Feedback Message Passing for Inference in Gaussian Graphical Models

Ying Liu, *Student Member, IEEE*, Venkat Chandrasekaran, *Member, IEEE*, Animashree Anandkumar, *Member, IEEE*, and Alan S. Willsky, *Fellow, IEEE*

***Abstract*—While loopy belief propagation (LBP) performs reasonably well for inference in some Gaussian graphical models with cycles, its performance is unsatisfactory for many others. In particular for some models LBP does not converge, and in general when it does converge, the computed variances are incorrect (except for cycle-free graphs for which belief propagation (BP) is non-iterative and exact). In this paper we propose *feedback message passing* (FMP), a message-passing algorithm that makes use of a special set of vertices (called a *feedback vertex set* or *FVS*) whose removal results in a cycle-free graph. In FMP, standard BP is employed several times on the cycle-free subgraph excluding the FVS while a special message-passing scheme is used for the nodes in the FVS. The computational complexity of exact inference is $\mathcal{O}(k^2 n)$, where k is the number of feedback nodes, and n is the total number of nodes. When the size of the FVS is very large, FMP is computationally costly. Hence we propose *approximate FMP*, where a pseudo-FVS is used instead of an FVS, and where inference in the non-cycle-free graph obtained by removing the pseudo-FVS is carried out approximately using LBP. We show that, when approximate FMP converges, it yields exact means *and* variances on the pseudo-FVS and exact means throughout the remainder of the graph. We also provide theoretical results on the convergence and accuracy of approximate FMP. In particular, we prove error bounds on variance computation. Based on these theoretical results, we design efficient algorithms to select a pseudo-FVS of bounded size. The choice of the pseudo-FVS allows us to explicitly trade off between efficiency and accuracy. Experimental results show that using a pseudo-FVS of size no larger than $\log(n)$, this procedure converges much more often, more quickly, and provides more accurate results than LBP on the entire graph.**

***Index Terms*—Belief propagation, feedback vertex set, Gaussian graphical models, graphs with cycles, Markov random field.**

Manuscript received May 09, 2011; revised January 09, 2012; accepted April 03, 2012. Date of publication May 03, 2012; date of current version July 10, 2012. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Raviv Raich. This research was supported in part by AFOSR through Grant FA9550-08-1-1080 and in part by Shell International Exploration and Production, Inc. This paper was presented in part at the International Symposium of Information Theory, Austin, Texas, 2010 [1].

Y. Liu and A. S. Willsky are with the Stochastic Systems Group, Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: liu_ying@mit.edu; willsky@mit.edu).

V. Chandrasekaran is with the Department of Computing and Mathematical Sciences at the California Institute of Technology, Pasadena, CA 91125 USA (e-mail: venkatc@caltech.edu).

A. Anandkumar is with the Center for Pervasive Communications and Computing, University of California, Irvine, CA 92697 USA (e-mail: a.anandkumar@uci.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSP.2012.2195656

I. Introduction

GAUSSIAN graphical models are used to represent the conditional independence relationships among a collection of normally distributed random variables. They are widely used in many fields such as computer vision and image processing [2], gene regulatory networks [3], medical diagnostics [4], oceanography [5], and communication systems [6]. Inference in Gaussian graphical models refers to the problem of estimating the means and variances of all random variables given the model parameters in information form (see Section II-A for more details). Exact inference results in Gaussian graphical models of moderate size can be obtained by algorithms such as direct matrix inversion, nested dissection, Cholesky factorization, and Gauss–Seidel iteration [7]. However, these algorithms can be computationally prohibitive for very large problems involving millions of random variables, especially if variances are sought [5], [8], [9]. The development of efficient algorithms for solving such large-scale inference problems is thus of great practical importance.

Belief propagation (BP) is an efficient message-passing algorithm that gives exact inference results in linear time for tree-structured graphs [10]. The Kalman filter for linear Gaussian estimation and the forward-backward algorithm for hidden Markov models can be viewed as special instances of BP. Though widely used, tree-structured models (also known as cycle-free graphical models) possess limited modeling capabilities, and many stochastic processes and random fields arising in real-world applications cannot be well-modeled using cycle-free graphs.

Loopy belief propagation (LBP) is an application of BP on loopy graphs using the same local message update rules. Empirically, it has been observed that LBP performs reasonably well for certain graphs with cycles [11], [12]. Indeed, the decoding method employed for turbo codes has also been shown to be a successful instance of LBP [13]. A desirable property of LBP is its distributed nature—as in BP, message updates in LBP only involve local model parameters or local messages, so all nodes can update their messages in parallel.

However, the convergence and correctness of LBP are not guaranteed in general, and many researchers have attempted to study the performance of LBP [14]–[17]. For Gaussian graphical models, even if LBP converges, it is known that only the means converge to the correct values while the variances obtained are incorrect in general [15]. In [17], a walk-sum analysis framework is proposed to analyze the performance of LBP in Gaussian graphical models. Based on such a walk-sum analysis, other algorithms have been proposed to obtain better inference results [18].

LBP has fundamental limitations when applied to graphs with cycles: Local information cannot capture the global structure of cycles, and thus can lead to convergence problems and inference errors. There are several questions that arise naturally: Can we use more memory to track the paths of messages? Are there some nodes that are more important than other nodes in terms of reducing inference errors? Can we design an algorithm accordingly without losing too much decentralization?

Motivated by these questions, we consider a particular set of "important" nodes called a *feedback vertex set* (FVS). A feedback vertex set is a subset of vertices whose removal breaks all the cycles in a graph. In our *feedback message passing* (FMP) algorithm, nodes in the FVS use a different message passing scheme than other nodes. More specifically, the algorithm we develop consists of several stages. In the first stage on the cycle-free graph (i.e., that excluding the FVS), we employ standard inference algorithms such as BP but in a non-standard manner: Incorrect estimates for the nodes in the cycle-free portion are computed while other quantities are calculated and then fed back to the FVS. In the second stage, nodes in FVS use these quantities to perform exact mean and variance computations in the FVS and to produce quantities used to initiate the third stage of BP processing on the cycle-free portion in order to correct the means and variances. Though communication among the "important" nodes is needed, the messages within the cycle-free portion are completely local and distributed.[1] If the number of feedback nodes is bounded, the means and variances can be obtained exactly in linear time by using FMP. In general, the complexity is $\mathcal{O}(k^2 n)$, where k is the number of the feedback nodes and n is the total number of nodes.

For graphs with large feedback vertex sets (e.g., for large two-dimensional grids), FMP becomes computationally costly. We develop *approximate FMP* using a pseudo-FVS (i.e., a set of nodes of moderate size that break some but not all of the cycles). The resulting algorithm has the same structure as the exact algorithm except that the inference algorithm on the remainder of the graph, (excluding the pseudo-FVS), which contains cycles, needs to be specified. In this paper we simply use LBP, although any other inference algorithm could also be used. As we will show, assuming convergence of LBP on the remaining graph, the resulting algorithm always yields the correct means *and* variances on the pseudo-FVS, and the correct means elsewhere. Using these results and ideas motivated by the work on walk-summability (WS) [17], we develop simple rules for selecting nodes for the pseudo-FVS in order to ensure and enhance convergence of LBP in the remaining graph (by ensuring WS in the remaining graph) and high accuracy (by ensuring that our algorithm "collects the most significant walks"; see Section II-C for more details). This pseudo-FVS selection algorithm allows us to trade off efficiency and accuracy in a simple and natural manner. Experimental results suggest that this algorithm performs exceedingly well—including for non-WS models for which LBP on the entire graph fails catastrophically—using a pseudo-FVS of size no larger than $\log(n)$.

Inference algorithms based on dividing the nodes of a graphical model into subsets have been explored previously [19], [20]. The approach presented in this paper is distinguished by the fact that our methods can be naturally modified to provide efficient approximate algorithms with theoretical analysis on convergence and error bounds.

The remainder of the paper is organized as follows. In Section II, we first introduce some basic concepts in graph theory and Gaussian graphical models. Then we briefly review BP, LBP, and walk-sum analysis. We also define the notion of an FVS and state some relevant results from the literature. In Section III, we show that for a class of graphs with small FVS, inference problems can be solved efficiently and exactly by FMP. We start with the single feedback node case, and illustrate the algorithm using a concrete example. Then we describe the general algorithm with multiple feedback nodes. We also prove that the algorithm converges and produces correct estimates of the means and variances. In Section IV, we introduce approximate FMP, where we use a pseudo-FVS of bounded size. We also present theoretical results on convergence and accuracy of approximate FMP. Then we provide an algorithm for selecting a good pseudo-FVS. In Section V, we present numerical results. The experiments are performed on two-dimensional grids, which are widely used in various research areas including image processing. We design a series of experiments to analyze the convergence and accuracy of approximate FMP. We also compare the performance of the algorithm with different choices of pseudo-FVS, and demonstrate that excellent performance can be achieved with a pseudo-FVS of modest size chosen in the manner we describe. Finally, in Section VI, we conclude with a discussion of our main contributions and future research directions.

II. Background

A. Gaussian Graphical Models

The set of conditional independence relationships among a collection of random variables can be represented by a graphical model [21]. An undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ consists of a set of nodes (or vertices) $\mathcal{V}$ and a set of edges $\mathcal{E}$. Each node $s \in \mathcal{V}$ corresponds to a random variable x_s. We say that a set $C \subset \mathcal{V}$ separates sets $A, B \subset \mathcal{V}$ if every path connecting A and B passes through C. The random vector[2] $\mathbf{x}_{\mathcal{V}}$ is said to be Markov with respect to $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ if for any subset $A, B, C \subset \mathcal{V}$, where C separates A and B, we have that $\mathbf{x}_A$ and $\mathbf{x}_B$ are independent conditioned on $\mathbf{x}_C$, i.e., $p(\mathbf{x}_A, \mathbf{x}_B|\mathbf{x}_C) = p(\mathbf{x}_A|\mathbf{x}_C)p(\mathbf{x}_A|\mathbf{x}_C)$. Such Markov models on undirected graphs are also commonly referred to as undirected graphical models or Markov random fields.

In a Gaussian graphical model, the random vector $x_{\mathcal{V}}$ is jointly Gaussian. The probability density function of a jointly Gaussian distribution is given by $p(\mathbf{x}) \propto \exp\{-\frac{1}{2}\mathbf{x}^T J\mathbf{x} + \mathbf{h}^T\mathbf{x}\}$, where J is the *information, concentration or precision matrix* and $\mathbf{h}$ is the *potential vector*. We refer to these parameters as the model parameters in information form. The mean vector $\boldsymbol{\mu}$ and covariance matrix P are related to J and $\mathbf{h}$ by $\boldsymbol{\mu} = J^{-1}\mathbf{h}$ and $P = J^{-1}$. For Gaussian graphical models, the graph structure is sparse with respect to the information matrix J, i.e., $J_{i,j} \neq 0$ if and only if there is an edge between i and j. For example, Fig. 1(a) is the underlying

[1] See Section VI for a brief discussion of a recursive extension to the methods developed here that suggests how one can construct fully distributed algorithms.

[2] We use the notation $\mathbf{x}_A$, where $A \subset \mathcal{V}$, to denote the collection of random variables $\{x_s | s \in A\}$.

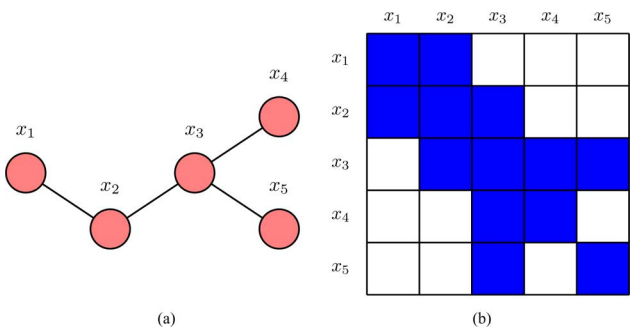


Fig. 1. The relationship between the sparsity pattern in the underlying graph and the sparsity pattern in the information matrix of a Gaussian graphical model. Conditional independence can be directly read from either the sparsity pattern of the graph structure or the sparsity pattern of the information matrix. (a) The sparsity pattern of the underlying graph. (b) The sparsity pattern of the information matrix.

graph for the information matrix J with sparsity pattern shown in Fig. 1(b). For a non-degenerate Gaussian distribution, J is positive definite. The conditional independences of a collection of Gaussian random variables can be read immediately from the graph as well as from the sparsity pattern of the information matrix. If $J_{ij} = 0$, $i \neq j$, then x_i and x_j are independent conditioned on all other variables [22]. Inference in Gaussian graphical models refers to the problem of estimating the means μ_i and variances P_{ii} of every random variable x_i given J and $\mathbf{h}$.

B. Belief Propagation and Loopy Belief Propagation

BP is a message passing algorithm for solving inference problems in graphical models. Messages are updated at each node according to incoming messages from neighboring nodes and local parameters. It is known that for tree-structured graphical models, BP runs in linear time (in the cardinality $n = |\mathcal{V}|$ of the node set) and is exact. When there are cycles in the graph, LBP is used instead, where the same local message update rules as BP are used neglecting the existence of cycles. However, convergence and correctness are not guaranteed when there are cycles.

In Gaussian graphical models, the set of messages can be represented by $\{\Delta J_{i\to j} \cup \Delta h_{i\to j}\}_{(i,j)\in\mathcal{E}}$, where $\Delta J_{i\to j}$ and $\Delta h_{i\to j}$ are scalar values. Consider a Gaussian graphical model: $p(\mathbf{x}) \propto \exp\{-\frac{1}{2}\mathbf{x}^T J\mathbf{x} + \mathbf{h}^T\mathbf{x}\}$. BP (or LBP) proceeds as follows [17]:

1) Message Passing:
The messages are initialized as $\Delta J_{i\to j}^{(0)}$ and $\Delta h_{i\to j}^{(0)}$, for all $(i,j) \in \mathcal{E}$. These initializations may be chosen in different ways. In our experiments we initialize all messages with the value 0.
At each iteration t, the messages are updated based on previous messages as

$$\Delta J_{i\to j}^{(t)} = -J_{ji}\left(\hat{J}_{i\backslash j}^{(t-1)}\right)^{-1} J_{ij} \tag{1}$$

$$\Delta h_{i\to j}^{(t)} = -J_{ji}\left(\hat{J}_{i\backslash j}^{(t-1)}\right)^{-1} \hat{h}_{i\backslash j}^{(t-1)} \tag{2}$$

where

$$\hat{J}_{i\backslash j}^{(t-1)} = J_{ii} + \sum_{k\in\mathcal{N}(i)\backslash j} \Delta J_{k\to i}^{(t-1)} \tag{3}$$

$$\hat{h}_{i\backslash j}^{(t-1)} = h_i + \sum_{k\in\mathcal{N}(i)\backslash j} \Delta h_{k\to i}^{(t-1)}. \tag{4}$$

Here, $\mathcal{N}(i) = \{j \in \mathcal{V} : (i,j) \in \mathcal{E}\}$ denotes the set of neighbors of node i. The fixed-point messages are denoted as $\Delta J_{i\to j}$ and $\Delta h_{i\to j}$ if the messages converge.

2) Computation of Means and Variances:
The variances and means are computed based on the fixed-point messages as

$$\hat{J}_i = J_{ii} + \sum_{k\in\mathcal{N}(i)} \Delta J_{k\to i}, \quad \hat{h}_i = h_i + \sum_{k\in\mathcal{N}(i)} \Delta h_{k\to i}. \tag{5}$$

The variances and means can then be obtained by $P_{ii} = \hat{J}_i^{-1}$ and $\mu_i = \hat{J}_i^{-1}\hat{h}_i$.

C. Walk-Sum Analysis

Computing means and variances for a Gaussian graphical model corresponds to solving a set of linear equations and obtaining the diagonal elements of the inverse of J respectively. There are many ways in which to do this—e.g., by direct solution, or using various iterative methods. As we outline in this section, one way to interpret the exact or approximate solution of this problem is through walk-sum analysis, which is based on a simple power series expansion of J^{-1}. In [17] and [18], walk-sum analysis is used to interpret the computations of means and variances formally as collecting all required "walks" in a graph. The analysis in [17] identifies when LBP fails, in particular when the required walks cannot be summed in arbitrary orders, i.e., when the model is not walk-summable.[3] One of the important benefits of walk-sum analysis is that it allows us to understand what various algorithms compute and relate them to the required exact computations. For example, as shown in [17], LBP collects all of the required walks for the computation of the means (and, hence, always yields the correct means if it converges) but only some of the walks required for variance computations for loopy graphs (so, if it converges, its variance calculations are not correct).

For simplicity, in the rest of the paper, we assume without loss of generality that the information matrix J has been normalized such that all its diagonal elements are equal to unity. Let $R = I - J$, and note that R has zero diagonal. The matrix R is called the *edge-weight matrix*.[4]

A walk of length $l \geq 0$ is defined as a sequence of vertices $w = (w_0, w_1, w_2, \ldots, w_l)$ where each step (w_i, w_{i+1}) is an edge in the graph. The *weight of a walk* is defined as the product of the edge weights,

$$\phi(w) = \prod_{l=1}^{l(w)} R_{w_{l-1},w_l} \tag{6}$$

where $l(w)$ is the length of walk w. Also, we define the weight of a zero-length walk, i.e., a single node, as one. By the Neumann

[3]Walk-summability corresponds to the absolute convergence of the series corresponding to the walk-sums needed for variance computation in a graphical model [17].

[4]The matrix R, which has the same off-diagonal sparsity pattern as J, is a matrix of partial correlation coefficients: R_{ij} is the conditional correlation coefficient between x_i and x_j conditioned on all of the other variables in the graph.

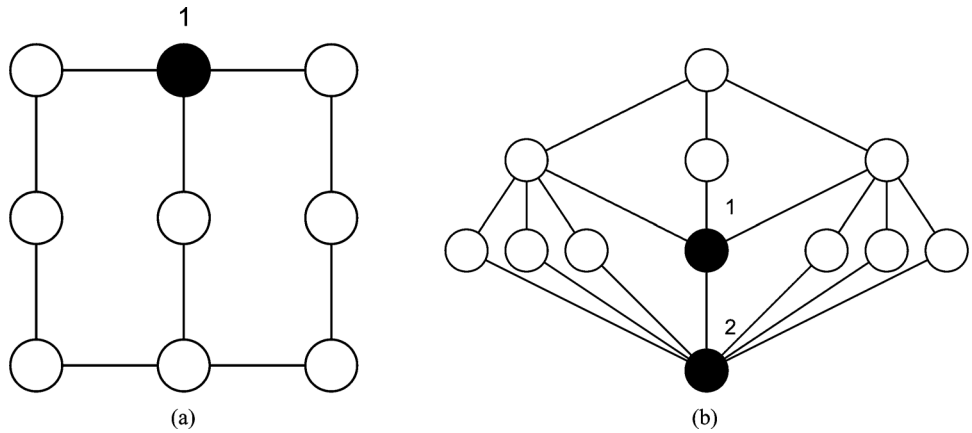


Fig. 2. Examples of FVS's of different sizes. After removing the nodes in an FVS and their incident edges, the reminder of the graph is cycle-free. (a) A graph with an FVS of size one (b) A graph with an FVS of size two.

power series for matrix inversion, the covariance matrix can be expressed as

$$P = J^{-1} = (I - R)^{-1} = \sum_{l=0}^{\infty} R^l. \tag{7}$$

This formal series converges (although not necessarily absolutely) if the spectral radius, $\rho(R)$, i.e., the magnitude of the largest eigenvalue of R, is less than 1.

Let $\mathcal{W}$ be a set of walks. We define the walk-sum of $\mathcal{W}$ as

$$\phi(\mathcal{W}) \triangleq \sum_{w \in \mathcal{W}} \phi(w). \tag{8}$$

We use $\phi(i \to j)$ to denote the sum of all walks from node i to node j. In particular, we call $\phi(i \to i)$ the *self-return walk-sum* of node i. It is easily checked that the (i, j) entry of R^l equals $\phi^l(i \to j)$, the sum of all walks of length l from node i to node j. Hence,

$$P_{ij} = \phi(i \to j) = \sum_{l=0}^{\infty} \phi^l(i \to j). \tag{9}$$

A Gaussian graphical model is *walk-summable* (WS) if for all $i, j \in \mathcal{V}$, the walk-sum $\phi(i \to j)$ converges for any order of the summands in (9) (note that the summation in (9) is ordered by walk-length). In walk-summable models, $\phi(i \to j)$ is well-defined for all $i, j \in \mathcal{V}$. The covariances and the means can be expressed as

$$P_{ij} = \phi(i \to j) \tag{10}$$

$$\mu_i = \sum_{j \in \mathcal{V}} h_j P_{ij} = \sum_{j \in \mathcal{V}} h_j \phi(i \to j). \tag{11}$$

As shown in [17] for non-WS models, LBP may not converge and can, in fact, yield oscillatory variance estimates that take on negative values. Here we list some useful results from [17] that will be used in this paper.

1) The following conditions are equivalent to walk-summability:
   i) $\sum_{w \in \mathcal{W}_{i \to j}} |\phi(w)|$ converges for all $i, j \in \mathcal{V}$, where $\mathcal{W}_{i \to j}$ is the set of walks from i to j.
   ii) $\rho(\bar{R}) < 1$, where $\bar{R}$ is the matrix whose elements are the absolute values of the corresponding elements in R.
2) A Gaussian graphical model is walk-summable if it is attractive, i.e., every edge weight R_{ij} is nonnegative; a valid Gaussian graphical model is walk-summable if the underlying graph is cycle-free.
3) For a walk-summable Gaussian graphical model, LBP converges and gives the correct means.
4) In walk-summable models, the estimated variance from LBP for a node is the sum over all backtracking walks[5], which is a subset of all self-return walks needed for computing the correct variance.

D. Feedback Vertex Set

A *feedback vertex set* (FVS), also called a loop cutset, is defined as a set of vertices whose removal (with the removal of incident edges) results in a cycle-free graph [23]. For example, in Fig. 2(a), node 1 forms an FVS by itself since it breaks all cycles. In Fig. 2(b), the set consisting of nodes 1 and 2 is an FVS.The problem of finding the FVS of the minimum size is called the *minimum feedback vertex set* problem, which has been widely studied in graph theory and computer science. For a general graph, the decision version of the minimum FVS problem, i.e., deciding whether there exists an FVS of size at most k, has been proven to be NP-complete [24]. Finding the minimum FVS for general graphs is still an active research area. To the best of the authors' knowledge, the fastest algorithm for finding the minimum FVS runs in time $\mathcal{O}(1.7548^n)$, where n is the number of nodes [25].

Despite the difficulty of obtaining the minimal FVS, approximate algorithms have been proposed to give an FVS whose size is bounded by a factor times the minimum possible size [26]–[28]. In [28], the authors proposed an algorithm that gives an FVS of size at most two times the minimum size. The complexity of this algorithm is $\mathcal{O}(\min\{m \log n, n^2\})$, where m and n are respectively the number of edges and vertices. In addition, if one is given prior knowledge of the graph structure, optimal or near optimal solutions can be found efficiently or even in linear

[5]A backtracking walk of a node is a self-return walk that can be reduced consecutively to a single node. Each reduction is to replace a subwalk of the form $\{i, j, i\}$ by the single node $\{i\}$. For example, a self-return walk of the form 12321 is backtracking, but a walk of the form 1231 is not.

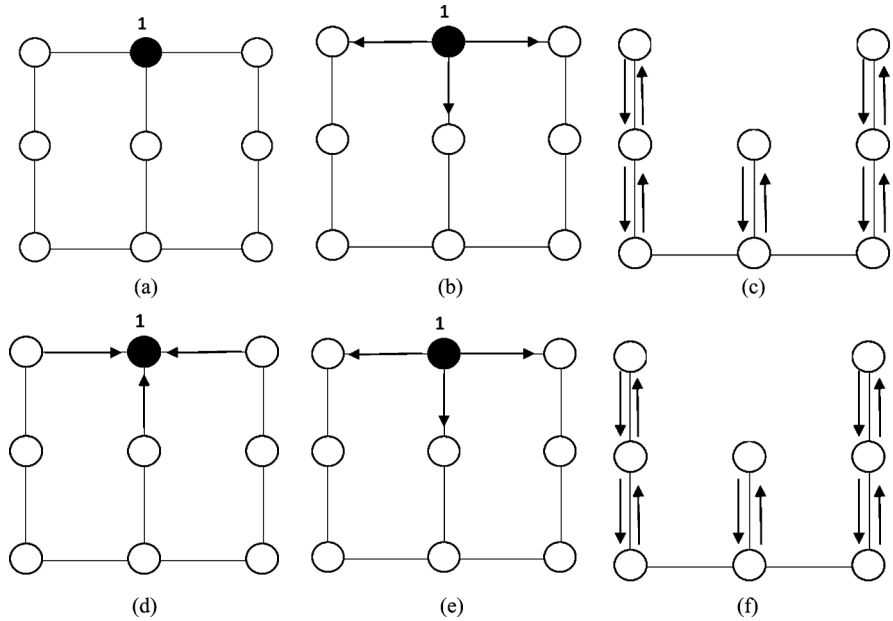


Fig. 3. The FMP algorithm with a single feedback node: (a) A graph with cycles; (b) message initialization; (c) first round of BP; (d) forward messages; (e) feedback messages; and (f) second round of BP.

time for many special graph structures [29]–[31]. Fixed-parameter polynomial-time algorithms are also developed to find the minimum FVS if the minimum size is known to be bounded by a parameter [32].

III. Exact Feedback Message Passing

In this section, we describe the exact FMP algorithm (or simply FMP), which gives the exact inference results for all nodes. We initialize FMP by selecting an FVS, $\mathcal{F}$, using any one of the algorithms mentioned in Section II-D. The nodes in the FVS are called feedback nodes.

We use a special message update scheme for the feedback nodes while using standard BP messages (although, as we will see, not in a standard way) for the non-feedback nodes. In FMP, two rounds of BP message passing are performed with different parameters. In the first round of BP, we obtain inaccurate "partial variances" and "partial means" for the nodes in the cycle-free graph as well as some "feedback gains" for the non-feedback nodes. Next we compute the exact inference results for the feedback nodes. In the second round of standard BP, we make corrections to the "partial variances" and "partial means" of the non-feedback nodes. Exact inference results are then obtained for all nodes.

Before describing FMP, we introduce some notation. With a particular choice, $\mathcal{F}$, of FVS and with $\mathcal{T} = \mathcal{V} \setminus \mathcal{F}$ as the remaining cycle-free graph, we can define submatrices and subvectors respectively of J and $\mathbf{h}$. In particular, let $J_{\mathcal{F}}$ denote the information matrix restricted to nodes in $\mathcal{F}$—i.e., for convenience we assume we have ordered the nodes in the graph so that $\mathcal{F}$ consists of the first k nodes in $\mathcal{V}$, so that $J_{\mathcal{F}}$ corresponds to the upper-left $k \times k$ block of J, and similarly $J_{\mathcal{T}}$, the information matrix restricted to nodes in $\mathcal{T}$ corresponds to the lower right $(n-k) \times (n-k)$ block of J. We can also define $J_{\mathcal{T}\mathcal{F}}$, the lower left cross-information matrix, and its transpose (the upper-right cross-information matrix) $J_{\mathcal{F}\mathcal{T}}$. Analogously we can define the subvectors $\mathbf{h}_{\mathcal{F}}$ and $\mathbf{h}_{\mathcal{T}}$. In addition, for the graph $\mathcal{G}$ and any node j, let $\mathcal{N}(j)$ denote the neighbors of j, i.e., the nodes connected to j by edges.

In this section we first describe FMP for the example in Fig. 3(a), in which the FVS consists of a single node. Then we describe the general FMP algorithm with multiple feedback nodes. We also prove the correctness and analyze the complexity.

A. The Single Feedback Node Case

Consider the loopy graph in Fig. 3(a) and a Gaussian graphical model, with information matrix J and potential vector $\mathbf{h}$, defined on it. In this graph every cycle passes through node 1, and thus node 1 forms an FVS by itself. We use $\mathcal{T}$ to denote the subgraph excluding node 1 and its incident edges. Graph $\mathcal{T}$ is a tree, which does not have any cycles.[6] Using node 1 as the feedback node, FMP consists of the following steps:

Step 1: Initialization

We construct an additional potential vector $\mathbf{h}^1 = J_{\mathcal{T},1}$ on $\mathcal{T}$, i.e., $\mathbf{h}^1$ is the submatrix (column vector) of J with column index 1 and row indices corresponding to $\mathcal{T}$. Note that, since in this case $\mathcal{F} = \{1\}$, this new potential vector is precisely $J_{\mathcal{T}\mathcal{F}}$. In the general case $J_{\mathcal{T}\mathcal{F}}$ will consist of a set of columns, one for each element of the FVS, where each of those columns is indexed by the nodes in $\mathcal{T}$. Note that $h_i^1 = J_{1i}$ for all $i \in \mathcal{N}(1)$ and $h_i^1 = 0$ for all $i \notin \mathcal{N}(1)$. We can view this step as node 1 sending messages to its neighbors to obtain $\mathbf{h}^1$. See Fig. 3(b) for an illustration.

Step 2: First Round of BP on $J_{\mathcal{T}}$ [Fig. 3(c)]

We now perform BP on $\mathcal{T}$ twice, both times using the information matrix $J_{\mathcal{T}}$, but two different potential vectors. The first of these is simply the original potential vector restricted to $\mathcal{T}$, i.e., $\mathbf{h}_{\mathcal{T}}$. The

[6]More generally, the cycle-free graph used in FMP can be a collection of disconnected trees, i.e., a forest.

second uses $\mathbf{h}^1$ as constructed in Step 1.[7] The result of the former of these BP sweeps yields for each node i in $\mathcal{T}$ its "partial variance" $P_{ii}^{\mathcal{T}} = (J_{\mathcal{T}}^{-1})_{ii}$ and its "partial mean" $\mu_i^{\mathcal{T}} = (J_{\mathcal{T}}^{-1}\mathbf{h}_{\mathcal{T}})_i$ by standard BP message passing on $\mathcal{T}$. Note that these results are not the true variances and means since this step does not involve the contributions of node 1. At the same time, BP using $\mathbf{h}^1$ yields a "feedback gain" g_i^1, where $g_i^1 = (J_{\mathcal{T}}^{-1}\mathbf{h}^1)_i$ by standard BP on $\mathcal{T}$.[8] Since $\mathcal{T}$ is a tree-structured graph, BP terminates in linear time.

Step 3: Exact Inference for the Feedback Node

Feedback node 1 collects the "feedback gains" from its neighbors as shown in Fig. 3(d). Node 1 then calculates its *exact* variance and mean as follows:

$$P_{11} = (J_{11} - \sum_{j\in\mathcal{N}(1)} J_{1j}g_j^1)^{-1}, \tag{12}$$

$$\mu_1 = P_{11}(h_1 - \sum_{j\in\mathcal{N}(1)} J_{1j}\mu_j^{\mathcal{T}}). \tag{13}$$

In this step, all the computations involve only the parameters local to node i, the "feedback gains" from, and the "partial means" of node 1's neighbors.

Step 4: Feedback Message Passing [Fig. 3(e)]

After feedback node 1 obtains its own variance and mean, it passes the results to all other nodes in order to correct their "partial variances" $P_{ii}^{\mathcal{T}}$ and "partial means" $\mu_i^{\mathcal{T}}$ computed in Step 2. The neighbors of node 1 revise their node potentials as follows:

$$\widetilde{h}_j = \begin{cases} h_j - J_{1j}\mu_1, & \forall\, j \in \mathcal{N}(1) \\ h_j, & \forall\, j \notin \mathcal{N}(1). \end{cases} \tag{14}$$

From (14) we see that only node 1's neighbors revise their node potentials. The revised potential vector $\widetilde{\mathbf{h}}_{\mathcal{T}}$ and $J_{\mathcal{T}}$ are then used in the second round of BP.

Step 5: Second Round of BP on $J_{\mathcal{T}}$ [Fig. 3(f)]

We perform BP on $\mathcal{T}$ with $J_{\mathcal{T}}$ and $\widetilde{\mathbf{h}}_{\mathcal{T}}$. The means $\mu_i = (J_{\mathcal{T}}^{-1}\widetilde{\mathbf{h}}_{\mathcal{T}})_i$, obtained from this round of BP are the *exact* means. The *exact* variances can be computed by adding correction terms to the "partial variances" as

$$P_{ii} = P_{ii}^{\mathcal{T}} + g_i^1 P_{11} g_i^1, \qquad \forall\, i \in \mathcal{T} \tag{15}$$

where the "partial variance" $P_{ii}^{\mathcal{T}}$ and the "feedback gain" g_i^1 are computed in Step 2. There is only one correction term in this single feedback node case. We will see that when the size of FVS is larger than one, there will be multiple correction terms.

B. Feedback Message Passing for General Graphs

For a general graph, the removal of a single node may not break all cycles. Hence, the FVS may consist of multiple nodes. In this case, the FMP algorithm for a single feedback node can be generalized by adding extra feedback messages, where each extra message corresponds to one extra feedback node in the FVS.

Assume an FVS, $\mathcal{F}$, has been selected, and, as indicated previously, we order the nodes such that $\mathcal{F} = \{1, \ldots, k\}$. The FMP algorithm with multiple feedback nodes is essentially the same as the FMP algorithm with a single feedback node. When there are k feedback nodes, we compute k sets of feedback gains each corresponding to one feedback node. More precisely, Step 1 in the algorithm now involves performing BP on $\mathcal{T}$ $k+1$ times, all with the same information matrix, $J_{\mathcal{T}}$, but with different potential vectors, namely $\mathbf{h}_{\mathcal{T}}$ and $\mathbf{h}^p$, $p = 1, \ldots, k$, where these are the successive columns of $J_{\mathcal{T}\mathcal{F}}$. To obtain the exact inference results for the feedback nodes, we then need to solve an inference problem on a smaller graph, namely $\mathcal{F}$, of size k, so that Step 3 in the algorithm becomes one of solving a k-dimensional linear system. Step 4 then is simply modified from the single-node case to provide a revised potential vector on $\mathcal{T}$ taking into account corrections from each of the nodes in the FVS. Step 5 then involves a single sweep of BP on $\mathcal{T}$ using this revised potential vector to compute the exact means on $\mathcal{T}$, and the feedback gains, together with the variance computation on the FVS, provide corrections to the partial variances for each node in $\mathcal{T}$. The general FMP algorithm with a given FVS $\mathcal{F}$ is summarized in Algorithm 1.

Algorithm 1: The FMP algorithm with a given FVS

Input: information matrix J, potential vector $\mathbf{h}$ and feedback vertex set $\mathcal{F}$ of size k

Output: mean μ_i and variance P_{ii} for every node i

1. Construct k extra potential vectors: $\forall\, p \in \mathcal{F}$, $\mathbf{h}^p = J_{\mathcal{T},p}$, each corresponding to one feedback node.
2. Perform BP on $\mathcal{T}$ with $J_{\mathcal{T}}$, $\mathbf{h}_{\mathcal{T}}$ to obtain $P_{ii}^{\mathcal{T}} = (J_{\mathcal{T}}^{-1})_{ii}$ and $\mu_i^{\mathcal{T}} = (J_{\mathcal{T}}^{-1}\mathbf{h}_{\mathcal{T}})_i$ for each $i \in \mathcal{T}$. With the k extra potential vectors, calculate the feedback gains $g_i^1 = (J_{\mathcal{T}}^{-1}\mathbf{h}^1)_i, g_i^2 = (J_{\mathcal{T}}^{-1}\mathbf{h}^2)_i, \ldots, g_i^k = (J_{\mathcal{T}}^{-1}\mathbf{h}^k)_i$ for $i \in \mathcal{T}$ by BP.
3. Obtain a size-k subgraph with $\widehat{J}_{\mathcal{F}}$ and $\widehat{\mathbf{h}}_{\mathcal{F}}$ given by
$$(\widehat{J}_{\mathcal{F}})_{pq} = J_{pq} - \sum_{j\in\mathcal{N}(p)\cap\mathcal{T}} J_{pj}g_j^q, \ \ \forall\, p, q \in \mathcal{F}$$
$$(\widehat{\mathbf{h}}_{\mathcal{F}})_p = h_p - \sum_{j\in\mathcal{N}(p)\cap\mathcal{T}} J_{pj}\mu_j^{\mathcal{T}}, \ \ \forall\, p \in \mathcal{F}$$
and solve the inference problem on the small graph by $P_{\mathcal{F}} = \widehat{J}_{\mathcal{F}}^{-1}$ and $\boldsymbol{\mu}_{\mathcal{F}} = \widehat{J}_{\mathcal{F}}^{-1}\widehat{\mathbf{h}}_{\mathcal{F}}$.
4. Revise the potential vector on $\mathcal{T}$ by
$$\tilde{h}_i = h_i - \sum_{j\in\mathcal{N}(i)\cap\mathcal{F}} J_{ij}(\boldsymbol{\mu}_{\mathcal{F}})_j, \ \ \forall\, i \in \mathcal{T}.$$
5. Another round of BP with the revised potential vector $\widetilde{\mathbf{h}}_{\mathcal{T}}$ gives the exact means for nodes on $\mathcal{T}$. Add correction terms to obtain the exact variances for nodes in $\mathcal{T}$:
$$P_{ii} = P_{ii}^{\mathcal{T}} + \sum_{p\in\mathcal{F}}\sum_{q\in\mathcal{F}} g_i^p (P_{\mathcal{F}})_{pq} g_i^q, \ \ \forall\, i \in \mathcal{T}.$$

C. Correctness and Complexity of FMP

It is worth noting that BP on a tree, if organized to have computations that proceed from leaf nodes to a common root and then back to leaves, can be interpreted as performing Gaussian elimination, without fill, and back-substitution. An intuition for

[7] Note that since both BP passes here—and, in the general case, the set of $k+1$ BP passes in this step—use the same information matrix, there are economies in the actual BP message-passing as the variance computations are the same for all.

[8] The superscript 1 of g_i^1 means this feedback gain corresponds to the feedback node 1.

FMP is that it performs Gaussian elimination on the tree-part of the graph (excluding the FVS) with fill only among the FVS nodes. The message-passing structure we provide not only realizes this in a way that exposes ties to and non-standard uses of BP-like computations, but also allows us to examine and exploit walk-sum interpretations of the computations. In this subsection, we analyze the correctness and computational complexity of FMP.

Theorem 1: The feedback message passing algorithm described in Algorithm 1 results in the exact means and exact variances for all nodes.

Proof: To make the notation less cluttered, let $J_M = J_{\mathcal{T}\mathcal{F}}$ and J'_M be the transpose of J_M so that we can write

$$J = \begin{bmatrix} J_{\mathcal{F}} & J'_M \\ J_M & J_{\mathcal{T}} \end{bmatrix} \text{ and } \mathbf{h} = \begin{bmatrix} \mathbf{h}_{\mathcal{F}} \\ \mathbf{h}_{\mathcal{T}} \end{bmatrix}. \tag{16}$$

Similarly, we can write

$$P = \begin{bmatrix} P_{\mathcal{F}} & P'_M \\ P_M & P_{\mathcal{T}} \end{bmatrix} \text{ and } \boldsymbol{\mu} = \begin{bmatrix} \boldsymbol{\mu}_{\mathcal{F}} \\ \boldsymbol{\mu}_{\mathcal{T}} \end{bmatrix}. \tag{17}$$

By the construction of $\mathbf{h}^1, \mathbf{h}^2, \ldots, \mathbf{h}^k$ in FMP and (16),

$$J_M = [\mathbf{h}^1, \mathbf{h}^2, \ldots, \mathbf{h}^k]. \tag{18}$$

The feedback gains $\mathbf{g}^1, \mathbf{g}^2, \ldots, \mathbf{g}^k$ in FMP are computed by BP with $\mathbf{h}^1, \mathbf{h}^2, \ldots, \mathbf{h}^k$ as potential vectors. Since BP gives the exact means on trees,

$$[\mathbf{g}^1, \mathbf{g}^2, \ldots, \mathbf{g}^k] = \left[J_{\mathcal{T}}^{-1}\mathbf{h}^1, J_{\mathcal{T}}^{-1}\mathbf{h}^2, \ldots, J_{\mathcal{T}}^{-1}\mathbf{h}^k\right] \tag{19}$$

$$= J_{\mathcal{T}}^{-1} J_M. \tag{20}$$

In FMP, $\boldsymbol{\mu}^{\mathcal{T}}$ is computed by BP with potential vector $\mathbf{h}_{\mathcal{T}}$, so

$$\boldsymbol{\mu}^{\mathcal{T}} = J_{\mathcal{T}}^{-1}\mathbf{h}_{\mathcal{T}}. \tag{21}$$

The diagonal of $J_{\mathcal{T}}^{-1}$ is also calculated exactly in the first round of BP in FMP as $P_{ii}^{\mathcal{T}} = (J_{\mathcal{T}}^{-1})_{ii}$. Since $P = J^{-1}$, by matrix computations, we have

$$P_{\mathcal{T}} = J_{\mathcal{T}}^{-1} + (J_{\mathcal{T}}^{-1} J_M) P_{\mathcal{F}} (J_{\mathcal{T}}^{-1} J_M)'. \tag{22}$$

Substituting (20) into (22), we have

$$P_{ii} = P_{ii}^{\mathcal{T}} + \sum_{p\in\mathcal{F}} \sum_{q\in\mathcal{F}} g_i^p \, (P_{\mathcal{F}})_{pq} \, g_i^q, \quad \forall\, i \in \mathcal{T} \tag{23}$$

where $P_{ii}^{\mathcal{T}}$ is the "partial variance" of node i and g_i^p the "feedback gain" in FMP. Here $P_{\mathcal{F}}$ is the exact covariance matrix of the feedback nodes in $\mathcal{F}$. This is the same equation as in Step 5 of FMP. We need to show that $P_{\mathcal{F}}$ is indeed calculated exactly in FMP. By Schur's complement,

$$\widehat{J}_{\mathcal{F}} \triangleq P_{\mathcal{F}}^{-1} = J_{\mathcal{F}} - J'_M J_{\mathcal{T}}^{-1} J_M \tag{24}$$

$$\widehat{\mathbf{h}}_{\mathcal{F}} \triangleq P_{\mathcal{F}}^{-1}\boldsymbol{\mu}_{\mathcal{F}} = \mathbf{h}_{\mathcal{F}} - J'_M J_{\mathcal{T}}^{-1}\mathbf{h}_{\mathcal{T}}. \tag{25}$$

By (20) and (21),

$$\widehat{J}_{\mathcal{F}} = J_{\mathcal{F}} - J'_M[\mathbf{g}^1, \mathbf{g}^2, \ldots, \mathbf{g}^k] \tag{26}$$

$$\widehat{\mathbf{h}}_{\mathcal{F}} = \mathbf{h}_{\mathcal{F}} - J'_M\boldsymbol{\mu}^{\mathcal{T}} \tag{27}$$

which is exactly the same formula as in Step 3 of FMP. Therefore, we obtain the exact covariance matrix and exact means for nodes in $\mathcal{F}$ by solving $P_{\mathcal{F}} = (\widehat{J}_{\mathcal{F}})^{-1}$ and $\boldsymbol{\mu}_{\mathcal{F}} = P_{\mathcal{F}}\widehat{\mathbf{h}}_{\mathcal{F}}$.

Since $\boldsymbol{\mu} = J^{-1}\mathbf{h}$, from (16) and (17), we can get

$$\boldsymbol{\mu}_{\mathcal{T}} = J_{\mathcal{T}}^{-1}(\mathbf{h}_{\mathcal{T}} - J_M\boldsymbol{\mu}_{\mathcal{F}}). \tag{28}$$

We define $\widetilde{\mathbf{h}}_{\mathcal{T}} = \mathbf{h}_{\mathcal{T}} - J_M\boldsymbol{\mu}_{\mathcal{F}}$, i.e.,

$$(\widetilde{\mathbf{h}}_{\mathcal{T}})_i = h_i - \sum_{j\in\mathcal{N}(i)\cap\mathcal{F}} J_{ij}(\mu_{\mathcal{F}})_j \tag{29}$$

where $\boldsymbol{\mu}_{\mathcal{F}}$ is the exact mean of nodes in $\mathcal{F}$. This step is equivalent to performing BP with parameters $J_{\mathcal{T}}$ and the revised potential vector $\widetilde{\mathbf{h}}_{\mathcal{T}}$ as in Step 4 of FMP. This completes the proof. ■

We now analyze the computational complexity of FMP with k denoting the size of the FVS and n the total number of nodes in the graph. In Steps 1 and 2, BP is performed on $\mathcal{T}$ with $k+2$ messages (one for J, one with $\mathbf{h}_{\mathcal{T}}$, and one for each $\mathbf{h}^p$). The total complexity is $\mathcal{O}(k(n-k))$. In Step 3, $\mathcal{O}(k^2(n-k))$ computations are needed to obtain $\widehat{J}_{\mathcal{F}}$ and $\widehat{h}_{\mathcal{F}}$ and $\mathcal{O}(k^3)$ operations to solve the inference problem on a graph of size k. In Steps 4 and 5, it takes $\mathcal{O}(k(n-k))$ computations to give the exact means and $\mathcal{O}(k^2(n-k))$ computations to add correction terms. Therefore, the total computational complexity of FMP is $\mathcal{O}(k^2 n)$. This is a significant reduction from $\mathcal{O}(n^3)$ of direct matrix inversion when k is small.

IV. Approximate Feedback Message Passing

As we have seen from Theorem 1, FMP always gives correct inference results. However, FMP is intractable if the size of the FVS is very large. This motivates our development of *approximate FMP*, which uses a pseudo-FVS instead of an FVS.

A. Approximate FMP with a Pseudo-FVS

There are at least two steps in FMP which are computationally intensive when k, the size of the FVS, is large: solving a size-k inference problem in Step 3 and adding k^2 correction terms to each non-feedback node in Step 5. One natural approximation is to use a set of feedback nodes of smaller size. We define a *pseudo-FVS* as a subset of an FVS that does not break all the cycles. A useful pseudo-FVS has a small size, but breaks the most "crucial" cycles in terms of the resulting inference errors. We will discuss how to select a good pseudo-FVS in Section IV-D. In this subsection, we assume that a pseudo-FVS is given.

Consider a Gaussian graphical model Markov on a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. We use $\widetilde{\mathcal{F}}$ to denote the given pseudo-FVS, and use $\widetilde{\mathcal{T}}$ to denote the pseudo-tree (i.e., a graph with cycles) obtained by eliminating nodes in $\widetilde{\mathcal{F}}$ from $\mathcal{G}$. With a slight abuse of terminology, we still refer to the nodes in $\widetilde{\mathcal{F}}$ as the feedback nodes. A natural extension is to replace BP by LBP in Step 2 and Step 5 of FMP.[9]

The total complexity of approximate FMP depends on the size of the graph, the cardinality of the pseudo-FVS, and the

[9]Of course, one can insert other algorithms for Steps 2 and 5—e.g., iterative algorithms such as embedded trees [18] which can yield exact answers. However, here we focus on the use of LBP for simplicity.

number of iterations of LBP within the pseudo-tree. Let k be the size of the pseudo-FVS, n be the number of nodes, m be the number of edges in the graph, and D be the maximum number of iterations in Step 2 and Step 5. By a similar analysis as for FMP, the total computational complexity for approximate FMP is $\mathcal{O}(k^2n + kmD)$. Assuming that we are dealing with relatively sparse graphs, so that $m = \mathcal{O}(n)$, reductions in complexity as compared to a use of a full FVS rely on both k and D being of moderate size. Of course the choices of those quantities must also take into account the tradeoff with the accuracy of the computations.

B. Convergence and Accuracy

In this subsection, we provide theoretical results on convergence and accuracy of approximate FMP. We first provide a result assuming convergence that makes several crucial points, namely on the exactness of means throughout the entire graph, the exactness of variances on the pseudo-FVS, and on the interpretation of the variances on the remainder of the graph as augmenting the LBP computation with a rich set of additional walks, roughly speaking those that go through the pseudo-FVS:

Theorem 2: Consider a Gaussian graphical model with parameters J and $\mathbf{h}$. If approximate FMP converges with a pseudo-FVS $\widetilde{\mathcal{F}}$, it gives the correct means for all nodes and the correct variances on the pseudo-FVS. The variance of node i in $\widetilde{\mathcal{T}}$ calculated by this algorithm equals the sum of all the backtracking walks of node i within $\widetilde{\mathcal{T}}$ plus all the self-return walks of node i that visit $\widetilde{\mathcal{F}}$, so that the only walks missed in the computation of the variance at node i are the non-backtracking walks within $\widetilde{\mathcal{T}}$.

Proof: We have

$$J = \begin{bmatrix} J_{\widetilde{\mathcal{F}}} & J'_M \\ J_M & J_{\widetilde{\mathcal{T}}} \end{bmatrix} \text{ and } \mathbf{h} = \begin{bmatrix} \mathbf{h}_{\widetilde{\mathcal{F}}} \\ \mathbf{h}_{\widetilde{\mathcal{T}}} \end{bmatrix}. \tag{30}$$

By Result 3) in Section II-C, when LBP converges, it gives the correct means. Hence, after convergence, for $i = 1, 2, \ldots, k$, we have

$$\mathbf{g}^i = J_{\widetilde{\mathcal{T}}}^{-1} J_{\widetilde{\mathcal{T}},i}, \text{ and } \boldsymbol{\mu}^{\widetilde{\mathcal{T}}} = J_{\widetilde{\mathcal{T}}}^{-1} \mathbf{h}_{\widetilde{\mathcal{T}}}$$

where $\mathbf{g}^i$ is the feedback gain corresponding to feedback node i and $\boldsymbol{\mu}^{\widetilde{\mathcal{T}}}$ is the partial mean in approximate FMP. These quantities are exact after convergence. Since $\mathbf{g}^i$ and $\boldsymbol{\mu}^{\widetilde{\mathcal{T}}}$ are computed exactly, following the same steps as in the proof of Theorem 1, we can obtain the exact means and variances for nodes in $\widetilde{\mathcal{F}}$.

From the proof of Theorem 1, we also have

$$\boldsymbol{\mu}_{\widetilde{\mathcal{T}}} = J_{\widetilde{\mathcal{T}}}^{-1}(\mathbf{h}_{\widetilde{\mathcal{T}}} - J_M \boldsymbol{\mu}_{\widetilde{\mathcal{F}}}). \tag{31}$$

We have shown that $\boldsymbol{\mu}_{\widetilde{\mathcal{F}}}$ is computed exactly in Step 3 in approximate FMP, so $\mathbf{h}_{\widetilde{\mathcal{T}}} - J_M \boldsymbol{\mu}_{\widetilde{\mathcal{F}}}$ is computed exactly. Since LBP on $\widetilde{\mathcal{T}}$ gives the exact means for any potential vector, the means of all nodes in $\widetilde{\mathcal{T}}$ are exact. As in the proof of Theorem 1, we have that the exact covariance matrix on $\widetilde{\mathcal{T}}$ is given by

$$P_{\widetilde{\mathcal{T}}} = J_{\widetilde{\mathcal{T}}}^{-1} + (J_{\widetilde{\mathcal{T}}}^{-1} J_M) P_{\widetilde{\mathcal{F}}} (J_{\widetilde{\mathcal{T}}}^{-1} J_M)'. \tag{32}$$

As noted previously, the exact variance of node $i \in \widetilde{\mathcal{T}}$ equals the sum of all the self-return walks of node i. We partition these walks into two classes: self-return walks of node i within $\widetilde{\mathcal{T}}$, and self-return walks that visit at least one node in $\widetilde{\mathcal{F}}$. The diagonal of $J_{\widetilde{\mathcal{T}}}^{-1}$ captures exactly the first class of walks. Hence, the second term in the right-hand side of (32) corresponds to the sum of the second class of walks. Let us compare each of these terms to what is computed by the approximate FVS algorithm. By Result 4) in Section II-C, LBP on $\widetilde{\mathcal{T}}$ gives the sums of all the backtracking walks after convergence. So the first term in (32) is approximated by backtracking walks. However, note that the terms $J_{\widetilde{\mathcal{T}}}^{-1} J_M$ and $P_{\widetilde{\mathcal{F}}}$ are obtained exactly.[10] Hence, the approximate FMP algorithm computes the second term exactly and thus provides precisely the second set of walks. As a result, the only walks missing from the exact computation of variances in $\widetilde{\mathcal{T}}$ are non-backtracking walks within $\mathcal{T}$. This completes the proof. ∎

We now state several conditions under which we can guarantee convergence.

Proposition 1: Consider a Gaussian graphical model with graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and model parameters J and $\mathbf{h}$. If the model is walk-summable, approximate FMP converges for any pseudo-FVS $\widetilde{\mathcal{F}} \subset \mathcal{V}$.

Proof: Let $R = I - J$ and $(\bar{R})_{ij} = |R_{ij}|$. In approximate FMP, LBP is performed on the pseudo-tree induced by $\widetilde{\mathcal{T}} = \mathcal{V} \backslash \widetilde{\mathcal{F}}$. The information matrix on the pseudo-tree is $J_{\widetilde{\mathcal{T}}}$, which is a submatrix of J. By [33, Corollary 8.1.20], for any $\widetilde{\mathcal{T}}$

$$\rho(\bar{R}_{\widetilde{\mathcal{T}}}) \leq \rho(\bar{R}) < 1. \tag{33}$$

By Result 3) in Section II-C, LBP on $\widetilde{\mathcal{T}}$ is guaranteed to converge. All other computations in approximate FMP terminate in a finite number of steps. Hence, approximate FMP converges for any pseudo-FVS $\widetilde{\mathcal{F}} \subset \mathcal{V}$. ∎

For the remainder of the paper we will refer to the quantities as in (33) as the spectral radii of the corresponding graphs (in this case $\widetilde{\mathcal{T}}$ and the original graph $\mathcal{G}$). Walk-summability on the entire graphical model is actually far stronger than is needed for approximate FMP to converge. As the proof of Proposition 1 suggests, all we really need is for the graphical model on the graph excluding the pseudo-FVS to be walk-summable. As we will discuss in Section IV-D, this objective provides one of the drivers for a very simple algorithm for choosing a pseudo-FVS in order to enhance the walk-summability of the remaining graph and as well as accuracy of the resulting LBP variance computations.

Remarks: The following two results follow directly from Proposition 1.

1) Consider a walk-summable Gaussian graphical model. Let $\widetilde{\mathcal{F}}_j$ be a pseudo-FVS consisting of j nodes and $\emptyset \neq \widetilde{\mathcal{F}}_1 \subseteq \widetilde{\mathcal{F}}_2 \subseteq \cdots \subseteq \widetilde{\mathcal{F}}_k \subseteq \mathcal{F}$, where $\mathcal{F}$ is an FVS, then $W_i^{\mathrm{LBP}} \subseteq W_i^{\widetilde{\mathcal{F}}_1} \subseteq W_i^{\widetilde{\mathcal{F}}_2} \subseteq \cdots \subseteq W_i^{\widetilde{\mathcal{F}}_k} \subseteq W_i^{\mathcal{F}}$ for any node i in the graph. Here W_i^{LBP} is the set of walks captured by LBP for calculating the variance of node i; $W_i^{\widetilde{\mathcal{F}}_j}$ is the set of walks captured by approximate FMP with pseudo-FVS $\widetilde{\mathcal{F}}_j$; and $W_i^{\mathcal{F}}$ is the set of walks captured by FMP with FVS $\mathcal{F}$.
2) Consider an attractive Gaussian graphical model (i.e., one in which all elements of R are non-negative). Let $\widetilde{\mathcal{F}}_1 \subseteq$

[10] Note that the columns of the former are just the feedback gains computed by LBP for each of the additional potential vectors on $\widetilde{\mathcal{T}}$ corresponding to columns of $J_{\widetilde{\mathcal{T}}\widetilde{\mathcal{F}}}$, which we have already seen are computed exactly, as we have for the covariance on the pseudo-FVS.

$\widetilde{\mathcal{F}}_2 \subseteq \cdots \subseteq \widetilde{\mathcal{F}}_k \subseteq \mathcal{F}$ denote the pseudo-FVS (FVS), and $P_{ii}^{\mathrm{LBP}}, P_{ii}^{\widetilde{\mathcal{F}}_1}, \ldots, P_{ii}^{\widetilde{\mathcal{F}}_k}, P_{ii}^{\mathcal{F}}$ denote the corresponding variances calculated for node i by LBP, approximate FMP and FMP respectively. P_{ii} represents the exact variance of node i. We have $P_{ii}^{\mathrm{LBP}} \leq P_{ii}^{\widetilde{\mathcal{F}}_1} \leq P_{ii}^{\widetilde{\mathcal{F}}_2} \leq \cdots \leq P_{ii}^{\widetilde{\mathcal{F}}_k} \leq P_{ii}^{\mathcal{F}} = P_{ii}$ for any node i in $\mathcal{V}$.

The above results show that with approximate FMP, we can effectively trade off complexity and accuracy by selecting pseudo-FVS of different sizes.

C. Error Bounds for Variance Computation

We define the measure of the error of an inference algorithm for Gaussian graphical models as the average absolute error of variances for all nodes:

$$\epsilon = \frac{1}{n}\sum_{i\in\mathcal{V}} |\widehat{P}_{ii} - P_{ii}| \tag{34}$$

where n is the number of nodes, $\widehat{P}_{ii}$ is the computed variance of node i by the algorithm and P_{ii} is the exact variance of node i.

Proposition 2: Consider a walk-summable Gaussian graphical model with n nodes. Assume the information matrix J is normalized to have unit diagonal. Let ϵ_{FMP} denote the error of approximate FMP and $\widehat{P}_{ii}^{\mathrm{FMP}}$ denote the estimated variance of node i. Then

$$\epsilon_{\mathrm{FMP}} = \frac{1}{n}\sum_{i\in\mathcal{V}} |\widehat{P}_{ii}^{\mathrm{FMP}} - P_{ii}| \leq \frac{n-k}{n}\frac{\tilde{\rho}^{\tilde{g}}}{1-\tilde{\rho}}$$

where k is the number of feedback nodes, $\tilde{\rho}$ is the spectral radius corresponding to the subgraph $\widetilde{\mathcal{T}}$, and $\tilde{g}$ denotes the girth of $\widetilde{\mathcal{T}}$, i.e., the length of the shortest cycle in $\widetilde{\mathcal{T}}$. In particular, when $k = 0$, i.e., LBP is used on the entire graph, we have

$$\epsilon_{\mathrm{LBP}} = \frac{1}{n}\sum_{i\in\mathcal{V}} |\widehat{P}_{ii}^{\mathrm{LBP}} - P_{ii}| \leq \frac{\rho^g}{1-\rho}$$

where the notation is similarly defined.

Some of the following proof techniques are motivated by the proof of the error bound on determinant estimation with the so-called orbit-product representation in [34].

Proof: By Theorem 2,

$$\epsilon_{\mathrm{LBP}} = \frac{1}{n}\sum_{i\in\mathcal{V}} |\phi(i \xrightarrow{\mathrm{NB}} i)| \tag{35}$$

where $\phi(i \xrightarrow{\mathrm{NB}} i)$ denotes the sum of all non-backtracking self-return walks of node i.

We have

$$\epsilon_{\mathrm{LBP}} = \frac{1}{n}\sum_{i\in\mathcal{V}} |\phi(i \xrightarrow{\mathrm{NB}} i)| \leq \frac{1}{n}\sum_{i\in\mathcal{V}} \bar{\phi}(i \xrightarrow{\mathrm{NB}} i) \tag{36}$$

where $\bar{\phi}(\cdot)$ denotes the sum of absolute weight of walks, or walk-sums defined on $\bar{R}$.

Non-backtracking self-return walks must contain at least one cycle. So the minimum length of a non-backtracking walk is g, which is the minimum length of cycles. Thus

$$\epsilon_{\mathrm{LBP}} \leq \frac{1}{n}\sum_{i\in\mathcal{V}} \bar{\phi}(i \xrightarrow{\mathrm{NB}} i) \leq \frac{1}{n}\sum_{i\in\mathcal{V}}\sum_{m=g}^{\infty} (\bar{R}^m)_{ii} \tag{37}$$

$$= \frac{1}{n}\mathrm{Tr}(\sum_{m=g}^{\infty}(\bar{R}^m)) = \frac{1}{n}\sum_{m=g}^{\infty}\mathrm{Tr}(\bar{R}^m). \tag{38}$$

Let $\lambda_i(\cdot)$ denotes the ith largest eigenvalue of a matrix. Since $\lambda_i(\bar{R}^m) = \lambda_i(\bar{R})^m$ and $\lambda_i(\bar{R}) \leq \rho$, we have

$$\mathrm{Tr}(\bar{R}^m) = \sum_{i=1}^{n}\lambda_i(\bar{R})^m \leq n\rho^m. \tag{39}$$

Therefore,

$$\epsilon_{\mathrm{LBP}} \leq \frac{1}{n}\sum_{m=g}^{\infty} n\rho^m = \frac{\rho^g}{1-\rho}. \tag{40}$$

When approximate FMP is used with a size-k pseudo-FVS, the variances of nodes in the pseudo-FVS are computed exactly, while the variance errors for other nodes are the same as performing LBP on the subgraph excluding the pseudo-FVS. Therefore,

$$\epsilon_{\mathrm{FMP}} = \frac{1}{n}\sum_{i\in\mathcal{V}} |\hat{P}_{ii} - P_{ii}| = \frac{1}{n}\sum_{i\in\widetilde{\mathcal{T}}} |\hat{P}_{ii} - P_{ii}| \tag{41}$$

$$= \frac{1}{n}(n-k)\epsilon_{LBP} \leq \frac{n-k}{n}\frac{\tilde{\rho}^{\tilde{g}}}{1-\tilde{\rho}}. \tag{42}$$

■

An immediate conclusion of Proposition 2 is that if a graph is cycle-free (i.e., $g = \infty$), the error ϵ_{LBP} is zero.

We can also analyze the performance of FMP on a Gaussian graphical model that is Markov on a Erdös–Rényi random graph $\mathfrak{G}\left(n, \frac{c}{n}\right)$. Each edge in such a random graph with n nodes appears with probability $\frac{c}{n}$, independent of every other edge in the graph [35].

Proposition 3: Consider a sequence of graphs $\{\mathcal{G}_n\}_{n=1}^{\infty}$ drawn from Erdös–Rényi model $\mathfrak{G}\left(n, \frac{c}{n}\right)$ with fixed c. Suppose we have a sequence of Gaussian graphical models parameterized by $\{(J_n, \mathbf{h}_n)\}_{n=1}^{\infty}$ that are Markov on $\{\mathfrak{G}_n\}_{n=1}^{\infty}$ and are strictly walk-summable (i.e., the spectral radii $\rho(\bar{R}_n)$ are uniformly upper bounded away from unity). Then asymptotically almost surely there exists a sequence of pseudo-FVS $\{\widetilde{\mathcal{F}}_n\}_{n=1}^{\infty}$ with $\widetilde{\mathcal{F}}_n$ of size $\mathcal{O}(\log n)$, with which the error of approximate FMP as in (34) approaches zero.

Proof: We can obtain a graph with girth greater than l by removing one node at every cycle of length up to l. The number of cycles of length up to l in $\mathfrak{G}\left(n, \frac{c}{n}\right)$ is $\mathcal{O}(c^l)$ asymptotically almost surely (Corollary 4.9 in [35]). So we can obtain a graph of girth $\log\log n$ by removing $\mathcal{O}(\log n)$ nodes. By Proposition 2, the error approaches zero when n approaches infinity. ■

D. Finding a Good Pseudo-FVS of Bounded Size

One goal of choosing a good pseudo-FVS is to ensure that LBP converges on the remaining subgraph; the other goal is to obtain smaller inference errors. In this subsection we discuss a local selection criterion motivated by these two goals and show that the two goals are consistent.

Let $\bar{R}$ denote the absolute edge weight matrix. Since $\rho(\bar{R}) < 1$ is a sufficient condition for LBP to converge on graph $\mathcal{G}$, obtaining convergence reduces to that of removing the minimum number of nodes such that $\rho(\bar{R}_{\widetilde{\mathcal{T}}}) < 1$ for the remaining

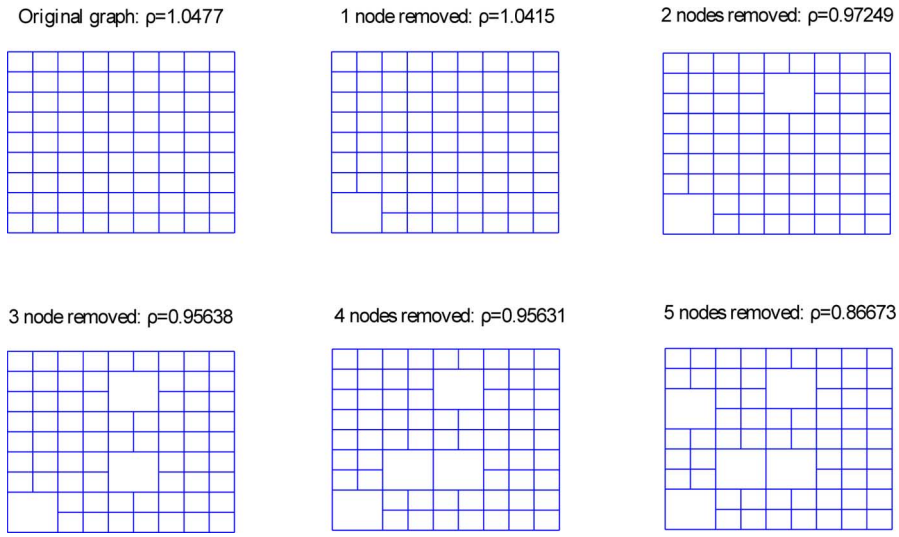


Fig. 4. Size of the pseudo-FVS and the spectral radius of the corresponding remaining graph.

graph $\widetilde{\mathcal{T}}$. However, searching and checking this condition over all possible sets of pseudo-FVS's up to a desired cardinality is a prohibitively expensive, and instead we seek a local method (i.e., using only quantities associated with individual nodes) for choosing nodes for our pseudo-FVS, one at a time, to enhance convergence. The principal motivation for our approach is the following bound [33] on the spectral radius of a nonnegative matrix, which can be evaluated efficiently:

$$\min_i \sum_j \bar{R}_{ij} \leq \rho(\bar{R}) \leq \max_i \sum_j \bar{R}_{ij}. \tag{43}$$

We further simplify this problem by a greedy heuristic: one feedback node is chosen at each iteration. This provides a basis for a simple greedy method for choosing nodes for our pseudo-FVS. In particular, at each stage, we examine the graph excluding the nodes already included in the pseudo-FVS and select the node with the largest sum of edge weights, i.e., $\operatorname{argmax}_i \sum_j \bar{R}_{ij}$. We then remove the node from the graph and put it into $\widetilde{\mathcal{F}}$. We continue the same procedure on the remaining graph until the maximum allowed size k of $\widetilde{\mathcal{F}}$ is reached or the remaining graph does not have any cycles.

Algorithm 2: The pseudo-FVS selection criterion

Input: information matrix J and the maximum size k of the pseudo-FVS

Output: a pseudo-FVS $\widetilde{\mathcal{F}}$

1. Let $\widetilde{\mathcal{F}} = \emptyset$ and normalize J to have unit diagonal.
2. Repeat until $|\widetilde{\mathcal{F}}| = k$ or the remaining graph is empty.
   a) Clean up the current graph by eliminating all the tree branches.
   b) Update the scores $s(i) = \sum_{j \in \mathcal{N}(i)} |J_{ij}|$.
   c) Put the node with the largest score into $\widetilde{\mathcal{F}}$ and remove it from the current graph.

The selection algorithm is summarized in Algorithm 2. Note that while the motivation just given for this method is to enhance convergence of LBP on $\widetilde{\mathcal{T}}$, we are also enhancing the accuracy of the resulting algorithm, as Proposition 2 suggests, since the bound on the spectral radius $\rho(\bar{R})$ is reduced with the removal of nodes. In addition, as shown in Theorem 2, the only approximation our algorithm makes is in the computation of variances for nodes in $\widetilde{\mathcal{T}}$, and those errors correspond to non-backtracking self-return walks confined to $\widetilde{\mathcal{T}}$ (i.e., we do capture non-backtracking self-return walks that exit $\widetilde{\mathcal{T}}$ and visit nodes in the pseudo-FVS). Thus, as we proceed with our selection of nodes for our pseudo-FVS, it makes sense to choose nodes with the largest scores, which is precisely what this approach accomplishes.

The complexity of the selection algorithms is $\mathcal{O}(km)$, where m is the number of edges and k is the size of the pseudo FVS. In particular, the complexity is $\mathcal{O}(kn)$ for sparse graphs. As a result, constructing a pseudo-FVS in this manner for sparse graphs such as two-dimensional grids is computationally inexpensive compared with the inference algorithm that then exploits it.

Finding a suitable pseudo-FVS is important. We will see in Section V that there is a huge performance difference between a good selection and a bad selection of $\widetilde{\mathcal{F}}$. In addition, experimental results show that with a good choice of pseudo-FVS (using the algorithms just described), we not only can get excellent convergence and accuracy results but can do this with pseudo-FVS of cardinality k and number of iterations D that scale well with the graph size n. Empirically, we find that we only need $\mathcal{O}(\log n)$ feedback nodes as well as very few iterations to obtain excellent performance, and thus the complexity is $\mathcal{O}(n \log^2(n))$.

V. Numerical Results

In this section, we apply approximate FMP to graphical models that are Markov on two-dimensional grids and present results detailing the convergence and correctness of our proposed algorithm. Two-dimensional grids are sparse since each node is connected to a maximum of four neighbors. There have been many studies of inference problems on grids [36]. However, inference cannot, in general, be solved exactly in linear time due to the existence of many cycles of various lengths. It is known that the size of the FVS for a grid grows linearly with the number of nodes on the grid [37]. Hence, we use approximate FMP with a pseudo-FVS of bounded size to

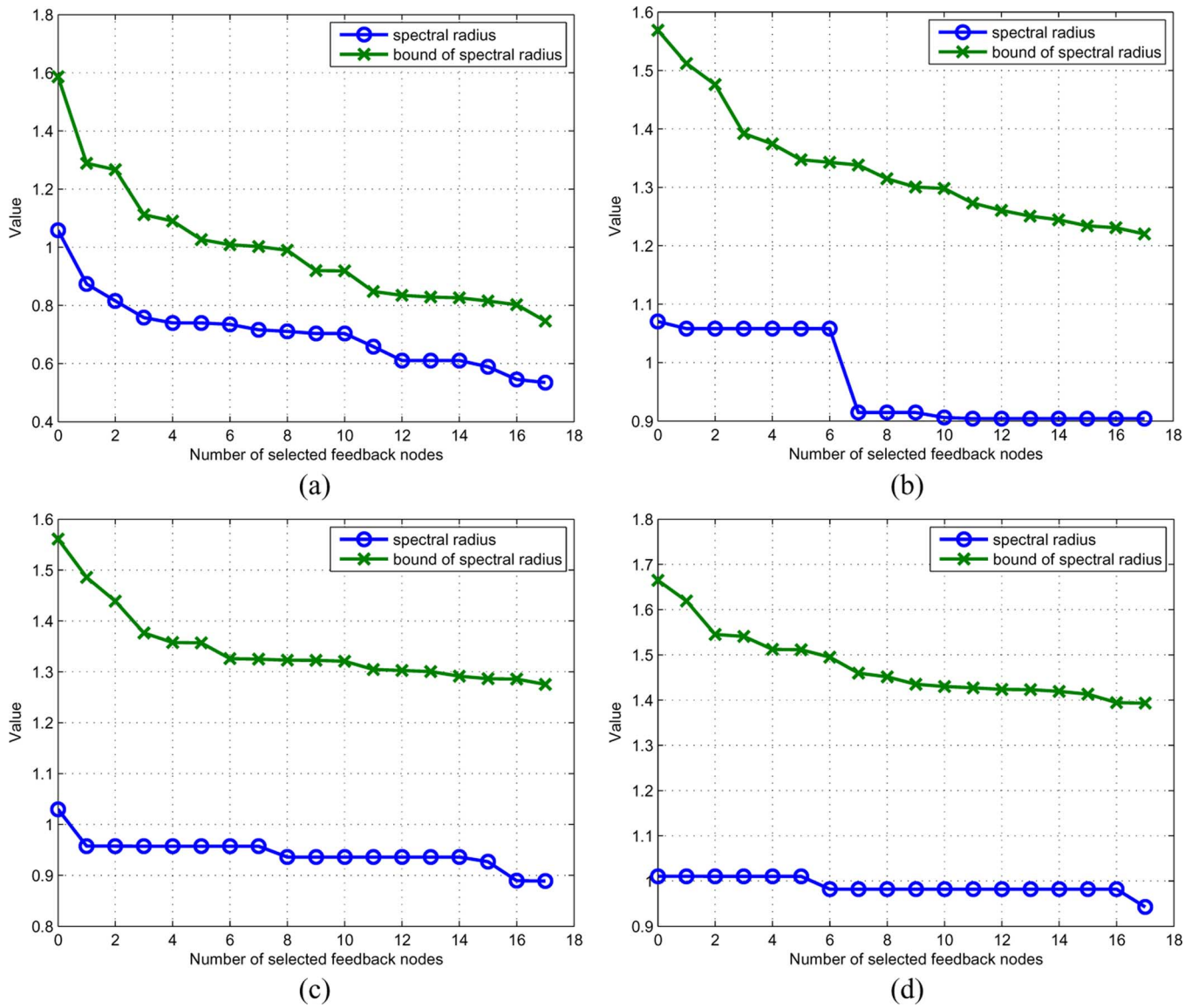


Fig. 5. Number of selected feedback nodes versus the spectral radius and its bound. (a) 10 × 10 grid. (b) 20 × 20 grid. (c) 40 × 40 grid. (d) 80 × 80 grid.

ensure that inference is tractable. Moreover, regular structures such as grids have no graphical "pinch points," i.e., nodes whose removal breaks significantly more cycles than other nodes. Hence, grids represent potentially challenging graphical structures for our approximate FMP algorithm.

In our simulations, we consider $l \times l$ grids with different values of l. The size of the graph is thus $n = l^2$. We randomly generate an information matrix J that has the sparsity pattern corresponding to a grid. Its nonzero off-diagonal entries are drawn from an i.i.d. uniform distribution with support in $[-1, 1]$. We ensure J is positive definite by adding λI. Without loss of generality, we choose λ just large enough to make the sum positive definite so that we can focus on the cases where LBP often fails and that also challenge approximate FMP, as the choice of an effective but comparatively small pseudo-FVS (of size $\mathcal{O}(\log(n))$ compared with $\mathcal{O}(n)$ nodes required for a full FVS) is crucial. We also generate a potential vector $\mathbf{h}$, whose entries are drawn i.i.d. from a uniform distribution with support in $[-1, 1]$. We then normalize the information matrix to have unit diagonal.

A. Convergence of Approximate FMP

In Fig. 4, we illustrate our pseudo-FVS selection procedure to remove one node at a time for a graphical model constructed as just-described on a 10 × 10 grid. The remaining graphs, after removing 0, 1, 2, 3, 4, and 5 nodes, and their corresponding spectral radii $\rho(\bar{R})$ are shown in the figures. LBP does not converge on the entire graph and the corresponding spectral radius is $\rho(\bar{R}) = 1.0477$. When one feedback node is chosen, the spectral radius corresponding to the remaining graph is reduced to 1.0415. After removing one more node from the graph, the spectral radius is further reduced to 0.97249, which ensures convergence. In all experiments on 10 × 10 grids, we observe that by choosing only a few nodes (at most three empirically) for our pseudo-FVS, we can obtain convergence even if LBP on the original graph diverges.

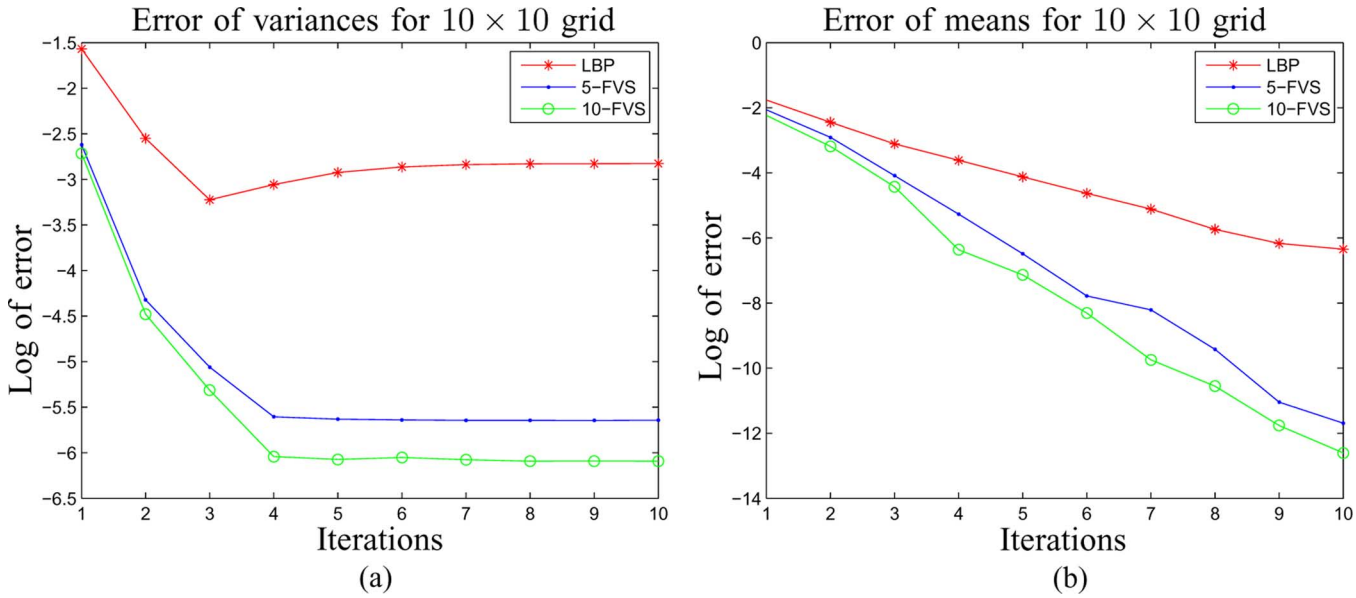


Fig. 6. Inference errors of a 10×10 grid. (a) Evolution of variance errors with iterations. (b) Evolution of mean errors with iterations.

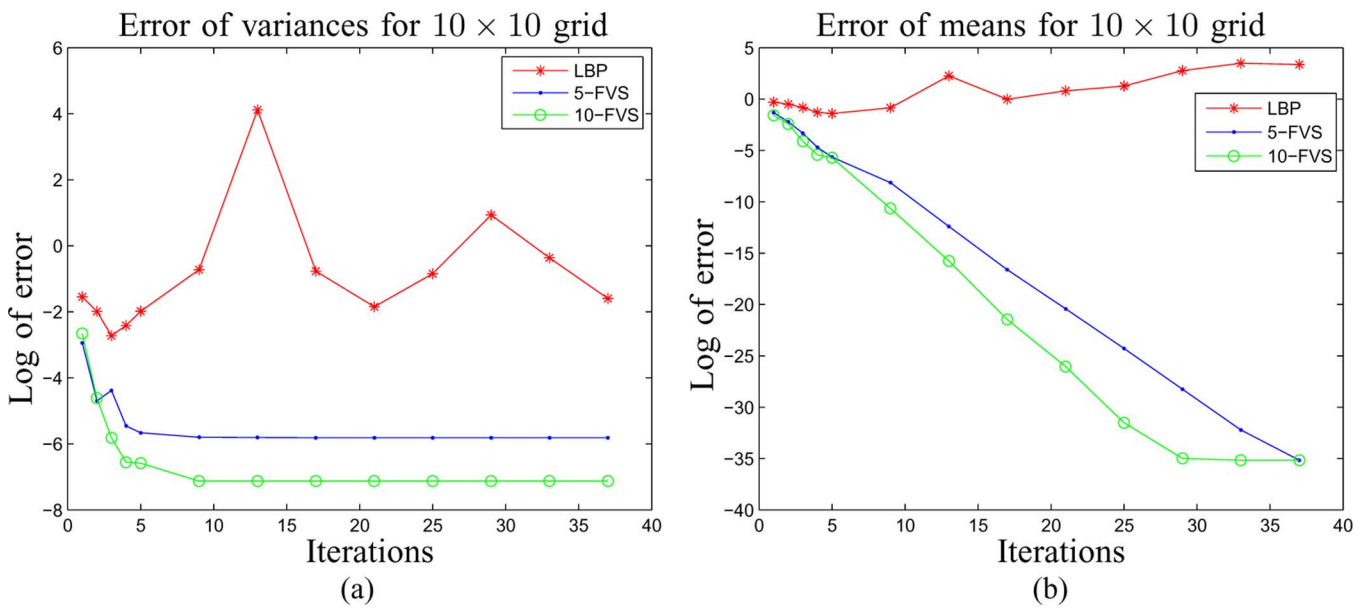


Fig. 7. Inference errors of a 10×10 grid. (a) Evolution of variance errors with iterations. (b) Evolution of mean errors with iterations.

In Fig. 5 we show that the spectral radius and its upper bound given in (43) decrease when more nodes are included in the pseudo-FVS. Convergence of approximate FMP is immediately guaranteed when the spectral radius is less than one.

B. Accuracy of Approximate FMP

In this subsection, we show numerical results of the inference errors defined in (34). On each grid, LBP and the approximate FMP algorithms with two different sets of feedback nodes are performed. One set has $k = \lceil \log n \rceil$ feedback nodes while the other has $k = \sqrt{n}$ feedback nodes. The horizontal axis shows the number of message passing iterations. The vertical axis shows the errors for both variances and means on a logarithmic scale.[11]

In Figs. 6–10, numerical results are shown for 10×10, 20×20, 40×40 and 80×80 grids respectively.[12] Except for the model in Fig. 6, LBP fails to converge for all models. With $k = \lceil \log n \rceil$ feedback nodes, approximate FMP converges for all the grids and gives much better accuracy than LBP. In Fig. 6 where LBP converges on the original graph, we obtain more accurate variances and improved convergence rates using approximate FMP. In Fig. 7 to 10, LBP diverges while approximate FMP gives inference results with small errors. When $k = \sqrt{n}$ feedback nodes are used, we obtain even better approximations but with more computations in each iteration. We performed approximate FMP on different graphs with different parameters, and empirically observed that

[11]The error of means is defined in the manner as variances—the average of the absolute errors of means for all nodes.

[12]Here we use shorthand terminology, where k-FVS refers to running our approximate FMP algorithm with a pseudo-FVS of cardinality k. Figs. 6 and 7 are different random simulation results with the same parameter setup.

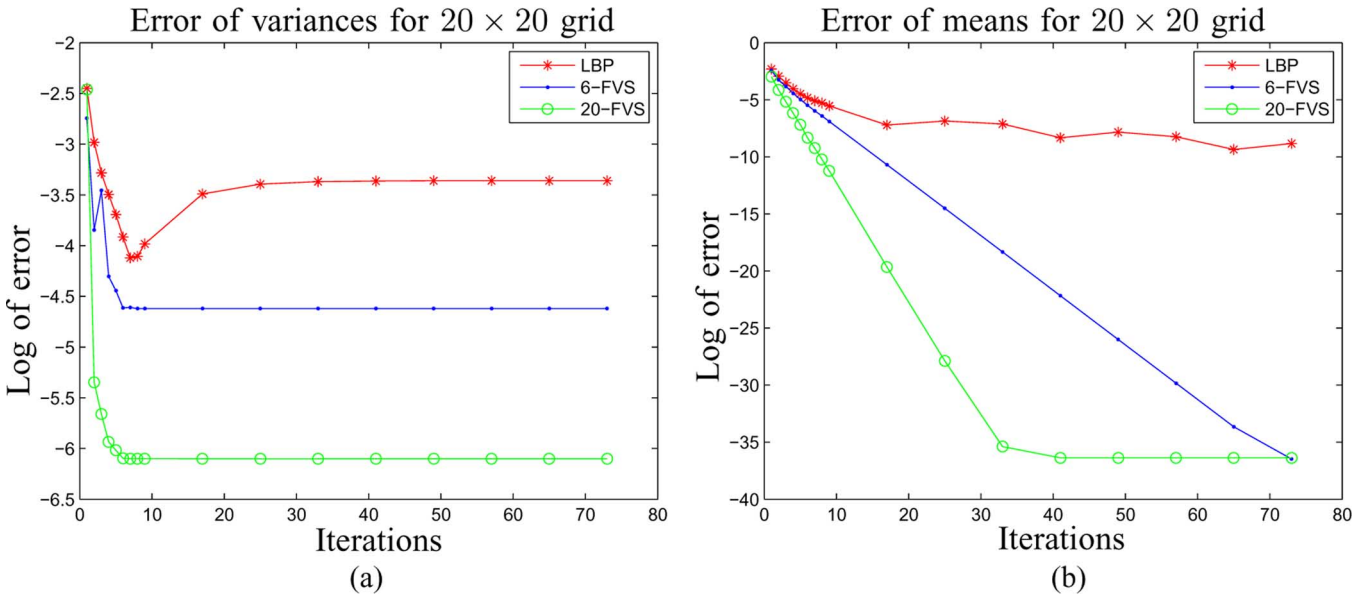


Fig. 8. Inference errors of a 20 × 20 grid. (a) Evolution of variance errors with iterations. (b) Evolution of mean errors with iterations.

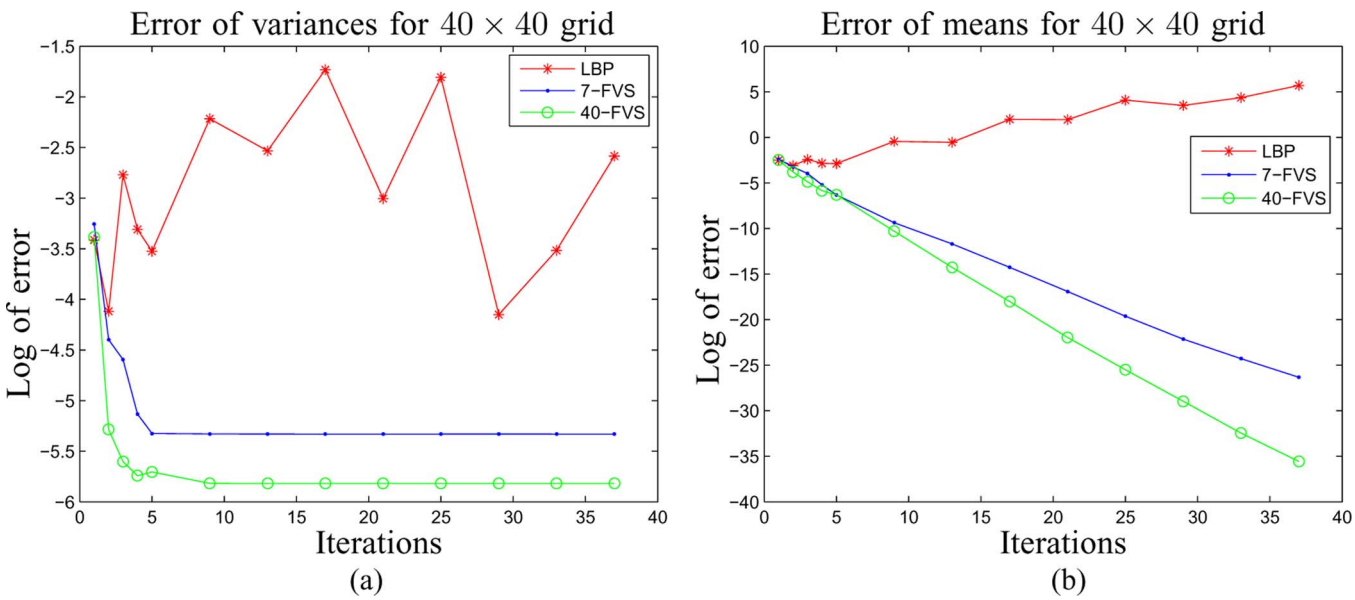


Fig. 9. Inference errors of a 40 × 40 grid. (a) Evolution of variance errors with iterations. (b) Evolution of mean errors with iterations.

$k = \lceil \log n \rceil$ feedback nodes seem to be sufficient to give a convergent algorithm and good approximations.

Remarks: The question, of course, arises as to whether it is simply the *size* of the pseudo-FVS that is important. However, numerical results show that approximate FMP does not give satisfactory results if we choose a "bad" pseudo-FVS. In Fig. 11, we present results to demonstrate that the approximate FMP algorithm with a badly selected pseudo-FVS indeed performs poorly. The pseudo-FVS is selected by the opposite criterion of the algorithm in Algorithm 2, i.e., the node with the smallest score is selected at each iteration. We can see that LBP, 7-FVS, and 40-FVS algorithms all fail to converge. These results suggest that when a suitable set of feedback nodes are selected, we can leverage the graph structure and model parameters to dramatically improve the quality of inference in Gaussian graphical models.

VI. Conclusions and Future Directions

In this paper we have developed the feedback message passing algorithm where we first identify a set of feedback nodes. The algorithm structure involves first employing BP algorithms on the remaining graph (excluding the FVS), although with several different sets of node potentials at nodes that are neighbors of the FVS; then using the results of these computations to perform exact inference on the FVS; and then employing BP on the remaining graph again in order to correct the answers on those nodes to yield exact answers. The feedback message passing algorithm solves the inference problem exactly in a Gaussian graphical model in linear time if the graph has a FVS of bounded size. Hence, for a graph with a large FVS, we propose an approximate feedback message passing algorithm that chooses a smaller "pseudo-FVS" and

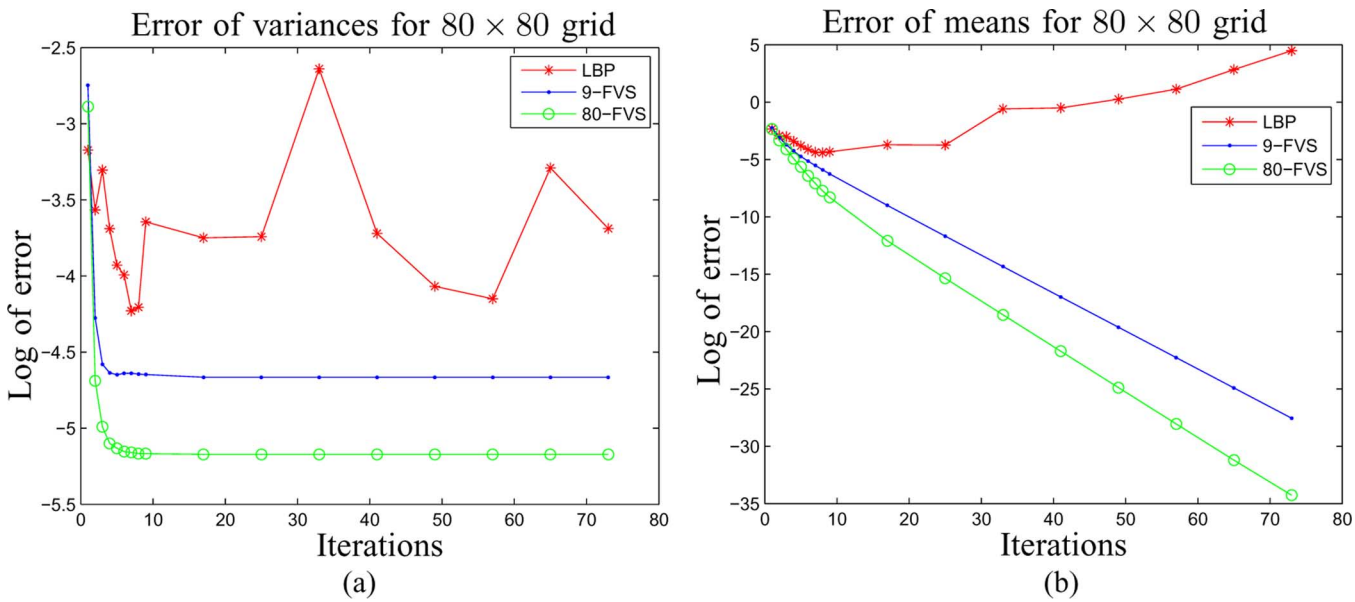


Fig. 10. Inference errors of an 80 × 80 grid. (a) Evolution of variance errors with iterations. (b) Evolution of mean errors with iterations.

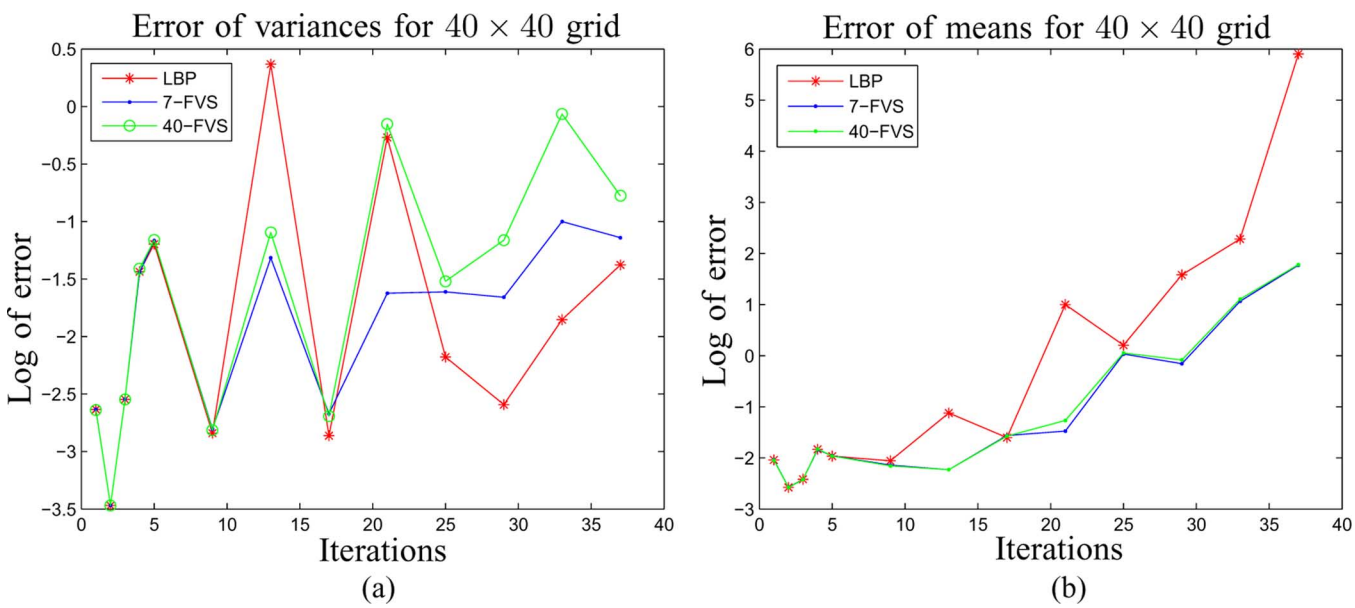


Fig. 11. Inference errors with a bad selection of feedback nodes. (a) Evolution of variance errors with iterations. (b) Evolution of means errors with iterations.

replaces BP on the remaining graph with its loopy counterpart LBP. We provide theoretical results that show that, assuming convergence of the LBP, we still obtain exact inference results (means and variances) on the pseudo-FVS, exact means on the entire graph, and approximate variances on the remaining nodes that have precise interpretations in terms of the additional "walks" that are collected as compared to LBP on the entire graph. We also provide bounds on accuracy, and these, together with an examination of the walk-summability condition, provide an algorithm for choosing nodes to include in the pseudo-FVS. Our experimental results demonstrate that these algorithms lead to excellent performance (including for models in which LBP diverges) with pseudo-FVS size that grows only logarithmically with graph size.

There are many future research directions based on the ideas of this paper. For examples, more extensive study of the performance of approximate FMP on random graphs is of great interest. Our Proposition 3 shows the existence of a pseudo-FVS of size $\mathcal{O}(\log(n))$ that works well asymptotically for Erdös–Rényi graphs. However, that Proposition does not provide construction of such an FVS. Also, an open theoretical question is whether the method for choosing a pseudo-FVS used in our experiments provides such a construction for the conditions of the Proposition.

In addition, as we have pointed out, LBP is only one possibility for the inference algorithm used on the remaining graph after a pseudo-FVS is chosen. One intriguing possibility is to indeed use approximate FMP itself on this remaining graph—i.e., a recursive applications of this algorithm. At the beginning of the recursive algorithm, only one node is selected as a feedback node. Instead of using LBP on the remaining subgraph to compute the feedback gains and correction terms (which can be reduced to the computation of certain means and variances), we can recursively apply approximate FMP on the remaining sub-

graph. Hence, more feedback nodes are selected one by one from the current remaining subgraph until the maximum allowed recursion depth is reached or we obtain a cycle-free subgraph. This recursive algorithm is currently under investigation, as are the use of these algorithmic constructs for other important problems, including the learning of graphical models with small FVS's and using an FVS or pseudo-FVS for efficient sampling of Gaussian graphical models.

ACKNOWLEDGMENT

The authors would like to thank D. Shah, J. Dauwels, V. Tan for helpful discussions, and the reviewers for their constructive comments.

REFERENCES

[1] Y. Liu, V. Chandrasekaran, A. Anandkumar, and A. Willsky, "Feedback message passing for inference in Gaussian graphical models," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Austin, TX, Jun. 2010.

[2] M. Sonka, V. Hlavac, and R. Boyle, *Image Processing, Analysis, and Machine Vision*, 2nd ed. London, U.K.: International Thomson, 1999.

[3] A. Werhli, M. Grzegorczyk, and D. Husmeier, "Comparative evaluation of reverse engineering gene regulatory networks with relevance networks, graphical Gaussian models and Bayesian networks," *Bioinformatics*, vol. 22, no. 20, pp. 2523–2523, 2006.

[4] D. Heckerman and J. Breese, "Causal independence for probability assessment and inference using Bayesian networks," *IEEE Trans. Syst., Man, Cybern., A: Syst., Humans*, vol. 26, no. 6, pp. 826–831, 1996.

[5] C. Wunsch and P. Heimbach, "Practical global oceanic state estimation," *Phys. D, Nonlinear Phenom.*, vol. 230, no. 1–2, pp. 197–208, 2007.

[6] H. E. Gamal and A. Hammons, "Analyzing the turbo decoder using the Gaussian approximation," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 671–686, 2001.

[7] T. Davis, *Direct methods for sparse linear systems*. Philadelphia, PA: SIAM, 2006, vol. 2.

[8] J. Fry and E. Gaztanaga, "Biasing and hierarchical statistics in large-scale structure," *Astrophys. J.*, vol. 413, pp. 447–447, 1993.

[9] L. Yang, X. Liu, C. Jursa, M. Holliman, A. Rader, H. Karimi, and I. Bahar, "iGNM: A database of protein functional motions based on Gaussian network model," *Bioinformatics*, vol. 21, no. 13, pp. 2978–2978, 2005.

[10] M. Jordan, "Graphical models," *Stat. Sci.*, pp. 140–155, 2004.

[11] K. Murphy, Y. Weiss, and M. Jordan, "Loopy belief propagation for approximate inference: An empirical study," *Proc. Uncertainty Art. Intell.*, pp. 467–475, 1999.

[12] C. Crick and A. Pfeffer, "Loopy belief propagation as a basis for communication in sensor networks," *Uncertainty Art. Intell.*, vol. 18, 2003.

[13] R. McEliece, D. MacKay, and J. Cheng, "Turbo decoding as an instance of Pearl's 'belief propagation' algorithm," *IEEE J. Sel. Areas Commun.*, vol. 16, no. 2, pp. 140–152, 1998.

[14] A. Ihler, J. Fisher, and A. Willsky, "Loopy belief propagation: Convergence and effects of message errors," *J. Mach. Learn. Res.*, vol. 6, no. 1, pp. 905–905, 2006.

[15] Y. Weiss and W. Freeman, "Correctness of belief propagation in Gaussian graphical models of arbitrary topology," *Neural Comput.*, vol. 13, no. 10, pp. 2173–2200, 2001.

[16] S. Tatikonda and M. Jordan, "Loopy belief propagation and Gibbs measures," *Uncertainty Art. Intell.*, vol. 18, pp. 493–500, 2002.

[17] D. Malioutov, J. Johnson, and A. Willsky, "Walk-sums and belief propagation in Gaussian graphical models," *J. Mach. Learn. Res.*, vol. 7, pp. 2031–2064, 2006.

[18] V. Chandrasekaran, J. Johnson, and A. Willsky, "Estimation in Gaussian graphical models using tractable subgraphs: A walk-sum analysis," *IEEE Trans. Signal Process.*, vol. 56, no. 5, pp. 1916–1930, 2008.

[19] J. Pearl, "A constraint propagation approach to probabilistic reasoning," *Uncertainty Art. Intel.*, 1986.

[20] A. Darwiche, "Recursive conditioning," *Art. Intell.*, vol. 126, no. 1–2, pp. 5–41, 2001.

[21] S. Lauritzen, *Graphical Models*. New York: Oxford Univ. Press, 1996.

[22] T. Speed and H. Kiiveri, "Gaussian Markov distributions over finite graphs," *Ann. Stat.*, vol. 14, no. 1, pp. 138–150, 1986.

[23] V. Vazirani, *Approximation Algorithms*. New York: Springer, 2004.

[24] R. Karp, "Reducibility among combinatorial problems," *Complexity Comput. Comput.*, vol. 43, pp. 85–103, 1972.

[25] F. Fomin, S. Gaspers, A. Pyatkin, and I. Razgon, "On the minimum feedback vertex set problem: Exact and enumeration algorithms," *Algorithmica*, vol. 52, no. 2, pp. 293–307, 2008.

[26] P. Erdös and L. Pósa, "On the maximal number of disjoint circuits of a graph," *Publicationes Mathematicae Debrecen*, vol. 9, pp. 3–12, 1962.

[27] R. Bar-Yehuda, D. Geiger, J. Naor, and R. Roth, "Approximation algorithms for the vertex feedback set problem with applications to constraint satisfaction and Bayesian inference," in *Proc. 5th Annu. ACM-SIAM Symp. Discrete Algorithms*, 1994, pp. 344–354.

[28] V. Bafna, P. Berman, and T. Fujito, "A 2-approximation algorithm for the undirected feedback vertex set problem," *SIAM J. Discrete Math.*, vol. 12, pp. 289–289, 1999.

[29] A. Shamir, "A linear time algorithm for finding minimum cutsets in reducible graphs," *SIAM J. Comput.*, vol. 8, pp. 645–645, 1979.

[30] C. Wang, E. Lloyd, and M. Soffa, "Feedback vertex sets and cyclically reducible graphs," *J. ACM*, vol. 32, no. 2, pp. 296–313, 1985.

[31] D. Kratsch, H. Müller, and I. Todinca, "Feedback vertex set on AT-free graphs," *Discrete Appl. Math.*, vol. 156, no. 10, pp. 1936–1947, 2008.

[32] F. Dehne, M. Fellows, M. Langston, F. Rosamond, and K. Stevens, "An $O(2^{O(k)}n^3)$ FPT algorithm for the undirected feedback vertex set problem," *Theory Comput. Syst.*, vol. 41, no. 3, pp. 479–492, 2007.

[33] R. Horn and C. Johnson, *Matrix Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 1990.

[34] J. Johnson, V. Chernyak, and M. Chertkov, "Orbit-product representation and correction of Gaussian belief propagation," in *Proc. 26th Annu. Int. Conf. Mach. Learn.*, 2009, pp. 473–480.

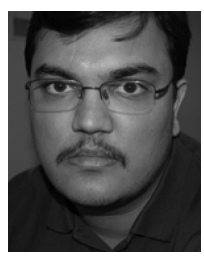
[35] B. Bollobás, *Random Graphs*. Cambridge, U.K.: Cambridge Univ. Press, 2001.

[36] A. George, "Nested dissection of a regular finite element mesh," *SIAM J. Numer. Anal.*, vol. 10, no. 2, pp. 345–363, 1973.

[37] F. Madelaine and I. Stewart, "Improved upper and lower bounds on the feedback vertex numbers of grids and butterflies," *Discrete Math.*, vol. 308, no. 18, pp. 4144–4164, 2008.

Ying Liu (S'09) received the B.E. degree in electrical engineering from Tsinghua University, Beijing, China, in 2008, and the S.M. degree in electrical engineering and computer science from the Massachusetts Institute of Technology, Cambridge, MA, in 2010, where he is currently working towards the Ph.D. degree.

Since 2008, he has been a Research Assistant with the Stochastic Systems Group under the guidance of Prof. A. Willsky. His research interests include machine learning, graphical models, stochastic signal processing, and distributed algorithms.



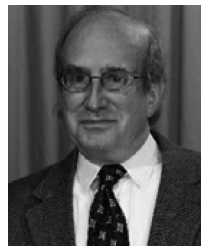
Venkat Chandrasekaran (S'03–M'11) received the B.S. degree in electrical and computer engineering and the B.A. degree in mathematics from Rice University, Houston, TX, in 2005 and the Ph.D. degree in electrical engineering and computer science from the Massachusetts Institute of Technology (MIT), Cambridge, in 2011.

He is currently an Assistant Professor in the Department of Computing and Mathematical Sciences at the California Institute of Technology, Pasadena. His research interests lie in the areas of optimization, statistics, and signal processing.

Animashree Anandkumar (S'02–M'09) received the B.Tech. degree in electrical engineering from the Indian Institute of Technology, Madras, in 2004 and the Ph.D. degree in electrical and computer engineering with a minor in applied math from Cornell University, Ithaca, NY, in 2009.

She was a Postdoctoral Researcher at the Stochastic Systems Group at the Massachusetts Institute of Technology (MIT), Cambridge, between 2009 and 2010. She is currently an Assistant Professor at the Electrical Engineering and Computer Science Department at the University of California, Irvine. Her research interests are in the area of high-dimensional statistics, statistical-signal processing, and information theory with a focus on probabilistic graphical models.

Dr. Anandkumar is the recipient of the 2008 IEEE Signal Processing Society (SPS) Young Author award for a paper that she coauthored with L. Tong which appeared in the IEEE TRANSACTIONS ON SIGNAL PROCESSING. She is the recipient of the Fran Allen IBM Ph.D. fellowship 2008–2009, presented in conjunction with the IBM Ph.D. Fellowship Award. She received the Best Thesis Award 2009 and Best Paper Award 2011 by the ACM Sigmetrics Society. She has served as a reviewer for the IEEE TRANSACTIONS ON SIGNAL PROCESSING, the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, the IEEE TRANSACTIONS ON INFORMATION THEORY, the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, and the IEEE SIGNAL PROCESSING LETTERS.



Alan S. Willsky (S'70–M'73–SM'82–F'86) received the S.B. degree in 1969 and the Ph.D. degree in 1973, both from the Department of Aeronautics and Astronautics, Massachusetts Institute of Technology (MIT).

He joined MIT, Cambridge, in 1973 and is the Edwin Sibley Webster Professor of Electrical Engineering and Director of the Laboratory for Information and Decision Systems. He was a founder of Alphatech, Inc., and Chief Scientific Consultant, a role in which he continues at BAE Systems Advanced Information Technologies. From 1998–2002, he served on the U.S. Air Force Scientific Advisory Board. He has received a number of awards, including the 1975 American Automatic Control Council Donald P. Eckman Award, the 1979 ASCE Alfred Noble Prize, the 1980 IEEE Browder J. Thompson Memorial Award, the IEEE Control Systems Society Distinguished Member Award in 1988, the 2004 IEEE Donald G. Fink Prize Paper Award, Doctorat Honoris Causa from Université de Rennes in 2005, and the 2009 Technical Achievement Award from the IEEE Signal Processing Society. His research interests are in the development and application of advanced methods of estimation, machine learning, and statistical signal and image processing.

Prof. Willsky is also a member of the National Academy of Engineering. He and his students have received a variety of Best Paper Awards at various conferences and for papers in journals, including the 2001 IEEE Conference on Computer Vision and Pattern Recognition, the 2003 Spring Meeting of the American Geophysical Union, the 2004 Neural Information Processing Symposium, Fusion 2005, and the 2008 award from the journal Signal Processing for the outstanding paper in the year 2007. He has delivered numerous keynote addresses and is coauthor of the text *Signals and Systems* (Prentice-Hall, 1996).