Hybrid Generalized Approximate Message Passing with Applications to Structured Sparsity

Sundeep Rangan, Alyson K. Fletcher, Vivek K Goyal, and Philip Schniter

***Abstract*—Gaussian and quadratic approximations of message passing algorithms on graphs have attracted considerable attention due to their computational simplicity, analytic tractability, and wide applicability in optimization and statistical inference problems. This paper summarizes a systematic framework for incorporating such approximate message passing (AMP) methods in general graphical models. The key concept is a partition of dependencies of a general graphical model into strong and weak edges, with each weak edge representing a small, linearizable coupling of variables. AMP approximations based on the central limit theorem can be applied to the weak edges and integrated with standard message passing updates on the strong edges. The resulting algorithm, which we call hybrid generalized approximate message passing (Hybrid-GAMP), can yield significantly simpler implementations of sum-product and max-sum loopy belief propagation. By varying the partition between strong and weak edges, a performance–complexity trade-off can be achieved. Structured sparsity problems are studied as an example of this general methodology where there is a natural partition of edges.**

I. Introduction

Message passing algorithms on graphical models have become widely-used in high-dimensional optimization and inference problems in a range of fields [1], [2]. The fundamental principle of graphical models is to factor high-dimensional problems into sets of problems of lower dimension. The factorization is represented via a graph where the problem variables and factors are represented by the graph vertices, and the dependencies between them represented by edges. Message passing methods such as loopy belief propagation (BP) use this graphical structure to perform approximate inference or optimization in an iterative manner. In each iteration, inference or optimization is performed "locally" on the sub-problems associated with each factor, and "messages" are passed between the variables and factors to account for the coupling between the local problems.

Although effective in a range of problems, loopy BP is only as simple as the problems in the constituent factors; if the factors themselves are of high dimensions, exact implementation of loopy BP will be computationally intractable.

To reduce the complexity of loopy BP, this paper presents a *hybrid generalized approximate message passing* (Hybrid-GAMP) algorithm for what we call *graphical models with linear mixing*. The basic idea is that when factors depend on large numbers of variables, the dependencies are often through aggregates of small, linearizable contributions. In the proposed framework, these weak, linear interactions are identified by partitioning the graph edges into *weak* and *strong* edges, with the dependencies on the weak edges being described by a linear transform. Under the assumption that the components of the linear transform are small, it is argued that the computations for the messages of standard loopy BP along the weak edges can be significantly simplified. Approximate messages along the weak edges are integrated with standard messages on the strong edges.

The Hybrid-GAMP methodology can be applied to any variant of loopy BP, including the sum-product algorithm for inference (e.g., computation of a posterior mean) and the max-sum algorithm for optimization (e.g., computation of a posterior mode). For the sum-product loopy BP algorithm, we show that the messages along the weak edges can be approximated as Gaussian random variables and the computations for these messages can be simplified via the central limit theorem. For max-sum loopy BP, we argue that one can use quadratic approximations of the messages and perform the computations via a simple least-squares solution.

These approximations can dramatically simplify the computations. The complexity of standard loopy BP generically grows exponentially with the maximum degree of the factor nodes. With the GAMP approximation, however, the complexity is exponential only in the maximum degree from the strong edges, while it is linear in the number of weak edges. As a result, Hybrid-GAMP algorithms on a graphical model with linear mixing can remain tractable even with very large numbers of weak, linearizable interactions.

Gaussian and quadratic approximations for message passing algorithms with linear dependencies are not new. The purpose of this paper is to provide a systematic and general framework for these approximations that incorporates and extends many earlier algorithms. Many previous works have considered Gaussian approximations of loopy BP for the problem of estimating vectors with independent components observed through noisy, linear measurements [3]–[9]. In the terminology of this paper, these algorithms apply to graphs where all the non-trivial edges are weak. By enabling graphs that have mixes of both strong and weak edges, the framework of this paper significantly generalizes these methods. For example, instead of the unknown vector simply having independent components, the presence of strong edges can enable the vector to have any prior describable with a graphical model.

S. Rangan (email: srangan@poly.edu) is with Polytechnic Institute of New York University, Brooklyn, NY.

A. K. Fletcher (email: alyson@eecs.berkeley.edu) is with the Department of Electrical Engineering and Computer Sciences, Univ. of California, Berkeley.

V. K. Goyal (email: v.goyal@ieee.org) is with the Research Laboratory of Electronics, Massachusetts Institute of Technology.

P. Schniter (email: schniter@ece.osu.edu) is with the Department of Electrical and Computer Engineering, The Ohio State University.

978-1-4673-2579-0/12/$31.00 ©2012 IEEE

The approach here of combining approximate message passing methods and standard graphical models with linear mixing is closest to the methods developed in [10]–[13] for wavelet image denoising and turbo equalization. These works also considered graphical models that had both linear and nonlinear components, and applied approximate message passing techniques along the lines of [7], [8] to the linearizable portions while maintaining standard BP updates in the remainder of the graph. The use of approximate message passing methods on portions of a factor graph has also been applied with joint parameter estimation and decoding for CDMA multiuser detection in [14]; in a wireless interference coordination problem in [15], and proposed in [16, Section 7] in the context of compressed sensing. The framework presented here unifies and extends all of these examples and thus provides a systematic procedure for incorporating Gaussian approximations of message passing in a modular manner in general graphical models.

The remainder of this paper develops only the sum-product case; the reader is referred to [17] for parallel development of the max-sum case as well as proofs omitted for brevity and more examples and details.

II. Graphical Model Problems with Linear Mixing

Let $\mathbf{x}$ and $\mathbf{z}$ be real-valued block column vectors

$$\mathbf{x} = (\mathbf{x}_1^*, \ldots, \mathbf{x}_n^*)^*, \qquad \mathbf{z} = (\mathbf{z}_1^*, \ldots, \mathbf{z}_m^*)^*, \tag{1}$$

and consider a function of these vectors of the form

$$F(\mathbf{x}, \mathbf{z}) := \textstyle\sum_{i=1}^m f_i(\mathbf{x}_{\alpha(i)}, \mathbf{z}_i), \tag{2}$$

where, for each i, $f_i(\cdot)$ is a real-valued function; $\alpha(i)$ is a subset of the indices $\{1, \ldots, n\}$; and $\mathbf{x}_{\alpha(i)}$ is the concatenation of the vectors $\{\mathbf{x}_j,\, j \in \alpha(i)\}$. We are interested in computations subject to linear constraints of the form

$$\mathbf{z}_i = \textstyle\sum_{j=1}^n \mathbf{A}_{ij}\mathbf{x}_j = \mathbf{A}_i\mathbf{x}, \tag{3}$$

where each $\mathbf{A}_{ij}$ is a real-valued matrix and $\mathbf{A}_i$ is the block column matrix with components $\mathbf{A}_{ij}$. We will also let $\mathbf{A}$ be the block matrix with components $\mathbf{A}_{ij}$ so that we can write the linear constraints as $\mathbf{z} = \mathbf{A}\mathbf{x}$.

The function $F(\mathbf{x}, \mathbf{z})$ is naturally described via a graphical model as shown in Fig. 1. Specifically, we associate with $F(\mathbf{x}, \mathbf{z})$ a bipartite *factor graph* $G = (V, E)$ whose vertices V consist of n *variable nodes* corresponding to the (vector-valued) variables $\mathbf{x}_j$, and m *factor nodes* corresponding to the factors $f_i(\cdot)$ in (2). There is an edge $(i, j) \in E$ in the graph if and only if the variable $\mathbf{x}_j$ has some influence on the factor $f_i(\mathbf{x}_{\alpha(i)}, \mathbf{z}_i)$. This influence can occur in one of two mutually exclusive ways:

- The index j is in $\alpha(i)$, so that the variable $\mathbf{x}_j$ directly appears in the sub-vector $\mathbf{x}_{\alpha(i)}$ in the factor $f_i(\mathbf{x}_{\alpha(i)}, \mathbf{z}_i)$. In this case, (i, j) will be called a *strong edge*, since $\mathbf{x}_j$ can have an arbitrary and potentially-large influence on the factor.
- The matrix $\mathbf{A}_{ij}$ is nonzero, so $\mathbf{x}_j$ affects $f_i(\mathbf{x}_{\alpha(i)}, \mathbf{z}_i)$ through its linear influence on $\mathbf{z}_i$ in (3). In this case, (i, j) will be called a *weak edge*, since the approximations we will make in the algorithms below assume that $\mathbf{A}_{ij}$ is small. The set of weak edges into the factor node i will be denoted $\beta(i)$.

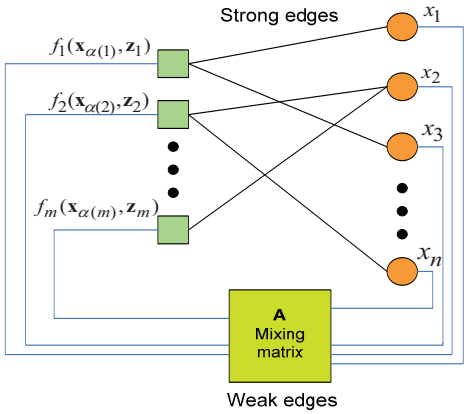

Fig. 1. Factor graph representation of the linear mixing estimation and optimization problems. The variable nodes (circles) are connected to the factor nodes (squares) either directly (strong edges) or via the output of the linear mixing matrix $\mathbf{A}$ (weak edges).

Together $\alpha(i)$ and $\beta(i)$ comprise the set of all indices j such that the variable node $\mathbf{x}_j$ is connected to the factor node $f_i(\cdot)$ in the graph G. The union $\partial(i) = \alpha(i) \cup \beta(i)$ is thus the neighbor set of $f_i(\cdot)$. Similarly, for any variable node $\mathbf{x}_j$, we let $\alpha(j)$ be the set of indices i such that that the factor node $f_i(\cdot)$ is connected to $\mathbf{x}_j$ via a strong edge, and let $\beta(j)$ be the set of indices i such that there is a weak edge. We let $\partial(j) = \alpha(j) \cup \beta(j)$ be the union of these sets, which is the neighbor set of $\mathbf{x}_j$.

Given these definitions, we are interested in the **Expectation problem P-EXP**: Given a function $F(\mathbf{x}, \mathbf{z})$ of the form (2), a matrix $\mathbf{A}$, and *scale factor* $u > 0$, define the joint distribution

$$p(\mathbf{x}) := (1/Z(u)) \exp\left[uF(\mathbf{x}, \mathbf{z})\right], \qquad \mathbf{z} = \mathbf{A}\mathbf{x}, \tag{4}$$

where $Z(u)$ is a normalization constant called the partition function (it is a function of u). For this distribution, compute the expectations

$$\widehat{\mathbf{x}} = \mathbb{E}[\mathbf{x}], \qquad \widehat{\mathbf{z}} = \mathbb{E}[\mathbf{z}]. \tag{5}$$

Also, for each j, compute the log marginal

$$\Delta_j(\mathbf{x}_j) := (1/u) \log \textstyle\int \exp\left[uF(\mathbf{x}, \mathbf{z})\right] d\mathbf{x}_{\setminus j}, \tag{6}$$

where the integral is over all variables $\mathbf{x}_r$ for $r \neq j$.

P-EXP arises naturally in statistical inference: Suppose we are given a probability distribution $p(\mathbf{x})$ of the form (4) for some function $F(\mathbf{x}, \mathbf{z})$. The function $F(\mathbf{x}, \mathbf{z})$ may depend implicitly on some observed vector $\mathbf{y}$, so that $p(\mathbf{x})$ represents the posterior distribution of $\mathbf{x}$ given $\mathbf{y}$. In this context, the solution $(\widehat{\mathbf{x}}, \widehat{\mathbf{z}})$ to the problem P-EXP is precisely the *minimum mean squared error* (MMSE) estimate when $u = 1$. The function $\Delta_j(\mathbf{x}_j)$ is the log marginal distribution of $\mathbf{x}_j$.

In the analysis below, we will assume that, for all factor nodes $f_i(\cdot)$, the strong and weak neighbors, $\alpha(i)$ and $\beta(i)$, are disjoint. This assumption introduces no loss of generality: If an edge (i,j) is both weak and strong, we can modify the function $f_i(\mathbf{x}_\alpha(i), \mathbf{z}_i)$ to move the influence of $\mathbf{x}_j$ from the term $\mathbf{z}_i$ into the direct term $\mathbf{x}_{\alpha(i)}$ [17].

Even when the dependence of a factor $f_i(\mathbf{x}_{\alpha(i)}, \mathbf{z}_i)$ on a variable $\mathbf{x}_j$ is only through the linear term $\mathbf{z}_i$, we may still wish to move the dependence to a strong edge to improve accuracy at the expense of greater computation.

Since $\mathbf{A}_{ij} \neq 0$ only when $j \in \beta(i)$, we may sometimes write the summation (3) as

$$\mathbf{z}_i = \textstyle\sum_{j\in\beta(i)} \mathbf{A}_{ij}\mathbf{x}_j = \mathbf{A}_{i,\beta(i)}\mathbf{x}_{\beta(i)}, \tag{7}$$

where $\mathbf{x}_{\beta(i)}$ is the sub-vector of $\mathbf{x}$ with components $j \in \beta(i)$ and $\mathbf{A}_{i,\beta(i)}$ is the corresponding portion of the ith block-row of $\mathbf{A}$.

III. Review of Loopy Belief Propagation

The sum-product loopy BP algorithm is based on iteratively passing estimates of the log marginals $\Delta_j(\mathbf{x}_j)$ in (6). We index the iterations by $t = 0, 1, \ldots$, and denote the estimate "message" from the factor node f_i to the variable node $\mathbf{x}_j$ in the tth iteration by $\Delta_{i\to j}(t, \mathbf{x}_j)$ and the reverse message by $\Delta_{i\leftarrow j}(t, \mathbf{x}_j)$. The messages in loopy BP are equivalent up to a constant factor. That is, adding any constant term that does not depend on $\mathbf{x}_j$ to either the message $\Delta_{i\to j}(t, \mathbf{x}_j)$ or $\Delta_{i\leftarrow j}(t, \mathbf{x}_j)$ has no effect on the algorithm. We will thus use the notation

$$\Delta(\mathbf{x}) \equiv g(\mathbf{x}) \quad \Longleftrightarrow \quad \Delta(\mathbf{x}) = g(\mathbf{x}) + C,$$

for some constant C that does not depend on $\mathbf{x}$. Similarly, we write $p(\mathbf{x}) \propto q(\mathbf{x})$ when $p(\mathbf{x}) = Cq(\mathbf{x})$ for some constant C. We fix the scale factor $u > 0$ and, for any function $\Delta(\cdot)$, we will write $\mathbb{E}[g(\mathbf{x})\,;\,\Delta(\cdot)]$ to denote the expectation of $g(\mathbf{x})$ with respect to a distribution specified indirectly by $\Delta(\cdot)$:

$$\mathbb{E}[g(\mathbf{x})\,;\,\Delta(\cdot)] = \textstyle\int g(\mathbf{x})p(\mathbf{x})\,d\mathbf{x}, \tag{8}$$

where $p(\mathbf{x})$ is the probability distribution

$$p(\mathbf{x}) = (1/Z(u))\exp\left[u\Delta(\mathbf{x})\right]$$

and $Z(u)$ is a normalization constant.

For each edge $(i,j) \in E$, the factor node update is a computation of $\Delta_{i\to j}(t, \mathbf{x}_j)$ by integration over variables $\mathbf{x}_r$ with $r \in \partial(i)$ and $r \neq j$. The variable node update is a computation of $\Delta_{i\leftarrow j}(t+1, \mathbf{x}_j)$ by combining influences of $\Delta_{\ell\to j}(t, \mathbf{x}_j)$ for $\ell \in \partial(j) \setminus i$ followed by computation of a point estimate $\widehat{\mathbf{x}}_j(t+1)$ as a scalar expectation.

When the graph G is acyclic, the sum-product loopy BP algorithm converges to the exact solution of P-EXP. For graphs with cycles, the loopy BP algorithm is, in general, only approximate; see, for example, [2], [18], [19].

Brute force solutions to P-EXP involve an expectation over *all* n variables $\mathbf{x}_j$. Loopy BP reduces this "global" problem to a sequence of "local" problems associated with each of the factors $f_i(\cdot)$. The local expectation problems may be significantly lower in dimension than the global problem. In particular, the factor $f_i(\mathbf{x}_{\alpha(i)}, \mathbf{z}_i)$ is a function of $d_i = |\partial(i)|$ variables $\mathbf{x}_j$, either through one of the $|\alpha(i)|$ strong edges or $|\beta(i)|$ weak edges. For each $j \in \partial(i)$, the factor node update involves an integration over $d_i - 1$ variables. The complexity in general grows exponentially in d_i. Thus, standard loopy BP is only typically tractable when the degrees d_i of the factor nodes are small or the factors have some convenient form.

IV. Hybrid-GAMP

The Hybrid-GAMP algorithm reduces the cost of loopy BP by exploiting complexity-reducing approximations of the cumulative effect of the weak edges. We saw in the previous section that the loopy BP update at each factor node $f_i(\cdot)$ has a cost that may be exponential in the degree d of the node, which consists of $|\alpha(i)|$ strong edges and $|\beta(i)|$ weak edges. The Hybrid-GAMP algorithm with edge partitioning uses the linear mixing property to eliminate the exponential dependence on the $|\beta(i)|$ weak edges, so the only exponential dependence is on the $|\alpha(i)|$ strong edges. Thus, the edge partitioning makes Hybrid-GAMP computationally tractable as long as the number of strong edges is small. The number of weak edges can be arbitrary. In particular, the mixing matrix $\mathbf{A}$ can be dense.

The basis of the Hybrid-GAMP approximation is to assume that the matrix $\mathbf{A}_{ij}$ is small along any weak edge (i,j). Under this assumption, one can apply a Gaussian approximation of the weak edge messages and use the central limit theorem at the factor nodes. A heuristic derivation of the Hybrid-GAMP approximations is given in [17, App. B].

We need additional notation: The Hybrid-GAMP algorithm produces a sequence of estimates $\widehat{\mathbf{x}}_j(t)$ and $\widehat{\mathbf{z}}_i(t)$ for the variables $\mathbf{x}_j$ and $\mathbf{z}_i$. Several other intermediate variables $\widehat{\mathbf{p}}_i(t)$, $\widehat{\mathbf{s}}_i(t)$ and $\widehat{\mathbf{r}}_j(t)$ are also produced. Associated with each of the variables are matrices $\mathbf{Q}_j^x(t)$, $\mathbf{Q}_i^z(t), \ldots$, that represent certain covariances. When we need to take the inverse of the matrices, we will use the notation $\mathbf{Q}_j^{-x}(t)$ to mean $(\mathbf{Q}_j^x(t))^{-1}$. Finally, for any positive definite matrix $\mathbf{Q}$ and vector $\mathbf{a}$, we will let $\|\mathbf{a}\|_{\mathbf{Q}}^2 = \mathbf{a}^*\mathbf{Q}^{-1}\mathbf{a}$, which is a weighted two norm.

Algorithm 1: **Hybrid-GAMP:** Consider the problem P-EXP for some function $F(\mathbf{x}, \mathbf{z})$ of the form (2), matrix $\mathbf{A}$, and scale factor $u > 0$.

1) *Initialization:* Set $t = 0$ and select some initial values $\Delta_{i\to j}(t-1, \mathbf{x}_j)$ for all strong edges (i,j) and values $\widehat{r}_j(t-1)$ and $\mathbf{Q}_j^r(t-1)$ for all variable node indices j.
2) *Variable node update, strong edges:* For all strong edges (i,j), compute

$$\Delta_{i\leftarrow j}(t, \mathbf{x}_j) \equiv \textstyle\sum_{\ell\in\alpha(j)\neq i} \Delta_{\ell\to j}(t-1, \mathbf{x}_j) - \frac{1}{2}\|\widehat{\mathbf{r}}_j(t-1) - \mathbf{x}_j\|^2_{\mathbf{Q}_j^r(t-1)}. \tag{9}$$

3) *Variable node update, weak edges:* For all variable nodes j, compute

$$\Delta_j(t, \mathbf{x}_j) \equiv H_j^x(t, \mathbf{x}_j, \widehat{\mathbf{r}}_j(t-1), \mathbf{Q}_j^r(t-1)), \tag{10}$$

$$H_j^x(t,\mathbf{x}_j,\widehat{\mathbf{r}}_j,\mathbf{Q}_j^r) = \sum_{i\in\alpha(j)}\Delta_{i\to j}(t-1,\mathbf{x}_j) - \tfrac{1}{2}\|\widehat{\mathbf{r}}_j-\mathbf{x}_j\|^2_{\mathbf{Q}_j^r}, \quad (11)$$

$$\widehat{\mathbf{x}}_j(t) = \mathbb{E}\left(\mathbf{x}_j\,;\,\Delta_j(t,\cdot)\right), \quad (12)$$

$$\mathbf{Q}_j^x(t) = u\,\mathbf{var}\left(\mathbf{x}_j\,;\,\Delta_j(t,\cdot)\right). \quad (13)$$

4) *Factor node update, linear step:* For all factor nodes i, compute

$$\widehat{\mathbf{z}}_i(t) = \sum_{j\in\beta(i)}\mathbf{A}_{ij}\widehat{\mathbf{x}}_j(t), \quad (14a)$$

$$\widehat{\mathbf{p}}_i(t) = \widehat{\mathbf{z}}_i(t) - \mathbf{Q}_i^p(t)\widehat{\mathbf{s}}_i(t-1), \quad (14b)$$

$$\mathbf{Q}_i^p(t) = \sum_{j\in\beta(i)}\mathbf{A}_{ij}\mathbf{Q}_j^x(t)\mathbf{A}_{ij}^*, \quad (14c)$$

where, initially, we set $\widehat{\mathbf{s}}_i(-1)=0$.

5) *Factor node update, strong edges:* For all strong edges (i,j), compute:

$$\Delta_{i\to j}(t,\mathbf{x}_j) \equiv \tfrac{1}{u}\log\int p_{i\to j}(t,\mathbf{x}_{\alpha(i)},\mathbf{z}_i)d\mathbf{x}_{\alpha(i)\setminus j}d\mathbf{z}_i \quad (15)$$

where the integral is over $\mathbf{z}_i$ and all components $\mathbf{x}_r$ with $r\in\alpha(i)\setminus j$, and $p_{i\to j}(0,\mathbf{x}_j)$ is the probability distribution function

$$p_{i\to j}(t,\mathbf{x}_{\alpha(i)},\mathbf{z}_i) \propto \exp\left(uH^z_{i\to j}(t,\mathbf{x}_{\alpha(i)}.\mathbf{z}_i,\widehat{\mathbf{p}}_i(t),\mathbf{Q}_i^p(t))\right). \quad (16)$$

6) *Factor node update, weak edges:* For all factor nodes i, compute

$$H_i^z(t,\mathbf{x}_{\alpha(i)},\mathbf{z}_i,\widehat{\mathbf{p}}_i,\mathbf{Q}_i^p) := f_i(\mathbf{x}_{\alpha(i)},\mathbf{z}_i) + \sum_{r\in\alpha(i)}\Delta_{i\leftarrow r}(t,\mathbf{x}_r) - \tfrac{1}{2}\|\mathbf{z}_i-\widehat{\mathbf{p}}_i\|^2_{\mathbf{Q}_i^p}. \quad (17)$$

Then, let

$$\widehat{\mathbf{z}}_i^0(t)=\mathbb{E}(\mathbf{z}_i), \qquad \mathbf{Q}_i^z(t)=u\,\mathbf{var}(\mathbf{z}_i), \quad (18)$$

where $\mathbf{z}_i$ is the component of the pair $(\mathbf{x}_{\alpha(i)},\mathbf{z}_i)$ with the joint distribution

$$p_i(t,\mathbf{x}_{\alpha(i)},\mathbf{z}_i) \propto \exp\left(uH_i^z(t,\mathbf{x}_{\alpha(i)},\mathbf{z}_i,\widehat{\mathbf{p}}_i(t),\mathbf{Q}_i^p(t))\right). \quad (19)$$

Then, compute

$$\widehat{\mathbf{s}}_i(t) = \mathbf{Q}_i^{-p}(t)\left[\widehat{\mathbf{z}}_i^0(t)-\widehat{\mathbf{p}}_i(t)\right], \quad (20a)$$

$$\mathbf{Q}_i^s(t) = \mathbf{Q}_i^{-p}(t) - \mathbf{Q}_i^{-p}(t)\mathbf{D}_i^{-z}(t)\mathbf{Q}_i^{-p}(t). \quad (20b)$$

7) *Variable node update, linear step:* For all variables nodes j compute

$$\mathbf{Q}_j^{-r}(t) = \sum_{i\in\beta(j)}\mathbf{A}_{ij}^*\mathbf{Q}_i^s(t)\mathbf{A}_{ij}, \quad (21a)$$

$$\widehat{\mathbf{r}}_j(t) = \widehat{\mathbf{x}}(t) + \mathbf{Q}_j^r(t)\sum_{i\in\beta(j)}\mathbf{A}_{ij}^*\widehat{\mathbf{s}}_i(t). \quad (21b)$$

Increment t and return to step 2 for some number of iterations.

Although the Hybrid-GAMP algorithm above appears more complicated than standard loopy BP, Hybrid-GAMP can be computationally dramatically cheaper. The main computational difficulty of loopy BP is the factor update; this involves an expectation over $|\partial(i)|$ variables, where $\partial(i)$ is the set of all variables connected to the factor node i. In the Hybrid-GAMP algorithm, these computations are replaced by (15), where the expectation is over the strong edge variables $\alpha(i)$. If the number of edges is large, the computational savings can be dramatic. The other steps of the Hybrid-GAMP algorithms are all linear least-squares operations, or componentwise nonlinear functions on the individual variables.

For illustration, we have only presented one form of the Hybrid-GAMP procedure. Variants with discrete distributions and other message scheduling are possible.

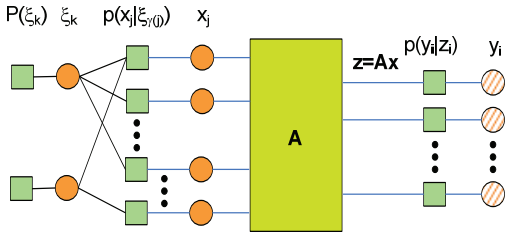

Fig. 2. Graphical model for the group sparsity problem with overlapping groups. The group dependencies between components of the vector $\mathbf{x}$ are modeled via a set of binary latent variables $\boldsymbol{\xi}$.

Method	Complexity
Group-OMP [24]	$O(\rho mn^2)$
Group-Lasso [20], [21], [25]	$O(mn)$ per iteration
Relaxed BP with vector components [26]	$O(mn^2)$ per iteration
Hybrid-GAMP with vector components	$O(mnd)$ per iteration
Hybrid-GAMP with scalar components	$O(mn)$ per iteration

TABLE I
COMPLEXITY COMPARISON FOR GROUP SPARSITY ESTIMATION OF A SPARSE VECTOR WITH K GROUPS, EACH GROUP OF DIMENSION d. THE NUMBER OF MEASUREMENTS IS m AND THE SPARSITY RATIO IS ρ.

V. Application to Structured Sparsity

Hybrid-GAMP is a flexible and general methodology. To enable comparisons against existing algorithms, we consider the group sparse estimation problem [20], [21]. This is a highly-structured problem in which dependencies beyond linear mixing are captured by binary latent variables ξ_k as shown in Fig. 2. The setting and the specialization of Algorithm 1 to this case are described in detail in [17].

In addition to its generality, the Hybrid-GAMP procedure is among the most computationally efficient. Consider the special case when there are K non-overlapping groups of d elements each; that is, each ξ_k is connected to d variables x_j, and the sets are disjoint. In this case, the total vector dimension for $\mathbf{x}$ is $n=Kd$. We consider the non-overlapping case since there are many algorithms that apply to this case that we can compare against. For non-overlapping uniform groups, Table I compares the computational cost of the Hybrid-GAMP algorithm to other methods. Because of the block separability of this special case, the analysis in [22] applies. For $d=2$, correlated variables can be considered the real and imaginary parts of a complex variable, enabling the methods of [23].

Of course, a complete comparison requires that we consider the number of iterations, not just the computation per iteration.

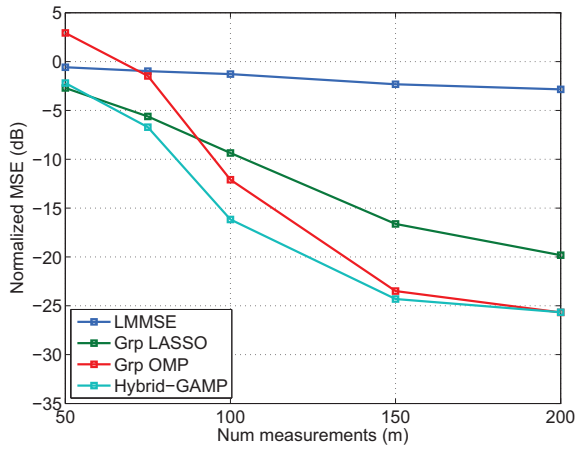

Fig. 3. Comparison of performances of various estimation algorithms for group sparsity with $n = 100$ groups of dimension $d = 4$ with a sparsity fraction of $\rho = 0.1$.

This comparison requires further study beyond the scope of this paper. However, it is possible that the Hybrid-GAMP procedure will be favorable in this regard. Our simulations below show good convergence after only 10–20 iterations.

Fig. 3 shows the results of a simple simulation comparison of algorithms. The simulation used a vector $\mathbf{x}$ with $n = 100$ groups of size $d = 4$ and i.i.d. Bernoulli variables ξ_k with $\rho = \Pr(\xi_k = 1) = 0.1$. The matrix was i.i.d. Gaussian and the observations were with AWGN noise at an SNR of 20 dB. The number of measurements m was varied from 50 to 200, and the plot shows the MSE for each of the methods. The Hybrid-GAMP method was run with 20 iterations. In group LASSO, at each value of m, the algorithm was simulated with several values of the regularization parameter and the plot shows the minimum MSE. In Group-OMP, the algorithm was run with the true value of the number of nonzero coefficients. It can be seen that the Hybrid-GAMP method is consistently as good or better than the other methods. All code for the simulations can be found in the SourceForge open website [27].

VI. Conclusions

A general model for optimization and statistical inference based on graphical models with linear mixing was presented. The linear mixing components of the graphical model account for interactions through aggregates of large numbers of small, linearizable contributions. Gaussian and second-order approximations are shown to greatly simplify the implementation of loopy BP for these interactions, and the Hybrid-GAMP framework presented here enables systematic incorporation of these approximations in general graphical models. Simulations were presented for group sparsity where the Hybrid-GAMP method has equal or superior performance to existing methods. However, the generality of the method will enable Hybrid-GAMP to be applied to much more complex models where few algorithms are available. In addition to experimenting with such models, future work will focus on establishing rigorous theoretical analyses along the lines of [9], [28].

[1] B. J. Frey, *Graphical Models for Machine Learning and Digital Communication*. MIT Press, 1998.

[2] M. J. Wainwright and M. I. Jordan, *Graphical Models, Exponential Families, and Variational Inference*, ser. Foundations and Trends in Machine Learning. NOW Publishers, 2008, vol. 1.

[3] J. Boutros and G. Caire, "Iterative multiuser joint decoding: Unified framework and asymptotic analysis," *IEEE Trans. Inform. Theory*, vol. 48, no. 7, pp. 1772–1793, Jul. 2002.

[4] T. Tanaka and M. Okada, "Approximate belief propagation, density evolution, and neurodynamics for CDMA multiuser detection," *IEEE Trans. Inform. Theory*, vol. 51, no. 2, pp. 700–706, Feb. 2005.

[5] D. Guo and C.-C. Wang, "Asymptotic mean-square optimality of belief propagation for sparse linear systems," in *Proc. IEEE Inform. Theory Workshop*, Chengdu, China, Oct. 2006, pp. 194–198.

[6] ——, "Random sparse linear systems observed via arbitrary channels: A decoupling principle," in *Proc. IEEE ISIT*, Jun. 2007, pp. 946–950.

[7] S. Rangan, "Estimation with random linear mixing, belief propagation and compressed sensing," arXiv:1001.2228v1 [cs.IT]., Jan. 2010.

[8] D. L. Donoho, A. Maleki, and A. Montanari, "Message-passing algorithms for compressed sensing," *Proc. Nat. Acad. Sci.*, vol. 106, no. 45, pp. 18 914–18 919, Nov. 2009.

[9] M. Bayati and A. Montanari, "The dynamics of message passing on dense graphs, with applications to compressed sensing," *IEEE Trans. Inform. Theory*, vol. 57, no. 2, pp. 764–785, Feb. 2011.

[10] P. Schniter, "Turbo reconstruction of structured sparse signals," in *Proc. Conf. on Inform. Sci. & Sys.*, Mar. 2010.

[11] J. Ziniel, L. C. Potter, and P. Schniter, "Tracking and smoothing of time-varying sparse signals via approximate belief propagation," in *Conf. Rec. Asilomar Conf. on Signals, Syst. & Computers*, Nov. 2010.

[12] S. Som, L. C. Potter, and P. Schniter, "Compressive imaging using approximate message passing and a Markov-tree prior," in *Conf. Rec. Asilomar Conf. on Signals, Syst. & Computers*, Nov. 2010.

[13] P. Schniter, "A message-passing receiver for BICM-OFDM over unknown clustered-sparse channels," in *Proc. IEEE Workshop Signal Process. Adv. Wireless Commun.*, Jun. 2011.

[14] G. Caire, A. Tulino, and E. Biglieri, "Iterative multiuser joint detection and parameter estimation: a factor-graph approach," in *Proc. IEEE Inform. Theory Workshop*, Sep. 2001, pp. 36–38.

[15] S. Rangan and R. K. Madan, "Belief propagation methods for intercell interference coordination," in *Proc. IEEE Infocom*, Apr. 2011.

[16] A. Montanari, "Graphical models concepts in compressed sensing," arXiv:1011.4328v3 [cs.IT], Mar. 2011.

[17] S. Rangan, A. K. Fletcher, V. K. Goyal, and P. Schniter, "Hybrid approximate message passing with applications to structured sparsity," arXiv:1111.2581 [cs.IT], Nov. 2011.

[18] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988.

[19] J. S. Yedidia, W. T. Freeman, and Y. Weiss, "Understanding belief propagation and its generalizations," in *Exploring Artificial Intelligence in the New Millennium*. Morgan Kaufmann, 2003, pp. 239–269.

[20] M. Yuan and Y. Lin, "Model selection and estimation in regression with grouped variables," *J. Royal Statist. Soc.*, vol. 68, pp. 49–67, 2006.

[21] P. Zhao, G. Rocha, and B. Yu, "The composite absolute penalties family for grouped and hierarchical variable selection," *Ann. Stat.*, vol. 37, no. 6, pp. 3468–3497, 2009.

[22] D. Donoho, I. Johnstone, and A. Montanari, "Accurate prediction of phase transitions in compressed sensing via a connection to minimax denoising," arXiv:1111.1041v1 [cs.IT]., Nov. 2011.

[23] A. Maleki, L. Anitori, Z. Yang, and R. G. Baraniuk, "Asymptotic analysis of complex LASSO via complex approximate message passing (CAMP)," arXiv:1108.0477v1 [cs.IT], Aug. 2011.

[24] A. C. Lozano, G. Świrszcz, and N. Abe, "Group orthogonal matching pursuit for variable selection and prediction," in *Proc. Neural Information Process. Syst.*, Dec. 2008.

[25] S. J. Wright, R. D. Nowak, and M. Figueiredo, "Sparse reconstruction by separable approximation," *IEEE Trans. Signal Process.*, vol. 57, no. 7, pp. 2479–2493, Jul. 2009.

[26] J. Kim, W. Chang, B. Jung, D. Baron, and J. C. Ye, "Belief propagation for joint sparse recovery," arXiv:1102.3289, Feb. 2011.

[27] S. Rangan *et al.*, "Generalized approximate message passing," SourceForge.net project gampmatlab, available on-line at `http://gampmatlab.sourceforge.net/`.

[28] S. Rangan, "Generalized approximate message passing for estimation with random linear mixing," arXiv:1010.5141v1 [cs.IT]., Oct. 2010.