Embedded Trees: Estimation of Gaussian Processes on Graphs with Cycles

Erik B. Sudderth, *Student Member, IEEE*, Martin J. Wainwright, *Member, IEEE*, and Alan S. Willsky, *Fellow, IEEE*

***Abstract*—Graphical models provide a powerful general framework for encoding the structure of large-scale estimation problems. However, the graphs describing typical real-world phenomena contain many cycles, making direct estimation procedures prohibitively costly. In this paper, we develop an iterative inference algorithm for general Gaussian graphical models. It operates by exactly solving a series of modified estimation problems on spanning trees embedded within the original cyclic graph. When these subproblems are suitably chosen, the algorithm converges to the correct conditional means. Moreover, and in contrast to many other iterative methods, the tree-based procedures we propose can also be used to calculate exact error variances. Although the conditional mean iteration is effective for quite densely connected graphical models, the error variance computation is most efficient for sparser graphs. In this context, we present a modeling example suggesting that very sparsely connected graphs with cycles may provide significant advantages relative to their tree-structured counterparts, thanks both to the expressive power of these models and to the efficient inference algorithms developed herein.**

The convergence properties of the proposed tree-based iterations are characterized both analytically and experimentally. In addition, by using the basic tree-based iteration to precondition the conjugate gradient method, we develop an alternative, accelerated iteration that is finitely convergent. Simulation results are presented that demonstrate this algorithm's effectiveness on several inference problems, including a prototype distributed sensing application.

***Index Terms*—Belief propagation, error variances, Gaussian processes, graphical models, Markov random fields, multiscale, optimal estimation, tree-based preconditioners.**

I. Introduction

GAUSSIAN processes play an important role in a wide range of practical, large-scale statistical estimation problems. For example, in such fields as computer vision [3], [4] and oceanography [5], Gaussian priors are commonly used to model the statistical dependencies among hundreds of thousands of random variables. Since direct linear algebraic methods are intractable for very large problems, families of structured statistical models, and associated classes of efficient estimation algorithms, must be developed.

For problems with temporal, Markovian structure, linear state space models [6] provide a popular and effective framework for encoding statistical dependencies. Temporal Markov models correspond to perhaps the simplest class of *graphical models* [7], in which nodes index a collection of random variables (or vectors), and edges encode the structure of their statistical relationships (as elucidated in Section II). Specifically, temporal state-space models are associated with simple chain graphs (see $\mathcal{G}_{\mathrm{ss}}$ in Fig. 1). When a collection of random variables has more complex statistical structure, more complex graphs—often containing loops or cycles and many paths between pairs of nodes—are generally required.

For graphs without loops, including Markov chains and more general tree-structured graphs, very efficient optimal inference algorithms exist. For example, for Gaussian processes, the Kalman filter and Rauch–Tung–Striebel (RTS) smoother for state space models [6], and their generalizations to arbitrary trees [8], produce both optimal estimates and error variances with constant complexity per graph node. However, for large graphs with cycles, exact (noniterative) methods become prohibitively complex, leading to the study of iterative algorithms. Although one can apply standard linear algebraic methods (such as those discussed in Section II-B), there is also a considerable literature on iterative algorithms specialized for statistical inference on loopy graphs.

In this paper, we present a new class of iterative inference algorithms for arbitrarily structured Gaussian graphical models. As we illustrate, these algorithms have excellent convergence properties. Just as importantly, and in contrast to existing methods, our algorithms iteratively compute not only optimal estimates but also exact error variances. Moreover, we show that our algorithms can be combined with classical linear algebraic methods (in particular conjugate gradient) to produce accelerated, preconditioned iterations with very attractive performance.

In some contexts, such as the sensor network problem presented in Section VI-C, the structure of the graph relating a set of variables may be determined *a priori* by physical constraints. In many other situations, however, the choice of the graph is also part of the signal processing problem. That is, in many cases, the model used does not represent "truth" but rather a tradeoff between the accuracy with which the model captures important features of the phenomenon of interest and the tractability of the resulting signal processing algorithm. At one extreme are tree-structured graphs, which admit very efficient estimation algorithms but have comparatively limited modeling

Manuscript received April 28, 2003; revised September 13, 2003. This work was supported in part by the Office of Naval research under Grant N00014-00-1-0089, by the Air Force Office of Scientific Research under Grant F49620-00-1-0362, and by an ODDR&E MURI funded through the Army Research Office under Grant DAAD19-00-1-0466. E. B. Sudderth was supported in part by an NDSEG fellowship. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Hamid Krim.

E. B. Sudderth and A. S. Willsky are with the Laboratory for Information and Decision Systems, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: esuddert@mit.edu; willsky@mit.edu).

M. J. Wainwright is with the Department of Statistics and the Department of Electrical Engineering and Computer Science, University of California at Berkeley, Berkeley, CA 94720 USA (e-mail: wainwrig@eecs.berkeley.edu).

Digital Object Identifier 10.1109/TSP.2004.836539

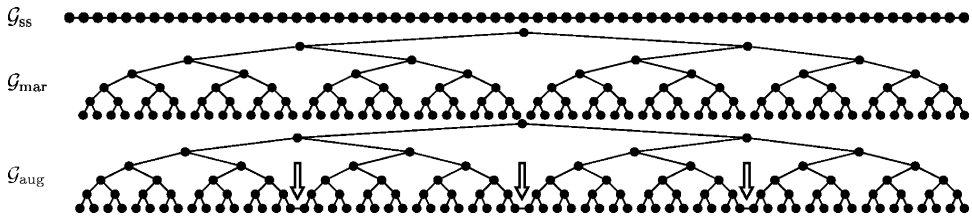

Fig. 1. Three different graphs for modeling 128 samples of a 1-D process: a Markov chain (state space model) $\mathcal{G}_{\mathrm{ss}}$, a multiscale autoregressive (MAR) model $\mathcal{G}_{\mathrm{mar}}$, and an augmented multiscale model $\mathcal{G}_{\mathrm{aug}}$, which adds three additional edges (below arrows) to the finest scale of $\mathcal{G}_{\mathrm{mar}}$. All nodes represent Gaussian vectors of dimension 2.

power. The addition of edges, and creation of loops, tends to increase modeling power, but also leads to more complex inference algorithms.

The following example explores these tradeoffs in more detail and demonstrates that for many statistical signal processing problems, it is possible to construct graphical models that effectively balance the conflicting goals of model accuracy and algorithmic tractability. Although these modeling issues are not resolved by this paper, they provide explicit motivation for the inference algorithms that we develop. In particular, as we demonstrate in Section VI-A, our methods are especially effective for this example.

A. Graphical Modeling Using Chains, Trees, and Graphs with Cycles

Consider the following one-dimensional (1-D), isotropic covariance function, which has been used to model periodic "hole effect" dependencies arising in geophysical estimation problems [9], [10]:

$$C(\tau;\omega) = \frac{1}{\omega\tau}\sin(\omega\tau), \qquad \omega > 0. \tag{1}$$

Fig. 2 shows the covariance matrix corresponding to 128 samples of a process with this covariance. In the same figure, we illustrate the approximate modeling of this process with a chain-structured graph $\mathcal{G}_{\mathrm{ss}}$. More precisely, we show the covariance of a temporal state space model with parameters chosen to provide an optimal[1] approximation to the exact covariance, subject to the constraint that each state variable has dimension 2. Notice that although the Markov chain covariance P_{ss} accurately captures short-range correlations, it is unable to model important long-range dependencies inherent in this covariance.

Multiscale autoregressive (MAR) models provide an alternative modeling framework that has been demonstrated to be powerful, efficient and widely applicable [12]. MAR models define state-space recursions on hierarchically organized trees, as illustrated by $\mathcal{G}_{\mathrm{mar}}$ in Fig. 1. Auxiliary or *hidden* variables are introduced at "coarser" scales of the tree in order to capture more accurately the statistical properties of the "finest" scale process defined on the leaf nodes.[2] Since MAR models are defined on cycle-free graphs, they admit efficient optimal estimation algorithms and, thus, are attractive alternatives for

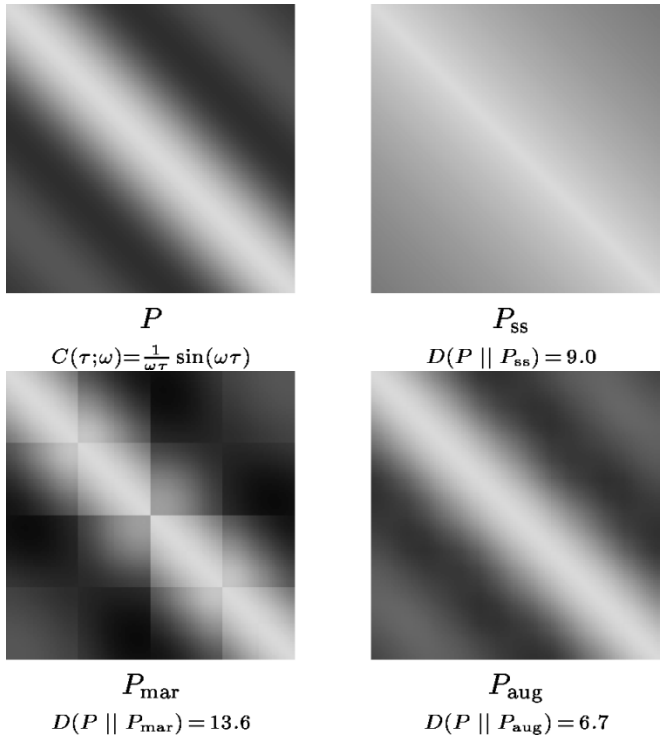

Fig. 2. Approximate modeling of the target covariance matrix P (sampled from $C(\tau;\omega)$) using each of the three graphs in Fig. 1. The KL divergence of each approximating distribution is given below an intensity plot of the corresponding covariance matrix (largest values in white).

many problems. Multiscale methods are particularly attractive for two-dimensional (2-D) processes, where quadtrees may be used to approximate notoriously difficult nearest-neighbor grids [4], [5].

Fig. 2 shows a multiscale approximation P_{mar} to the hole effect covariance, where the dimension of each node is again constrained to be 2. The MAR model captures the long-range periodic correlations much better than the state-space model. However, this example also reveals a key deficiency of MAR models: Some spatially adjacent fine-scale nodes are widely separated in the tree structure. In such cases, the correlations between these nodes may be inadequately modeled, and blocky boundary artifacts are produced. Blockiness can be reduced by increasing the dimension of the coarse scale nodes, but this often leads to an unacceptable increase in computational cost.

One potential solution to the boundary artifact problem is to add edges between pairs of fine-scale nodes where discontinuities are likely to arise. Such edges should be able to account for short-range dependencies neglected by standard MAR

[1]For all modeling examples, optimality is measured by the Kullback–Leibler (KL) divergence [11]. We denote the KL divergence between two zero mean Gaussians with covariances P and Q by $D(P\|Q)$.

[2]In some applications, coarse scale nodes provide an efficient mechanism for modeling nonlocal measurements [13], but in this example, we are only interested in the covariance matrix induced at the finest scale.

models. To illustrate this idea, we have added three edges to the tree graph $\mathcal{G}_{\mathrm{mar}}$ across the largest fine-scale boundaries, producing an "augmented" graph $\mathcal{G}_{\mathrm{aug}}$ (see Fig. 1). Fig. 2 shows the resulting excellent approximation P_{aug} to the original covariance. This augmented multiscale model retains the accurate long-range correlations of the MAR model while completely eliminating the worst boundary artifacts.

The previous example suggests that very sparsely connected graphs with cycles may offer significant modeling advantages relative to their tree-structured counterparts. Unfortunately, as the resulting graphs do have cycles, the extremely efficient inference algorithms that made tree-structured multiscale models so attractive are not available. The main goal of this paper is to develop inference techniques that allow both estimates and the associated error variances to be quickly calculated for the widest possible class of graphs. The algorithms we develop are particularly effective for graphs, like that presented in this example, which are nearly tree-structured.

B. Outline of Contributions

The primary contribution of this paper is to demonstrate that tree-based inference routines provide a natural basis for the design of estimation algorithms that apply to much broader classes of graphs. All of the algorithms depend on the fact that, within any graph with cycles, there are many embedded subgraphs for which optimal inference is tractable. Each embedded subgraph can be revealed by removing a different subset of the original graph's edges. We show that by appropriately combining sequences of exact calculations on tractable subgraphs, it is possible to solve statistical inference problems defined on the original graph with cycles exactly and efficiently. Without question, the most tractable subgraphs are trees, and for simplicity of terminology, we will refer to our methods as *embedded tree* (ET) algorithms. Similarly, we will often think of extracted subgraphs as being trees. However, all of the ideas developed in this paper carry over immediately to the more general case in which the extracted subgraph contains cycles but is still tractable, as illustrated in Section VI-C.

After presenting the necessary background regarding Gaussian graphical models and numerical linear algebra (Section II), we develop the details of our ET algorithm for iterative, exact calculation of both means (Section III) and error variances (Section IV). The performance of this algorithm is analyzed both theoretically and experimentally, demonstrating the convergence rate improvements achieved through the use of multiple embedded trees. In Section V, we then show how the basic ET algorithm may be used to precondition the conjugate gradient method, producing a much more rapidly convergent iteration. To emphasize the broad applicability of our methods, we provide an experimental evaluation on several different inference problems in Section VI. These include the augmented multiscale model of Fig. 1, as well as a prototype distributed sensing application.

II. Gaussian Graphical Models

A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ consists of a node or vertex set $\mathcal{V}$ and a corresponding edge set $\mathcal{E}$. In a graphical model [7], a random variable x_s is associated with each node $s \in \mathcal{V}$. In this paper, we focus on the case where $\{x_s | s \in \mathcal{V}\}$ is a jointly Gaussian process, and the random vector x_s at each node has dimension d (assumed uniform for notational simplicity). Given any subset $\mathcal{A} \subset \mathcal{V}$, let $x_{\mathcal{A}} \triangleq \{x_s | s \in \mathcal{A}\}$ denote the set of random variables in $\mathcal{A}$. If the $N \triangleq |\mathcal{V}|$ nodes are indexed by the integers $\mathcal{V} = \{1, 2, \ldots, N\}$, the Gaussian process defined on the overall graph is given by $x \triangleq [x_1^T x_2^T \cdots x_N^T]^T$. Let $x \sim \mathcal{N}(\mu, P)$ indicate that x is a Gaussian process with mean μ and covariance P. With graphical models, it is often useful to consider an alternative *information* parameterization $x \sim \mathcal{N}^{-1}(h, J)$, where $J = P^{-1}$ is the inverse covariance matrix, and the mean is equal to $\mu = J^{-1}h$.

Graphical models implicitly use edges to specify a set of conditional independencies. Each edge $(s, t) \in \mathcal{E}$ connects two nodes $s, t \in \mathcal{V}$, where $s \neq t$. In this paper, we exclusively employ *undirected* graphical models for which the edges (s, t) and (t, s) are equivalent.[3] Fig. 3(a) shows an example of an undirected graph representing five different random variables. Such models are also known as *Markov random fields* (MRFs) or, for the special case of jointly Gaussian random variables, as *covariance selection models* in the statistics literature [16]–[18].

In undirected graphical models, conditional independence is associated with *graph separation*. Suppose that $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$ are subsets of $\mathcal{V}$. Then, $\mathcal{B}$ separates $\mathcal{A}$ and $\mathcal{C}$ if there are no paths between sets $\mathcal{A}$ and $\mathcal{C}$ that do not pass through $\mathcal{B}$. The stochastic process x is said to be Markov with respect to $\mathcal{G}$ if $x_{\mathcal{A}}$ and $x_{\mathcal{C}}$ are independent conditioned on the random variables $x_{\mathcal{B}}$ in any separating set. For example, in Fig. 3(a), the random variables x_1 and $\{x_4, x_5\}$ are conditionally independent given $\{x_2, x_3\}$. If the *neighborhood* of a node s is defined to be $\Gamma(s) \triangleq \{t | (s, t) \in \mathcal{E}\}$, which is the set of all nodes that are directly connected to s, it follows immediately that

$$p(x_s | x_{\mathcal{V}\setminus s}) = p\left(x_s | x_{\Gamma(s)}\right). \tag{2}$$

That is, conditioned on its immediate neighbors, the probability distribution of the random vector at any given node is independent of the rest of the process.

For general graphical models, the Hammersley–Clifford theorem [16] relates the Markov properties implied by $\mathcal{G}$ to a factorization of the probability distribution $p(x)$ over *cliques*, or fully connected subsets, of $\mathcal{G}$. For Gaussian models with positive definite covariance matrices, this factorization takes a particular form that constrains the structure of the *inverse* covariance matrix. Given a Gaussian process $x \sim \mathcal{N}(\mu, P)$, we partition the inverse covariance $J \triangleq P^{-1}$ into an $N \times N$ grid of $d \times d$ submatrices $\{J_{s,t} | s, t \in \mathcal{V}\}$. We then have the following result [16], [18]:

Theorem 1: Let x be a Gaussian stochastic process with inverse covariance J, which is Markov with respect to $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Assume that $\mathcal{E}$ is minimal so that x is not Markov with respect to any $\mathcal{G}' = (\mathcal{V}, \mathcal{E}')$ such that $\mathcal{E}' \subsetneq \mathcal{E}$. Then, for any $s, t \in \mathcal{V}$ such that $s \neq t$, $J_{s,t} = J_{t,s}^T$ will be nonzero if and only if $(s, t) \in \mathcal{E}$.

[3]There is another formalism for associating Markov properties with graphs that uses directed edges. Any directed graphical model may be converted into an equivalent undirected model, although some structure may be lost in the process [14], [15].

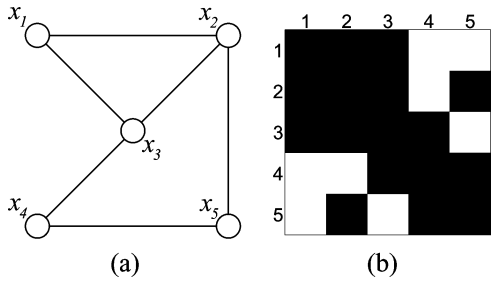

Fig. 3. (a) Graphical model representing five jointly Gaussian random vectors. (b) Structure of the corresponding inverse covariance matrix P^{-1}, where black squares denote nonzero entries.

Fig. 3 illustrates Theorem 1 for a small sample graph. In most graphical models, each node is only connected to a small subset of the other nodes. Theorem 1 then shows that P^{-1} will be a *sparse* matrix with a small (relative to N) number of nonzero entries in each row and column.

Any Gaussian distribution satisfying Theorem 1 can be written as a product of positive pairwise *potential functions* involving adjacent nodes:

$$p(x) = \frac{1}{Z} \prod_{(s,t)\in\mathcal{E}} \psi_{s,t}(x_s, x_t). \tag{3}$$

Here, Z is a normalization constant. For any inverse covariance matrix J, the pairwise potentials can be expressed in the form

$$\psi_{s,t}(x_s, x_t) = \exp\left\{-\frac{1}{2}\begin{bmatrix} x_s^T & x_t^T \end{bmatrix} \begin{bmatrix} J_{s(t)} & J_{s,t} \\ J_{t,s} & J_{t(s)} \end{bmatrix} \begin{bmatrix} x_s \\ x_t \end{bmatrix}\right\} \tag{4}$$

where the $J_{s(t)}$ terms are chosen so that for all $s \in \mathcal{V}$, $\sum_{t\in\Gamma(s)} J_{s(t)} = J_{s,s}$. Note that there are many different ways to decompose $p(x)$ into pairwise potentials, each corresponding to a different partitioning of the block diagonal entries of J. However, all of the algorithms and results presented in this paper are invariant to the specific choice of decomposition. See [2] for further discussion of this parameterization.

A. Graph-Based Inference Algorithms

Graphical models may be used to represent the prior distributions underlying Bayesian inference problems. Let $x \sim \mathcal{N}(0, P)$ be an unobserved random vector that is Markov with respect to $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. We assume that the graphical prior is parameterized by the graph-structured inverse covariance matrix $J = P^{-1}$ or equivalently by pairwise potential functions as in (4). Given a vector of noisy observations $y = Cx + v$, $v \sim \mathcal{N}(0, R)$, the conditional distribution $p(x|y) \sim \mathcal{N}(\widehat{x}, \widehat{P})$ may be calculated from the information form of the normal equations:

$$\widehat{P}^{-1}\widehat{x} = C^T R^{-1} y \tag{5}$$

$$\widehat{P} = (J + C^T R^{-1} C)^{-1}. \tag{6}$$

The conditional mean $\widehat{x}$ provides both the *Bayes' least squares* and *maximum a posteriori* (MAP) estimates of x. The *error covariance* matrix $\widehat{P}$ measures the expected deviation of x from the estimate $\widehat{x}$.

We assume, without loss of generality,[4] that the observation vector y decomposes into a set $\{y_s\}_{s=1}^N$ of local observations of the individual variables $\{x_s\}_{s=1}^N$. In this case, C and R are block diagonal matrices. We would like to compute the *marginal* distributions $p(x_s|y) \sim \mathcal{N}(\widehat{x}_s, \widehat{P}_s)$ for all $s \in \mathcal{V}$. Note that each $\widehat{x}_s$ is a subvector of $\widehat{x}$, whereas each $\widehat{P}_s$ is a block diagonal element of $\widehat{P}$. These marginal distributions could be directly calculated from (5) and (6) by matrix inversion in $\mathcal{O}((Nd)^3)$ operations. For large problems, however, this cost is prohibitively high, and the structure provided by $\mathcal{G}$ must be exploited.

When $\mathcal{G}$ is a Markov chain, efficient dynamic programming-based recursions may be used to exactly compute $p(x_s|y)$ in $\mathcal{O}(Nd^3)$ operations. For example, when the potentials are specified by a state-space model, a standard Kalman filter may be combined with a complementary reverse-time recursion [6]. These algorithms may be directly generalized to any graph that contains no cycles [8], [14], [19], [20]. We refer to such graphs as *tree-structured*. Tree-based inference algorithms use a series of local message-passing operations to exchange statistical information between neighboring nodes. One of the most popular such methods is known as the sum-product [19] or *belief propagation* (BP) [14] algorithm. The *junction tree* algorithm [7], [16] extends tree-based inference procedures to general graphs by first clustering nodes to break cycles and then running the BP algorithm on the tree of clusters. However, in order to ensure that the junction tree is probabilistically consistent, the dimension of the clustered nodes must often be quite large [7]. In these cases, the computational cost is generally comparable to direct matrix inversion.

The intractability of exact inference methods has motivated the development of alternative iterative algorithms. One of the most popular is known as *loopy belief propagation* [21]. Loopy BP iterates the local message-passing updates underlying tree-based inference algorithms until they (hopefully) converge to a fixed point. For many graphs, especially those arising in error-correcting codes [19], these fixed points very closely approximate the true marginal distributions [22]. For Gaussian graphical models, it has been shown that when loopy BP does converge, it always calculates the correct conditional means [21], [23]. However, the error variances are incorrect because the algorithm fails to account properly for the correlations induced by the graph's cycles. For more general graphical models, recently developed connections to the statistical physics literature have led to a deeper understanding of the approximations underlying loopy BP [15], [24], [25].

Recently, two independent extensions of the loopy BP algorithm have been proposed that allow exact computation of error variances, albeit with greater computational cost. Welling and Teh [26] have proposed propagation rules for computing linear response estimates of the joint probability of all pairs of nodes. For Gaussian models, their method iteratively computes the full error covariance matrix at a cost of $\mathcal{O}(N|\mathcal{E}|)$ operations per iteration. However, for connected graphs, $|\mathcal{E}|$ is at least $\mathcal{O}(N)$, and the resulting $\mathcal{O}(N^2)$ cost is undesirable when only the N marginal variances are needed. Plarre and Kumar [27]

[4]Any observation involving multiple nodes may be represented by a set of pairwise potentials coupling those nodes and handled identically to potentials arising from the prior.

have proposed a different extended message passing algorithm that exploits the correspondence between recursive inference and Gaussian elimination [12]. We discuss their method in more detail in Section V.

B. Linear Algebraic Inference Algorithms

As shown by (5), the conditional mean $\widehat{x}$ of a Gaussian inference problem can be viewed as the solution of a linear system of equations. Thus, any algorithm for solving sparse, positive definite linear systems may be used to calculate such estimates. Letting $\widehat{J} \triangleq \widehat{P}^{-1}$ denote the inverse error covariance matrix and $\bar{y} \triangleq C^T R^{-1} y$ the normalized observation vector, (5) may be rewritten as

$$\widehat{J}\widehat{x} = \bar{y}. \tag{7}$$

A wide range of iterative algorithms for solving linear systems may be derived using a *matrix splitting* $\widehat{J} = M - K$. The unique solution $\widehat{x}$ of (7) is also the only solution of

$$(\widehat{J} + K)\widehat{x} = K\widehat{x} + \bar{y}. \tag{8}$$

Assuming $M = \widehat{J} + K$ is invertible, (8) naturally suggests the generation of a sequence of iterates $\{\widehat{x}^n\}_{n=1}^{\infty}$ according to the recursion

$$\widehat{x}^n = M^{-1}(K\widehat{x}^{n-1} + \bar{y}). \tag{9}$$

The matrix M is known as a *preconditioner*, and (9) is an example of a preconditioned Richardson iteration [28], [29]. Many classic algorithms, including the Gauss–Jacobi and successive overrelaxation methods, are Richardson iterations generated by specific matrix splittings.

The convergence of the Richardson iteration (9) is determined by the eigenvalues $\{\lambda_i(M^{-1}K)\}$ of the matrix $M^{-1}K$. Letting $\rho(M^{-1}K) \triangleq \max_{\lambda \in \{\lambda_i(M^{-1}K)\}} |\lambda|$ denote the spectral radius, $\widehat{x}^n$ will converge to $\widehat{x}$, for arbitrary $\widehat{x}^0$, if and only if $\rho(M^{-1}K) < 1$. The asymptotic convergence rate is

$$\rho(M^{-1}K) = \rho(I - M^{-1}\widehat{J}) \tag{10}$$

If M is chosen so that $M^{-1}\widehat{J} \approx I$, the Richardson iteration will converge after a small number of iterations. However, at each iteration, it is necessary to multiply $K\widehat{x}^{n-1}$ by M^{-1} or, equivalently, to solve a linear system of the form $M\bar{x} = \bar{b}$. The challenge, then, is to determine a preconditioner M that well approximates the original system $\widehat{J}$ but whose solution is much simpler.

Although Richardson iterations are often quite effective, more sophisticated algorithms have been proposed. For positive definite systems, the *conjugate gradient* (CG) iteration is typically the method of choice [28], [30]. Each iteration of the CG algorithm chooses $\widehat{x}^n$ to minimize the weighted error metric $\|\widehat{J}\widehat{x}^n - \bar{y}\|_{\widehat{J}^{-1}}$ over subspaces of increasing dimension. This minimization can be performed in $\mathcal{O}(Nd^2)$ operations per iteration, requiring only a few matrix-vector products involving the matrix $\widehat{J}$ and some inner products [28]. CG is guaranteed to converge (with exact arithmetic) in at most Nd iterations. However, as with Richardson iterations, any symmetric preconditioning matrix may be used to modify the spectral properties of $\widehat{J}$, thereby accelerating convergence.

The standard CG algorithm, like Richardson iterations, does not explicitly calculate any entries of $\widehat{J}^{-1} = \widehat{P}$ and, thus, does not directly provide error variance information. In principle, it is possible to extend CG to iteratively compute error variances along with estimates [31]. However, these error variance formulas are very sensitive to finite precision effects, and for large or poorly conditioned problems, they typically produce highly inaccurate results (see the simulations in Section VI).

With any iterative method, it is necessary to determine when to stop the iteration, i.e., to decide that $\widehat{x}^n$ is a sufficiently close approximation to $\widehat{J}^{-1}\bar{y}$. For all of the simulations in this paper, we follow the standard practice [30] of iterating until the residual $r^n = \bar{y} - \widehat{J}\widehat{x}^n$ satisfies

$$\frac{\|r^n\|_2}{\|\bar{y}\|_2} \leq \epsilon \tag{11}$$

where ϵ is a tolerance parameter. The final error is then upper bounded as $\|\widehat{J}^{-1}\bar{y} - \widehat{x}^n\|_2 \leq \epsilon \lambda_{\min}(\widehat{J})\|\bar{y}\|_2$.

III. Calculating Conditional Means Using Embedded Trees

In this section, we develop the class of ET algorithms for finding the conditional mean of Gaussian inference problems defined on graphs with cycles. Complementary ET algorithms for the calculation of error variances are discussed in Section IV. The method we introduce explicitly exploits graphical structure inherent in the problem to form a series of tractable approximations to the full model. For a given graphical model, we do not define a single iteration but a family of nonstationary generalizations of the Richardson iteration introduced in Section II-B. Our theoretical and empirical results establish that this nonstationarity can substantially improve the ET algorithm's performance.

A. Graph Structure and Embedded Trees

As discussed in Section II-A, inference problems defined on tree-structured graphs may be efficiently solved by direct, recursive algorithms. Each iteration of the ET algorithm exploits this fact to perform exact computations on a tree embedded in the original graph. For a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, an embedded tree $\mathcal{G}_{\mathcal{T}} = (\mathcal{V}, \mathcal{E}_{\mathcal{T}})$ is defined to be a subgraph ($\mathcal{E}_{\mathcal{T}} \subset \mathcal{E}$) that has no cycles. We use the term tree to include both spanning trees in which $\mathcal{G}_{\mathcal{T}}$ is connected, as well as disconnected "forests" of trees. As Fig. 4 illustrates, there are typically a large number of trees embedded within graphs with cycles. More generally, the ET algorithm can exploit any embedded subgraph for which exact inference is tractable. We provide an example of this generality in Section VI-C. For clarity, however, we frame our development in the context of tree-structured subgraphs.

For Gaussian graphical models, embedded trees are closely connected to the structural properties of the inverse covariance matrix. Consider a Gaussian process $x \sim \mathcal{N}^{-1}(0, J)$ that is Markov with respect to an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. By Theorem 1, for any $s, t \in \mathcal{V}$ such that $s \neq t$, $J_{s,t}$ will be nonzero if and only if $(s, t) \in \mathcal{E}$. Thus, modifications of the edge set $\mathcal{E}$ are precisely equivalent to changes in the locations of the nonzero off-diagonal entries of J. In particular, consider a modified stochastic process $x_{\mathcal{T}} \sim \mathcal{N}^{-1}(0, J_{\mathcal{T}})$ that is Markov with respect

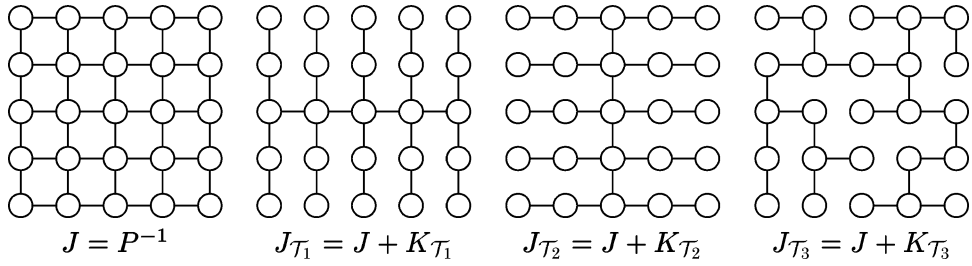

Fig. 4. Embedded trees produced by three different cutting matrices $\{K_{\mathcal{T}_i}\}_{i=1}^{3}$ for a nearest-neighbor grid.

to an embedded tree $\mathcal{G}_{\mathcal{T}} = (\mathcal{V}, \mathcal{E}_{\mathcal{T}})$. For any tree-structured inverse covariance $J_{\mathcal{T}}$, there exists a symmetric matrix $K_{\mathcal{T}}$ such that

$$J_{\mathcal{T}} = J + K_{\mathcal{T}}. \tag{12}$$

Because it acts to remove edges from the graph, $K_{\mathcal{T}}$ is called a *cutting matrix*. As Fig. 4 illustrates, different cutting matrices may produce different trees embedded within the original graph. Note that the cutting matrix $K_{\mathcal{T}}$ also defines a matrix splitting $J = (J_{\mathcal{T}} - K_{\mathcal{T}})$ as introduced in Section II-B.

Certain elements of the cutting matrix, such as the off-diagonal blocks corresponding to discarded edges, are uniquely defined by the choice of embedded tree $\mathcal{G}_{\mathcal{T}}$. However, other entries of $K_{\mathcal{T}}$, such as the diagonal elements, are not constrained by graph structure. Consequently, there exist many different cutting matrices $K_{\mathcal{T}}$ and associated inverse covariances $J_{\mathcal{T}}$, corresponding to a given tree $\mathcal{G}_{\mathcal{T}}$. In later sections, it will be useful to define a restricted class of *regular* cutting matrices;

Definition 1: For a regular cutting matrix $K_{\mathcal{T}}$ corresponding to an embedded tree $\mathcal{G}_{\mathcal{T}}$, all off-diagonal entries not corresponding to cut edges must be zero. In addition, the block diagonal entries for nodes from which no edge is cut must be zero.

Thus, for a given $\mathcal{G}_{\mathcal{T}}$, elements of the corresponding family of regular cutting matrices may differ only in the block diagonal entries corresponding to nodes involved in at least one cut edge.

As discussed in Section I, many potentially interesting classes of graphical models are "nearly" tree-structured. For such models, it is possible to reveal an embedded tree by removing a small (relative to N) number of edges. Let $E \triangleq |\mathcal{E} \setminus \mathcal{E}_{\mathcal{T}}|$ denote the number of discarded or "cut" edges. Clearly, any regular cutting matrix $K_{\mathcal{T}}$ removing E edges may have nonzero entries in at most $2Ed$ columns, implying that $\text{rank}(K_{\mathcal{T}})$ is at most $\mathcal{O}(Ed)$. Thus, for sparsely connected graphical models where $E \ll N$, cutting matrices may always be chosen to have *low rank*. This fact is exploited in later sections of this paper.

B. Tree-Based Stationary and Nonstationary Richardson Iterations

Consider the graphical inference problem introduced in Section II-A. As before, let $\widehat{J} \triangleq \widehat{P}^{-1}$ denote the inverse error covariance matrix and $\bar{y} \triangleq C^T R^{-1} y$ the normalized observation vector. As discussed in the previous section, any embedded tree $\mathcal{G}_{\mathcal{T}}$, and associated cutting matrix $K_{\mathcal{T}}$, defines a matrix splitting $\widehat{J} = (\widehat{J}_{\mathcal{T}} - K_{\mathcal{T}})$. The standard Richardson iteration (9) for this splitting is given by

$$\widehat{x}^n = \widehat{J}_{\mathcal{T}}^{-1}(K_{\mathcal{T}}\widehat{x}^{n-1} + \bar{y}). \tag{13}$$

By comparison to (7), we see that, assuming $\widehat{J}_{\mathcal{T}}$ is positive definite, this iteration corresponds to a tree-structured Gaussian inference problem, with a set of perturbed observations given by $(K_{\mathcal{T}}\widehat{x}^{n-1} + \bar{y})$. Thus, each iteration can be efficiently computed. More generally, it is sufficient to choose $K_{\mathcal{T}}$ so that $\widehat{J}_{\mathcal{T}}$ is invertible. Although the probabilistic interpretation of (13) is less clear in this case, standard tree-based inference recursions will still correctly solve this linear system. In Section III-D, we discuss conditions that guarantee invertibility of $\widehat{J}_{\mathcal{T}}$.

Because there are many embedded trees within any graph cycles, there is no reason that the iteration of (13) must use the same matrix splitting at every iteration. Let $\{\mathcal{G}_{\mathcal{T}_n}\}_{n=1}^{\infty}$ be a sequence of trees embedded within $\mathcal{G}$, and let $\{K_{\mathcal{T}_n}\}_{n=1}^{\infty}$ be a corresponding sequence of cutting matrices such that $\widehat{J}_{\mathcal{T}_n} = (\widehat{J} + K_{\mathcal{T}_n})$ is Markov with respect to $\mathcal{G}_{\mathcal{T}_n}$. Then, from some initial guess $\widehat{x}^0$, we may generate a sequence of iterates $\{\widehat{x}^n\}_{n=1}^{\infty}$ using the recursion

$$\widehat{x}^n = \widehat{J}_{\mathcal{T}_n}^{-1}(K_{\mathcal{T}_n}\widehat{x}^{n-1} + \bar{y}). \tag{14}$$

We refer to this nonstationary generalization of the standard Richardson iteration as the ET algorithm [1], [2]. The cost of computing $\widehat{x}^n$ from $\widehat{x}^{n-1}$ is $\mathcal{O}(Nd^3 + Ed^2)$, where $E = |\mathcal{E} \setminus \mathcal{E}_{\mathcal{T}_n}|$ is the number of cut edges. Typically, E is at most $\mathcal{O}(N)$; therefore, the overall cost of each iteration is $\mathcal{O}(Nd^3)$.

Consider the evolution of the error $e^n \triangleq (\widehat{x}^n - \widehat{x})$ between the estimate $\widehat{x}^n$ at the n^{th} iteration and the solution $\widehat{x}$ of the original inference problem. Combining (7) and (14), we have

$$e^n = \widehat{J}_{\mathcal{T}_n}^{-1} K_{\mathcal{T}_n} e^{n-1}. \tag{15}$$

From the invertibility of $\widehat{J}$ and $\widehat{J}_{\mathcal{T}_n}$, it follows immediately that $\widehat{x}$, which is the conditional mean of the original inference problem (7), is the unique fixed point of the ET recursion. One natural implementation of the ET algorithm cycles through a fixed set of T embedded trees $\{\mathcal{G}_{\mathcal{T}_n}\}_{n=1}^{T}$ in a periodic order so that

$$\mathcal{G}_{\mathcal{T}_{n+kT}} = \mathcal{G}_{\mathcal{T}_n} \qquad k \in \mathbb{Z}^+. \tag{16}$$

In this case, e^n evolves according to a linear periodically varying system whose convergence can by analyzed as follows:

Proposition 1: Suppose the ET mean recursion (14) is implemented by periodically cycling through T embedded trees, as in (16). Then, the error $e^n \triangleq \widehat{x}^n - \widehat{x}$ evolves according to

$$e^{Tn+T} = \left[\prod_{j=1}^{T} \widehat{J}_{\mathcal{T}_j}^{-1} K_{\mathcal{T}_j}\right] e^{Tn} \triangleq S e^{Tn}. \tag{17}$$

If $\rho(S) < 1$, then for arbitrary $\widehat{x}^0$, $e^n \overset{n\to\infty}{\longrightarrow} 0$ at an asymptotic rate of at most $\gamma \triangleq \rho(S)^{(1/T)}$.

Thus, the convergence rate of the ET algorithm may be optimized by choosing the cutting matrices $K_{\mathcal{T}_n}$ such that $\rho(S)$ is as small as possible.

As discussed earlier, when the ET iteration uses the same cutting matrix $K_{\mathcal{T}}$ at every iteration [as in (13)], it is equivalent to a stationary preconditioned Richardson iteration. The following proposition shows that when the recursion is implemented by periodically cycling through $T > 1$ cutting matrices, we may still recover a stationary Richardson iteration by considering every T^{th} iterate.

Proposition 2: Suppose that the ET recursion (14) is implemented by periodically cycling through T cutting matrices $\{K_{\mathcal{T}_n}\}_{n=1}^{T}$. Consider the subsampled sequence of estimates $\{\widehat{x}^{nT}\}_{n=0}^{\infty}$ produced at every Tth iteration. The ET procedure generating these iterates is equivalent to a preconditioned Richardson iteration

$$\widehat{x}^{Tn} = \left(I - M_T^{-1}\widehat{J}\right)\widehat{x}^{T(n-1)} + M_T^{-1} C^T R^{-1} y \tag{18}$$

where the preconditioner M_T^{-1} is defined according to the recursion

$$M_n^{-1} = \left(\widehat{J} + K_{\mathcal{T}_n}\right)^{-1} K_{\mathcal{T}_n} M_{n-1}^{-1} + \left(\widehat{J} + K_{\mathcal{T}_n}\right)^{-1} \tag{19}$$

with initial condition $M_1^{-1} = (\widehat{J} + K_{\mathcal{T}_1})^{-1}$.

Proof: This result follows from an induction argument; see [2, Th. 3.3] for details. □

Several classic Richardson iterations can be seen as special cases of the ET algorithm. For example, if the cutting matrix removes every edge, producing a disconnected forest of single-node trees, the result is the well known Gauss–Jacobi algorithm [28], [29]. For nearest-neighbor grids (as in Fig. 4), one possible ET iteration alternates between two cutting matrices, the first removing all vertical edges and the second all horizontal ones. It can be shown that this iteration is equivalent to the alternating direction implicit (ADI) method [29], [30], [32]. For more details on these connections, see [2, Sec. 3.2.5].

There are also heuristic similarities between the ET iteration, which uses the BP algorithm to exactly solve a sequence of tree-structured subproblems, and loopy BP. This relationship is particularly apparent when loopy BP's messages are updated according to tree-based schedules [25]. However, it is straightforward to show that the cutting matrix used by ET to connect subsequent iterations [see (14)] is *not* equivalent to any set of BP message updates. The precise relationship between these methods is complex and depends on the numerical structure of the chosen cutting matrix. However, as we demonstrate in Section III-D, the ET cutting matrix may always be chosen to guarantee convergence, whereas there exist graphs for which loopy BP is known to diverge. In addition, the ET algorithm may be extended to compute exact error variances (see Section IV), whereas loopy BP's variance estimates are approximate.

C. Motivating Example: Single-Cycle Inference

In this section, we explore the ET algorithm's behavior on a simple graph in order to motivate later theoretical results; more extensive simulations are presented in Section VI. Our results provide a dramatic demonstration of the importance of allowing for nonstationary Richardson iterations.

Consider a 20-node single-cycle graph, with randomly chosen inhomogeneous potentials. Although, in practice, single cycle problems can be solved easily by direct methods [33], they provide a useful starting point for understanding iterative algorithms. The cutting matrix $K_{\mathcal{T}}$ must only remove a single edge to reveal a spanning tree of a single cycle graph. In this section, we consider only regular cutting matrices, for which all off-diagonal entries are zero except for the pair required to remove the chosen edge. However, we may freely choose the two diagonal entries $(K_{\mathcal{T}})_{s,s}$, $(K_{\mathcal{T}})_{t,t}$ corresponding to the nodes from which the single edge (s,t) is cut. We consider three different possibilities, corresponding to positive semidefinite ($(K_{\mathcal{T}})_{s,s} = (K_{\mathcal{T}})_{t,t} = |(K_{\mathcal{T}})_{s,t}|$), zero diagonal ($(K_{\mathcal{T}})_{s,s} = (K_{\mathcal{T}})_{t,t} = 0$), and negative semidefinite ($(K_{\mathcal{T}})_{s,s} = (K_{\mathcal{T}})_{t,t} = -|(K_{\mathcal{T}})_{s,t}|$) cutting matrices.

When the ET iteration is implemented with a single cutting matrix $K_{\mathcal{T}}$, Proposition 1 shows that the convergence rate is given by $\gamma = \rho(\widehat{J}_{\mathcal{T}}^{-1} K_{\mathcal{T}})$. Fig. 5(a) plots γ as a function of the magnitude $|\widehat{J}_{s,t}|$ of the off-diagonal error covariance entry corresponding to the cut edge. Intuitively, convergence is fastest when the magnitude of the cut edge is small. For this example, zero diagonal cutting matrices always lead to the fastest convergence rates.

When the ET algorithm is implemented by periodically cycling between two cutting matrices $K_{\mathcal{T}_1}$, $K_{\mathcal{T}_2}$, Proposition 1 shows that the convergence rate is given by $\gamma = \rho(\widehat{J}_{\mathcal{T}_2}^{-1} K_{\mathcal{T}_2} \widehat{J}_{\mathcal{T}_1}^{-1} K_{\mathcal{T}_1})^{1/2}$. Figs. 5(b) and 5(c) plot these convergence rates when $K_{\mathcal{T}_1}$ is chosen to cut the cycle's weakest edge, and $K_{\mathcal{T}_2}$ is varied over all other edges. When plotted against the magnitude of the second edge cut, as in Fig. 5(b), the γ values display little structure. Fig. 5(c) shows these same γ values plotted against an index number showing the ordering of edges in the cycle. Edge 7 is the weak edge removed by $K_{\mathcal{T}_1}$. Notice that for the zero diagonal case, cutting the same edge at every iteration is the *worst* possible choice, despite the fact that every other edge in the graph is stronger and leads to slower single-tree iterations. The best performance is obtained by choosing the second cut edge to be as far from the first edge as possible.

In Fig. 5(d) and (e), we examine the convergence behavior of the zero diagonal two-tree iteration corresponding to edges 7 and 19 in more detail. For Fig. 5(d) and (e), the error at each iteration is measured using the normalized residual introduced in (11). Fig. 5(d) shows that even though the single-tree iteration generated by edge 19 converges rather slowly relative to the edge 7 iteration, the composite iteration is orders of magnitude faster than either single-tree iteration. In Fig. 5(e), we compare

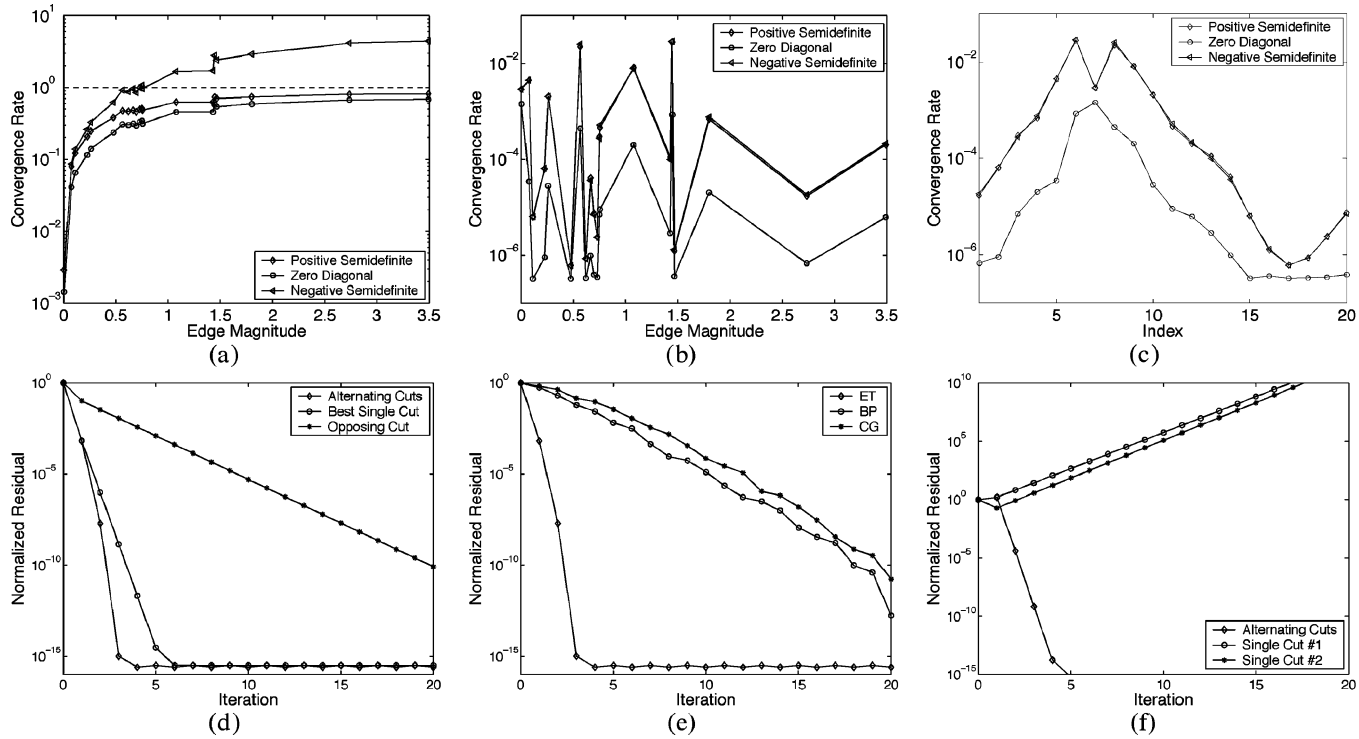

Fig. 5. Behavior of the ET algorithm on a single 20-node cycle. (a) Single-tree convergence rate versus edge strength. (b) Two-tree convergence rate versus edge strength. (c) Two-tree convergence rate versus edge index. (d) Comparison of zero diagonal single-tree and two-tree iterations. (e) Comparison to belief propagation (BP) and conjugate gradient (CG). (f) Two individually divergent cutting matrices produce a convergent two-tree iteration.

the performance of the parallel BP and unpreconditioned CG iterations, showing that for this problem, the ET algorithm is much faster.

The previous plots suggest that a two-tree ET iteration may exhibit features quite different from those observed in the corresponding single-tree iterations. This is dramatically demonstrated by Fig. 5(f), which considers the negative semidefinite cutting matrices corresponding to the two *strongest* edges in the graph. As predicted by Fig. 5(a), the single-tree iterations corresponding to these edges are divergent. However, because these strong edges are widely separated in the original graph (indexes 1 and 12), they lead to a two-tree iteration that outperforms even the best single-tree iteration.

D. Convergence Criteria

When the ET mean recursion periodically cycles through a fixed set of T cutting matrices, Proposition 1 shows that its convergence depends entirely on the eigenstructure of $S = \prod_{n=1}^{T} \widehat{J}_{\mathcal{T}_n}^{-1} K_{\mathcal{T}_n}$. However, this matrix is never explicitly calculated by the ET recursion, and for large-scale problems, direct determination of its eigenvalues is intractable. In this section, we derive several conditions that allow a more tractable assessment of the ET algorithm's convergence.

Consider first the case where the ET algorithm uses the same cutting matrix $K_{\mathcal{T}}$ at every iteration. We then have a standard Richardson iteration, for which the following theorem, proved by Adams [34], provides a simple necessary and sufficient convergence condition:

Theorem 2: Let $\widehat{J}$ be a symmetric positive definite matrix and $K_{\mathcal{T}}$ be a symmetric cutting matrix such that $(\widehat{J} + K_{\mathcal{T}})$ is nonsingular. Then

$$\rho\left((\widehat{J} + K_{\mathcal{T}})^{-1} K_{\mathcal{T}}\right) < 1, \quad \text{if and only if} \quad \widehat{J} + 2K_{\mathcal{T}} \succ 0.$$

The following two corollaries provide simple procedures for satisfying the conditions of Theorem 2:

Corollary 1: If the ET algorithm is implemented with a single positive semidefinite cutting matrix $K_{\mathcal{T}}$, the resulting iteration will be convergent for any positive definite inverse error covariance $\widehat{J}$.

Proof: If $\widehat{J}$ is positive definite and $K_{\mathcal{T}}$ is positive semidefinite, then $(\widehat{J} + 2K_{\mathcal{T}})$ is positive definite. □

Corollary 2: Suppose that $\widehat{J}$ is diagonally dominant so that

$$\widehat{J}_{s,s} > \sum_{t \in \Gamma(s)} |\widehat{J}_{s,t}| \tag{20}$$

for all $s \in \mathcal{V}$. Then, any regular cutting matrix with non-negative diagonal entries will produce a convergent embedded trees iteration.

Proof: Regular cutting matrices only modify the off-diagonal entries of $\widehat{J}$ by setting certain elements to zero. Therefore, the entries set to zero in $(\widehat{J} + K_{\mathcal{T}})$ will simply have their signs flipped in $(\widehat{J} + 2K_{\mathcal{T}})$, leaving the summation in (20) unchanged. Then, by the assumption that $K_{\mathcal{T}}$ has non-negative diagonal entries, we are assured that $(\widehat{J} + 2K_{\mathcal{T}})$ is diagonally dominant and, hence, positive definite. □

Note that Corollary 2 ensures that if $\widehat{J}$ is diagonally dominant, the zero diagonal cutting matrices of Section III-C will produce convergent ET iterations.

Although Theorem 2 completely characterizes the conditions under which the single tree ET iteration converges, it says nothing about the resulting convergence rate. The following theorem allows the convergence rate to be bounded in certain circumstances.

Theorem 3: When implemented with a single positive semidefinite cutting matrix $K_{\mathcal{T}}$, the convergence rate of the ET algorithm is bounded by

$$\frac{\lambda_{\max}(K_{\mathcal{T}})}{\lambda_{\max}(K_{\mathcal{T}})+\lambda_{\max}(\widehat{J})} \leq \rho\left(\widehat{J}_{\mathcal{T}}^{-1}K_{\mathcal{T}}\right) \leq \frac{\lambda_{\max}(K_{\mathcal{T}})}{\lambda_{\max}(K_{\mathcal{T}})+\lambda_{\min}(\widehat{J})}.$$

Proof: This follows from [35, Th. 2.2]; see [2, Th. 3.14] for details. □

Increasing the diagonal entries of $K_{\mathcal{T}}$ will also increase the upper bound on $\rho(\widehat{J}_{\mathcal{T}}^{-1}K_{\mathcal{T}})$ provided by Theorem 3. This matches the observation made in Section III-C that positive semidefinite cutting matrices tend to produce slower convergence rates.

When the ET algorithm employs multiple trees, obtaining a precise characterization of its convergence behavior is more difficult. The following theorem provides a simple set of sufficient conditions for two-tree iterations.

Theorem 4: Consider the embedded trees iteration generated by a pair of cutting matrices $\{K_{\mathcal{T}_1}, K_{\mathcal{T}_2}\}$. Suppose that the following three matrices are positive definite:

$$\widehat{J}+K_{\mathcal{T}_1}+K_{\mathcal{T}_2} \succ 0 \quad \widehat{J}+K_{\mathcal{T}_1}-K_{\mathcal{T}_2} \succ 0 \quad \widehat{J}-K_{\mathcal{T}_1}+K_{\mathcal{T}_2} \succ 0.$$

Then, the resulting iteration is convergent ($\rho(S) < 1$).

Proof: See [2, App. C.4]. □

The conditions of this theorem show that in the multiple tree case, there may be important interactions between the cutting matrices that affect the convergence of the composite iteration. These interactions were demonstrated by the single cycle inference results of Section III-C and are further explored in Section VI. Note that it is *not* necessary for the cutting matrices to be individually convergent, as characterized by Theorem 2, in order for the conditions of Theorem 4 to be satisfied. When more than two trees are used, a singular value analysis may be used to provide sufficient conditions for convergence; see [2, Sec. 3.4.2] for details.

IV. Exact Error Variance Calculation

The ET algorithm introduced in the preceding section used a series of exact computations on spanning trees to calculate the conditional mean $\widehat{x}$ of a loopy Gaussian inference problem. In this section, we examine the complementary problem of determining marginal error variances[5] $\{\widehat{P}_s | s \in \mathcal{V}\}$. We develop a class of iterative algorithms for calculating error variances that are particularly efficient for very sparsely connected loopy graphs, like that of Section I-A.

Due to the linear algebraic structure underlying Gaussian inference problems, any procedure for calculating $\widehat{x}$ may be easily adapted to the calculation of error variances. In particular, suppose that the full error covariance matrix $\widehat{P}$ is partitioned into columns $\{\hat{p}_i\}_{i=1}^{Nd}$. Then, letting e_i be an Nd-dimensional vector of zeros with a one in the ith position, the ith column of $\widehat{P}$ is equal to $\widehat{P}e_i$, or equivalently

$$\widehat{J}\hat{p}_i = e_i. \tag{21}$$

By comparison to (7), we see that $\hat{p}_i$ is equal to the conditional mean of a particular inference problem defined by the synthetic observation vector e_i. Thus, given an inference procedure like the ET algorithm that calculates conditional means at a cost of $\mathcal{O}(Nd^3)$ per iteration, we may calculate a series of approximations to $\widehat{P}$ using $\mathcal{O}(N^2d^4)$ operations per iteration.

While the procedure described in the previous paragraph is theoretically sound, the computational cost may be too large for many applications. We would prefer an algorithm that only calculates the N desired marginal error variances $\widehat{P}_s$, avoiding the $\mathcal{O}(N^2)$ cost that any algorithm calculating all of $\widehat{P}$ must require. Consider the ET iteration generated by a single cutting matrix $K_{\mathcal{T}}$ chosen so that $\rho(\widehat{J}_{\mathcal{T}}^{-1}K_{\mathcal{T}}) < 1$. When $\widehat{x}^0 = 0$, a simple induction argument shows that subsequent iterations $\widehat{x}^n$ may be expressed as a series of linear functions of the normalized observation vector $\bar{y}$:

$$\widehat{x}^n = \left[\widehat{J}_{\mathcal{T}}^{-1} + F_n\right]\bar{y} \tag{22}$$

$$F_n = \widehat{J}_{\mathcal{T}}^{-1}K_{\mathcal{T}}\left[\widehat{J}_{\mathcal{T}}^{-1} + F_{n-1}\right]$$

$$F_1 = \mathbf{0}. \tag{23}$$

Since we have assumed that $\rho(\widehat{J}_{\mathcal{T}}^{-1}K_{\mathcal{T}}) < 1$, Proposition 1 guarantees that $\widehat{x}^n \overset{n\to\infty}{\longrightarrow} \widehat{x}$ for any $\bar{y}$. Therefore, the sequence of matrices defined by (22), (23) must converge to $\widehat{P}$:

$$\widehat{P} = \sum_{n=0}^{\infty} \widehat{J}_{\mathcal{T}}^{-1}\left[K_{\mathcal{T}}\widehat{J}_{\mathcal{T}}^{-1}\right]^n = \lim_{n\to\infty}\left(\widehat{J}_{\mathcal{T}}^{-1} + F_n\right). \tag{24}$$

In fact, these matrices correspond exactly to the series expansion of $\widehat{P}$ generated by the following fixed-point equation:

$$\widehat{P} = \widehat{J}_{\mathcal{T}}^{-1} + \widehat{J}_{\mathcal{T}}^{-1}K_{\mathcal{T}}\widehat{P}. \tag{25}$$

The matrix sequence of (24) may be derived by repeatedly using (25) to expand itself.

Clearly, if we could somehow track the matrices composing the ET series expansion (24), we could recover the desired error covariance matrix $\widehat{P}$. In order to perform this tracking efficiently, we must exploit the fact that, as discussed in Section III, for many models, the cutting matrix $K_{\mathcal{T}}$ is low rank. This allows us to focus our computation on a particular low-dimensional subspace, leading to computational gains when $\text{rank}(K_{\mathcal{T}}) < N$. We begin in Section IV-A by presenting techniques for explicitly constructing rank-revealing decompositions of cutting matrices. Then, in Section IV-B, we use these decompositions to calculate the desired error variances.

A. Low-Rank Decompositions of Cutting Matrices

For any cutting matrix $K_{\mathcal{T}}$, there exist many different additive decompositions into rank-one terms:

$$K_{\mathcal{T}} = \sum_i \omega_i u_i u_i^T, \qquad u_i \in \mathbb{R}^{Nd}. \tag{26}$$

[5]When nodes represent Gaussian vectors ($d > 1$), $\widehat{P}_s$ is actually a d-dimensional covariance matrix. However, to avoid confusion with the full error covariance matrix $\widehat{P}$, we always refer to $\{\widehat{P}_s | s \in \mathcal{V}\}$ as error variances.

For example, because any symmetric matrix has an orthogonal set of eigenvectors, the eigendecomposition $K_{\mathcal{T}} = UDU^T$ is of this form. However, for large graphical models, the $\mathcal{O}(N^3d^3)$ cost of direct eigenanalysis algorithms is intractable. In this section, we provide an efficient decomposition procedure that exploits the structure of $K_{\mathcal{T}}$ to reduce the number of needed terms.

Consider a regular cutting matrix $K_{\mathcal{T}}$ that acts to remove the edges $\mathcal{E} \setminus \mathcal{E}_{\mathcal{T}}$ from the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, and let $E = |\mathcal{E} \setminus \mathcal{E}_{\mathcal{T}}|$. The decomposition we construct depends on a set of *key nodes* defined as follows.

Definition 2: Consider a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with associated embedded tree $\mathcal{G}_{\mathcal{T}} = (\mathcal{V}, \mathcal{E}_{\mathcal{T}})$. A subset $\mathcal{W} \subset \mathcal{V}$ of the vertices forms a key node set if for any cut edge $(s,t) \in \mathcal{E} \setminus \mathcal{E}_{\mathcal{T}}$, either s or t (or both) belong to $\mathcal{W}$.

In other words, at least one end of every cut edge must be a key node. In most graphs, there will be many ways to choose $\mathcal{W}$ (see Fig. 6). In such cases, the size of the resulting decomposition is minimized by minimizing $W \triangleq |\mathcal{W}|$. Note that $\mathcal{W}$ may always be chosen so that $W \leq E$.

Given a set of key nodes $\mathcal{W}$ of dimension d, we decompose $K_{\mathcal{T}}$ as

$$K_{\mathcal{T}} = \sum_{w \in \mathcal{W}} H_w \tag{27}$$

where H_w is chosen so that its only nonzero entries are in the d rows and columns corresponding to w. Because every cut edge adjoins a key node, this is always possible. The following proposition shows how each H_w may be further decomposed into rank-one terms.

Proposition 3: Let H be a symmetric matrix whose only nonzero entries lie in a set of d rows and the corresponding columns. Then, the rank of H is at most $2d$.

Proof: See Appendix. □

Thus, the rank of a cutting matrix with W corresponding key nodes can be at most $2Wd$. The proof of Proposition 3 is constructive, providing an explicit procedure for determining a rank-one decomposition as in (26). The cost of this construction is at most $\mathcal{O}(NWd^3)$ operations. However, in the common case where the size of each node's local neighborhood does not grow with N, this reduces to $\mathcal{O}(Wd^3)$. Note that because at most one key node is needed for each cut edge, Proposition 3 implies that a cutting matrix $K_{\mathcal{T}}$, which cuts E edges, may always be decomposed into at most $2Ed$ terms. When $d = 1$, the number of terms may be reduced to only E by appropriately choosing the diagonal entries of $K_{\mathcal{T}}$ (see [2, Sec. 3.3.1] for details), which is a fact we exploit in the simulations of Section VI-B.

B. Fixed-Point Error Variance Iteration

In this section, we provide an algorithm for tracking the terms of the ET series expansion (24). Since the block diagonal entries of $\widehat{J}_{\mathcal{T}}^{-1}$ may be determined in $\mathcal{O}(Nd^3)$ operations by an exact recursive algorithm, one obvious solution would be to directly track the evolution of the F_n matrices. This possibility is explored in [2, Sec. 3.3.3], where it is shown that low-rank decompositions of F_n may be recursively updated in $\mathcal{O}(NW^2d^5)$ operations per iteration. Here, however, we focus on a more efficient approach based on a specially chosen set of synthetic inference problems.

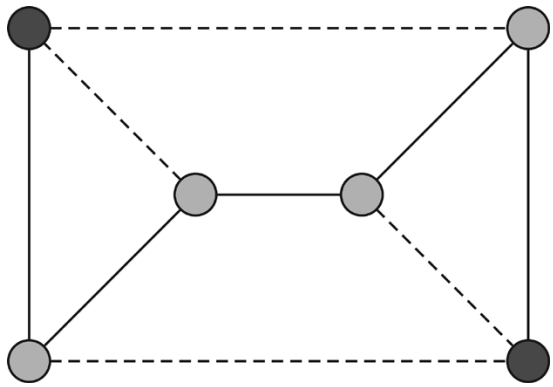

Fig. 6. Graphical representation of a spanning tree (solid) and the corresponding cut edges (dashed). Both of the shaded groups of nodes define key node sets, but the darker one would produce a more efficient decomposition.

We begin by considering the fixed point (25) characterizing the single tree ET series expansion. Using the low-rank decomposition of $K_{\mathcal{T}}$ given by (26), we have

$$\begin{aligned}\widehat{P} &= \widehat{J}_{\mathcal{T}}^{-1} + \widehat{J}_{\mathcal{T}}^{-1}\left(\sum_i \omega_i u_i u_i^T\right)\widehat{P} \\ &= \widehat{J}_{\mathcal{T}}^{-1} + \sum_i \omega_i \left(\widehat{J}_{\mathcal{T}}^{-1} u_i\right)(\widehat{P}u_i)^T. \end{aligned} \tag{28}$$

As discussed in Section IV-A, the decomposition of $K_{\mathcal{T}}$ may always be chosen to have at most $\mathcal{O}(Wd)$ vectors u_i. Each of the terms in (28) has a probabilistic interpretation. For example, $\widehat{J}_{\mathcal{T}}^{-1}$ is the covariance matrix of a tree-structured graphical model. Similarly, $\widehat{J}_{\mathcal{T}}^{-1}u_i$ and $\widehat{P}u_i$ are the conditional means of synthetic inference problems with observations u_i.

Algorithm 1 shows how all of the terms in (28) may be efficiently computed using the inference algorithms developed earlier in this paper. The computational cost is dominated by the solution of the synthetic estimation problems on the graph with cycles [step 3(b)]. Any inference algorithm can be used in this step, including the ET algorithm (using one or multiple trees), loopy BP, or conjugate gradient. Thus, this fixed-point algorithm can effectively transform any method for computing conditional means into a fast error variance iteration.

Although (28) was motivated by the ET algorithm, nothing about this equation requires that the cutting matrix produce a tree-structured graph. All that is necessary is that the remaining edges form a structure for which exact error variance calculation is tractable. In addition, note that (28) gives the correct perturbation of $\widehat{J}_{\mathcal{T}}^{-1}$ for calculating the entire error covariance $\widehat{P}$ and not just the block diagonals $\widehat{P}_s$. Thus, if the BP algorithm is extended to calculate a particular set of off-diagonal elements of $\widehat{J}_{\mathcal{T}}^{-1}$, the exact values of the corresponding entries of $\widehat{P}$ may be found in the same manner.

Given a tree-structured matrix splitting $\widehat{J} = (\widehat{J} - K_{\mathcal{T}})$, where the cutting matrix $K_{\mathcal{T}}$ has W key nodes:

1) Compute the block diagonal entries of $\widehat{J}_{\mathcal{T}}^{-1}$ using the BP algorithm ($\mathcal{O}(Nd^3)$ operations).

2) Determine a rank one decomposition of $K_{\mathcal{T}}$ into $\mathcal{O}(Wd)$ vectors u_i, as in (26).

3) For each of the $\mathcal{O}(Wd)$ vectors u_i:

a) Compute $\widehat{J}_{\mathcal{T}}^{-1} u_i$ using the BP algorithm ($\mathcal{O}(Nd^3)$ operations).

b) Compute $\widehat{P} u_i$ using *any* convenient conditional mean estimation algorithm (typically $\mathcal{O}(Nd^3)$ operations per iteration).

4) Using the results of the previous steps, calculate the block diagonal entries of $\widehat{P}$ as in (28) ($\mathcal{O}(NWd^2)$ operations).

Algorithm 1. ET fixed-point algorithm for computing error variances. The overall computational cost is $\mathcal{O}(NWd^4)$ operations for each conditional mean iteration in step 3(b).

V. Tree-Structured Preconditioners

Preconditioners play an important role in accelerating the convergence of the CG method. In Section III-B, we demonstrated that whenever the ET algorithm is implemented by periodically cycling through a fixed set of T cutting matrices, it is equivalent to a preconditioned Richardson iteration. An explicit formula for the implicitly generated preconditioning matrix is given in Proposition 2. By simply applying the standard ET iteration in (14) once for each of the T cutting matrices, we can compute the product of the ET preconditioning matrix with any vector in $\mathcal{O}(TNd^3)$ operations. This is the only operation needed to use the ET iteration as a preconditioner for CG [28], [30].

In this section, we explore the theoretical properties of embedded tree-based preconditioners (see Section VI for simulation results). For a preconditioning matrix to be used with CG, it must be symmetric. While any single-tree ET iteration leads to a symmetric preconditioner, the preconditioners associated with multiple-tree iterations are in general nonsymmetric. It is possible to choose multiple-tree cutting matrices so that symmetry is guaranteed. However, in dramatic contrast to the tree-based Richardson iterations of Section III, multiple tree preconditioners typically perform slightly *worse* than their single-tree counterparts [2, Sec. 4.3]. Thus, in this paper, we focus our attention on single tree preconditioners.

If a single spanning tree, generated by cutting matrix $K_{\mathcal{T}}$, is used as a preconditioner, the system which CG effectively solves is given by

$$(\widehat{J} + K_{\mathcal{T}})^{-1} \widehat{J} \widehat{x} = (\widehat{J} + K_{\mathcal{T}})^{-1} \overline{y}. \tag{29}$$

The convergence rate of CG is determined by how well the eigenspectrum of the *preconditioned* system $(\widehat{J} + K_{\mathcal{T}})^{-1} \widehat{J}$ can be fit by a polynomial [28]. Effective preconditioners produce smoother, flatter eigenspectra that are accurately fit by a lower order polynomial.

Several authors have recently rediscovered [36], analyzed [37], and extended [38], [39] a technique called support graph theory, which provides powerful methods for constructing effective preconditioners from maximum-weight spanning trees. Support graph theory is especially interesting because it provides a set of results guaranteeing the effectiveness of the resulting preconditioners. Preliminary experimental results [37] indicate that tree-based preconditioners can be very effective for many, but not all, of the canonical test problems in the numerical linear algebra literature.

In general, the support graph literature has focused on tree-based preconditioners for relatively densely connected graphs, such as nearest-neighbor grids. However, for graphs that are nearly tree-structured ($K_{\mathcal{T}}$ is low rank), it is possible to make stronger statements about the convergence of preconditioned CG, as shown by the following theorem.

Theorem 5: Suppose that the conjugate gradient algorithm is used to solve the Nd-dimensional preconditioned linear system given by (29). Then, if the cutting matrix $K_{\mathcal{T}}$ has $\text{rank}(K_{\mathcal{T}}) = m$, the preconditioned CG method will converge to the exact solution in at most $m + 1$ iterations.

Proof: Let λ be any eigenvalue of $(\widehat{J} + K_{\mathcal{T}})^{-1} \widehat{J}$, and let v be one of its corresponding eigenvectors. By definition, we have

$$\begin{aligned} (\widehat{J} + K_{\mathcal{T}})^{-1} \widehat{J} v &= \lambda v \\ (1 - \lambda) \widehat{J} v &= \lambda K_{\mathcal{T}} v. \end{aligned} \tag{30}$$

Since $\text{rank}(K_{\mathcal{T}}) = m$, there exist $N - m$ linearly independent eigenvectors in the null space of $K_{\mathcal{T}}$. Each one of these eigenvectors satisfies (30) when $\lambda = 1$. Thus, $\lambda = 1$ is an eigenvalue of $(\widehat{J} + K_{\mathcal{T}})^{-1} \widehat{J}$, and its multiplicity is at least $N - m$.

Let $\{\lambda_i\}_{i=1}^{m}$ denote the m eigenvalues of $(\widehat{J} + K_{\mathcal{T}})^{-1} \widehat{J}$ not constrained to equal one. Consider the $(m+1)$st-order polynomial $p_{m+1}(\lambda)$ defined as

$$p_{m+1}(\lambda) = (1 - \lambda) \prod_{i=1}^{m} \frac{(\lambda_i - \lambda)}{\lambda_i}. \tag{31}$$

By construction, $p_{m+1}(0) = 1$. Let $A \triangleq (\widehat{J} + K_{\mathcal{T}})^{-1} \widehat{J}$. Then, from [28, p. 313], we see that

$$\frac{\|r^{m+1}\|_{A^{-1}}}{\|r^0\|_{A^{-1}}} \leq \max_{\lambda \in \{\lambda_i(A)\}} |p_{m+1}(\lambda)| \tag{32}$$

where r^{m+1} is the residual at iteration $m+1$. Since $p_{m+1}(\bar{\lambda}) = 0$ for all $\bar{\lambda} \in \{\lambda_i(A)\}$, we must have $r^{m+1} = 0$. Therefore, CG must converge by iteration $m + 1$. □

Thus, when the cutting matrix rank is smaller than the dimension of $\widehat{J}$, the preconditioned CG iteration is guaranteed to converge in strictly fewer iterations than the unpreconditioned method.

When combined with the results of the previous sections, Theorem 5 has a number of interesting implications. In particular, from Section IV-A, we know that a cutting matrix $K_{\mathcal{T}}$ with W associated key nodes can always be chosen so that $\text{rank}(K_{\mathcal{T}}) \leq \mathcal{O}(Wd)$. Then, if this cutting matrix is used to precondition the CG iteration, we see that the conditional mean

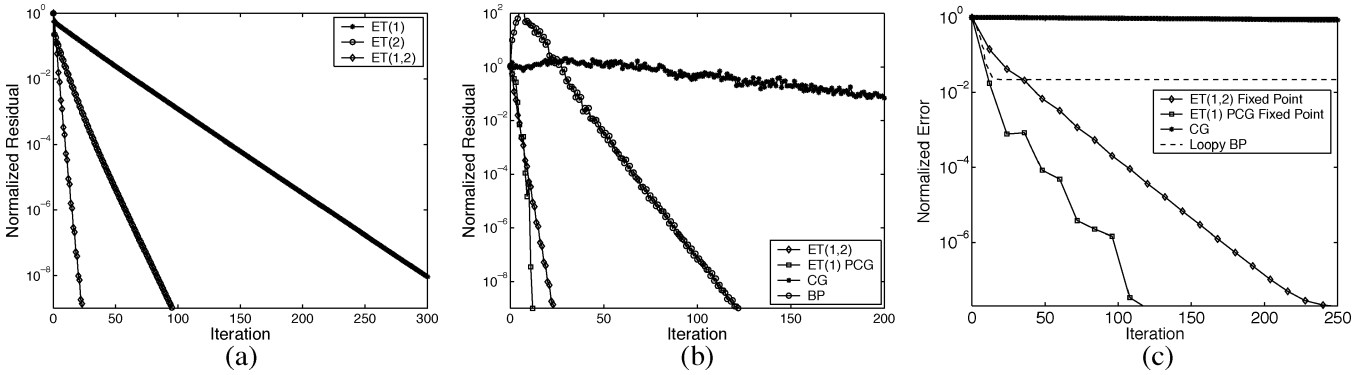

Fig. 7. Comparison of inference methods on the augmented multiscale model of Figs. 1 and 2. (a) Single and alternating tree ET iterations. (b) Comparison to other conditional mean iterations. (c) Error variance methods.

can be *exactly* (noniteratively) calculated in $\mathcal{O}(NWd^4)$ operations. Similarly, if single-tree preconditioned CG is used to solve the synthetic estimation problems [step 3(b)] of the fixed-point error variance algorithm (Section IV-B), error variances can be exactly calculated in $\mathcal{O}(NW^2d^5)$ operations. Note that in problems where W is reasonably large, we typically hope (and find) that iterative methods will converge in less than $\mathcal{O}(Wd)$ iterations. Nevertheless, it is useful to be able to provide such guarantees of worst-case computational cost. See the following section for examples of models where finite convergence dramatically improves performance.

Exact error variances may also be calculated by a recently proposed extended message passing algorithm [27], which defines additional messages to cancel the "fill" associated with Gaussian elimination on graphs with cycles [12]. Like the ET fixed point iteration of Algorithm 1, the extended message-passing procedure is based on accounting for perturbations from a spanning tree of the graph. The cost of extended message passing is $\mathcal{O}(NL^2)$, where L is the number of nodes adjacent to *any* cut edge. In contrast, Algorithm 1 requires $\mathcal{O}(NW^2)$ operations, where the number of key nodes W is strictly less than L (and sometimes much less, as in Section VI-C). More importantly, extended message passing is noniterative and always requires the full computational cost, whereas Algorithm 1 produces a series of approximate error variances that are often accurate after far fewer than W iterations.

VI. Simulations

In this section, we present several computational examples that explore the empirical performance of the inference algorithms developed in this paper. When calculating conditional means, we measure convergence using the normalized residual of (11). For error variance simulations, we use the normalized error metric

$$\frac{\left(\sum_{s\in\mathcal{V}}\left|\widehat{P}_s^n-\widehat{P}_s\right|^2\right)^{\frac{1}{2}}}{\left(\sum_{s\in\mathcal{V}}|\widehat{P}_s|^2\right)^{\frac{1}{2}}} \tag{33}$$

where $\widehat{P}_s$ are the true error variances, and $\widehat{P}_s^n$ are the approximations at the nth iteration. Error variance errors are always plotted versus the number of equivalent BP iterations; therefore, we properly account for the extra cost of solving multiple synthetic inference problems in the ET fixed point method [Algorithm 1, step 3(b)].

A. Multiscale Modeling

In Fig. 7, we compare the performance of the inference methods developed in this paper on the augmented multiscale model $\mathcal{G}_{\text{aug}}$ (see Figs. 1 and 2) constructed in Section I-A. To do this, we associate a 2-D observation vector $y_s = x_s + v_s$ with each of the 64 finest scale nodes, where v_s is independent noise with variance equal to the marginal prior variance of x_s. The resulting inverse error covariance matrix is not well conditioned[6] ($\kappa(\widehat{J}) \approx 3.9 \times 10^4$), making this a challenging inference problem for iterative methods.

We first consider the ET Richardson iteration of Section III. We use zero-diagonal regular cutting matrices to create two different spanning trees: the multiscale tree $\mathcal{G}_{\text{mar}}$ of Fig. 1 and an alternative tree where three of the four edges connecting the second and third coarsest scales are removed. These cutting matrices provide a pair of single-tree iterations [denoted by ET(1) and ET(2)], as well as a two-tree iteration [denoted by ET(1,2)], which alternates between trees. As shown in Fig. 7(a), despite the large condition number, all three iterations converge at an asymptotically linear rate as predicted by Proposition 1. However, as in Section III-C, the ET(1,2) iteration converges much faster than either single-tree method.

In Fig. 7(b), we compare the ET(1,2) iteration to CG and loopy BP, as well as to the single-tree preconditioned CG (PCG) algorithm of Section V. Due to this problem's large condition number, BP converges rather slowly, whereas standard CG behaves extremely erratically, requiring over 1000 iterations for convergence. In contrast, since the cutting matrices are rank 12 for this graph, Theorem 5 guarantees that (ignoring numerical issues) the ET PCG iteration will converge in at most 13 iterations. Since the preconditioned system is much better conditioned ($\kappa((\widehat{J}+K_{\mathcal{T}})^{-1}\widehat{J}) \approx 273$), this finite convergence is indeed observed.

Finally, we compare the ET fixed-point error variance iteration (Section IV-B, Algorithm 1) to loopy BP's approximate

[6]For a symmetric positive definite matrix J, the condition number is defined as $\kappa(J) \triangleq (\lambda_{\max}(J)/\lambda_{\min}(J))$.

TABLE I
CONDITIONAL MEAN SIMULATION RESULTS: NUMBER OF ITERATIONS BEFORE CONVERGENCE TO A NORMALIZED RESIDUAL OF 10^{-10}. FOR THE DISORDERED CASE, WE REPORT THE AVERAGE NUMBER OF ITERATIONS ACROSS 100 RANDOM PRIOR MODELS, PLUS OR MINUS TWO STANDARD DEVIATIONS

Inference	Augmented Tree		Nearest–Neighbor Grid	
Algorithm	Homogeneous	Disordered	Homogeneous	Disordered
ET(1)	55	123.0 ± 145.4	331	219.3 ± 107.0
ET(2)	37	82.9 ± 110.0	346	216.8 ± 114.5
ET(1,2)	13	11.1 ± 6.0	314	110.8 ± 34.7
ET(1) PCG	4	4 ± 0.0	59	47.7 ± 4.8
CG	60	95.0 ± 8.0	78	85.8 ± 8.1
BP	54	42.5 ± 10.6	380	62.6 ± 19.6

error variances, as well as variances derived from the CG iteration, as in [31]. We consider two options for solving the synthetic inference problems [step 3(b)]: the ET(1,2) Richardson iteration and single-tree PCG. As shown in Fig. 7(c), this problem's poor conditioning causes the CG error variance formulas to produce very inaccurate results that never converge to the correct solution. Although loopy BP converges to somewhat more accurate variances, even a single iteration of the PCG fixed-point iteration produces superior estimates. We also see that the PCG method's advantages over ET(1,2) for conditional mean estimation translate directly to more rapid error variance convergence.

B. Sparse versus Dense Graphs with Cycles

This section examines the performance of the ET algorithms on graphs with randomly generated potentials. We consider two different graphs: the sparse augmented multiscale graph $\mathcal{G}_{\text{aug}}$ of Fig. 1 and a more densely connected, 20×20 nearest-neighbor grid (analogous to the small grid in Fig. 4). For grids, one must remove $\mathcal{O}(N)$ edges to reveal an embedded tree so that the ET fixed-point error variance algorithm requires $\mathcal{O}(N^2)$ operations per iteration. This cost is intractable for large N; therefore, in this section, we focus solely on the calculation of conditional means.

For each graph, we assume all nodes represent scalar Gaussian variables and consider two different potential function assignments. In the first case, we create a *homogeneous* model by assigning the same attractive potential

$$\psi_{s,t}(x_s, x_t) = \exp\left\{-\frac{1}{2}(x_s - x_t)^2\right\} \tag{34}$$

to each edge. We also consider *disordered* models, where each edge is randomly assigned a different potential of the form

$$\psi_{s,t}(x_s, x_t) = \exp\left\{-\frac{1}{2}w_{st}(x_s - a_{st}x_t)^2\right\}. \tag{35}$$

Here, w_{st} is sampled from an exponential distribution with mean 1, whereas a_{st} is set to $+1$ or -1 with equal probability. These two model types provide extreme test cases, each revealing different qualitative features of the proposed algorithms.

To create test inference problems, we assign a measurement of the form $y_s = x_s + v_s, v_s \sim (0, 10)$ to each node in the graph. We consider high noise levels because they lead to the most (numerically) challenging inference problems. For the augmented multiscale model, the ET algorithms use the same spanning trees as in Section VI-A, whereas the grid simulations use trees similar to the first two spanning trees in Fig. 4.

Table I lists the number of iterations required for each algorithm to converge to a normalized residual of 10^{-10}. For the disordered case, we generated 100 different random graphical priors, and we report the mean number of required iterations, plus or minus two standard deviations. For the augmented multiscale model, as in Section VI-A, we find that the ET(1,2) and ET PCG methods both dramatically outperform their competitors. In particular, since the cutting matrix must remove only three edges and nodes represent scalar variables, preconditioned CG always finds the exact solution in four iterations.

For the nearest-neighbor grid, the performance of the ET Richardson iterations is typically worse than other methods. However, even on this densely connected graph, we find that single-tree preconditioning improves CGs convergence rate, particularly for disordered potentials. This performance is not predicted by Theorem 5 but is most likely attributable to the spectral smoothing provided by the tree-based preconditioner.

C. Distributed Sensor Networks

In this section, we examine an inference problem motivated by applications involving networks of distributed sensors. Fig. 8(a) shows a graph in which each of the 600 nodes is connected to its spatially nearest neighbors, except for the three central nodes, which produce long-range connections (dashed). Most of the nodes are clustered near the perimeter of a square region: a configuration suggestive of surveillance applications in which nodes represent simple, range-limited sensors. The few long-range relationships might be caused by the presence of nodes with long-range communications abilities or spatial sensitivities or could represent objects sensed in the environment.[7] We assign potentials to this graph using the disordered distribution of Section VI-B. We assume that exact inference is possible for the graph consisting of only the black nodes (e.g. via local clustering) but difficult when the longer range relationships are introduced. Thus, for this example, the "tree-based" algorithms developed earlier in this paper will

[7]While distributed sensor networks motivate the structure of this example, our goal here is to illustrate the features of the ET estimation algorithms, and we do not resolve many issues presented by real sensor networks. In particular, while our algorithms involve message passing, truly distributed implementation of ET algorithms involves issues beyond the scope of this paper. For example, in real networks, nodes representing objects cannot participate in communication and computation. This is a topic of ongoing research.

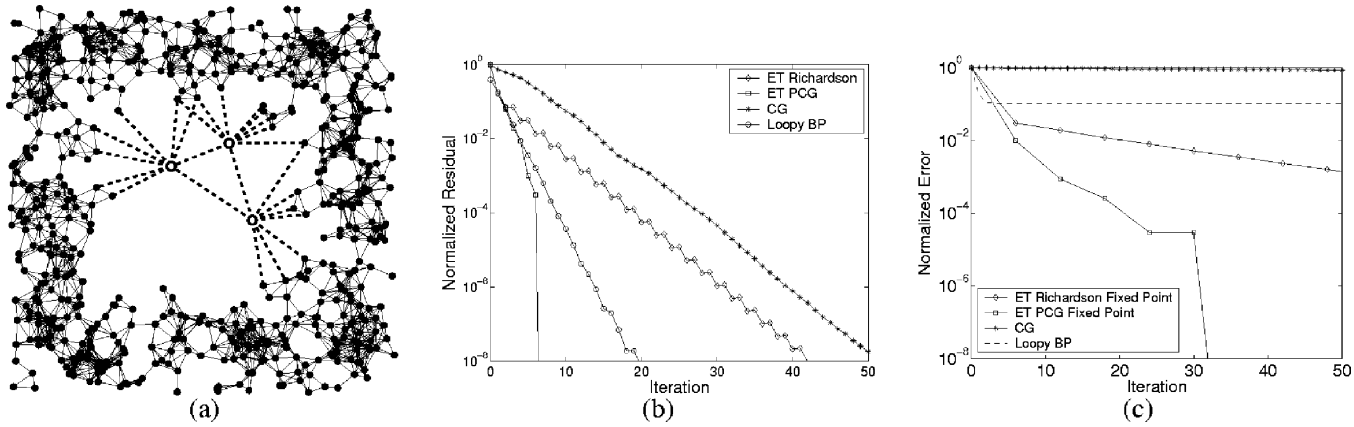

Fig. 8. Distributed sensing example. (a) Sensor network where the edges (dashed) connected to three nodes (open circles) lead to nonlocal interactions. The solid nodes form a core structure that is solved exactly at each step of the presented inference results. (b) Conditional mean results. (c) Error variance results.

actually solve this (loopy) system of local interactions at each iteration.

Fig. 8(b) compares several different methods for calculating conditional means on this graph. BP performs noticeably better than either unpreconditioned CG or the single "tree" Richardson iteration. However, since the three central nodes form a key node set (see Section IV-A), the cutting matrix has rank 6, and preconditioned CG converges to the exact answer in only seven iterations. For the error variance calculation [see Fig. 8(c)], the gains are more dramatic. BP quickly converges to a suboptimal answer, but after only one iteration of the six synthetic inference problems, the fixed-point covariance method finds a superior solution. The CG error variance formulas of [31] are again limited by finite precision effects.

VII. Discussion

We have proposed and analyzed a new class of embedded trees (ET) algorithms for iterative, exact inference on Gaussian graphical models with cycles. Each ET iteration exploits the presence of a subgraph for which exact estimation is tractable. Analytic and experimental results demonstrate that the ET preconditioned conjugate gradient method rapidly computes conditional means even on very densely connected graphs. The complementary ET error variance method is most effective for sparser graphs. We provide two examples, drawn from the fields of multiscale modeling and distributed sensing, which show how such graphs may naturally arise.

Although we have developed inference algorithms based on tractable embedded subgraphs, we have not provided a procedure for choosing these subgraphs. Our results indicate that there are important interactions among cut edges, suggesting that simple methods (e.g. maximal spanning trees) may not provide the best performance. Although support graph theory [36], [37] provides some guarantees for embedded trees, extending these methods to more general subgraphs is an important open problem.

The multiscale modeling example of Section I-A suggests that adding a small number of edges to tree-structured graphs may greatly increase their effectiveness, in particular alleviating the commonly encountered boundary artifact problem. The ET algorithms demonstrate that it is indeed possible to perform efficient, exact inference on such augmented multiscale models. However, our methods also indicate that this computational feasibility may depend on how quickly the number of "extra" edges must grow as a function of the process size. This edge allocation problem is one example of a more general modeling question: How should hidden graphical structures be chosen to best balance the goals of model accuracy and computational efficiency?

Appendix

We present the proof of Proposition 3. Without loss of generality, assume that the nonzero entries of H lie in the first d rows and columns. Let H be partitioned as $H = \begin{bmatrix} A & B^T \\ B & 0 \end{bmatrix}$, where A is a $d \times d$ symmetric matrix. Similarly, partition the eigenvectors of H as $v = \begin{bmatrix} v_a \\ v_b \end{bmatrix}$, where v_a is of dimension d. The eigenvalues of H then satisfy

$$Av_a + B^T v_b = \lambda v_a \tag{36}$$

$$Bv_a = \lambda v_b. \tag{37}$$

Suppose that $\lambda \neq 0$. From (37), v_b is uniquely determined by λ and v_a. Plugging (37) into (36), we have

$$\lambda^2 v_a - \lambda A v_a - B^T B v_a = 0. \tag{38}$$

This d-dimensional symmetric quadratic eigenvalue problem has at most $2d$ distinct solutions. Thus, H can have at most $2d$ nonzero eigenvalues, and $\text{rank}(H) \leq 2d$.

Acknowledgment

The authors would like to thank Dr. M. Schneider for many helpful discussions.

References

[1] M. J. Wainwright, E. B. Sudderth, and A. S. Willsky, "Tree-based modeling and estimation of Gaussian processes on graphs with cycles," in *Neural Information Processing Systems 13*. Cambridge, MA: MIT Press, 2001, pp. 661–667.

[2] E. B. Sudderth, "Embedded trees: Estimation of Gaussian processes on graphs with cycles," Master's thesis, Mass. Inst. Technol., Feb. 2002. [Online] Available: http://ssg.mit.edu/~esuddert/.

[3] R. Szeliski, "Bayesian modeling of uncertainty in low-level vision," *Int. J. Comput. Vision*, vol. 5, no. 3, pp. 271–301, 1990.
[4] M. R. Luettgen, W. C. Karl, and A. S. Willsky, "Efficient multiscale regularization with applications to the computation of optical flow," *IEEE Trans. Image Processing*, vol. 3, pp. 41–64, Jan. 1994.
[5] P. W. Fieguth, W. C. Karl, A. S. Willsky, and C. Wunsch, "Multiresolution optimal interpolation and statistical analysis of TOPEX/POSEIDON satellite altimetry," *IEEE Trans. Geosci. Remote Sens.*, vol. 33, pp. 280–292, Mar. 1995.
[6] T. Kailath, A. H. Sayed, and B. Hassibi, *Linear Estimation*. Englewood Cliffs, NJ: Prentice-Hall, 2000.
[7] M. Jordan, "Graphical models," *Statistical Sci.*, vol. 19, pp. 140–155, 2004.
[8] K. C. Chou, A. S. Willsky, and A. Benveniste, "Multiscale recursive estimation, data fusion, and regularization," *IEEE Trans. Automat. Contr.*, vol. 39, no. 3, pp. 464–478, Mar. 1994.
[9] N. A. C. Cressie, *Statistics for Spatial Data*. New York: Wiley, 1993.
[10] P. Abrahamsen, "A review of Gaussian random fields and correlation functions," Norwegian Computing Center, Oslo, Norway, Tech. Rep. 917, Apr. 1997.
[11] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. New York: Wiley, 1991.
[12] A. S. Willsky, "Multiresolution Markov models for signal and image processing," *Proc. IEEE*, vol. 90, pp. 1396–1458, Aug. 2002.
[13] M. M. Daniel and A. S. Willsky, "A multiresolution methodology for signal-level fusion and data assimilation with applications to remote sensing," *Proc. IEEE*, vol. 85, pp. 164–180, Jan. 1997.
[14] J. Pearl, *Probabilistic Reasoning in Intelligent Systems*. San Mateo, CA: Morgan Kaufman, 1988.
[15] J. S. Yedidia, W. T. Freeman, and Y. Weiss, "Understanding belief propagation and its generalizations," in *Exploring Artificial Intelligence in the New Millennium*, G. Lakemeyer and B. Nebel, Eds. San Mateo, CA: Morgan Kaufmann, 2002.
[16] S. L. Lauritzen, *Graphical Models*. Oxford, U.K.: Oxford Univ. Press, 1996.
[17] A. P. Dempster, "Covariance selection," *Biometrics*, vol. 28, pp. 157–175, Mar. 1972.
[18] T. P. Speed and H. T. Kiiveri, "Gaussian Markov distributions over finite graphs," *Ann. Statist.*, vol. 14, no. 1, pp. 138–150, Mar. 1986.
[19] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inform. Theory*, vol. 47, pp. 498–519, Feb. 2001.
[20] S. M. Aji and R. J. McEliece, "The generalized distributive law," *IEEE Trans. Inform. Theory*, vol. 46, pp. 325–343, Mar. 2000.
[21] Y. Weiss and W. T. Freeman, "Correctness of belief propagation in Gaussian graphical models of arbitrary topology," *Neural Comput.*, vol. 13, pp. 2173–2200, 2001.
[22] K. P. Murphy, Y. Weiss, and M. I. Jordan, "Loopy belief propagation for approximate inference: an empirical study," in *Uncertainty in Artificial Intelligence 15*. San Mateo, CA: Morgan Kaufmann, 1999, pp. 467–475.
[23] P. Rusmevichientong and B. Van Roy, "An analysis of belief propagation on the turbo decoding graph with Gaussian densities," *IEEE Trans. Inform. Theory*, vol. 47, pp. 745–765, Feb. 2001.
[24] J. S. Yedidia, W. T. Freeman, and Y. Weiss, "Constructing free energy approximations and generalized belief propagation algorithms," MERL, Tech. Rep. 2004-040, May 2004.
[25] M. J. Wainwright, T. S. Jaakkola, and A. S. Willsky, "Tree-based reparameterization framework for analysis of sum-product and related algorithms," *IEEE Trans. Inform. Theory*, vol. 49, pp. 1120–1146, May 2003.
[26] M. Welling and Y. W. Teh, "Linear response algorithms for approximate inference in graphical models," *Neural Comput.*, vol. 16, pp. 197–221, 2004.
[27] K. H. Plarre and P. R. Kumar, "Extended message passing algorithm for inference in loopy Gaussian graphical models," *Ad Hoc Networks*, vol. 2, pp. 153–169, 2004.
[28] J. W. Demmel, *Applied Numerical Linear Algebra*. Philadelphia, PA: SIAM, 1997.
[29] D. M. Young, *Iterative Solution of Large Linear Systems*. New York: Academic, 1971.
[30] R. Barrett *et al.*, *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*. Philadelphia, PA: SIAM, 1994.
[31] J. G. Berryman, "Analysis of approximate inverses in tomography II: iterative inverses," *Optimiz. Eng.*, vol. 1, pp. 437–473, 2000.
[32] D. W. Peaceman and H. H. Rachford Jr., "The numerical solution of parabolic and elliptic differential equations," *J. SIAM*, vol. 3, no. 1, pp. 28–41, Mar. 1955.
[33] R. Nikoukhah, A. S. Willsky, and B. C. Levy, "Kalman filtering and Riccati equations for descriptor systems," *IEEE Trans. Automat. Contr.*, vol. 37, no. 9, pp. 1325–1342, Sept. 1992.
[34] L. Adams, "m-Step preconditioned conjugate gradient methods," *SIAM J. Sci. Statist. Comput.*, vol. 6, pp. 452–463, Apr. 1985.
[35] O. Axelsson, "Bounds of eigenvalues of preconditioned matrices," *SIAM J. Matrix Anal. Applicat.*, vol. 13, no. 3, pp. 847–862, July 1992.
[36] M. Bern, J. R. Gilbert, B. Hendrickson, N. Nguyen, and S. Toledo, "Support-graph preconditioners," *SIAM J. Matrix Anal. Applicat.*, Jan. 2001, submitted for publication.
[37] D. Chen and S. Toledo, "Vaidya's preconditioners: implementation and experimental study," *Electron. Trans. Numer. Anal.*, vol. 16, pp. 30–49, 2003.
[38] E. Boman and B. Hendrickson, "Support theory for preconditioning," *SIAM J. Matrix Anal. Applicat.*, vol. 25, no. 3, pp. 694–717, 2003.
[39] E. Boman, D. Chen, B. Hendrickson, and S. Toledo, "Maximum-weight-basis preconditioners," *Numerical Linear Algebra Applicat.*, to be published.

Erik B. Sudderth (S'97) received the B.S. degree (summa cum laude) in electrical engineering from the University of California at San Diego, La Jolla, in 1999 and the M.S. degree in 2002 from the Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, where he is currently pursuing the Ph.D. degree.

His research interests include statistical modeling and machine learning and, in particular, the application of graphical models to problems in computer vision and remote sensing.

Martin J. Wainwright (M'03) received the Ph.D. degree in electical engineering and computer science (EECS) from the Massachusetts Institute of Technology (MIT), Cambridge, in January 2002.

He is currently an assistant professor with the Department of Statistics and the Department of Electrical Engineering and Computer Science, University of California, Berkeley. His research interests include statistical signal and image processing, variational methods and convex optimization, machine learning, and information theory.

Dr. Wainwright received the George M. Sprowls Award in 2002 from the MIT EECS Department for his doctoral dissertation.

Alan S. Willsky (S'70–M'73–SM'82–F'86) joined the faculty of the Massachusetts Institute of Technology (MIT), Cambridge, in 1973 and is currently the Edwin Sibley Webster Professor of Electrical Engineering. He is a founder, member of the Board of Directors, and Chief Scientific Consultant of Alphatech, Inc. From 1998 to 2002, he served as a member of the U.S. Air Force Scientific Advisory Board. He has held visiting positions in England and France. He has delivered numerous keynote addresses and is co-author of the undergraduate text *Signals and Systems* (Englewood Cliffs, NJ: Prentice-Hall, 1996, Second ed.). His research interests are in the development and application of advanced methods of estimation and statistical signal and image processing. Methods he has developed have been successfully applied in a variety of applications including failure detection, surveillance systems, biomedical signal and image processing, and remote sensing.

Dr. Willsky has received several awards, including the 1975 American Automatic Control Council Donald P. Eckman Award, the 1979 ASCE Alfred Noble Prize, and the 1980 IEEE Browder J. Thompson Memorial Award. He has held various leadership positions in the IEEE Control Systems Society (which made him a Distinguished Member in 1988).