Inference for Generalized Linear Models via Alternating Directions and Bethe Free Energy Minimization

Sundeep Rangan*, Alyson K. Fletcher†, Philip Schniter‡ and Ulugbek S. Kamilov§
*NYU-Poly, Elec. & Comp. Engineering, srangan@NYU.edu
†U.C. Santa Cruz., Elec. Engineering, afletcher@ucsc.edu
‡The Ohio State Univ., Elec. & Comp. Engineering, schniter@ece.osu.edu
§EPFL, Biomed. Imaging Group, ulugbek.kamilov@epfl.ch

***Abstract*—Generalized Linear Models (GLMs), where a random vector $\mathbf{x}$ is observed through a noisy, possibly nonlinear, function of a linear transform $\mathbf{z} = \mathbf{A}\mathbf{x}$ arise in a range of applications in nonlinear filtering and regression. Approximate Message Passing (AMP) methods, based on loopy belief propagation, are a promising class of approaches for approximate inference in these models. AMP methods are computationally simple, general, and admit precise analyses with testable conditions for optimality for large i.i.d. transforms $\mathbf{A}$. However, the algorithms can easily diverge for general transforms. This paper presents a convergent approach to the generalized AMP (GAMP) algorithm based on direct minimization of a large-system limit approximation of the Bethe Free Energy (LSL-BFE). The proposed method uses a double-loop procedure, where the outer loop successively linearizes the LSL-BFE and the inner loop minimizes the linearized LSL-BFE using the Alternating Direction Method of Multipliers (ADMM). The proposed method, called ADMM-GAMP, is similar in structure to the original GAMP method, but with an additional least-squares minimization. It is shown that for strictly convex, smooth penalties, ADMM-GAMP is guaranteed to converge to a local minima of the LSL-BFE, thus providing a convergent alternative to GAMP that is stable under arbitrary transforms. Simulations are also presented that demonstrate the robustness of the method for non-convex penalties as well.**

***Index Terms*—Belief propagation, ADMM, variational optimization, message passing, generalized linear models.**

I. Introduction

Consider the problem of estimating a random vector $\mathbf{x} \in \mathbb{R}^n$ from observations $\mathbf{y} \in \mathbb{R}^m$ as shown in Fig. 1. The unknown vector is assumed to have a prior density of the form $p(\mathbf{x}) = \mathrm{e}^{-f_x(\mathbf{x})}$ and the observations $\mathbf{y} \in \mathbb{R}^m$ are described by a likelihood function of the form $p(\mathbf{y}|\mathbf{A}\mathbf{x}) = \mathrm{e}^{-f_z(\mathbf{A}\mathbf{x})}$, for some known transform $\mathbf{A} \in \mathbb{R}^{m\times n}$. In statistics, this model is a special case of a generalized linear model (GLM) and arises in a range of applications including statistical regression, filtering, inverse problems, and nonlinear forms of compressed sensing. The posterior density of $\mathbf{x}$ given $\mathbf{y}$ in the GLM model is given by

$$p_{\mathbf{x}|\mathbf{y}}(\mathbf{x}|\mathbf{y}) = \frac{1}{Z}\exp\left[-f_x(\mathbf{x}) - f_z(\mathbf{A}\mathbf{x})\right], \qquad (1)$$

where Z is a normalization constant. In this work, we consider the inference problem of estimating the posterior marginal distributions, $p_{x_j|\mathbf{y}}(x_j|\mathbf{y})$. From these posterior marginals, one can compute the posterior means and variances

$$\widehat{x}_j \triangleq \mathbb{E}(x_j|\mathbf{y}), \quad \tau_{x_j} \triangleq \mathrm{var}(x_j|\mathbf{y}). \qquad (2)$$

We study this inference problem in the case where the functions f_x and f_z are separable, in that they are of the form

$$f_x(\mathbf{x}) = \sum_{j=1}^{n} f_{x_j}(x_j), \quad f_z(\mathbf{z}) = \sum_{i=1}^{m} f_{z_i}(z_i), \qquad (3)$$

for some scalar functions f_{x_j} and f_{z_i}.

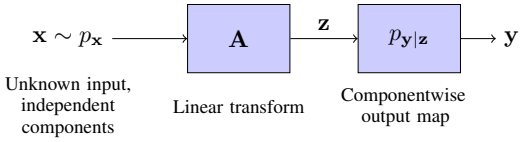

Fig. 1. Generalized Linear Model (GLM) where an unknown random vector $\mathbf{x}$ is observed via a linear transform followed by componentwise likelihood to yield a measurement vector $\mathbf{y}$.

In recent years, Bayesian forms of approximate message passing (AMP) have been considered for approximate inference in GLMs [1]–[4]. AMP methods are based on Gaussian and quadratic approximations to loopy belief propagation (loopy BP) in graphical models and are both computationally simple and applicable to arbitrary separable penalty functions f_x and f_z. In addition, for certain large i.i.d. transforms $\mathbf{A}$, they have the benefit that the behavior of the algorithm can be exactly predicted by a state evolution analysis, which then provides testable conditions for Bayes optimality [4]–[6].

Unfortunately, for general $\mathbf{A}$, AMP methods may diverge [7], [8] – a situation that is not surprising given that AMP is based on loopy BP, which also may diverge. Several recent modifications have been proposed to improve the stability of AMP, including damping [7], sequential updating [9], and adaptive damping [10], some of which have been instrumental in applications such as [11]–[13]. However, while these methods appear to perform empirically well, little has been proven rigorously about their convergence.

The main goal of this paper is to provide a provably convergent approach to AMP. We focus on the generalized AMP (GAMP) method of [4], which allows arbitrary separable functions for both f_x and f_z. Our approach to stabilizing GAMP is based on reconsidering the inference problem as a type of free-energy minimization. Specifically, it is known that GAMP can be considered as an iterative procedure that attempts to minimize a large-system-limit approximation of the so-called Bethe Free Energy (BFE) [14], [15], which we abbreviate as "LSL-BFE" in the sequel.

To minimize the LSL-BFE, we propose a double-loop algorithm, similar to the well-known Convex Concave Procedure (CCCP) [16]. Our main theoretical result shows that, for strictly convex penalties, the proposed algorithm is guaranteed to converge to at least a local minima of the LSL-BFE. In this way, we obtain a variant of the GAMP method with a provable convergence guarantee for arbitrary transforms $\mathbf{A}$.

II. ADMM GAMP

A. Bethe Free Energy Minimization

Loopy belief propagation (loopy BP) in graphical models provides a generic method for approximately computing marginals $p(x_j|\mathbf{y})$ of a potentially high-dimensional density $p(\mathbf{x}|\mathbf{y})$ by minimizing an energy function called the Bethe Free Energy (BFE) [17]. Unfortunately, for the GLM model in the previous section, the computations in loopy BP may be difficult for dense matrices $\mathbf{A}$. The sum-product GAMP algorithm from [4] can be interpreted as a method for minimizing an approximation of the BFE that applies to certain large, dense $\mathbf{A}$ [14], [15]. Specifically, the sum-product GAMP algorithm finds estimates $\widehat{b}_x(\mathbf{x})$ and $\widehat{b}_z(\mathbf{z})$ of the posterior densities $p(\mathbf{x}|\mathbf{y})$ and $p(\mathbf{z}|\mathbf{y})$ via the minimization,

$$(\widehat{b}_x, \widehat{b}_z) \triangleq \underset{b_x, b_z}{\arg\min}\, J(b_x, b_z) \text{ such that} \tag{4a}$$

$$\mathbb{E}(\mathbf{z}|b_z) = \mathbf{A}\mathbb{E}(\mathbf{x}|b_x) \tag{4b}$$

where b_x and b_z are product densities, i.e.,

$$b_x(\mathbf{x}) = \prod_{j=1}^{n} b_{x_j}(x_j),\ b_z(\mathbf{z}) = \prod_{i=1}^{m} b_{z_i}(z_i), \tag{5}$$

and the objective function $J(b_x, b_z)$ is given by

$$\begin{aligned} J(b_x, b_z) &\triangleq D(b_x \| \mathrm{e}^{-f_x}) + D(b_z \| Z_z^{-1}\mathrm{e}^{-f_z}) \\ &\quad + H\big(\mathrm{var}(\mathbf{x}|b_x), \mathrm{var}(\mathbf{z}|b_z)\big) \end{aligned} \tag{6}$$

$$H(\boldsymbol{\tau}_x, \boldsymbol{\tau}_z) \triangleq \frac{1}{2}\sum_{i=1}^{m}\left[\frac{\tau_{z_i}}{\sum_{j=1}^{n} S_{ij}\tau_{x_j}} + \ln\left(2\pi\sum_{j=1}^{n} S_{ij}\tau_{x_j}\right)\right] \tag{7}$$

$$\boldsymbol{\tau}_x \triangleq (\tau_{x_1}, \ldots, \tau_{x_n})^{\mathsf{T}}, \quad \tau_{x_j} \triangleq \mathrm{var}(x_j|b_{x_j}) \tag{8}$$

$$\boldsymbol{\tau}_z \triangleq (\tau_{z_1}, \ldots, \tau_{z_m})^{\mathsf{T}}, \quad \tau_{z_i} \triangleq \mathrm{var}(z_i|b_{z_i}) \tag{9}$$

$$S_{ij} = [\mathbf{S}]_{ij} \triangleq [\mathbf{A}]_{ij}^2 \ \forall ij, \tag{10}$$

where $D(\cdot\|\cdot)$ denotes KL divergence, $Z_z \triangleq \int_{\mathbb{R}^m} \mathrm{e}^{-f_z(\mathbf{z})}\,\mathrm{d}\mathbf{z}$ is the scale factor that makes $Z_z^{-1}\mathrm{e}^{-f_z(\mathbf{z})}$ a valid density over $\mathbf{z} \in \mathbb{R}^m$, and $H(\boldsymbol{\tau}_x, \boldsymbol{\tau}_z)$ is an upper bound on the entropy of b_z that is tight for Gaussian b_z with $\boldsymbol{\tau}_z = \mathbf{S}\boldsymbol{\tau}_x$. The objective function of the optimization in (6) can be interpreted as an approximation of the BFE for the GLM from Section I in a certain large-system limit, where $m, n \to \infty$ and $\mathbf{A}$ has i.i.d. components [15]. We thus call this approximate BFE the *large-system limit Bethe Free Energy* or LSL-BFE.

B. Outer Loop Minimization via Iterative Linearization

While the fixed points of the sum-product GAMP algorithm correspond to local minima of the LSL-BFE minimization in (4), the sum-product GAMP algorithm may not always converge (see, e.g., the negative results in [7], [8], [10]). We thus consider an alternative minimization strategy based on a generalization of the classic double-loop method known as the concave convex procedure (CCCP) [16].

The proposed algorithm for the GLM problem is shown in Algorithm 1, and can be briefly explained as follows (see the full paper [18] for a complete discussion). First, as shown in [18], the $\overline{\boldsymbol{\tau}}_r^k$ and $\overline{\boldsymbol{\tau}}_p^k$ computed in lines 9-11 of Algorithm 1 are the inverse gradients of the Gaussian entropy term $H(\boldsymbol{\tau}_x, \boldsymbol{\tau}_z)$ in (7) at the current estimates of the variances $\boldsymbol{\tau}_x^k, \boldsymbol{\tau}_z^k$:

$$\frac{1}{2\overline{\boldsymbol{\tau}}_r^k} = \frac{\partial H(\boldsymbol{\tau}_x^k, \boldsymbol{\tau}_z^k)}{\partial \boldsymbol{\tau}_x}, \quad \frac{1}{2\overline{\boldsymbol{\tau}}_p^k} = \frac{\partial H(\boldsymbol{\tau}_x^k, \boldsymbol{\tau}_z^k)}{\partial \boldsymbol{\tau}_z}. \tag{11}$$

The $\boldsymbol{\tau}_r^{k+1}$ and $\boldsymbol{\tau}_p^{k+1}$ computed in lines 15 and 16 of Algorithm 1 are "damped" versions of $\overline{\boldsymbol{\tau}}_p^k$ and $\overline{\boldsymbol{\tau}}_r^k$, controlled by damping parameter θ^k. The belief estimates (b_x^k, b_z^k) are then computed by minimizing the functional

$$\begin{aligned} J(b_x, b_z, \boldsymbol{\tau}_r, \boldsymbol{\tau}_p) &\triangleq D(b_x\|\mathrm{e}^{-f_x}) + D(b_z\|Z_z^{-1}\mathrm{e}^{-f_z}) \\ &\quad + \left(\frac{1}{2\boldsymbol{\tau}_r}\right)^{\mathsf{T}} \mathrm{var}(\mathbf{x}|b_x) + \left(\frac{1}{2\boldsymbol{\tau}_p}\right)^{\mathsf{T}} \mathrm{var}(\mathbf{z}|b_z), \end{aligned} \tag{12}$$

subject to the linear constraints $(b_x, b_z) \in B$ where

$$B = \{(b_x, b_z) \mid \mathbb{E}(\mathbf{z}|b_z) = \mathbf{A}\mathbb{E}(\mathbf{x}|b_x)\}. \tag{13}$$

The functional (12) is a modified form of the LSL-BFE in (6) where the Gaussian entropy term $H(\boldsymbol{\tau}_x, \boldsymbol{\tau}_z)$ is replaced by a first-order approximation based on the approximate gradients $(1/2\boldsymbol{\tau}_r, 1/2\boldsymbol{\tau}_p)$. Hence, Algorithm 1 can be interpreted as a method that iteratively minimizes a partial linearization of the LSL-BFE, with a damped update on the gradient term in the linear approximation.

This approach is closely related to the CCCP method from [16], which also iteratively minimizes a partial linear approximation of the BFE. (A complete discussion can be found in our full paper [18].) Similar to the CCCP method, Algorithm 1 can be considered as a "double loop" method, since each iteration involves a minimization in line 5 that itself is typically solved iteratively.

C. Inner-Loop Minimization via ADMM

Algorithm 1, requires that, in each iteration of the outer loop, we solve the constrained optimization

$$(b_x, b_z) = \underset{b_x, b_z}{\arg\min}\, J(b_x, b_z, \boldsymbol{\tau}_r, \boldsymbol{\tau}_p) \text{ s.t. } \mathbb{E}(\mathbf{z}|b_z) = \mathbf{A}\mathbb{E}(\mathbf{x}|b_x). \tag{14}$$

Algorithm 1 Minimizing LSL-BFE via Iterative Linearization

Require: LSL-BFE objective function (6) with a matrix $\mathbf{A}$.

$k \leftarrow 0$
Select initial linearization $\boldsymbol{\tau}_p^0$, $\boldsymbol{\tau}_r^0$.
repeat
  {Minimize the linearized LSL-BFE}
  $(b_x^k, b_z^k) \leftarrow \arg\min_{(b_x,b_z)\in B} J(b_x, b_z, \boldsymbol{\tau}_r^k, \boldsymbol{\tau}_p^k)$

  {Compute the gradient terms}
  $\boldsymbol{\tau}_x^k \leftarrow \mathrm{var}(\mathbf{x}|b_x^k)$, $\boldsymbol{\tau}_z^k \leftarrow \mathrm{var}(\mathbf{z}|b_z^k)$
  $\overline{\boldsymbol{\tau}}_p^k \leftarrow \mathbf{S}\boldsymbol{\tau}_x^k$
  $\boldsymbol{\tau}_s^k \leftarrow (\mathbf{1} - \boldsymbol{\tau}_z^k/\overline{\boldsymbol{\tau}}_p^k)/\overline{\boldsymbol{\tau}}_p^k$
  $\overline{\boldsymbol{\tau}}_r^k \leftarrow 1/(\mathbf{S}^{\mathsf{T}}\boldsymbol{\tau}_s^k)$

  {Update the linearization}
  Select a damping parameter $\theta^k \in (0, 1]$
  $1/\boldsymbol{\tau}_r^{k+1} \leftarrow \theta^k/\overline{\boldsymbol{\tau}}_r^k + (1-\theta^k)/\boldsymbol{\tau}_r^k$
  $1/\boldsymbol{\tau}_p^{k+1} \leftarrow \theta^k/\overline{\boldsymbol{\tau}}_p^k + (1-\theta^k)/\boldsymbol{\tau}_p^k$
until Terminated

This optimization can be performed by the Alternating Direction Method of Multipliers (ADMM) [19] as follows. First, we replace the constraint $\mathbb{E}(\mathbf{z}|b_z) = \mathbf{A}\mathbb{E}(\mathbf{x}|b_x)$ with two constraints: $\mathbb{E}(\mathbf{z}|b_z) = \mathbf{A}\mathbf{v}$ and $\mathbb{E}(\mathbf{x}|b_x) = \mathbf{v}$. Variable splittings of this form are commonly used in the context of monotropic programming [20]. With this splitting, the augmented Lagrangian for the linearized LSL-BFE (14) becomes

$$\begin{aligned} &L(b_x, b_z, \mathbf{s}, \mathbf{q}, \mathbf{v}; \boldsymbol{\tau}_p, \boldsymbol{\tau}_r) \\ &\triangleq J(b_x, b_z, \boldsymbol{\tau}_r, \boldsymbol{\tau}_p) + \mathbf{q}^{\mathsf{T}}(\mathbb{E}(\mathbf{x}|b_x) - \mathbf{v}) + \mathbf{s}^{\mathsf{T}}(\mathbb{E}(\mathbf{z}|b_z) - \mathbf{A}\mathbf{v}) \\ &\quad + \frac{1}{2}\|\mathbb{E}(\mathbf{x}|b_x) - \mathbf{v}\|^2_{\boldsymbol{\tau}_r} + \frac{1}{2}\|\mathbb{E}(\mathbf{z}|b_z) - \mathbf{A}\mathbf{v}\|^2_{\boldsymbol{\tau}_p}, \end{aligned} \tag{15}$$

where $\mathbf{s}$ and $\mathbf{q}$ represent the dual variables. From [19], the standard ADMM recursion for this problem is

$$(b_x^{t+1}, b_z^{t+1}) = \arg\min_{b_x, b_z} L(b_x, b_z, \mathbf{s}^t, \mathbf{q}^t, \mathbf{v}^t; \boldsymbol{\tau}_p, \boldsymbol{\tau}_r) \tag{16a}$$

$$\mathbf{s}^{t+1} = \mathbf{s}^t + (1/\boldsymbol{\tau}_p)\big(\mathbb{E}(\mathbf{z}|b_z^{t+1}) - \mathbf{A}\mathbf{v}^t\big) \tag{16b}$$

$$\mathbf{q}^{t+1} = \mathbf{q}^t + (1/\boldsymbol{\tau}_r)\big(\mathbb{E}(\mathbf{x}|b_x^{t+1}) - \mathbf{v}^t\big) \tag{16c}$$

$$\mathbf{v}^{t+1} = \arg\min_{\mathbf{v}} L(b_x^{t+1}, b_z^{t+1}, \mathbf{s}^{t+1}, \mathbf{q}^{t+1}, \mathbf{v}; \boldsymbol{\tau}_p, \boldsymbol{\tau}_r). \tag{16d}$$

The full paper [18] details all of the minimizations in (16). It shows, e.g., that the belief estimates in (16a) are given by

$$b_x^{t+1}(\mathbf{x}) \propto \exp\big(-f_x(\mathbf{x}) - \tfrac{1}{2}\|\mathbf{x} - \mathbf{r}^t\|^2_{\boldsymbol{\tau}_r}\big) \tag{17a}$$

$$b_z^{t+1}(\mathbf{z}) \propto \exp\big(-f_z(\mathbf{z}) - \tfrac{1}{2}\|\mathbf{z} - \mathbf{p}^t\|^2_{\boldsymbol{\tau}_p}\big), \tag{17b}$$

where the vectors $\mathbf{r}^t$ and $\mathbf{p}^t$ are computed in Algorithm 2, and that (16d) reduces to the least-squares problem

$$\begin{aligned} \mathbf{v}^{t+1} = \arg\min_{\mathbf{v}} &\|\mathbf{z}^{t+1} + \boldsymbol{\tau}_p\mathbf{s}^{t+1} - \mathbf{A}\mathbf{v}\|^2_{\boldsymbol{\tau}_p} \\ &+ \|\mathbf{x}^{t+1} + \boldsymbol{\tau}_r\mathbf{q}^{t+1} - \mathbf{v}\|^2_{\boldsymbol{\tau}_r}. \end{aligned} \tag{18}$$

Due to the separable nature of b_x and b_z (recall (5)), the posterior means in (16b)-(16c) can be evaluated one component at a time, and for many problems the resulting scalar integral can be computed in closed form. These computations are written in line 9 of Algorithm 2 as

$$\mathbf{x}^{t+1} = g_x(\mathbf{r}^t, \boldsymbol{\tau}_r^t) \triangleq \mathbb{E}(\mathbf{x}|b_x^{t+1}) \tag{19a}$$

$$\mathbf{z}^{t+1} = g_z(\mathbf{p}^t, \boldsymbol{\tau}_p^t) \triangleq \mathbb{E}(\mathbf{z}|b_z^{t+1}), \tag{19b}$$

where we call g_x and g_z the *MMSE estimation functions*. The general "g_x, g_z" notation is useful because, with slightly different definitions of g_x and g_z, the algorithm can also be used for MAP estimation, as discussed in the full paper [18].

We note that Algorithm 2 is written in a "parallel" form that allows the computation of one ADMM inner-loop update per outer-loop update. However, as discussed in [18], the update schedule can be controlled by the damping parameter θ^t. In particular, by setting $\theta^t = 0$ for a specified number of iterations t, many inner-loop updates can be exectured with fixed value of the outer-loop linearization parameters $\boldsymbol{\tau}_r, \boldsymbol{\tau}_p$.

Interestingly, Algorithm 2 has close similarities to the original GAMP algorithm from [4]. For example, most of the steps in Algorithm 2 can also be found in the GAMP algorithm, and the definitions of the estimation functions g_x and g_z remain the same. We thus call Algorithm 2 "ADMM-GAMP." The main differences between GAMP and ADMM-GAMP is that the latter i) incorporates a least-squares minimization (18), ii) adds damping to the updates of $\boldsymbol{\tau}_r^t$ and $\boldsymbol{\tau}_p^t$, and iii) uses $\overline{\boldsymbol{\tau}}_p^{t+1}$ rather than $\boldsymbol{\tau}_p^t$ in the update of $\boldsymbol{\tau}_s$. A complete discussion of the differences can be found in [18].

Algorithm 2 ADMM-GAMP

Require: Matrix $\mathbf{A}$, estimation functions g_x and g_z.

1: $\mathbf{S} \leftarrow |\mathbf{A}|^2$ (componentwise magnitude squared)
2: Initialize $\boldsymbol{\tau}_r^0 > \mathbf{0}, \boldsymbol{\tau}_p^0 > \mathbf{0}$, $\mathbf{v}^0$
3: $\mathbf{q}^0 \leftarrow \mathbf{0}$, $\mathbf{s}^0 \leftarrow \mathbf{0}$
4: $t \leftarrow 0$
5: **repeat**
6: {ADMM inner iteration}
7: $\mathbf{r}^t \leftarrow \mathbf{v}^t - \boldsymbol{\tau}_r^t\mathbf{q}^t$
8: $\mathbf{p}^t \leftarrow \mathbf{A}\mathbf{v}^t - \boldsymbol{\tau}_p^t\mathbf{s}^t$
9: $\mathbf{x}^{t+1} \leftarrow g_x(\mathbf{r}^t, \boldsymbol{\tau}_r^t)$, $\mathbf{z}^{t+1} \leftarrow g_z(\mathbf{p}^t, \boldsymbol{\tau}_p^t)$
10: $\mathbf{q}^{t+1} \leftarrow \mathbf{q}^t + (1/\boldsymbol{\tau}_r^t)(\mathbf{x}^{t+1} - \mathbf{v}^t)$
11: $\mathbf{s}^{t+1} \leftarrow \mathbf{s}^t + (1/\boldsymbol{\tau}_p^t)(\mathbf{z}^{t+1} - \mathbf{A}\mathbf{v}^t)$
12: Compute $\mathbf{v}^{t+1}$ from (18)
13:
14: {Compute the gradient terms}
15: $\boldsymbol{\tau}_x^{t+1} \leftarrow \boldsymbol{\tau}_r^t g_x'(\mathbf{r}^t, \boldsymbol{\tau}_r^t)$, $\boldsymbol{\tau}_z^{t+1} \leftarrow \boldsymbol{\tau}_p^t g_z'(\mathbf{p}^t, \boldsymbol{\tau}_p^t)$
16: $\overline{\boldsymbol{\tau}}_p^{t+1} \leftarrow \mathbf{S}\boldsymbol{\tau}_x^{t+1}$
17: $\boldsymbol{\tau}_s^{t+1} \leftarrow (1 - \boldsymbol{\tau}_z^{t+1}/\overline{\boldsymbol{\tau}}_p^{t+1})/\overline{\boldsymbol{\tau}}_p^{t+1}$
18: $\overline{\boldsymbol{\tau}}_r^{t+1} \leftarrow 1/(\mathbf{S}^{\mathsf{T}}\boldsymbol{\tau}_s^{t+1})$
19:
20: {Update the linearization}
21: Select a damping parameter $\theta^t \in [0, 1]$
22: $1/\boldsymbol{\tau}_r^{t+1} \leftarrow \theta^t/\overline{\boldsymbol{\tau}}_r^{t+1} + (1-\theta^t)/\boldsymbol{\tau}_r^t$
23: $1/\boldsymbol{\tau}_p^{t+1} \leftarrow \theta^t/\overline{\boldsymbol{\tau}}_p^{t+1} + (1-\theta^t)/\boldsymbol{\tau}_p^t$
24: **until** Terminated

III. Convergence Analysis for Strictly Convex Penalties

A. Convergence of the ADMM Inner Loop

It is shown in the full paper that the fixed points of ADMM-GAMP with the MMSE estimation functions (19) correspond to local minima of the LSL-BFE optimization (6). To understand the convergence of the algorithm to a fixed point, we first establish convergence of the inner loop. For this, we make the following assumptions.

Assumption 1: The functions f_x and f_z are strictly convex, separable functions, in that they are of the form (3), where the components have continuous second derivatives such that

$$A \leq f''_{x_j}(x_j) \leq B \;\forall x_j, \quad A \leq f''_{z_i}(z_i) \leq B \;\forall z_i, \tag{20}$$

for some $0 < A \leq B < \infty$.

Under this assumption, we have the following convergence result. A slightly more general result that applies to a larger class of estimation functions g_x and g_z is given in the full paper [18].

Theorem 1: Consider Algorithm 2 with only ADMM updates (i.e., $\theta^t = 0$ for all t), so that the linearization terms remain constant, (i.e., $\boldsymbol{\tau}_p^t = \boldsymbol{\tau}_p, \boldsymbol{\tau}_r^t = \boldsymbol{\tau}_r$ for all t for some vectors $\boldsymbol{\tau}_p$ and $\boldsymbol{\tau}_r$). Then, under Assumption 1, the algorithm with the MMSE estimation functions (19) converges to a unique fixed point at a linear rate of convergence.

B. Outer Loop Convergence

We next consider the convergence of the outer loop, Algorithm 1, assuming that the inner minimization (i.e., line 5 of Algorithm 1) is computed exactly.

Theorem 2: Suppose that the functions f_x and f_z satisfy Assumption 1 and the matrix $\mathbf{S}$ has positive components (i.e., $S_{ij} = |A_{ij}|^2 > 0 \;\forall ij$). Then, there exists a $\overline{\theta}$ such that, if $\theta^k < \overline{\theta}$, the sequence of belief estimates (b_x^k, b_z^k) generated by Algorithm 1 yields a monotonically non-increasing LSL-BFE, i.e., $J(b_x^{k+1}, b_z^{k+1}) \leq J(b_x^k, b_z^k)$.

Together, Theorems 1 and 2 demonstrate that ADMM-GAMP will converge under sufficient damping. Specifically, suppose that iterations $t_1 < t_2 < \cdots$ are infinitely far apart. Then, for all t between each t_k and t_{k+1}, set $\theta^t = 0$ so that the ADMM inner-loop is run to completion and, at each $t = t_k$, select θ^t to be a sufficiently small positive value. It is of course impossible to use an infinite number of inner-loop iterations in practice. Fortunately, our numerical experiments in Section IV suggest that a fixed number of inner-loop iterations is sufficient.

IV. Numerical Experiments

We illustrate the performance of ADMM-GAMP by considering three numerical experiments. While our theoretical results assumed strictly convex penalties, we numerically demonstrate the stability of ADMM-GAMP for the non-convex penalty corresponding to a Bernoulli-Gaussian prior on $\mathbf{x}$, i.e.,

$$e^{-f_{x_i}(x_i)} = (1-\rho)\delta(x_i) + \rho\mathcal{N}(x_i; 0, 1), \tag{21}$$

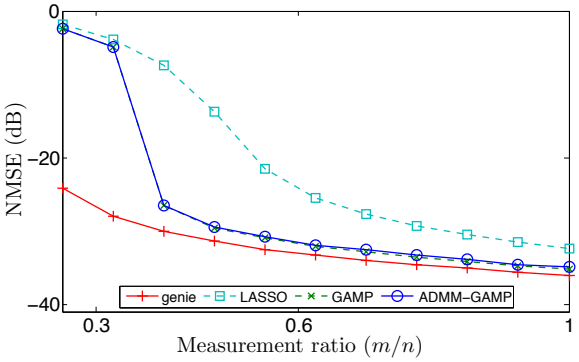

Fig. 2. Average NMSE versus measurement rate m/n when recovering a length $n = 1000$ Bernoulli-Gaussian signal $\mathbf{x}$ from AWGN-corrupted measurements $\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{e}$ under i.i.d. $\mathbf{A}$.

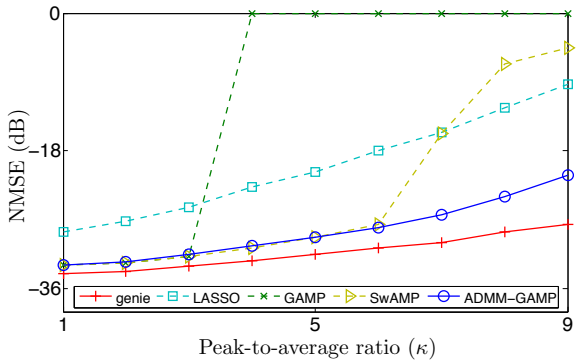

Fig. 3. Average NMSE versus peak-to-average squared-singular-value ratio $\kappa(\mathbf{A})$ when recovering a length $n = 1000$ Bernoulli-Gaussian signal $\mathbf{x}$ from $m = 600$ AWGN-corrupted measurements $\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{e}$. Note the superior performance of ADMM-GAMP relative to both the original GAMP and SwAMP, and the proximity of ADMM-GAMP to the support-aware genie.

where $\rho \in (0, 1]$ is the sparsity ratio and δ is the Dirac delta distribution. In our experiments, we fix the parameters to $n = 1000$ and $\rho = 0.2$, and we numerically compare the normalized MSE

$$\text{NMSE (dB)} \triangleq 10\log_{10}\left(\frac{\|\mathbf{x} - \widehat{\mathbf{x}}\|_2^2}{\|\mathbf{x}\|_2^2}\right)$$

of ADMM-GAMP to four other recovery schemes: the original GAMP method [4]; de-biased LASSO [21]; swept AMP (SwAMP) [9]; and the support-aware MMSE estimator, labeled "genie" – see [18] for details.

The first experiment considers recovering sparse $\mathbf{x}$ from $\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{e}$, where $\mathbf{e}$ is AWGN with variance set to achieve an SNR of 30 dB, and where the measurement matrix $\mathbf{A}$ is drawn with i.i.d. $\mathcal{N}(0, 1/m)$ entries. Figure 2 shows the NMSE performance of the algorithms under test after averaging the results of 100 Monte Carlo trials. The case of i.i.d. $\mathbf{A}$ is the "ideal" scenario for both AMP and GAMP, where the convergence can be rigorously guaranteed [4]–[6] as $m, n \to \infty$. In Figure 2, since m and n are sufficiently

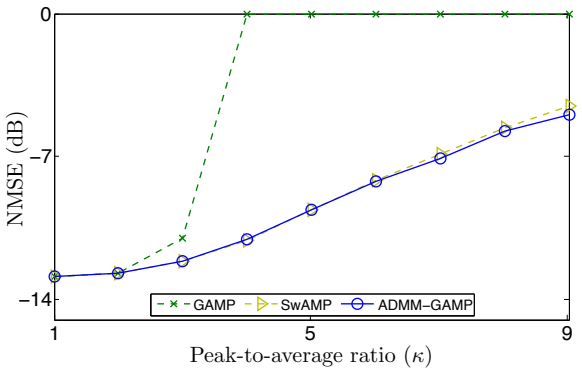

Fig. 4. Average NMSE versus peak-to-average squared-singular-value ratio $\kappa(\mathbf{A})$ when recovering a length $n = 1000$ Bernoulli-Gaussian signal $\mathbf{x}$ from $m = 2000$ noiseless 1-bit measurements $\mathbf{y} = \mathrm{sgn}(\mathbf{Ax})$. Note the superior performance of ADMM-GAMP relative to the original GAMP and SwAMP.

large, it is not surprising to see that GAMP performs well over all measurement ratios m/n.

The benefits of ADMM-GAMP become apparent in our second experiment, which uses non-i.i.d. matrices $\mathbf{A}$. We first recall that [7] established that the convergence of GAMP can be predicted by the peak-to-average ratio of the squared singular values,

$$\kappa(\mathbf{A}) \triangleq \frac{\sigma_1^2(\mathbf{A})}{\sum_{i=1}^{r} \sigma_i^2(\mathbf{A})/r}, \tag{22}$$

where $r = \min\{m, n\}$ and $\sigma_i(\mathbf{A})$ is the i-th largest singular value of $\mathbf{A}$. When this ratio κ is sufficiently large, the algorithm will diverge. Thus, to test the robustness of ADMM-GAMP, we constructed a sequence of matrices $\mathbf{A}$ with varying κ – see [18] for details. As a function of κ, the NMSE performance of the various algorithms under test is illustrated in Figure 3 for the case of $m = 600$ measurements. There it can be seen that, for larger values of κ, the NMSE performance of the original GAMP algorithm deteriorated, which was a result of the algorithm diverging. The ADMM-GAMP method, however, converged over the entire range of κ values, achieving NMSE performance relatively close to the support-aware genie. Fig. 4 repeats the experiment for a "one-bit" measurement output $\mathbf{y} = \mathrm{sgn}(\mathbf{Ax})$, where sgn is the *sign function*, as considered in, e.g., [22] and [23] – again, see [18] for details. We again see that ADMM-GAMP being stable over a wide range of values of κ.

Conclusions

A major stumbling block to more widespread use of AMP methods is their convergence and numerical stability. While several methods have been proposed to improve the convergence, this paper provides a method with provable guarantees under arbitrary transforms. Nevertheless, there is still much work to be done. Most obviously, the proposed ADMM-GAMP method comes with higher computational cost and our simulations indicate that other methods can be equally effective. One line of future work would thus be see to whether the proof techniques in this paper can be extended to these methods as well as other variants of GAMP.

References

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

[2] ——, "Message passing algorithms for compressed sensing I: motivation and construction," in *Proc. Info. Theory Workshop*, Jan. 2010.

[3] S. Rangan, "Estimation with random linear mixing, belief propagation and compressed sensing," in *Proc. Conf. on Inform. Sci. & Sys.*, Princeton, NJ, Mar. 2010, pp. 1–6.

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

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

[6] A. Javanmard and A. Montanari, "State evolution for general approximate message passing algorithms, with applications to spatial coupling," arXiv:1211.5164 [math.PR]., Nov. 2012.

[7] S. Rangan, P. Schniter, and A. Fletcher, "On the convergence of approximate message passing with arbitrary matrices," in *Proc. ISIT*, Jul. 2014, pp. 236–240.

[8] F. Caltagirone, L. Zdeborová, and F. Krzakala, "On convergence of approximate message passing," in *Proc. ISIT*, Jul. 2014, pp. 1812–1816.

[9] A. Manoel, F. Krzakala, E. W. Tramel, and L. Zdeborová, "Sparse estimation with the swept approximated message-passing algorithm," *arXiv:1406.4311*, Jun. 2014.

[10] J. Vila, P. Schniter, S. Rangan, F. Krzakala, and L. Zdeborová, "Adaptive damping and mean removal for the generalized approximate message passing algorithm," in *Proc. IEEE ICASSP*, 2015, to appear.

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

[12] P. Schniter and S. Rangan, "Compressive phase retrieval via generalized approximate message passing," *IEEE Trans. Signal Process.*, vol. 63, no. 4, pp. 1043–1055, 2015.

[13] J. Ziniel, P. Schniter, and P. Sederberg, "Binary linear classification and feature selection via generalized approximate message passing," *IEEE Trans. Signal Process.*, vol. 63, no. 8, pp. 2020–2032, 2015.

[14] S. Rangan, P. Schniter, E. Riegler, A. Fletcher, and V. Cevher, "Fixed points of generalized approximate message passing with arbitrary matrices," in *Proc. ISIT*, Jul. 2013, pp. 664–668.

[15] F. Krzakala, A. Manoel, E. W. Tramel, and L. Zdeborová, "Variational free energies for compressed sensing," in *Proc. ISIT*, Jul. 2014, pp. 1499–1503.

[16] A. L. Yuille and A. Rangarajan, "The concave-convex procedure (CCCP)," *Proc. NIPS*, vol. 2, pp. 1033–1040, 2002.

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

[18] S. Rangan, A. K. Fletcher, P. Schniter, and U. S. Kamilov, "Inference for generalized linear models via alternating directions and Bethe free energy minimization," *arXiv:1501.01797*, Jan. 2015.

[19] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2010.

[20] R. T. Rockafellar, "Monotropic programming: Descent algorithms and duality," in *Nonlinear Programming*, O. L. Mangasarian, R. R. Meyer, and S. M. Robinson, Eds. Academic Press, 1981, vol. 4, pp. 327–366.

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

[22] P. T. Boufounos and R. G. Baraniuk, "1-bit compressive sensing," in *Proc. Conf. on Inform. Sci. & Sys.*, 2008, pp. 16–21.

[23] U. S. Kamilov, V. K. Goyal, and S. Rangan, "Message-passing dequantization with applications to compressed sensing," *IEEE Trans. Signal Process.*, vol. 60, no. 12, pp. 6270–6281, Dec. 2012.