-wavelet like expansion, instead of solving sparse linear systems $J\mu = h$, as we have done in Chapter 5.

On another note, this also brings up the very general question of relating our work on finite graphs to fields defined on continuous manifolds – as many GMRFs are obtained by discretization of the corresponding stochastic processes on continuous manifolds. In that setting covariances are not matrices but rather continuous functions, and the role of information matrices is taken by differential operators.

Appendix A

Proofs and details

A.1 Proofs for Chapter 3

Proof of Proposition 3.1.1 *Proof of (i) ⇒ (ii).* We examine convergence of the matrix series in (ii) element-wise. First note that $(\bar{R}^l)_{ij}$ is an absolute walk-sum over all walks of length l from i to j:

$$(\bar{R}^l)_{ij} = \sum_{w:i\xrightarrow{l}j} |\phi(w)|$$

(there are a finite number of these walks so the sum is well-defined). Now, if (i) holds then using properties of absolute convergence we can order the sum $\sum_{w:i\to j} |\phi(w)|$ however we wish and it still converges. If we order walks by their length and then group terms for walks of equal lengths (each group has a finite number of terms) we obtain:

$$\sum_{w:i\to j} |\phi(w)| = \sum_{l} \sum_{w:i\xrightarrow{l}j} |\phi(w)| = \sum_{l} (\bar{R}^l)_{ij} \tag{A.1}$$

Therefore, the series $\sum_l (\bar{R}^l)_{ij}$ converges for all i, j.

Proof of (ii) ⇒ (i). To show convergence of the sum $\sum_{w:i\to j} |\phi(w)|$ it is sufficient to test convergence for any convenient ordering of the walks. As shown in (A.1), $\sum_l (\bar{R}^l)_{ij}$ corresponds to one particular ordering of the walks which converges by (ii). Therefore, the walk-sums in (i) converge absolutely.

Proof of (ii) ⇔ (iii). This is a standard result in matrix analysis [127].

Proof of (iii) ⇔ (iv). Note that λ is an eigenvalue of $\bar{R}$ if and only if $1-\lambda$ is an eigenvalue of $I-\bar{R}$ ($\bar{R}x = \lambda x \Leftrightarrow (I-\bar{R})x = (1-\lambda)x$). Therefore, $\lambda_{\min}(I-\bar{R}) = 1-\lambda_{\max}(\bar{R})$. According to the Perron-Frobenius theorem, $\varrho(\bar{R}) = \lambda_{\max}(\bar{R})$ because $\bar{R}$ is non-negative. Thus, $\varrho(\bar{R}) = 1 - \lambda_{\min}(I-\bar{R})$ and we have that $\varrho(\bar{R}) < 1 \Leftrightarrow \lambda_{\min}(I-\bar{R}) > 0$. □

Proof of Corollary 3.1.2 We will show that for any non-frustrated model there exists a diagonal D with $D_{ii} = \pm 1$, i.e. a signature matrix, such that $DRD = \bar{R}$. Hence, R and $\bar{R}$ have the same eigenvalues, because $DRD = DRD^{-1}$ is a similarity transform which preserves the eigenvalues of a matrix. It follows that $I - R \succ 0$ implies $I - \bar{R} \succ 0$ and walk-summability of J by Proposition 3.1.1(iv).

Now we describe how to construct a signature similarity which makes R attractive for non-frustrated models. We show how to split the vertices into two sets V^+ and V^- such that negating V^- makes the model attractive. Find a spanning tree T of the graph G. Pick a node i. Assign it to V^+. For any other node j, there is a unique path to i in T. If the product of edge-weights along the path is positive, then assign j to V^+, otherwise to V^-. Now, since the model is non-frustrated, all edges $\{j,k\}$ in G such that $j,k \in V^+$ are positive, all edges with $j,k \in V^-$ are positive, and all edges with $j \in V^+$ and $k \in V^-$ are negative. This can be seen by constructing the cycle that goes from j to i to k in T and crosses the edge $\{k,j\}$ to close itself. If $j,k \in V^+$ then the paths j to i and i to k have a positive weight, hence in order for the cycle to have a positive weight, the last step $\{k,j\}$ must also have a positive weight. The other two cases are similar. Now let D be diagonal with $D_{ii} = 1$ for $i \in V^+$, and $D_{ii} = -1$ for $i \in V^-$. Then $DRD = \begin{bmatrix} R_{V^+} & -R_{V^+,V^-} \\ -R_{V^-,V^+} & R_{V^-} \end{bmatrix} \geq 0$, i.e. $DRD = \bar{R}$. □

Proof of Proposition 3.1.2 *Proof of WS ⇒ (i).* WS is equivalent to $\varrho(\bar{R}) < 1$ by Proposition 3.1.1. But $\varrho(R) \leq \varrho(\bar{R})$ by (3.1). Hence, $\varrho(\bar{R}) < 1 \Rightarrow \varrho(R) < 1$.

Proof of (i) ⇒ (ii). Given $J = I - R$, it holds that $\lambda_{\min}(J) = 1 - \lambda_{\max}(R)$. Also, $\lambda_{\max}(R) \leq \varrho(R)$. Hence, $\lambda_{\min}(J) = 1 - \lambda_{\max}(R) \geq 1 - \varrho(R) > 0$ for $\varrho(R) < 1$.

Proof of (i) ⇒ (iii). This is a standard result in matrix analysis. □

Proof of Proposition 3.1.5 Assume that G is connected (otherwise we apply the proof to each connected component, and the spectral radii are the maxima over the respective connected components). We prove that $\varrho(\bar{R}) = \varrho(\hat{R})$. By the Perron-Frobenius theorem, there exists a positive vector x such that $\bar{R}x = \varrho(\bar{R})x$. Let $\hat{x} = (x; x)$. Then $\hat{R}\hat{x} = \varrho(\bar{R})\hat{x}$ because

$$(\hat{R}\hat{x})_{\pm} = (R_+ + R_-)x = \bar{R}x = \varrho(\bar{R})x$$

Hence, $\varrho(\bar{R})$ is an eigenvalue of $\hat{R}$ with positive eigenvector $\hat{x}$. First suppose that $\hat{G}$ is connected. Then, by the Perron-Frobenius theorem, $\varrho(\bar{R}) = \varrho(\hat{R})$ because $\hat{R}$ has a unique positive eigenvector which has eigenvalue equal to $\varrho(\hat{R})$. Now, $\hat{J} = I - \hat{R} \succ 0 \Leftrightarrow \hat{J}$ is WS $\Leftrightarrow \varrho(\hat{R}) < 1 \Leftrightarrow \varrho(\bar{R}) < 1 \Leftrightarrow J = I - R$ is WS. If $\hat{G}$ is disconnected then $\hat{R}$ is a block-diagonal matrix with two copies of $\bar{R}$ (after relabeling the nodes), so $\varrho(\hat{R}) = \varrho(\bar{R})$. □

Proof of Proposition 3.1.6 We partition walk-sums into sums over "even" and "odd" walks according to the number of negative edges crossed by the walk. Thus a walk w is even if $\phi(w) > 0$ and is odd if $\phi(w) < 0$. The graph $\hat{G}$ is defined so that every walk

from i_+ to j_+ is even and every walk from i_+ to j_- is odd. Thus,

$$\begin{aligned} P_{ij} &= \sum_{\text{even } w:i\to j} \phi(w) + \sum_{\text{odd } w:i\to j} \phi(w) \\ &= \sum_{w:i_+\to j_+} \hat{\phi}(w) - \sum_{w:i_+\to j_-} \hat{\phi}(w) \\ &= \hat{P}_{i_+,j_+} - \hat{P}_{i_+,j_-} \end{aligned}$$

The second part of the the proposition follows by similar logic. Now we classify a walk as even if $h_{w_0}\phi(w) > 0$ and as odd if $h_{w_0}\phi(w) < 0$. Note also that setting $\hat{h} = (h_+; h_-)$ has the effect that all walks with $h_{w_0} > 0$ begin in V_+ and all walks with $h_{w_0} < 0$ begin in V_-. Consequently, every even walk ends in V_+ and every odd walk ends in V_-. Thus,

$$\begin{aligned} \mu_i &= \sum_{\text{even } w:*\to i} h_*\phi(w) + \sum_{\text{odd } w:*\to i} h_*\phi(w) \\ &= \sum_{w:*\to i_+} \hat{h}_*\hat{\phi}(w) - \sum_{w:*\to i_-} \hat{h}_*\hat{\phi}(w) \\ &= \hat{\mu}_{i_+} - \hat{\mu}_{i_-} \quad \square \end{aligned}$$

Proof of Proposition 3.1.7 Take J_1 and J_2 pairwise-normalizable. Take any $\alpha, \beta \geq 0$ such that at least one of them is positive. Then $\alpha J_1 + \beta J_2$ is also pairwise-normalizable simply by taking the same weighted combinations of each of the J_e matrices for J_1 and J_2. Setting $\beta = 0$ shows that $\mathcal{J}_{PN}$ is a cone, and setting $\beta = 1 - \alpha$ shows convexity. The cone is pointed since it is a subset of the cone of semidefinite matrices, which is pointed [7]. $\square$

Proof of Proposition 3.1.8 *Proof of PN $\Rightarrow$ WS.* It is evident that any J matrix which is pairwise-normalizable is positive definite. Furthermore, reversing the sign of the partial correlation coefficient on edge e simply negates the off-diagonal element of J_e which does not change the value of $\det J_e$ so that we still have $J_e \succeq 0$. Thus, we can make all the negative coefficients positive and the resulting model $I - \bar{R}$ is still pairwise-normalizable and hence positive-definite. Then, by Proposition 3.1.1(iv), $J = I - R$ is walk-summable.

Proof of WS $\Rightarrow$ PN. Given a walk-summable model $J = I - R$ we construct a pairwise-normalized representation of the information matrix. We may assume the graph is connected (otherwise, we may apply the following construction for each connected component of the graph). Hence, by the Perron-Frobenius theorem there exists a positive eigenvector $x > 0$ of $\bar{R}$ such that $\bar{R}x = \lambda x$ and $\lambda = \varrho(\bar{R}) > 0$. Given (x, λ) we construct a representation $J = \sum_e [J_e]$ where for $e = \{i, j\}$ we set:

$$J_e = \begin{pmatrix} \frac{|r_{ij}|x_j}{\lambda x_i} & -r_{ij} \\ -r_{ij} & \frac{|r_{ij}|x_i}{\lambda x_j} \end{pmatrix}$$

This is well-defined (there is no division by zero) since x and λ are positive. First, we verify that $J = \sum_{e\in\mathcal{E}} [J_e]$. It is evident that the off-diagonal elements of the edge matrices sum to $-R$. We check that the diagonal elements sum to one:

$$\sum_e [J_e]_{ii} = \frac{1}{\lambda x_i} \sum_j |r_{ij}| x_j = \frac{(\bar{R}x)_i}{\lambda x_i} = \frac{(\lambda x)_i}{\lambda x_i} = 1$$

Next, we verify that each J_e is positive-definite. This matrix has positive diagonal and determinant:

$$\det J_e = \left(\frac{|r_{ij}|x_j}{\lambda x_i}\right)\left(\frac{|r_{ij}|x_i}{\lambda x_j}\right) - (-r_{ij})^2 = r_{ij}^2\left(\frac{1}{\lambda^2} - 1\right) > 0$$

The inequality follows from walk-summability because $0 < \lambda < 1$ and hence $\left(\frac{1}{\lambda^2} - 1\right) > 0$. Thus, $J_e \succ 0$. □

Proof of Proposition 3.1.9 Let $a_i = J_{ii} - \sum_{j\neq i} |J_{ij}|$. Note that $a_i > 0$ follows from diagonal-dominance. Let $\deg(i)$ denote the degree of node i in G. Then, $J = \sum_{e\in\mathcal{E}} [J_e]$ where for edge $e = \{i, j\}$ we set

$$J_e = \begin{pmatrix} |J_{ij}| + \frac{a_i}{\deg(i)} & J_{ij} \\ J_{ij} & |J_{ij}| + \frac{a_j}{\deg(j)} \end{pmatrix}$$

with all other elements of $[J_e]$ set to zero. Note that:

$$\sum_e [J_e]_{ii} = \sum_{j\in\mathcal{N}(i)} \left(|J_{ij}| + \frac{a_i}{\deg(i)}\right) = a_i + \sum_{j\in\mathcal{N}(i)} |J_{ij}| = J_{ii}$$

Also, J_e has positive diagonal elements and has determinant $\det(J_e) > 0$. Hence, $J_e \succ 0$. Thus, J is pairwise-normalizable. □

Proof of Proposition 3.2.2 To calculate the walk-sum for multiple-revisit self-return walks in $T_{i\backslash j}$, we can use the single-revisit counterpart:

$$\gamma_{i\backslash j} = \phi(i \to i \mid T_{i\backslash j}) = \frac{1}{1 - \phi\left(i \overset{\backslash i}{\to} i \mid T_{i\backslash j}\right)} \tag{A.2}$$

Now, we decompose the single-revisit walks in the subtree $T_{i\backslash j}$ in terms of the possible first step of the walk (i, k), where $k \in \mathcal{N}(i)\backslash j$. Hence,

$$\phi(i \overset{\backslash i}{\to} i \mid T_{i\backslash j}) = \sum_{k\in\mathcal{N}(i)\backslash j} \phi(i \overset{\backslash i}{\to} i \mid T_{k\to i}) \tag{A.3}$$

Using (3.4), (A.2), and (A.3), we are able to represent the walk-sum $\phi(j \overset{\setminus j}{\to} j \mid T_{i\to j})$ in $T_{i\to j}$ in terms of the walk-sums $\phi(i \overset{\setminus i}{\to} i \mid T_{k\to i})$ on smaller subtrees $T_{k\to i}$. This is the basis of the recursive calculation:

$$\alpha_{i\to j} = r_{ij}^2 \frac{1}{1 - \sum_{k\in\mathcal{N}(i)\setminus j} \alpha_{k\to i}}$$

These equations look strikingly similar to the belief propagation updates. Combining (2.29) and (2.30) from Section 2.3.1 we have:

$$-\Delta J_{i\to j} = J_{ij}^2 \frac{1}{J_{ii} + \sum_{k\in\mathcal{N}(i)\setminus j} \Delta J_{k\to i}}$$

It is evident that the recursive walk-sum equations can be mapped exactly to belief propagation updates. In normalized models $J_{ii} = 1$. We have the message update $\alpha_{i\to j} = -\Delta J_{i\to j}$, and the variance estimate in the subtree $T_{i\setminus j}$ is $\gamma_{i\setminus j} = \hat{J}_{i\setminus j}^{-1}$. □

Proof of Proposition 3.2.3 A multiple-revisit walk in $T_{i\setminus j}$ can be written in terms of single-visit walks:

$$\phi_h(* \to i \mid T_{i\setminus j}) = \left(h_i + \phi_h(* \overset{\setminus i}{\to} i \mid T_{i\setminus j})\right) \phi(i \to i \mid T_{i\setminus j})$$

We already have $\gamma_{i\setminus j} = \phi(i \to i \mid T_{i\setminus j})$ from (A.2). The remaining term $\phi_h(* \overset{\setminus i}{\to} i \mid T_{i\setminus j})$ can be decomposed according to the subtrees in which the walk lives:

$$\phi_h(* \overset{\setminus i}{\to} i \mid T_{i\setminus j}) = \sum_{k\in\mathcal{N}(i)\setminus j} \phi_h(* \overset{\setminus i}{\to} i \mid T_{k\to i})$$

Thus we have the recursion:

$$\beta_{i\to j} = r_{ij}\gamma_{i\setminus j}(h_i + \sum_{k\in\mathcal{N}(i)\setminus j} \beta_{k\to i})$$

To compare this to the Gaussian BP updates, let us combine (2.29) and (2.30) in Section 2.3.1:

$$\Delta h_{i\to j} = -J_{ij}\hat{J}_{i\setminus j}^{-1}\left(h_i + \sum_{k\in\mathcal{N}(i)\setminus j} \Delta h_{k\to i}\right)$$

Thus BP updates for the means can also be mapped exactly into recursive walk-sum updates via $\beta_{i\to j} = \Delta h_{i\to j}$. □

Proof of Lemma 3.2.1 First, we note that for every walk w which ends at the root node of $T_i^{(n)}$ there is a corresponding walk in G which ends at i. The reason is that the neighbors of a given node j in $T_i^{(n)}$ correspond to a subset of the neighbors of j in G. Hence, for each step (w_k, w_{k+1}) of the walk in $T_i^{(n)}$ there is a corresponding step in G.

Next, we show that every walk $w = (w_0, \ldots, w_l)$ in G is contained in $T_{w_l}^{(n)}$ for some n. First consider the parallel message schedule, for which the computation tree $T_{w_l}^{(n)}$ grows uniformly. Then for any walk in G that ends at w_l and has length n there is a walk in $T_{w_l}^{(n)}$ that ends at the root.

The intuition for other message schedules is that every step (i, j) of the walk will appear eventually in any proper message schedule $\mathcal{M}$. A formal proof is somewhat technical. First we unwrap the walk w into a tree T_w rooted at w_l in the following way: start at w_l, the end of the walk, and traverse the walk in reverse. First add the edge $\{w_l, w_{l-1}\}$ to T_w. Now, suppose we are at node w_k in T_w and the next step in w is $\{w_k, w_{k-1}\}$. If w_{k-1} is already a neighbor of w_k in T_w then set the current node in T_w to w_{k-1}. Otherwise create a new node w_{k-1} and add the edge to T_w. It is clear that loops are never made in this procedure, so T_w is a tree.

We now show for any proper message schedule $\mathcal{M}$ that T_w is part of the computation tree $T_{w_l}^{(n)}$ for some n. Pick a leaf-edge $\{i_1, j_1\}$ of T_w. Since $\{\mathcal{M}^{(n)}\}$ is proper, there exist n_1 such that $(i_1, j_1) \in \mathcal{M}^{(n_1)}$. Now $(i_1, j_1) \in T_{i_1 \to j_1}^{(n_1)}$, and the edge appears at the root of $T_{i_1 \to j_1}^{(n_1)}$. Also, $T_{i_1 \to j_1}^{(n_1)} \subset T_{i_1 \to j_1}^{(m)}$ for $m > n_1$, so this holds for all subsequent steps as well. Now remove $\{i_1, j_1\}$ from T_w and pick another leaf edge $\{i_2, j_2\}$. Again, since $\{\mathcal{M}^{(n)}\}$ is proper, there exist $n_2 > n_1$ such that $(i_2, j_2) \in \mathcal{M}^{(n_2)}$. Remove $\{i_2, j_2\}$ from T_w, and continue similarly. At each such point n_k of eliminating some new edge $\{i_k, j_k\}$ of T_w, the whole eliminated subtree of T_w extending from $\{i_k, j_k\}$ has to belong to $T_{i_k \to j_k}^{(n_k)}$. Continue until just the root of T_w remains at step n. Now the computation tree $T_{w_l}^{(n)}$ (which is created by splicing together $T_{i \to j}^{(n)}$ for all edges (i, j) coming into the root of T_w) contains T_w, and hence it contains the walk w. □

Proof of Lemma 3.2.3 This result comes as an immediate corollary of Proposition A.1.1, which states that $\varrho(R_i^{(n)}) \leq \varrho(\bar{R})$ (here $R_i^{(n)}$ is the partial correlation matrix for $T_i^{(n)}$). For WS models, $\varrho(\bar{R}) < 1$ and the result follows. □

Proof of Lemma 3.3.1 The fact that the sequence $\{\varrho(R_i^{(n)})\}$ is bounded by $\varrho(\bar{R})$ is a nontrivial fact, proven in Appendix A.1.1 using a *k-fold graph* construction. To prove monotonicity, note first that for trees $\varrho(R_i^{(n)}) = \varrho(\bar{R}_i^{(n)})$. Also, note that all of the variables in the computation tree $T_i^{(n)}$ are also present in T_i^{n+1}. We zero-pad $\bar{R}_i^{(n)}$ to make it the same size as $\bar{R}_i^{(n+1)}$ (this does not change the spectral radius). Then it holds that $\bar{R}_i^{(n)} \leq \bar{R}_i^{(n+1)}$ element-wise. Using (3.1), it follows that $\varrho(\bar{R}_i^{(n)}) \leq \varrho(\bar{R}_i^{(n+1)})$, establishing monotonicity. □

Proof of Lemma 3.3.2 Let $T_i^{(n)}(\mathcal{M})$ denote the n-th computation tree under a proper message schedule $\mathcal{M}$ rooted at node i. We use the following simple extension of Lemma 3.2.1: Let $T_i^{(n)}(\mathcal{M}_1)$ be the nth computation tree rooted at i under message schedule $\mathcal{M}_1$. Take any node in $T_i^{(n)}(\mathcal{M}_1)$ which is a replica of node j in G. Then there exists m such that $T_i^{(n)}(\mathcal{M}_1) \subset T_j^{(m)}(\mathcal{M}_2)$, where $\mathcal{M}_2$ is another message schedule. The proof parallels that of Lemma 3.2.1: the tree $T_i^{(n)}(\mathcal{M}_1)$ has a finite number of edges, and we use induction adding one edge at a time.

Consider message schedule $\mathcal{M}_1$. By Lemma 3.3.1, $\varrho_i \triangleq \lim_{n\to\infty} \varrho(R_i^{(n)}(\mathcal{M}_1))$ exists. For any ϵ pick an L such that for $n \geq L$ it holds that $|\varrho(R_i^{(n)}(\mathcal{M}_1)) - \varrho_i| \leq \frac{\epsilon}{2}$. Pick a replica of node j inside $T_i^{(L)}(\mathcal{M}_1)$. Then using the property from the previous paragraph, there exists M such that $T_i^{(L)}(\mathcal{M}_1) \subset T_j^{(M)}(\mathcal{M}_2)$. Similarly there exists N such that $T_j^{(M)}(\mathcal{M}_2) \subset T_i^{(N)}(\mathcal{M}_1)$. It follows that $\bar{R}_i^{(L)}(\mathcal{M}_1) \leq \bar{R}_j^{(M)}(\mathcal{M}_2) \leq \bar{R}_i^{(N)}(\mathcal{M}_1)$, where we zero-pad the first two matrices to have the same size as the last one. Then, $\varrho(\bar{R}_i^{(L)}(\mathcal{M}_1)) \leq \varrho(\bar{R}_j^{(M)}(\mathcal{M}_2)) \leq \varrho(\bar{R}_i^{(N)}(\mathcal{M}_1))$. Then it holds that $\varrho_i - \frac{\epsilon}{2} \leq \varrho(\bar{R}_j^{(M)}(\mathcal{M}_2)) \leq \varrho_i + \frac{\epsilon}{2}$. Hence, $|\varrho(\bar{R}_j^{(M)}(\mathcal{M}_2)) - \varrho_i| \leq \epsilon$, and $\lim_{n\to\infty} \varrho(\bar{R}_j^{(n)}(\mathcal{M}_2)) = \varrho_i$. □

■ A.1.1 K-fold Graphs and Proof of Boundedness of $\varrho(R_i^{(n)})$.

Consider an arbitrary graph $G = (V, \mathcal{E})$. Suppose that we have a pairwise MRF defined on G with self potentials $\psi_i(x_i)$, for $v_i \in V$ and pairwise potentials $\psi_{ij}(x_i, x_j)$ for $(v_i, v_j) \in \mathcal{E}$. We construct a family of K-fold graphs based on G as follows:

1. Create K disconnected copies G_k, $k \in \{1, .., K\}$ of G, with nodes $v_i^{(k)}$, and edges $(v_i^{(k)}, v_j^{(k)})$. The nodes and the edges of G_k are labeled in the same way as the ones of G. The potentials ψ_i and ψ_{ij} are copied to the corresponding nodes and edges in all G_k.

2. Pick some pair of graphs G_k, G_l, and choose an edge (v_i, v_j) in G. We interconnect the corresponding edges in G_k and G_l: edges $(v_i^{(k)}, v_j^{(k)})$ and $(v_i^{(l)}, v_j^{(l)})$ become $(v_i^{(k)}, v_j^{(l)})$ and $(v_i^{(l)}, v_j^{(k)})$. The pairwise potentials are adjusted accordingly.

3. Repeat step 2 an arbitrary number of times for a different pair of graphs G_k, or a different edge in G.

An illustration of the procedure appears in Figure A.1. The original graph G is a 4-cycle with a chord. We create a 2-fold graph based on G by flipping the edges $(1, 2)$ in G_1 and $(1', 2')$ in G_2.

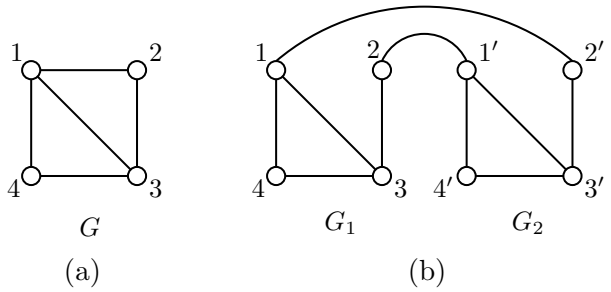


Figure A.1. Illustration of (a) graph G and (b) a 2-fold graph of G.

Now we apply the K-fold graph construction to Gaussian MRF models. Suppose that we have a model with information parameters J and h on G. Suppose that J is normalized to have unit-diagonal. Let G^K be a K-fold graph based on G with the information matrix J^K (which is also unit-diagonal by construction). Also, let $T_i^{(n)}$ be the nth computation tree for the original graph, and $J_i^{(n)}$ the corresponding information matrix (also unit-diagonal). Let $R = I - J$, $R^K = I^K - J^K$, and $R_i^{(n)} = I^{(n)} - J_i^{(n)}$ (here I, I^K, and $I^{(n)}$ are identity matrices of appropriate dimensions).

Lemma A.1.1 (Spectral radii of R and R^K). *For any K-fold graph G^K based on G: $\varrho(\bar{R}^K) = \varrho(\bar{R})$.*

Proof. Suppose that G is connected (otherwise apply the proof to each connected component of G, and the spectral radius for G will be the maximum of the spectral radii for the connected components).

Then, by the Perron-Frobenius theorem there exists a vector $x > 0$ such that $\bar{R}x = \varrho(\bar{R})x$. Create a K-fold vector x^K by copying entry x_i into each of the K corresponding entries of x^K. Then x^K is positive, and it also holds that $\bar{R}^K x^K = \varrho(\bar{R})x^K$ (since the local neighborhoods in G and G^K are the same). Now $\bar{R}^K$ is a non-negative matrix, and x^K is a positive eigenvector, hence it achieves the spectral radius of $\bar{R}^K$ by the Perron-Frobenius theorem. Thus, $\varrho(\bar{R}) = \varrho(\bar{R}^K)$. □

The construction of a K-fold graph based on G has parallels with the computation tree on G. The K-fold graph is locally equivalent to G and the computation tree, except for its leaf-nodes, is also locally equivalent to G. We show next that the computation tree $T_i^{(n)}$ is contained in some G^K for K large enough.

Lemma A.1.2 (K-fold graphs and computation trees). *Consider a computation tree $T_i^{(n)}$ corresponding to graph G. There exists a K-fold graph G^K, which contains $T_i^{(n)}$ as a subgraph, for K large enough.*

Proof. We provide a simple construction of a K-fold graph, making no attempt to minimize K. Let $T_i^{(n)} = (V_n, \mathcal{E}_n)$. Each node $v' \in V_n$ corresponds to some node

$v \in V$ in G. We create a K-fold graph G^K by making a copy $G_{v'}$ of G for every node $v' \in T_i^{(n)}$. Hence $K = |V_n|$. For each edge $(u', v') \in \mathcal{E}_n$ in the computation tree, we make an edge flip between nodes in graphs $G_{u'}$ and $G_{v'}$ that correspond to u and v in G. This operation is well-defined because edges in $T_i^{(n)}$ that map to the same edge in G do not meet. Thus, the procedure creates G^K which contains $T_i^{(n)}$ as a subgraph. □

Finally, we use the preceding lemmas to prove a bound on the spectral radii of the matrices $R_i^{(n)}$ for the computation tree $T_i^{(n)}$.

Proposition A.1.1 (Bound on $\varrho(R_i^{(n)})$). *For computation tree $T_i^{(n)}$: $\varrho(R_i^{(n)}) \leq \varrho(\bar{R})$.*

Proof. Consider a computation tree $T_i^{(n)}$. Recall that $\varrho(R_i^{(n)}) = \varrho(\bar{R}_i^{(n)})$, since $T_i^{(n)}$ is a tree. Use Lemma A.1.2 to construct a K-fold graph G^K which has $T_i^{(n)}$ as a subgraph. Zero-padding $\bar{R}_i^{(n)}$ to have the same size as $\bar{R}^K$, it holds that $\bar{R}_i^{(n)} \leq \bar{R}^K$. Since $\bar{R}_i^{(n)} \leq \bar{R}^K$, using (3.1) and Lemma A.1.1 we have: $\varrho(R_i^{(n)}) \leq \varrho(\bar{R}^K) = \varrho(\bar{R})$. □

■ A.2 Proofs and Details for Chapter 4

■ A.2.1 Scalar Walk-sums with Non-zero-diagonal

In Chapter 3 we have worked with the normalized matrix J, such that in the decomposition $J = I - R$, the matrix R is zero-diagonal. We now show that in the scalar case using other decompositions does not improve performance of LBP, and does not give better sufficient conditions for its convergence. This will also introduce walk-sums on graphs with self-loops which we use in Chapter 4.

Consider an arbitrary decomposition $J = D - K$, where we only require that D is diagonal with strictly positive elements (and, of course, $J \succ 0$). In particular, J may have unnormalized diagonal, i.e. diagonal elements not equal to 1, and K may have non-zero elements on its diagonal. One can use a more general version of Neumann power series:

$$J^{-1} = (D - K)^{-1} = D^{-T/2} \left(\sum_k (D^{-1/2} K D^{-T/2})^k \right) D^{-1/2}. \tag{A.4}$$

If we normalize J by $D^{-1/2}$, then we get $\tilde{J} = D^{-1/2} J D^{-T/2} = I - D^{-1/2} K D^{-T/2} \triangleq I - \tilde{R}$. Note that the expansion in (A.4) depends precisely on powers of $\tilde{R}$, hence it is just a rescaling of the power series for $\tilde{J}$. Thus, without loss of generality, we limit the discussion to the decomposition $J = I - R$. This applies to both the scalar and the vector case (with D block-diagonal).

We have another degree of freedom – whether or not to allow R to have non-zero diagonal entries. Starting from $J = I - R$ with zero-diagonal R, we obtain such alternative decompositions by $\tilde{J} = SJS$, and $\tilde{R} = I - \tilde{J} = I - SJS$, where S is some

positive diagonal scaling. Note that in this case J is not unit-diagonal, and $\tilde{R}$ *is not* the matrix of partial correlation coefficients.

Walk-summability with self-loops and LBP Suppose $J = I - R$, and R has non-zero diagonal entries. As we have done in Chapter 3, we assign weights r_{ij} to edges in the graph. To account for the non-zero diagonal entries of R, we introduce self-loops with weight r_{ii} at each node i, and allow walks to make a step (i, i). The necessary and sufficient condition for walk-summability of the non-zero diagonal decomposition is still $\varrho(\bar{R}) < 1$.

It is an easy exercise to extend the recursive walk-sum calculations from Section 3.2.1 to the non-zero diagonal case. We restrict the single-revisit self-return walks at node i not to take the self-loop step i-i. Multiple revisit self-return walks are allowed to follow the self-loop. With this definition, the only change in the equations for recursive variance and mean calculation (A.2) is the introduction of r_{ii} in the formula which computes multiple-revisit self-return walk-sums based on single-revisit ones:

$$\phi(i \to i \mid T_{i\setminus j}) = \frac{1}{1 - r_{ii} - \phi\left(i \overset{\setminus i}{\to} i \mid T_{i\setminus j}\right)} \tag{A.5}$$

All other equations in the proofs of Propositions 3.2.2, and 3.2.3 in Appendix A.1 are the same using the new definition of $\phi\left(i \to i \mid T_{i\setminus j}\right)$.

We use the following potential specification for LBP: $J_v = 1 - r_v$, and $J_e = \left[\begin{smallmatrix} 0 & -r_{ij} \\ -r_{ij} & 0 \end{smallmatrix}\right]$. With these definitions, all the results for zero-diagonal scalar case immediately extend to the non-zero diagonal case: if the model is walk-summable then LBP converges, and if the computation tree is valid, then variances converge. We next consider whether allowing R to have non-zero diagonal may improve the sufficient condition for LBP convergence.

Optimality of whitening for scalar-WS Suppose that the zero-diagonal model is $J = I - R$, and we rescale the variables: $\tilde{x} = D^{-1}x$, with D diagonal. Then $\tilde{J} = DJD$, and $\tilde{R} = I - D(I - R)D$ has a non-zero diagonal. The condition for walk-summability becomes $\varrho(\bar{\tilde{R}}) = \varrho(\overline{I - DJD}) < 1$. We now show that if the zero-diagonal model is walk-summable (WS), then allowing diagonal transformations can only increase the spectral radius; if the zero-diagonal model is not WS, then the model obtained by any diagonal transformations is also non-WS. Thus arbitrary diagonal scaling does not improve the sufficient condition for walk-summability in the scalar case[1]:

Lemma A.2.1 (Optimal diagonal scaling). *Let R have zero diagonal. Then for any diagonal matrix D: if $\varrho(\bar{R}) < 1$, then $\varrho(\bar{R}) = \varrho(\overline{I - J}) \leq \varrho(\overline{I - DJD})$. If $\varrho(\bar{R}) \geq 1$, then $\varrho(\overline{I - DJD}) \geq 1$.*

Proof. Consider the objective function that we would like to minimize:

$$\min_D \varrho(\overline{I - DJD}), \text{ where } D \text{ is diagonal.} \tag{A.6}$$

[1]Furthermore, in Lemma 4.2.6 in Chapter 4 we show that LBP is invariant to such rescalings.

Since R has zeros on the diagonal, we have:

$$\varrho(\overline{I - DJD}) = \varrho(\overline{I - D(I-R)D}) = \varrho(\overline{I - D^2 + DRD}) = \varrho(\overline{I - D^2} + \overline{DRD}) \quad \text{(A.7)}$$

The last equality holds because the matrix $I - D^2$ is diagonal, and the matrix DRD is zero-diagonal, therefore their non-zero entries do not overlap. First, we restrict the diagonal matrices $D = \text{diag}([d_1, ..., d_n])$ to have $d_i \geq 0$. This does not change the value of the optimization problem in (A.6): neither D^2 nor $\overline{DRD}$ are affected by the change of sign of any d_i.

In addition, we can restrict $d_i \leq 1$, without affecting the optimum value of the optimization problem in (A.6): suppose some $d_i > 1$, i.e. $d_i = 1 + e_i$, with $e_i > 0$. Let $\tilde{d}_i = 1 - e_i$, and define $\tilde{D} = \text{diag}(\tilde{d})$. Then $\overline{I - D^2} = \overline{I - \tilde{D}^2}$, because $|1 - d_i| = |-e_i| = e_i$, and $|1 - \tilde{d}_i| = |e_i| = e_i$. However, $\overline{DRD} \geq \overline{\tilde{D}R\tilde{D}}$ (elementwise), and thus

$$\overline{I - D^2} + \overline{DRD} \geq \overline{I - \tilde{D}^2} + \overline{\tilde{D}R\tilde{D}} \quad \text{(A.8)}$$

and

$$\varrho(\overline{I - D^2} + \overline{DRD}) \geq \varrho(\overline{I - \tilde{D}^2} + \overline{\tilde{D}R\tilde{D}}) \quad \text{(A.9)}$$

Hence whenever some $d_i = 1 + e_i > 1$, a better solution can be found by taking $d_i = 1 - e_i < 1$. Therefore we can assume $d_i \in [0, 1]$. This means that $\overline{I - D^2} = I - D^2$. Next, we recall that for positive matrices the spectral radius ϱ is achieved at λ_{max}, the maximum eigenvalue:

$$\varrho(I - D^2 + D\bar{R}D) = \lambda_{max}(I - D^2 + \overline{DRD}) = 1 + \lambda_{max}(D(\bar{R} - I)D) \quad \text{(A.10)}$$

Our problem reduces to minimize $\lambda_{max}(D(\bar{R} - I)D)$, where $d_i \in [0, 1]$. Now consider two cases:

(1) If $\varrho(\bar{R}) < 1$, then $\bar{R} - I \prec 0$, and the optimal D is obtained by maximizing all d_i, i,e, $D = I$.

(2) If $\varrho(\bar{R}) \geq 1$, then $\lambda_{max}(\bar{R} - I) \geq 0$, i.e. $\exists\ x$ such that $x^T(\bar{R} - I)x \geq 0$. Then, since $d_i \geq 0$, we have $x^T D(\bar{R} - I)Dx \geq 0$ and $\lambda_{max}(D(\bar{R} - I)D) \geq 0$. Thus, if J is non-walk-summable, then any DJD is also non-walk-summable. □

■ A.2.2 Proofs for Section 4.2.3

Proof of Lemma 4.2.5 Suppose we have some finite computation tree T with the information matrix J. We use 0 to denote the root-node. The self-return walk-sum at the root node in the tree is equal to the marginal covariance block at the root. Using hat notation '^' to denote marginal quantities, we have that $\hat{P}_0 = (\hat{J}_0)^{-1}$.

Now, suppose we add a new node j to the tree and connect it with node i in the tree by a new edge (i, j) with weight R_{ij}. Then the information matrix of this *u*pdated

tree T^u is $J^u = \begin{bmatrix} J & -R_{ij} \\ -R_{ji} & I \end{bmatrix}$. The root-node marginal of the new tree $\hat{P}_0^u = (\hat{J}_0^u)^{-1}$ is the self-return walk-sum at the root in the new tree. By integrating out the new variable we do not change the marginal at the root node: $\hat{J}^u_{T^u \backslash j} = J - R_{ij}R_{ji}$ (here $T^u \backslash j = T$). We have that $J - R_{ij}R_{ji} \preceq J$, hence $\hat{P}_T^u \succeq P$. Hence the same inequality holds for the principal submatrices corresponding to the root-node: $\hat{P}_0^u \succeq \hat{P}_0$. Thus $\phi(0 \to 0 \mid T^u) \succeq \phi(0 \to 0 \mid T)$. This means that the walk-sum over the additional walks is positive-semi-definite. □

Proof of Proposition 4.2.1 Our proof of Proposition 3.2.4 based on the sum-partition theorem can be extended to the vector case as well. Here we provide an alternative proof which gives more insight into vector-LBP.

Covariances. We first show that the LBP covariance estimates (i.e. covariances over the diagonal blocks) are monotonically increasing and bounded above in the positive-definite sense, and hence converge, as in the scalar case. LBP covariance estimates after k-steps correspond to the root-node covariances in the k-step computation tree. We have shown in Lemma 4.2.5 that after adding a new node the covariances at the root node increase. When going from k-step computation tree to a $k+1$-step computation tree, we are adding one or more nodes (depending on the message schedule), so covariances increase.

Since the model is vector walk-summable, the absolute walk-sum $\bar{\phi}(i \to i)$ converges and is finite. Hence, the absolute walk-sum is an upper bound on the the walk-sum over back-tracking (BT) walks: $\bar{\phi}(i \to i, \mathrm{BT}) \leq \bar{\phi}(i \to i)$, where the inequality is elementwise. Note a slight discrepancy – monotonicity is in terms of a positive-definite ordering, while boundedness is in terms of the elementwise ordering. However, if $0 \leq A$, and $A \leq B$, then $A \preceq B$ by the Perron-Frobenius theorem. Thus we have monotonicity and boundedness in the same positive-definite ordering which immediately implies convergence of self-return walk-sums for the covariances.

Means. Now we establish convergence of the means. In walk-summable models the series $\sum_{l=0}^{\infty} \bar{\phi}_h(* \stackrel{l}{\to} i)$ converges absolutely, as it contains linear combinations of the terms of the absolutely convergent series $\sum_{l=0}^{\infty} \bar{\phi}(j \stackrel{l}{\to} i)$:

$$\sum_{l=0}^{\infty} \bar{\phi}_h(* \stackrel{l}{\to} i) = \sum_{j \in V} \sum_{l=0}^{\infty} \bar{\phi}(j \stackrel{l}{\to} i)|h_j| \tag{A.11}$$

Therefore, the tail of the series, $\sum_{l>n} \bar{\phi}_h(i \stackrel{l}{\to} i)$, containing walk-sums for walks of length exceeding n, approaches zero as n increases: $\lim_{n\to\infty} \bar{\phi}_h(* \stackrel{l>n}{\to} i) = 0$.

The walks which are missing after n steps of LBP are a subset of walks with length exceeding n:[2]

$$\mathcal{W}(* \to i) \setminus \mathcal{W}(* \to i \mid T_i^{[n]}) \subset \mathcal{W}(* \stackrel{l>n}{\to} i) \tag{A.12}$$

[2] After n steps LBP captures all walks of length n as well as many longer walks (which live in $T_i^{(n)}$).

This implies that the absolute walk-sum over missing walks is bounded above by $\bar{\phi}_h(* \overset{l>n}{\to} i)$:

$$\left|\phi_h(* \to i) - \phi_h(* \to i \mid T_i^{[n]})\right| \leq \bar{\phi}_h(* \to i) - \bar{\phi}_h(* \to i \mid T_i^{[n]}) \leq \bar{\phi}_h(* \overset{l>n}{\to} i) \quad \text{(A.13)}$$

As $n \to \infty$, the error bound $\bar{\phi}_h(* \overset{l>n}{\to} i)$ approaches zero proving that belief propagation mean estimates converge and are correct in the limit.[3] $\square$

Proof of Lemma 4.2.8 Let $\hat{Q} = \arg\min_Q \varrho(\overline{QRQ^T})$, and $R_{\hat{Q}} = \hat{Q}R\hat{Q}^T$. If $\min_Q \varrho(\overline{QRQ^T}) < 1$ then $R_{\hat{Q}}$ is scalar walk-summable and by Proposition 3.1.8 it is also scalar-PN with $\varrho(\bar{R}_{\hat{Q}}) = 1 - \epsilon_{\max}(I - R_{\hat{Q}}) < 1$.

A model which is scalar-PN is also vector-PN, and the strength of PN can only increase by blocking: $\epsilon^{vec}_{\max}(I - R_{\hat{Q}}) \geq \epsilon_{\max}(I - R_{\hat{Q}})$. To see this, let $\tilde{e} \in \tilde{\mathcal{E}}$ be the vector edges, and $e \in \mathcal{E}$ the scalar edges. Hence $\tilde{\mathcal{E}}$ is a partition of $\mathcal{E}$, i.e. each scalar edge belongs to one and only one vector edge, $e \in \tilde{e}$. Now $J = \epsilon I + \sum_{\tilde{e}}[J_{\tilde{e}}] = \epsilon I + \sum_{\tilde{e}} \left(\sum_{e \in \tilde{e}}[J_e]\right)$. And $J_{\tilde{e}} = \sum_{e \in \tilde{e}}[J_e]_{\tilde{e}} \succeq 0$.

Finally, the strength of vector-PN is invariant to block-orthogonal transformations: $\epsilon^{vec}_{\max}(I - R_{\hat{Q}}) = \epsilon^{vec}_{\max}(I - \hat{Q}R\hat{Q}^T) = \epsilon^{vec}_{\max}(I - R) > 0$. To see this, take $J = \epsilon I + \sum_e [J_e]$, then $QJQ^T = \epsilon QQ^T + \sum_e Q[J_e]Q^T = \epsilon I + \sum_e [Q_e J Q_e^T]$, with $Q_e J Q_e^T \succeq 0$. We conclude that the model $I - R$ is vector-PN.

For the case of arbitrary block-invertible transformations a similar proof applies. If $\min_S \varrho(\overline{I - SJS^T}) < 1$ then the model SJS^T is scalar walk-summable and hence scalar-PN with $\epsilon_{\max}(SJS^T) > 0$. This implies vector-PN: $\epsilon^{vec}_{\max}(SJS^T) > 0$. The strength of vector-PN does depend on the transformation for non-orthogonal case, but if the model is vector-PN then it remains vector-PN after arbitrary block-invertible transformations. To show this we consider an equivalent definition of vector-PN: a model is vector-PN if there exists a decomposition $J = \sum[J_e]$ with $J_e \succ 0$. Now apply the transformation: $SJS^T = \sum[S_e J_e S_e^T]$, with $S_e J_e S_e^T \succ 0$. Thus if there exists a PN decomposition of J then there also exists one for SJS^T. This implies that $\epsilon^{vec}_{\max}(J) > 0$. $\square$

Proof of Lemma 4.2.10 We use canonical correlation analysis. Since we are dealing with J instead of P we refer to it as canonical partial correlation analysis. Suppose $J = \begin{bmatrix} J_{11} & J_{12} \\ J_{21} & J_{22} \end{bmatrix}$. We can rotate each of the vector-variables to bring the model into the form which displays the canonical partial correlations. Compute the singular value decompositions of the diagonal blocks: $J_{11} = S_1 D_1 S_1^T$ and $J_{22} = S_2 D_2 S_2^T$. Whiten the diagonal blocks by $D_i^{-1/2} S_i^T J_{ii} S_i D_i^{-1/2} = I$. Now compute the SVD of the resulting whitened off-diagonal block: $D_1^{-1/2} S_1^T J_{12} S_2 D_2^{-1/2} = Q_1 D Q_2^T$. Now $-D$ is the (in

[3] The proof applies to the serial version of LBP with computation trees of non-uniform depth: the walks which LBP misses after n steps are all longer than the minimum depth, which grows to ∞ as $n \to \infty$.

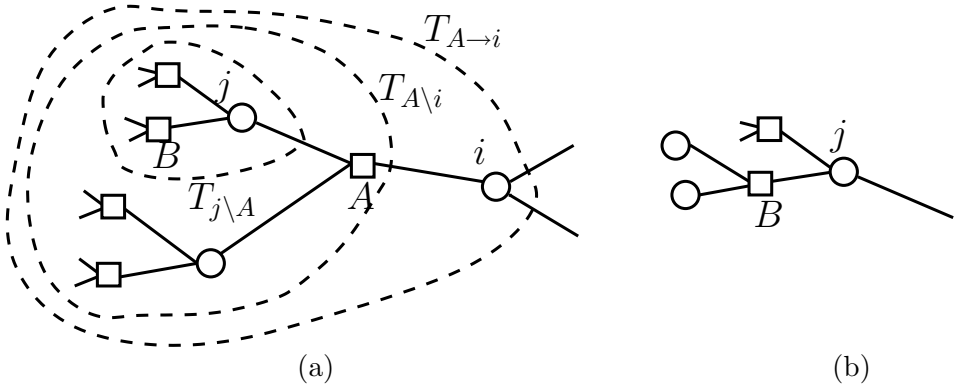


Figure A.2. (a) Illustration of the subtree notation, $T_{A\to i}$ and $T_{A\setminus i}$. (b) B is a leaf-factor node.

general rectangular) matrix with canonical partial correlations on the diagonal, and zeros elsewhere. Let $T = \text{blockdiag}(Q_i D_i^{-1/2} S_i^T)$ then

$$TJT^T = \left[\begin{smallmatrix} I & D \\ D^T & I \end{smallmatrix}\right] \tag{A.14}$$

Now $\bar{R} = \overline{I - TJT^T} = \left[\begin{smallmatrix} 0 & \bar{D} \\ \bar{D}^T & 0 \end{smallmatrix}\right]$. The eigenvalues of $\bar{R}$ are the diagonal elements of $\bar{D}$. Thus, since canonical partial correlations for a valid model are strictly below 1, we have $\varrho(\bar{R}) < 1$. □

■ A.2.3 Walk-sum Interpretation of FG-LBP in Trees

For simplicity we assume that J is normalized to have unit-diagonal, with R and all R_F being zero-diagonal. Also, we use the factorization which separately lists single-node and higher-order factors: $p(x) \propto \prod_{i\in V} \psi_i(x_i) \prod_{F\in F} \psi_F(x_F)$, with $|F| > 1$. We use the following potential specification for LBP: $J = \sum_i [J_i] + \sum_F [J_F]$, with $J_F = -R_F$, and $J_i = 1$. The vector h can in principle be partitioned arbitrarily among the single and higher-order factors, but for convenience, in the following derivations we only use single-node factors for h, and set $h_F = 0$.

Proof of Lemma 4.3.1 First we consider the variances in a tree-structured factor graph model. We show the equivalence of messages in FG-LBP to walk-sums in certain subtrees. Refer to Figure A.2 (a) for the definition of various subtrees. We do not show the self-factors in the figure. We say that a variable node is a leaf if it is connected to at most one higher-order factor. We say that a higher-order factor is a leaf, if it is connected to at most one non-leaf variable node, see Figure A.2 (b). Note that we do not use the term leaf to refer to singleton factors.

Suppose B is a leaf-factor, which is connected to the non-leaf variable j. Using our definition of potentials, and the FG-LBP equations (4.28), the message $\Delta J_{B\to j}$ is equal

to

$$\Delta J_{B\to j} = [J_B]_j - [R_B]_{j,B\backslash j}\ \hat{J}^{-1}_{B\backslash j}\ [R_B]_{B\backslash j,j} = -[R_B]_{j,B\backslash j}\ \hat{J}^{-1}_{B\backslash j}\ [R_B]_{B\backslash j,j} \tag{A.15}$$

Here $\hat{J}_{B\backslash j}$ is equal to $I - [R_B]_{B\backslash j}$ since each of the leaf nodes contributes 1 to the diagonal of $\hat{J}_{B\backslash j}$. We have $[J_B]_j = 0$ since J is normalized and R_B has zero-diagonal.[4] The variance estimate at node j in the subtree $T_{B\to j}$ is equal to

$$P_j(T_{B\to j}) = (J_j - [R_B]_{j,B\backslash j}\ \hat{J}^{-1}_{B\backslash j}\ [R_B]_{B\backslash j,j})^{-1} = \frac{1}{1 - [R_B]_{j,B\backslash j}\ \hat{J}^{-1}_{B\backslash j}\ [R_B]_{B\backslash j,j}} \tag{A.16}$$

The variance is equal to the multiple-revisit self-return walk-sum in the subtree: $P_j(T_{B\to j}) = \phi(j \to j \mid T_{B\to j})$. This means that the single-revisit walk-sum is $\phi(j \overset{\backslash j}{\to} j \mid T_{B\to j}) = [R_B]_{j,B\backslash j}\ \hat{J}^{-1}_{B\backslash j}\ [R_B]_{B\backslash j,j} = -\Delta J_{B\to j}$.

Now consider the factor graph in Figure A.2 (a), and suppose that B is some internal (i.e. non-leaf) factor. We use induction: we assume that all the incoming messages $\Delta J_{B\to j}$ for $B \in \mathcal{N}(j)\backslash A$ to node j correspond to single-revisit walk-sums in the subtree $T_{B\to j}$ (we have already shown this for leaf-factors). We then show that the outgoing messages $\Delta J_{j\to A}$ and $\Delta J_{A\to i}$ have a walk-sum interpretation as well, thus inductively proving the walk-sum interpretation for FG-LBP.

Combining the single-revisit walk-sums over all subtrees $T_{B\to j}$ for $B \in \mathcal{N}(j)\backslash A$, we get the single revisit walk-sum at node j in $T_{j\backslash A}$:

$$\phi(j \overset{\backslash j}{\to} j \mid T_{j\backslash A}) = \sum_{B\in\mathcal{N}(j)\backslash A} \phi(j \overset{\backslash j}{\to} j \mid T_{B\to j}) \tag{A.17}$$

We also have $\Delta J_{j\to A} = 1 + \sum_{B\in\mathcal{N}(j)\backslash A} \Delta J_{B\to j}$, where $\Delta J_{B\to j} = -\phi(j \overset{\backslash j}{\to} j \mid T_{B\to j})$. Hence, $\Delta J_{j\to A} = 1 - \phi(j \overset{\backslash j}{\to} j \mid T_{j\backslash A})$.

Next we compute the multiple-revisit self-return walk-sums at $j \in A\backslash i$ in the subtree $T_{A\backslash i}$. Each eliminated subtree $T_{j\backslash A}$ introduces a self-loop at node j in $[R_A]_j$ – every time the self-loop is traversed, it contributes the weight of all single-revisit walks in the eliminated subtree:

$$\phi(j \to j \mid T_{A\backslash i}) = \left((I - [R_A]_{A\backslash i} - \sum_{j\in\mathcal{N}(A)\backslash i} \phi(j \overset{\backslash j}{\to} j \mid T_{j\backslash A}))^{-1} \right)_{j,j} \tag{A.18}$$

The matrix $\hat{P}_{A\backslash i} = \left(I - [R_A]_{A\backslash i} - \sum_{j\in\mathcal{N}(A)\backslash i} \phi(j \overset{\backslash j}{\to} j \mid T_{j\backslash A}) \right)^{-1}$ contains these walk-sums for all pairs $j_1, j_2 \in A\backslash i$. Finally, the single-revisit walk at node i in $T_{A\to i}$ is

[4] Note that $J_j \neq [J_B]_j$. The first term corresponds to a self-factor, whereas the later is a diagonal element of a higher-order factor.

obtained by adding the first and last step to multiple-revisit walks in $T_{A\backslash i}$, so the walk-sum is $\phi(i \overset{\backslash i}{\rightarrow} i \mid T_{A\rightarrow i}) = [R_A]_{i,A\backslash i}\ \hat{P}_{A\backslash i}\ [R_A]_{A\backslash i,i}$. Now we compare this to the expression for $\Delta J_{A\rightarrow i}$ in (4.28):

$$\Delta J_{A\rightarrow i} = [J_A]_i - [J_A]_{i,A\backslash i}\ \hat{J}^{-1}_{A\backslash i}\ [J_A]_{A\backslash i,i} \tag{A.19}$$

The term $[J_A]_i = 0$ due to our normalization, $[J_A]_{i,A\backslash i} = -[R_A]_{i,A\backslash i}$, and $\hat{J}_{A\backslash i} = ([J_A]_{A\backslash i} + \sum_{j\in\mathcal{N}(A)\backslash i}[\Delta J_{j\rightarrow A}]_{A\backslash i})^{-1}$. Making these substitutions, we obtain that $\Delta J_{A\rightarrow i} = -\phi(i \overset{\backslash i}{\rightarrow} i \mid T_{A\rightarrow i})$, thus establishing the walk-sum interpretation for FG-LBP variances.

Now we repeat the same analysis for the means. The message $\Delta h_{B\rightarrow j} = -\phi_h(* \overset{\backslash j}{\rightarrow} j \mid T_{B\rightarrow j})$ – this is the single-visit walk-sum to node j in $T_{B\rightarrow j}$. Combining the subtrees $T_{B\rightarrow j}$ for all $B \in \mathcal{N}(i)\backslash A$, we obtain $\Delta h_{j\rightarrow A} = h_j + \sum_{B\in\mathcal{N}(j)\backslash A} \Delta h_{B\rightarrow j}$. Hence, $\Delta h_{j\rightarrow A} = h_i - \phi(* \overset{\backslash j}{\rightarrow} j \mid T_{j\backslash A})$.

To get the multiple-revisit walk-sums at node j in $T_{A\backslash i}$ we combine the single-visit walks terminating at j which have walk-sums $\phi_h(* \overset{\backslash j}{\rightarrow} j|T_{A\backslash i})$ and the 0-walk starting at j with weight h_j, with the multiple-revisit self-return walks in $T_{A\backslash i}$. We use the vector $\hat{h}_{A\backslash i}$, which contains $h_j + \phi_h(* \overset{\backslash j}{\rightarrow} j|T_{A\backslash i})$ in entry corresponding to j. Then, $\phi_h(* \rightarrow j \mid T_{A\backslash i}) = [\hat{P}_{A\backslash i}\ \hat{h}_{A\backslash i}]_j$. It is equal to $[\hat{\mu}_{A\backslash i}]_j$.

Finally, the walk-sum $\phi(* \overset{\backslash i}{\rightarrow} i \mid T_{A\rightarrow i}) = [R_A]_{i,A\backslash i}\hat{\mu}_{A\backslash i}$ is obtained by appending a single step (j, A, i) to any multiple-visit walk in $T_{A\backslash i}$. This walk-sum is equal to $-\Delta h_{A\rightarrow i} = [J_A]_{i,A\backslash i}\hat{\mu}_{A\backslash i}$. This completes the argument. □

Proof of Proposition 4.3.2 The proof parallels the proof of the result in the scalar case. First we note that walks in the factor graph which end at node i have a one-to-one correspondence to walks in the FG computation tree which end at the root in the computation tree. All walks for the means are captured in the computation tree, while only those walks for the variances are captured that both start and end at the root in the computation tree – these are the factor graph analog of back-tracking self-return walks. We omit the the proofs of these results, as they closely follow the proofs of Lemmas 3.2.1 and 3.2.2 in the scalar case.

Let $\mathcal{W}(i \overset{BT}{\rightarrow} i)$ denote the back-tracking self-return walks at node i (again we stress that these back-tracking walks are defined with respect to the factor graph computation tree – they have to start and end at the root node). Let $T_i^{(n)}$ denote the n-step computation tree rooted at node i. In the same way as we have done in Sections 3.2.1 and 3.2.2, we can express walk-sums for FG-LBP means and variances as:

$$\begin{aligned}\mathcal{W}(* \rightarrow i) &= \cup_n \mathcal{W}(* \rightarrow 0 \mid T_i^{(n)}) \\ \mathcal{W}(i \overset{BT}{\rightarrow} i) &= \cup_n \mathcal{W}(0 \rightarrow 0 \mid T_i^{(n)})\end{aligned}$$

The computation trees $T_i^{(n)}$ at node i are nested, $T_i^{(n)} \subset T_i^{(n+1)}$ for all n. Hence, $\mathcal{W}(* \to 0|T_i^{(n)}) \subset \mathcal{W}(* \to 0|T_i^{(n+1)})$ and $\mathcal{W}(0 \to 0|T_i^{(n)}) \subset \mathcal{W}(0 \to 0|T_i^{(n+1)})$. Then, since our model is factor graph walk-summable, by Lemma 3.1.2, we obtain the result:

$$\begin{aligned} \mu_i = \phi_h(* \to i) &= \lim_{n\to\infty} \phi_h(* \to 0|T_i^{(n)}) = \lim_{n\to\infty} \hat{\mu}_i^{(n)} \\ P_i^{(BT)} \triangleq \phi(i \overset{BT}{\to} i) &= \lim_{n\to\infty} \phi(0 \to 0|T_i^{(n)}) = \lim_{n\to\infty} \hat{P}_i^{(n)}. \qquad \square \end{aligned}$$

Note that although the theorem parallels the scalar case, there are some differences: unlike the scalar case, in the FG-case some self-return walks in the computation tree may not have positive weights. It is still true however, that as the computation tree grows, the variances at the root-node monotonically increase.

■ A.2.4 Factor Graph Normalizability and LBP

Proof of Proposition 4.3.3 We prove convergence of the variances by showing that they decrease monotonically, and are bounded below.

Step 1. (Boundedness from below) Let $T^{(n)}$ denote the n-step computation tree, and $J^{(n)}$ denote the corresponding information matrix. At each step we are adding positive-definite factors J_f corresponding to the added edges. So the computation tree always has a positive-definite[5] information matrix, $J^{(n)} \succ 0$. It follows that the variance at the root, and hence LBP variance estimates, are bounded below by zero.

Step 2. (Monotonic decrease) Consider the computation tree as it increases from depth n to depth $n+1$. At step n the information matrix is $J^{(n)}$. Let us decompose the variables in the tree into internal variables and the leaves at step n, denoted by $x_{I(n)}$ and $x_{L(n)}$. Then $J^{(n)} = \begin{bmatrix} J_{I(n)} & J_{I(n),L(n)} \\ J_{L(n),I(n)} & J_{L(n)} \end{bmatrix}$. At step $n+1$ new leaves $L(n+1)$ are introduced, and the previous leaves become internal nodes: $I(n+1) = I(n) \cup L(n)$. The information matrix at step $n+1$ becomes:

$$J^{(n+1)} = \begin{bmatrix} J_{I(n)} & J_{I(n),L(n)} & \\ J_{L(n),I(n)} & J_{L(n)}+J_{L(n)}^{+} & J_{L(n),L(n+1)} \\ & J_{L(n+1),L(n)} & J_{L(n+1)} \end{bmatrix}, \quad \text{where} \quad \begin{bmatrix} J_{L(n)}^{+} & J_{L(n),L(n+1)} \\ J_{L(n+1),L(n)} & J_{L(n+1)} \end{bmatrix} \succ 0 \tag{A.20}$$

The second matrix is composed of all the factor potentials that are added from step n to $n+1$. It is positive-definite since every variable is covered by at least one positive-definite factor potential. We need to show that the variance at the root node decreases when going from n to $n+1$.

Marginalize all the leaves at step $n+1$ to get the information matrix $\hat{J}_{I(n+1)}^{(n+1)}$. The matrix $\hat{J}_{I(n+1)}^{(n+1)}$ has the same size as $J^{(n)}$, and gives the same marginal variance at the

[5]The computation tree is connected, so the case $J^{(n)}$ being positive semi-definite but not strictly positive-definite is ruled out.

root node as $J^{(n+1)}$. Now

$$\hat{J}^{(n+1)}_{I(n+1)} = \begin{bmatrix} J_{I(n)} & J_{I(n),L(n)} \\ J_{L(n),I(n)} & J_{L(n)}+A \end{bmatrix}, \quad \text{where} \quad A = J^{+}_{L(n)} - J_{L(n),L(n+1)}\, J^{-1}_{L(n+1)}\, J_{L(n+1),L(n)} \succ 0 \tag{A.21}$$

Here, A is obtained by the Schur complement-formula. $A \succ 0$ since the update matrix (second matrix in (A.20)) is positive-definite. We have shown that $\hat{J}^{(n+1)}_{I(n+1)} \succeq J^{(n)}$. The corresponding covariance matrices (their inverses) for the computation tree satisfy $\hat{P}^{(n+1)}_{I(n+1)} \preceq P^{(n)}$. Thus the principal submatrices (in particular at the root node of the computation tree) satisfy the same relationship: $\hat{P}^{(n+1)}_{0} \preceq \hat{P}^{(n)}_{0}$. Hence variances decrease from step n to step $n+1$. In summary, variances decrease monotonically, and are bounded from below, hence they converge. □

■ A.2.5 Complex Representation of CAR Models

We show how to formulate a thin-plate model (an example of a CAR model) in the complex-valued Gaussian framework of [98] described in Section 4.3.4. The thin plate model is:

$$p(x) \propto \exp\left(-\sum_i \frac{1}{2\sigma_y^2}(y_i - x_i)^2 - \frac{1}{2\sigma_z^2}\sum_{i\in V}(x_i - \frac{1}{d_i}\sum_{j\in N(i)} x_j)^2 \right). \tag{A.22}$$

We introduce an auxiliary variable $z_i = (x_i - \frac{1}{d_i}\sum_{j\in N(i)} x_j)$. Hence $z = Hx$. Now we can construct an equivalent model as follows: $x \sim \mathcal{N}(y, \sigma_y^2 I)$, $z \sim \mathcal{N}(0, \sigma_z^2 I)$, and we have a hard constraint: $z = Hx$. So the joint density can be written as

$$p(x,z) \sim p(y \mid x)p(z)\delta(Hx - z) \tag{A.23}$$

Changing the sign on the term Hx, and following the same steps as in Section 4.3.4, we get a model in the following form:

$$p(x) \propto \int_w e^{-\frac{1}{2}\left[\begin{smallmatrix} x \\ w \end{smallmatrix}\right]^T \begin{pmatrix} \frac{1}{\sigma_y^2} I & -jH^T \\ -jH & \sigma_z^2 I \end{pmatrix} \left[\begin{smallmatrix} x \\ w \end{smallmatrix}\right] + \left[\begin{smallmatrix} y \\ 0 \end{smallmatrix}\right]^T \left[\begin{smallmatrix} x \\ w \end{smallmatrix}\right]} dw \tag{A.24}$$

Upon marginalizing w out of the complex-valued joint function $p(x,w)$ we obtain the correct real-valued marginal density for x. The graph in the above model is bipartite, with edges connecting x variables to w variables.

■ A.3 Details for Chapter 5

Stability of errors in wavelet based approximation We provide the analysis for Proposition 5.2.1. Recall the expression for $\mathbb{E}_\sigma[E_{ii}^2]$ that we obtained in (5.19), and decompose it according to scale, s:

$$\mathbb{E}_\sigma[E_{ii}^2] = \sum_s \sum_{l \in s} \sum_{k \in C(l) \setminus l} R_{ik}^2 W_{il}^2.$$

Since W_l has compact support, W_{il} is non-zero only for some constant (independent of N and s) number of wavelets at each scale that contain i in the support. Let K be an upper bound on this constant. Also, W_{il}^2 at scale s is bounded by 2^{-s} since $\|W_l\|_2 = 1$, and $\psi_{s,k}(t) = \frac{1}{2^{s/2}}\psi(2^{-s}t - k)$. Thus we have:

$$\mathbb{E}_\sigma[E_{ii}^2] \leq K \sum_s \sum_{k \in C(l_s^*) \setminus l_s^*} R_{ik}^2 2^{-s}. \tag{A.25}$$

Here l_s^* is the index that achieves the maximum sum over k at scale s. We bound the other terms in the summation over l by this maximum, giving the factor of K in front.

First, suppose that we are dealing with a one-dimensional GMRF, and that outside the region of disturbance P_{ij} decays exponentially with $d(i,j)$, i.e. $P_{ij} = A\beta^{d(i,j)}$, $|\beta| < 1$. Then the response $R_k(i)$ also decays exponentially with the same decay rate outside the region of disturbance, $R_k(i) = A_s\beta^{d(i,j(k))}$, where $j(k)$ corresponds to the peak of W_k. This happens because exponentials are eigen-functions of LTI filters. However, the constant A_s decreases rapidly with each finer scale. If our wavelet has m vanishing moments then $A_s = O(2^{((s-N_{sc})(m+1/2)})$ for k that belongs to scale s, $s \in \{1, ..., N_{sc}\}$. Recall that N_{sc} is the number of scales we use in the wavelet basis, which depends on the correlation length L of the procees: we set $N_{sc} \propto \log_2(L)$.

We can write $\sum_k R_{ik}^2 = A_s^2 Q_\beta(s)$, where we define $Q_\beta(s) = \sum_{k \in C(l_s) \setminus l_s} \beta^{2d(i,j(k))} = \sum_{n \neq 0} \beta^{2d_s|n|}$, with n indexing the aliased terms, and we use d_s to denote the separation length at scale s. Consider how $Q_\beta(s)$ depends on s. The separation length in our construction is $d_s = d_1 2^{s-1}$, where d_1 is the separation at the finest scale. The number of aliased terms doubles with each finer scale, and the distance between them decreases by a factor of two. For one-dimensional signals this (un-scaled) error roughly doubles with each finer scale: $Q_\beta(s) = \sum_{n \neq 0} \beta^{(|n| d_1 2^s)}$ satisfies $Q_\beta(s+1) \leq \frac{1}{2} Q_\beta(s)$.

Hence $Q_\beta(s) \leq 2^{-(s-1)} Q_\beta(1)$. Note that the term $Q_\beta(1)$ is equal to the error in the original (wavelet-less) construction with separation distance d_1. Putting all the pieces

together, the total error becomes:

$$\begin{aligned} K\sum_{s=1}^{N_{sc}} 2^{-s}\sum_k R_{ik}^2 &\le K\sum_s 2^{-s}A_s^2 Q_\beta(1)2^{(1-s)} \\ &\le 2KQ_\beta(1)\sum_s 2^{-2s}2^{(s-N_{sc})(2m+1)} \\ &\le 2KQ_\beta(1)2^{-2N_{sc}}\sum_{s=1}^{N_{sc}} 2^{(s-N_{sc})(2m-1)} \\ &\le 4KQ_\beta(1)2^{-2N_{sc}}, \quad \text{if } m \ge 1. \end{aligned} \tag{A.26}$$

In the last line, if the number of vanishing moments satisfies $m \ge 1$ then the sum $\sum_{s=1}^{N_{sc}} 2^{(s-N_{sc})(2m-1)} \le 2$ for any N_{sc}. That means that the total error is bounded by a constant multiple of $2^{-2N_{sc}}Q_\beta(1)$. Since $N_{sc} \propto \log_2(L)$, this is roughly $L^{-2}Q_\beta(1)$. As we mentioned $Q_\beta(1)$ roughly corresponds to the error in the standard basis construction (without wavelets). From Section 5.1 we know that $Q_\beta(1)$ is bounded so the errors in wavelet-based construction are also bounded, and it can be seen that using wavelets we get a much smaller error. We also know that by controlling d_1 the error $Q_\beta(1)$ can be made arbitrarily small. Hence, the same is true for the error in the wavelet-based construction.

Now let us consider power-law decay of correlations (again outside of the disturbance region), $P_{ij} = Ad(i,j)^{-p}$, with $p > 0$. In contrast to the exponential decay, the power-law decay changes when wavelets are applied. A wavelet with m vanishing moments acts as local smoothing followed by m-th order differentiation [89], so if P_{ij} decays as $d(i,j)^{-p}$, then $R_k(i)$ decays as $d(i,j(k))^{-(p+m)}$. This means that the tails of $R_k(i)$ decay faster than the tails of P_{ij}. We define $Q_p(s) = \sum_n (2d_s|n|)^{-(p+m)}$. The bound for $Q_p(s)$ in terms of $Q_p(1)$ changes: $\sum_n ((d_s/2)|n|)^{-(p+m)} = 2^{(p+m)}\sum_n (d_s|n|)^{-(p+m)}$, hence $Q_p(s) = 2^{-(p+m)(s-1)}Q_p(1)$. Putting everything together, we have:

$$\begin{aligned} K\sum_{s=1}^{N_{sc}} 2^{-s}R_{ik}^2 &\le K\sum_s 2^{-s}A_s^2 Q_p(1)2^{(p+m)(1-s)} \\ &\le KQ_p(1)2^p\sum_s 2^{-(p+m+1)s}2^{(s-N_{sc})(2m+1)} \\ &\le KQ_p(1)2^p 2^{-(p+m+1)N_{sc}}\sum_{s=1}^{N_{sc}} 2^{(s-N_{sc})(m-p)} \\ &\le KQ_p(1)2^{p+1}2^{-(p+m+1)N_{sc}}, \quad \text{if } m > p+1. \end{aligned} \tag{A.27}$$

In the last line if $m > p+1$ then the sum $\sum_{s=1}^{N_{sc}} 2^{(s-N_{sc})(m-p)} \le 2$, and the total error is bounded by a constant multiple of $2^{-(p+m+1)N_{sc}}Q_p(1)$, or roughly $L^{-(p+m+1)}Q_p(1)$. If $1 \le m < p$, then the sum is dominated by the largest term, $\sum_{s=1}^{N_{sc}} 2^{(s-N_{sc})(m-p)} \approx$

$2^{(p-m)N_{sc}}$, and the total error is a constant multiple of $2^{-(p+m+1)N_{sc}}2^{(p-m)N_{sc}}Q_p(1) = 2^{(-2m-1)N_{sc}}Q_p(1)$, or roughly $L^{-2m}Q_p(1)$.

In either case, the total error is bounded by a small multiple of $Q_p(1)$. For the power-law decay, $Q_p(1)$ is in fact smaller than the error using the standard basis, as using wavelets we change the power of decay from p to $p+m$. Hence $Q_p(1)$ roughly corresponds to the error in the single-scale construction with p replaced by $p+m$. Using our results for the standard basis in Section 5.1 we can conclude that the total errors are bounded, and can be made arbitrarily small by controlling d_1.

Also note that wavelet-based construction is especially advantageos in lattices of higher dimension (with d dimensions): there, the convergence of $Q_p(1)$ requires $p > \frac{d}{2}$, see footnote after equation (5.8). However, with the wavelet construction we only need $p + m > \frac{d}{2}$. This means, that for the case where the errors in the standard-basis construction diverge, we can still make them converge using wavelets with sufficient number of vanishing moments.

Appendix B

Miscelaneous appendix

■ B.1 Properties of Gaussian Models

In this section we summarize some useful facts about Gaussian random vectors, and list important identities from linear algebra.

Gaussian density An N-dimensional jointly Gaussian random vector $x \sim \mathcal{N}(\mu, P)$ has probability density given by:

$$p(x) = \frac{1}{\sqrt{(2\pi)^N \det(P)}} \exp(-\frac{1}{2}(x-\mu)^T P^{-1}(x-\mu)) \tag{B.1}$$

The information form of this density, with information parameters $J = P^{-1}$ and $h = P^{-1}\mu$, is given by:

$$p(x) = \frac{1}{Z} \exp\{-\tfrac{1}{2}x^T Jx + h^T x\} \tag{B.2}$$

where $Z = \exp\left(\frac{1}{2}(N\log(2\pi) - \log\det(J) + h^T J^{-1} h)\right)$.

The entropy is equal to

$$H(p) = -\int p(x)\log p(x)dx = \frac{1}{2}\log\left[(2\pi e)^N \det(P)\right] \tag{B.3}$$

The Kullback-Leibler divergence between two Gaussian random vectors $x_1 \sim \mathcal{N}(\mu_1, P_1)$, $x_2 \sim \mathcal{N}(\mu_2, P_2)$ is given by:

$$D(p_1 \parallel p_2) = \int p_1(x)\log\left(\frac{p_1(x)}{p_2(x)}\right)dx = \tag{B.4}$$

$$\frac{1}{2}\left[\log\left(\det(P_2 P_1^{-1})\right) + \mathrm{tr}(P_2^{-1}P_1) + (\mu_2-\mu_1)^T P_2^{-1}(\mu_2-\mu_1) - N\right] \tag{B.5}$$

Marginal and conditional densities. Suppose that we have a pair of jointly Gaussian random vectors (x, y), with mean and covariance

$$\mu = \mathbb{E}[\begin{pmatrix} x \\ y \end{pmatrix}] = \begin{bmatrix} \mu_x \\ \mu_y \end{bmatrix} \quad \text{and} \quad P = \mathbb{E}[\begin{pmatrix} x \\ y \end{pmatrix}(x\ y)\ ] = \begin{bmatrix} P_x & P_{x,y} \\ P_{y,x} & P_y \end{bmatrix} \tag{B.6}$$

Then the marginal density of x is specified by $\hat{\mu}_x = \mu_x$ and $\hat{P}_x = P_x$, i.e. marginal mean and covariance are submatrices of μ and P. However, the conditional density of x given y is specified by:

$$P_{x|y} = P_x - P_{x,y}P_y^{-1}P_{y,x} \quad \text{and} \quad \mu_{x|y} = \mu_x - P_{x,y}P_x^{-1}(y - \mu_y) \tag{B.7}$$

Now consider the same operations in information form. Suppose that the information parameters of the joint density of x and y are given by:

$$h = \begin{bmatrix} h_x \\ h_y \end{bmatrix} \quad \text{and} \quad J = \begin{bmatrix} J_x & J_{x,y} \\ J_{y,x} & J_y \end{bmatrix} \tag{B.8}$$

Then the conditionals have simple expressions:

$$J_{x|y} = J_x, \quad \text{and} \quad h_{x|y} = h_x - J_{x,y}y. \tag{B.9}$$

However, the marginals are more complicated:

$$\hat{J}_x = J_x - J_{x,y}J_y^{-1}J_{y,x} \quad \text{and} \quad \hat{h}_x = h_x - J_{x,y}J_y^{-1}h_y \tag{B.10}$$

Matrix identities. Consider a partitioned matrix:

$$P = \begin{bmatrix} P_x & P_{x,y} \\ P_{y,x} & P_y \end{bmatrix} \tag{B.11}$$

Let $M = P_x - P_{x,y}P_y^{-1}P_{y,x}$. The matrix M is called the Schur complement of y. The inverse of P is given by

$$P^{-1} = \begin{bmatrix} M^{-1} & -M^{-1}P_{x,y}P_y^{-1} \\ -P_y^{-1}P_{y,x}M^{-1} & P_y^{-1} + P_y^{-1}P_{y,x}M^{-1}P_{x,y}P_y^{-1} \end{bmatrix} \tag{B.12}$$

Also, the following determinant decomposition holds:

$$\det(P) = \det(P_x)\det(P_y - P_{y,x}P_x^{-1}P_{x,y}) \tag{B.13}$$

Finally, we mention the matrix inversion lemma:

$$(A + BCD)^{-1} = A^{-1} - A^{-1}B(C^{-1} + DA^{-1}B)^{-1}DA^{-1} \tag{B.14}$$

■ B.2 Bethe Free Energy for Gaussian Graphical Models

We briefly summarize the variational formulation of inference, the Bethe free energy approximation, and the Gaussian Bethe free energy.

Overview: variational inference and Bethe free energy Consider a factor graph specification of a Gaussian graphical model:

$$p(x) = \frac{1}{Z} \prod_i \psi_i(x_i) \prod_F \psi_F(x_F) \tag{B.15}$$

with variables x_i, $i \in V$ with $V = \{1, .., N\}$. Here $\psi_i(x_i)$ are single-variable factors, and $\psi_F(x_F)$ are higher-order factors, with $F \subset V$ and $|F| > 1$. We define local energies

$$E_i(x_i) = -\log(\psi_i(x_i)), \quad \text{and} \quad E_F(x_F) = -\log(\psi_F(x_F)), \tag{B.16}$$

and the total energy $E(x) = \sum_F E_F(x_F) + \sum_i E_i(x_i)$. For an arbitrary probability distribution $b(x)$ over x define the average energy to be $U(b(x)) = \mathbb{E}_b[E(x)]$ (expectation is taken under $b(x)$), and let $H(b(x)) = -\int b(x) \log b(x) dx$ denote the entropy of $b(x)$. Consider the following functional of a trial probability distribution $b(x)$, called the Gibbs free energy:

$$F_{Gibbs}(b(x)) = U(b(x)) - H(b(x)) \tag{B.17}$$

It can be shown [138] that the minimum of $F_{Gibbs}(\cdot)$ is achieved with $b(x) = \frac{1}{Z} \exp(-E(x))$, i.e. this optimization recovers $p(x)$ in (B.15), and the minimum is $-\log(Z)$. This suggests a variational principle for inference: given ψ_i and ψ_F one can optimize (B.17) and obtain the marginals of $p(x)$. While this variational approach is intractable for general graphs, it suggests a variety of principled approximations [131]. We describe the Bethe free energy approximation, which has been shown to have ties to LBP [138].

Instead of optimizing over full joint distribution $b(x)$, we consider single node beliefs $b_i(x_i)$ and factor beliefs $b_F(x_F)$. While there may or may not exist a joint distribution $b(x)$ with these beliefs as its marginals, we require that at least these beliefs are consistent. We require that if $i \in F$ then $\int p_F(x_F) dx_{F \setminus i} = p_i(x_i)$ for every variable in each factor. Also, $\int p_i(x_i) dx_i = 1$ and $\int p_F(x_F) dx_F = 1$ for every variable and factor.

Bethe free energy is an approximation of the Gibbs free energy, and it is only a function of these local beliefs. Bethe free energy consists of a local average energy term, and an approximate entropy term:

$$F_{Bethe}(b_i, b_F) = U_{Bethe}(b_i, b_F) - H_{Bethe}(b_i, b_F) \tag{B.18}$$

The Bethe average energy term is

$$U_{Bethe} = \sum_F \sum_{x_F} b_F(x_F) E_F(x_F) + \sum_i \sum_{x_i} b_i(x_i) E_i(x_i) \tag{B.19}$$

The Bethe entropy term, H_{Bethe}, is given by:

$$H_{Bethe} = -\sum_i (1 - z_i) \sum_{x_i} b_i(x_i) \log b_i(x_i) - \sum_F b_F(x_F) \log b_F(x_F) \tag{B.20}$$

Here z_i is the number of higher-order factors ψ_F which depend on the variable x_i. Belief propagation has been shown to be related to Bethe free energy: fixed points of LBP are stationary points of the Bethe free energy [63, 138, 139].

Gaussian Bethe free energy We now describe the Bethe free energy for a Gaussian graphical model. Consider a Gaussian distribution in the information form:

$$p(x) = \frac{1}{Z} \exp(-\frac{1}{2} x^T J x + h^T x) \tag{B.21}$$

We consider factorizations of the density according to a factor graph, as described in Section 2.3:

$$p(x) = \frac{1}{Z} \prod_i \psi_i(x_i) \prod_f \psi_F(x_F) = \frac{1}{Z} \prod_i \exp(-\frac{1}{2} A_i x_i^2 + h_i x_i) \prod_F \exp(-\frac{1}{2} x_F^T B_F x_F) \tag{B.22}$$

where $x^T J x = \sum_i A_i x_i^2 + \sum_F x_F^T B_F x_F$.

To obtain the Bethe free energy, introduce Gaussian beliefs $b_i(x_i) \propto \mathcal{N}(x_i; \hat{\mu}_i, \hat{P}_i)$, $b_F(x_F) \propto \mathcal{N}(x_F; \hat{\mu}_F, \hat{P}_F)$, where the notation $\mathcal{N}(x; \mu, P)$ represents a Gaussian density with mean μ and covariance P written explicitly as a function of x. The Gaussian Bethe free energy, $F_{Bethe} = U_{Bethe} - H_{Bethe}$, reduces to:

$$F_{Bethe} = \frac{1}{2} \sum_F \left(\mathrm{tr}[B_F \hat{P}_F] + \hat{\mu}_F^T B_F \hat{\mu}_F \right) + \sum_i \left(\frac{1}{2} A_i (\hat{P}_i + \hat{\mu}_i^2) - \hat{\mu}_i h_i \right) \tag{B.23}$$

$$- \sum_F \frac{1}{2} \log \det(\hat{P}_F) + \frac{1}{2} \sum_i (z_i - 1) \log \hat{P}_i + \mathrm{const} \tag{B.24}$$

Recall that z_i is the number of higher-order factors ψ_F which depend on the variable x_i. The Bethe free energy is minimized subject to consistency constraints on the beliefs; for the Gaussian case that translates into the requirements $[\hat{P}_F]_i = \hat{P}_i$, and $[\hat{\mu}_F]_i = \hat{\mu}_i$, i.e. the diagonal elements of $\hat{P}_F$ have to agree with the corresponding $\hat{P}_i$, and the vectors $\hat{\mu}_F$ have to agree with $\hat{\mu}_i$.

We mention that [37] has considered the Bethe free energy for fractional Gaussian belief propagation (which includes ordinary Gaussian belief propagation as a special case) in the scalar pairwise case, and has shown that when the model is pairwise normalizable, then the Bethe free energy is bounded below, and when the model is not PN, then the Bethe free energy is unbounded.

Bibliography

[1] S. Amari. Information geometry on hierarchy of probability distributions. *IEEE Trans. Information Theory*, 47(5):1701–1711, Jul. 2001.

[2] S. Amari and H. Nagaoka. *Methods of Information geometry*. Oxford University press, 2000.

[3] S. Arnborg, D. G. Corneil, and A. Proskurowski. Complexity of finding embeddings in a k-tree. *SIAM J. Alg. Disc. Meth.*, 8:277–284, 1987.

[4] B. Aspvall and J. R. Gilbert. Graph coloring using eigenvalue decomposition. *SIAM J. Alg. Disc. Meth.*, 5:526–538, 1984.

[5] F. R. Bach and M. I. Jordan. Thin junction trees. In *NIPS*, 2001.

[6] O. Barndorff-Nielsen. *Information and exponential families in statistical theory*. Wiley, 1978.

[7] A. Ben-Tal and A. Nemirovski. *Lectures on Modern Convex Optimization: Analysis, Algorithms, and Engineering Applications*. MPS-SIAM Series on Optimization, SIAM, 2001.

[8] C. Berge. *Graphs and Hypergraphs*. North Holland Publishing Company, 1976.

[9] J. Besag. Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society, Series B*, 36(2):192–225, 1974.

[10] D. Bickson, D. Dolev, and E. Yom-Tov. A Gaussian belief propagation for large scale support vector machines. Technical report, Hebrew University, 2007.

[11] D. Bickson, D. Dolev, and E. Yom-Tov. Solving large scale kernel ridge regression using a Gaussian belief propagation solver. NIPS workshop of efficient machine learning, 2007.

[12] N. Biggs. *Algebraic Graph Theory*. Cambridge University Press, 1993.

[13] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[14] D. M. Blei, T. Griffiths, M. I. Jordan, and J. Tenenbaum. Hierarchical topic models and the nested chinese restaurant process. In *NIPS*, 2004.

[15] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, Mar. 2003.

[16] B. Bollobas. *Modern Graph Theory*. Springer, 1998.

[17] B. Bollobas. *Random Graphs*. Cambridge University Press, 2001.

[18] E. G. Boman, D. Chen, O. Parekh, and S. Toledo. On factor width and symmetric H-matrices. *Linear Algebra and its Applications*, 405, 2005.

[19] C. A. Bouman and M. Shapiro. A multiscale random field model for Bayesian image segmentation. *IEEE Trans. Imag. Proc.*, 3(2), Mar. 1994.

[20] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[21] P. Bremaud. *Markov chains, Gibbs fields, Monte Carlo simulation, and queues*. Springer-Varlag, 2001.

[22] I.N. Bronshtein, K.A. Semendyayev, G. Musiol, and H. Muehlig. *Handbook of mathematics*. Springer, 2007.

[23] D. C. Brydges, J. Frohlich, and A. D. Sokal. The random-walk representation of classical spin systems and correlation inequalities. *Communications in mathematical physics*, 91, 1983.

[24] M. Cetin, L. Chen, J. Fisher, A. Ihler, R. Moses, M. J. Wainwright, and A. S. Willsky. Distributed fusion in sensor networks. *IEEE Signal Processing Magazine*, 23(4):43–55, July 2006.

[25] V. Chandrasekaran, J. K. Johnson, and A. S. Willsky. Adaptive embedded subgraph algorithms using walk-sum analysis. In *NIPS*, 2007.

[26] V. Chandrasekaran, J. K. Johnson, and A. S. Willsky. Estimation in Gaussian graphical models using tractable subgraphs: A walk-sum analysis. *IEEE Trans. Signal Processing*, 2008. To appear.

[27] R. Chellappa and S. Chatterjee. Classification of textures using Gaussian markov random fields. *IEEE Trans. on Acoust, Speech and Signal Proc.*, 33:959–963, 1985.

[28] R. Chellappa and A.K. Jain. *Markov Random Fields: Theory and Application*. Academic Press, 1993.

[29] J. Chen, A. Khisti, D. M. Malioutov, and J. S. Yedidia. Distributed source coding using serially-concatenated-accumulate codes. In *IEEE Inf. Theory Workshop*, Oct. 2004.

[30] M. Chertkov and V. Y. Chernyak. Loop series for discrete statistical models on graphs. *J. Stat. Mech.*, P06009, June 2006.

[31] M. J. Choi. Multiscale Gaussian graphical models and algorithms for large-scale inference. Master's thesis, MIT, 2007.

[32] M. J. Choi, V. Chandrasekaran, D. Malioutov, J. K. Johnson, and A. S. Willsky. Multiscale stochastic modeling for tractable inference and data assimilation. *Computer Methods in Applied Mechanics and Engineering*, 2008. accepted for publication.

[33] C. K. Chow and C. N. Liu. Approximating discrete probability distributions with dependence trees. *IEEE Trans. Information Theory*, 14(3):462–467, 1968.

[34] R. R. Coifman and M. Maggioni. Diffusion wavelets. *Appl. Comp. Harm. Anal.*, 21(1):53–94, 2006.

[35] R. G. Cowell, A. P. Dawid, S. L. Lauritzen, and D. J. Spiegelhater. *Probabilistic Networks and Expert Systems*. Springer, 2003.

[36] N. Cressie. *Statistics for Spatial Data*. Wiley, 1993.

[37] B. Cseke and T. M. Heskes. Bounds on the bethe free energy for Gaussian networks. In *NIPS workshop*, 2007.

[38] I. Csiszar. A geometric interpretation of Darroch and Ratcliff's generalized iterative scaling. *Annals of Statistics*, 17(3):1409–1413, Sep. 1989.

[39] M. Dahleh, M. A. Dahleh, and G. Verghese. Lectures on dynamic systems and control. 6.241 course notes. Massachusetts Institute of Technology.

[40] N. Darroch and D. Ratcliff. Generalized iterative scaling for log-linear models. *Annals of Math. Statistics*, 43:1470–1480, 1972.

[41] A. d'Aspremont, O. Banerjee, and L. El Ghaoui. First-order methods for sparse covariance selection. *SIAM Journal on Matrix Analysis and its Applications*, 30(1):56–66, Feb. 2008.

[42] I. Daubechies. *Ten Lectures on Wavelets*. SIAM, 1992.

[43] A. P. Dempster. Covariance selection. *Biometrics*, 28:157–175, March 1972.

[44] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Stat. Society. Series B*, 39(1):1–38, 1977.

[45] R. W. Dijkerman and R. R. Mazumdar. Wavelet representation of stochastic processes and multiresolution stochastic models. *IEEE Trans. Signal Processing*, 42(7), July 1994.

[46] A. Dobra, B. Jones, C. Hans, J. Nevins, and M. West. Sparse graphical models for exploring gene expression data. *Journal of Multivariate Analysis, special issue on Multivariate Methods in Genomic Data Analysis*, 90:196–212, 2003.

[47] P. Doukhan, G. Oppenheim, and M. S. Taqqu. *Theory and Applications of Long-range dependence*. Birkhauser, 2003.

[48] I. S. Duff, Erisman A. M., and J. K. Reid. *Direct Methods for Sparse Matrices*. Monographs on Numerical Analysis. Oxford University Press, 1989.

[49] M. E. Fisher. Critical temperatures of anisotropic Ising lattices ii, general upper bounds. *Physical Review*, 162(2), 1967.

[50] P. Flandrin. Wavelet analysis and synthesis of fractional Brownian motion. *IEEE Trans. Information Theory*, 38(2), Mar. 1992.

[51] William T. Freeman, Egon C. Pasztor, and Owen T. Carmichael. Learning low-level vision. *IJCV*, 40(1):25–47, 2000.

[52] B. Frey and F. Kschischang. Probability propagation and iterative decoding. In *Proc. Allerton Conf. Communications, Control and Computing*, pages 482–493, Oct. 1996.

[53] B. J. Frey, R. Koetter, and N. Petrovic. Very loopy belief propagation for unwrapping phase images. In *NIPS*, 2001.

[54] N. Friedman. Inferring cellular networks using probabilistic graphical models. *Science*, 303:799–805, 2004.

[55] R. G. Gallager. *Low-density parity check codes*. MIT Press, 1963.

[56] B. Gidas. A renormalization group approach to image processing problems. *IEEE Trans. Patt. Anal. Mach. Intell.*, 11(2), Feb. 1989.

[57] W.R. Gilks, S. Richardson, and D. Spiegelhalter. *Markov Chain Monte Carlo in Practice*. Chapman and Hall, 1996.

[58] R. Godement. *Analysis I*. Springer-Verlag, 2004.

[59] J. Hammersley and P. Clifford. Markov fields on finite graphs and lattices. Unpublished manuscript, 1971.

[60] G. H. Hardy. *Divergent Series*. Clarendon Press, Oxford, 1949.

[61] L. He, X. Liu, and G. Strang. Laplacian eigenvalues of growing trees. In *Conf. on Math. Theory of Networks and Systems*, 2000.

[62] P. Heggernes. Minimal triangulations of graphs: a survey. *Discrete Mathematics*, 306(3):297–317, 2006.

[63] T. M. Heskes. Stable fixed points of loopy belief propagation are minima of the Bethe free energy. In *NIPS*, 2002.

[64] R. Horn and C. Johnson. *Matrix Analysis*. Cambridge University Press, 1985.

[65] R. Horn and C. Johnson. *Topics in Matrix Analysis*. Cambridge University Press, 1991.

[66] A. Ihler. *Inference in Sensor Networks: Graphical Models and Particle Methods*. PhD thesis, MIT, June 2005.

[67] A. Ihler. Accuracy bounds for belief propagation. In *UAI*, 2007.

[68] A. Ihler and V. Chandrasekaran. private communication.

[69] A. Ihler, J. Fisher III, and A. Willsky. Message errors in belief propagation. In *NIPS*, 2004.

[70] C. T. Ireland and S. Kullback. Contingency tables with given marginals. *Biometrika*, 55:179–188, 1968.

[71] F. V. Jensen. *An introduction to Bayesian networks*. Springer, 1996.

[72] J. K. Johnson. Walk-summable Gauss-Markov random fields. Unpublished Manuscript, available at `http://www.mit.edu/people/jasonj`, December 2001.

[73] J. K. Johnson. *Convex Relaxation Methods for Graphical Models: Lagrangian and Maximum-Entropy Approaches*. PhD thesis, MIT, July 2008.

[74] J. K. Johnson, Malioutov D. M., and A. S. Willsky. Walk-sum interpretation and analysis of Gaussian belief propagation. In *NIPS*, 2006.

[75] J. K Johnson, D. M. Malioutov, and A. S. Willsky. Lagrangian relaxation for MAP estimation in graphical models. In *Proc. Allerton Conf. Communications, Control and Computing*, 2007.

[76] J.K. Johnson and A.S. Willsky. A recursive model-reduction method for approximate inference in Gaussian Markov random fields. *IEEE Trans. Imag. Proc.*, 17(1), January 2008.

[77] B. Jones and M. West. Covariance decomposition in undirected Gaussian graphical models. *Biometrika*, 92(4), 2005.

[78] M. I. Jordan. Graphical models. *Statistical Science, Special Issue on Bayesian Statistics*, 19:140–155, 2004.

[79] K. Jung and D. Shah. Approximate message-passing inference algorithm. In *Information Theory Workshop*, Sep. 2007.

[80] T. Kailath, A. H. Sayed, and B. Hassibi. *Linear Estimation*. Prentice Hall, 2000.

[81] S. Kirkland, J. J. McDonald, and M. Tsatsomeros. Sign patterns that require positive eigenvalues. *Linear and Multilinear Algebra*, 41, 1996.

[82] F. Kschischang, B. J. Frey, and H. A Loeliger. Factor graphs and the sum-product algorithm. *IEEE Trans. Information Theory*, 47(2):498–519, Feb. 2001.

[83] S. Lauritzen. *Graphical Models*. Oxford Statistical Science Series. Oxford University Press, 1996.

[84] S. L. Lauritzen and N. S. Sheehan. Graphical models in genetics. In D. J. Balding, M. Bishop, and C. Cannings, editors, *Handbook of Statistical Genetics*, pages 808–842. Wiley, 2007.

[85] D. M. Malioutov, J. K. Johnson, and A. S. Willsky. Low-rank variance estimation in large-scale GMRF models. In *IEEE ICASSP*, May 2006.

[86] D. M. Malioutov, J. K. Johnson, and A. S. Willsky. Walk-sums and belief propagation in Gaussian graphical models. *Journal of Machine Learning Research*, 7, Oct. 2006.

[87] D. M. Malioutov, J. K. Johnson, and A. S. Willsky. GMRF variance approximation using spliced wavelet bases. In *IEEE ICASSP*, April 2007.

[88] D. M. Malioutov, J.K. Johnson, M. J. Choi, and A. S. Willsky. Low-rank variance approximation in GMRF models: Single and multi-scale approaches. *under review for IEEE Transactions on Signal Processing*, 2008.

[89] S. Mallat. *A wavelet tour of signal processing*. Academic Press, 1998.

[90] E. Masry. The wavelet transform of stochastic processes with stationary increments and its application to fractional Brownian motion. *IEEE Trans. Information Theory*, 39(1), Jan. 1993.

[91] R. J. McElice, D. J. C. MacKay, and J. F. Cheng. Turbo decoding as an instance of Pearl's belief propagation algorithm. *IEEE J. Select. Areas Commun.*, 16:140–152, Feb. 1998.

[92] N. Meinshausen and P. Buhlmann. High dimensional graphs and variable selection with the lasso. *Annals of Statistics*, 34(3):1436–1462, 2006.

[93] M. Mezard. Statistical physics of the glass phase. *Physica, A: Statistical Mechanics and its Applications*, 306:25–38, Apr. 2002.

[94] M. Mezard and R. Zecchina. Random k-satisfiability: from an analytic solution to an efficient algorithm. *Physical Review: E*, 66, Nov. 2002.

[95] T. Minka. Expectation propagation for approximate Bayesian inference. In *Proceedings of the seventeenth conference on Uncertainty in Artificial Intelligence*, 2001.

[96] C. Moallemi and B. Van Roy. Consensus propagation. In *NIPS*, 2006.

[97] C. Moallemi and B. Van Roy. Convergence of the min-sum message passing algorithm for quadratic optimization. Manuscript, 2006.

[98] A. Montanari, B. Prabhakar, and D. Tse. Belief propagation based multi–user detection. In *Proc. Allerton Conf. Communications, Control and Computing*, Oct. 2005. E-print: cs.IT/0510044.

[99] J. Mooij and H. Kappen. Sufficient conditions for convergence of loopy belief propagation. In *Proc. UAI*, 2005.

[100] K. Murphy, A. Torralba, and W. T. Freeman. Using the forest to see the trees: a graphical model relating features, objects and scenes. In *NIPS*, 2003.

[101] K. Murphy, Y. Weiss, and M. Jordan. Loopy belief propagation for approximate inference: an empirical study. In *UAI*, 1999.

[102] A. V. Oppenheim and A. S. Willsky. *Signals and Systems*. Prentice Hall, 1997.

[103] J. Pearl. *Probabilistic inference in intelligent systems*. Morgan Kaufmann, 1988.

[104] Alessandro Pelizzola. Cluster variation method in statistical physics and probabilistic graphical models. *Journal of Physics A: Mathematical and General*, 38(33):R309–R339, 2005.

[105] K. Plarre and P. R. Kumar. Extended message passing algorithm for inference in loopy Gaussian graphical models. *Ad Hoc Networks*, 2:153–169, 2004.

[106] L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. IEEE*, 77(2):257–286, 1989.

[107] T. Richardson and R. Urbanke. The capacity of low-density parity check codes under message- passing decoding. *IEEE Trans. Information Theory*, 47(5):599–619, 2001.

[108] C. Robert and G. Casella. *Monte Carlo Statistical Methods*. Springer, 2004.

[109] W. Rudin. *Principles of Mathematical Analysis*. McGraw Hill, 3rd edition, 1976.

[110] H. Rue and L. Held. *Gaussian Markov Random Fields Theory and Applications*. Chapman and Hall, CRC, 2005.

[111] P. Rusmevichientong and B. Van Roy. An analysis of belief propagation on the turbo decoding graph with Gaussian densities. *IEEE Trans. Information Theory*, 48(2):745–765, Feb. 2001.

[112] S. R. Sanghavi, D. M. Malioutov, and A. S. Willsky. Linear programming analysis of loopy belief propagation for weighted matching. In *NIPS*, 2007.

[113] S. R. Sanghavi, D. Shah, and A. S. Willsky. Message passing for max-weight independent set. In *NIPS*, 2007.

[114] F. Sha and F. Pereira. Shallow parsing with conditional random fields. In *Proc. HLT/NAACL-03*, 2003.

[115] T. Speed and H. Kiiveri. Gaussian Markov distributions over finite graphs. *The Annals of Statistics*, 14(1), 1986.

[116] D. Spielman and S. H. Teng. Nearly-linear time algorithms for graph partitioning, graph sparsification, and solving linear systems. In *Proc. ACM Symposium on Theory of Computing*, 2004.

[117] N. Srebro. Maximum likelihood Markov networks: An algorithmic approach. Master's thesis, MIT, 2000.

[118] R. P. Stanley. *Enumerative combinatorics*. Cambridge University Press, 1997.

[119] E. Sudderth, A. Ihler, W. T. Freeman, and A. S. Willsky. Nonparametric belief propagation. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, 2003.

[120] E. Sudderth, M. J. Wainwright, and A. S. Willsky. Embedded trees: Estimation of Gaussian processes on graphs with cycles. *IEEE Trans. Signal Proc.*, 52:3136–3150, Nov. 2004.

[121] E. B. Sudderth. *Graphical Models for Visual Object Recognition and Tracking*. PhD thesis, MIT, Dec. 2006.

[122] E. B. Sudderth, M. J. Wainwright, and A. S. Willsky. Loop series and Bethe variational bounds in attractive graphical models. In *NIPS*, 2007.

[123] M. M. Syslo. On cycle bases of a graph. *Networks*, 9:123–132, 1979.

[124] S. C. Tatikonda and M. I. Jordan. Loopy belief propagation and Gibbs measures. In *UAI*, 2002.

[125] S. ten Brink. Convergence of iterative decoding. *Electronics Letters*, 35(10), May 1999.

[126] U. Trottenberg, C.W. Oosterlee, and A. Schuller. *Multigrid*. Academic Press, 2001.

[127] R. S. Varga. *Matrix iterative analysis*. Springer-Verlag, 2000.

[128] P. O. Vontobel. Interior-point algorithms for linear-programming decoding. In *Proc. Information Theory and its Applications*, 2008. Available at http://arxiv.org/abs/0802.1369.

[129] M. Wainwright, T. Jaakkola, and A. Willsky. Tree-based reparameterization framework for analysis of sum-product and related algorithms. *IEEE Trans. Information Theory*, 49(5), 2003.

[130] M. J. Wainwright, T. S. Jaakkola, and A. S. Willsky. MAP estimation via agreement on (hyper)trees: Message-passing and linear-programming approaches. *IEEE Trans. on Inf. Theory*, 51(11), Nov. 2005.

[131] M. J. Wainwright and M. I. Jordan. A variational principle for graphical models. In S. Haykin, J. Principe, T. Sejnowski, and J. McWhirter, editors, *New Directions in Statistical Signal Processing: From Systems to Brain*. MIT Press, 2005.

[132] M. J. Wainwright, P. Ravikumar, and J. Lafferty. High-dimensional graphical model selection using l1-regularized logistic regression. In *NIPS*, 2006.

[133] Y. Weiss and W. Freeman. Correctness of belief propagation in Gaussian graphical models of arbitrary topology. *Neural Computation*, 13, 2001.

[134] Y. Weiss and W. T. Freeman. On the optimality of solutions of the max-product belief propagation algorithm in arbitrary graphs. *IEEE Trans. on Inf. Theory*, 47(2):723–735, 2001.

[135] A. S. Willsky. Multiresolution Markov models for signal and image processing. *IEEE Proceedings*, 90(8):1396–1458, Aug 2002.

[136] J. Winn and C. M. Bishop. Variational message passing. *Journal of Machine Learning Research*, 6:661–694, 2005.

[137] C. Yanover and Y. Weiss. Approximate inference and protein folding. In *NIPS*, 2002.

[138] J. Yedidia, W. Freeman, and Y. Weiss. Understanding belief propagation and its generalizations. *Exploring AI in the new millennium*, 2003.

[139] J. S. Yedidia, W. T. Freeman, and Y. Weiss. Constructing free-energy approximations and generalized belief propagation algorithms. *IEEE Trans. Inf. Theory*, 51(7):2282–2312, July 2005.

[140] A. H. Zemanian. *Distribution theory and transform analysis*. McGraw-Hill, 1965.