Generalized Approximate Message Passing for Estimation with Random Linear Mixing

Sundeep Rangan, *Member, IEEE*

arXiv:1010.5141v2 [cs.IT] 13 Aug 2012

Abstract—We consider the estimation of an i.i.d. random vector observed through a linear transform followed by a component-wise, probabilistic (possibly nonlinear) measurement channel. A novel algorithm, called generalized approximate message passing (GAMP), is presented that provides computationally efficient approximate implementations of max-sum and sum-problem loopy belief propagation for such problems. The algorithm extends earlier approximate message passing methods to incorporate arbitrary distributions on both the input and output of the transform and can be applied to a wide range of problems in nonlinear compressed sensing and learning.

Extending an analysis by Bayati and Montanari, we argue that the asymptotic componentwise behavior of the GAMP method under large, i.i.d. Gaussian transforms is described by a simple set of state evolution (SE) equations. From the SE equations, one can *exactly* predict the asymptotic value of virtually any componentwise performance metric including mean-squared error or detection accuracy. Moreover, the analysis is valid for arbitrary input and output distributions, even when the corresponding optimization problems are non-convex. The results match predictions by Guo and Wang for relaxed belief propagation on large sparse matrices and, in certain instances, also agree with the optimal performance predicted by the replica method. The GAMP methodology thus provides a computationally efficient methodology, applicable to a large class of non-Gaussian estimation problems with precise asymptotic performance guarantees.

Index Terms—Optimization, random matrices, estimation, belief propagation, graphical models, compressed sensing.

I. INTRODUCTION

The problem of estimating vectors from linear transforms followed by random, possibly nonlinear, measurements, arises in a range of problems in signal processing, communications, and learning. This paper considers a general class of such estimation problems in a Bayesian setting shown in Fig. 1. An input vector $\mathbf{q} \in Q^n$ has components $q_j \in Q$ for some set Q and generates an unknown random vector $\mathbf{x} \in \mathbb{R}^n$ through a componentwise input channel described by a conditional distribution $p_{X|Q}(x_j|q_j)$. The vector $\mathbf{x}$ is then passed through a linear transform

$$\mathbf{z} = \mathbf{A}\mathbf{x}, \qquad (1)$$

where $\mathbf{A} \in \mathbb{R}^{m \times n}$ is a known transform matrix. Finally, each component of z_i of $\mathbf{z}$ randomly generates an output component y_i of a vector $\mathbf{y} \in Y^m$ through a second scalar conditional distribution $p_{Y|Z}(y_i|z_i)$, where Y is some output set. The problem is to estimate the transform input $\mathbf{x}$ and output $\mathbf{z}$

from the system input vector $\mathbf{q}$, system output $\mathbf{y}$ and linear transform $\mathbf{A}$.

The formulation is general and has wide applicability – we will present several applications in Section II. However, for many non-Gaussian instances of the problem, exact computations of quantities such as the posterior mode or mean of $\mathbf{x}$ is computationally prohibitive. The primary difficulty is that the matrix $\mathbf{A}$ "couples" or "mixes" the coefficients of $\mathbf{x}$ into $\mathbf{z}$. If the transform matrix were the identity matrix (i.e. $m = n$ and $\mathbf{A} = I$), then the estimation problem would *decouple* into $m = n$ scalar estimation problems, each defined by the Markov chain:

$$q_j \sim p_Q(q_j) \xrightarrow{p_{X|Q}(x_j|q_j)} x_j = z_j \xrightarrow{p_{Y|Z}(y_j|z_j)} y_j.$$

Since all the quantities in this Markov chain are scalars, one could numerically compute the exact posterior distribution on any component $x_j(= z_j)$ from q_j and y_j with one-dimensional integrals. However, for a general (i.e. non-identity) matrix $\mathbf{A}$, the components of $\mathbf{x}$ are coupled into the vector $\mathbf{z}$. In this case, the posterior distribution of any particular component x_j or z_i would involve a high-dimensional integral that is, in general, difficult to evaluate.

This paper presents a novel algorithm, *generalized approximate message passing* or GAMP, that *decouples* the vector-valued estimation problem into a sequence of scalar problems and linear transforms. The GAMP algorithm is an extension of several earlier Gaussian and quadratic approximations of loopy belief propagation (loopy BP) that have been successful in many previous applications, including, most recently, compressing sensing [1]–[9] (See the "Previous Works" subsection below). The proposed methodology extends these methods to provide several valuable features:

- *Generality:* Most importantly, the GAMP methodology applies to essentially arbitrary priors $P_{X|Q}$ and output channel distributions $P_{Y|Z}$. The algorithm can thus incorporate arbitrary non-Gaussian inputs as well as output nonlinearities. Moreover, the algorithm can be used to realize approximations of both max-sum loopy BP for *maximum a posteriori* (MAP) estimation and sum-product loopy BP for minimum mean-squared error (MMSE) estimates and approximations of the posterior marginals.
- *Computational simplicity:* The algorithm is computationally simple with each iteration involving only scalar estimation computations on the components of $\mathbf{x}$ and $\mathbf{z}$ along with linear transforms. Indeed, the iterations are similar in form to the fastest known methods for such

problems [10]–[14], while being potentially more general. Moreover, our simulations indicate excellent convergence in a small number, typically 10 to 20, iterations.

- *Analytic tractability:* Our main theoretical result, Claim 1, shows that for large Gaussian i.i.d. transforms, $\mathbf{A}$, the asymptotic behavior of the components of the GAMP algorithm are described exactly by a simple *scalar equivalent model*. The parameters in this model can be computed by a set of scalar *state evolution* (SE) equations, analogous to the density evolution equations for BP decoding of LDPC codes [15], [16]. From this scalar equivalent model, one can asymptotically *exactly* predict any componentwise performance metric such as mean-squared error (MSE) or detection accuracy. Moreover, the SE analysis generalizes results on earlier approximate message passing-like algorithms [1]–[9], and also, in certain instances, show a match with the optimal performance as predicted by the replica method in statistical physics [8], [17]–[21].

The GAMP algorithm thus provides a general and systematic approach to a large class of estimation problems with linear mixing that is computationally tractable and widely-applicable. Moreover, in the case of large random $\mathbf{A}$, the method admits exact asymptotic performance characterizations.

A. Prior Work

The GAMP algorithm belongs to a long line of methods based on Gaussian and quadratic approximations of loopy belief propagation (loopy BP). Loopy BP is a general methodology for estimation problems where the statistical relationships between variables are represented via a graphical model. The loopy BP algorithm iteratively updates estimates of the variables via a message passing procedure along the graph edges [22], [23]. For the linear mixing estimation problem, the graph is precisely the incidence matrix of the matrix $\mathbf{A}$ and the loopy BP messages are passed between nodes corresponding to the input variables x_j and outputs variables z_i.

The GAMP algorithm provides approximations of two important variants of loopy BP:

- *Max-sum* loopy BP for approximately computing MAP estimates of $\mathbf{x}$ and $\mathbf{z}$ as well as sensitivities of the maximum of the posterior distribution to the individual components of $\mathbf{x}$ and $\mathbf{z}$; and
- *Sum-product* loopy BP for approximations of the minimum mean-squared error (MMSE) estimates of $\mathbf{x}$ and $\mathbf{z}$ and the marginal posteriors of their individual components.

We will call the GAMP approximations of the two algorithms, max-sum GAMP and sum-product GAMP, respectively.

In communications and signal processing, BP is best known for its connections to iterative decoding in turbo and LDPC codes [24]–[26]. However, while turbo and LDPC codes typically involve computations over finite fields, BP has also been successfully applied in a number of problems with linear real-valued mixing, including CDMA multiuser detection [1],
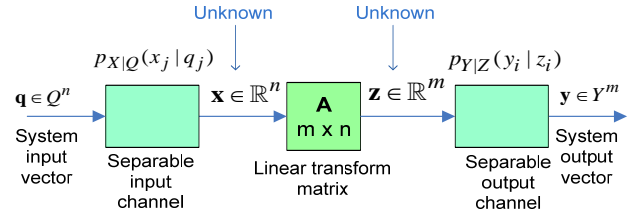

Fig. 1. General estimation problem with linear mixing. The problem is to estimate the transform input and output vectors $\mathbf{x}$ and $\mathbf{z}$ from the system input and output vectors $\mathbf{q}$ and $\mathbf{y}$, channel transition probability functions $p_{X|Q}(\cdot|\cdot)$, $p_{Y|Z}(\cdot|\cdot)$ and transform matrix $\mathbf{A}$.

[27], [28], lattice codes [29] and compressed sensing [6], [30], [31].

Although exact implementation of loopy BP for dense graphs is generally computationally difficult, Gaussian and quadratic approximations of loopy BP have been widely and successfully used for many years. Such approximate methods originated in CDMA multiuser detection problems in [1]–[5], and, more recently, have attracted considerable interest in compressed sensing [6]–[9], [20], [21]. The methods have appeared under a variety of names including "approximate BP", "relaxed BP" and, most recently, "approximate message passing". Gaussian approximations are also used in expectation propagation [32], [33], as well as the analysis and design of LDPC codes [34]–[36].

The GAMP algorithm presented here is most closely related to the approximate message passing (AMP) methods in [6], [37], [38] as well as the relaxed BP methods in [4], [5], [8]. The AMP method [6] used a quadratic approximation of max-sum loopy BP to derive an efficient implementation of the LASSO estimator [39], [40]. The LASSO estimate is equivalent to a MAP estimate of $\mathbf{x}$ assuming a Laplacian prior. This method was then generalized in [37], [38] to implement Bayesian estimation with arbitrary priors using Gaussian approximations of sum-product loopy BP. The relaxed BP method in [5], [8] offers a further extension to nonlinear output channels, but the algorithm is limited to sum-product approximations of loopy BP and the analysis is limited to certain random, sparse matrices.

The GAMP method proposed in this paper provides a unified methodology for estimation with dense matrices that can incorporate arbitrary input and output distributions and provide approximations of both max-sum and sum-product loopy BP.

Moreover, similar to previous analyses of AMP-like algorithms, we argue that that the asymptotic behavior of GAMP has a sharp characterization for certain large, i.i.d. matrices. Specifically, for large random $\mathbf{A}$, the componentwise behavior of many AMP techniques can be asymptotically described exactly by a set of state evolution (SE) equations. Such analyses have been presented for both large sparse matrices [1], [4], [5], [8], and, more recently, for dense i.i.d. matrices [6], [7]. The validity of the SE analysis on dense matrices was particularly difficult to rigorously prove since conventional

density evolution techniques such as [15], [16] need graphs that are locally cycle-free. The key breakthrough was an analysis by Bayati and Montanari in [7] that provided the first completely rigorous analysis of the dynamics of message passing in dense graphs using a new conditioning argument from [41]. That work also provided a rigorous justification of many predictions [17]–[21], [42] of the behavior of optimal estimators using the replica method from statistical physics.

The main theoretical contribution of this paper, Claim 1, can be seen as an extension of Bayati and Montanari's analysis in [7] to GAMP – the novelty being the incorporation of arbitrary output channels. Specifically, we present a generalization of the SE equations to arbitrary output channels recovering several earlier results on AWGN channels as special instances. The SE equations also agree with the results in [5], [8] for relaxed BP applied to arbitrary output channels for large sparse matrices.

The extension of Bayati and Montanari's analysis in [7] to incorporate arbitrary output channels is relatively straightforward. Unfortunately, a full re-derivation of their result would be long and beyond the scope of this paper. We thus only provide a brief of sketch of the main steps and our result is thus not fully rigorous. To emphasize the lack of rigor, we use the term Claim instead of Theorem. The predictions, however, are confirmed in numerical simulations.

A conference version of this paper appeared in [43]. This paper includes all the proofs and derivations along with more detailed discussions and simulations.

B. Outline

The outline of the remainder of this paper is as follows. As motivation, Section II provides some examples of the linear mixing problems. The GAMP method is then introduced in Section III, and precise equations for the max-sum and sum-product versions are given in Section IV. The asymptotic state evolution analysis is presented in Section V and shown to recover many previous predictions of the SE analysis in several special cases discussed in Section VI. A simple numerical simulation to confirm the predictions is presented in Section VII which considers a nonlinear compressed sensing problem. Finally, since the original publication of the paper, there has been considerable work on the topic. The conclusions summarize this work and present avenues for future research.

II. EXAMPLES AND APPLICATIONS

The linear mixing model is extremely general and can be applied in a range of circumstances. We review some simple examples for both the output and input channels from [8].

A. Output Channel Examples

a) AWGN output channel: For an additive white Gaussian noise (AWGN) output channel, the output vector $\mathbf{y}$ can be written as

$$\mathbf{y} = \mathbf{z} + \mathbf{w} = \mathbf{A}\mathbf{x} + \mathbf{w}, \tag{2}$$

where $\mathbf{w}$ is a zero mean, Gaussian i.i.d. random vector independent of $\mathbf{x}$. The corresponding channel transition probability distribution is given by

$$p_{Y|Z}(y|z) = \frac{1}{\sqrt{2\pi\tau^w}} \exp\left(-\frac{(y-z)^2}{2\tau^w}\right), \tag{3}$$

where $\tau^w > 0$ is the variance of the components of $\mathbf{w}$.

b) Non-Gaussian noise models: Since the output channel can incorporate an arbitrary separable distribution, the linear mixing model can also include the model (2) with non-Gaussian noise vectors $\mathbf{w}$, provided the components of $\mathbf{w}$ are i.i.d. One interesting application for a non-Gaussian noise model is to study the bounded noise that arises in quantization as discussed in [8].

c) Logistic channels: A quite different channel is based on a *logistic* output. In this model, each output y_i is 0 or 1, where the probability that $y_i = 1$ is given by some sigmoidal function such as

$$p_{Y|Z}(y_i = 1|z_i) = \frac{1}{1 + a\exp(-z_i)}, \tag{4}$$

for some constant $a > 0$. Thus, larger values of z_i result in a higher probability that $y_i = 1$.

This logistic model can be used in classification problems as follows [44]: Suppose one is given m samples, with each sample being labeled as belonging to one of two classes. Let $y_i = 0$ or 1 denote the class of sample i. Also, suppose that the ith row of the transform matrix $\mathbf{A}$ contains a vector of n data values associated with the ith sample. Then, using a logistic channel model such as (4), the problem of estimating the vector $\mathbf{x}$ from the labels $\mathbf{y}$ and data $\mathbf{A}$ is equivalent to finding a linear dependence on the data that classifies the samples between the two classes. This problem is often referred to as logistic regression and the resulting vector $\mathbf{x}$ is called the regression vector. By adjusting the prior on the components of $\mathbf{x}$, one can then impose constraints on the components of $\mathbf{x}$ including, for example, sparsity constraints.

B. Input Channel Examples

The distributions $p_{X|Q}(x_j|q_j)$ models the prior on x_j, with the variable q_j being a parameter of that prior that varies over the components, but is known to the estimator. When all the components of x_j are identically distributed, we can ignore q_j and use a constant distribution $p_X(x_j)$.

d) Sparse priors and compressed sensing: One class of non-Gaussian priors that have attracted considerable recent attention is sparse distributions. A vector $\mathbf{x}$ is sparse if a large fraction of its components are zero or close to zero. Sparsity can be modeled probabilistically with a variety of heavy-tailed distributions $p_X(x_j)$ including Gaussian mixture models, generalized Gaussians and Bernoulli distributions with a high probability of the component being zero. The estimation of sparse vectors with random linear measurements is the basic subject of compressed sensing [45]–[47] and fits naturally into the linear mixing framework.

e) Discrete distributions: The linear mixing model can also incorporate discrete distributions on the components of **x**. Discrete distribution arise often in communications problems where discrete messages are modulated onto the components of **x**. The linear mixing with the transform matrix **A** comes into play in CDMA spread spectrum systems and lattice codes mentioned above.

f) Power variations and dynamic range: For multiuser detection problems in communications, the parameter q_j can be used to model *a priori* power variations amongst the users that are known to the receiver. For example, suppose $q_j > 0$ almost surely and $x_j = \sqrt{q_j} u_j$, with u_j being identically distributed across all components j and independent of q_j. Then, $\mathbb{E}(x_j^2 | q_j) = q_j \mathbb{E}(u_j^2)$ and q_j thus represents a power scaling. This model has been used in a variety of analyses of CDMA multiuser detection [4], [18], [48] and random access channels [49].

III. GENERALIZED APPROXIMATE MESSAGE PASSING

We begin with a description of the generalized approximate message passing (GAMP) algorithm, which is an extension of the AMP procedure in [6], [7]. Similar to the AMP method, the idea of the algorithm is to iteratively decompose the vector valued estimation problem into a sequence of scalar operations at the input and output nodes. In the GAMP algorithm, the scalar operations are defined by two functions, $g_{\text{out}}(\cdot)$ and $g_{\text{in}}(\cdot)$, that we call the *scalar estimation* functions. We will see in Section IV that with appropriate choices of these functions, the GAMP algorithm can provide Gaussian and quadratic approximations of either sum-product and max-sum loopy BP.

The steps of the GAMP method are shown in Algorithm 1. The algorithm produces a sequence of estimates, $\widehat{\mathbf{x}}(t)$ and $\widehat{\mathbf{z}}(t)$, for the unknown vectors **x** and **z**. The algorithm also outputs vectors $\boldsymbol{\tau}^x(t)$ and $\boldsymbol{\tau}^s(t)$. As we will see in the case of the sum-product GAMP (Section IV-B), these have interpretations as certain variances. Although our analysis later is for real-valued matrices, we have written the equations for the complex case.

A. Computational Complexity

We will discuss the selection of the scalar estimation functions and provide a detailed analysis of the algorithm performance later. We first describe the algorithm's computational simplicity – which is a key part of the algorithm appeal.

Each iteration of the algorithm has four steps. The first step, the output linear step, involves only a matrix-vector multiplications by **A** and $|\mathbf{A}|^2$, where the squared magnitude is taken componentwise. The worst case complexity is $O(mn)$ and would be smaller for structured transforms such as Fourier, wavelet or sparse. The next step — the nonlinear step — involves componentwise applications of the nonlinear output estimation function $g_{\text{out}}(\cdot)$ on each of the m components of the output vector $\widehat{\mathbf{p}}$. As we will see, the function $g_{\text{out}}(\cdot)$ does not change with the dimension, so the total complexity of the output nonlinear step is complexity of $O(m)$. Similarly, the input steps involve matrix-vector multiplications by $\mathbf{A}^T$ and $(|\mathbf{A}|^2)^T$ along with componentwise scalar operations at the

---

Algorithm 1 Generalized AMP (GAMP)

Given a matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$, system inputs and outputs **q** and **y** and scalar estimation functions $g_{\text{in}}(\cdot)$, and $g_{\text{in}}(\cdot)$, generate a sequence of estimates $\widehat{\mathbf{x}}(t)$, $\widehat{\mathbf{z}}(t)$, for $t = 0, 1, \ldots$ through the following recursions:

1) *Initialization:* Set $t = 0$ and set $\widehat{x}_j(t)$ and $\tau_j^x(t)$ to some initial values.
2) *Output linear step:* For each i, compute:

$$\tau_i^p(t) = \sum_j |a_{ij}|^2 \tau_j^x(t) \tag{5a}$$

$$\widehat{p}_i(t) = \sum_j a_{ij} \widehat{x}_j(t) - \tau_i^p(t) \widehat{s}_i(t-1), \tag{5b}$$

$$\widehat{z}_i(t) = \sum_j a_{ij} \widehat{x}_j(t) \tag{5c}$$

where initially, we take $\widehat{s}(-1) = 0$.
3) *Output nonlinear step:* For each i,

$$\widehat{s}_i(t) = g_{\text{out}}(t, \widehat{p}_i(t), y_i, \tau_i^p(t)) \tag{6a}$$

$$\tau_i^s(t) = -\frac{\partial}{\partial \widehat{p}} g_{\text{out}}(t, \widehat{p}_i(t), y_i, \tau_i^p(t)). \tag{6b}$$

4) *Input linear step:* For each j,

$$\tau_j^r(t) = \left[\sum_i |a_{ij}|^2 \tau_i^s(t) \right]^{-1} \tag{7a}$$

$$\widehat{r}_j(t) = \widehat{x}_j(t) + \tau_j^r(t) \sum_i a_{ij} \widehat{s}_i(t). \tag{7b}$$

5) *Input nonlinear step:* For each j,

$$\widehat{x}_j(t+1) = g_{\text{in}}(t, \widehat{r}_j(t), q_j, \tau_j^r(t)) \tag{8a}$$

$$\tau_j^x(t+1) = \tau_j^r(t) \frac{\partial}{\partial \widehat{r}} g_{\text{in}}(t, \widehat{r}_j(t), q_j, \tau_j^r(t)). \tag{8b}$$

Then increment $t = t + 1$ and return to step 2 until a sufficient number of iterations have been performed.

---

input. The complexity is again dominated by the transforms with a worst-case complexity of $O(mn)$.

Thus, we see the GAMP algorithm reduces the vector-valued operation to a sequence of linear transforms and scalar estimation functions. The worst-case total complexity per iteration is thus $O(mn)$ and smaller for structured transforms. Moreover, as we will see in the state evolution analysis, the number of iterations required for the same per component performance does not increase with the problem size, so the total complexity is bounded by the matrix-vector multiplication. In addition, we will see in the simulations that good performance can be obtained with a small number of iterations, usually 10 to 20.

It should be pointed out that the structure of the GAMP iterations – transforms followed by scalar operations – underly many of the most of successful methods for linear-mixing type problems. In particular, the separable approximation method of [10] and alternating direction methods in [11]–[14] are all based on iterations of this form. The contribution of the current paper is to show that specific instance of these transform +

separating algorithms can be interpreted as an approximation of loopy BP and admits precise asymptotic analyses.

B. Further Simplifications

To further reduce computational complexity, the GAMP algorithm can be approximated by a modified procedure shown in Algorithm 2. In the modified procedure, the variance vectors, $\boldsymbol{\tau}^r(t)$ and $\boldsymbol{\tau}^p(t)$, are replaced with scalars $\tau^r(t)$ and $\tau^p(t)$, thus forcing all the variance components to be the same. This approximation would be valid when the components of the transform matrix $\mathbf{A}$ are the approximately equal so that $|a_{ij}|^2 \approx \|\mathbf{A}\|_F^2 / mn$ for all i, j.

The approximations in Algorithm 2 can also be heuristically justified when $\mathbf{A}$ has i.i.d. components. Specifically, if $\mathbf{A}$ is i.i.d., m and n are large, and the dependence between the components of $|A_{ij}|^2$ and the vectors $\boldsymbol{\tau}^x(t)$ and $\boldsymbol{\tau}^s(t)$ can be ignored, the simplification in Algorithm 2 can be justified through the Law of Large Numbers. Of course, $|A_{ij}|^2$ is not independent of $\boldsymbol{\tau}^x(t)$ and $\boldsymbol{\tau}^s(t)$, but in our simulation below for an i.i.d. matrix, we will see very little difference between Algorithm 1 and the simplified version, Algorithm 2.

The benefit of the simplification is that we can eliminate the matrix multiplications by $|\mathbf{A}|^2$ and $(|\mathbf{A}|^2)^T$ – reducing the number of transforms per iteration from four to two. This savings can be significant, particularly when the $\mathbf{A}^2$ and $(\mathbf{A}^2)^T$ have no particularly simple implementation. The simplified algorithm, Algorithm 2, also more closely matches the AMP procedure of [6], and will be more amenable to analysis later in Section V.

IV. SCALAR ESTIMATION FUNCTIONS TO APPROXIMATE LOOPY BP

As discussed in the previous section, through proper selection of the scalar estimation functions $g_{\text{in}}(\cdot)$ and $g_{\text{out}}(\cdot)$, GAMP can provide approximations of either max-sum or sum-product loopy BP. With these selections, we can thus realize the two most useful special cases of GAMP:

- **Max-sum GAMP**: An approximation of max-sum loopy BP for computations of the MAP estimates and computations of the marginal maxima of the posterior; and
- **Sum-product GAMP**: An approximation of sum-product loopy BP for computations of the MMSE estimates and the marginal posterior distributions.

Heuristic "derivations" of the scalar estimation functions for both of these algorithms are sketched in Appendices C and D and summarized here. The selection functions are also summarized in Table I. We emphasize that the approximations are entirely heuristic – we don't claim any formal properties of the approximation. However, the analysis of the GAMP algorithm with these or other functions will be more rigorous.

A. Max-Sum GAMP for MAP Estimation

To describe the MAP estimator, observe that for the linear mixing estimation problem in Section I, the posterior density

---

Algorithm 2 GAMP with Scalar Variances

Given a matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$, a system input and output vectors $\mathbf{q}$ and $\mathbf{y}$, selection functions $g_{\text{in}}(\cdot)$, and $g_{\text{in}}(\cdot)$, generate a sequence of estimates $\widehat{\mathbf{x}}(t)$, $\widehat{\mathbf{z}}(t)$, for $t = 0, 1, \ldots$ through the recursions:

1) *Initialization:* Set $t = 0$ and let $\widehat{x}_j(t)$ and $\tau^x(t)$ be any initial sequences.
2) *Output linear step:* For each i, compute:

$$\tau^p(t) = (1/m)\|\mathbf{A}\|_F^2 \tau^x(t) \tag{9a}$$

$$\widehat{p}_i(t) = \sum_j a_{ij}\widehat{x}_j(t) - \tau^p(t)\widehat{s}_i(t-1), \tag{9b}$$

$$\widehat{z}_i(t) = \sum_j a_{ij}\widehat{x}_j(t) \tag{9c}$$

where initially, we take $\widehat{s}(-1) = 0$.
3) *Output nonlinear step:* For each i,

$$\widehat{s}_i(t) = g_{\text{out}}(t, \widehat{p}_i(t), y_i, \tau^p(t)) \tag{10a}$$

$$\tau^s(t) = -\frac{1}{m}\sum_{i=1}^m \frac{\partial}{\partial \widehat{p}} g_{\text{out}}(t, \widehat{p}_i(t), y_i, \tau^p(t)). \tag{10b}$$

4) *Input linear step:* For each j,

$$1/\tau^r(t) = (1/n)\|\mathbf{A}\|_F^2 \tau^s(t) \tag{11a}$$

$$\widehat{r}_j(t) = \widehat{x}_j(t) + \tau^r(t)\sum_i a_{ij}\widehat{s}_i(t). \tag{11b}$$

5) *Input nonlinear step:*

$$\widehat{x}_j(t+1) = g_{\text{in}}(t, \widehat{r}_j(t), q_j, \tau^r(t)) \tag{12a}$$

$$\tau^x(t+1) =$$
$$\frac{\tau^r(t)}{n}\sum_{j=1}^n \frac{\partial}{\partial \widehat{r}} g_{\text{in}}(t, \widehat{r}_j(t), q_j, \tau^r(t)). \tag{12b}$$

Then increment $t = t + 1$ and return to step 2 until a sufficient number of iterations have been performed.

---

of the vector $\mathbf{x}$ given the system input $\mathbf{q}$ and output $\mathbf{y}$ is given by the conditional density function

$$p_{\mathbf{x}|\mathbf{q},\mathbf{y}}(\mathbf{x}|\mathbf{q},\mathbf{y}) := \frac{1}{Z(\mathbf{q},\mathbf{y})} \exp\left(F(\mathbf{x}, \mathbf{A}\mathbf{x}, \mathbf{q}, \mathbf{y})\right), \tag{13}$$

where

$$F(\mathbf{x}, \mathbf{z}, \mathbf{q}, \mathbf{y}) := \sum_{j=1}^n f_{\text{in}}(x_j, q_j) + \sum_{i=1}^m f_{\text{out}}(z_i, y_i), \tag{14}$$

and

$$f_{\text{out}}(z, y) := \log p_{Y|Z}(y|z) \tag{15a}$$

$$f_{\text{in}}(x, q) := \log p_{X|Q}(x|q). \tag{15b}$$

The constant $Z(\mathbf{q}, \mathbf{y})$ in (13) is a normalization constant. Given this distribution, the *maximum a posteriori* (MAP) estimator is the maxima of (13) which is given by the optimization

$$\widehat{\mathbf{x}}^{\text{map}} := \arg\max_{\mathbf{x} \in \mathbb{R}^n} F(\mathbf{x}, \mathbf{z}, \mathbf{q}, \mathbf{y}), \quad \widehat{\mathbf{z}} = \mathbf{A}\widehat{\mathbf{x}}. \tag{16}$$

Method	Input scalar estimation functions		Output scalar estimation functions	
	$g_{\text{in}}(\widehat{r}, q, \tau^r)$	$\tau^r g'_{\text{in}}(\widehat{r}, q, \tau^r)$	$g_{\text{out}}(\widehat{p}, y, \tau^p)$	$-g'_{\text{out}}(\widehat{p}, y, \tau^p)$
Max-sum GAMP for MAP estimation	$\arg\max_x F_{\text{in}}(z, \widehat{r}, q, \tau^r)$	$\tau^r / (1 - \tau^r f''_{\text{in}}(\widehat{x}, q))$	$(\widehat{z}^0 - \widehat{p})/\tau^p$ $\widehat{z}^0 := \arg\max_z F_{\text{out}}(z, \widehat{p}, y, \tau^p)$	$f''_{\text{out}}(\widehat{z}^0, y)/(\tau^p f''_{\text{out}}(\widehat{z}^0, y) - 1)$
Sum-product GAMP for MMSE estimation	$E(x|\widehat{r}, q, \tau^r)$ $R = X + \mathcal{N}(0, \tau^r)$	$\text{var}(x|\widehat{r}, q, \tau^r)$	$(\widehat{z}^0 - \widehat{p})/\tau^p$ $\widehat{z}^0 := \mathbb{E}(z|\widehat{p}, y, \tau^p)$ $Y \sim P_{Y|Z}, Z \sim \mathcal{N}(\widehat{p}, \tau^p)$	$(\tau^p(t) - \text{var}(z|\widehat{p}, y))/(\tau^p(t))^2$
AWGN	$\tau^{x0}(\widehat{r}-q)/(\tau^r+\tau^{x0})+q$ $X = \mathcal{N}(q, \tau^{x0})$	$\tau^{x0}\tau^r/(\tau^r + \tau^{x0})$	$(y - \widehat{p})/(\tau^w + \tau^p)$ $Y = Z + \mathcal{N}(0, \tau^w)$	$1/(\tau^w + \tau^p)$

TABLE I
SCALAR INPUT AND OUTPUT ESTIMATION FUNCTIONS MAX-SUM AND SUM-PRODUCT GAMP. FOR AWGN INPUTS AND OUTPUTS, THE SCALAR
ESTIMATION FUNCTIONS FOR BOTH MAX-SUM AND SUM-PRODUCT ALGORITHMS ARE THE SAME AND HAVE A PARTICULARLY SIMPLE FORM.

For each component j, we will also be interested in the *marginal maxima* of the posterior distribution

$$\Delta_j(x_j) := \max_{\mathbf{x}\backslash x_j} F(\mathbf{x}, \mathbf{z}, \mathbf{q}, \mathbf{y}), \quad \widehat{\mathbf{z}} = \mathbf{A}\widehat{\mathbf{x}}, \qquad (17)$$

where the maximization is over vector $\mathbf{x} \in \mathbb{R}^n$ with a fixed value for the coefficient x_j. Note that the component $\widehat{x}_j$ of the MAP estimate is given by $\widehat{x}_j = \arg\max_{x_j} \Delta_j(x_j)$. This, $\Delta_j(x_j)$ can be interpreted as the sensitivity of the maxima to the value of the component x_j.

Note that one may also be interested in the optimization (16) where the objective is of the form (14), but the functions $f_{\text{in}}(\cdot)$ and $f_{\text{out}}(\cdot)$ are not derived from any density functions. The max-sum GAMP method can be applied to these general optimization problems as well.

Now, an approximate implementation of max-sum BP for the MAP estimation problem (16) is described in Appendix C. It is suggested there that a possible input function to approximately implement max-sum BP is given by

$$g_{\text{in}}(\widehat{r}, q, \tau^r) := \arg\max_x F_{\text{in}}(x, \widehat{r}, q, \tau^r) \qquad (18)$$

where

$$F_{\text{in}}(x, \widehat{r}, q, \tau^r) := f_{\text{in}}(x, q) - \frac{1}{2\tau^r}(\widehat{r} - x)^2. \qquad (19)$$

Here, and below, we have dropped the dependence on the iteration number t when it is not needed. The Appendix also shows that the function (18) has a derivative satisfying

$$\tau^r \frac{\partial}{\partial \widehat{r}} g_{\text{in}}(\widehat{r}, q, \tau^r) = \frac{\tau^r}{1 - \tau^r f''_{\text{in}}(\widehat{x}, q)}, \qquad (20)$$

where the second derivative $f''_{\text{in}}(x, q)$ is with respect to x and $\widehat{x} = g_{\text{in}}(r, q, \tau^r)$. Also, the marginal maxima (17) is approximately given by

$$\Delta_j(x_j) \approx F_{\text{in}}(x_j, \widehat{r}_j, q_j, \tau^r_j) + \text{const}, \qquad (21)$$

where the constant term does not depend on x_j.

The initial condition for the GAMP algorithm should be taken as

$$\widehat{x}_j(0) = \arg\max_{x_j} f_{\text{in}}(x_j, q_j), \quad \tau^x_j(0) = \frac{1}{f''_{\text{in}}(\widehat{x}_j(0), q_j)}. \qquad (22)$$

This initial conditions corresponds to the output of (8) with $t = 0$, the functions in (18) and (20) and $\tau^r(-1) \to \infty$.

Observe that when f_{in} is given by (15b), $g_{\text{in}}(\widehat{r}, q, \tau^r)$ is precisely the scalar MAP estimate of a random variable X given $Q = q$ and $\widehat{R} = \widehat{r}$ for the random variables

$$\widehat{R} = X + V, \quad V \sim \mathcal{N}(0, \tau^r), \qquad (23)$$

where $X \sim p_{X|Q}(x|q)$, $Q \sim p_Q(q)$ and with V independent of X and Q. With this definition, $\widehat{R}$ can be interpreted as a Gaussian noise-corrupted version of X with noise level τ^r.

Appendix C shows that the output function for the approximation of max-sum BP is given by

$$g_{\text{out}}(\widehat{p}, y, \tau^p) := \frac{1}{\tau^p}(\widehat{z}^0 - \widehat{p}), \qquad (24)$$

where

$$\widehat{z}^0 := \arg\max_z F_{\text{out}}(z, \widehat{p}, y, \tau^p), \qquad (25)$$

and

$$F_{\text{out}}(z, \widehat{p}, y, \tau^p) := f_{\text{out}}(z, y) - \frac{1}{2\tau^p}(z - \widehat{p})^2. \qquad (26)$$

The negative derivative of this function is given by

$$-\frac{\partial}{\partial \widehat{p}} g_{\text{out}}(\widehat{p}, y, \tau^p) = \frac{-f''_{\text{out}}(\widehat{z}^0, y)}{1 - \tau^p f''_{\text{out}}(\widehat{z}^0, y)}, \qquad (27)$$

where the second derivative $f''_{\text{out}}(z, y)$ is with respect to z.

Now, when $f_{\text{out}}(z, y)$ is given by (15a), $F_{\text{out}}(z, \widehat{p}, y, \tau^p)$ in (26) can be interpreted as the log posterior of a random variable Z given $Y = y$ and

$$Z \sim \mathcal{N}(\widehat{p}, \tau^p), \qquad Y \sim p_{Y|Z}(y|z). \qquad (28)$$

In particular, $\widehat{z}^0$ in (25) is the MAP estimate of Z.

We see that the max-sum GAMP algorithm reduces the vector MAP estimation problem to a sequence of scalar MAP estimations problems at the inputs and outputs. The scalar parameters $\tau^r(t)$ and $\tau^p(t)$ represent effective Gaussian noise levels in these problems. The equations for the scalar estimation functions are summarized in Table I.

B. Sum-Product GAMP for MMSE Estimation

The minimum mean squared error (MMSE) estimate is the conditional expectation

$$\widehat{\mathbf{x}}^{\text{mmse}} := \mathbb{E}[\mathbf{x} \mid \mathbf{y}, \mathbf{q}], \qquad (29)$$

relative to the conditional density (13). We are also interested in the log posterior marginals

$$\Delta_j(x_j) := \log p(x_j | \mathbf{q}, \mathbf{y}). \tag{30}$$

The selection of the functions $g_{\text{in}}(\cdot)$ and $g_{\text{out}}(\cdot)$ to approximately implement sum-product loopy BP to compute the MMSE estimates and posterior marginals is described in Appendix D. Heuristic arguments in that section, show that the input function to implement BP-based MMSE estimation is given by

$$g_{\text{in}}(\widehat{r}, q, \tau^r) := \mathbb{E}[X \mid \widehat{R} = \widehat{r}, \; Q = q], \tag{31}$$

where the expectation is over the variables in (23). Also, the derivative is given by the variance,

$$\tau^r \frac{\partial}{\partial \widehat{r}} g_{\text{in}}(\widehat{r}, q, \tau^r) := \text{var}[X \mid \widehat{R} = \widehat{r}, \; Q = q]. \tag{32}$$

In addition, the log posterior marginal (30) is approximately given by (21). From the definition of $F_{\text{in}}(\cdot)$ in (19) we see that the posterior marginal is approximately given by

$$p(x_j | \mathbf{q}, \mathbf{y}) \approx \frac{1}{Z} p_{X|Q}(x_j | q_j) \exp\left[-\frac{1}{2\tau_r}(\widehat{r}_j - x_j)^2\right], \tag{33}$$

where Z is a normalization constant.

The initial condition for the GAMP algorithm for MMSE estimation should be taken as

$$\widehat{x}_j(0) = \mathbb{E}(X \mid Q = q_j) \quad \tau_j^x(0) = \text{var}(X \mid Q = q_j), \tag{34}$$

where the expectations are over the distribution $p_{X|Q}(x_j | q_j)$. Thus, the algorithm is initialized to the the prior mean and variance on x_j based on the parameter q_j but no observations. Equivalently, the initial conditions (34) corresponds to the output of (8) with $t = 0$, the functions in (31) and (32) and $\tau^r(-1) \to \infty$.

To describe the output function $g_{\text{out}}(\widehat{p}, y, \tau^p)$, consider a random variable z with conditional probability density

$$p(z | \widehat{p}, y, \tau^p) \propto \exp F_{\text{out}}(z, \widehat{p}, y, \tau^p), \tag{35}$$

where $F_{\text{out}}(z, \widehat{p}, y, \tau^p)$ is given in (26). The distribution (35) is the posterior density function of the random variable Z with observation Y in (28). Appendix D shows that the output function $g_{\text{out}}(\widehat{p}, y, \tau^p)$ to implement approximate BP for the MMSE problem is given by

$$g_{\text{out}}(\widehat{p}, y, \tau^p) := \frac{1}{\tau^p}(\widehat{z}^0 - \widehat{p}), \quad \widehat{z}^0 := \mathbb{E}(z | \widehat{p}, y, \tau^p), \tag{36}$$

where the expectation is over the density function (35). Also, the negative derivative of $g_{\text{out}}(\cdot)$ is given by

$$-\frac{\partial}{\partial \widehat{p}} g_{\text{out}}(\widehat{p}, y, \tau^p) = \frac{1}{\tau^p}\left(1 - \frac{\text{var}(z | \widehat{p}, y, \tau^p)}{\tau^p}\right). \tag{37}$$

Appendix D also provides an alternative definition for $g_{\text{out}}(\cdot)$: The function $g_{\text{out}}(\cdot)$ in (36) is given by

$$g_{\text{out}}(\widehat{p}, y, \tau^p) := \frac{\partial}{\partial \widehat{p}} \log p(y | \widehat{p}, \tau^p), \tag{38}$$

where $p(y | \widehat{p}, \tau^p)$ is the density is from the channel (28). As a result, its negative derivative is

$$-\frac{\partial}{\partial \widehat{p}} g_{\text{out}}(\widehat{p}, y, \tau^p) = -\frac{\partial^2}{\partial \widehat{p}^2} \log p(y | \widehat{p}, \tau^p). \tag{39}$$

Hence $g_{\text{out}}(\widehat{p}, y, \tau^p)$ has the interpretation of a *score function* of the parameter $\widehat{p}$ in the distribution of the random variable Y in (28).

Thus, similar to the MAP estimation problem, the sum-product GAMP algorithm reduces the vector MMSE estimation problem to the evaluation of sequence of scalar estimation problems from Gaussian noise. Scalar MMSE estimation is performed at the input nodes, and a score function of an ML estimation problem is performed at the output nodes.

C. AWGN Output Channels

In the special case of an AWGN output channel, we will see that that the output functions for max-sum and sum-product GAMP are identical and reduce to the update in the AMP algorithm of Bayati and Montanari in [7]. Suppose that the output channel is described by the AWGN distribution (3) for some output variance $\tau^w > 0$. Then, it can be checked that the distribution $p(z | \widehat{p}, y, \tau^p)$ in (35) is the Gaussian

$$p(z | \widehat{p}, y, \tau^p) \sim \mathcal{N}(\widehat{z}^0, \tau^z), \tag{40}$$

where

$$\widehat{z}^0 \quad := \quad \widehat{p} + \frac{\tau^p}{\tau^w + \tau^p}(y - \widehat{p}), \tag{41a}$$

$$\tau^z \quad := \quad \frac{\tau^w \tau^p}{\tau^w + \tau^p} \tag{41b}$$

It can be verified that the conditional mean $\widehat{z}^0$ in (41a) agrees with both $\widehat{z}^0$ in (25) for the MAP estimator and $\widehat{z}^0$ in (36) for the MMSE estimator. Therefore, the output function g_{out} for both the MAP estimate in (24) and MMSE estimate in (36) is given by

$$g_{\text{out}}(\widehat{p}, y, \tau^p) := \frac{y - \widehat{p}}{\tau^w + \tau^p}. \tag{42}$$

The negative derivative of the function is given by

$$-\frac{\partial}{\partial \widehat{p}} g_{\text{out}}(\widehat{p}, y, \tau^p) = \frac{1}{\tau^p + \tau^w}. \tag{43}$$

Therefore, for both max-sum and sum-product GAMP $g_{\text{out}}(\widehat{p}, y, \tau^p)$ and its derivative are given by (42) and (43). When, we apply these equations into the input updates (8), we precisely obtain the original AMP algorithm (with some scalings) given in [7].

D. AWGN Input Channels

Now suppose that the input density function $p_{X|Q}(x|q)$ is a Gaussian density

$$p_{X|Q}(x|q) = \mathcal{N}(q, \tau^{x0}),$$

for some variance $\tau^{x0} > 0$. Then, it is easily checked that the input estimate function $g_{\text{in}}(\cdot)$ and its derivative are identical for both max-sum and sum-product GAMP and given by

$$g_{\text{in}}(\widehat{r}, q, \tau^r) \quad := \quad \frac{\tau^{x0}}{\tau^{x0} + \tau^r}(\widehat{r} - q) + q \tag{44a}$$

$$\tau^r g'_{\text{in}}(\widehat{r}, q, \tau^r) \quad := \quad \frac{\tau^{x0} \tau^r}{\tau^{x0} + \tau^r}. \tag{44b}$$

The functions are shown in Table I.

V. ASYMPTOTIC ANALYSIS

We now present our main theoretical result, which is the SE analysis of GAMP for large, Gaussian i.i.d. matrices $\mathbf{A}$.

A. Empirical Convergence of Random Variables

The analysis is a relatively minor modification of the results in Bayati and Montanari's paper [7]. The work [7] employs certain deterministic models on the vectors and then proves convergence properties of related empirical distributions. To apply the same analysis here, we need to review some of their definitions. We say a function $\phi : \mathbb{R}^r \to \mathbb{R}^s$ is *pseudo-Lipschitz* of order $k > 1$, if there exists an $L > 0$ such for any $\mathbf{x}, \mathbf{y} \in \mathbb{R}^r$,

$$\|\phi(\mathbf{x}) - \phi(\mathbf{y})\| \le L(1 + \|\mathbf{x}\|^{k-1} + \|\mathbf{y}\|^{k-1})\|\mathbf{x} - \mathbf{y}\|.$$

Now suppose that for each $n = 1, 2, \ldots$, $\mathbf{v}^{(n)}$ is a block vector with components $\mathbf{v}_i^{(n)} \in \mathbb{R}^s$, $i = 1, 2, \ldots, \ell(n)$ for some $\ell(n)$. So, the total dimension of $\mathbf{v}^{(n)}$ is $s\ell(n)$. We say that the components of the vectors $\mathbf{v}^{(n)}$ *empirically converges with bounded moments of order k* as $n \to \infty$ to a random vector $\mathbf{V}$ on $\mathbb{R}^s$ if: For all pseudo-Lipschitz continuous functions, ϕ, of order k,

$$\lim_{n \to \infty} \frac{1}{\ell(n)} \sum_{i=1}^{\ell(n)} \phi(\mathbf{v}_i^{(n)}) = \mathbb{E}(\phi(\mathbf{V})) < \infty.$$

When the nature of convergence is clear, we may write (with some abuse of notation)

$$\lim_{n \to \infty} \mathbf{v}_i^{(n)} \stackrel{d}{=} \mathbf{V}.$$

B. Assumptions

With these definitions, we can now formally state the modeling assumptions, which follow along the lines of the asymptotic model considered by Bayati and Montanari in [7]. Specifically, we consider a sequence of random realizations of the estimation problem in Section I indexed by the input signal dimension n. For each n, we assume the output dimension $m = m(n)$ is deterministic and scales linearly with the input dimension in that

$$\lim_{n \to \infty} n/m(n) = \beta, \tag{45}$$

for some $\beta > 0$ called the *measurement ratio*. We also assume that the transform matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ has i.i.d. Gaussian components $a_{ij} \sim \mathcal{N}(0, 1/m)$ and $\mathbf{z} = \mathbf{A}\mathbf{x}$.

We assume that for some order $k \ge 2$, the components of initial condition $\widehat{\mathbf{x}}(0)$, $\boldsymbol{\tau}^x(0)$ and input vectors $\mathbf{x}$ and $\mathbf{q}$ empirically converge with bounded moments of order $2k - 2$ as

$$\lim_{n \to \infty} (\widehat{x}_j(0), \tau_j^x(0), x_j, q_j) \stackrel{d}{=} (\widehat{X}_0, \overline{\tau}^x(0), X, Q), \tag{46}$$

for some random variable triple $(\widehat{X}_0, X, Q)$ with joint density $p_{\widehat{X}_0, X, Q}(\widehat{x}_0, x, q)$ and constant $\overline{\tau}^x(0)$.

To model the dependence of the system output vector $\mathbf{y}$ on the transform output $\mathbf{z}$, we assume that, for each n, there is a deterministic vector $\mathbf{w} \in W^m$ for some set W, which we can

think of as a noise vector. Then, for every $i = 1, \ldots, m$, we assume that

$$y_i = h(z_i, w_i) \tag{47}$$

where h is some function $h : \mathbb{R} \times W \to Y$ and Y is the output set. Finally, we assume that the components of $\mathbf{w}$ empirically converge with bounded moments of order $2k - 2$ to some random variable W with distribution $p_W(w)$. We will write $p_{Y|Z}(y|z)$ for the conditional distribution of Y given Z in this model.

We also need certain continuity assumptions. Specifically, we assume that the partial derivatives of the functions $g_{\text{in}}(t, \widehat{r}, q, \tau^r)$ and $g_{\text{out}}(t, \widehat{p}, h(z, w), \tau^p)$ with respect to $\widehat{r}$, $\widehat{p}$ and z exist almost everywhere and are pseudo-Lipschitz continuous of order k. This assumption implies that the functions $g_{\text{in}}(t, \widehat{r}, q, \tau^r)$ and $g_{\text{out}}(t, \widehat{p}, y, \tau^p)$ are Lipschitz continuous in $\widehat{r}$ and $\widehat{p}$ respectively.

C. State Evolution Equations

Similar to [7], the key result here is that the behavior of the GAMP algorithm is described by a set of state evolution (SE) equations. For the GAMP algorithm, the SE equations are easiest to describe algorithmically as shown in in Algorithm 3. To describe the equations, we introduce two random vectors – one corresponding to the input channel, and the other corresponding to the output channel. At the input channel, given α^r, $\xi^r \in \mathbb{R}$, define the random variable triple

$$\theta^r(\xi^r, \alpha^r) := (X, Q, \widehat{R}), \tag{48}$$

where the distribution (X, Q) are derived from the density in (46) and $\widehat{R}$ is given

$$\widehat{R} = \alpha^r X + V, \quad V \sim \mathcal{N}(0, \xi^r), \tag{49}$$

with V independent of X and Q. At the output channel, given a covariance matrix $\mathbf{K}^p \in \mathbb{R}^{2 \times 2}$, $\mathbf{K}^p > 0$, define the four-dimensional random vector

$$\theta^p(\mathbf{K}^p) := (Z, \widehat{P}, W, Y), \quad Y = h(Z, W), \tag{50}$$

such that W and $(Z, \widehat{P})$ are independent with distributions

$$(Z, \widehat{P}) \sim \mathcal{N}(0, \mathbf{K}^p), \quad W \sim p_W(w). \tag{51}$$

Since the computations in (53) and (54) are expectations over scalar random variables, they can be evaluated numerically given functions $g_{\text{in}}(\cdot)$ and $g_{\text{out}}(\cdot)$.

D. Main Result

To simplify the analysis, we consider the GAMP method with scalar variances, Algorithm 2. Our simulations indicate no difference between Algorithms 1 and 2 at moderate block lengths. We also assume the following minor modifications

- The variance $\tau^r(t)$ in (11) is replaced by its deterministic limit $\overline{\tau}^r(t)$;
- The variance $\tau^p(t)$ in (9) is replaced by its deterministic limit $\overline{\tau}^p(t)$; and
- The norm $\|\mathbf{A}\|_F^2$ is replaced by its expectation $\mathbb{E}\|\mathbf{A}\|_F^2 = n$.

Algorithm 3 GAMP State Evolution

Given scalar estimation functions $g_{\text{in}}(\cdot)$ and $g_{\text{out}}(\cdot)$, the density function $p_{\widehat{X}_0, X, Q}$ and initial value $\overline{\tau}^x(0)$ in (46), the density function $p_{Y|Z}$ at the output and the measurement ratio $\beta = m/n$, compute the state evolution parameters as follows:

1) *Initialization:* Set $t = 0$, let $\overline{\tau}^x(0)$ be the initial value in (46) and set

$$\mathbf{K}^x(0) = \text{cov}(X, \widehat{X}(0)), \qquad (52)$$

meaning the covariance matrix of the random variables $(X, \widehat{X}(0))$ in the limit (46).

2) *Output node update:* Compute

$$\overline{\tau}^p(t) = \beta\overline{\tau}^x(t), \quad \mathbf{K}^p(t) = \beta\mathbf{K}^x(t) \qquad (53a)$$

$$\overline{\tau}^r(t) = -\mathbb{E}^{-1}\left[\frac{\partial}{\partial\widehat{p}}g_{\text{out}}(t, \widehat{P}, Y, \overline{\tau}^p(t))\right] \qquad (53b)$$

$$\xi^r(t) = (\overline{\tau}^r(t))^2\mathbb{E}\left[g_{\text{out}}^2(t, \widehat{P}, Y, \overline{\tau}^p(t))\right] \qquad (53c)$$

where the expectations are over the random variable triples $\theta^p(\mathbf{K}^p(t)) = (Z, \widehat{P}, W, Y)$ in (50). Also, let

$$\alpha^r(t) = \overline{\tau}^r(t)$$
$$\times \; \mathbb{E}\left[\frac{\partial}{\partial z}g_{\text{out}}(t, \widehat{P}, h(z, W), \overline{\tau}^p(t))\Big|_{z=Z}\right]. \qquad (53d)$$

3) *Input node update:* Compute

$$\overline{\tau}^x(t+1) = \overline{\tau}^r(t)\mathbb{E}\left[\frac{\partial}{\partial\widehat{r}}g_{\text{in}}(t, Q, \widehat{R}, \overline{\tau}^r(t))\right] \qquad (54a)$$

$$\mathbf{K}^x(t+1) = \text{cov}(X, \widehat{X}(t+1)) \qquad (54b)$$

where the expectation is over the random variable triple $\theta^r(\xi^r(t), \alpha^r(t)) = (X, Q, \widehat{R})$ in (48), and

$$\widehat{X}(t+1) = g_{\text{in}}(t, Q, \widehat{R}, \overline{\tau}^r(t)).$$

Increment $t = t + 1$ and return to step 2.

These simplifications are likely not significant, since we will show that, under these simplifications, for all t, $\tau^r(t) \to \overline{\tau}^r(t)$ and $\tau^p(t) \to \overline{\tau}^p(t)$. Also, since $\mathbf{A}$ has i.i.d. components with variance $1/m$, $(1/n)\mathbb{E}\|\mathbf{A}\|^2 \to 1$. Moreover, it is possible that one could formally justify the simplification with the arguments in [50].

Claim 1: Consider the GAMP with scalar variances, Algorithm 2, under the assumptions in Section V-B and with the above modifications. Then, for any fixed iteration number t:

(a) Almost surely, we have the limits

$$\lim_{n\to\infty}\tau^r(t) = \overline{\tau}^r(t), \quad \lim_{n\to\infty}\tau^p(t) = \overline{\tau}^p(t). \qquad (55)$$

(b) The components of the vectors $\mathbf{x}$, $\mathbf{q}$, $\widehat{\mathbf{r}}$ and $\widehat{\mathbf{x}}$ empirically converge with bounded moments of order k as

$$\lim_{n\to\infty}(x_j, q_j, \widehat{r}_j(t)) \overset{d}{=} \theta^r(\xi^r(t), \alpha^r(t)), \qquad (56)$$

where $\theta^r(\xi^r(t), \alpha^r(t)) = (X, Q, \widehat{R})$ is the random variable triple in (48) and $\xi^r(t)$ is from the SE equations.
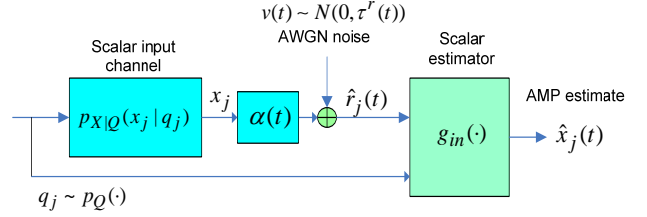

Fig. 2. Scalar equivalent model. The joint distribution of any one component x_j and its estimate $\widehat{x}_j(t)$ from the tth iteration of the AMP algorithm is identical asymptotically to an equivalent estimator with Gaussian noise.

(c) The components of the vectors $\mathbf{z}$, $\widehat{\mathbf{p}}$, $\mathbf{w}$ and $\mathbf{y}$ empirically converge with bounded moments of order k as

$$\lim_{n\to\infty}(z_i, \widehat{p}_i(t), w_i, y_i) \overset{d}{=} \theta^p(\mathbf{K}^p(t)), \qquad (57)$$

where $\theta^p(\mathbf{K}^p(t)) = (Z, \widehat{P}, W, Y)$ is the random vector in (50) and $\mathbf{K}^p(t)$ is given by the SE equations.

A proof of the claim is sketched in Appendix A. We use the term "claim" here since the proof relies on an extension of a general result from [7] to vector-valued quantities. Due to space considerations, we only sketch a proof of the extension. The term "claim", as opposed to "theorem," is used to emphasize that the details have been omitted.

A useful interpretation of Claim 1 is that it provides a *scalar equivalent model* for the behavior of the GAMP estimates. The equivalent scalar model is illustrated in Fig. 2. To understand this diagram, observe that part (b) of Claim 1 shows that the joint empirical distribution of the components $(x_j, q_j, \widehat{r}_j(t))$ converges to $\theta^r(\xi^r(t), \alpha^r(t)) = (X, Q, \widehat{R})$ in (48). This distribution is identical to $\widehat{r}_j(t)$ being a scaled and noise-corrupted version of x_j. Then, the estimate $\widehat{x}_j(t) = g_{\text{in}}(t, \widehat{r}_j(t), q_j, \tau_j^r(t))$ is a nonlinear scalar function of $\widehat{r}_j(t)$ and q_j. A similar interpretation can be drawn at the output nodes.

VI. SPECIAL CASES

We now show that several previous results can be recovered from the general SE equations in Section V-C as special cases.

A. AWGN Output Channel

First consider the case where the output function (47) is given by additive noise model

$$y_i = h(z_i, w_i) = z_i + w_i. \qquad (58)$$

Assume the components w_i of the output noise vector empirically converge to a random variable W with zero mean and variance $\tau^w > 0$. The output noise W need not be Gaussian. But, suppose the GAMP algorithm uses an output function $g_{\text{out}}(\cdot)$ in (42) corresponding to an AWGN output for some *postulated* output variance τ^w_{post} that may differ from τ^w. This is the scenario analyzed in Bayati and Montanari's paper [7].

Substituting (43) with the postulated noise variance τ^w_{post} into (53b) we get

$$\overline{\tau}^r(t) = \tau^w_{\text{post}} + \overline{\tau}^p(t) = \tau^w_{\text{post}} + \beta\overline{\tau}^x(t). \qquad (59)$$

Also, using (42) and (58), we see that

$$\frac{\partial}{\partial z} g_{\mathrm{out}}(\widehat{p}, h(z,y), \overline{\tau}^p(t)) = \frac{\partial}{\partial z} \frac{z + w - \widehat{p}}{\tau_{\mathrm{post}}^w + \overline{\tau}^p(t)} = \frac{1}{\overline{\tau}^r(t)}.$$

Therefore, (53d) implies that $\alpha^r(t) = 1$.

Now define

$$\xi^x(t) := [1 \ \ -1] \mathbf{K}^x(t) \begin{bmatrix} 1 \\ -1 \end{bmatrix} \tag{60a}$$

$$\xi^p(t) := \beta \xi^x(t) = [1 \ \ -1] \mathbf{K}^p(t) \begin{bmatrix} 1 \\ -1 \end{bmatrix}. \tag{60b}$$

With these definitions, note that if $(Z, \widehat{P}) \sim \mathcal{N}(0, \mathbf{K}^p(t))$, then

$$\xi^p(t) = \mathbb{E}(Z - \widehat{P})^2. \tag{61}$$

Also, with $\mathbf{K}^x(t)$ defined in (54b),

$$\xi^x(t+1) = \mathbb{E}(X - \widehat{X}(t))^2$$
$$= \mathbb{E}\left[X - g_{\mathrm{in}}(t, \widehat{R}, Q, \overline{\tau}^r(t))\right]^2. \tag{62}$$

Therefore,

$$\xi^r(t+1) \overset{(a)}{=} (\overline{\tau}^r(t+1))^2 \frac{\mathbb{E}(Y - \widehat{P})^2}{(\tau_{\mathrm{post}}^w + \overline{\tau}^p(t+1))^2}$$
$$\overset{(b)}{=} \mathbb{E}(Y - \widehat{P})^2$$
$$\overset{(c)}{=} \mathbb{E}(W + Z - \widehat{P})^2 = \tau^w + \mathbb{E}(Z - P)^2$$
$$\overset{(d)}{=} \tau^w + \xi_p(t+1)$$
$$\overset{(e)}{=} \tau^w + \beta \mathbb{E}\left[X - g_{\mathrm{in}}(t, \widehat{R}, Q, \overline{\tau}^r(t))\right]^2, \tag{63}$$

where (a) follows from substituting (42) into (53c); (b) follows from (59); (c) follows from the output channel model assumption (58); (d) follows from (61) and (e) follows from (60) and (62). The expectation in (63) is over $\theta^r(\xi^r(t), \alpha^r(t)) = (X, Q, \widehat{R})$ in (48). Since $\alpha^r(t) = 1$, the expectation is over random variables $(X, Q, \widehat{R})$ where $X \sim p_{X|Q}(x|q)$, $Q \sim p_Q(q)$ and

$$\widehat{R} = X + V, \quad V \sim \mathcal{N}(0, \xi^r(t)), \tag{64}$$

and V is independent of X and Q. With this expectation, (63) is precisely the SE equation in [7] for the AMP algorithm with a general input function $g_{\mathrm{in}}(\cdot)$.

B. Sum-Product GAMP with AWGN Output Channels

The SE equation (63) applies to a general input function $g_{\mathrm{in}}(\cdot)$. Now suppose that the estimator uses an input function $g_{\mathrm{in}}(\cdot)$ based on the sum-product estimator (31). To account for possible mismatches with the estimator, suppose that the sum-product GAMP algorithm is run with some postulated distribution $p_{X|Q}^{\mathrm{post}}(\cdot)$ that may differ from the true distribution $p_{X|Q}(\cdot)$. Let $\widehat{x}_{\mathrm{mmse}}^{\mathrm{post}}(\widehat{r}, q, \tau^r)$ be the corresponding scalar MMSE estimator

$$\widehat{x}_{\mathrm{mmse}}^{\mathrm{post}}(\widehat{r}, q, \tau^r) := \mathbb{E}\left(X | \widehat{R} = \widehat{r}, Q = q\right),$$

where the expectation is over the random variable

$$\widehat{R} = X + V, \quad V \sim \mathcal{N}(0, \tau^r), \tag{65}$$

where X given Q follows the postulated distribution $p_{X|Q}^{\mathrm{post}}(x|q)$ and V is independent of X and Q. Let $\mathcal{E}_{\mathrm{mmse}}^{\mathrm{post}}(\widehat{r}, q, \tau^r)$ denote the corresponding postulated variance. Then, (31) and (32) can be re-written

$$g_{\mathrm{in}}(\widehat{r}, q, \tau^r) = \widehat{x}_{\mathrm{mmse}}^{\mathrm{post}}(\widehat{r}, q, \tau^r), \tag{66a}$$

and

$$\tau^r \frac{\partial}{\partial \widehat{r}} g_{\mathrm{in}}(\widehat{r}, q, \tau^r) = \mathcal{E}_{\mathrm{mmse}}^{\mathrm{post}}(\widehat{r}, q, \tau^r). \tag{66b}$$

With this notation,

$$\overline{\tau}^r(t+1) \overset{(a)}{=} \tau_{\mathrm{post}}^w + \beta \overline{\tau}^x(t+1)$$
$$\overset{(b)}{=} \tau_{\mathrm{post}}^w + \beta \mathbb{E}\left[\mathcal{E}_{\mathrm{mmse}}^{\mathrm{post}}(\widehat{r}, q, \overline{\tau}^r(t))\right] \tag{67}$$

where (a) follows from (59) and (b) follows from substituting (66b) into (54a). Also, substituting (66a) into (63), we obtain

$$\xi^r(t+1) = \tau^w + \beta \mathbb{E}\left[X - \widehat{x}_{\mathrm{mmse}}^{\mathrm{post}}(\widehat{R}, Q, \overline{\tau}^r(t))\right]^2. \tag{68}$$

The fixed point of the updates (67) and (68) are precisely the equations given by Guo and Verdú in their replica analysis of MMSE estimation with AWGN outputs and non-Gaussian priors [18]. Their work uses the replica method from statistical physics to argue the following: Let $\widehat{\mathbf{x}}^{\mathrm{post}}$ be the exact MMSE estimate of the vector $\mathbf{x}$ based on the postulated distributions $p_{X|Q}^{\mathrm{post}}$ and postulated noise variance τ_{post}^w. By "exact," we mean the actual MMSE estimate for that postulated prior – not the GAMP or any other approximation. Then, in the limit of large i.i.d. random matrices $\mathbf{A}$, it is claimed that the joint distribution of (x_j, q_j) for some component j and the corresponding exact MMSE estimate $\widehat{x}_j^{\mathrm{post}}$, follows the same scalar model as Fig. 2 for the GAMP algorithm. Moreover, the effective noise variance of the exact MMSE estimator is a fixed point of the updates (67) and (68). This result of Guo and Verú generalized several earlier results including analyses of linear estimators in [51] and [48] based on random matrix theory and BPSK estimation in [17] based on the replica method. As discussed in [7], the SE analysis of the AMP algorithm provides some rigorous basis for the replica analysis, along with an algorithm to achieve the performance.

Also, when the postulated distribution matches the true distribution, the general equations (67) and (68) reduce to the SE equations for Gaussian approximated BP on large sparse matrices derived originally by Boutros and Caire [1] along with other works including [4] and [52]. In this regard, the analysis here can be seen as an extension of that work to handle dense matrices and the case of possibly mismatched distributions.

C. Max-Sum GAMP with AWGN Output Channels

Now suppose that the estimator uses an input function $g_{\mathrm{in}}(\cdot)$ based on the MAP estimator (18) again with a postulated distribution $p_{X|Q}^{\mathrm{post}}(\cdot)$ that may differ from the true distribution $p_{X|Q}(\cdot)$. Let $\widehat{x}_{\mathrm{map}}^{\mathrm{post}}(\widehat{r}, q, \tau^r)$ be the corresponding scalar MAP estimator

$$\widehat{x}_{\mathrm{map}}^{\mathrm{post}}(\widehat{r}, q, \tau^r) := \arg\max_{x \in \mathbb{R}} \left[f_{\mathrm{in}}(x, q) - \frac{1}{2\tau^r}(\widehat{r} - x)^2\right], \tag{69}$$

where
$$f_{\text{in}}(x,q) = \log p_{X|Q}^{\text{post}}(x|q).$$

With this definition, $\widehat{x}_{\text{map}}^{\text{post}}$ is the scalar MAP estimate of the random variable X from the observations $\widehat{R} = \widehat{r}$ and $Q = q$ in the model (65). Also, following (20) define

$$\mathcal{E}_{\text{map}}^{\text{post}}(\widehat{r}, q, \tau^r) := \frac{\tau^r}{1 - f_{\text{in}}''(\widehat{x}, q)\tau^r},$$

where $\widehat{x} = \widehat{x}_{\text{map}}^{\text{post}}$. Then, (18) and (20) can be re-written as

$$g_{\text{in}}(\widehat{r}, q, \tau^r) = \widehat{x}_{\text{map}}^{\text{post}}(\widehat{r}, q, \tau^r), \qquad (70\text{a})$$

and

$$\tau^r \frac{\partial}{\partial \widehat{r}} g_{\text{in}}(\widehat{r}, q, \tau^r) = \mathcal{E}_{\text{map}}^{\text{post}}(\widehat{r}, q, \tau^r). \qquad (70\text{b})$$

Following similar computations to the previous subsection, we obtain the SE equations

$$\overline{\tau}^r(t{+}1) = \tau_{\text{post}}^w + \beta \mathbb{E}\left[\mathcal{E}_{\text{map}}^{\text{post}}(\widehat{R}, Q, \overline{\tau}^r(t))\right] \qquad (71\text{a})$$

$$\xi^r(t{+}1) = \tau^w + \beta \mathbb{E}\left[X - \widehat{x}_{\text{map}}^{\text{post}}(\widehat{R}, Q, \overline{\tau}^r(t))\right]^2 (71\text{b})$$

Similar to the MMSE case, the fixed point of these equations precisely agree with the equations for replica analysis of MAP estimation with AWGN output noise given in [53] and related works in [19]. Thus, again, the GAMP framework provides a rigorous justification of the replica results along with an algorithm to achieve the predictions by the replica method.

D. General Output Channels with Matched Distributions

Finally, let us return to the case of general (possibly non-AWGN) output channels and consider the sum-product GAMP algorithm with the estimations functions in Section IV-B. In this case, we will assume that the postulated distributions for $p_{X|Q}$ and $p_{Y|Z}$ match the true distributions. Guo and Wang in [5] derived SE equations for standard BP in this case for large sparse random matrices. The work [8] derived identical equations for a relaxed version of BP. We will see that the same SE equations will follow as a special case of the more general SE equations above.

To prove this, we will show by induction that, for all t, $\mathbf{K}^x(t)$ has the form

$$\mathbf{K}^x(t) = \begin{bmatrix} \tau^{x0} & \tau^{x0} - \overline{\tau}^x(t) \\ \tau^{x0} - \overline{\tau}^x(t) & \tau^{x0} - \overline{\tau}^x(t) \end{bmatrix}, \qquad (72)$$

where τ^{x0} is the variance of X. For $t = 0$, recall that for MMSE estimation, $\overline{\tau}^x(0) = \tau^{x0}$ and $\widehat{x}_j(0) = 0$ (the mean of X) for all j. Therefore, (52) shows that (72) holds for $t = 0$.

Now suppose that (72) holds for some t. Then, (53a) shows that $\mathbf{K}^p(t)$ has the form

$$\mathbf{K}^p(t) = \begin{bmatrix} \tau^{z0} & \tau^{z0} - \overline{\tau}^p(t) \\ \tau^{z0} - \overline{\tau}^p(t) & \tau^{z0} - \overline{\tau}^p(t) \end{bmatrix}, \qquad (73)$$

where $\tau^{z0} = \beta\tau^{x0}$. Now, if $(Z, \widehat{P}) \sim \mathcal{N}(0, \mathbf{K}^p(t))$ with $\mathbf{K}^p(t)$ given by (73), then the conditional distribution of Z given $\widehat{P}$ is $Z \sim \mathcal{N}(\widehat{P}, \overline{\tau}^p(t))$. Therefore, $g_{\text{out}}(\cdot)$ in (38) can be interpreted as

$$g_{\text{out}}(\widehat{p}, y, \tau^p) := \frac{\partial}{\partial \widehat{p}} \log p_{Y|\widehat{P}}(y|\widehat{P} = \widehat{p}), \qquad (74)$$

where $p_{Y|\widehat{P}}(\cdot|\cdot)$ is the likelihood of Y given $\widehat{P}$ for the random variables $\theta^p(\mathbf{K}^p(t))$ in (50).

Now let $F(\overline{\tau}^p(t))$ be the *Fisher information*

$$F(\overline{\tau}^p(t)) := \mathbb{E}\left[\frac{\partial}{\partial \widehat{p}} \log p_{Y|\widehat{P}}(Y|\widehat{P})\right]^2, \qquad (75)$$

where the expectation is also over $\theta^p(\mathbf{K}^p(t))$ with $\mathbf{K}^p(t)$ given by (73). A standard property of the Fisher information is that

$$F(\overline{\tau}^p(t)) = \mathbb{E}\left[\frac{\partial^2}{\partial \widehat{p}^2} \log p_{Y|\widehat{P}}(Y|\widehat{P})\right]. \qquad (76)$$

Substituting (75) and (76) into (53) we see that

$$\overline{\tau}^r(t) = \xi^r(t) = \frac{1}{F(\overline{\tau}^p(t))}. \qquad (77)$$

We will show in Appendix E that

$$\mathbb{E}\left[\frac{\partial}{\partial \widehat{p}} g_{\text{out}}(t, h(Z, W), \widehat{P}, \overline{\tau}^p(t))\right]$$
$$= \mathbb{E}\left[\frac{\partial}{\partial z} g_{\text{out}}(t, h(Z, W), \widehat{P}, \overline{\tau}^p(t))\right]. \qquad (78)$$

It then follows from (53d) that

$$\alpha^r(t) = 1. \qquad (79)$$

Now consider the input update (54). Since $\alpha^r(t) = 1$ and $\overline{\tau}^r(t) = \xi^r(t)$, the expectation in (31) agrees with the expectations in (54). Substituting (32) into (54a), we obtain

$$\overline{\tau}^x(t{+}1) = \mathbb{E}\left[\text{var}(X|Q, \widehat{R})\right], \qquad (80)$$

where the expectation is over $\theta^r(\xi^r(t), \alpha^r(t))$ in (48) with $\alpha = 1$. Also, if $\widehat{X}(t{+}1) = \mathbb{E}(X|Q, \widehat{R})$ then

$$\mathbb{E}(X - \widehat{X}(t{+}1))^2 = \mathbb{E}\left[\text{var}(X|Q, \widehat{R})\right] = \overline{\tau}^x(t{+}1)$$
$$\mathbb{E}\left[\widehat{X}(t{+}1)(X - \widehat{X}(t{+}1))\right] = 0.$$

These relations, along with the definition $\tau^{x0} = \mathbb{E}(X^2)$, show that the covariance $\mathbf{K}^x(t{+}1)$ in (54b) is of the form (72). This completes the induction argument to show that (72) and the other equations above hold for all t.

The updates (77) and (80) are precisely the SE equations given in [5] and [8] for sum-product BP estimation with matched distributions on large sparse random matrices. The current work thus shows that the identical SE equations hold for dense matrices $\mathbf{A}$. In addition, the results here extend the earlier SE equations by considering the cases where the distributions postulated by the estimator may not be identical to the true distribution.

VII. NONLINEAR COMPRESSED SENSING EXAMPLE

To validate the GAMP algorithm and its SE analysis, we considered the following simple example of a nonlinear compressed sensing problem. The distribution on the components of the input, $\mathbf{x}$, is taken as a Bernoulli-Gaussian:

$$x_j \sim \begin{cases} 0 & \text{prob} = 1 - \rho, \\ \mathcal{N}(0, 1) & \text{prob} = \rho, \end{cases} \qquad (81)$$
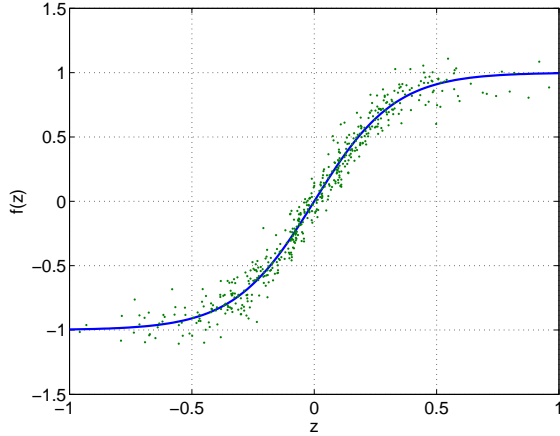
Fig. 3.   Sigmoidal output function for the nonlinear compressed sensing example, along with a scatter plot of the noisy points (z_i, y_i) in one typical realization.
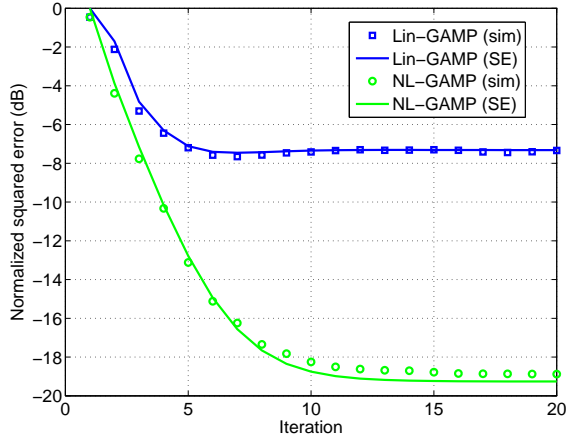

Fig. 4.   Normalized squared error of the of the sparse vector estimates in the nonlinear compressed sensing problem. Plotted are the median squared errors for Monte Carlo trials of the GAMP algorithm using both a linear approximation of the output (Lin-GAMP) and the true nonlinear output (NL-GAMP). Also plotted are state evolution (SE) predictions of the asymptotic performance.

where, in the experiments below, we set $\rho = 0.1$. The distribution (81) provides a simple model of a sparse vector with ρ being the fraction of non-zero components. To match the state evolution (SE) theory, the linear transform $\mathbf{A} \in \mathbb{R}^{m \times n}$ is generated as an i.i.d. Gaussian matrix with $A_{ij} \sim \mathcal{N}(0, 1/n)$. For the output channel, we consider nonlinear measurements of the form

$$y_i = f(z_i) + w_i, \qquad f(z) = 1/(1 + \exp(-az)), \quad (82)$$

and $w_i \sim \mathcal{N}(0, \tau^w)$. The function $f(z)$ in (82) is a sigmoidal function that arises commonly in neural networks and provides a good test of the ability of GAMP to handle nonlinear measurements. The output variance is taken as $\tau^w = 0.01$, or 20 dB below the full range of the output of $f(z)$. The scale factor in (82) is set to $a = 6.1$. The output function $f(z)$ along with a scatter plot of the noisy points (z_i, y_i) is shown in Fig. 3. Since we are recovering a sparse vector $\mathbf{x}$ from a linear transform followed by noisy, nonlinear measurements, we can consider the problem a nonlinear compressed sensing

problem. We set $(m, n) = (500, 1000)$ so that the sparse vector is undersampled by a factor of two.

For this problem, we consider two estimators, both based on the GAMP algorithm:

- *NL-GAMP:* The sum-product GAMP algorithm matched to the true input and output distributions.
- *Lin-GAMP:* The sum-product GAMP algorithm matched to the true input distribution, but the estimator assumes a linear channel of the form

$$y_i = f'(0)z_i + w_i, \qquad w_i \sim \mathcal{N}(0, \tau^w).$$

Since the estimator Lin-GAMP assumes an AWGN output channel, it is equivalent to the Bayesian-AMP estimator of [37], [38]. For both algorithms, we use the full algorithm, Algorithm 1. Other simulations, not reported here, show virtually no difference to the simplified algorithm, Algorithm 2.

Fig. 4 plots the normalized squared error for both algorithms over 100 Monte Carlo simulations. In each Monte Carlo simulation we computed the normalized squared error (NSE),

$$\text{NSE} = 10 \log_{10}(\mathbb{E}\|\mathbf{x} - \widehat{\mathbf{x}}\|^2 / \mathbb{E}\|\mathbf{x}\|^2),$$

where the expectation is over the components of the vectors. Fig. 4 then plots the median normalized squared error over the 100 trials. The median is used since there is a significant variation from between trials, and the median removes outliers.

The first point to notice is that there is a significant gain from incorporating the nonlinearity in the output relative to using simple linear approximations. Indeed, NL-GAMP shows an asymptotic gain of over 11 dB relative to the Lin-GAMP method that approximates the output as a linear function. Secondly, we see that both algorithms converge very quickly, within approximately 10 to 15 iterations.

Finally, we see that for both methods, the state evolution (SE) analysis predicts the per iteration performance extremely well. For Lin-GAMP, the SE predicts the median SE within 0.2 dB and for NL-GAMP, the prediction is within 0.4 dB. Although the SE analysis theoretically only provides predictions for a modified version of the simplified Algorithm 2, we see that the prediction holds, at least in this simulation, for the full algorithm, Algorithm 1.

Also note that since NL-GAMP is the sum-product GAMP algorithm where the postulated and true distributions are matched, the SE equations fall under the special case given in Section VI-D. The SE equations in this special case were derived for sparse matrices in [5] and [8]. However, since the Lin-GAMP is applied to a nonlinear output where the true and postulated distributions are not matched, there are no previous SE analyses that can predict the performance of the method. Interestingly, the squared error for Lin-GAMP does not monotonically decrease; There is a slight increase in squared error after iteration 7 and the increase is actually predicted by the SE analysis.

The simulation thus demonstrates that the GAMP method can provide a tractable approach to a difficult nonlinear compressed sensing problem with significant gains over AMP methods based on naïve linear approximations. Moreover,

the SE analysis can precisely predict the performance of the method and quantify the value of incorporating the nonlinearity. All code for the simulation is available in the sourceforge GAMP repository [54].

Conclusions and Future Work

We have considered a general linear mixing estimation problem of estimating an i.i.d. (possibly non-Gaussian) vector observed through a linear transform followed by a componentwise (possibly random and nonlinear) measurements. The formulation is extremely general and encompasses many problems in compressed sensing, classification, learning and signal processing. We have presented a novel algorithm, called GAMP, that unifies several earlier methods to realize computationally efficient and systematic approximations of max-sum and sum-product loopy BP. Our main theoretical contribution is an extension of Bayati and Montanari's state evolution (SE) analysis in [7] to precisely describes the asymptotic behavior of the GAMP algorithm for large Gaussian i.i.d. matrices. The GAMP algorithm thus provides a computationally simple method that can apply to a large class of estimation problems with provable exact performance characterizations. As a result, we believe that the GAMP algorithm can have wide ranging applicability. Indeed, applications of the GAMP methodology have been used in nonlinear wireless scheduling problems [55], compressed sensing with quantization [56], and neural estimation [57] to name a few.

Nevertheless, there are a large number of open issues for future research, some of which have been begun consideration since the original publication of this paper in [58].

Non-Gaussian matrices: Our SE analysis currently only applies to Gaussian i.i.d. matrices and a significant open question is whether the analyses can be extended to more general matrices. Recently, [59] extended the replica analysis in [17], [18] to obtain predictions of the behavior of optimal estimators for linear mixing problems with general free matrices. One avenue of future research is to see if a AMP-like algorithm can be constructed that can provably obtain the performance predicted for these matrices.

Learning Distributions: One of the main limitations of the GAMP method is that both the sum-product and max-sum variants require that the true distributions on the input an output channels are known. When the distributions are not known, one approach is to find a minimax estimator over a class of distributions, as was performed for classes of sparse priors in [60]. A second approach is to attempt to adaptively estimate the distributions assuming some parametric model. One of the most promising methods in this regard is to combine GAMP with expectation-maximization (EM) estimation as discussed in [20], [21], [50], [61], [62].

Connections to graphical models: Since the GAMP method is derived from graphical model techniques, it can be incorporated as a component of a larger graphical model where the approximate message passing is combined with standard belief propagation updates. Such hybrid approaches have been explored in a number of works including [63]–[67] for extending compressed sensing problems with other

statistical dependencies between components or estimation of additional latent variables.

Optimality: A significant outstanding theoretical issue is to obtain a performance lower bound. An appealing feature of the SE analysis of sparse random matrices such as [1], [4], [5], [8] as well as well as the replica analysis in [17], [18], [42] is that they provide lower bounds on the performance of the optimal estimator. These lower bounds are also described by SE equations. Then, using a sandwiching argument – a technique used commonly in the study of LDPC codes [68] – shows that when fixed points of the SE equations are unique, BP is optimal. Finding a similar lower bound for the GAMP algorithm is a possible avenue of future research.

Appendix A
Vector-Valued AMP

A. SE Equations for Vector-Valued AMP

The analysis of the GAMP requires a vector-valued version of the recursion analyzed by Bayati and Montanari. Fix dimensions n_d and n_b, and let Θ^u and Θ^v be any two sets. Let $G_{\text{in}}(t, \mathbf{d}, \theta^u) \in \mathbb{R}^{n_b}$ and $G_{\text{out}}(t, \mathbf{b}, \theta^v) \in \mathbb{R}^{n_d}$ be vector-valued functions over the arguments $t = 0, 1, 2, \ldots$, $\mathbf{b} \in \mathbb{R}^{n_b}$, $\mathbf{d} \in \mathbb{R}^{n_d}$, $\theta^u \in \Theta^u$ and $\theta \in \Theta^v$. Assume $G_{\text{in}}(t, \mathbf{d}, \theta^u)$ and $G_{\text{out}}(t, \mathbf{b}, \theta^v)$ are Lipschitz continuous in $\mathbf{d}$ and $\mathbf{b}$, respectively. Let $\mathbf{A} \in \mathbb{R}^{m \times n}$ and generate a sequence of vectors $\mathbf{b}_i(t)$ and $\mathbf{d}_j(t)$ by the iterations

$$\mathbf{b}_i(t) = \sum_j a_{ij} \mathbf{u}_j(t) - \boldsymbol{\lambda}(t) \mathbf{v}_i(t-1), \tag{83a}$$

$$\mathbf{d}_j(t) = \sum_i a_{ij} \mathbf{v}_i(t) - \boldsymbol{\xi}(t) \mathbf{u}_j(t) \tag{83b}$$

where

$$\mathbf{u}_j(t+1) = G_{\text{in}}(t, \mathbf{d}_j(t), \theta_j^u), \tag{84a}$$

$$\mathbf{v}_i(t) = G_{\text{out}}(t, \mathbf{b}_i(t), \theta_i^v), \tag{84b}$$

and

$$\boldsymbol{\xi}(t) = \frac{1}{m} \sum_{i=1}^{m} \frac{\partial}{\partial \mathbf{b}} G_{\text{out}}(t, \mathbf{b}_i(t), \theta_i^v) \tag{85a}$$

$$\boldsymbol{\lambda}(t+1) = \frac{1}{m} \sum_{j=1}^{n} \frac{\partial}{\partial \mathbf{d}} G_{\text{in}}(t, \mathbf{d}_j(t), \theta_j^v). \tag{85b}$$

Here, we interpret the derivatives as matrices $\boldsymbol{\xi}(t) \in \mathbb{R}^{n_d \times n_b}$ and $\boldsymbol{\lambda}(t+1) \in \mathbb{R}^{n_b \times n_d}$. The recursion is initialized with $t = 0$, $\mathbf{v}_i(t-1) = 0$ and some values for $\mathbf{u}_j(0)$.

Now similar to Section V, consider a sequence of random realizations of the parameters indexed by the input dimension n. For each n, we assume that the output dimension $m =$

$m(n)$ is deterministic and scales linearly as in (45) for some $\beta \geq 0$. Assume that the transform matrix $\mathbf{A}$ has i.i.d. Gaussian components $a_{ij} \sim \mathcal{N}(0, 1/m)$. Also assume that the following components converge empirically with bounded moments of order $2k - 2$ with the limits

$$\lim_{n \to \infty} \theta_i^v \stackrel{d}{=} \theta^v, \quad \lim_{n \to \infty} \theta_j^u \stackrel{d}{=} \theta^u, \quad (86a)$$

$$\lim_{n \to \infty} \mathbf{u}_i(0) \stackrel{d}{=} \mathbf{U}_0, \quad (86b)$$

for random variables θ^v, θ^u and U_0. We also assume $\mathbf{U}_0$ is zero mean.

Under these assumptions, we will argue that the SE equations for the vector-valued recursion are given by:

$$\mathbf{K}^d(t)$$
$$:= \mathbb{E}\left[G_{\text{out}}(t, \mathbf{B}(t), \theta^v) G_{\text{out}}(t, \mathbf{B}(t), \theta^v)^T \right] \quad (87a)$$
$$\mathbf{K}^b(t+1)$$
$$:= \beta \mathbb{E}\left[G_{\text{in}}(t, \mathbf{D}(t), \theta^u) G_{\text{in}}(t, \mathbf{D}(t), \theta^u)^T \right] \quad (87b)$$

where the expectations are over the random variables θ^u and θ^v in the limits (86), and $\mathbf{B}(t)$ and $\mathbf{D}(t)$ are Gaussian vectors

$$\mathbf{B}(t) \sim \mathcal{N}(0, \mathbf{K}^b(t)), \quad \mathbf{D}(t) \sim \mathcal{N}(0, \mathbf{K}^d(t)) \quad (88)$$

independent of θ^u and θ^v. The SE equations (87) are initialized with

$$\mathbf{K}^b(0) := \beta \mathbb{E}\left[\mathbf{U}_0 \mathbf{U}_0^T \right]. \quad (89)$$

The matrices in (85) are derived *empirically* from the variables $\mathbf{b}(t)$ and $\mathbf{d}(t)$. We will also be interested in the case where the algorithm directly uses the *expected* values

$$\boldsymbol{\xi}(t) = \mathbb{E}\left[\frac{\partial}{\partial \mathbf{b}} G_{\text{out}}(t, \mathbf{B}(t), \theta^v) \right] \quad (90a)$$

$$\boldsymbol{\lambda}(t+1) = \mathbb{E}\left[\frac{\partial}{\partial \mathbf{d}} G_{\text{in}}(t, \mathbf{D}(t), \theta^v) \right], \quad (90b)$$

where, again, the expectations are over random variables θ^u and θ^v in the limits (86), and $\mathbf{B}(t)$ and $\mathbf{D}(t)$ are Gaussian vectors in (88) independent of θ^u and θ^v.

Claim 2: Consider the recursion in (83) and (84) with either the empirical update (85) or expected update (90). Then, under the above assumptions, for any fixed iteration number t, the variables in the recursions with either the empirical or expected update converge empirically as

$$\lim_{n \to \infty} (\mathbf{d}_j(t), \theta_j^u) \stackrel{d}{=} (\mathbf{D}(t), \theta^u) \quad (91a)$$

$$\lim_{n \to \infty} (\mathbf{b}_i(t), \theta_i^v) \stackrel{d}{=} (\mathbf{B}(t), \theta^v), \quad (91b)$$

where are θ^u and θ^v are the random variables in the limits (86), and $\mathbf{B}(t)$ and $\mathbf{D}(t)$ are Gaussians (88) independent of θ^u and θ^v.

The scalar case when $n_b = n_d = 1$ is rigorously proven by Bayati and Montanari [7]. The modifications for the vector-valued case is straightforward but tedious. However, we only provide a sketch of the arguments in Appendix F. As discussed above, a full re-derivation of the proof from [7] would be long and beyond the scope of the paper. Thus, the result is not fully rigorous, and we use the term Claim instead of Theorem to emphasize the lack of rigor. A complete proof would be a valuable avenue of future work.

APPENDIX B
PROOF OF CLAIM 1

Claim 1 is a special case of the general result, Claim 2, above. Although we have not provided a complete proof of Claim 2, the implication from Claim 2 to 1 is completely rigorous.

Let $n_b = 2$ and $n_d = 1$ and define the variables

$$\mathbf{u}_j(t) = \left[\begin{array}{c} x_j \\ \widehat{x}_j(t) \end{array} \right], \quad \mathbf{b}_i(t) = \left[\begin{array}{c} z_i \\ \widehat{p}_i(t) \end{array} \right] \quad (92a)$$

$$v_i(t) = \widehat{s}_i(t) \quad (92b)$$

$$d_j(t) = \frac{1}{\overline{\tau}^r(t)}(\widehat{r}_j(t) - \alpha^r(t)x_j) \quad (92c)$$

$$\theta_j^u = (x_j, q_j), \quad \theta_i^v = w_i \quad (92d)$$

$$\boldsymbol{\lambda}(t+1) = \left[\begin{array}{c} 0 \\ \tau^p(t) \end{array} \right] \quad (92e)$$

$$\boldsymbol{\xi}(t) = \frac{1}{\overline{\tau}^r(t)}\left[\alpha^r(t) \; -1 \right]. \quad (92f)$$

Also, for $\theta^u = (x, q)$, $\theta^v = w$ and $\mathbf{b} = (z \; \widehat{p})^T$, define the functions

$$G_{\text{in}}(t, d, \theta^u)$$
$$:= \left[\begin{array}{c} x \\ g_{\text{in}}(t, \overline{\tau}^r(t)d + \alpha^r(t)x, q, \overline{\tau}^r(t)) \end{array} \right], \quad (93a)$$

and

$$G_{\text{out}}(t, \mathbf{b}, \theta^v) := g_{\text{out}}(t, \widehat{p}, h(z, w), \overline{\tau}^p(t)). \quad (93b)$$

With these definitions, it is easily checked that simplified GAMP algorithm, Algorithm 2, with the modifications in Section V-D agrees with the recursion described by equations (83), and (84), with $\boldsymbol{\lambda}(t)$ being defined with the empirical update in (85) and $\boldsymbol{\xi}(t)$ being defined by the expected value in (90). For example,

$$\mathbf{b}_i(t) \stackrel{(a)}{=} \left[\begin{array}{c} z_i \\ \widehat{p}_i(t) \end{array} \right]$$
$$\stackrel{(b)}{=} \sum_j a_{ij} \left[\begin{array}{c} x_j \\ \widehat{x}_j(t) \end{array} \right] - \left[\begin{array}{c} 0 \\ \overline{\tau}^p(t)\widehat{s}_i(t-1) \end{array} \right]$$
$$\stackrel{(c)}{=} \sum_j a_{ij}\mathbf{u}_j(t) - \boldsymbol{\lambda}(t)v_i(t-1),$$

where (a) follows from the definition of $b_i(t)$ in (92a); (b) follows and (9) and the fact that $\mathbf{z} = \mathbf{A}\mathbf{x}$ and (c) follows from the remaining definitions in (92). Therefore, the variables in (92) satisfy (83a). Similarly,

$$d_j(t) \stackrel{(a)}{=} \frac{1}{\overline{\tau}^r(t)}(\widehat{r}_j(t) - \alpha^r(t)x_j)$$
$$\stackrel{(b)}{=} \sum_i a_{ij}\widehat{s}_i(t) + \frac{1}{\overline{\tau}^r(t)}(\widehat{x}_j(t) - \alpha^r(t)x_j)$$
$$\stackrel{(c)}{=} \sum_i a_{ij}v_i(t) - \boldsymbol{\xi}(t)\mathbf{u}_j(t),$$

where (a) follows from the definition of $d_j(t)$ in (92c); (b) follows from (11b) with the modification that $\tau^r(t)$ is replaced with $\overline{\tau}^r(t)$; and (c) follows from the other definitions in (92)

Hence the variables in (92) satisfy (83b). The equations in (84) can also easily verified.

We next consider $\boldsymbol{\lambda}(t)$ and $\boldsymbol{\xi}(t)$. For $\boldsymbol{\lambda}(t)$, first observe that (92c) shows that

$$\frac{\partial}{\partial d} g_{\text{in}}(t, \overline{\tau}^r(t)d + \alpha^r(t)x_j, q_j, \overline{\tau}^r(t))|_{d=d_j(t)}$$
$$= \overline{\tau}^r(t)\frac{\partial}{\partial \widehat{r}} g_{\text{in}}(t, \widehat{r}_j(t), q_j, \overline{\tau}^r(t)). \quad (94)$$

Hence

$$\tau^p(t+1) \stackrel{(a)}{=} \frac{n}{m}\overline{\tau}^r(t+1)$$

$$= \frac{1}{n}\sum_{j=1}^{n} \overline{\tau}^r(t)\frac{\partial}{\partial \widehat{r}} g_{\text{in}}(t, \widehat{r}_j(t), q_j, \overline{\tau}^r(t)) \quad (95)$$

where (a) follows from (9a) with the modification that $\|\mathbf{A}\|_F^2$ has been replaced by its expectation $\mathbb{E}\|\mathbf{A}\|_F^2 = n$ and (b) follows from (12b) with the modification that $\tau^r(t)$ is replaced by $\overline{\tau}^r(t)$. Combining (94) and (95), with the definition of $G_{\text{in}}(\cdot)$ in (93a), we see that $\boldsymbol{\lambda}(t)$ in (92e) satisfies (85b).

Similarly, for $\boldsymbol{\xi}(t)$, observe that (53d) and (53b) show that

$$\mathbb{E}\left[\frac{\partial}{\partial z} g_{\text{out}}(t, \widehat{P}, h(z, W), \overline{\tau}^p(t))\bigg|_{z=Z}\right] = \frac{\alpha^r(t)}{\overline{\tau}^r(t)}$$

$$\mathbb{E}\left[\frac{\partial}{\partial \widehat{p}} g_{\text{out}}(t, \widehat{p}, h(Z, W), \overline{\tau}^p(t))\bigg|_{\widehat{p}=P}\right] = \frac{-1}{\overline{\tau}^r(t)}.$$

Combining these relations with the definition of $G_{\text{out}}(\cdot)$ in (93b), shows that $\boldsymbol{\xi}(t)$ defined in (92f) satisfies (90).

Therefore, we can apply Claim 2 which shows that the limits (91) hold for the matrices $\mathbf{K}^b(t)$ and $\mathbf{K}^d(t)$ from the SE equations (87) with initial condition (89). Since the definitions in (92) set $\theta_j^u = (x_j, q_j)$ and $\theta_i^v = w_i$, the expectations in (87) are over the random variables

$$\theta^u \sim (X, Q), \quad \theta^v \sim W, \quad (96)$$

where (X, Q) and W are the limiting random variables in Section V-B.

Now using the SE equations (52), (53) and (54), one can easily show by induction that

$$\mathbf{K}^p(t) = \mathbf{K}^b(t) = \beta\mathbf{K}^x(t) \quad (97a)$$
$$\xi^r(t) = (\overline{\tau}^r(t))^2\mathbf{K}^d(t). \quad (97b)$$

For example, one can show that (97a) holds for $t = 0$ by comparing the initial conditions (89) with (52) and using the definition of $\mathbf{u}(0)$ in (92a). Now, suppose that (97a) holds or some t. Then, (97a) and (96) show that $(\mathbf{B}(t), \theta^v)$ in the expectation (87b) is identically distributed to $((Z, \widehat{P}), W)$ in $\theta^p(\mathbf{K}^p(t))$. Therefore,

$$\xi^r(t) \stackrel{(a)}{=} (\overline{\tau}^r(t))^2\mathbb{E}\left[g_{\text{out}}^2(t, Y, \widehat{P}, \overline{\tau}^p(t))\right]$$
$$\stackrel{(b)}{=} (\overline{\tau}^r(t))^2\mathbb{E}\left[G_{\text{out}}^2(t, \mathbf{B}(t), \theta^v)\right]$$
$$\stackrel{(c)}{=} (\overline{\tau}^r(t))^2\mathbf{K}^d(t),$$

where (a) follows from (53c); (b) follows from the definition of $\mathbf{b}(t)$ and θ^v in (92) and $G_{\text{out}}(\cdot)$ in (93b); and (c) follows

from (87a). Similarly, one can show that, if (97b) holds for some t, then (97a) holds for $t + 1$.

With these equivalences, we can now prove the assertions in Claim 1. To prove (56), first observe that the limit (91) along with (96) and the definitions in (92) show that

$$\lim_{n\to\infty}\left(\frac{1}{\overline{\tau}^r(t)}(\widehat{r}_j(t) - \alpha^r(t)x_j), x_j, q_j\right) \stackrel{d}{=} (D(t), X, Q),$$

where the limit is in the sense of empirical convergence of order k and $D(t) \sim \mathcal{N}(0, \mathbf{K}^d(t))$ is independent of (X, Q). But, this limit is equivalent to

$$\lim_{n\to\infty}(x_j, q_j, \widehat{r}_j(t)) \stackrel{d}{=} (X, Q, \widehat{R}), \quad (98)$$

where

$$\widehat{R} = \alpha^r(t)X + \overline{\tau}^r(t)D(t).$$

Since $D(t) \sim \mathcal{N}(0, \mathbf{K}^d(t))$,

$$\overline{\tau}^r(t)D(t) \sim \mathcal{N}(0, (\overline{\tau}^r(t))^2\mathbf{K}^d(t)) = \mathcal{N}(0, \xi^r(t)),$$

where the last equality is due to (97b). Therefore, $(X, Q, \widehat{R})$ in (98) is identically distributed to $\theta^r(\xi^r(t), \alpha^r(t))$ in (48). This proves (56), and part (b) of Claim 1. Part (c) of Claim 1 is proven similarly.

To prove (55) in part (a),

$$\lim_{n\to\infty}\frac{1}{\tau^r(t)} \stackrel{(a)}{=} \lim_{n\to\infty}\tau^s(t)$$
$$\stackrel{(b)}{=} -\lim_{n\to\infty}\frac{1}{m}\sum_{i=1}^{m}\frac{\partial}{\partial \widehat{p}} g_{\text{out}}(t, \widehat{p}_i(t), y_i, \overline{\tau}^p(t))$$
$$\stackrel{(c)}{=} -\mathbb{E}\left[\frac{\partial}{\partial \widehat{p}} g_{\text{out}}(t, \widehat{P}, Y, \overline{\tau}^p(t))\right]$$
$$\stackrel{(d)}{=} \frac{1}{\overline{\tau}^r(t)},$$

where (a) follows from (11a) with the modification that $\|\mathbf{A}\|_F^2 = n$, (b) follows from (10b) and that fact that we are considering the modified algorithm where $\tau^p(t)$ is replaced with $\overline{\tau}^p(t)$; (c) follows the limit (57) and the assumption that the derivative of $g_{\text{out}}(t, \widehat{p}, y, \overline{\tau}^p(t))$ is of order k; and (d) follows from (53). Similarly, one can show

$$\lim_{n\to\infty}\tau^p(t) = \overline{\tau}^p(t).$$

This proves (55) and completes the proof of Claim 1.

APPENDIX C
MAX-SUM GAMP

In this section, we show that with the functions g_{in} and g_{out} in (18) and (24), the GAMP algorithm can be seen heuristically as a first-order approximation of max-sum loopy BP for the MAP estimation problem. The derivations in this section are not rigorous, since we do not formally claim any properties of this approximation.
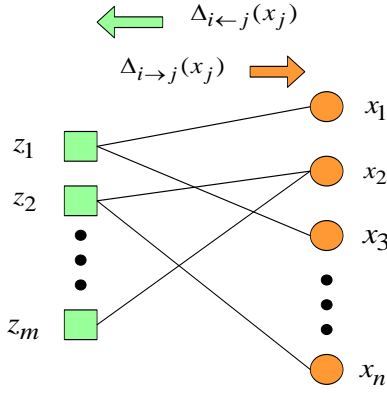
Fig. 5.   Factor or Tanner graph for the linear mixing estimation problem.

A. Max-Sum BP for MAP Estimation

We first review how we would apply standard max-sum loopy BP for the MAP estimation problem (16). For both the MAP and MMSE estimation problems, standard loopy BP operates by associating with the transform matrix $\mathbf{A}$ a bipartite graph $G = (V, E)$ called the *factor* or *Tanner* graph as illustrated in Fig. 5. The vertices V in this graph consists of n "input" or "variable" nodes associated with the variables x_j, $j = 1, \ldots, n$, and m "output" or "measurements" nodes associated with the transform outputs z_i, $i = 1, \ldots, m$. There is an edge $(i, j) \in E$ between the input node x_j and output node z_i if and only if $a_{ij} \neq 0$.

Now let $\Delta_j(x_j)$ be the marginal maxima in (17), which we can interpret as a "value" function on the variable x_j. Max-sum loopy BP iteratively sends messages between the input nodes x_j and output nodes z_i representing estimates of the value function $\Delta_j(x_j)$. The value message from the input node x_j to output node z_i in the tth iteration is denoted $\Delta_{i \leftarrow j}(t, x_j)$ and the reverse message for z_i to x_j is denoted $\Delta_{i \rightarrow j}(t, x_j)$.

Loopy BP updates the value messages with the following simple recursions: The messages from the output nodes are given by

$$\Delta_{i \rightarrow j}(t, x_j) = \text{const} \\ + \ \max_{\mathbf{x}} f_{\text{out}}(z_i, y_i) + \sum_{r \neq j} \Delta_{i \leftarrow r}(t, x_r) \qquad (99)$$

where the maximization is over vectors $\mathbf{x}$ with the jth component equal to x_j and $z_i = \mathbf{a}_i^T \mathbf{x}$, where $\mathbf{a}_i^T$ is the ith row of the matrix $\mathbf{A}$. The constant term is any term that does not depend on x_j, although it may depend on t or the indices i and j. The messages from the input nodes are given by

$$\Delta_{i \leftarrow j}(t+1, x_j) = \text{const} \\ + \ f_{\text{in}}(x_j, q_j) + \sum_{\ell \neq i} \Delta_{\ell \rightarrow j}(t, x_j) \qquad (100)$$

The BP iterations are initialized with $t = 0$ and $\Delta_{i \rightarrow j}(-1, x_j) = 0$.

The BP algorithm is terminated after some finite number of iterations. After the final tth iteration, the final estimate for x_j can be taken as the maximum of

$$\Delta_j(t+1, x_j) = f_{\text{in}}(x_j, q_j) + \sum_i \Delta_{i \rightarrow j}(t, x_j). \qquad (101)$$

B. Quadratic Legendre Transforms

To approximate the BP algorithm, we need the following simple result. Given a function $f : \mathbb{R} \rightarrow \mathbb{R}$, define the functions

$$\begin{align} (Lf)(x, r, \tau) &:= f(x) - \frac{1}{2\tau}(r - x)^2 & (102a) \\ (\Gamma f)(r, \tau) &:= \arg\max_x (Lf)(x, r, \tau) & (102b) \\ (\Lambda f)(r, \tau) &:= \max_x (Lf)(x, r, \tau) & (102c) \\ (\Lambda^{(k)} f)(r, \tau) &:= \frac{\partial^k}{\partial r^k}(\Lambda f)(r, \tau), & (102d) \end{align}$$

over the variables $r, \tau \in \mathbb{R}$ with $\tau > 0$ and $k = 1, 2, \ldots$. The function Λf can be interpreted as a quadratic variant of the standard Legendre transform of f [69].

Lemma 1: Let $f : \mathbb{R} \rightarrow \mathbb{R}$ be twice differentiable and assume that all the maximizations in (102) exist and are unique. Then,

$$\begin{align} (\Lambda^{(1)} f)(r, \tau) &= \frac{\widehat{x} - r}{\tau} & (103a) \\ (\Lambda^{(2)} f)(r, \tau) &= \frac{f''(\widehat{x})}{1 - \tau f''(\widehat{x})}, & (103b) \\ \frac{\partial}{\partial r}\widehat{x} &= \frac{1}{1 - \tau f''(\widehat{x})} & (103c) \end{align}$$

where $\widehat{x} = (\Gamma f)(r, \tau)$.

Proof: Equation (103a) follows from the fact that

$$(\Lambda^{(1)} f)(r, \tau) = \left. \frac{\partial L(r, x)}{\partial r} \right|_{x = \widehat{x}} = \frac{\widehat{x} - r}{\tau}.$$

Next observe that

$$\begin{align} \frac{\partial^2}{\partial x^2} L(x, r) &= f''(x) - \frac{1}{\tau} \\ \frac{\partial^2}{\partial x \partial r} L(x, r) &= \frac{1}{\tau}. \end{align}$$

So the derivative of $\widehat{x}$ is given by

$$\begin{align} \frac{\partial}{\partial r}\widehat{x} &= -\left[\frac{\partial^2}{\partial x^2} L(\widehat{x}, r) \right]^{-1} \frac{\partial^2}{\partial x \partial r} L(\widehat{x}, r) \\ &= \frac{1/\tau}{1/\tau - f''(x)} = \frac{1}{1 - \tau f''(x)}, \end{align}$$

which proves (103c). Hence

$$\begin{align} (\Lambda^{(2)} f)(r, \tau) &= \frac{\partial}{\partial r}(\Lambda^{(1)} f)(r, \tau) \\ &= \frac{\partial}{\partial r}\left[\frac{\widehat{x} - r}{\tau} \right] \\ &= \frac{1}{\tau}\left(\frac{1}{1 - \tau f''(\widehat{x})} - 1 \right) = \frac{f''(\widehat{x})}{1 - \tau f''(\widehat{x})}, \end{align}$$

which shows (103b). ∎

C. GAMP Approximations

We now show that the GAMP algorithm with the functions in (18) and (24) can be seen as a quadratic approximation of the max-sum loopy BP updates (100). The derivation is similar

to the one given in [70] for the Laplacian AMP algorithm [6]. We begin by considering the output node update (99). Let

$$\widehat{x}_j(t) \; := \; \arg\max_{x_j} \Delta_j(t, x_j), \tag{104a}$$

$$\widehat{x}_{i\leftarrow j}(t) \; := \; \arg\max_{x_j} \Delta_{i\leftarrow j}(t, x_j), \tag{104b}$$

$$\frac{1}{\tau_j^x(t)} \; := \; -\frac{\partial^2}{\partial x_j^2} \Delta_j(t, x_j)|_{x_j=\widehat{x}_j(t)}. \tag{104c}$$

$$\frac{1}{\tau_{i\leftarrow j}^x(t)} \; := \; -\frac{\partial^2}{\partial x_j^2} \Delta_{i\leftarrow j}(t, x_j)|_{x_j=\widehat{x}_{i\leftarrow j}(t)}. \tag{104d}$$

Now, for small a_{ir}, the values of x_r in the maximization (99) will be close to $\widehat{x}_{i\leftarrow j}(t)$. So, we can approximate each term $\Delta_{i\leftarrow r}(t, x_r)$ with the second order approximation

$$\Delta_{i\leftarrow r}(t, x_r) \approx \Delta_{i\leftarrow r}(t, \widehat{x}_{i\leftarrow r}(t)) - \frac{1}{2\tau_r^x(t)}(x_r - \widehat{x}_{i\leftarrow r}(t))^2, \tag{105}$$

where we have additionally made the approximation $\tau_{i\leftarrow r}^x(t) \approx \tau_r^x(t)$ for all i. Now, given x_j and z_i, consider the minimization

$$J := \min_{\mathbf{x}} \sum_{r \neq j} \frac{1}{2\tau_r}(x_r - \widehat{x}_{i\leftarrow r})^2, \tag{106}$$

subject to

$$z_i = a_{ij}x_j + \sum_{r \neq j} a_{ir}x_r.$$

A standard least squares calculation shows that the minimization (106) is given by

$$J = \frac{1}{2} \sum_{r \neq j} \frac{1}{2\tau_{i\rightarrow j}^p} \left(z_i - \widehat{p}_{i\rightarrow j} - a_{ij}x_j\right)^2$$

where

$$\widehat{p}_{i\rightarrow j} = \sum_{r \neq j} a_{ir}\widehat{x}_r, \quad \tau_{i\rightarrow j}^p = \sum_{r \neq j} |a_{ir}|^2 \tau_r^x.$$

So, the approximation (105) reduces to

$$\Delta_{i\rightarrow j}(t, x_j)$$
$$\approx \max_{z_i} \left[f_{\mathrm{out}}(z_i, y_i) - \frac{1}{2\tau_{i\rightarrow j}^p(t)}(z_i - \widehat{p}_{i\rightarrow j}(t) - a_{ij}x_j)^2, \right]$$
$$+ \mathrm{const}$$
$$= \; H\left(\widehat{p}_{i\rightarrow j}(t) + a_{ij}x_j, y_i, \tau_{i\rightarrow j}^p(t)\right) + \mathrm{const} \tag{107}$$

where

$$\widehat{p}_{i\rightarrow j}(t) \; := \; \sum_{r \neq j} a_{ir}\widehat{x}_{i\leftarrow r}(t) \tag{108a}$$

$$\tau_{i\rightarrow j}^p(t) \; := \; \sum_{r \neq j} |a_{ir}|^2 \tau_r^x(t), \tag{108b}$$

and

$$H(\widehat{p}, y, \tau^p) := \max_z \left[f_{\mathrm{out}}(z, y) - \frac{1}{2\tau^p}(z - \widehat{p})^2 \right]. \tag{109}$$

The constant term in (107) does not depend on z_i. Now let

$$\widehat{p}_i(t) := \sum_j a_{ij}\widehat{x}_{i\leftarrow j}(t), \tag{110}$$

and $\tau_i^p(t)$ be given as in (5a). Then it follows from (108) that

$$\widehat{p}_{i\rightarrow j}(t) \; = \; \widehat{p}_i(t) - a_{ij}\widehat{x}_{i\leftarrow j}(t) \tag{111a}$$

$$\tau_{i\rightarrow j}^p(t) \; = \; \tau_i^p(t) - a_{ij}^2 \tau_j^x(t). \tag{111b}$$

Using (111) and neglecting terms of order $O(a_{ij}^2)$, (107) can be further approximated as

$$\Delta_{i\rightarrow j}(t, x_j) \approx H\left(\widehat{p}_i(t) + a_{ij}(x_j - \widehat{x}_j), y_i, \tau_i^p(t)\right) + \mathrm{const}. \tag{112}$$

Now let

$$g_{\mathrm{out}}(\widehat{p}, y, \tau^p) := \frac{\partial}{\partial \widehat{p}} H(\widehat{p}, y, \tau^p). \tag{113}$$

Comparing $H(\cdot)$ in (109) with the definitions in (102) and using the properties in (103), it can be checked that

$$H(\widehat{p}, y, \tau^p) = (\Lambda f_{\mathrm{out}}(\cdot, y))(\widehat{p}, \tau^p),$$

and that $g_{\mathrm{out}}(\cdot)$ in (113) and its derivative agree with the definitions in (24) and (27).

Now define $\widehat{s}_i(t)$ and $\tau_i^s(t)$ as in (6). Then, a first order approximation of (112) shows that

$$\Delta_{i\rightarrow j}(t, x_j) \approx \mathrm{const}$$
$$+ s_i(t)a_{ij}(x_j - \widehat{x}_j(t)) - \frac{\tau_i^s(t)}{2}a_{ij}^2(x_j - \widehat{x}_j(t))^2.$$
$$= \; \mathrm{const} \left[s_i(t)a_{ij} + a_{ij}^2 \tau_i^s(t)\widehat{x}_j(t) \right] x_j$$
$$- \frac{\tau_i^s(t)}{2}a_{ij}^2 x_j^2, \tag{114}$$

where again the constant term does not depend on x_j. We next consider the input update (100). Substituting the approximation (114) into (100) we obtain

$$\Delta_{i\leftarrow j}(t+1, x_j) \approx \mathrm{const}$$
$$+ \; f_{\mathrm{in}}(x_j, q_j) - \frac{1}{2\tau_{i\leftarrow j}^r(t)}(\widehat{r}_{i\leftarrow j}(t) - x_j)^2, \tag{115}$$

where the constant term does not depend on x_j and

$$\frac{1}{\tau_{i\leftarrow j}^r(t)} \; = \; \sum_{\ell \neq i} a_{\ell j}^2 \tau_\ell^s(t) \tag{116a}$$

$$\widehat{r}_{i\leftarrow j}(t) \; = \; \tau_{i\leftarrow j}^r(t) \sum_{\ell \neq i} \left[s_\ell(t)a_{\ell j} + a_{\ell j}^2 \tau_\ell^s(t)\widehat{x}_j(t) \right]$$
$$= \; \widehat{x}_j(t) + \tau_{i\leftarrow j}^r(t) \sum_{\ell \neq i} s_\ell(t)a_{\ell j}. \tag{116b}$$

Now define $g_{\mathrm{in}}(r, q, \tau^r)$ as in (18). Then, using (115) and (18), $\widehat{x}_{i\leftarrow j}(t)$ in (104b) can be re-written as

$$\widehat{x}_{i\leftarrow j}(t+1) \approx g_{\mathrm{in}}\left(\widehat{r}_{i\leftarrow j}(t), q_j, \tau_{i\leftarrow j}^r(t)\right). \tag{117}$$

Then, if we define $\widehat{r}_j(t)$ and $\tau_j^r(t)$ as in (7), $\widehat{r}_{i\leftarrow j}(t)$ and $\tau_{i\leftarrow j}^r(t)$ in (116) can be re-written as

$$\tau_{i\leftarrow j}^r(t) \; \approx \; \tau_j^r(t). \tag{118a}$$

$$\widehat{r}_{i\leftarrow j}(t) \; \approx \; \widehat{x}_j(t) + \tau_j^r(t) \sum_{\ell \neq i} s_\ell(t)a_{\ell j}$$
$$= \; \widehat{r}_j(t) - \tau_j^r(t)a_{ij}s_i(t), \tag{118b}$$

where in the approximations we have ignored terms of order $O(a_{ij}^2)$. We can then simplify (117) as

$$
\begin{aligned}
\widehat{x}_{i\leftarrow j}&(t+1) \\
&\stackrel{(a)}{\approx} \quad g_{\text{in}}\left(\widehat{r}_j(t) - a_{ij}s_i(t)\tau_j^r(t), q_j, \tau_j^r(t)\right) \\
&\stackrel{(b)}{\approx} \quad \widehat{x}_j(t+1) - a_{ij}s_j(t)D_j(t+1)
\end{aligned}
\tag{119}
$$

where (a) follows from substituting (118) into (117) and (b) follows from a first-order approximation with the definitions

$$
\begin{aligned}
\widehat{x}_j(t+1) &:= g_{\text{in}}\left(\widehat{r}_j(t), q_j, \tau_j^r(t)\right) & \text{(120a)} \\
D_j(t+1) &:= \tau_j^r(t)\frac{\partial}{\partial \widehat{r}}g_{\text{in}}\left(\widehat{r}_j(t), q_j, \tau_j^r(t)\right). & \text{(120b)}
\end{aligned}
$$

Now

$$
\begin{aligned}
D_j(t+1) &\stackrel{(a)}{\approx} \tau_j^r(t)\frac{\partial}{\partial \widehat{r}}(\Gamma f_{\text{in}}(\cdot, q_j))\left(\widehat{r}_j(t), \tau_j^r(t)\right) \\
&\stackrel{(b)}{=} \frac{\tau_j^r(t)}{1 - \tau_j^r(t)f_{\text{in}}''(\widehat{x}_j(t+1), q_j)} \\
&\stackrel{(c)}{\approx} \left[-\frac{\partial^2}{\partial x_j^2}\Delta_{i\leftarrow j}(t+1, \widehat{x}_j(t+1))\right]^{-1} \\
&\stackrel{(d)}{\approx} \tau_j^x(t+1),
\end{aligned}
\tag{121}
$$

where (a) follows from (120b) and by comparing (18) to (102b); (b) follows from (103c); (c) follows from (115) and (d) follows from (104d). Substituting (121) into (119) and (110), we obtain

$$
\widehat{p}_i(t) = \sum_j a_{ij}\widehat{x}_j(t) - \tau_i^p(t)\widehat{s}_i(t-1),
\tag{122}
$$

which agrees with the definition in (5c). In summary, we have shown that with the choice of $g_{\text{in}}(\cdot)$ and $g_{\text{out}}(\cdot)$ in (18) and (24), the GAMP algorithm can be seen as a quadratic approximation of max-sum loopy BP for MAP estimation.

APPENDIX D
SUM-PRODUCT GAMP

A. Preliminary Lemma

Our analysis will needs the following standard result.

Lemma 2: Consider a random variable U with a conditional probability density function of the form

$$
p_{U|V}(u|v) := \frac{1}{Z(v)}\exp\left(\phi_0(u) + uv\right),
$$

where $Z(v)$ is a normalization constant (called the partition function). Then,

$$
\begin{aligned}
\frac{\partial}{\partial v}\log Z(v) &= \mathbb{E}(U|V = v) & \text{(123a)} \\
\frac{\partial^2}{\partial v^2}\log Z(v) &= \frac{\partial}{\partial v}\mathbb{E}(U|V = v) & \text{(123b)} \\
&= \text{var}(U|V = v). & \text{(123c)}
\end{aligned}
$$

Proof: The relations are standard properties of exponential families [23]. ■

B. Sum-Product BP for MMSE Estimation

The sum-product loopy BP algorithm for MMSE estimation is similar to max-sum algorithm in Appendix C. For the sum-product algorithm, the BP messages can also be represented as functions $\Delta_{i\rightarrow j}(t, x_j)$ and $\Delta_{i\leftarrow j}(t, x_j)$. However, these messages are to be interpreted as estimates of the log likelihoods of the variables x_j, conditioned on the system input and output vectors, $\mathbf{q}$ and $\mathbf{y}$, and the transform matrix $\mathbf{A}$.

The updates for the messages are also slightly different between the max-sum and sum-product algorithms. For the sum-product algorithm, the output node update (99) is replaced by the update equation

$$
\Delta_{i\rightarrow j}(t, x_j) = \log \mathbb{E}(p_{Y|Z}(y_i, z_i)|x_j) + \text{const},
\tag{124}
$$

where the expectation is over the random variable $z_i = \mathbf{a}_i^T\mathbf{x}$ with the components x_r of $\mathbf{x}$ being independent with distributions

$$
p_{i\leftarrow r}(x_r) \propto \exp \Delta_{i\leftarrow r}(t, x_r).
\tag{125}
$$

The constant term in (124) can be any term that does not depend on x_j. The sum-product input node update is identical to the max-sum update (100). Similar to the max-sum estimation algorithm, sum-product BP is terminated after some iterations. The final estimate for the conditional distribution of x_j is given by

$$
p_j(x_j) \propto \exp \Delta_j(t, x_j),
$$

where $\Delta_j(t, x_j)$ is given in (101). From this conditional distribution, one can compute the conditional mean of x_j.

C. GAMP Approximation

We now show that with the $g_{\text{in}}(\cdot)$ in (31) and $g_{\text{out}}(\cdot)$ in (36), the GAMP algorithm can heuristically be regarded as a Gaussian approximation of the above updates. The calculations are largely the same as the approximations for max-sum BP in Appendix C, so we will just point out the key differences.

We begin with the output node update (124). Let

$$
\begin{aligned}
\widehat{x}_j(t) &:= \mathbb{E}[x_j|\Delta_j(t, \cdot)], & \text{(126a)} \\
\widehat{x}_{i\leftarrow j}(t) &:= \mathbb{E}[x_j|\Delta_{i\leftarrow j}(t, \cdot)], & \text{(126b)} \\
\tau_j^x(t) &:= \text{var}[x_j|\Delta_j(t, \cdot)] & \text{(126c)} \\
\tau_{i\leftarrow j}^x(t) &:= \text{var}[x_j|\Delta_{i\leftarrow j}(t, \cdot)], & \text{(126d)}
\end{aligned}
$$

where we have used the notation $\mathbb{E}(g(x)|\Delta(\cdot))$ to mean the expectation over a random variable x with a probability density function

$$
p_\Delta(x) \propto \exp\left(\Delta(x)\right).
\tag{127}
$$

Therefore, $\widehat{x}_{i\leftarrow r}(t)$ and $\tau_{i\leftarrow r}^x(t)$ are the mean and variance of the random variable x_r with density (125). Now, the expectation in (124) is over $z_i = \mathbf{a}_i^T\mathbf{x}$ with the components x_r being independent with probability density (125). So, for large n, the Central Limit Theorem suggests that the conditional distribution of z_i given x_j should be approximately Gaussian $z_i \sim \mathcal{N}(\widehat{p}_{i\leftarrow j}(t), \tau_{i\leftarrow j}^p(t))$, where $\widehat{p}_{i\leftarrow j}(t)$ and $\tau_{i\leftarrow j}^p(t))$ are defined in (108). Hence, $\Delta_{i\rightarrow j}(t, x_j)$ can be approximated as

$$
\Delta_{i\rightarrow j}(t, x_j) \approx H\left(\widehat{p}_{i\leftarrow j}(t), y_i, \tau_{i\leftarrow j}^p(t)\right),
\tag{128}
$$

where

$$H(\widehat{p}, y, \tau^p) := \log \mathbb{E}\left[p_{Y|Z}(y|z) | \widehat{p}, \tau^p \right], \qquad (129)$$

and the expectation is over random variable $z \sim \mathcal{N}(\widehat{p}, \tau^p)$. It is easily checked that

$$H(\widehat{p}, y, \tau^p) := \log p(z|\widehat{p}, y, \tau^p) + \text{const}, \qquad (130)$$

where $p(z|\widehat{p}, y, \tau^p)$ is given in (35) and the constant term does not depend on $\widehat{p}$.

Now, identical to the argument in Appendix IV-A, we can define $\widehat{p}_i(t)$ as in (110) and $\tau_i^p(t)$ as in (5a) so that $\Delta_{i \to j}(t, x_j)$ can be approximated as (112). Also, if we define $g_{\text{out}}(\widehat{p}, y, \tau^p)$ as in (113), and $\widehat{s}_i(t)$ and $\tau_i^s(t)$ as in (6), then $\widehat{s}_i(t)$ and $\tau_i^s(t)$ are respectively the first and second-order derivatives of $H(\widehat{p}_i(t), y_i, \tau_i^p(t))$ with respect to $\widehat{p}$. Then, taking a second order approximation of (112) results in (114).

Also, using (130), $g_{\text{out}}(\cdot)$ as defined in (113) agrees with the definition in (38). To show that $g_{\text{out}}(\cdot)$ is also equivalent to (36) note that we can rewrite $H(\cdot)$ in (130) as

$$H(\widehat{p}, y, \tau^p) := \log \left[\frac{Z(\widehat{p}, y, \tau^p)}{Z_0(\widehat{p}, y, \tau^p)} \right],$$

where

$$Z_0(\widehat{p}, y, \tau^p) := \int \exp \left[\frac{1}{2\tau^p}(2\widehat{p}z - z^2) \right] dz$$

$$Z(\widehat{p}, y, \tau^p)$$
$$:= \int \exp \left[f_{\text{out}}(z, y) + \frac{1}{2\tau^p}(2\widehat{p}z - z^2) \right] dz.$$

Then the relations (36) and (37) follow from the relations (123).

We next consider the input node update (100). Similar to Appendix C, we can substitute the approximation (114) into (100) to obtain (115) where $\widehat{r}_{i \leftarrow j}(t)$ and $\tau_{i \leftarrow j}^r(t)$ are defined in (116).

Using the approximation (115) and the definition of $F_{\text{in}}(\cdot)$ in (19), the probability distribution $p_\Delta(x_j)$ in (127) with $\Delta = \Delta_{i \leftarrow j}(t, x_j)$ is approximately

$$p_{\Delta_{i \leftarrow j}(t, \cdot)}(x_j)$$
$$\approx \frac{1}{Z} \exp F_{\text{in}}(x_j, \widehat{r}_{i \leftarrow j}(t), q_j, \tau_{i \leftarrow j}^r(t)),$$

where Z is a normalization constant. But, based on the form of $F_{\text{in}}(\cdot)$ in (19), this distribution is identical to the conditional distribution of X in $\theta^r(\tau^r)$ in (48) given $(\widehat{R}, Q) = (\widehat{r}, q)$. Therefore, if we define $g_{\text{in}}(\cdot)$ as in (31), it follows that $\widehat{x}_{i \leftarrow j}(t)$ in (126b) satisfies (117). The remainder of the proof now follows as in the proof of Appendix D.

APPENDIX E
PROOF OF (78)

Define

$$D(\mathbf{K}^p) := \mathbb{E}\left[\frac{\partial}{\partial \widehat{p}} g_{\text{out}}(t, h(Z, W), \widehat{P}, \overline{\tau}^p(t)) \right]$$

$$- \mathbb{E}\left[\frac{\partial}{\partial z} g_{\text{out}}(t, h(Z, W), \widehat{P}, \overline{\tau}^p(t)) \right], \qquad (131)$$

where the expectation is over $(Z, \widehat{P}) \sim \mathcal{N}(0, \mathbf{K}^p)$ and $W \sim p_W(w)$ independent of $(Z, \widehat{P})$. We must show that when $\mathbf{K}^p$ is of the form (73) and $g_{\text{out}}(\cdot)$ is given by (36), then $D(\mathbf{K}^p) = 0$.

To this end, let $\mathbf{Q}$ be the two-dimensional random vector

$$\mathbf{Q} = [Z \quad \widehat{P}]^T \sim \mathcal{N}(0, \mathbf{K}^p) \qquad (132)$$

and let $\mathbf{G} \in \mathbb{R}^{1 \times 2}$ be the derivative

$$\mathbf{G} = \mathbb{E}\left[\frac{\partial}{\partial \mathbf{q}} g_{\text{out}}(t, h(Z, W), \widehat{P}, \overline{\tau}^p(t)) \right],$$

which is the row vector whose two components are the partial derivatives with respect to z and $\widehat{p}$. Thus, $D(\mathbf{K}^p)$ in (131) can be re-written as

$$D(\mathbf{K}^p) = \mathbf{G} \begin{bmatrix} 1 \\ -1 \end{bmatrix}. \qquad (133)$$

Also, using Stein's Lemma (Lemma 4 below) and the covariance (132),

$$\mathbb{E}\left[\mathbf{Q} g_{\text{out}}(t, h(Z, W), \widehat{P}, \overline{\tau}^p(t)) \right]$$
$$= \mathbb{E}\left[\mathbf{Q} \mathbf{Q}^T \right] \mathbf{G}^T = \mathbf{K}^p \mathbf{G}'^T.$$

Applying this equality to (133) we obtain

$$D(\mathbf{K}^p) := \mathbb{E}\left[\phi(Z, \widehat{P}) g_{\text{out}}(t, h(Z, W), \widehat{P}, \overline{\tau}^p(t)) \right], \quad (134)$$

where $\phi(\cdot)$ is the function:

$$\phi(z, \widehat{p}) := [z \ \widehat{p}] (\mathbf{K}^p)^{-1} \begin{bmatrix} 1 \\ -1 \end{bmatrix}.$$

When $\mathbf{K}^p$ is of the form (73), then it is easily checked that

$$\phi(z, \widehat{p}) = \frac{\widehat{p}}{\overline{\tau}^p(t)}. \qquad (135)$$

Therefore,

$$D(\mathbf{K}^p) \overset{(a)}{=} \frac{1}{\overline{\tau}^p(t)} \mathbb{E}\left[\widehat{P} g_{\text{out}}(t, h(Z, W), \widehat{P}, \overline{\tau}^p(t)) \right]$$

$$\overset{(b)}{=} \frac{1}{\overline{\tau}^p(t)} \mathbb{E}\left[\widehat{P} \frac{\partial}{\partial \widehat{P}} \log p_{Y|\widehat{P}}(Y|\widehat{P}) \right]$$

$$\overset{(c)}{=} \frac{1}{\overline{\tau}^p(t)} \mathbb{E}\left[\widehat{P} \frac{\partial}{\partial \widehat{p}} \int p_{Y|\widehat{P}}(y|\widehat{P}) dy \right]$$

$$= \frac{1}{\overline{\tau}^p(t)} \mathbb{E}\left[\widehat{P} \frac{\partial}{\partial \widehat{p}}(1) \right] = 0$$

where (a) follows from substituting (135) into (134); (b) follows from (74); (c) follows from taking the derivative of the logarithm. This shows that $D(\mathbf{K}^p) = 0$ and proves (78).

APPENDIX F
PROOF SKETCH FOR CLAIM 2

As stated earlier, Bayati and Montanari in [7] already proved the result for scalar case when $n_b = n_d = 1$. Only very minor modifications are required for the vector-valued case, so we will just provide a sketch of the key changes.

For the vector case, it is convenient to introduce the notation $\mathbf{b}(t)$ to denote the matrix with columns $\mathbf{b}_i(t)$:

$$\mathbf{b}(t) := \begin{bmatrix} \mathbf{b}_1(t)^T \\ \vdots \\ \mathbf{b}_m(t)^T \end{bmatrix} \in \mathbb{R}^{m \times n_b}.$$

We can define $\mathbf{u}(t)$, $\mathbf{v}(t)$ and $\mathbf{d}(t)$ similarly. Then, in analogy with the definitions in [7], we let

$$
\begin{aligned}
\mathbf{x}(t) &:= \mathbf{d}(t) + \mathbf{u}(t)\boldsymbol{\xi}(t)^T \in \mathbb{R}^{n \times n_d} \\
\mathbf{y}(t) &:= \mathbf{b}(t) + \mathbf{v}(t-1)\boldsymbol{\lambda}(t)^T \in \mathbb{R}^{m \times n_b}
\end{aligned}
$$

and define the matrices

$$
\begin{aligned}
\mathbf{X}(t) &:= [\mathbf{x}(0)| \cdots |\mathbf{x}(t-1)] \in \mathbb{R}^{n \times tn_d} \\
\mathbf{Y}(t) &:= [\mathbf{y}(0)| \cdots |\mathbf{y}(t-1)] \in \mathbb{R}^{m \times tn_b} \\
\mathbf{U}(t) &:= [\mathbf{u}(0)| \cdots |\mathbf{u}(t-1)] \in \mathbb{R}^{n \times tn_b} \\
\mathbf{V}(t) &:= [\mathbf{v}(0)| \cdots |\mathbf{v}(t-1)] \in \mathbb{R}^{m \times tn_d}.
\end{aligned}
$$

The updates (83) can then be re-written in matrix form as

$$
\mathbf{X}(t) = \mathbf{A}^T \mathbf{V}(t), \quad \mathbf{Y}(t) = \mathbf{A}\mathbf{U}(t). \tag{136}
$$

Next, also following the proof in [7], let $\mathbf{v}_{||}(t)$ be the projection of each column of $\mathbf{v}(t)$ onto the column space of $\mathbf{V}(t)$, and let $\mathbf{v}_\perp(t)$ be its orthogonal component, $\mathbf{v}_\perp(t) = \mathbf{v}(t) - \mathbf{v}_{||}(t)$. Thus, we can write

$$
\mathbf{v}_{||}(t) = \sum_{i=0}^{t-1} \mathbf{v}(i)\boldsymbol{\alpha}_i(t), \tag{137}
$$

for matrices $\boldsymbol{\alpha}_i(t) \in \mathbb{R}^{n_d \times n_d}$. Similarly, let $\mathbf{u}_{||}(t)$ and $\mathbf{u}_\perp(t)$ be the parallel and orthogonal components of $\mathbf{u}(t)$ with respect to the column space of $\mathbf{U}(t)$ so that

$$
\mathbf{u}_{||}(t) = \sum_{i=0}^{t-1} \mathbf{u}(i)\boldsymbol{\beta}_i(t), \tag{138}
$$

where $\boldsymbol{\beta}_i(t) \in \mathbb{R}^{n_b \times n_b}$.

Now let $\mathfrak{G}(t_1, t_2)$ be the sigma algebra generated by $\mathbf{b}(0), \ldots, \mathbf{b}(t_1)$, $\mathbf{v}(0), \ldots, \mathbf{v}(t_1)$, $\mathbf{d}(0), \ldots, \mathbf{d}(t_2 - 1)$ and $\mathbf{u}(0), \ldots, \mathbf{u}(t_2)$. Also for any sigma-algebra, $\mathfrak{G}$, and random variables X and Y, we will say that X is equal in distribution to Y conditional on $\mathfrak{G}$ if $\mathbb{E}(\phi(X)Z) = \mathbb{E}(\phi(Y)Z)$ for any Z that is $\mathfrak{G}$-measurable. In this case, we write $X|_{\mathfrak{G}} \overset{d}{=} Y$.

With these definitions, the vector-valued analogy of the main technical lemma [7, Lemma 1] can be stated as follows:

Lemma 3: Under the assumptions of Claim 2, the following hold for all $t \geq 0$:

(a) The conditional distribution of $\mathbf{d}(t+1)$ is given by

$$
\mathbf{d}(t)|_{\mathfrak{G}(t,t)}
$$
$$
\overset{d}{=} \sum_{i=0}^{t-1} \mathbf{d}(i)\boldsymbol{\alpha}_i(t) + \widetilde{\mathbf{A}}^T \mathbf{v}_\perp(t) + \widetilde{\mathbf{U}}(t)\overrightarrow{\sigma}_t(1)
$$
$$
\mathbf{b}(t)|_{\mathfrak{G}(t-1,t)}
$$
$$
\overset{d}{=} \sum_{i=0}^{t-1} \mathbf{b}(i)\boldsymbol{\beta}_i(t) + \widetilde{\mathbf{A}}\mathbf{u}_\perp(t) + \widetilde{\mathbf{V}}(t)\overrightarrow{\sigma}_t(1)
$$

where $\widetilde{\mathbf{A}}$ is an independent copy of $\mathbf{A}$, and the matrices $\boldsymbol{\alpha}_i(t)$ and $\boldsymbol{\beta}_i(t)$ are the coefficients in the expansion (137) and (138). The matrix $\widetilde{\mathbf{U}}(t)$ is such that its columns form an orthogonal basis for the column space of $\mathbf{U}(t)$ with $\widetilde{\mathbf{U}}(t)^T \widetilde{\mathbf{U}}(t) = nI_{tn_d}$. Similarly, $\widetilde{\mathbf{V}}(t)$ is such that its columns form an orthogonal basis for the column space

of $\mathbf{V}(t)$ with $\widetilde{\mathbf{V}}(t)^T \widetilde{\mathbf{V}}(t) = mI_{tn_b}$. The vector $\overrightarrow{\sigma}_t(1)$ goes to zero almost surely.

(b) For any pseudo-Lipschitz functions $\phi_d : \mathbb{R}^{n_d(t+1)} \times \Theta^u \to \mathbb{R}$ and $\phi_b : \mathbb{R}^{n_b(t+1)} \times \Theta^v \to \mathbb{R}$:

$$
\lim_{n \to \infty} \frac{1}{n} \sum_{j=1}^{n} \phi_d(\mathbf{d}_j(0), \cdots, \mathbf{d}_j(t), \theta_j^u)
$$
$$
= \mathbb{E}[\phi_d(\mathbf{D}(0), \ldots, \mathbf{D}(t), \theta^u)]
$$
$$
\lim_{n \to \infty} \frac{1}{m} \sum_{i=1}^{m} \phi_b(\mathbf{b}_j(0), \cdots, \mathbf{b}_j(t), \theta_j^v)
$$
$$
= \mathbb{E}[\phi_b(\mathbf{B}(0), \ldots, \mathbf{B}(t), \theta^v)]
$$

where the expectations are over Gaussian random vectors $(\mathbf{D}(0), \ldots, \mathbf{D}(t))$ and $(\mathbf{B}(0), \ldots, \mathbf{B}(t))$ and the random variables θ^u and θ^v in the limit (86). The variables θ^u and θ^v are independent of $\mathbf{B}(r)$ and $\mathbf{D}(r)$ and the marginal distributions of $\mathbf{B}(r)$ and $\mathbf{D}(r)$ are given by (88).

(c) For all $0 \leq r, s \leq t$, the following limit exists hold are bounded and are non-random

$$
\lim_{n \to \infty} \frac{1}{n} \sum_{j=1}^{n} \mathbf{d}_j(r)\mathbf{d}_j(s)^T
$$
$$
= \lim_{m \to \infty} \frac{1}{m} \sum_{i=1}^{m} \mathbf{v}_i(r)\mathbf{v}_i(s)^T
$$
$$
\lim_{n \to \infty} \frac{1}{m} \sum_{i=1}^{m} \mathbf{b}_i(r)\mathbf{b}_i(s)^T
$$
$$
= \beta \lim_{m \to \infty} \frac{1}{n} \sum_{i=1}^{n} \mathbf{u}_j(r)\mathbf{u}_j(s)^T.
$$

(d) Suppose $\varphi_d : \mathbb{R}^{n_d} \times \Theta^u \to \mathbb{R}$ and $\varphi_b : \mathbb{R}^{n_b} \times \Theta^v \to \mathbb{R}$ are almost everywhere continuously differentiable with a bounded derivative with respect to the first argument. Then, for all all $0 \leq r, s \leq t$ the following limits exists are bounded and non-random:

$$
\lim_{n \to \infty} \frac{1}{n} \sum_{j=1}^{n} \mathbf{d}_j(r)\varphi_d(\mathbf{d}_j(s))^T
$$
$$
= \lim_{n \to \infty} \frac{1}{n} \sum_{j=1}^{n} \mathbf{d}_j(r)\mathbf{d}_j(s)^T \mathbf{G}_d(s)^T
$$
$$
\lim_{m \to \infty} \frac{1}{m} \sum_{i=1}^{m} \mathbf{b}_i(r)\varphi_b(\mathbf{b}_j(s))^T
$$
$$
= \lim_{m \to \infty} \frac{1}{m} \sum_{i=1}^{m} \mathbf{b}_i(r)\mathbf{b}_i(s)^T \mathbf{G}_b(s)^T
$$

where $\mathbf{G}_d(s)$ and $\mathbf{G}_b(s)$ are the empirical derivatives

$$
\mathbf{G}_d(s) := \lim_{n \to \infty} \frac{1}{n} \sum_{j=1}^{n} \frac{\partial}{\partial \mathbf{d}} \varphi_d(\mathbf{d}_j(s), \theta^u)
$$
$$
\mathbf{G}_b(s) := \lim_{n \to \infty} \frac{1}{m} \sum_{i=1}^{m} \frac{\partial}{\partial \mathbf{b}} \varphi_b(\mathbf{b}_i(s), \theta^v).
$$

(e) For $\ell = k - 1$, the following bounds hold almost surely

$$\lim_{n \to \infty} \frac{1}{n} \sum_{j=1}^{n} \|\mathbf{d}_j(t)\|^{2\ell} < \infty$$

$$\lim_{m \to \infty} \frac{1}{m} \sum_{i=1}^{m} \|\mathbf{b}_i(t)\|^{2\ell} < \infty$$

This lemma is a verbatim copy of [7, Lemma 1] with some minor changes for the vector-valued case. Observe that Claim 2 is a special case of part (b) of Lemma 3 by considering functions of the form

$$\phi_d(\mathbf{d}(0), \ldots, \mathbf{d}(t), \theta^u) = \phi_d(\mathbf{d}(t), \theta^u)$$
$$\phi_b(\mathbf{b}(0), \ldots, \mathbf{b}(t), \theta^u) = \phi_b(\mathbf{b}(t), \theta^v).$$

That is, we consider functions that only depend on the most recent iteration number.

The proof of Lemma 3 also follows follows almost identically to the proof of the analogous lemma in the scalar case in [7]. The key idea in that proof is the following conditioning argument, originally used by Bolthausen in [41]: To evaluate the conditional distributions of $\mathbf{d}(t)$ with respect to $\mathfrak{G}(t, t)$ and $\mathbf{b}(t)$ with respect to $\mathfrak{G}(t-1, t)$ in part (a) of Lemma 3, one evaluates the corresponding conditional distribution of the matrix $\mathbf{A}$. But, this conditional distribution is precisely identical to the distribution of $\mathbf{A}$ conditioned on *linear* constraints of the form (136). But, this distribution is just the distribution of a Gaussian random vector conditioned on it lying on an affine subspace. That distribution has a simple expression as a deterministic offset plus a projected Gaussian. The detailed expression are given in [7, Lemma 6] and the identical equations can be used here.

The proof uses this conditioning principle along with an induction argument along the iteration number t and the statements (a) to (e) of the lemma. We omit the details as they are involved but follow with only minor changes from the original scalar proof in [7].

The only one other non-trivial extension that is needed is the matrix form of Stein's Lemma, which can be stated as:

Lemma 4 (Stein's Lemma [71]): Suppose $\mathbf{Z}_1 \in \mathbb{R}^{n_1}$ and $\mathbf{Z}_2 \in \mathbb{R}^{n_2}$ are jointly Gaussian random vectors and $\varphi : \mathbb{R}^{n_2} \to \mathbb{R}^{n_3}$ is any function such that the expectations

$$\mathbf{G} := \mathbb{E}\left[\frac{\partial}{\partial \mathbf{z}_2} \varphi(\mathbf{Z}_2)\right] \in \mathbb{R}^{n_3 \times n_2}$$

and

$$\mathbb{E}(\mathbf{Z}_1 \varphi(\mathbf{Z}_2)^T) \in \mathbb{R}^{n_1 \times n_3}$$

exists. Then,

$$\mathbb{E}(\mathbf{Z}_1 \varphi(\mathbf{Z}_2)^T) = \mathbb{E}\left((\mathbf{Z}_1 - \overline{\mathbf{Z}}_1)(\mathbf{Z}_2 - \overline{\mathbf{Z}}_2)^T\right) \mathbf{G}^T,$$

where $\overline{\mathbf{Z}}_i$ is the expectation of $\mathbf{Z}_i$.

Note that part (d) of Lemma 3 is of the same form of this lemma.

REFERENCES

[1] 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.

[2] J. P. Neirotti and D. Saad, "Improved message passing for inference in densely connected systems," *Europhys. Lett.*, vol. 71, no. 5, pp. 866–872, Sep. 2005.

[3] 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.

[4] 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.

[5] ——, "Random sparse linear systems observed via arbitrary channels: A decoupling principle," in *Proc. IEEE Int. Symp. Inform. Theory*, Nice, France, Jun. 2007, pp. 946–950.

[6] 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.

[7] 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.

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

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

[10] 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.

[11] M. Fortin and R. Glowinski, *Augmented Lagrangian Methods*. Amsterdam: North-Holland Publishing Co., 1983, vol. 15.

[12] R. Glowinski and P. L. Tallec, *Augmented Lagrangian and Operator-Splitting Methods in Nonlinear Mechanics*, ser. SIAM Studies in Applied Mathematics. Philadelphia, PA: SIAM, 1989.

[13] B. He, L.-Z. Liao, D. Han, and H. Yang, "A new inexact alternating directions method for monotone variational inequalities," *Math. Program.*, vol. 92, no. 1, Ser A, pp. 103–108, 2002.

[14] Z. Wen, D. Goldfarb, and W. Yin, "Alternating direction augmented Lagrangian methods for semidefinite programming," *Math. Program. Comp.*, vol. 2, no. 3–4, pp. 203–230, 2010.

[15] S. ten Brink, "Convergence behavior of iteratively decoded parallel concatenated codes," *IEEE Trans. Commun.*, vol. 49, no. 10, pp. 1727–1737, Oct. 2001.

[16] A. Ashikhmin, G. Kramer, and S. ten Brink, "Extrinsic information transfer functions: Model and erasure channel properties," *IEEE Trans. Inform. Theory*, vol. 50, no. 11, pp. 2657–2673, Nov. 2004.

[17] T. Tanaka, "A statistical-mechanics approach to large-system analysis of CDMA multiuser detectors," *IEEE Trans. Inform. Theory*, vol. 48, no. 11, pp. 2888–2910, Nov. 2002.

[18] D. Guo and S. Verdú, "Randomly spread CDMA: Asymptotics via statistical physics," *IEEE Trans. Inform. Theory*, vol. 51, no. 6, pp. 1983–2010, Jun. 2005.

[19] Y. Kabashima, T. Wadayama, and T. Tanaka, "Typical reconstruction limit of compressed sensing based on l_p-norm minimization," arXiv:0907.0914 [cs.IT]., Jun. 2009.

[20] F. Krzakala, M. Mézard, F. Sausset, Y. Sun, and L. Zdeborová, "Statistical physics-based reconstruction in compressed sensing," arXiv:1109.4424, Sep. 2011.

[21] ——, "Probabilistic reconstruction in compressed sensing: Algorithms, phase diagrams, and threshold achieving matrices," arXiv:1206.3953, Jun. 2012.

[22] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Mateo, CA: Morgan Kaufmann Publ., 1988.

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

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

[25] D. J. C. MacKay, "Good error-correcting codes based on very sparse matrices," *IEEE Trans. Inform. Theory*, vol. 45, no. 3, pp. 399–431, Mar. 1999.

[26] D. J. C. MacKay and R. M. Neal, "Near Shannon limit performance of low density parity check codes," *Electron. Letters*, vol. 33, pp. 457–458, 1997.

[27] M. Yoshida and T. Tanaka, "Analysis of sparsely-spread CDMA via statistical mechanics," in *Proc. IEEE Int. Symp. Inform. Theory*, Seattle, WA, Jun. 2006, pp. 2378–2382.

[28] G. Guo and C. C. Wang, "Multiuser detection of sparsely spread CDMA," *IEEE J. Sel. Areas Comm.*, vol. 26, no. 3, pp. 421–431, Mar. 2008.

[29] N. Sommer, M. Feder, and O. Shalvi, "Low-density lattice codes," *IEEE Trans. Inform. Theory*, vol. 54, no. 4, pp. 1561–1585, Apr. 2008.

[30] D. Baron, S. Sarvotham, and R. G. Baraniuk, "Bayesian compressive sensing via belief propagation," *IEEE Trans. Signal Process.*, vol. 58, no. 1, pp. 269–280, Jan. 2010.

[31] D. Guo, D. Baron, and S. Shamai, "A single-letter characterization of optimal noisy compressed sensing," in *Proc. 47th Ann. Allerton Conf. on Commun., Control and Comp.*, Monticello, IL, Sep.–Oct. 2009.

[32] T. P. Minka, "A family of algorithms for approximate Bayesian inference," Ph.D. dissertation, Massachusetts Institute of Technology, Cambridge, MA, 2001.

[33] M. Seeger, "Bayesian inference and optimal design for the sparse linear model," *J. Machine Learning Research*, vol. 9, pp. 759–813, Sep. 2008.

[34] S.-Y. Chung, T. J. Richardson, and R. L. Urbanke, "Analysis of sum-product decoding of low-density parity-check codes using a Gaussian approximation," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 657–670, Feb. 2001.

[35] H. El Gamal and R. Hammons, "Analyzing the turbo decoder using the Gaussian approximation," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 671–686, Feb. 2001.

[36] L. R. Varshney, "Performance of LDPC codes under noisy message-passing decoding," in *Proc. Inform. Th. Workshop*, Lake Tahoe, CA, Sep. 2007, pp. 178–183.

[37] D. L. Donoho, A. Maleki, and A. Montanari, "Message passing algorithms for compressed sensing I: motivation and construction," in *Proc. Info. Theory Workshop*, Jan. 2010.

[38] ——, "Message passing algorithms for compressed sensing II: analysis and validation," in *Proc. Info. Theory Workshop*, Jan. 2010.

[39] R. Tibshirani, "Regression shrinkage and selection via the lasso," *J. Royal Stat. Soc., Ser. B*, vol. 58, no. 1, pp. 267–288, 1996.

[40] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," Feb. 1996.

[41] E. Bolthausen, "On the high-temperature phase of the Sherrington–Kirkpatrick model," Seminar at Eurandom, Eindhoven, Sep. 2009.

[42] S. Rangan, A. Fletcher, and V. K. Goyal, "Asymptotic analysis of MAP estimation via the replica method and applications to compressed sensing," *IEEE Trans. Inform. Theory*, vol. 58, no. 3, pp. 1902–1923, Mar. 2012.

[43] S. Rangan, "Generalized approximate message passing for estimation with random linear mixing," in *Proc. IEEE Int. Symp. Inform. Theory*, Saint Petersburg, Russia, Jul.–Aug. 2011, pp. 2174–2178.

[44] C. M. Bishop, *Pattern Recognition and Machine Learning*, ser. Information Science and Statistics.   New York, NY: Springer, 2006.

[45] E. J. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Trans. Inform. Theory*, vol. 52, no. 2, pp. 489–509, Feb. 2006.

[46] D. L. Donoho, "Compressed sensing," *IEEE Trans. Inform. Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.

[47] E. J. Candès and T. Tao, "Near-optimal signal recovery from random projections: Universal encoding strategies?" *IEEE Trans. Inform. Theory*, vol. 52, no. 12, pp. 5406–5425, Dec. 2006.

[48] D. Tse and S. Hanly, "Linear multiuser receivers: Effective interference, effective bandwidth and capacity," *IEEE Trans. Inform. Theory*, vol. 45, no. 3, pp. 641–675, Mar. 1999.

[49] A. K. Fletcher, S. Rangan, and V. K. Goyal, "On–off random access channels: A compressed sensing framework," arXiv:0903.1022v1 [cs.IT]., Mar. 2009.

[50] U. S. Kamilov, S. Rangan, A. K. Fletcher, and M. Unser, "Approximate message passing with consistent parameter estimation and applications to sparse learning," arXiv:1207.3859 [cs.IT], Jul. 2012.

[51] S. Verdú and S. Shamai, "Spectral efficiency of CDMA with random spreading," *IEEE Trans. Inform. Theory*, vol. 45, no. 3, pp. 622–640, Mar. 1999.

[52] A. Montanari and D. Tse, "Analysis of belief propagation for non-linear problems: The example of CDMA (or: How to prove Tanaka's formula)," arXiv:cs/0602028v1 [cs.IT], Feb. 2006.

[53] S. Rangan, A. K. Fletcher, and V. K. Goyal, "Asymptotic analysis of MAP estimation via the replica method and applications to compressed sensing," arXiv:0906.3234v1 [cs.IT]., Jun. 2009.

[54] S. Rangan, "Generalized approximate message passing wiki page," http://gampmatlab.sourceforge.net/wiki/index.php, 2011.

[55] S. Rangan and R. Madan, "Belief propagation methods for intercell interference coordination in femtocell networks," *IEEE J. Sel. Areas Comm.*, vol. 30, no. 3, pp. 631–640, Apr. 2012.

[56] U. Kamilov, V. K. Goyal, and S. Rangan, "Message-passing estimation from quantized samples," arXiv:1105.6368v1 [cs.IT]., May 2011.

[57] A. K. Fletcher, S. Rangan, L. Varshney, and A. Bhargava, "Neural reconstruction with approximate message passing (NeuRAMP)," in *Proc. Neural Information Process. Syst.*, Granada, Spain, Dec. 2011.

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

[59] G. Caire, S. Shamai, A. Tulino, and S. Verdú, "Support recovery in compressed sensing: Information-theoretic bounds," in *Proc. UCSD Workshop Inform. Theory & Its Applications*, La Jolla, CA, Jan. 2011.

[60] D. Donoho, I. Johnstone, A. Maleki, and A. Montanari, "Compressed sensing over ℓ^p-balls: Minimax mean square error," in *Proc. ISIT*, St. Petersburg, Russia, Jun. 2011.

[61] J. P. Vila and P. Schniter, "Expectation-maximization Bernoulli-Gaussian approximate message passing," in *Conf. Rec. 45th Asilomar Conf. Signals, Syst. & Comput.*, Pacific Grove, CA, Nov. 2011, pp. 799–803.

[62] ——, "Expectation-maximization Gaussian-mixture approximate message passing," in *Proc. Conf. on Inform. Sci. & Sys.*, Princeton, NJ, Mar. 2012.

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

[64] J. Ziniel, L. C. Potter, and P. Schniter, "Tracking and smoothing of time-varying sparse signals via approximate belief propagation," in *Conf. Rec. 44th Asilomar Conf. Signals, Syst. & Comput.*, Pacific Grove, CA, Nov. 2010, pp. 802–812.

[65] S. Som, L. C. Potter, and P. Schniter, "Compressive imaging using approximate message passing and a Markov-tree prior," in *Conf. Rec. 44th Asilomar Conf. Signals, Syst. & Comput.*, Pacific Grove, CA, Nov. 2010, pp. 243–247.

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

[67] S. Rangan, A. K. Fletcher, V. K. Goyal, and P. Schniter, "Hybrid generalized approximation message passing with applications to structured sparsity," in *Proc. IEEE Int. Symp. Inform. Theory*, Cambridge, MA, Jul. 2012, pp. 1241–1245.

[68] T. Richardson and R. Urbanke, "The capacity of low-density parity check codes under message-passing decoding," Bell Laboratories, Lucent Technologies, Tech. Rep. BL01121710-981105-34TM, Nov. 1998.

[69] R. T. Rockafellar, *Convex Analysis*.   Princeton, NJ: Princeton Univ. Press, 1970.

[70] A. Montanari, "Graphical model concepts in compressed sensing," in *Compressed Sensing: Theory and Applications*, Y. C. Eldar and G. Kutyniok, Eds.   Cambridge Univ. Press, Jun. 2012, pp. 394–438.

[71] C. Stein, "A bound for the error in the normal approximation to the distribution of a sum of dependent random variables," in *Proc. Sixth Berkeley Symposium on Mathematical Statistics and Probability*, Berkeley, CA, 1972.