Local Convergence of an AMP Variant to the LASSO Solution in Finite Dimensions

Yanting Ma, 1 Min Kang, 2 Jack W. Silverstein, 2 and Dror Baron 3

Abstract—A common sparse linear regression formulation is the $\ell_1$ regularized least squares, which is also known as least absolute shrinkage and selection operator (LASSO). Approximate message passing (AMP) has been proved to asymptotically achieve the LASSO solution when the regression matrix has independent and identically distributed (i.i.d) Gaussian entries in the sense that the averaged per-coordinate $\ell_2$ distance between the AMP iterates and the LASSO solution vanishes as the signal dimension goes to infinity before the iteration number. However, in finite dimensional settings, characterization of AMP iterates in the limit of large iteration number has not been established. In this work, we propose an AMP variant by including a parameter that depends on the largest singular value of the regression matrix. The proposed algorithm can also be considered as a primal dual hybrid gradient algorithm with adaptive stepsizes. We show that whenever the AMP variant converges, it converges to the LASSO solution for arbitrary finite dimensional regression matrices. Moreover, we show that the AMP variant is locally stable around the LASSO solution under the condition that the LASSO solution is unique and that the regression matrix is drawn from a continuous distribution. Our local stability result implies that in the special case where the regression matrix is large and has i.i.d. random entries, the original AMP, which is a special case of the proposed AMP variant, is locally stable around the LASSO solution.

I. INTRODUCTION

Least absolute shrinkage and selection operator (LASSO) is a common formulation for sparse linear regression, which is defined as the optimization problem:

$$\mathbf{x}^* \in \arg\min \{ F(\mathbf{x}) : \ell_2(\mathbf{y} - \mathbf{A}\mathbf{x})^2 + \gamma \|\mathbf{x}\|_1 \}, \quad (1)$$

where $\mathbf{A} \in \mathbb{R}^{n \times N}$ is the regression matrix, $\mathbf{y} \in \mathbb{R}^n$ is the data vector, $\gamma > 0$ is the regularization parameter, and $\| \cdot \|_p$, for $p = 1, 2$, denotes the $\ell_p$ norm. Applications of LASSO include model selection and image processing, since natural images are usually sparse in some transform domain. While numerous standard convex optimization algorithms such as the class of proximal gradient methods [1]–[3], alternating direction method of multipliers (ADMM) [4], and primal dual hybrid gradient (PDHG) [5]–[7] can be used to solve (1), it is also of both theoretical and practical interest to study approximate message passing (AMP) for solving (1), since AMP was initially introduced by Donoho et al. [8] as a LASSO solver and it usually enjoys fast empirical convergence when it converges.

Existing theoretical convergence analysis of standard optimization algorithms and that of AMP are considered in different problem settings. Specifically, the quantity of optimization algorithms is usually $\lim_{t \to \infty} \|\mathbf{x}^t - \mathbf{x}^*\|_2$, where $\mathbf{x}^t$ is the estimate at the $t^{th}$ iteration of an iterative algorithm, for any fixed and finite $n$ and $N$. It is usually assumed in the AMP framework that the data vector $\mathbf{y}$ is generated according to a linear system, $\mathbf{y} = \mathbf{A}\mathbf{x}_0 + \mathbf{w}$, with some underlying ground-truth $\mathbf{x}_0$ and some noise $\mathbf{w}$. Under these assumptions, the analysis of AMP shows that when $\mathbf{A}$ has independent and identically distributed (i.i.d) Gaussian entries, the quantity $\lim_{N \to \infty} \frac{1}{N} \|\mathbf{x}^t - \mathbf{x}_0\|_2$ with $\frac{t}{N} \to c \in (0, \infty)$ converges to a deterministic number predicted by a scalar recursion referred to as state evolution with probability one [9]; this is later extended to a large deviation result [10]. For the LASSO problem, Bayati and Montanari [11] have proven the convergence of AMP iterates to the LASSO solution $\mathbf{x}^*$ in the sense that $\lim_{t \to \infty} \lim_{N \to \infty} \frac{1}{N} \|\mathbf{x}^t - \mathbf{x}^*\|^2 = 0$ with probability one, which has also been extended to a large deviation result in recent work [12]. However, this large deviation result only holds for $t = O\left(\frac{\log N}{\log \log N}\right)$. Therefore, the convergence of AMP for finite $N$ as $t \to \infty$ is still unknown. In fact, using matrices with i.i.d. Gaussian entries and following the calibration method proposed in [11] for choosing the threshold of the soft-thresholding function at each AMP iteration, we performed 2000 trials of Monte Carlo simulations with $N = 2000, n = 1000$, and AMP never converged to the LASSO solution in terms of $\ell_2$ error.

The connection between AMP and standard convex optimization algorithms has enabled the design of AMP variants that have convergence guarantees for more practical settings such as the ones with non-Gaussian finite dimensional matrices. Most such results have been developed in a more general algorithmic framework known as generalized AMP (GAMP) [13]. In the context of solving optimization problems, GAMP considers objective functions of the form $\sum_{i=1}^n g((\mathbf{A}\mathbf{x})_i) + \sum_{i=1}^n f(x_i)$. Consider now that $\mathbf{A}$ is arbitrary and finite dimensional. When both $g$ and $f$ are quadratic

\[ \text{The big O notation } O(\cdot) \text{ means that there exists an } N_0 \in \mathbb{N} \text{ and a positive real number } B \text{ such that } t(N) \leq B \frac{\log N}{\log \log N} \text{ for all } N \geq N_0. \]
functions, the damped GAMP [14], which defines the current iterate as a convex combination of the current estimate and the iterate from the previous iteration, has global convergence guarantees. When \( g \) and \( f \) are strictly convex and twice continuously differentiable, assuming that the derivatives of the nonlinear functions used in each GAMP iteration are bounded within the open interval \((0, 1)\), the damped GAMP with fixed stepsize [14] is proved to be locally stable around the equilibrium point, and the ADMM-GAMP [15], which combines an ADMM inner loop within each GAMP iteration, is guaranteed to achieve global convergence.

The interpretation of AMP as a PDHG algorithm was first mentioned by Rangan et al. [14], which has inspired the current work. The differences between our work and the local stability analysis in [14] are as follows:

1) The objective function of LASSO is non-differentiable, hence is not covered in [14].
2) Instead of damping the iterates while keeping the stepsizes fixed as in [14], we do not make changes to the updates of the iterates but design a stepsize updating schedule based on AMP.
3) The result in [14] holds only when the stepsize is fixed over all iterations, which loses the main advantage of AMP over standard optimization algorithms for fast convergence, whereas our result allows keeping the structure of the stepsize updating schedule of AMP. As we will see in Section IV, numerical results show that the nonlinear functions used in each GAMP iteration are continuously differentiable, assuming that the derivatives of the PDHG solves (4) by alternating between the estimation of \( s \) and \( x \) as \( s^{t+1} = \arg\max_{s \in \mathbb{R}^n} \left\{ F(s, x^t) + \frac{1}{2\tau_s} \| s-s^t \|_2^2 \right\} \) and \( x^{t+1} = \arg\min_{x \in \mathbb{R}^n} \left\{ F(s^{t+1}, x) + \frac{1}{2\tau_x} \| x-x^t \|_2^2 \right\} \), respectively, which is equivalent to
\[
\begin{align*}
\tau_s^{t+1} & = \arg\max_{\tau_s} \left\{ g^*(s^t) + \frac{1}{2\tau_s} \| s - (s^t + \tau_s A x^t) \|_2^2 \right\}, \\
\tau_x^{t+1} & = \arg\min_{x \in \mathbb{R}^n} \left\{ f(x) + \frac{1}{2\tau_x} \| x - (x^t - \tau_x^t A^T s^{t+1}) \|_2^2 \right\}.
\end{align*}
\]

In the above, the stepizes \( \tau_s^t \) and \( \tau_x^t \) can stay constant for all iterations or be updated at every iteration. One feature of PDHG algorithms is that each equilibrium point of the algorithm is a saddle-point of (4). This can be explained as follows. Let \((\tilde{x}, \tilde{s}, \tau_x, \tau_s)\) be an equilibrium point of the algorithm (6), then
\[
0 \in \partial_s \left( g^*(s) + \frac{1}{2\tau_s} \| s - (\tilde{s} + \tau_s A \tilde{x}) \|_2^2 \right) \big|_{s=\tilde{s}},
\]
\[
0 \in \partial_x \left( f(x) + \frac{1}{2\tau_x} \| x - (\tilde{x} - \tau_x A^T \tilde{s}) \|_2^2 \right) \big|_{x=\tilde{x}},
\]
where \( \partial_s \) denotes sub-differential with respect to \( s \). The above implies that \( A \tilde{x} \in \partial g^*(\tilde{s}) \) and \( -A^T \tilde{s} \in \partial f(\tilde{x}) \), which is the necessary and sufficient condition for \((\tilde{x}, \tilde{s})\) to be a saddle-point of (4).

The choice of stepizes is crucial for the convergence of an optimization algorithm. AMP can be interpreted as a special case of PDHG with an adaptive stepsize updating schedule [14]. Specifically, let
\[
\tau_s^t = \frac{1}{\tau_s^t - 1}, \quad \tau_x^{t+1} = 1 + \frac{d^t}{c} \tau_x^t,
\]
where \( d^t = \| x^{t+1} \|_0/N \) with \( \| x^{t+1} \|_0 \) (the \( \ell_0 \) quasi-norm) denoting the number of nonzero coordinates of \( x^{t+1} \). By (2) and (3), we have \( g^*(s) = (y,s) + \frac{1}{2} \| s \|_2^2 \). For easy comparison, we use the same notation for the soft-thresholding function (proximal operator for the \( \ell_1 \)-norm) as in [11]. For any \( \theta > 0 \), \( u \in \mathbb{R}^N \), define
\[
\eta(u; \theta) := \arg\min_{x \in \mathbb{R}^N} \| x \|_1 + \frac{1}{2\theta} \| x - u \|_2^2.
\]
Then (6) can be written as
\[
\begin{align*}
\tau_s^{t+1} & = \frac{1}{\tau_x^t} (A^T x^t - y) + \left( 1 - \frac{1}{\tau_x^t} \right) s^t, \\
x^{t+1} & = \eta \left( x^t - \tau_x^t A^T s^{t+1}; \gamma \tau_x^t \right).
\end{align*}
\]
Let \( z^t := -\tau_s^t s^{t+1} \) for all \( t \geq 1 \) and notice from (7) that \( (\tau_x - 1)/\tau_x^{-1} = d^t - 1/c \). Then we can see that (9) matches the AMP algorithm (see [8] and [11]), but with a different choice for the threshold of the soft-thresholding function. We emphasize that the choice of the threshold in [11] does not guarantee that AMP will converge to the LASSO solution.
for finite dimensional problems, whereas the choice in (9) guarantees that whenever (9) converges, it converges to the LASSO solution for arbitrary finite dimensional $A$.

We notice that in many optimization algorithms, the stepsize usually depends on the largest singular value of $A$, whereas the algorithm defined in (6) with $\tau_s^t$ and $\tau_x^t$ being updated according to (7), which is equivalent to AMP (9), does not depend on the largest singular value of $A$. Therefore, in order to have an algorithm that is more robust than AMP with arbitrary finite dimensional $A$ while retaining the fast convergence feature of AMP, we introduce a parameter $e$ to (7) that depends on the largest singular value of $A$ and still keep the general structure of the updating schedule in (7).

Specifically, choosing $0 < e < \min\{1, 4/(\sigma^2_{\max}(A) + 2)\}$, where $\sigma_{\max}(A)$ is the largest singular value of $A$, we modify (7) as

$$\tau_s^t = \frac{e}{\tau_x^t - e}, \quad \tau_x^t = 1 + \frac{d^t}{c} \tau_x^t. \quad (10)$$

Note that such a choice of $e$ ensures local stability of our proposed algorithm; details about the local stability analysis will be discussed in Section III. Our proposed AMP variant is (6) with the stepsize updating schedule defined in (10). Similar to the derivation of (9) from (6) and (7), we can write the proposed AMP variant as in Algorithm 1.

Algorithm 1 Proposed AMP Variant

**Input:** $A$, $y$, $0 < e < \min\{1, 4/(\sigma_{\max}^2(A) + 2)\}$, $t_{\max}$

**Initialization:** $x^0$, $s^0$, $x_0^t$

for $0 \leq t \leq t_{\max}$ do

$$s^{t+1} = \frac{e}{\tau_x^t} (Ax^t - y) + \left(1 - \frac{e}{\tau_x^t}\right)s^t$$

$$x^{t+1} = \eta \left( x^t - \tau_x^t A^T s^{t+1}; \gamma \tau_x^t \right)$$

$$\tau_x^{t+1} = 1 + \frac{d^t}{c} \tau_x^t \quad \text{with} \quad d^t = \|x^{t+1}\|_0/N$$

end for

**Output:** $x^{t_{\max}}$

III. LOCAL STABILITY ANALYSIS

In this section, we study the local stability of Algorithm 1 around its equilibrium point, which is the LASSO solution. We first discuss conditions under which our analysis is valid, and then show that Algorithm 1 is locally stable under these conditions.

A. Assumptions

When Null($A$), the null space of $A$, contains nonzero components, the objective function is not strictly convex in $x$, hence there may be multiple solutions. Conditions for the uniqueness of the LASSO solution have been studied by Tibshirani [17], which states that a sufficient [17, Lemma 2] and necessary [17, Lemma 6] condition for (1) to admit a unique solution is that Null($C_K(A)$) = $\{0\}$, where $K := \{i \in \{1, \ldots, N\} | x_i^* \neq 0\}$ with $x_i^*$ the $i^{th}$ coordinate of $x^*$, and $C_K(A)$ is the submatrix of $A$ formed by deleting the $i^{th}$ column of $A$ for all $i \notin K$. Notice that the necessary condition implies that when the LASSO solution is unique, we have that $|K| \leq \min\{n, N\}$; this is a condition that we need to prove our result. A more explicit sufficient condition for uniqueness in the almost sure sense is also provided in [17, Lemma 4]: if entries of $A$ are drawn from a continuous probability distribution on $\mathbb{R}^{n \times N}$, then the LASSO solution is unique with probability one regardless of the dimension of $A$.

Note that with $\eta(\cdot)$ being the soft-thresholding function (8), the definition of $d^t$ can be written as

$$d^t = \{ i \in \{1, \ldots, N\} : |x^t - \tau_x^t A^T s^{t+1}|_i > \gamma \tau_x^t \},$$

where we can further replace $s^{t+1}$ by $\frac{e}{\tau_x^t} (Ax^t - y) + \left(1 - \frac{e}{\tau_x^t}\right)s^t$, so that $d^t$ only depends on the iterates at the $t^{th}$ iteration. Similarly, the update of $x^{t+1}$ in (11) can also be written as a function of iterates at the $t^{th}$ iteration only. Therefore, letting $v^t \in \mathbb{R}^{n \times N+1}$ be defined as $v^t := \{s^t; x^t; \tau_x^t\}$ for all $t \geq 0$, we have that (11) defines a nonlinear operator $G : \mathbb{R}^{n \times N+1} \rightarrow \mathbb{R}^{n \times N+1}$ such that $v^{t+1} = G(v^t)$. Note that $G$ is differentiable at $\{s^t; x^t; \tau_x^t\}$ except when there exists an $i \in \{1, \ldots, N\}$, such that

$$|x^t - A^T (e(Ax^t - y) + (\tau_x^t - e)s^t)|_i = \pm \gamma \tau_x^t,$$

which has probability zero if $A$ obeys a continuous distribution.

To summarize, we make the following two assumptions on components in (1):

1) The matrix $A$ is drawn from a continuous probability distribution on $\mathbb{R}^{n \times N}$.
2) The regularization parameter $\gamma > 0$.

B. Stability around the Equilibrium Point

Having clarified our assumptions, we now prove our main result, which is the local stability guarantee of Algorithm 1 as stated in the following proposition.

**Proposition 1.** Consider the LASSO problem defined in (1) and suppose that the conditions in Section III-A are satisfied. Then Algorithm 1 is stable around its equilibrium point with probability one.

**Proof.** Suppose that $G$ is differentiable around the equilibrium point $\tilde{v}$. Then the local stability of $G$ around $\tilde{v}$ is determined by the largest eigenvalue (in modulus) of the Jacobian matrix $J$ of $G$ evaluated at $\tilde{v}$. The expression for $J$ is

$$J = \begin{bmatrix}
(1 - \frac{e}{\tau_x^t}) I_n & \frac{e}{\tau_x^t} A \\
\frac{e}{\tau_x^t} D A^T & -(\frac{e}{\tau_x^t} D A^T) \\
0_{1 \times n} & 0_{1 \times N}
\end{bmatrix},$$

where $D$ is a diagonal matrix defined as $D_{ii} = \mathbb{I}(|x_i^*| \neq 0)$ with $\mathbb{I}$ being the indicator function, $I_n$ is the $n \times n$ identity matrix, and $0_{n \times m}$ is an $n \times m$ matrix with all zero-valued coordinates. In what follows, we show that with an appropriate choice of $e$, the eigenvalue of $J$ with the largest modulus is within the unit circle of the complex plane.
Let $\alpha = 1 - e/\tau_x$ and let $|A|$ denote the determinant of a matrix $A$. Then

$$
|J - \lambda I| = \left| \begin{array}{cc}
(\alpha - \lambda)I & (e/\tau_x)A \\
(e - \tau_x)DA & \widehat{D} - \lambda I_N - e\bar{D}A^TA
\end{array} \right|
$$

where the last inequality holds because $e \in (0, 1]$ and $\tau_x > 1$, hence $1 - e/\tau_x > 0$. Now that $\lambda \neq 0$, we divide both sides of (15) by $(e\lambda)^{\hat{N} \hat{d}}$.

$$
|\frac{(\alpha - \lambda)(1 - \lambda)}{e\lambda}|I_{N\hat{d}} + H = 0.
$$

(16)

Therefore, $\lambda$ is a solution to (15) if and only if $-\frac{(\alpha - \lambda)(1 - \lambda)}{e\lambda}$ is an eigenvalue of $H$. Let $\text{sp}(H)$ denote the spectrum (i.e., set of eigenvalues) of $H$, and define

$$
m := \min_{\lambda \in \text{sp}(H)} \lambda, \quad M := \max_{\lambda \in \text{sp}(H)} \lambda.
$$

(17)

Note that $H = [A^T A]_\alpha = (C_K(A))^T C_K(A)$. By condition of the uniqueness of LASSO solution, we have that $C_K(A)$ is non-singular, hence $m > 0$. Let $u = -\frac{(\alpha - \lambda)(1 - \lambda)}{e\lambda}$, then

$$
\lambda^2 + (eu - \alpha - 1)\lambda + \alpha = 0.
$$

(18)

Let $b := eu - \alpha - 1$, and we solve (18) for $\lambda$. When $|b| < 2\sqrt{\alpha}$, we have complex roots $\lambda = \frac{1}{2}(-b \pm i\sqrt{4\alpha - b^2})$, hence $|\lambda| = \frac{1}{2}\sqrt{4\alpha - b^2} < \sqrt{\alpha} < 1$. When $|b| \geq 2\sqrt{\alpha}$, we have real roots. Define

$$
h_1(b) := \frac{1}{2}(-b + \sqrt{b^2 - 4\alpha}), \quad h_2(b) := \frac{1}{2}(-b - \sqrt{b^2 - 4\alpha}).
$$

(19)

Notice that

$$
h'_1(b) = \frac{1}{2} \begin{cases} -1 + \frac{\sqrt{b^2 - 4\alpha}}{b} & \text{if } b > 2\sqrt{\alpha} \\ -1 + \frac{\sqrt{b^2 - 4\alpha}}{b} & \text{if } b < -2\sqrt{\alpha} \end{cases}
$$

$$
h'_2(b) = \frac{1}{2} \begin{cases} -1 + \frac{\sqrt{b^2 - 4\alpha}}{b} & \text{if } b > 2\sqrt{\alpha} \\ -1 + \frac{\sqrt{b^2 - 4\alpha}}{b} & \text{if } b < -2\sqrt{\alpha} \end{cases}
$$

(20)

Also notice that $h_1(b) \uparrow 0$ as $b \uparrow \infty$, and $h_2(b) \downarrow 0$ as $b \downarrow -\infty$. The graph of $h_1$ and $h_2$ as a function of $b$, respectively, is shown in Fig. 1. It can be seen that

$$
\max(|h_1(b)|, |h_2(b)|) = \begin{cases} h_1(b), & \text{if } b \leq -2\sqrt{\alpha} \\ -h_2(b), & \text{if } b \geq 2\sqrt{\alpha} \end{cases}
$$

First consider $b \leq -2\sqrt{\alpha}$. Recall that $b = eu - \alpha - 1 > em - \alpha - 1 > -\alpha - 1$, since $e > 0$ and $m > 0$. Notice from (19) and (20) that $h_1(-\alpha - 1) = 1$, $h_1(-2\sqrt{\alpha}) = -\sqrt{\alpha} < 1$, and that $h_1(b)$ is monotone decreasing when $b \leq -2\sqrt{\alpha}$. Therefore, we have that $|h_1(b)| < 1$, when $b \leq -2\sqrt{\alpha}$.

Next consider $b \geq 2\sqrt{\alpha}$. Notice that $h_2(1 + \alpha) = -1$ and that $h_2(b)$ is monotone decreasing when $b \geq 2\sqrt{\alpha}$. Therefore, in order to have $|h_2(b)| < 1$, we need $b = eu - \alpha - 1 < 1 + \alpha, \forall u \in \text{sp}(H)$. This condition is satisfied if we let $e < \frac{2(1 + \alpha)}{M}$. Recall that $\alpha = 1 - e/\tau_x$. By (11), we have $1/\tau_x = 1 - d/c$. Combining the analysis on $h_1$ and $h_2$ given above, it follows that the condition on the parameter $e$ for all eigenvalues of $J$ to be within the unit circle of the complex plane is

$$
0 < e < \min \left\{ 1, \frac{4}{M + 2(1 - d/c)} \right\}.
$$

(21)
The upper-bound for \( e \) in (21) is tight. However, it depends on the equilibrium point. It is desirable to have a condition on \( e \) that does not depend on knowledge about the equilibrium point, so that an appropriate value for \( e \) can be chosen before running the algorithm (11). Notice that \( H = \{A^TA\}_{K} \) is a principal submatrix of the symmetric matrix \( A^TA \). The interlacing property of eigenvalues implies that \( M \leq L := \max_{\lambda \in sp(A^TA)} \lambda \). Moreover, the condition for the uniqueness of the LASSO solution implies that \( \tilde{d}/c \in [0,1] \). Therefore, local stability is guaranteed if

\[
0 < e \leq \min\{1, 4/(L + 2)\}. \tag{22}
\]

C. Random Matrices with i.i.d. Entries

In the special case where the matrix \( A \in \mathbb{R}^{n \times N} \) is the upper left corner of a doubly infinite array\(^2\) of i.i.d. zero-mean random variables with finite fourth moment and is normalized such that the variance is 1/n, the asymptotic largest singular value of \( C_K(A) \in \mathbb{R}^{n \times Nd} \), where \( Nd < n \), is

\[
1 + \sqrt{Nd/n} = 1 + \sqrt{d/c} \text{ with probability one} \quad [18].
\]

It follows that the denominator of the upper-bound in (21) is

\[
M + 2(1 - \tilde{d}/c) = 1 + \tilde{d}/c + 2\sqrt{\tilde{d}/c} - 2\tilde{d}/c = 3 + 2\sqrt{d/c} - \tilde{d}/c.
\]

Let \( x = \tilde{d}/c \), hence \( x \in [0,1] \). Define \( f(x) = 2\sqrt{x} - x \) and notice that \( f \) is monotone increasing on \( [0,1] \). Therefore, \( f(x) < f(1) = 1 \), which implies that \( M + 2(1 - \tilde{d}/c) \leq 4 \).

That is, local stability for large zero-mean random matrices with variance 1/n is guaranteed by setting \( e = 1 \), which, as mentioned before, makes Algorithm 1 coincide with the original AMP (9), as seen in [8] and [11].

To demonstrate the efficiency of the proposed AMP variant, we compare it with the original AMP that uses the calibration method proposed in [11], a PDHG algorithm with a fixed stepsize that guarantees convergence (see [7]), and a popular convex optimization algorithm, fast iterative shrinkage and thresholding algorithm (FISTA) [3]. Because our proposed algorithm is inspired by AMP and depends on a parameter \( e \), we will refer to it as “eAMP” and we choose \( e \) according to (22). In addition, we also include results for eAMP with \( e = 1 \), which we recall is of the same form as the standard AMP but the threshold for the thresholding function at each iteration is different from that in [11].

In all the simulations, the problem dimension is set to be \( N = 2000 \), \( n = 1000 \). The data vector \( y \) is obtained by \( y = Ax_0 + w \), where entries of \( w \) are independent realizations of a Gaussian distribution with mean zero and variance \( \sigma_w^2 \) and entries of \( x_0 \) are independent realizations of a Bernoulli(0.1)-Uniform[-1,1] distribution (i.e., \( X_0 = BU \) with \( B \sim \text{Bernoulli}(0.1) \) and \( U \sim \text{Uniform}[-1,1] \)). The value of \( \sigma_w^2 \) is chosen such that 10 log\(_10\) \( \frac{\|Ax_0\|_{\infty}}{\sqrt{n\sigma_w^2}} \) = 25 (i.e., the signal-to-noise ratio is 25dB). All tested algorithms are initialized with an all-zero vector. Since FISTA and PDHG have theoretical convergence guarantees, we present their results only for comparison of empirical convergence speed, hence we sometimes stop these two algorithms early when the convergence speed comparison is clear.

For the first set of simulations, we use matrices \( A \) whose entries are independent realizations of a zero-mean Gaussian distribution, which is the case studied in [11] in the limit as \( N, n \to \infty \). The simulation results are shown in Fig. 2a. We notice that while AMP seems to have converged, it does not converge to the LASSO solution \( x^* \), whereas eAMP with both choices of \( e \) has converged to the LASSO solution. Moreover, the empirical convergence speed (in terms of number of iterations) of eAMP is much faster than that of FISTA or PDHG.

For the second set of simulations, we use matrices \( A \) whose rows are independent realizations of a zero-mean multivariate Gaussian distribution, where diagonal entries of the covariance matrix have value 1/n and off-diagonal entries have value 0.01/n. The simulation results are shown in Fig. 2b. In this case, AMP and eAMP with the inappropriate choice of \( e = 1 \) have diverged, whereas eAMP with \( e \) chosen according to (22) has converged to the LASSO solution and requires far fewer iterations than FISTA and PDHG.

While our analysis in Section III only guarantees local stability for eAMP, the encouraging simulation results suggest that a global convergence result might be possible; we leave the global convergence analysis for future work.

V. Conclusion

In this paper, we proposed an AMP variant (Algorithm 1) for solving the LASSO problem (1). Unlike the work in [11] that analyzes the limiting behavior of AMP iterates as...
the iteration number goes to infinity for infinite dimensional problems, we focused on more practical finite dimensional problems. Specifically, for any finite dimensional matrix $A$, whenever our algorithm converges, it converges to the LASSO solution. We emphasize that this is not the case for AMP with $\epsilon$, whenever our algorithm converges, it converges to the LASSO solution with probability one when entries of $A$ under which the algorithm is locally stable around the PDHG [7].

Fig. 2: Comparison of different algorithms for solving the LASSO problem (1).

(a) Entries of the matrix $A$ are drawn independently from a zero-mean Gaussian distribution.

(b) Rows of the matrix $A$ are drawn independently from a zero-mean multivariate Gaussian distribution.

the iteration number goes to infinity for infinite dimensional problems, we focused on more practical finite dimensional problems. Specifically, for any finite dimensional matrix $A$, whenever our algorithm converges, it converges to the LASSO solution. We emphasize that this is not the case for AMP with finite dimensional $A$ even when $A$ has i.i.d. Gaussian entries, as we have seen in Fig. 2a. The proposed algorithm contains a parameter $\epsilon$ that depends on the largest singular value of the matrix $A$. In Proposition 1, we provided conditions on $\epsilon$ under which the algorithm is locally stable around the LASSO solution with probability one when entries of $A$ are drawn from a continuous distribution. Finally, simulation results showed that the number of iterations required for our algorithm to converge is orders of magnitude smaller than that of widely used optimization algorithms such as FISTA [3] and PDHG [7].

REFERENCES