Estimation with Random Linear Mixing, Belief Propagation and Compressed Sensing

Sundeep Rangan

***Abstract*—We apply Guo and Wang's relaxed belief propagation (BP) method to the estimation of a random vector from linear measurements followed by a componentwise probabilistic measurement channel. Relaxed BP uses a Gaussian approximation in standard BP to obtain significant computational savings for dense measurement matrices. The main contribution of this paper is to extend the relaxed BP method and analysis to general (non-AWGN) output channels. Specifically, we present detailed equations for implementing relaxed BP for general channels and show that relaxed BP has an identical asymptotic large sparse limit behavior as standard BP, as predicted by the Guo and Wang's state evolution (SE) equations. Applications are presented to compressed sensing and estimation with bounded noise.**

***Index Terms*—Non-Gaussian estimation, belief propagation, density evolution, compressed sensing, sparsity, bounded noise.**

I. Introduction

Consider the problem of estimating a random vector $\mathbf{x} \in \mathbb{R}^n$ from the vector $\mathbf{y} \in \mathbb{R}^m$ shown in Fig. 1. As depicted in the figure, the input vector $\mathbf{x}$ is first passed through a linear transform,

$$\mathbf{z} = \Phi\mathbf{x}, \tag{1}$$

where $\Phi \in \mathbb{R}^{m \times n}$ is a known transform matrix, and then passed through an *output channel* or *measurement channel* described by a conditional distribution $p_{\mathbf{Y}|\mathbf{Z}}(\mathbf{y}|\mathbf{z})$. Suppose that the distributions of both the input vector $\mathbf{x}$ and output channel are *separable* in that the probability distributions factor as

$$p_{\mathbf{X}}(\mathbf{x}) = \prod_{j=1}^{n} p_X(x_j), \quad p_{\mathbf{Y}|\mathbf{Z}}(\mathbf{y}|\mathbf{z}) = \prod_{i=1}^{m} p_{Y|Z}(y_i|z_i), \tag{2}$$

where $p_X(x_j)$ and $p_{Y|Z}(y_i|z_i)$ are scalar distribution functions, and x_j, y_i and z_i are the components of the vectors $\mathbf{x}$, $\mathbf{y}$ and $\mathbf{z}$, respectively.

If $m = n$ and the mixing matrix Φ is the identity matrix, then the problem of estimating the input vector $\mathbf{x}$ from the output vector $\mathbf{y}$ reduces to n scalar estimation problems. However, for general Φ, optimal estimation of $\mathbf{x}$ is usually intractable because the transform matrix Φ "couples" or "mixes" the n components of $\mathbf{x}$ into the m components of the output vector $\mathbf{y}$. We thus call the problem of estimating the vector $\mathbf{x}$ from the coupled output vector $\mathbf{y}$ the *linear mixing estimation problem*.

One natural approach to the linear mixing estimation problem is *belief propagation* (BP), which iteratively updates estimates of the variables based on message passing along a graph [1], [2]. In communications and signal processing, BP is best known for its connections to iterative decoding in turbo and LDPC codes [3]–[5]. 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 [6], [7], lattice codes [8] and compressed sensing [9]–[11].

A key theoretical justification for applying BP to the specific problem of linear mixing estimation came with the work of Montanari and Tse [12]. That worked considered BP estimation of binary ±1 vectors with AWGN measurements and large sparse random mixing matrices. In this setting, Montanari and Tse derived state evolution (SE) equations for the mean-squared error of BP as a function of the iteration number. Their analysis revealed that BP is asymptotically optimal in mean-square when the SE equations have a unique fixed point. The large sparse limit analysis was extended by Guo and Wang first to general priors and power levels [13], and then to arbitrary (non-AWGN) output channels [14]. These results provided the first rigorous conditions for the optimality of BP for estimation with linear mixing and confirmed earlier predictions given by the replica method from statistical physics [15], [16].

Guo and Wang's work [13] also presented the important result that the mean-square optimality of BP could be achieved by a significantly simpler algorithm that they called *relaxed BP*. One of the problems of applying standard BP to the linear mixing estimation problem is that the computations grow exponentially with the density of the transform matrix Φ. Relaxed BP overcomes this problem by using a Gaussian approximation of the messages to linearize the computations at the output nodes. Gaussian approximations had been used in earlier BP-based methods in CDMA multiuser detection [17]–[19] and also occasionally appear in the analysis and design of LDPC codes [20], [21].

The main contribution of this paper is to extend the relaxed BP method and analysis:

- *Extensions to non-AWGN output channels:* The relaxed BP algorithm described in Guo and Wang's first paper [13] considers only AWGN output channels. The second paper [14] considers arbitrary output channels, but focuses on standard BP and only briefly mentions how to apply the relaxed BP approximations. In this paper, we work out the relaxed BP equations in full detail for general (non-AWGN) channels. Moreover, we offer some additional simplifications (see Section IV-D) to reduce the computations of relaxed BP even further. The ability to incorporate non-AWGN channels extends the scope of the relaxed BP method significantly. For example, it

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

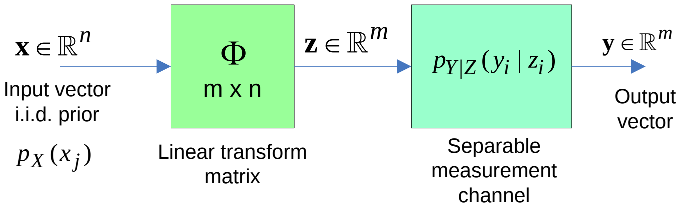

Fig. 1. Linear mixing estimation problem. A random input vector $\mathbf{x}$ is transformed by a matrix Φ, and the transformed vector $\mathbf{z}$ is passed through a separable measurement channel yielding a final output vector $\mathbf{y}$. The linear mixing estimation problem is to estimate the input vector $\mathbf{x}$ from the output vector $\mathbf{y}$ given the transform matrix Φ, prior $p_X(x_j)$ and measurement channel transition distribution $p_{Y|Z}(y_i|z_i)$.

enables the study of non-Gaussian noise processes, as well as discrete output channels that arise, for example, in pattern classification problems. We suggest some possible applications in Section II.

- *Improved convergence analysis:* A key result of Guo and Wang's state evolution (SE) analysis in [13] and [14] is that, when the measurement ratio $\beta = m/n$ is sufficiently small, relaxed BP asymptotically achieves the minimum mean-squared error (MSE) in the limit of large sparse random mixing matrices. Moreover, this minimum MSE is described by a unique fixed point to the SE equations. In this work, we extend the analysis to general β. Specifically, we show that for *any* β, there are upper and lower fixed point solutions to the SE equations. The asymptotic MSE of the relaxed BP method always converges to the upper fixed point, while the lower fixed point always provides a lower bound to the MSE of any estimator. Hence, in the case that the fixed point solution is unique, relaxed BP is asymptotically optimal.
- *Applications to compressed sensing and bounded noise estimation:* Although relaxed BP was originally developed for CDMA multiuser detection, the method can be applied to non-Gaussian estimation in a variety of applications. In this paper, we simulate relaxed BP for compressed sensing and estimation with bounded noise.

An algorithm closely related to relaxed BP is the recently developed *approximate message passing* (AMP) method proposed in [11] and analyzed further in [22]. The AMP algorithm generalizes the relaxed BP method and analysis in the special case of AWGN measurements. Unlike relaxed BP, the AMP algorithm can be applied with an arbitrary scalar estimation function so that the prior on the components of $\mathbf{x}$ do not need to be known. Also, as we will discuss in Section V, the analysis of relaxed BP is only valid under a certain large sparse limit model. This model is an approximation to the case where Φ is dense. The analysis of the AMP algorithm in [22] provides rigorous results for dense measurement matrices. An interesting open problem is whether the analysis of AMP can be extended to general output channels considered here.

A. Organization

The remainder of this paper is organized as follows. In Section II, we introduce some specific examples of the linear mixing estimation problem. Section III reviews how to apply standard BP to estimation with linear mixing. The relaxed BP algorithm is introduced in Section IV. The large sparse limit analysis is described in Section V. Section VI presents some simple simulations of the algorithm to validate the analytic results. All the proofs are developed in appendices.

II. Examples and Applications

The linear mixing model is extremely general and can be applied in a range of circumstances. We illustrate some simple examples for both the measurement channel and prior on $\mathbf{x}$.

A. Measurement Channel Examples

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

$$\mathbf{y} = \mathbf{z} + \mathbf{w} = \Phi \mathbf{x} + \mathbf{w}, \tag{3}$$

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

$$p_{Y|Z}(y_i|z_i) = \phi(y_i - z_i\,;\,\mu_w), \tag{4}$$

where $\mu_w > 0$ is the variance of the components of $\mathbf{w}$ and $\phi(v\,;\,\mu)$ is the Gaussian distribution,

$$\phi(v\,;\,\mu) = \frac{1}{\sqrt{2\pi\mu}} \exp\left(-\frac{1}{2\mu}|v|^2\right). \tag{5}$$

The AWGN channel is precisely the model considered by Guo and Wang in their original relaxed BP paper [13].

b) Non-Gaussian noise models: Since the output channel can incorporate an arbitrary separable distribution, the linear mixing model can also include the model (3) 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. We will consider this application in the numerical simulations in Section VI-C

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

$$p_{Y|Z}(y_i = 1|z_i) = \frac{1}{1 + a\exp(-z_i)}, \tag{6}$$

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 [23]: 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 Φ contains a vector of n data values associated with the ith sample. Then, using a logistic channel model such as (6), the problem of estimating the vector $\mathbf{x}$ from the labels $\mathbf{y}$ and data Φ 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

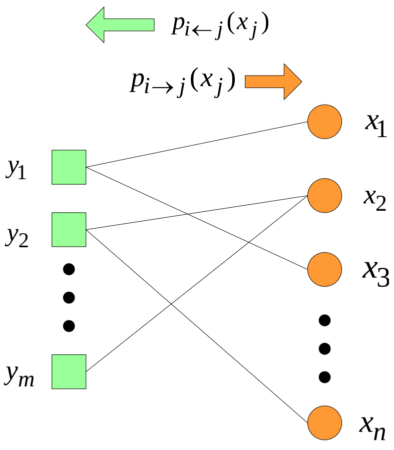

Fig. 2. Factor or Tanner graph for the linear mixing estimation problem.

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. Examples of Priors

d) Sparse priors and compressed sensing: As discussed in the introduction, one class of priors that we will consider in some detail in the simulations 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 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 [24]–[26] and fits naturally into the linear mixing framework.

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

III. Review of Standard Belief Propagation

Before describing the relaxed BP algorithm, it is useful to first review how standard BP would be applied to the linear mixing estimation problem. Standard BP associates with the transform matrix Φ a bipartite graph $G = (V, E)$, called the *factor* or *Tanner* graph illustrated in Fig. 2. 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 observations y_i, $i = 1,\ldots,m$. There is an edge $(i,j) \in E$ between the input node x_j and output node y_i if and only if $\Phi_{i,j} \neq 0$.

Given this graph, define the neighbor sets of the input and output nodes as

$$N_{\rm in}(j) = \{\, i \;:\; (i,j) \in E \,\}, \quad (7a)$$
$$N_{\rm out}(i) = \{\, j \;:\; (i,j) \in E \,\}, \quad (7b)$$

so that $N_{\rm in}(j)$ is the set of neighbors of the input index j, and $N_{\rm out}(i)$ is the set of neighbors of the output index i.

The standard BP algorithm works by iteratively passing "messages" along the edges of this graph represented as probability distributions on the variables x_j. The messages are sometimes called *beliefs*. For the linear mixing estimation problem, the standard BP algorithm can be described as follows:

1) *Initialization:* Set $t = 1$ and initialize the outgoing messages from the input nodes to

$$p^x_{i\leftarrow j}(t,x_j) = p^x_j(t,x_j) = p_X(x_j), \quad (8)$$

for all input node indices j and edges $(i,j) \in E$. This initialization simply sets the messages to the priors on the variables x_j.

2) *Mixing update:* For each edge $(i,j) \in E$, compute $p^z_{i\rightarrow j}(t, z_{i\rightarrow j})$, the distribution of the random variable

$$z_{i\rightarrow j} = \sum_{r\in N_{\rm out}(i)\neq j} \Phi_{ir}x_r, \quad (9)$$

assuming the variables x_r are independent with distributions $x_r \sim p^x_{i\leftarrow r}(t,x_r)$. Here, the sum is over indices $r \in N_{\rm out}(i)$ with $r \neq j$. Also, for each i, compute $p^z_i(t,z_i)$, the distribution of the random variable

$$z_i = \sum_{r\in N_{\rm out}(i)} \Phi_{ir}x_r. \quad (10)$$

3) *Output update:* For each edge $(i,j) \in E$, compute the likelihood function

$$p^u_{i\rightarrow j}(t,u_i) = \int p_{Y|Z}(y_i \mid u_i + z_{i\rightarrow j}) \times p^z_{i\rightarrow j}(t, z_{i\rightarrow j})\, dz_{i\rightarrow j}. \quad (11)$$

4) *Input update:* For each edge $(i,j) \in E$, compute the distribution

$$p^x_{i\leftarrow j}(t+1,x_j) \propto p_X(x_j) \prod_{\ell\in N_{\rm in}(j)\neq i} p^u_{\ell\rightarrow j}(t,\Phi_{\ell j}x_j). \quad (12)$$

Here, the $\propto$ sign indicates that the distribution is to be normalized so that it has unit integral. Also, compute the total distribution

$$p^x_j(t+1,x_j) \propto p_X(x_j) \prod_{\ell\in N_{\rm in}(j)} p^u_{\ell\rightarrow j}(t,\Phi_{\ell j}x_j). \quad (13)$$

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

When the graph G is acyclic, then it can be shown that the distributions $p^x_j(x_j)$ and $p^z_i(z_i)$ eventually converge to the true marginal distributions of the random variables x_j and z_i given the observations $\mathbf{y}$. However, for graphs with cycles, the BP algorithm in general only returns an approximation to the true marginals. An analysis of the BP algorithm is beyond the scope of this work and is covered extensively elsewhere. See, for example, [1], [2] and [27].

What is important here is to recognize the complexity of the algorithm. The difficult step is the computations of the distributions of the variables $z_{i\rightarrow j}$ and z_i in (9) and (10) in the mixing update. Suppose the output node y_i has in-degree d. That is, there are d indices j such that Φ_{ij} is non-zero. Then, the evaluation of the distributions on $z_{i\to j}$ and z_i involve the integration over $d-1$ and d components x_r. Since the complexity of this computation grows exponentially in d, the BP algorithm is only tractable when the transform matrix Φ is sparse (i.e., d is small). The point of relaxed BP is to provide an approximation to BP suitable for large, dense Φ.

IV. Relaxed Belief Propagation

A. Scalar Estimation Functions

Before describing the relaxed BP algorithm, we need to define certain functions related to scalar estimation problems at the input and output nodes. At the input nodes, we consider the problem of estimating a scalar random variable $x \sim p_X(x)$ from some scalar observation of the form

$$q = x + v, \quad v \sim \mathcal{N}(0,\mu), \tag{14}$$

where $\mu > 0$ is a noise-level and v is additive Gaussian noise independent of x. Let $F_{\rm in}(q,\mu)$ and $\mathcal{E}_{\rm in}(q,\mu)$ be the conditional mean and variance of the random variable x given the scalar observation q. Although, the functions $F_{\rm in}(q,\mu)$ and $\mathcal{E}_{\rm in}(q,\mu)$ may not have closed form expressions, they can be evaluated with one-dimensional integrals.

To analyze the output nodes, suppose that z is a scalar Gaussian random variable $z \sim \mathcal{N}(\widehat{z},\mu)$ and y has the conditional distribution $p_{Y|Z}(y|z)$. Let $p_{Y|\widehat{Z},\mu}(y|\widehat{z},\mu)$ be the likelihood function

$$p_{Y|\widehat{Z},\mu}(y|\widehat{z},\mu) = \int p_{Y|Z}(y|z)\phi(z-\widehat{z};\,\mu)\,dz, \tag{15}$$

where $\phi(z-\widehat{z};\,\mu)$ is the Gaussian p.d.f. in (5) with mean $\widehat{z}$ and variance μ. The relaxed BP algorithm is based on the derivatives of log likelihood or *score function*

$$D_r(y,\widehat{z},\mu) = -\frac{\partial^r}{\partial \widehat{z}^r}\log p_{y|\widehat{Z},\mu}(y|\widehat{z},\mu), \tag{16}$$

for $r > 0$. Again, this function can in general be evaluated numerically.

B. Algorithm

We can now describe the relaxed BP algorithm. The algorithm produces a sequence of estimates $\widehat{x}_j(t)$, $t = 0,1,\ldots$ for each variable x_j as well as estimates $\widehat{z}_i(t)$ for each transformed variable z_i. Several other intermediate estimates and variances are also computed. The steps are as follows:

1) *Initialization:* Set $t = 1$, and for every input node index j and every $(i,j) \in E$, initialize

$$\begin{aligned} \widehat{x}_{i\leftarrow j}(t) &= \widehat{x}_j(t) = \widehat{x}_{\rm init}, && (17a)\\ \mu^x_{i\leftarrow j}(t) &= \mu^x_j(t) = \mu^x_{\rm init}, && (17b)\end{aligned}$$

where $\widehat{x}_{\rm init}$ and $\mu^x_{\rm init}$ are the mean and variance of the prior $p_X(x)$.

2) *Output node, linear step:* For every $(i,j) \in E$ compute

$$\begin{aligned} \widehat{z}_{i\to j}(t) &= \sum_{r\neq j}\Phi_{ir}\widehat{x}_{i\leftarrow r}(t), && (18a)\\ \mu^z_{i\to j}(t) &= \sum_{r\neq j}|\Phi_{ir}|^2\mu^x_{i\leftarrow r}(t). && (18b)\end{aligned}$$

Also compute $\widehat{z}_i(t)$ and $\mu^z_i(t)$ similarly, but with the summation over all $r \in \{1,\ldots,n\}$.

3) *Output node, non-linear step:* For every $(i,j) \in E$ compute

$$\begin{aligned} \widehat{u}_{i\to j}(t) &= -\frac{D_1(y_i,\widehat{z}_{i\to j}(t),\mu^z_{i\to j}(t))}{D_2(y_i,\widehat{z}_{i\to j}(t),\mu^z_{i\to j}(t))}, && (19a)\\ \mu^u_{i\to j}(t) &= \frac{1}{D_2(y_i,\widehat{z}_{i\to j}(t),\mu^z_{i\to j}(t))}, && (19b)\end{aligned}$$

where $D_r(y,\widehat{z},\mu)$ are the derivatives of the negative log likelihood function in (16).

4) *Input node, linear step:* For every $(i,j) \in E$ compute

$$\begin{aligned} \widehat{q}_{i\leftarrow j}(t) &= \mu^q_{i\leftarrow j}(t)\sum_{\ell\neq i}\frac{\Phi^*_{\ell j}\widehat{u}_{\ell\to j}(t)}{\mu^u_{\ell\to j}(t)}, && (20a)\\ \mu^q_{i\leftarrow j}(t) &= \left(\sum_{\ell\neq i}\frac{|\Phi_{\ell j}|^2}{\mu^u_{\ell\to j}(t)}\right)^{-1}. && (20b)\end{aligned}$$

Also, compute $\widehat{q}_j(t)$ and $\mu^q_j(t)$ similarly, but with the summation over all $\ell \in \{1,\ldots,m\}$.

5) *Input node, non-linear step:* For every $(i,j) \in E$ compute

$$\begin{aligned} \widehat{x}_{i\leftarrow j}(t+1) &= F_{\rm in}(\widehat{q}_{i\leftarrow j}(t),\mu^q_{i\leftarrow j}(t)), && (21a)\\ \mu^x_{i\leftarrow j}(t+1) &= \mathcal{E}_{\rm in}(\widehat{q}_{i\leftarrow j}(t),\mu^q_{i\leftarrow j}(t)). && (21b)\end{aligned}$$

Similarly, for every $j = 1,\ldots,n$, compute $\widehat{x}_j(t+1)$ and $\mu^x_j(t+1)$ using $\widehat{q}_j(t)$ and $\mu^q_j(t)$. Set $t = t+1$ and return to step 2.

C. Heuristic Justification

Although we will formally analyze the relaxed BP algorithm below, it is useful to first provide a heuristic understanding of the steps. The relaxed BP algorithm is a simplification of the standard BP method where only the means and variances of the probability distributions are passed. Specifically, the terms $\widehat{x}_{i\leftarrow j}(t)$ and $\mu^x_{i\leftarrow j}(t)$ are approximations of the mean and variance of the distribution $p^x_{i\leftarrow j}(t,x_j)$ in (13) in the standard BP algorithm. In step 1, these are initialized based on the prior $p_X(x_j)$. The relaxed BP approximation does not assume that $p^x_{i\leftarrow j}(t,x_j)$ itself is Gaussian. However, relaxed BP does assume that there is a sufficient number of terms in the summation in (9) that $p^z_{i\to j}(t,z_{i\to j})$ of the standard BP algorithm is well-approximated as Gaussian. The terms $\widehat{z}_{i\to j}(t)$ and $\mu^z_{i\to j}(t)$ in (18) in step 2 of the relaxed BP algorithm are the mean and variance of this Gaussian distribution. Under this Gaussian assumption, the likelihood function $p^u_{i\to j}(t,u_i)$ in (11) is approximately given by

$$p^u_{i\to j}(t,u_i) \approx p_{Y|\widehat{Z},\mu}(y_i|\widehat{z}_{i\to j}(t),\mu^z_{i\to j}(t)), \tag{22}$$

where $p_{Y|\widehat{Z},\mu}(y|\widehat{z},\mu)$ is given in (15). Then, using the derivatives in (16), the second order approximation of (22) is given by

$$\begin{aligned}\log p^u_{i\rightarrow j}(t,u_i) &\approx -\frac{1}{2\mu^u_{i\rightarrow j}(t)}|u_i-\widehat{u}_{i\rightarrow j}(t)|^2 \\ &\qquad + O(|u_i|^3) + \text{const}, \qquad (23)\end{aligned}$$

where $\widehat{u}_{i\rightarrow j}(t)$ and $\mu^u_{i\rightarrow j}(t)$ are given in (19) in step 3 of the relaxed BP algorithm and the constant term does not depend on u_i. Now summing the log likelihoods in (23), the distribution $p^x_{i\leftarrow j}(t,x_j)$ in (13) is given by

$$\begin{aligned}\log p^x_{i\leftarrow j}(t,x_j) &= \text{const} + \log p_X(x_j) \\ &\qquad + \sum_{\ell\neq i}\log p^u_{\ell\rightarrow j}(t,\Phi_{\ell j}x_j) \\ &\approx \log p_X(x_j) - \frac{1}{2\mu^q_{i\leftarrow j}(t)}|x_j-\widehat{q}_{i\leftarrow j}(t)|^2 + \text{const}, \quad (24)\end{aligned}$$

where $q_{i\leftarrow j}(t)$ and $\mu^q_{i\leftarrow j}(t)$ are the outputs (20). The approximation in (24) is due to the fact that the sum of the $O(|\Phi_{ij}x_j|^3)$ terms is asymptotically negligible for large n. This implies that

$$p^x_{i\leftarrow j}(t,x_j) \propto p_X(x_j)\phi(x_j-\widehat{q}_{i\leftarrow j}(t)\,;\,\mu^q_{i\leftarrow j}(t)),$$

where again $\phi(\cdot\,;\cdot)$ is the Gaussian distribution in (5). This implies that, conditional on x_j, $\widehat{q}_{i\leftarrow j}(t)$ is distributed as $\mathcal{N}(x_j,\mu^q_{i\leftarrow j}(t))$. The final step, step 5, uses the scalar estimation functions in Section IV-A to compute $\widehat{x}_{i\leftarrow j}(t)$ and $\mu^x_{i\leftarrow j}(t)$, the mean and variance of x_j given $\widehat{q}_{i\leftarrow j}(t)$.

D. Algorithm Complexity and Simplifications

We next consider the complexity of the relaxed BP algorithm. The most computationally demanding steps are the nonlinear mean and variance computations in $F_{\rm in}(\cdot)$, $\mathcal{E}_{\rm in}(\cdot)$ in (21) and the derivatives, $D_r(\cdot)$ of the log likelihood function in (19). Each of these functions can be computed by a one-dimensional numerical integral. Moreover, each iteration of the relaxed BP algorithm requires exactly one evaluation of the input node functions and one evaluation of the output log likelihood derivatives per edge of the Tanner graph. Thus, the computations grow linearly with the density of the graph and unlike standard BP, the relaxed BP algorithm is tractable even for dense matrices Φ.

The relaxed BP algorithm can actually be further simplified with some small approximations. Following Tanaka and Okada's approximate BP algorithm in [18], the relaxed BP algorithm can be approximately implemented using one evaluation of the input and output nonlinear functions per vertex, as opposed to one evaluation per edge.

To implement this simplified relaxed BP algorithm, we first assume that the outgoing variances are the same to all destinations. Specifically, we replace the variance computations in the relaxed BP algorithm with

$$\begin{aligned}\mu^z_{i\rightarrow j}(t) &= \mu^z_i(t) \\ \mu^u_{i\rightarrow j}(t) &= \frac{1}{D_2(y_i,\widehat{z}_i(t),\mu^z_i(t))} \\ \mu^q_{i\leftarrow j}(t) &= \mu^q_j(t) \\ \mu^x_{i\leftarrow j}(t) &= \mu^x_j(t),\end{aligned}$$

where $\mu^q_j(t)$ and $\mu^x_j(t)$ are still computed as in the relaxed BP algorithm. In this way, the variance function $\mathcal{E}_{\rm in}(\cdot)$ and second derivative $D_2(y,\widehat{z},\mu)$ are each only computed once per vertex per iteration, as opposed to once per edge.

To reduce the evaluations of the input node function $F_{\rm in}(\cdot)$, we first observe from (20) and the definition of $q_j(t)$ that

$$q_{i\leftarrow j}(t) = q_j(t) - \mu^q_{i\leftarrow j}(t)\frac{\Phi^*_{ij}u_{i\rightarrow j}(t)}{\mu^u_{i\rightarrow j}(t)}.$$

Therefore, we can approximate the update in (21) with

$$\begin{aligned}\widehat{x}_{i\leftarrow j}(t) &= F_{\rm in}(\widehat{q}_{i\leftarrow j}(t),\mu^q_{i\leftarrow j}(t)) \\ &\approx F_{\rm in}(\widehat{q}_{i\leftarrow j}(t),\mu^q_j(t)) \\ &\approx F_{\rm in}(\widehat{q}_j(t),\mu^q_j(t)) \\ &\quad -\mu^q_{i\leftarrow j}(t)\frac{\Phi^*_{ij}u_{i\rightarrow j}(t)}{\mu^u_{i\rightarrow j}(t)}\left.\frac{\partial}{\partial q}F_{\rm in}(q,\mu^q_j(t))\right|_{q=\widehat{q}_j(t)},\!(25)\end{aligned}$$

where we have used a first order approximation for $F_{\rm in}(\cdot)$. Moreover, the partial derivative can be evaluated with the following lemma.

Lemma 1: The input MSE function defined in Section IV-A satisfies

$$\frac{\partial}{\partial q}F_{\rm in}(q,\mu) = \frac{1}{\mu}\mathcal{E}_{\rm in}(q,\mu). \qquad (26)$$

Proof: See Appendix C. ■

For the output node, we can use the fact that

$$\widehat{z}_{i\rightarrow j}(t) = \widehat{z}_i(t) - \Phi_{ij}\widehat{x}_{i\leftarrow j}(t).$$

Therefore, we can make the approximation

$$\begin{aligned}\widehat{u}_{i\rightarrow j}(t) &= D_1(y_i,\widehat{z}_{i\rightarrow j}(t),\mu^z_{i\rightarrow j}(t)) \\ &\approx D_1(y_i,\widehat{z}_{i\rightarrow j}(t),\mu^z_i(t)) \\ &\approx D_1(y_i,\widehat{z}_i(t),\mu^z_i(t)) \\ &\quad -\Phi_{ij}\widehat{x}_{i\leftarrow j}(t)D_2(y_i,\widehat{z}_i(t),\mu^z_i(t)). \quad (27)\end{aligned}$$

Using (25) and (27), we only need to evaluate the nonlinear functions $F_{\rm in}(\cdot)$ and $D_1(\cdot)$ once per edge. The analysis that we will present later does not apply to the relaxed BP algorithm with these approximations. Nevertheless, we will see in the simulations that the approximate relaxed BP algorithms behave closely to the exact implementation.

V. Large Sparse Limit Analysis

A. Modeling Assumptions

We analyze the relaxed BP algorithm in the large sparse limit developed in [12]–[14]. The large sparse limit model considers a sequence of problems parameterized by n and d. For each n and d, the transform matrix $\Phi = \Phi(n,d) \in \mathbb{R}^{m\times n}$ is of the form

$$\Phi = \frac{1}{\sqrt{d}}\mathbf{A}\mathbf{S}^{1/2}, \quad \mathbf{S} = \text{diag}(s_1,\ldots,s_n), \qquad (28)$$

where $\mathbf{A} \in \mathbb{R}^{m\times n}$ and $\mathbf{S} \in \mathbb{R}^{n\times n}$ are two matrices, and $m = m(n)$ is a deterministic function of n. The number of measurements is assumed to grow linearly in n in that

$$\lim_{n\rightarrow\infty}\frac{n}{m(n)} = \beta \qquad (29)$$

for some $\beta \geq 0$.

The components s_j in (28) are called the *scale factors* and are assumed to be i.i.d. with some probability distribution $p_S(s_j)$ that does not depend on n or d. We assume $s_j > 0$ almost surely. The diagonal scale factor matrix $\mathbf{S}$ is used to scale the powers of the components of $\mathbf{x}$. Specifically, multiplication by $\mathbf{S}^{1/2}$ scales the variance of the jth component of $\mathbf{x}$ by a factor s_j. In this way, the scale factors can be used to capture variations in the power of the components of $\mathbf{x}$ that are known *a priori* at the estimator. Variations in the power of $\mathbf{x}$ that are not known to the estimator should be captured in the distribution of $\mathbf{x}$.

The matrix $\mathbf{A}$ is deterministic, and we evaluate the performance of the relaxed BP algorithm on a deterministic sequence of input and output indices $i = i(n,d)$ and $j = j(n,d)$. The sequence of matrices $\mathbf{A}$ and indices i and j are assumed to satisfy the following conditions:

Assumption 1: Let $\mathbf{A} = \mathbf{A}(n,d)$ be a sequence of deterministic matrices in the factorization of $\Phi = \Phi(n,d)$ in (28). Let $i = i(n,d)$ and $j = j(n,d)$ be a deterministic sequence of indices. Let $t > 0$ be some fixed iteration number of the relaxed BP algorithm. Assume that $\mathbf{A}$, i and j satisfy the following:

(a) For every n and d, (i,j) is an edge in the Tanner graph G associated with Φ.

(b) The *computation subgraphs* $G_i(t)$ and $G_j(t)$ of the Tanner graph taken a depth of $2t$ hops from the output node i and input node j are trees. Precise definitions of these computation subgraphs are given in Appendix D.

(c) All the nodes in the subgraphs $G_i(t)$ and $G_j(t)$ have degrees bounded above by d.

(d) For all output nodes ℓ in the subgraph $G_i(t)$, we have the limits

$$\lim_{d\to\infty}\lim_{n\to\infty}\frac{1}{d}\sum_{r\in N_{\rm out}(\ell)}|a_{\ell r}|^2 = \beta, \tag{30a}$$

$$\lim_{d\to\infty}\lim_{n\to\infty}\frac{1}{d^{3/2}}\sum_{r\in N_{\rm out}(\ell)}|a_{\ell r}|^3 = 0. \tag{30b}$$

For all input nodes r in the subgraph $G_j(t)$, we have the limits

$$\lim_{d\to\infty}\lim_{n\to\infty}\frac{1}{d}\sum_{\ell\in N_{\rm in}(r)}|a_{\ell r}|^2 = 1, \tag{30c}$$

$$\lim_{d\to\infty}\lim_{n\to\infty}\frac{1}{d^{3/2}}\sum_{\ell\in N_{\rm in}(r)}|a_{\ell r}|^3 = 0. \tag{30d}$$

As in [12]–[14], the key assumption is that the Tanner graph G associated with the transform matrix Φ is locally tree-like around the components of interest i and j. The assumption is common in the study of BP algorithms as it makes the messages independent. This local tree-like property is only possible with the graph being sparse. This sparsity assumption is brought out explicitly by bounding the input and output degrees of the Tanner graph.

Assumption 1 uses a deterministic model for the $\mathbf{A}$ as opposed to the random matrix model with i.i.d. components studied in [12]–[14]. The deterministic model simplifies some of the proofs. In particular, the input and output degrees are deterministically bounded as opposed to be bounded on average – which simplifies some of the convergence arguments.

In the large sparse limit analysis, we first let $n \to \infty$ with m growing linearly with n and keeping d fixed. This enables the local-tree like properties. We then let $d \to \infty$, which will enable the use of a Central Limit Theorem approximation.

This order of limits is critical. Unfortunately, to analyze dense matrices, one would like an analysis where d can grow with n. Indeed, if the matrix is completely dense, we would like $d = m(n)$. Unfortunately, the large sparse limit analysis that we rely on here requires that we consider the two limits separately; it thus represents an approximation to the actual problem. Nevertheless, we will see in simulations that large sparse limit analysis appears to predict the behavior with dense matrices as well.

More sophisticated analysis techniques developed recently in [22] enable the study of dense matrices without the order of limits above. One possible avenue of future research would be to see if that analysis can be applied to the relaxed BP algorithm with general (non-AWGN) output channels as well.

B. Large Sparse Limit Convergence

Under the large sparse limit model, define the random vectors

$$\begin{aligned}
\theta^x_{i\leftarrow j}(n,d,t) &= (x_j, s_j, \widehat{x}_{i\leftarrow j}(t), \mu^x_{i\leftarrow j}(t)) &(31a)\\
\theta^x_{j}(n,d,t) &= (x_j, s_j, \widehat{x}_{j}(t), \mu^x_{j}(t)) &(31b)\\
\theta^z_{i\to j}(n,d,t) &= (z_{i\to j}, \widehat{z}_{i\to j}(t), \mu^z_{i\to j}(t)) &(31c)\\
\theta^z_{i}(n,d,t) &= (z_{i}, \widehat{z}_{i}(t), \mu^z_{i}(t)), &(31d)
\end{aligned}$$

where the dependence on n and d on the right-hand side of the equations is implicit. Our goal is to describe the large sparse limit behavior of these random vectors.

A key result of [14] is that the large sparse limit behavior of BP is described by a set of simple *state evolution* (SE) equations, which can be described as follows: Given $\mathcal{E}_{\rm in}(q,\mu)$ in Section IV-A, define

$$\begin{aligned}
\overline{\mathcal{E}}_{\rm in}(\mu,s) &= \mathbf{E}\left[\mathcal{E}_{\rm in}(q,\mu/s)|s\right] &(32a)\\
\overline{\mathcal{E}}_{\rm in}(\mu) &= \mathbf{E}\left[s\mathcal{E}_{\rm in}(q,\mu/s)\right], &(32b)
\end{aligned}$$

where the expectation is taken over the scalar random variables $s \sim p_S(s)$ and q given by (14) with $x \sim p_X(x)$. We will call $\overline{\mathcal{E}}_{\rm in}(\mu)$ the *input node MSE function*. In addition to the works [13], [14], this function appeared in Guo and Verdú's replica analysis of MSE estimation [16] and related works [28], [29]. Variants also appear in the analysis of the AMP algorithm [11], [22].

At the output node, let

$$\mu^z_{\rm init} = \beta\mathbf{E}(s)\mu^x_{\rm init}, \tag{33}$$

where $\mu^x_{\rm init}$ is variance of x_j according to the prior $p_X(x_j)$, and the expectation is over $s \sim p_S(s)$. Then, for $\mu \leq \mu^z_{\rm init}$, Guo and Wang [14] define the *output node MSE function* as

$$\overline{\mathcal{E}}_{\rm out}(\mu) = \frac{1}{\mathbf{E}\left[D_2(y,\widehat{z},\mu)\right]}, \tag{34}$$

where $D_2(y, \widehat{z}, \mu)$ is the derivative (16) of the score function. The expectation in (34) is taken over

$$(z, \widehat{z}) \sim \mathcal{N}(0, P_z(\mu)), \tag{35}$$

where $P_z(\mu)$ is the covariance matrix

$$P_z(\mu) = \left(\begin{array}{cc} \mu^z_{\rm init} & \mu^z_{\rm init} - \mu \\ \mu^z_{\rm init} - \mu & \mu^z_{\rm init} - \mu \end{array} \right), \tag{36}$$

and the conditional distribution of y given z is given by $p_{Y|Z}(y|z)$.

Now consider the recursion

$$\begin{aligned} \mu^q(t) &= \overline{\mathcal{E}}_{\rm out}(\mu^z(t)), & (37a) \\ \mu^x(t+1, s) &= \overline{\mathcal{E}}_{\rm in}(\mu^q(t), s), & (37b) \\ \mu^z(t+1) &= \beta\overline{\mathcal{E}}_{\rm in}(\mu^q(t)), & (37c) \end{aligned}$$

defined for $t \geq 1$. We can also write (37) with the single equation

$$\mu^z(t+1) = \beta\overline{\mathcal{E}}_{\rm in}\left[\overline{\mathcal{E}}_{\rm out}(\mu^z(t))\right]. \tag{38}$$

In [14], the equations (37) (or the single equation version (38)) are called the *state evolution* equations for BP as they describe the evolution of the error variances.

We consider two possible initial conditions for this recursion: one low value and one high value. The low sequence will be initialized with $\mu^z(1) = \mu^z_{\rm lo}(1) = 0$, and the high sequence will be initialized with $\mu^z(1) = \mu^z_{\rm hi}(1) = \mu^z_{\rm init}$ in (33). We will use the subscripts as in $\mu^z_{\rm lo}(t)$ and $\mu^z_{\rm hi}(t)$ to differentiate between the two sequences.

Now, for $t \in \mathbb{Z}^+$, let $\theta^x(t)$ be the random vector

$$\theta^x(t) = \left(x, s, F_{\rm in}(q, \mu), \mathcal{E}_{\rm in}(q, \mu)\right), \tag{39}$$

where $x \sim p_X(x)$, $s \sim p_S(s)$, q is distributed as (14), and $\mu = \mu^q(t-1)/s$ with $\mu^q(t-1)$ being the (deterministic) quantity in the state evolution (SE) equation (37). To initialize, let

$$\theta^x(1) = \left(x, s, \widehat{x}_{\rm init}, \mu^x_{\rm init}\right), \tag{40}$$

where $\widehat{x}_{\rm init}$ and $\mu^x_{\rm init}$ are the mean and variance of the prior of $p_X(x)$. We will see below that when we use $\mu^q(t) = \mu^q_{\rm hi}(t)$, the SE output with the "high" initial condition $\theta^x(t)$ describes the limit of the random vector $\theta^x_{i\leftarrow j}(n, d, t)$ in (31). When $\mu^q(t) = \mu^q_{\rm lo}(t)$, $\theta^x(t)$ is the limit of a related quantity for a certain "genie-aided" algorithm (see Appendix B).

Also, for all $t \in \mathbb{Z}^+$, define the random vector

$$\theta^z(t) = (z, \widehat{z}, \mu^z(t)), \tag{41}$$

where $\mu^z(t)$ is the output of the state evolution equations (37) and $(z, \widehat{z}) \sim \mathcal{N}(0, P_z(\mu^z(t)))$.

Theorem 1: Consider the relaxed BP algorithm under the large sparse limit model above with transform matrix Φ and indices i and j satisfying Assumption 1 for some *fixed* iteration number t. Then:

(a) The random vectors in (31) converge in distribution as follows:

$$\begin{aligned} \lim_{d\rightarrow\infty}\lim_{n\rightarrow\infty} \theta^x_{i\leftarrow j}(n, d, t) &= \theta^x(t) & (42a) \\ \lim_{d\rightarrow\infty}\lim_{n\rightarrow\infty} \theta^x_j(n, d, t) &= \theta^x(t) & (42b) \\ \lim_{d\rightarrow\infty}\lim_{n\rightarrow\infty} \theta^z_{i\leftarrow j}(n, d, t) &= \theta^z(t) & (42c) \\ \lim_{d\rightarrow\infty}\lim_{n\rightarrow\infty} \theta^z_i(n, d, t) &= \theta^z(t), & (42d) \end{aligned}$$

where the random vectors $\theta^x(t)$ and $\theta^z(t)$ are defined as above with $\mu^q(t) = \mu^q_{\rm hi}(t)$ and $\mu^z(t) = \mu^z_{\rm hi}(t)$.

(b) The error variances satisfy the limits

$$\begin{aligned} \lim_{d\rightarrow\infty}\lim_{n\rightarrow\infty} \mathbf{E}\left[|x_j - \widehat{x}_j(t)|^2 | s_j = s\right] = \mu^x_{\rm hi}(t, s), && (43a) \\ \lim_{d\rightarrow\infty}\lim_{n\rightarrow\infty} \mathbf{E}\left[|z_i - \widehat{z}_i(t)|^2\right] = \mu^z_{\rm hi}(t), && (43b) \end{aligned}$$

where $\mu^x_{\rm hi}(t, s)$ and $\mu^z_{\rm hi}(t)$ are the output of the SE equations (37) with the "hi" initial condition.

(c) The minimum conditional error variance of x_j and z_i given Φ and $\mathbf{y}$ satisfy the asymptotic lower bounds

$$\begin{aligned} \lim_{d\rightarrow\infty}\lim_{n\rightarrow\infty} \mathbf{E}\left[\mathbf{var}(x_j|\mathbf{y}, \Phi) | s_j = s\right] \geq \mu^x_{\rm lo}(t, s), && (44a) \\ \lim_{d\rightarrow\infty}\lim_{n\rightarrow\infty} \mathbf{E}\left[\mathbf{var}(z_i|\mathbf{y}, \Phi)\right] \geq \mu^z_{\rm lo}(t), && (44b) \end{aligned}$$

where $\mu^x_{\rm lo}(t, s)$ and $\mu^z_{\rm lo}(t)$ are the output of the SE equations (37) with the "lo" initial condition.

Proof: See Appendices E and F. ■

The performance bounds in parts (a) and (b) are largely identical to the results in [14] except that they apply to relaxed BP instead of BP. This is our main result: in the large sparse limit model, relaxed BP and standard BP have the identical asymptotic behavior. The lower bound in part (c) of the theorem is also very close to results in [14] and just repeated here for completeness.

Part (a) of the theorem provides a simple scalar characterization for this asymptotic behavior. Specifically, using the definition of $\theta^x(t)$ in (39), Theorem 1 shows that the componentwise behavior of the relaxed BP follows a *scalar equivalent model* as shown in Fig. 3: The component x_j is first corrupted by Gaussian noise yielding a noisy component q_j. The relaxed BP estimate $\widehat{x}_j(t)$ then behaves identically to the optimal scalar MMSE estimate of x_j from the AWGN measurement q_j. From this scalar equivalent joint distribution of the components and their estimates, one can compute any componentwise separable performance metric such as mean-squared error or probability of detection.

The effective Gaussian noise levels in the scalar models are described by $\mu^z_{\rm hi}(t)$ and $\mu^q_{\rm hi}(t)$ from the state evolution equations (37). Since the state evolution equations can be evaluated easily with numerical integration, Theorem 1 thus provides a simple, computationally-tractable method for exactly characterizing the performance of the relaxed BP algorithm.

Part (b) shows that the SE outputs $\mu^x_{\rm hi}(t, s)$ and $\mu^z_{\rm hi}(t)$ respectively describe the asymptotic estimation error on the components x_j and prediction error on the outputs z_i. Part (c) provides corresponding lower bounds on these error variances for any estimator.

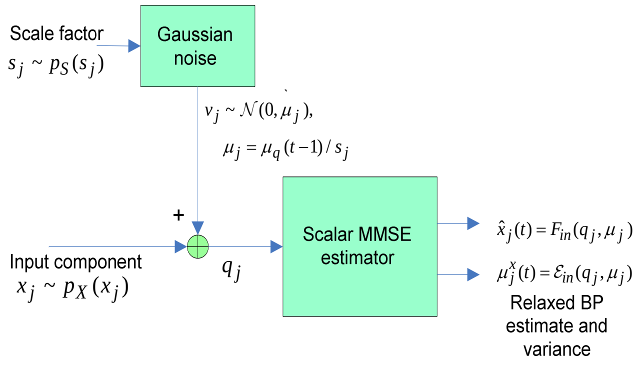
Scale factor
$s_j \sim p_S(s_j)$
Gaussian noise
$v_j \sim \mathcal{N}(0,\mu_j)$,
$\mu_j = \mu_q(t-1)/s_j$
+
Input component
$x_j \sim p_X(x_j)$
q_j
Scalar MMSE estimator
$\hat{x}_j(t) = F_{in}(q_j,\mu_j)$
$\mu_j^x(t) = \mathcal{E}_{in}(q_j,\mu_j)$
Relaxed BP estimate and variance

Fig. 3. Equivalent scalar model of the relaxed BP algorithm. The asymptotic behavior of the relaxed BP estimate $\widehat{x}_j(t)$ of a component x_j is identical to the output of an MMSE estimator with AWGN noise. The effective noise is scaled by the factor s_j corresponding to the component x_j.

C. Convergence over Iteration and Mean-Square Optimality

Theorem 1 describes the asymptotic behavior of the relaxed BP algorithm for a *fixed* iteration number t. Our second result describes the behavior of the relaxed BP estimates as $t \to \infty$.

Theorem 2: Consider the state evolution equations (37). Suppose that $\overline{\mathcal{E}}_{\rm in}(\mu)$ and $\overline{\mathcal{E}}_{\rm out}(\mu)$ are continuous. Then, the SE equations have at least one fixed point with $0 \leq \mu^z \leq \mu^z_{\rm init}$. Also:

(a) With the "hi" initial condition, $\mu^z(1) = \mu^z_{\rm hi}(1) = \mu^z_{\rm init}$, the sequences $\mu^z_{\rm hi}(t)$, $\mu^x_{\rm hi}(t,s)$ and $\mu^q_{\rm hi}(t)$ decrease monotonically to the largest fixed point of the SE equations (37).

(b) With the "lo" initial condition, $\mu^z(1) = \mu^z_{\rm lo}(1) = 0$, the sequences $\mu^z_{\rm lo}(t)$, $\mu^x_{\rm lo}(t,s)$ and $\mu^q_{\rm lo}(t)$ increase monotonically to the smallest fixed point of the SE equations (37).

Proof: See Appendix G. ■

The theorem is similar to the convergence result in [13] except that it applies to all β. The importance of the result is that it shows that relaxed BP provably converges in the limit of large iterations, and the asymptotic error variance of relaxed BP and the corresponding error lower bounds are both fixed points of the SE equations. A corollary of this result is that, when the fixed points of the SE equations (37) are unique, the error variance of relaxed BP and the corresponding lower bound agree. The result thus gives an easily verifiable condition under which relaxed BP is asymptotically mean-square optimal.

VI. Numerical Simulations

The large sparse limit analysis of the relaxed BP algorithm in Section V is theoretically exact only in the asymptotic limit of large dimensions. Also, the analysis assumes a certain scaling where the measurement matrix Φ remains sparse. To test the accuracy of the large sparse limit model for finite problems with dense Φ, we conducted the following simple numerical experiments.

A. Gauss-Bernoulli Prior with an AWGN Output Channel

In the first experiment, the vector $\mathbf{x}$ was generated with i.i.d. components x_j with a Gauss-Bernoulli distribution given by

$$x_j \sim \left\{ \begin{array}{ll} \mathcal{N}(0,1/\rho) & \text{with prob } = \rho, \\ 0 & \text{with prob } = 1-\rho. \end{array} \right. \tag{45}$$

Here ρ is the *sparsity ratio* and represents the average fraction of non-zero components in $\mathbf{x}$. The experiments below used the value $\rho = 0.1$. We chose this Gaussian mixture model since it is a simple example of a sparse prior used in compressed sensing. This prior is also used in the numerical validation of the replica method in [28].

The components of the measurement matrix Φ were generated as i.i.d. zero-mean Gaussians. Even though this matrix is dense, we will see that the large sparse limit analysis predicts the behavior of the relaxed BP estimator well.

For the measurement channel in this first experiment, we assumed an AWGN output channel (3), where the noise $\mathbf{w}$ also has i.i.d. zero-mean Gaussian components. The noise variance, μ_w, of the components of $\mathbf{w}$ was selected such that $\mathsf{SNR}_0 = 10$ dB, where SNR_0 is the signal-to-noise ratio,

$$\mathsf{SNR}_0 = 10\log_{10}\left(\frac{\mathbf{E}\|\Phi\mathbf{x}\|^2}{n\mu_w}\right). \tag{46}$$

As discussed in [28], SNR_0 is the effective SNR that an estimator would see in estimating any one component of x_j with the other $n-1$ components of $\mathbf{x}$ known.

Fig. 4 shows the median normalized squared-error (NSE) as a function of the iteration number in the relaxed BP algorithm for this model. In all the numerical experiments, we used the relaxed BP algorithm with the simplifications described in Section IV-D. The simulation was conducted with 1000 random realizations of the problem, and for each realization, we measured the NSE given by

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

where $\widehat{\mathbf{x}}$ is the estimate of $\mathbf{x}$. The NSE represents the average error over the n components of $\mathbf{x}$. Fig. 4 plots the median of these NSE values over the 1000 Monte Carlo trials. The figure shows the median NSE for vector dimensions $n = 100$ and 500 and $\beta = n/m = 2$ and 3.

The points marked "pred (SE)" are the NSE values as predicted by the state evolution equations (37). We see that the state evolution equations, which are theoretically exact for infinite n, provide an excellent match (within 0.1 dB) with the simulated values when $n = 500$. At the shorter length of $n = 100$, the SE equations still provide a good match, although there is a small steady error of about 0.2 dB when $\beta = 2$ and 0.8 dB when $\beta = 3$.

In Fig. 4, we plotted the median NSE since there is actually considerable variation in the NSE over the random realizations of the problem parameters. To illustrate the degree of variability, Fig. 5 shows the CDF of the NSE values over the 1000 Monte Carlo trials. We see that there is a large variation in the NSE, especially at the smaller dimension $n = 100$. This means that although the median performance may be

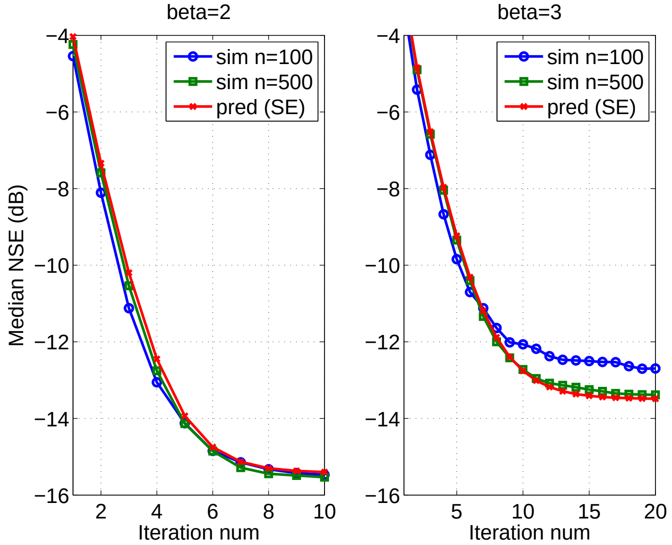

Fig. 4. State evolution prediction. The normalized squared-error as predicted by the state evolution equations (37) as a function of the iteration number is compared against the simulated value for $n = 100$ and 500. The simulation is based on a sparse Gauss-Bernoulli model. See text for details.

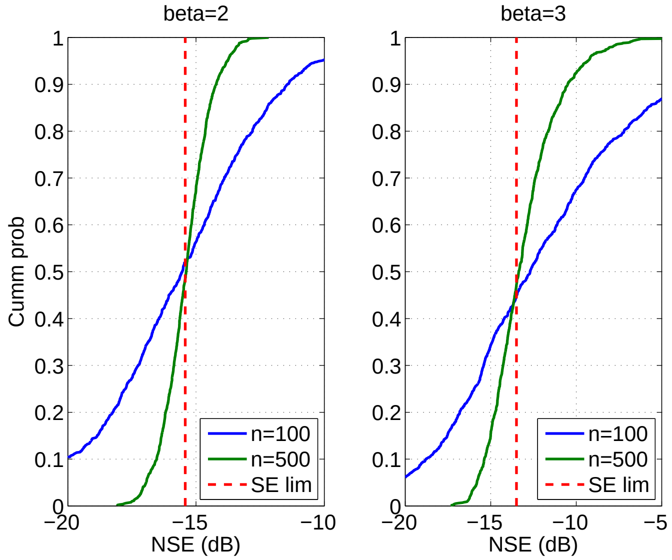

Fig. 5. Performance variation. Plotted is the simulated CDF of the NSE, averaged over the components in the vector $\mathbf{x}$, but accounting for variations in the random problem parameters. The CDF for the value of $n = 500$ shows less variation than $n = 100$ and closer to concentration around the state evolution limit (SE lim).

good, there is still a significant chance that the algorithm could perform well below the median on any particular realization.

As one might expect, at the higher dimension of $n = 500$, the level of variability is reduced and performance begins to concentrate around the density evolution limit. However, even at $n = 500$, the variation is not insignificant. As a result, caution should be exercised in using the SE predictions with short to medium block lengths.

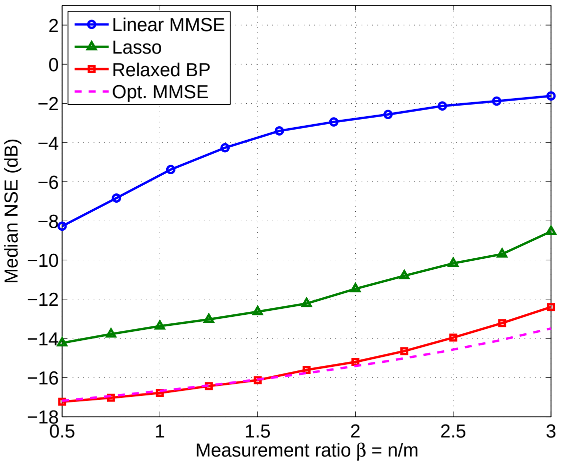

Fig. 6. Performance comparison of relaxed BP with other sparse estimation methods.

B. Comparison to Other Sparse Detection Methods

Fig. 6 plots the squared-error performance of the relaxed BP algorithm varying the measurement ratio $\beta = n/m$ and holding $n = 100$. For each value of β, the points labeled “relaxed BP” show the median NSE after 20 iterations of the relaxed BP algorithm.

Also plotted is the theoretical optimal MMSE performance as predicted by the lower bound, Theorem 1(c). In this experiment, we observed only one fixed point for the SE equations for all the values of β. So the lower bound on the MSE in Theorem 1(c) equals the theoretical asymptotic performance of relaxed BP given in Theorem 1(b). We see that the relaxed BP algorithm at $n = 100$ performs very close to the asymptotic optimal performance for values of β up to approximately 2. For larger values of β, there is a small gap between the performance of the relaxed BP algorithm and the optimal performance. The gap grows to 0.8 dB at $\beta = 3$. As discussed in Fig. 4, this gap decreases at higher values of n.

Fig. 6 also shows the performance of two other simple algorithms. The top curve is the median NSE for optimal linear MMSE estimation, and the curve labeled “lasso” is the MSE from the lasso algorithm of [30]. The lasso method is based on an ℓ_1-relaxation of the optimal estimator and is widely-used for sparse estimation problems in compressed sensing. In this experiment, the regularization weighting in the lasso estimator was optimized as described in [28].

We see that the relaxed BP algorithm offers some gain over either of these methods. Of course, with the interest in compressed sensing, there is now a plethora of methods for estimating sparse vectors. It is likely that other methods, including possible modifications of lasso, can obtain a similar performance as relaxed BP. A complete comparison of relaxed BP against these methods is beyond the scope of this work. What is important is that relaxed BP provides a unified, systematic method for a large class of problems, such that when applied to certain compressed sensing problems, it gives near optimal performance.

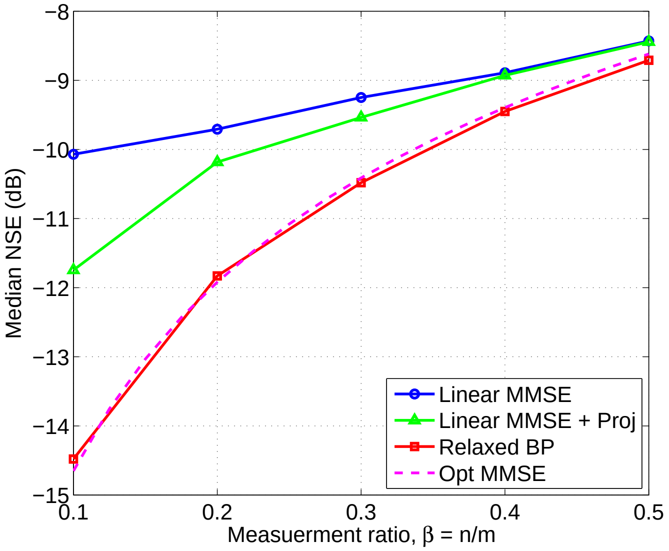

Fig. 7. Relaxed BP algorithm with a Gaussian prior and bounded noise output channel. The plot compares the simulated relaxed BP performance against the predicted performance based on density evolution. Also shown is the performance of the linear MMSE estimator with and without projection to the consistent set.

C. *Estimation with Bounded Noise*

To validate the relaxed BP method and analysis for non-AWGN output channels, we next considered a bounded, uniform noise channel. Specifically, we assumed that the output channel is given by (3) where the components of the noise vector $\mathbf{w}$ are i.i.d. and uniformly distributed in an interval $[-\delta, \delta]$ for some $\delta > 0$. Among other applications, this bounded noise model arises in the study of subtractive dithered quantization [31], [32], where the uncertainty interval corresponds to a quantization region.

Unfortunately, optimal MMSE estimation with bounded uniform noise involves an integration over an n-dimensional polytope, which is generally computationally intractable. However, the relaxed BP algorithm can be readily applied to the relaxed BP problem with bounded noise providing a simple, computationally-tractable algorithm for this problem.

Fig. 7 shows a simulation of the relaxed BP algorithm with a bounded uniform output noise channel. The simulation used a vector $\mathbf{x}$ with $n = 50$ zero-mean i.i.d. Gaussian components. Similar to the previous experiment, we again used a measurement matrix Φ with Gaussian i.i.d. components. Also, bounded uniform noise in the interval $[-\delta, \delta]$ results in a noise variance of $\mu_w = \delta^2/3$. In this experiment, the noise level δ was adjusted such that SNR_0 in (46) was equal to 10 dB. We varied the values of the measurement ratio $\beta = n/m$, and for each value of β, the points labeled “Relaxed BP” in Fig. 7 plots the median NSE over 1000 Monte Carlo trials of the relaxed BP algorithm, using 20 iterations in each relaxed BP run.

As in the previous experiment, the SE equations have a unique fixed point, and thus relaxed BP is theoretically asymptotically optimal with a minimum variance predicted by the SE fixed point. The curve labeled “Opt MMSE” shows this theoretical asymptotic minimum squared error. We see that the median squared error of relaxed BP at $n = 50$ matches the theoretical asymptotic performance well.

Fig. 7 also compares the relaxed BP method to two other simple algorithms. One is the linear MMSE estimator, which is equivalent to the MMSE estimator assuming Gaussian noise. The second estimator, shown in the curve labeled “Linear MMSE + Proj,” is the linear MMSE estimate followed by a projection step. A key observation of the work [33], [34] is that any estimate (including the linear MMSE estimate) can be improved by simply projecting the estimate onto the set of vectors $\mathbf{x}$ *consistent* with the bounded noise. An estimate $\widehat{\mathbf{x}}$ is consistent with the noise if $\|\mathbf{y} - \Phi\widehat{\mathbf{x}}\|_\infty \leq \delta$. The works [33], [34] show that projecting to a consistent estimate always reduces the squared-error and can offer significant gains at low values of β (what is called high oversampling). Similar results and algorithms have been reported elsewhere [35]–[37]. The figure shows that projecting the linear MMSE estimate does indeed offer reductions in the squared error, especially for small β. However, the relaxed BP algorithm, in comparison, is even better.

The reason that the relaxed BP algorithm shows a performance improvement over the projected linear MMSE estimate is that projecting the linear MMSE estimate will generally result in a point only on the boundary of the consistent set. In contrast, the relaxed BP algorithm will attempt to find the centroid of the consistent region, which will likely have a smaller error variance.

VII. Conclusions

We have presented an extension to Guo and Wang’s relaxed BP method in [13] to non-AWGN measurements. The algorithm applies to a large class of estimation problems involving linear mixing and arbitrary separable input and output distributions. Unlike standard BP, relaxed BP is computationally tractable even for dense measurement matrices. Our main result shows that, in the large sparse limit, relaxed BP achieves the same asymptotic behavior as standard BP as described in [14]. In particular, when certain state evolution equations have unique fixed points, relaxed BP is mean-square optimal. Given the generality of the algorithm, its computational simplicity and provable performance guarantees, we believe that relaxed BP can have wide ranging applications. We have demonstrated the algorithm in two well-known NP-hard problems: compressed sensing and estimation with bounded noise.

The main theoretical limitation of the work is that it applies to large sparse random matrices, where the density of the measurement matrix must grow at a much slower rate than the matrix dimension. An interesting avenue of future work would be to see if the dense matrix analysis of the AMP algorithm in [11] and [22] can be extended to relaxed BP.

Appendix A
Preliminary Convergence Results

Before proving our main result, the next number of appendices develop some preliminary results. We begin in this appendix with some simple extensions to the Law of Large

Numbers and the Central Limit Theorem. We omit the proofs as the results can be proven with minor modifications to standard arguments using characteristic functions [38].

Lemma 2 (Modified Law of Large Numbers): For each n and d, let $x^d_{n,i} \in \mathbb{R}$, $i = 1,\ldots,d$ be a set of independent (though not necessarily identically distributed) random variables satisfying

$$\lim_{d\to\infty}\lim_{n\to\infty} x^d_{n,i} = x \sim p_X(x),$$

where the convergence is in distribution and $p_X(x)$ is the distribution for the limiting random variable x and $i = i(n,d) \in \{1,\ldots,d\}$ is any deterministic sequence. Assume x has bounded second moments. Let $b^d_{n,i}$ be a set of non-negative deterministic constants such that

$$\lim_{d\to\infty}\lim_{n\to\infty} \frac{1}{d}\sum_{i=1}^{d} b^d_{n,i} = 1.$$

Then, the limit

$$\lim_{d\to\infty}\lim_{n\to\infty} \frac{1}{d}\sum_{i=1}^{d} b^d_{n,i}x^d_{n,i} = \mathbf{E}(x)$$

holds in distribution.

Lemma 3 (Modified Central Limit Theorem): Let $x^d_{n,i}$ be as in Lemma 2 such that, for any deterministic sequence of indices $i = i(n,d) \in \{1,\ldots,d\}$, we have the limit

$$\lim_{d\to\infty}\lim_{n\to\infty} \sqrt{d}|\mathbf{E}(x^d_{n,i}) - \mathbf{E}(x)| = 0.$$

Also, suppose that $a^d_{n,i}$ is a deterministic sequence of scalars such that that

$$\begin{aligned}\lim_{d\to\infty}\lim_{n\to\infty} \frac{1}{d}\sum_{i=1}^{d} |a^d_{n,i}|^2 &= 1,\\ \lim_{d\to\infty}\lim_{n\to\infty} \frac{1}{d^{3/2}}\sum_{i=1}^{d} |a^d_{n,i}|^3 &= 0.\end{aligned}$$

Then,

$$\lim_{d\to\infty}\lim_{n\to\infty} \frac{1}{\sqrt{d}}\sum_{i=1}^{d} a^d_{n,i}(x^d_{n,i} - \mathbf{E}(x)) = \mathcal{N}(0,\mathbf{var}(x))$$

where the limit is in distribution.

Appendix B
Genie Algorithm

As stated earlier, part (c) of Theorem 1 is not new and can be found in [13], [14]. Their proof is restated here only for completeness. Using similar arguments as [39], their proof considers a "genie" or "oracle-aided" algorithm that has, as side information, knowledge of certain subsets of the components x_j.

The genie algorithm is defined as follows: In step 1 of the relaxed BP algorithm in Section IV-B, we simply replace the initialization (17) at $t = 1$ with

$$\begin{aligned}\widehat{x}_{i\leftarrow j}(t) &= \widehat{x}_j(t) = x_j, &\text{(47a)}\\ \mu^x_{i\leftarrow j}(t) &= \mu^x_j(t) = 0, &\text{(47b)}\end{aligned}$$

where x_j is the true value of the component. Otherwise, all the steps of the algorithm are the same as the regular relaxed BP algorithm.

We will see that while the error in the regular BP algorithm improves with each iteration, the genie algorithm starts with zero error and then increases. The performance of the true optimal estimator is "sandwiched" somewhere between the genie and regular relaxed BP algorithms, and thus consideration of the regular and relaxed algorithms provide upper and lower bounds on the optimal performance.

Appendix C
Proof of Lemma 1

Fix $\mu > 0$ and for $r = 0,1,2,\ldots$, define the functions

$$A_r(q) = \int x^r p_X(x)\phi(q-x\,;\,\mu)\,dx. \tag{48}$$

Then, the conditional mean $F_{\rm in}(q,\mu)$ and variance $\mathcal{E}_{\rm in}(q,\mu)$ are given by

$$\begin{aligned}F_{\rm in}(q,\mu) &= \frac{A_1(q)}{A_0(q)}, &\text{(49a)}\\ \mathcal{E}_{\rm in}(q,\mu) &= \frac{A_2(q)}{A_0(q)} - \frac{A_2^2(q)}{A_0^2(q)}. &\text{(49b)}\end{aligned}$$

Now, taking the derivative of the Gaussian distribution $\phi(\cdot\,;\,\mu)$ in (5), it is easily verified that

$$\frac{\partial}{\partial q}\phi(q-x\,;\,\mu) = \frac{x-q}{\mu}\phi(q-x\,;\,\mu).$$

Bringing this derivative inside the integral (48) then shows that

$$\frac{\partial A_r(q)}{\partial q} = \frac{1}{\mu}\left(A_{r+1}(q) - qA_r(q)\right). \tag{50}$$

Applying (50) to (49) we obtain

$$\begin{aligned}\frac{\partial F_{\rm in}(q,\mu)}{\partial q} &= \frac{\partial}{\partial q}\frac{A_1(q)}{A_0(q)}\\ &= \frac{(A_2(q)-qA_1(q))A_0(q) - (A_1(q)-qA_0(q))A_1(q)}{\mu A_0^2(q)}\\ &= \frac{A_2(q)}{\mu A_0(q)} - \frac{A_2^2(q)}{\mu A_0^2(q)} = \frac{1}{\mu}\mathcal{E}_{\rm in}(q,\mu).\end{aligned}$$

Appendix D
Computation Subgraphs and Local Tree-like Properties

An essential assumption of the large sparse limit analysis is that the Tanner graph is *locally tree-like*. To describe this property more precisely, we review some standard definitions and results that can be found in any description of BP such as [40]. Consider the Tanner graph G for the linear mixing problem defined in Section III. For $t = 0,1,2,\ldots$, recursively define a sequence of *computation subgraphs*, $G_j(t)$, $G_i(t)$, $G_{i\leftarrow j}(t)$ and $G_{i\rightarrow j}(t)$, as follows:

1) *Initialize:* Set $t = 1$, and for all $(i,j) \in E$ let $G_{i\leftarrow j}(t)$ and $G_j(t)$ be the empty subgraphs.
2) *Output update:* For all $(i,j) \in E$, let $G_{i\rightarrow j}(t)$ be the subgraph containing, for all $r \in N_{\rm out}(i) \neq j$:
   (a) the edges (i,r); and

(b) the subgraphs $G_{i\leftarrow r}(t)$.

Similarly, define the subgraph $G_i(t)$ to be the subgraph using all $r \in N_{\rm out}(i)$.

3) *Input update:* For all $(i,j) \in E$, let $G_{i\leftarrow j}(t+1)$ be the subgraph containing the node x_j and for all $\ell \in N_{\rm in}(j) \neq i$:

(a) the edges (ℓ, j);

(b) the output nodes y_ℓ; and

(c) the subgraphs $G_{\ell\rightarrow j}(t)$.

Similarly, define the subgraph $G_j(t)$ to be the subgraph using all $\ell \in N_{\rm in}(j)$. Set $t = t+1$ and return to step 1.

Now let $H_{i\rightarrow j}(t)$ be the sigma algebra generated by the components y_ℓ contained in the computation subgraph $G_{i\rightarrow j}(t)$. Similarly, let $H_{i\leftarrow j}(t)$ be the sigma algebra generated by the components y_ℓ contained in the computation subgraph $G_{i\leftarrow j}(t)$. To analyze the BP algorithm with the "genie" initialization in Appendix B, let $H^{\rm genie}_{i\rightarrow j}(t)$ and $H^{\rm genie}_{i\leftarrow j}(t)$ be respectively the sigma algebras generated by the entire vector $\mathbf{y}$ and the variable x_r *not* in the computation subgraphs $G_{i\rightarrow j}(t)$ and $G_{i\leftarrow j}(t)$. The following results are standard for BP and can be proven using arguments as in [41].

Lemma 4: Consider the sigma algebras $H_{i\leftarrow j}(t)$ and $H_{i\rightarrow j}(t)$ defined above.

(a) In the standard BP algorithm in Section III, the distributions $p^x_{i\leftarrow j}(t, x_j)$ are $H_{i\leftarrow j}(t)$ measurable, and the distribution $p^z_{i\rightarrow j}(t, z_i)$ and likelihood function $p^u_{i\rightarrow j}(t, u_j)$ are $H_{i\rightarrow j}(t)$ measurable.

(b) In the relaxed BP algorithm, $\widehat{x}_{i\leftarrow j}(t)$ and $\widehat{q}_{i\leftarrow j}(t)$ are $H_{i\leftarrow j}(t)$ measurable and $\widehat{z}_{i\rightarrow j}(t)$ and $\widehat{u}_{i\rightarrow j}(t)$ are $H_{i\rightarrow j}(t)$ measurable.

Similarly, under the oracle initialization described in Appendix B, the above statements hold with $H_{i\leftarrow j}(t)$ and $H_{i\rightarrow j}(t)$ replaced by $H^{\rm genie}_{i\leftarrow j}(t)$ and $H^{\rm genie}_{i\rightarrow j}(t)$.

Lemma 5: Consider the standard BP algorithm in Section III and the computation subgraphs defined above.

(a) If $G_{i\leftarrow j}(t)$ is a tree, then $p^x_{i\leftarrow j}(t, x_j)$ is the conditional distribution of x_j given $H_{i\leftarrow j}(t)$.

(b) If $G_{i\rightarrow j}(t)$ is a tree, then $p^z_{i\rightarrow j}(t, z_i)$ is the conditional distribution of $z_{i\rightarrow j}$ given $H_{i\rightarrow j}(t)$.

Similarly, under the genie initialization, the above statements hold with $H_{i\leftarrow j}(t)$ and $H_{i\rightarrow j}(t)$ replaced by $H^{\rm genie}_{i\leftarrow j}(t)$ and $H^{\rm genie}_{i\rightarrow j}(t)$.

Lemma 6: Consider the relaxed BP algorithm in Section IV under the assumptions in Section V.

(a) If $G_{i\leftarrow j}(t)$ is a tree, then all the terms $(\widehat{z}_{\ell\rightarrow j}(t), \mu^z_{i\rightarrow j}(t))$ are independent for different values $\ell \in N_{\rm in}(j) \neq i$.

(b) If $G_{i\rightarrow j}(t)$ is a tree, then the random vectors $\theta^x_{i\leftarrow r}(t)$ are independent for different values $r \in N_{\rm out}(i) \neq j$.

Appendix E
Proof of Theorem 1(a)

We prove this by induction. It is clear that (42a) holds for $t = 1$. In part B below we will show that if (42a) holds for some t, then so does (42c) and (42d). Then, in part C, we will show that if (42c) holds for some t, then (42a) holds for $t+1$. This will complete the induction argument. Part A provides a preliminary calculation that we need in part C.

A. Derivatives of the Score Function

The following lemma characterizes the derivatives $D_r(y, \widehat{z}, \mu)$ in (16). The result can also be found in [14], but we sketch the proof here for completeness. We will use this result below in the analysis of the output node update.

Lemma 7: Fix $\widehat{z}$ and μ and consider random variables y and z generated by $z \sim \mathcal{N}(\widehat{z} + u, \mu)$ and $y \sim p_{Y|Z}(y|z)$ for some $u \in \mathbb{R}$. Consider the derivative of the score function in (16). Then,

$$\begin{aligned} \mathbf{E}\left[D_1(y, \widehat{z}, \mu)|u\right] &= -u\mathbf{E}\left[D_2(y, \widehat{z}, \mu)|u = 0\right] \\ &\quad + O(u^2), \qquad (51a) \\ \mathbf{var}\left[D_1(y, \widehat{z}, \mu)|u\right] &= \mathbf{E}\left[D_2(y, \widehat{z}, \mu)|u = 0\right] \\ &\quad + O(u^2). \qquad (51b) \end{aligned}$$

Proof: To simplify the notation, we will drop the dependence on $\widehat{z}$ and μ. So, we will write, for example, $D_1(y)$ for $D_1(y, \widehat{z}, \mu)$. To prove (51a), we first note that

$$\begin{aligned} \mathbf{E}\left[D_1(y)|u\right] &= \int p_{Y|U}(y|u) D_1(y)\, dy \\ &= \int p_{Y|U}(y|0) D_1(y)\, dy \\ &\quad + u \int \left. \frac{\partial}{\partial u} p_{Y|U}(y|u) \right|_{u=0} D_1(y)\, dy + O(u^2). \qquad (52) \end{aligned}$$

Now, for the first term in (52), note that using the definition of $D_1(y)$ in (16), we have

$$\begin{aligned} &\int p_{Y|U}(y|0) D_1(y)\, dy \\ &= -\int p_{Y|U}(y|0) \left. \frac{\partial}{\partial u} \log p_{Y|U}(y|u) \right|_{u=0} dy \\ &= -\int \left. \frac{\partial}{\partial u} p_{Y|U}(y|u) \right|_{u=0} dy \\ &= -\frac{\partial}{\partial u} \int p_{Y|U}(y|u)\, dy \Big|_{u=0} \\ &= -\frac{\partial}{\partial u}(1) = 0. \qquad (53) \end{aligned}$$

Similarly, the second term in (52) can be simplified by evaluating the second-order derivative in the definition of $D_2(y)$ to obtain

$$\begin{aligned} &\mathbf{E}\left[D_2(y)|u = 0\right] \\ &= \int \frac{1}{p_{Y|U}(y|0)} \left(\frac{\partial}{\partial u}\, p_{Y|U}(y|u)\big|_{u=0} \right)^2 dy \\ &\quad - \int \frac{\partial^2}{\partial u^2}\, p_{Y|U}(y|u)\big|_{u=0}\, dy. \qquad (54) \end{aligned}$$

Now,

$$\begin{aligned} &\int \frac{\partial^2}{\partial u^2}\, p_{Y|U}(y|u)\big|_{u=0}\, dy \\ &= \frac{\partial^2}{\partial u^2} \int p_{Y|U}(y|u)\big|_{u=0}\, dy = \frac{\partial^2}{\partial u^2}(1) = 0. \qquad (55) \end{aligned}$$

Also,

$$\begin{aligned}
&\int \frac{1}{p_{Y|U}(y|0)} \left(\frac{\partial}{\partial u} \, p_{Y|U}(y|u)\big|_{u=0} \right)^2 dy \\
&\quad = \int \frac{\partial}{\partial u} \, p_{Y|U}(y|u)\big|_{u=0} \frac{\partial}{\partial u} \log p_{Y|U}(y|u)\big|_{u=0} \, dy \\
&\quad = -\int \frac{\partial}{\partial u} \, p_{Y|U}(y|u)\big|_{u=0} D_1(y) \, dy. \qquad (56)
\end{aligned}$$

Substituting (55) and (56) into (54), we obtain that

$$\mathbf{E}\left[D_2(y)|u=0\right] = -\int \frac{\partial}{\partial u} \, p_{Y|U}(y|u)\big|_{u=0} D_1(y) \, dy. \quad (57)$$

Then (51a) follows by substituting (53) and (57) into (52). Equation (51b) is proved by similar manipulations. ■

B. Analysis of the Output Node Update

Let $t \geq 1$ and suppose that (42a) holds for some t. We will show that this induction hypothesis implies (42c) and (42d). We will just prove this implication for $t > 1$. The proof for $t = 1$ is similar.

Under the induction hypothesis (42a), we first consider the convergence of the terms $\mu^z_{i\rightarrow j}(t)$. From the factorization (28) we have that

$$\Phi_{ij} = \frac{1}{\sqrt{d}} a_{ij} \sqrt{s_j}, \quad \forall j \in N_{\rm out}(i). \qquad (58)$$

Using (18) and (58), we have that

$$\begin{aligned}
\mu^z_{i\rightarrow j}(t) &= \sum_{r \in N_{\rm out}(i) \neq j} |\Phi_{ir}|^2 \mu^x_{i\leftarrow r} \\
&= \frac{1}{d} \sum_{r \in N_{\rm out}(i) \neq j} |a_{ir}|^2 s_r \mu^x_{i\leftarrow r}(t). \qquad (59)
\end{aligned}$$

By Lemma 6(b) and the assumption that $G_{i\rightarrow j}(t)$ is a tree, the terms in the summation in (59) are independent. Also, the induction hypothesis (42a) shows that their asymptotic distribution is given by

$$\lim_{d\rightarrow\infty} \lim_{n\rightarrow\infty} s_r \mu^x_{i\leftarrow r}(t) = s\mathcal{E}_{\rm in}(q, \mu^q(t-1)/s),$$

where the convergence is in distribution and the random variables s and q are the terms in $\theta^x(t)$ in (39). For the regular relaxed BP algorithm $\mu^q(t-1) = \mu^q_{\rm hi}(t-1)$ and, for the genie algorithm, $\mu^q(t-1) = \mu^q_{\rm lo}(t-1)$. In either case, the expectation of this limiting random variable is

$$\mathbf{E}\left[s\mathcal{E}_{\rm in}(q, \mu^q(t-1)/s)\right] = \overline{\mathcal{E}}_{\rm in}(\mu^q(t-1)),$$

where $\overline{\mathcal{E}}_{\rm in}(\mu^q(t-1))$ is defined in (32). Using (30c), we can apply the Modified Law of Large Numbers (Lemma 2), to the sum in (59) to obtain

$$\lim_{d\rightarrow\infty} \lim_{n\rightarrow\infty} \mu^z_{i\leftarrow j}(t) = \beta\overline{\mathcal{E}}_{\rm in}(\mu^q(t-1)) = \mu^z(t), \qquad (60)$$

where the convergence is in distribution and the last step follows from the definition of $\mu^z(t)$ in (37).

We next consider the convergence of the variables $\widehat{z}_{i\rightarrow j}(t)$. If we define $z_{i\rightarrow j}$ as in (9), then (18) and (58) show that

$$\begin{aligned}
z_{i\rightarrow j} - \widehat{z}_{i\rightarrow j}(t) &= \sum_{r \in N_{\rm out}(i) \neq j} \Phi_{ir}(x_r - \widehat{x}_{i\leftarrow r}) \\
&= \frac{1}{\sqrt{d}} \sum_{r \in N_{\rm out}(i) \neq j} a_{ir}\sqrt{s_r}(x_r - \widehat{x}_{i\leftarrow r}). \qquad (61)
\end{aligned}$$

By Lemma 6(b) the terms in the summation (61) are independent. Also assuming (42a) holds, the terms in the summation converge as

$$\lim_{d\rightarrow\infty} \lim_{n\rightarrow\infty} \sqrt{s_r}(x_r - \widehat{x}_{i\leftarrow r}) = \sqrt{s}(x - F_{\rm in}(q, \mu^q(t-1)/s)),$$

where the convergence is in distribution and the random variables x, s and q are the terms in $\theta^x(t)$ in (39). The variances of the terms converge as

$$\begin{aligned}
&\lim_{d\rightarrow\infty} \lim_{n\rightarrow\infty} \mathbf{E}\left[s_r|x_r - \widehat{x}_{i\leftarrow r}|^2\right] \\
&\quad = \mathbf{E}\left[s|x - F_{\rm in}(q, \mu^q(t-1)/s)|^2\right] = \overline{\mathcal{E}}_{\rm in}(\mu^q(t-1)).
\end{aligned}$$

Using (30a), (30b) and (60), we can apply the Modified Central Limit Theorem (Lemma 3) to (61) to obtain

$$\lim_{d\rightarrow\infty} \lim_{n\rightarrow\infty} z_{i\rightarrow j} - \widehat{z}_{i\rightarrow j}(t) = \mathcal{N}(0, \mu^z(t)), \qquad (62)$$

where the convergence is in distribution.

Similarly one can show that

$$\lim_{d\rightarrow\infty} \lim_{n\rightarrow\infty} z_{i\rightarrow j} = \mathcal{N}(0, \mu^z_{\rm init}) \qquad (63a)$$

and

$$\lim_{d\rightarrow\infty} \lim_{n\rightarrow\infty} (z_{i\rightarrow j} - \widehat{z}_{i\rightarrow j}(t))\widehat{z}_{i\rightarrow j}(t) = 0, \qquad (63b)$$

where $\mu^z_{\rm init}$ is defined in (33). Equations (62) and (63) imply that

$$\lim_{d\rightarrow\infty} \lim_{n\rightarrow\infty} (z_{i\rightarrow j}, \widehat{z}_{i\rightarrow j}(t)) = (z, \widehat{z}), \qquad (64)$$

where z and $\widehat{z}$ are the Gaussian random variables in (35) with $\mu = \mu_z(t)$. Combining (60) and (64) proves (42c).

This argument shows that if (42a) is true for some t, then so is (42c). A similar argument shows that (42a) also implies (42d), except we replace the summations over the sets $r \in N_{\rm out}(i) \neq j$ with $r \in N_{\rm out}(i)$.

C. Analysis of the Input Node Update

For the next step in the induction proof, we want to prove that if (42c) holds for some t, then (42a) holds for $t+1$.

Throughout this section, fix the input index j and variables s_j and x_j. For each output index $\ell \in N_{\rm in}(j)$ and $u \in \mathbb{R}$, define the Markov chain

$$\widehat{z}^G_\ell \rightarrow z^G_\ell(u) \rightarrow y^G_\ell(u),$$

where the random variables are distributed as

$$\begin{aligned}
\widehat{z}^G_\ell &\sim \mathcal{N}(0, \mu^z_{\rm init} - \mu^z(t)) \\
z^G_\ell(u) &\sim \mathcal{N}(\widehat{z}^G_\ell + u, \mu^z(t)) \\
y^G_\ell(u) &\sim p_{Y|Z}(y|z^G_\ell(u)).
\end{aligned}$$

Suppose the Markov chains are independent over different values of ℓ. We use the superscript "G" here to indicate that the random variables are Gaussian approximations to actual random variables in the problems. To be specific, first observe that

$$\Phi_{\ell j} = \frac{1}{\sqrt{d}} a_{\ell j}\sqrt{s_j}, \quad \forall \ell \in N_{\rm out}(j). \tag{66}$$

Combining (66) with the definition of $z_{i\to j}$ in (9) and the fact that $\mathbf{z} = \Phi\mathbf{x}$, we have

$$z_\ell = \sum_{r\in N_{\rm out}(\ell)} \Phi_{\ell j}x_j = u_\ell + z_{\ell\to j}, \tag{67}$$

where

$$u_\ell = \Phi_{\ell j}x_j = \frac{1}{\sqrt{d}}a_{\ell j}\sqrt{s_j}x_j. \tag{68}$$

The induction hypothesis (42c) and Lemma 6(a) then show that

$$\begin{aligned} &\lim_{d\to\infty}\lim_{n\to\infty} (\widehat{z}_{\ell\to j}(t), z_\ell, y_\ell, \mu^z_{\ell\to j}(t)) \\ &\quad = \ (\widehat{z}^G_\ell, z^G_\ell(u_\ell), y^G_\ell(u_\ell), \mu^z(t)), \end{aligned} \tag{69}$$

where the convergence is in distribution.

With these definitions, we first consider the convergence of $\mu^q_{i\leftarrow j}(t)$. Using (19b), (20b) and (66), we have that

$$\begin{aligned} \frac{1}{\mu^q_{i\leftarrow j}(t)} &= \sum_{\ell\in N_{\rm in}(j)\neq i} \frac{|\Phi_{\ell j}|^2}{\mu^u_{\ell\to j}(t)} \\ &= \frac{1}{d}\sum_{\ell\in N_{\rm in}(j)\neq i} |a_{\ell j}|^2 s_j D_2(y_\ell, \widehat{z}_{\ell\to j}(t), \mu^z_{\ell\to j}(t)). \end{aligned} \tag{70}$$

By Lemma 6(a), given x_j and s_j, all the terms in (70) are independent.

Also, using (69), we have the limit

$$\begin{aligned} &\lim_{d\to\infty}\lim_{n\to\infty} D_2(y_\ell, \widehat{z}_{\ell\to j}(t), \mu^z_{\ell\to j}(t)) \\ &\quad \overset{(a)}{=} \ D_2(y^G_\ell(u_\ell), \widehat{z}^G_\ell(u_\ell)\mu^z(t)) \\ &\quad \overset{(b)}{=} \ D_2(y^G_\ell(0), \widehat{z}^G_\ell(0), \mu^z(t)) + O(|u_\ell|^3) \\ &\quad \overset{(c)}{=} \ D_2(y^G_\ell(0), \widehat{z}^G_\ell(0), \mu^z(t)) \end{aligned} \tag{71}$$

where the convergence in (a) is in distribution; (b) follows from the assumption that $D_3(\cdot)$ is uniformly bounded and (c) follows from the fact that (68) shows that $|u_\ell|^3 = O(d^{-3/2}) \to 0$. We can apply the Modified Law of Large Numbers (Lemma 2) to the sum (70) to obtain the limit

$$\begin{aligned} &\lim_{d\to\infty}\lim_{n\to\infty} \frac{1}{\mu^q_{i\leftarrow j}(t)} \\ &\overset{(a)}{=} \ \lim_{d\to\infty}\lim_{n\to\infty} \frac{1}{d}\sum_{\ell\in N_{\rm in}(j)\neq i} |a_{\ell j}|^2 s_j \\ &\qquad\qquad \times D_2(y^G_\ell(0), \widehat{z}^G_\ell(0), \mu^z(t)) \\ &\overset{(b)}{=} \ s_j\mathbf{E}\left[D_2(y^G_\ell(0), \widehat{z}^G_\ell(0), \mu^z(t))\right] \\ &\overset{(c)}{=} \ s_j\overline{\mathcal{E}}_{\rm out}(\mu^z(t)) \overset{(d)}{=} \frac{s_j}{\mu^q(t)}, \end{aligned} \tag{72}$$

where the limit in (a) is in distribution and follows from (70) and (71); (b) follows from (30c) and the Modified Law of Large Numbers; (c) follows from the definition of $\overline{\mathcal{E}}_{\rm out}(\cdot)$ in (34) and the fact that the expectation over (y,z) in (34) is identical to $(y^G_\ell(0), z^G_\ell(0))$; and (d) follows from the SE equation (37a).

We next turn to the distribution of $\widehat{q}_{i\leftarrow j}(t)$. Using (19), the update (20a) can be simplified to

$$\begin{aligned} \widehat{q}_{i\leftarrow j}(t) &= \mu^q_{i\leftarrow j}(t)\sum_{\ell\neq i}\frac{\Phi^*_{\ell j}\widehat{u}_{\ell\to j}(t)}{\mu^u_{\ell\to j}(t)} \\ &= -\mu^q_{i\leftarrow j}(t)\sum_{\ell\neq i}\Phi^*_{\ell j}D_1(y_\ell, \widehat{z}_{\ell\to j}(t), \mu^z_{\ell\to j}(t)). \end{aligned} \tag{73}$$

So, using (69) and (72),

$$\begin{aligned} &\lim_{d\to\infty}\lim_{n\to\infty}\widehat{q}_{i\leftarrow j}(t) \\ &\quad = -\frac{\mu^q(t)}{s_j}\lim_{d,n\to\infty}\sum_{\ell\neq i}\Phi^*_{\ell j}D_1(y^G_\ell(u_\ell), \widehat{z}^G_\ell(u_\ell), \mu^z(t)), \end{aligned} \tag{74}$$

where here and below we use the shorthand $\lim_{d,n}$ for $\lim_d\lim_n$. Now define

$$\begin{aligned} e_\ell &= D_1(y^G_\ell(u_\ell), \widehat{z}^G_\ell(u_\ell), \mu^z(t)) \\ &\qquad + \Phi_{\ell j}x_j\mathbf{E}\left[D_2(y^G_\ell(0), \widehat{z}^G_\ell(0), \mu^z(t))\right], \end{aligned} \tag{75}$$

so we can rewrite (74) as

$$\begin{aligned} &\lim_{d\to\infty}\lim_{n\to\infty}\widehat{q}_{i\leftarrow j}(t) \\ &\quad = -\lim_{d,n\to\infty}\sum_{\ell\neq i}\frac{1}{s_j}\Phi^*_{\ell j}e_\ell \\ &\qquad - \lim_{d,n\to\infty}\sum_{\ell\neq i}\frac{\mu^q(t)|\Phi_{\ell j}|^2x_j}{s_j}\mathbf{E}\left[D_2(y^G_\ell(0), \widehat{z}^G_\ell(0), \mu^z(t))\right] \\ &\quad = -\lim_{d,n\to\infty}\frac{\mu^q(t)}{\sqrt{d s_j}}\sum_{\ell\neq i}a^*_{\ell j}e_\ell + x_j, \end{aligned} \tag{76}$$

where the last step follows from (74) and (58).

Now, applying Lemma 7 to e_ℓ in (75),

$$\begin{aligned} \mathbf{E}\left(e_\ell|u_\ell\right) &= O(|u_\ell|^2) \\ \mathbf{var}\left(e_\ell|u_\ell\right) &= \mathbf{E}\left[D_2(y^G_\ell(0), \widehat{z}^G_\ell(0), \mu^z(t))\right] + O(|u_\ell|^2) \\ &= \frac{1}{\mu^q(t)} + O(|u_\ell|^2). \end{aligned}$$

From the definition of u_ℓ in (68), the $O(|u_\ell|^2)$ terms are $O(1/d)$ and thus can be ignored. Applying the Modified Central Limit Theorem (Lemma 3) to the sum in (76) along with (30c) we obtain

$$\begin{aligned} \lim_{d\to\infty}\lim_{n\to\infty}\widehat{q}_{i\leftarrow j}(t) &= x_j + \frac{(\mu^q(t))^2}{s_j}\mathcal{N}\left(0, \frac{1}{\mu^q(t)}\right) \\ &= \mathcal{N}\left(x_j, \frac{\mu^q(t)}{s_j}\right). \end{aligned} \tag{77}$$

Also since $x_j \sim p_X(x_j)$ and $s_j \sim p_S(s_j)$, (72) and (77) together show that for any iteration t,

$$\begin{aligned} &\lim_{d\to\infty}\lim_{n\to\infty}(x_j, s_j, \widehat{q}_{i\leftarrow j}(t), \mu^q_{i\leftarrow j}(t)) \\ &\quad = \left(x, s, \mathcal{N}\left(x, \frac{1}{s}\mu_q(t)\right), \frac{1}{s}\mu_q(t)\right), \end{aligned} \tag{78}$$

where the convergence is in distribution, $x \sim p_X(x)$ and $s \sim p_S(s)$. Applying (21) to (78) shows (42a). Therefore, we have shown that if (42c) holds for t, then (42a) holds for $t+1$. One can also show that if (42c) holds for t, then (42b) holds for $t+1$ using similar arguments except we replace the summations over $\ell \in N_{\rm in}(j) \neq i$ with $\ell \in N_{\rm in}(j)$.

Appendix F
Proof of Theorem 1(b) and (c)

Parts (b) and (c) of Theorem 1 can be proven along the lines of Guo and Wang's analysis in [13] and [14] using the concept of an *asymptotically sufficient statistic* along with a standard *sandwiching* argument. Specifically, using their analysis, we will show that the regular and "genie" versions of the relaxed BP algorithm provide sufficient statistics for certain conditional distribution of the vectors $\mathbf{x}$ and $\mathbf{z}$. The regular BP algorithm provides a sufficient statistic relative to the sigma algebras $H_{i\leftarrow j}(t)$ and $H_{i\rightarrow j}(t)$, and the "genie" relaxed BP algorithm in Appendix B provides a sufficient statistic relative to $H^{\rm genie}_{i\leftarrow j}(t)$ and $H^{\rm genie}_{i\rightarrow j}(t)$. Moreover, the MSE relative to the sigma algebras is described by the state evolution equations starting from the "high" initial conditions for the regular algorithm and "low" initial condition for the genie algorithm. Since the sigma algebra generated by the actual observation vector $\mathbf{y}$ lies somewhere between these sigma algebras, H and $H^{\rm genie}$, the MSE of the optimal estimator is "sandwiched" between the two solutions to the state evolution equations.

Since the arguments in this section follow very closely with Guo and Wang's analysis in [13], [14], we will just sketch the proof. Similar sandwiching arguments can be found in the early analysis of LDPC codes in [39].

We begin with the following definition.

Definition 1: Suppose that (x_n, q_n, H_n) is a sequence where, for every n, x_n and q_n are random variables and H_n is a sigma algebra. We will say that q_n is an *asymptotically sufficient statistic* for x_n given H_n with limiting distribution $(x_n, q_n) \rightarrow (x, q)$ if:

(a) q_n is H_n-measurable;
(b) $(x_n, q_n) \rightarrow (x, q)$ in distribution;
(c) For any bounded continuous function $f(x)$,

$$\lim_{n\rightarrow\infty} \left(\mathbf{E}(f(x_n)|H_n) - \mathbf{E}(f(x)|q = q_n)\right) = 0$$

almost surely.

The definition is a natural generalization of the concept of a sufficient statistic. Specifically, it says that the conditional estimate $\mathbf{E}(f(x_n)|H_n)$ can be replaced by $\mathbf{E}(f(x)|q = q_n)$ with asymptotically vanishing error. That is, it is sufficient to use just q_n instead of the entire sigma algebra H_n and use just the limiting distribution (x, q) as opposed to the termwise distributions (x_n, q_n).

Following along the lines of Guo and Wang [13], [14], we now prove the following.

Theorem 3: For the relaxed BP algorithm:

(a) If $G_{i\rightarrow j}(t)$ is a tree, then $\widehat{z}_{i\rightarrow j}(t)$ is an asymptotically sufficient statistic for $z_{i\rightarrow j}$ given $H_{i\rightarrow j}(t)$ with the asymptotic distribution (42c).
(b) If $G_{i\leftarrow j}(t)$ is a tree, then $(\widehat{q}_{i\leftarrow j}(t), s_j)$ is an asymptotically sufficient statistic for x_j given $H_{i\leftarrow j}(t)$ with the asymptotic distribution (78).

The result also holds for the "genie" algorithm in Appendix B with $H_{i\rightarrow j}(t)$ and $H_{i\leftarrow j}(t)$ replaced by $H^{\rm genie}_{i\rightarrow j}(t)$ and $H^{\rm genie}_{i\leftarrow j}(t)$.

Similar to the proof of Theorem 1(a), we prove Theorem 3 by induction. For the initial step in the induction, note that, for the regular (non-Genie) algorithm, $H_{i\rightarrow j}(t)$ is empty and $\widehat{z}_{i\rightarrow j}(1)$ is the prior on $z_{i\rightarrow j}$. For the genie algorithm, $H^{\rm genie}_{i\rightarrow j}(t)$ contains the entire vector $\mathbf{x}$ and $\widehat{z}_{i\rightarrow j}(1) = z_{i\rightarrow j}$. Therefore, part (a) of Theorem 3 holds for $t = 1$. In part A below, we will show that if (a) holds for some t, (b) holds for $t + 1$. In part B, we will show the reverse implication that if (b) holds for some t so does (a). In part C, we apply Theorem 3 to prove Theorem 1(b) and (c).

A. Analysis of the Input Node Update

Suppose that part (a) of Theorem 3 holds for some $t \geq 1$. We will prove part (b) holds for $t + 1$. The asymptotic limit (78) has already been proven. We only need to show that $(\widehat{q}_{i\leftarrow j}(t), s_j)$ is asymptotically sufficient to describe the conditional distribution of x_j given $H_{i\leftarrow j}(t + 1)$.

To this end, suppose that $G_{i\leftarrow j}(t + 1)$ is a tree. By the construction of the computation subgraphs, $G_{\ell\rightarrow j}(t)$ must be a tree for every $\ell \in N_{\rm in}(j)$, $\ell \neq j$. Now define, for any $r \geq 1$, the "actual" derivatives of the likelihood

$$D^{\ell\rightarrow j}_{r,act}(t, y_\ell) = -\frac{\partial^r}{\partial u^r} \log p^u_{\ell\rightarrow j}(t, u)\big|_{u=0}, \tag{79}$$

where $p^u_{\ell\rightarrow j}(t, u)$ is defined in (11). Since $G_{\ell\rightarrow j}(t)$ is a tree, Lemma 5 shows that $p^z_{\ell\rightarrow j}(t, z_{\ell\rightarrow j})$ in (11) is the conditional distribution $z_{\ell\rightarrow j}$ given $H_{\ell\rightarrow j}(t)$. Bringing the derivatives in (79) inside the expectation in (11) we can rewrite (79) as

$$\begin{aligned} &D^{\ell\rightarrow j}_{r,act}(t, y_\ell) \\ &\quad = \; -\mathbf{E}\left[\frac{\partial^r}{\partial u^r} \log p_{Y|Z}(y_i|u + z_{\ell\rightarrow j})\big|_{u=0} \;\mid H_{\ell\rightarrow j}(t)\right], \end{aligned} \tag{80}$$

where the expectation is over the conditional distribution of $z_{\ell\rightarrow j}$ given $H_{\ell\rightarrow j}(t)$. Also, using (15) and (16), we can write

$$\begin{aligned} &D_r(y, \widehat{z}, \mu) \\ &\quad = \; -\mathbf{E}\left[\frac{\partial^r}{\partial u^r} \log p_{Y|Z}(y_i|u + z)\big|_{u=0} \;\mid \widehat{z}, \mu\right], \end{aligned} \tag{81}$$

where the expectation is over $z \sim \mathcal{N}(\widehat{z}, \mu)$.

Now, the induction hypothesis, Theorem 3(a), states that $\widehat{z}_{\ell\rightarrow j}(t)$ is asymptotically sufficient for $z_{\ell\rightarrow j}$ given $H_{\ell\rightarrow j}(t)$ with the asymptotic distribution

$$\lim_{d\rightarrow\infty} \lim_{n\rightarrow\infty} (z_{\ell\rightarrow j}, \widehat{z}_{\ell\rightarrow j}(t)) = \mathcal{N}(0, P_z(\mu^z(t)),$$

where $\mu^z(t) = \mu^z_{\rm hi}(t)$ for the regular algorithm and $\mu^z(t) = \mu^z_{\rm lo}(t)$ for the "genie algorithm". Applying this property to (80) to (81), we obtain that

$$\lim_{n\rightarrow\infty} \lim_{d\rightarrow\infty} D^{\ell\rightarrow j}_{r,act}(t, y_\ell) - D_r(y_\ell, \widehat{z}_{\ell\rightarrow j}(t), \mu^z(t)) = 0, \tag{82}$$

almost surely.

We can now rewrite $p^x_{i\leftarrow j}(t+1, x_j)$ in (13) as

$$\begin{aligned}
&\lim_{d\to\infty}\lim_{n\to\infty} -\log p^x_{i\leftarrow j}(t+1,x_j) + \log p_X(x_j) + \text{const} \\
&\overset{(a)}{=} \lim_{d\to\infty}\lim_{n\to\infty} \sum_{\ell\in N_{\rm in}(j)\neq i} \log p^u_{\ell\to j}(t, \Phi_{\ell j}x_j) \\
&\overset{(b)}{=} \lim_{d\to\infty}\lim_{n\to\infty} \sum_{\ell\in N_{\rm in}(j)\neq i} D^{\ell\to j}_{r,act}(t, y_\ell)\Phi_{\ell j}x_j \\
&\quad + \frac{1}{2}D^{\ell\to j}_{r,act}(t,y_\ell)|\Phi_{\ell j}x_j|^2 + O(|\Phi_{\ell j}x_j|^3) \\
&\overset{(c)}{=} \lim_{d,n\to\infty} \sum_{\ell\in N_{\rm in}(j)\neq i} D_r(y_\ell, \widehat{z}_{\ell\to j}(t), \mu^z(t))\Phi_{\ell j}x_j \\
&\quad + \frac{1}{2}D_r(y_\ell, \widehat{z}_{\ell\to j}(t), \mu^z(t))|\Phi_{\ell j}x_j|^2 \\
&\overset{(d)}{=} \lim_{d,n\to\infty} \frac{1}{2\mu^q_{i\leftarrow j}(t)}|x_j - \widehat{q}_{i\leftarrow j}(t)|^2 \\
&\overset{(e)}{=} \lim_{d,n\to\infty} \frac{s_j}{2\mu^q(t)}|x_j - \widehat{q}_{i\leftarrow j}(t)|^2
\end{aligned}$$

where the constant is independent of x_j; (a) follows from (13); (b) is the Taylor's series expansion of $\log p^u_{\ell\to j}(t, \Phi_{\ell j}x_j)$; (c) follows from (82) and (30d); (d) follows from (19) and (20) and (e) follows from (72). Here, with some abuse of notation, we have written $\lim A = \lim B$ in place of $\lim(A-B) = 0$. Using this same convention, the above equations show that

$$\begin{aligned}
&\lim_{d\to\infty}\lim_{n\to\infty} p^x_{i\leftarrow j}(t+1,x_j) \\
&\quad = \lim_{d\to\infty}\lim_{n\to\infty} \text{const} \\
&\qquad \times p_X(x_j)\exp\left[\frac{1}{2\mu^q_{i\leftarrow j}(t)}|x_j - \widehat{q}_{i\leftarrow j}(t)|^2\right]. \quad (83)
\end{aligned}$$

From Lemma 5, the left hand side of (83) precisely the conditional distribution of x_j given $H_{i\leftarrow j}(t{+}1)$ (or $H^{\rm genie}_{i\leftarrow j}(t)$). Therefore, (83) shows that this conditional distribution is asymptotically only a function of $\widehat{q}_{i\leftarrow j}(t)$ and s_j, and therefore $(\widehat{q}_{i\leftarrow j}(t), s_j)$ is asymptotically sufficient for x_j given $H_{i\leftarrow j}(t+1)$.

B. Analysis of the Output Update

Continuing the induction argument, we next show that if the part (b) of Theorem 3 holds for some t, then so does part (a). We have already proven the asymptotic distribution (42c). So, we just need to show that the conditional distribution of $z_{i\to j}$ given $H_{i\to j}(t)$ asymptotically depends only on $\widehat{z}_{i\to j}(t)$.

Now, from Lemma 5, the conditional distribution of $z_{i\to j}$ given $H_{i\to j}(t)$ is given by $p^z_{i\to j}(t, z_{i\to j})$ from the BP algorithm. But this distribution is described by the summation (9). The analysis in Appendix E-B shows that this summation has an asymptotic Gaussian distribution $\mathcal{N}(\widehat{z}_{i\to j}(t), \mu^z(t))$. So, $\widehat{z}_{i\to j}(t)$ is asymptotically sufficient to describe the distribution.

This completes the induction argument and proves Theorem 3.

C. MSE Relationships

Using Theorem 3, we can now prove parts (b) and (c) of Theorem 1. First observe that Theorem 3 along with the definition of an asymptotically sufficient statistic shows that

$$\begin{aligned}
&\lim_{d\to\infty}\lim_{n\to\infty} \mathbf{E}(x_j|H_{i\leftarrow j}(t)) - \widehat{x}_{i\leftarrow j}(t+1) \\
&\overset{(a)}{=} \lim_{d\to\infty}\lim_{n\to\infty} \mathbf{E}(x|q = \widehat{q}_{i\leftarrow j}(t), s = s_j) - \widehat{x}_{i\leftarrow j}(t+1) \\
&\overset{(b)}{=} \lim_{d\to\infty}\lim_{n\to\infty} F_{\rm in}(\widehat{q}_{i\leftarrow j}(t), \mu^q(t)/s_j) - \widehat{x}_{i\leftarrow j}(t+1) \\
&\overset{(c)}{=} 0 \qquad (84)
\end{aligned}$$

where in (a) the expectation is with respect to (x,q,s) distributed as (78); (b) follows from the definition of $F_{\rm in}(\cdot)$ in Section IV-A; and (c) follows from (21) and (78). The limit (84) shows that the conditional variance is given by

$$\begin{aligned}
&\lim_{d\to\infty}\lim_{n\to\infty} \mathbf{E}\left(\mathbf{var}(x_j|H_{i\leftarrow j}(t))|s_j = s\right) \\
&\quad = \lim_{d\to\infty}\lim_{n\to\infty} \mathbf{E}\left(|x_j - \mathbf{E}(x_j|H_{i\leftarrow j}(t))|^2|s_j = s\right) \\
&\quad \overset{(a)}{=} \lim_{d\to\infty}\lim_{n\to\infty} \mathbf{E}\left(|x_j - \widehat{x}_{i\leftarrow j}(t)|^2|s_j = s\right) \\
&\quad \overset{(b)}{=} \mathbf{E}\left(|x - F_{\rm in}(q, \mu^q(t)/s)|^2|s\right) \\
&\quad \overset{(c)}{=} \mathbf{E}\left(\mathcal{E}_{\rm in}(q, \mu^q(t)/s)|s\right) \\
&\quad \overset{(d)}{=} \overline{\mathcal{E}}_{\rm in}(\mu^q(t), s) \overset{(e)}{=} \mu^x(t+1, s), \qquad (85)
\end{aligned}$$

where (a) is due to the limit (84); (b) is due to the limit (78); (c) is the definition of $\mathcal{E}_{\rm in}(q,\mu)$; (d) follows from (32); and (e) is from (37b). The limit (85) holds for the regular algorithm with $\mu^x(t{+}1, s) = \mu^x_{\rm hi}(t{+}1, s)$ and for genie algorithm with $\mu^x(t+1, s) = \mu^x_{\rm lo}(t+1, s)$. With the regular (non-genie) algorithm, the limit (85) shows (43a). Also, for the genie algorithm, the sigma algebra $H^{\rm genie}_{i\leftarrow j}(t)$ is contains the sigma generated by just $\mathbf{y}$. Therefore,

$$\mathbf{var}(x_j|\mathbf{y}) \geq \mathbf{var}(x_j|H^{\rm genie}_{i\leftarrow j}(t)).$$

Combining this inequality with (85) shows (44a).

A similar argument can be used to show (43b) and (44b). We have thus shown part (b) and (c) of Theorem 1.

Appendix G
Proof of Theorem 2

The proof is based on a *degradation* argument, which is used commonly for convergence proofs of BP algorithms [40]. Suppose that $X \to Y \to Z$ is a Markov chain. Then, we say that Z is *degraded* with respect to Y, since estimates of X from Z are strictly worse than those from Y. The following lemma states a standard property of degraded random variables.

Lemma 8: Suppose that $X \to Y \to Z$ is a Markov chain. Then,

(a) The conditional variance of X satisfies

$$\mathbf{var}(X|Y) \leq \mathbf{var}(X|Z).$$

(b) Suppose the likelihood function of X given Y and the likelihood of X given Z both have continuous third derivatives. Then, for any x,

$$F(Z|X = x) \leq F(Y|X = x)$$

where, for any random variables X and W, $F(W|X = x)$ is the Fisher information

$$F(W|X = x) = -\mathbf{E}\left[\frac{\partial^2}{\partial x^2} \log p_{W|X}(w|x) \middle| X = x \right] \tag{86}$$

Proof: See, for example, [42]. ∎

We will combine this lemma with the following simple iteration result to prove the theorem.

Lemma 9: Suppose $G(\mu)$ is a monotonically increasing, continuous function with $0 \leq G(\mu) \leq \mu_{max}$ for all μ and some μ_{max}.

(a) Consider the sequence $\mu(t+1) = G(\mu(t))$ initialized with $\mu(1) = \mu_{max}$. Then

$$\lim_{t\to\infty} \mu(t) = \mu,$$

for some μ with $\mu = G(\mu)$. Moreover, the limiting value μ is the largest value satisfying $\mu = G(\mu)$ and $\mu \in [0, \mu_{max}]$.

(b) Similarly, if the above sequence is initialized with $\mu(1) = 0$, then $\mu(t) \to \mu$ where μ is the smallest value satisfying $\mu \in [0, \mu_{max}]$ and the fixed point equation $\mu = G(\mu)$.

Proof: We will just prove part (a) as part (b) is similar. We first prove by, induction, that $\mu(t+1) \leq \mu(t)$ for all t. For $t = 1$, since $\mu(1)$ is initialized to μ_{max} and $G(\mu) \leq \mu_{max}$ for all μ, we have that

$$\mu(2) = G(\mu(1)) \leq \mu_{max} = \mu(1).$$

Now suppose that $\mu(t+1) \leq \mu(t)$ for some t. Then, using the monotonicity of $G(\mu)$,

$$\mu(t+2) = G(\mu(t+1)) \leq G(\mu(t)) = \mu(t+1).$$

So, by induction, $\mu(t)$ is a monotonically decreasing sequence. Since it is bounded below by zero, it must converge to some μ. By the continuity of $G(\mu)$, the limit point must satisfy the fixed point equation $\mu = G(\mu)$.

It remains to show that the limiting value μ is the largest fixed point of G in the interval $[0, \mu_{max}]$. To this end, let μ_1 be any fixed point $\mu_1 = G(\mu_1)$ with $0 \leq \mu_1 \leq \mu_{max}$. Then, $\mu(1) = \mu_{max} \geq \mu_1$. Also, if $\mu(t) \geq \mu_1$, by the monotonicity of G,

$$\mu(t+1) = G(\mu(t)) \geq \mu_1.$$

So, by the induction the entire sequence $\mu(t) \geq \mu_1$. Taking the limits as $t \to \infty$, we have that $\mu \geq \mu_1$. Hence $\mu \geq \mu_1$ for any other fixed point of G. ∎

We can now prove Theorem 2. Define the function

$$G(\mu) = \beta\overline{\mathcal{E}}_{\rm in}\left[\overline{\mathcal{E}}_{\rm out}(\mu)\right], \tag{87}$$

so that we can rewrite the density evolution equation (38) as

$$\mu^z(t+1) = G(\mu^z(t)).$$

We will now apply Lemma 9 to show that $\mu_z(t) \to \mu$ to a fixed point $\mu = G(\mu)$. We first upper bound $G(\mu)$. For all $\mu > 0$,

$$\mathbf{E}\left[\mathcal{E}_{\rm in}(q, \mu)\right] \overset{(a)}{=} \mathbf{var}(X|Q) \overset{(b)}{\leq} \mathbf{var}(X) \overset{(c)}{=} \mu^x_{\rm init}.$$

where the expectation is over the random variable $q = x + v$, $v \sim \mathcal{N}(0, \mu)$; (a) follows from the definition of $\mathcal{E}_{\rm in}(q, \mu)$; (b) is the fact that conditioning cannot increase the variance; and (c) is from the definition of $\mu^x_{\rm init}$ in (17). Therefore, the definition of $\overline{\mathcal{E}}_{\rm in}(\mu)$ in (32) implies that

$$\overline{\mathcal{E}}_{\rm in}(\mu) = \mathbf{E}\left[s\mathcal{E}_{\rm in}(q, \mu/s)\right] \leq \mathbf{E}(s)\mu^x_{\rm init}.$$

As a result, $G(\mu)$ defined in (87) satisfies

$$G(\mu) \leq \beta\mathbf{E}(s)\mu^x_{\rm init} = \mu^z_{\rm init},$$

where $\mu^z_{\rm init}$ is defined in (33). So, we have that $G(\mu) \leq \mu^z_{\rm init}$ for all μ. Also, the "high" sequence $\mu^z_{\rm hi}(t)$ is initialized with $\mu^z_{\rm hi}(1) = \mu^z_{\rm init}$ and the "low" sequence with $\mu^z_{\rm lo}(1) = 0$ So, we will apply Lemma 9 with $\mu_{max} = \mu^z_{\rm init}$.

By the assumption of the theorem, $\overline{\mathcal{E}}_{\rm in}(\mu)$ and $\overline{\mathcal{E}}_{\rm out}(\mu)$ are continuous. Therefore, so is $G(\mu)$.

Hence, to apply Lemma 9, it remains to show that $G(\mu)$ is monotonically increasing. From (87), we need to show that $\overline{\mathcal{E}}_{\rm in}(\mu)$ and $\overline{\mathcal{E}}_{\rm out}(\mu)$ are monotonically increasing.

We first consider $\overline{\mathcal{E}}_{\rm in}(\mu)$. Let $\mu_2 \geq \mu_1$ and define the random variables

$$\begin{aligned} q_1 &= x + v_1, \quad v_1 \sim \mathcal{N}(0, \mu_1/s), \\ q_2 &= q_1 + w, \quad w \sim \mathcal{N}(0, (\mu_2 - \mu_1)/s), \end{aligned}$$

where $x \sim p_X(x)$, $s \sim p_S(s)$, and v_1 and v_2 are independent. We have that $x \to q_1 \to q_2$ is a Markov chain, so Lemma 8(a) shows that, for all s,

$$\mathbf{var}(X|Q_1, S = s) \leq \mathbf{var}(X|Q_2, S = s). \tag{88}$$

Also, q_2 is identically distributed to $q_2 = x + v_2$, $v_2 \sim \mathcal{N}(0, \mu_2/s)$ for some v_2 independent of x. The definition of $\overline{\mathcal{E}}_{\rm in}(\mu)$ shows that for $i = 1, 2$,

$$\overline{\mathcal{E}}_{\rm in}(\mu_i) = \mathbf{E}\left[s\,\mathbf{var}(X|Q_i, S = s)\right]. \tag{89}$$

Combining (88) and (89) shows that $\overline{\mathcal{E}}_{\rm in}(\mu)$ is monotonically increasing in μ.

The proof that $\overline{\mathcal{E}}_{\rm out}(\mu)$ is monotonically increasing is similar. Let μ_1 and μ_2 be variances such that

$$0 \leq \mu_1 \leq \mu_2 \leq \mu^z_{\rm init}.$$

For $u \in \mathbb{R}$, define the random variables

$$\begin{aligned} \widehat{z}_2 &\sim \mathcal{N}(0, \mu^z_{\rm init} - \mu_2) \\ \widehat{z}_1 &\sim \widehat{z}_2 + \mathcal{N}(0, \mu_2 - \mu_1) \\ z &\sim u + \widehat{z}_1 + \mathcal{N}(0, \mu_1) \end{aligned}$$

where all the Gaussian random variables are independent. Also, conditional on z, let y have the distribution $y \sim p_{Y|Z}(y|z)$. It can be verified that

$$u \to (\widehat{z}_1, y) \to (\widehat{z}_2, y)$$

is a Markov chain. It follows from Lemma 8(b) that

$$F(\widehat{Z}_1, Y|U = 0) \geq F(\widehat{Z}_2, Y|U = 0). \tag{90}$$

Also, the definitions of z, $\widehat{z}_1$ and $\widehat{z}_2$ above show that, for $i = 1, 2$, when $u = 0$,

$$(z, \widehat{z}_i) \sim \mathcal{N}(0, P_z(\mu_i)),$$

where $P_z(\mu)$ is defined in (36).

Now, the Fisher information satisfies

$$\begin{aligned} & F(\widehat{Z}_i, Y|U=0) \\ & \overset{(a)}{=} \quad -\mathbf{E}\left[\frac{\partial^2}{\partial u^2} \log p_{\widehat{Z}_i,Y|U}(\widehat{z}_i, y|u) \right| \left. u=0 \vphantom{\frac{\partial^2}{\partial u^2}} \right]. \\ & \overset{(b)}{=} \quad -\mathbf{E}\left[\frac{\partial^2}{\partial u^2} \log p_{Y|U,\widehat{Z}_i}(y|u, \widehat{z}_i) \right| \left. u=0 \vphantom{\frac{\partial^2}{\partial u^2}} \right]. \\ & \overset{(c)}{=} \quad \mathbf{E}\left[D_2(y, \widehat{z}_i, \mu_i) \right]. \\ & \overset{(d)}{=} \quad \frac{1}{\overline{\mathcal{E}}_{\rm out}(\mu_i)} \end{aligned} \tag{91}$$

where (a) follows from the definition of the Fisher information in (86); (b) follows from the fact that

$$\begin{aligned} & \log p_{\widehat{Z}_i,Y|U}(\widehat{z}_i, y|u) \\ & \quad = \quad \log p_{Y|\widehat{Z}_i,U}(y|u, \widehat{z}_i) + \log p_{\widehat{Z}_i|U}(\widehat{z}_i|u) \end{aligned}$$

and $\widehat{z}_i$ is independent of u; (c) is the definition of $D_r(y, \widehat{z}, \mu)$ in (16); and (d) follows from the definition of $\overline{\mathcal{E}}_{\rm out}(\mu)$ in (34). Equations (90) and (34) together now show that $\overline{\mathcal{E}}_{\rm out}(\mu)$ is monotonically increasing in μ. Since $\overline{\mathcal{E}}_{\rm in}(\mu)$ is also monotonically increasing in μ, so is $G(\mu)$.

Lemma 9 thus shows that $\mu^z_{\rm hi}(t)$ converges to the largest fixed point solution of the equation $\mu = G(\mu)$ and $\mu^z_{\rm lo}(t)$ converges to the smallest fixed point.

Acknowledgments

The author particularly thanks Dongning Guo and Chih-Chun Wang, whose ideas are the basis of this work, and Vivek Goyal for their feedback. The author also thanks Dror Baron, Andrea Montanari, Phil Schniter, John Sun, Lav Varshney and Martin Wainwright for their careful reading of the manuscript and Amin Shokrollahi and Martin Vetterli for their generous support at the Summer Research Institute at EPFL, 2009, where this research was initiated.

References

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

[2] 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.

[3] 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. 1988.

[4] 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.

[5] D. J. C. MacKay and R. M. Neal, "Near shannon limit performance of low density parity check codes," *Electron. Letters*, vol. 33.

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

[7] G. Guo and C. C. Wang, "Multiuser detection of sparsely spread CDMA," *IEEE J. Sel. Areas Comm.*, vol. 26, no. 3, p. 421431, Mar. 2008.

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

[9] 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.

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

[11] D. L. Donoho, A. Maleki, and A. Montanari, "Message passing algorithms for compressed sensing," arXiv:0907.3574v1 [cs.IT], Jul. 2009.

[12] 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.

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

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

[15] 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.

[16] 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.

[17] Y. Kabashima, "A CDMA multiuser detection algorithm on the basis of belief propagation," *J. Phys. A: Math. Gen*, vol. 36.

[18] 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.

[19] J. P. Neirotti and D. Saad, "Improved message passing for inference in densely connected systems," *Europhys. Lett.*, vol. 71, no. 5, p. 866872, Sep. 2005.

[20] 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.

[21] 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.

[22] M. Bayati and A. Montanari, "The dynamics of message passing on dense graphs, with applications to compressed sensing," arXiv:1001.3448v1 [cs.IT], Jan. 2010.

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

[24] 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.

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

[26] 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.

[27] J. Yedidia, W. T. Freeman, and Y. Weiss, *Understanding belief propagation and its generalizations*, ser. Exploring Artificial Intelligence in the New Millenium. San Francisco, CA: Morgan Kaufmann, 2003, ch. 8, pp. 239–269.

[28] S. Rangan, A. Fletcher, and V. K. Goyal, "Asymptotic analysis of MAP estimation via the replica method and applications to compressed sensing," arXiv:0906.3234v2 [cs.IT]., Aug. 2009.

[29] 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.

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

[31] R. M. Gray, "Quantization noise spectra," *IEEE Trans. Inform. Theory*, vol. 36, no. 6, pp. 1220–1244, Nov. 1990.

[32] S. P. Lipshitz, R. A. Wannamaker, and J. Vanderkooy, "Quantization and dither: A theoretical survey," *J. Audio Eng. Soc.*, vol. 40, no. 5, pp. 355–375, May 1992.

[33] N. T. Thao and M. Vetterli, "Reduction of the MSE in R-times oversampled A/D conversion from $O(1/R)$ to $O(1/R^2)$," *IEEE Trans. Signal Process.*, vol. 42, no. 1, pp. 200–203, Jan. 1994.

[34] ——, "Deterministic analysis of oversampled A/D conversion and decoding improvement based on consistent estimates," *IEEE Trans. Signal Process.*, vol. 42, no. 3, pp. 519–531, Mar. 1994.

[35] ——, "Lower bound on the mean-squared error in oversampled quantization of periodic signals using vector quantization analysis," *IEEE Trans. Inform. Theory*, vol. 42, no. 2, pp. 469–479, Mar. 1996.

[36] Z. Cvetković, "Source coding with quantized redundant expansions: Accuracy and reconstruction," in *Proc. IEEE Data Compression Conf.*, Snowbird, Utah, Mar. 1999, pp. 344–353.

[37] S. Rangan and V. K. Goyal, "Recursive consistent estimation with bounded noise," *IEEE Trans. Inform. Theory*, vol. 47, no. 1, pp. 457–464, Jan. 2001.

[38] G. R. Grimmett and D. R. Stirzaker, *Probability and Random Processes*, 2nd ed. Oxford Univ. Press, 1992.

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

[40] ——, *Modern Coding Theory*. Cambridge, UK: Cambridge University Press, 2009.

[41] ——, "The capacity of low-density parity check codes under message-passing decoding," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 599–618, Feb. 2001.

[42] E. Lehmann and G. Casella, *Theory of Point Estimation*, 2nd ed., ser. Springer Texts in Statistics. New York, NY: Springer, Sep. 2003.