

Article

# A Two-Stage Method to Test the Robustness of the Generalized Approximate Message Passing Algorithm

Qingshan You, Yongjie Luo and Qun Wan \*

School of Electronic Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China; youlin\_2001@163.com (Q.Y.); yongjie.luo@163.com (Y.L.)

\* Correspondence: wanqun@uestc.edu.cn; Tel.: +86-28-61830358

Academic Editor: Bruno Carpentieri

Received: 11 October 2016; Accepted: 17 November 2016; Published: 22 November 2016

**Abstract:** We propose a two-stage method to test the robustness of the generalized approximate message passing algorithm (GAMP). A pursuit process based on the marginal posterior probability is inserted in the standard GAMP algorithm to find the support of a sparse vector, and a revised GAMP process is used to estimate the amplitudes of the support. The numerical experiments with simulation and real world data confirm the robustness and performance of our proposed algorithm.

**Keywords:** compressed sensing; generalized approximate message passing; pursuit; two-stage method

## 1. Introduction

Compressed sensing (CS) has been a research focus in recent years [1]. Given the support of an unknown  $K$ -sparse vector  $\mathbf{x} \in \mathbb{R}^N$  as  $S = \{j: j \in \{1, \dots, N\}, x_j \neq 0\}$ , we consider an under-determined noisy linear system

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{n} \quad (1)$$

where  $\mathbf{y} \in \mathbb{R}^M$  is a compressed sampling observation,  $\mathbf{n} \in \mathbb{R}^M$  is an additive Gaussian noise,  $\mathbf{A} \in \mathbb{R}^{M \times N}$  is the sensing matrix. The object is to reconstruct  $\mathbf{x}$  from  $\mathbf{y}$  and  $\mathbf{A}$ . There are many reconstruction algorithms to solve the CS problem. The message passing based methods, including approximate message passing (AMP) algorithm [2] and generalized approximate message passing (GAMP) algorithm [3], have remained the most efficient ever since they were proposed. They work very well in the zero-mean Gaussian sensing matrix case, but become unstable and divergent in the more general sensing matrix case. For example, although an improvement to the non-zero mean Gaussian sensing matrix case has in a sense been proposed [4], the highly columns-correlated non-Gaussian sensing matrix case is still a problem. We propose a new GAMP-like algorithm, termed Bernoulli-Gaussian Pursuit GAMP (BGP-GAMP), to raise the robustness of standard GAMP algorithm. Our algorithm firstly utilizes the marginal posterior probability of  $\mathbf{x}$  to sequentially find the support  $S$ , and then estimates the amplitudes on  $S$  by a tiny revised GAMP algorithm, termed Fixed Support GAMP (FS-GAMP) which has been proposed in our previous work [5]. A recent work has been proposed by Rodger [6], which is similar to our work, but gives another viewpoint from a neural network statistical model. The numerical experiments verify our idea and show a good performance and robustness.

The rest of this paper is organized as follows. Firstly, we review the standard GAMP, especially Bernoulli-Gaussian prior GAMP, algorithm in Section 2. Secondly, we propose the FsGAMP and BGP-GAMP algorithms in Sections 3 and 4, respectively. Thirdly, experiments are conducted in Section 5. Finally, we draw a conclusion in the last section.

## 2. GAMP Algorithm Review

The GAMP algorithm can be classified to minimum mean error estimation form (MMSE-GAMP) and maximum posterior estimation form (MAP-GAMP) [3]. We restrict our attention to MMSE-GAMP, since the problems in the real world are noisy, such that the posterior function has a very complex shape with details which depend on the observation  $\mathbf{y}$ . However, MMSE is robust to such fluctuations.

Firstly, GAMP algorithm approximates the true marginal posterior  $p(z_m|\mathbf{y})$  by the output channel marginal posterior

$$p_{z_m|p_m}(z|\hat{p}_m, v_m^p) \triangleq \frac{p_{y_m|z_m}(y_m|z)\mathcal{N}(z; \hat{p}_m, v_m^p)}{\int p_{y_m|z_m}(y_m|z)\mathcal{N}(z; \hat{p}_m, v_m^p)dz} \tag{2}$$

where  $z_m$  is the estimation of  $(\mathbf{Ax})_m$  condition on the observation  $y_m$ , and  $p_{y_m|z_m}(y_m|z)$  is the likelihood of the output channel  $y_m = z_m + w_m$ . It is worth noting that  $w_m$  means a noise, a random variable, and it corresponds to the estimated random variable  $z_m$ , instead of the original noise  $n_m$  in Equation (1). Therefore,  $w_m$  is the residual after estimation of  $(\mathbf{Ax})_m$ . The iterative algorithm computes the conditional mean and variance of  $z_m$ . We want to obtain the posterior mean and variance estimates

$$\mathbb{E}\{z_m|p_m = \hat{p}_m, v_m^p\} = \int z_m p_{z_m|p_m}(z_m|\hat{p}_m, v_m^p)dz_m \tag{3}$$

$$\text{var}\{z_m|p_m = \hat{p}_m, v_m^p\} = \int (z_m - \mathbb{E}\{z_m|p_m = \hat{p}_m, v_m^p\})^2 p_{z_m|p_m}(z_m|\hat{p}_m, v_m^p)dz_m. \tag{4}$$

Secondly, GAMP algorithm approximates the true marginal posterior  $p(x_n|\mathbf{y})$  by the input channel marginal posterior

$$p_{x_n|r_n}(x|\hat{r}_n, v_n^r) \triangleq \frac{p_{x_n}(x)\mathcal{N}(x; \hat{r}_n, v_n^r)}{\int p_{x_n}(x)\mathcal{N}(x; \hat{r}_n, v_n^r)dx} \tag{5}$$

where  $p_{x_n}(x)$  is the prior of random variable  $x_n$  in the input channel  $r_n = x_n + v_n$ . We want to obtain the posterior mean and variance estimates

$$\mathbb{E}\{x_n|r_n = \hat{r}_n, v_n^r\} = \int x_n p_{x_n|r_n}(x_n|\hat{r}_n, v_n^r)dx_n \tag{6}$$

$$\text{var}\{x_n|r_n = \hat{r}_n, v_n^r\} = \int (x_n - \mathbb{E}\{x_n|r_n = \hat{r}_n, v_n^r\})^2 p_{x_n|r_n}(x_n|\hat{r}_n, v_n^r)dx_n. \tag{7}$$

GAMP algorithm is an intrinsic parallel computing process, since it decouples  $y_m = \sum_{n=1}^N a_{mn}x_n + w_m$  to the input channel  $r_n = x_n + v_n$  estimation and the output channel  $y_m = z_m + w_m$  estimation. According to the factor graph theory, the messages are probability measures. They can be classified to two types: from the variable node to the factor node, and from the factor node to the variable node. For AMP and GAMP algorithms, all the messages are passed simultaneously between the factor nodes and variable nodes along the edges on the factor graph. These probability measures can be approximated as Gaussian density, based on the derivation of AMP and GAMP algorithms. Therefore, the mean and variance of every entry of  $\mathbf{x}$  are i.i.d. Gaussian random variable. This intrinsic parallel computing implies a flaw: if  $\mathbf{A}$  does not satisfy restricted isometry property (RIP) very well, For example  $\mathbf{A}$  has highly correlated columns, GAMP will diverge. In real world problems, such as radar application, the sensing matrix  $\mathbf{A}$  may be deterministic, not statistics. Therefore the RIP analysis can not be applied to this situation, researchers use simulation to find the behavior of this kind of reconstruction. We will show it in Section 5.

The steps of GAMP are listed in Algorithm 1, where  $|\mathbf{A}|^2$  is a componentwise magnitude squared, the elementwise product and division are denoted as  $\odot$  and  $\oslash$ , respectively, notation  $[\cdot]$  means the componentwise form of a vector, e.g.,  $\mathbf{a} = [a_i]$ , and notation  $|\cdot|$  means the cardinality of a set, or the number of non-zero elements of a sparse vector.

---

**Algorithm 1:** Generalized Approximate Message Passing (GAMP).

---

**Input:**  $\mathbf{y}, \mathbf{A}$

**Description:**  $\mathbf{x}(t+1), \mathbf{v}^{\mathbf{x}}(t+1)$

- 1:  $\hat{\mathbf{x}}(t=1) \leftarrow \mathbf{0}; \mathbf{v}^{\mathbf{x}}(t=1) \leftarrow \mathbf{1}; \hat{\mathbf{s}}(t=0) \leftarrow \mathbf{0}$
  - 2: **repeat**
  - 3:  $\mathbf{v}^{\mathbf{P}}(t) \leftarrow |\mathbf{A}|^2 \mathbf{v}^{\mathbf{x}}(t)$
  - 4:  $\hat{\mathbf{p}}(t) \leftarrow \mathbf{A}\hat{\mathbf{x}}(t) - \hat{\mathbf{s}}(t-1) \odot \mathbf{v}^{\mathbf{P}}(t)$
  - 5:  $\hat{\mathbf{z}}(t) \leftarrow [\hat{z}_m(t)]$ , where  $\hat{z}_m(t) \triangleq \mathbb{E}\{z_m | p_m = \hat{p}_m(t), v_m^p(t)\}$
  - 6:  $\mathbf{v}^{\mathbf{z}}(t) \leftarrow [v_m^z(t)]$ , where  $v_m^z(t) \triangleq \text{var}\{z_m | p_m = \hat{p}_m(t), v_m^p(t)\}$
  - 7:  $\hat{\mathbf{s}}(t) \leftarrow (\mathbf{1} \odot \mathbf{v}^{\mathbf{P}}(t)) \odot (\hat{\mathbf{z}}(t) - \hat{\mathbf{p}}(t))$
  - 8:  $\mathbf{v}^{\mathbf{s}}(t) \leftarrow (\mathbf{1} \odot \mathbf{v}^{\mathbf{P}}(t)) \odot (1 - \mathbf{v}^{\mathbf{z}}(t) \odot \mathbf{v}^{\mathbf{P}}(t))$
  - 9:  $\mathbf{v}^{\mathbf{x}}(t) \leftarrow \mathbf{1} \odot (|\mathbf{A}|^2)^T \mathbf{v}^{\mathbf{s}}(t)$
  - 10:  $\hat{\mathbf{r}}(t) \leftarrow \hat{\mathbf{x}}(t) + \mathbf{v}^{\mathbf{x}}(t) \odot (\mathbf{A}^T \hat{\mathbf{s}}(t))$
  - 11:  $\hat{\mathbf{x}}(t+1) \leftarrow [\hat{x}_n(t+1)]$ , where  $\hat{x}_n(t+1) \triangleq \mathbb{E}\{x_n | r_n = \hat{r}_n(t), v_n^r(t)\}$
  - 12:  $\mathbf{v}^{\mathbf{x}}(t+1) \leftarrow [v_n^x(t+1)]$ , where  $v_n^x(t+1) \triangleq \text{var}\{x_n | r_n = \hat{r}_n(t), v_n^r(t)\}$
  - 13: **until** Terminated
- 

When the prior is a Bernoulli-Gaussian distribution [7]

$$p_{x_n}(x) = (1 - \lambda)\delta(x) + \lambda\mathcal{N}(x; \theta, \phi), \tag{8}$$

where  $\lambda$  is the sparsity-rate  $K/N$ , and the mean  $\theta$  and variance  $\phi$  are hyper-parameters which are set to zero and one respectively at the initiation and those parameters do not change in the algorithm running time. It is called BG-GAMP algorithm. One can obtain the input channel marginal posterior

$$p_{x_n|r_n}(x|\hat{r}_n, v_n^r) = \left( (1 - \lambda)\delta(x_n) + \lambda\mathcal{N}(x_n; \theta, \phi) \right) \frac{\mathcal{N}(x_n; \hat{r}_n, v_n^r)}{\zeta_n} \tag{9}$$

$$= (1 - \pi_n)\delta(x_n) + \pi_n\mathcal{N}(x_n; \gamma_n, \mu_n) \tag{10}$$

with the normalization factor

$$\zeta_n \triangleq \int p_{x_n}(x)\mathcal{N}(x; \hat{r}_n, v_n^r)dx = (1 - \lambda)\mathcal{N}(0; \hat{r}_n, v_n^r) + \lambda\mathcal{N}(0; \hat{r}_n - \theta, v_n^r + \phi) \tag{11}$$

and  $(\hat{r}_n, v_n^r, \lambda, \theta, \phi)$ -dependent variables

$$\pi_n \triangleq \frac{1}{1 + \left( \frac{\lambda\mathcal{N}(\hat{r}_n; \theta, v_n^r + \phi)}{(1-\lambda)\mathcal{N}(0; \hat{r}_n, v_n^r)} \right)^{-1}} \tag{12}$$

$$\gamma_n \triangleq \frac{\hat{r}_n/v_n^r + \theta/\phi}{1/v_n^r + 1/\phi} \tag{13}$$

$$\mu_n \triangleq \frac{1}{1/v_n^r + 1/\phi} \tag{14}$$

where  $\pi_n$  is the input channel marginal posterior support probability  $\Pr\{x_n \neq 0 | \hat{r}_n, v_n^r\}$ , according to Equation (10), and then  $\Pr\{x_n = 0 | \hat{r}_n, v_n^r\} = 1 - \pi_n$ . Substituting Equation (10) to Equations (6) and (7), we have

$$\hat{x}_n = \pi_n\gamma_n + (1 - \pi_n)\theta \tag{15}$$

$$v_n^x = \pi_n(|\gamma_n|^2 + \mu_n) + (1 - \pi_n)|\theta|^2 - |\hat{x}_n|^2. \tag{16}$$

The hypothesis of Bernoulli-Gaussian data  $\mathbf{x}$  is sparse. Sparsity-rate  $\lambda$  has been set to  $K/N$ . Although the sparsity  $K$  is unknown in practice, we can choose a small  $K$  value and an error bound,

and then we do the recovery algorithm. If the recovery result makes the final reconstruction error lower than the bound, this result can be accepted, otherwise we decrease the  $K$  value and do the recovery again until the bound has been achieved. This procedure is like the Least Angle Regression (LARS) algorithm which tests all the regularization hyper-parameters.

### 3. Fixed Support GAMP Algorithm

According to the belief propagation theory, if the number of cycles on a factor graph is low, then the robustness of message passing will be enhanced [8]. Therefore, we have an idea that the message passing can be restricted just on the support  $\Lambda$  if  $\Lambda$  has been determined in some way. That is the meaning of FS-GAMP. The steps of FS-GAMP are listed in Algorithm 2. It is very similar to the GAMP algorithm, but the major difference between FS-GAMP and GAMP is that the entries which do not belong to the support  $\Lambda$  are assigned to 0 directly.

---

**Algorithm 2:** Fixed Support GAMP (FS-GAMP).

---

**Input:**  $\mathbf{y}, \mathbf{A}, \Lambda$

**Description:**  $\mathbf{x}(t+1), \mathbf{v}^{\mathbf{x}}(t+1)$

1:  $\hat{\mathbf{x}}_{\Lambda}(t=1) \leftarrow \mathbf{0}^{|\Lambda| \times 1}; \mathbf{v}^{\mathbf{x}\Lambda}(t=1) \leftarrow \mathbf{1}^{|\Lambda| \times 1}; \hat{\mathbf{s}}(t=0) \leftarrow \mathbf{0}$

2: **repeat**

3:  $\mathbf{v}^{\mathbf{p}}(t) \leftarrow |\mathbf{A}_{:, \Lambda}|^2 \mathbf{v}_{\Lambda}^{\mathbf{x}}(t)$

4:  $\hat{\mathbf{p}}(t) \leftarrow \mathbf{A}_{:, \Lambda} \hat{\mathbf{x}}_{\Lambda}(t) - \hat{\mathbf{s}}(t-1) \odot \mathbf{v}^{\mathbf{p}}(t)$

5: {(line 5)–(line 8) of GAMP}

6:  $\mathbf{v}_{\Lambda}^{\mathbf{x}}(t) \leftarrow \mathbf{1} \oslash ((|\mathbf{A}_{:, \Lambda}|^2)^T \mathbf{v}^{\mathbf{s}}(t))$

7:  $\hat{\mathbf{r}}_{\Lambda}(t) \leftarrow \hat{\mathbf{x}}_{\Lambda}(t) + \mathbf{v}_{\Lambda}^{\mathbf{x}}(t) \odot (\mathbf{A}_{:, \Lambda}^T \hat{\mathbf{s}}(t))$

8:  $\hat{\mathbf{x}}_{\Lambda}(t+1) \leftarrow [\hat{x}_i(t+1)]$ , where  $i \in \Lambda, \hat{x}_i(t+1) \triangleq \mathbb{E}\{x_i | r_i = \hat{r}_i(t), v_i^{\mathbf{r}}(t)\}$

9:  $\mathbf{v}_{\Lambda}^{\mathbf{x}}(t+1) \leftarrow [v_i^{\mathbf{x}}(t+1)]$ , where  $i \in \Lambda, v_i^{\mathbf{x}}(t+1) \triangleq \text{var}\{x_i | r_i = \hat{r}_i(t), v_i^{\mathbf{r}}(t)\}$

10: **until**

---

### 4. Bernoulli-Gaussian Pursuit GAMP Algorithm

The hypothesis of FS-GAMP algorithm is a known support  $\Lambda$ , but we do not know it. Now back to BG-GAMP, Equations (15) and (16) use the marginal posterior probability  $\pi_n$  to weight the non-zero estimation part  $(\gamma_n, \mu_n)$  and the zero prior part  $\theta$ . At each iteration of BG-GAMP, when  $\pi_n$  is greater than zero-one switch threshold  $\zeta = 0.5$ , the non-zero estimation part takes over the major contribution. In this research we suppose that the non-zero prior probability of every entry in  $\mathbf{x}$  is 0.5. That implies the maximal entropy assumption. Therefore, the zero-one switch threshold is also set to 0.5 in accordance with the maximal entropy assumption. For a matrix which does not meet RIP conditions very well, we observe that the mean and the variance of one  $\pi_n > \zeta$  entry shows no sign of convergence, while some entries'  $\pi_j < \zeta, j \neq n$  already increase fast. When they are greater than  $\zeta$ , the means and the variances of these entries also show no sign of convergence. This phenomenon spreads to more and more entries, causing all entries to be totally diverged.

This observation inspires us to maintain the current computing status until one  $\pi_n > \zeta$  entry has converged, and then continue the next entry (or entries). In other words, we introduce a pursuit process of lines (4)–(6) in Algorithm 3 to sequentially find the support of  $\mathbf{x}$ , and then estimate the amplitudes on this support by FS-GAMP. BGP-GAMP means BG-GAMP with an intrinsic pursuit process. The steps of BGP-GAMP are listed in Algorithm 3. Note that the initial input parameters of FS-GAMP in line (7) are the output parameters of GAMP in line (3).

The computational complexity of GAMP part in BGP-GAMP algorithm is the same as the standard GAMP, that is  $O(MN)$ , at each iteration. The computational complexity of the FS-GAMP part in BGP-GAMP is  $O(M|\Lambda|)$  at each iteration. Because  $|\Lambda| \leq K$ , the computational complexity of BGP-GAMP at each iteration is less than  $O(M(N+K))$ .

**Algorithm 3:** Bernoulli-Gaussian Pursuit GAMP (BGP-GAMP).

---

**Input:**  $\mathbf{y}, \mathbf{A}$   
**Description:**  $\mathbf{x}(t+1), \nu^{\mathbf{x}}(t+1)$   
 1:  $\hat{\mathbf{x}}(t=1) \leftarrow \mathbf{0}; \nu^{\mathbf{x}}(t=1) \leftarrow \mathbf{1}; \hat{\mathbf{s}}(t=0) \leftarrow \mathbf{0}; \Lambda \leftarrow \emptyset$   
 2: **repeat**  
 3:   {(line 3)–(line 12) of GAMP}  
 4:    $\pi(t+1) \leftarrow [\pi_n(t+1)]$ , where  $\pi_n(t+1)$  obtained from (12)  
 5:    $J \leftarrow \{j : \pi_j(t+1) > 0.5\}$   
 6:    $\Lambda \leftarrow \Lambda \cup J$   
 7:   {FS-GAMP}  
 8: **until** Terminated.

---

**5. Experiments**

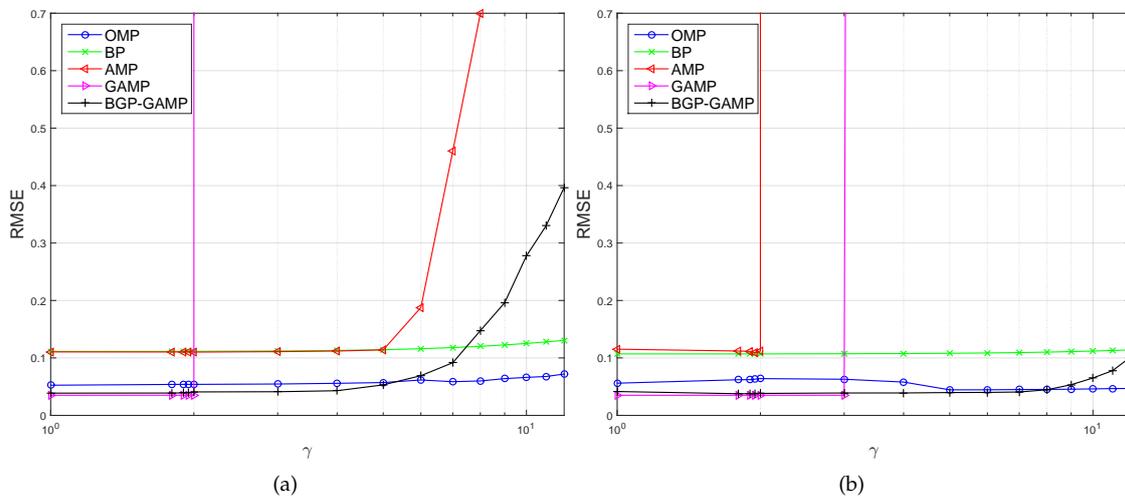
We investigate the performance of BGP-GAMP by three experiments. Four benchmark algorithms are adopted to compare with BGP-GAMP. Orthogonal matching pursuit (OMP) [9] and basis pursuit (BP) by interior point method to solve linear programming [10] are the most robust  $\ell_0$  and  $\ell_1$  optimization algorithms, other algorithms (old or new) are weaker than them in robustness. AMP and GAMP are not only the root cause of the robust problem, but also the basis of our method. The implementation of OMP comes from (<http://www.cmc.edu/pages/faculty/DNeedell>), BP comes from *l1magic* (<http://users.ece.gatech.edu/justin/l1magic>), AMP comes from Kamilov's work (<http://people.epfl.ch/ulugbek.kamilov>) and GAMP comes from *gamplab* (<http://eeweb.poly.edu/srangan/index.html>).

We do  $Q = 100$  trials in each experiment, and use the relative mean square error (RMSE)  $\xi = 1/Q \times \sum_{l=1}^Q \|\hat{\mathbf{x}}^l - \mathbf{x}^l\|_2 / \|\mathbf{x}^l\|_2$  as the performance metric. Given  $N/M = 3, M/K = 5$ , we set  $N = 256$ , and then  $M = 86, K = 18$ . Choose  $K$  locations at random as the support of  $\mathbf{x}$ , and the amplitude of each non-zero entry  $x_i$  is independently drawn from  $\mathcal{N}(0, 1)$ . We set the signal-to-noise ratio (SNR) to 25 dB.

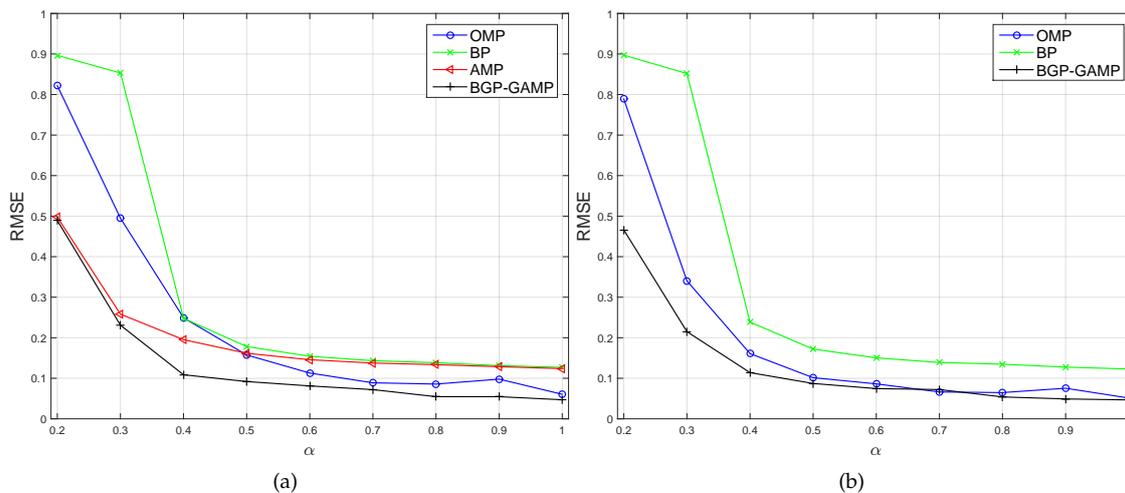
The first experiment, termed  $\gamma$  test, investigates non-zero mean Gaussian sensing matrix case. We create two types of sensing matrix: type 1 means  $A(:, j) = \gamma/N + \mathbf{a}_j/\sqrt{N}$ ; type 2 means  $A(:, j) = \gamma/N + \mathbf{a}_j/\|\mathbf{a}_j\|_2$ , where each entry of  $\mathbf{a}_j$  is i.i.d. drawn from  $\mathcal{N}(0, 1)$ . Set the range of  $\gamma$  to  $[0, 1, 1.8, 1.9, 1.95, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]$ . Figure 1a shows that GAMP violently diverges at  $\gamma = 2$ , AMP fast diverges at  $\gamma = 5$ . The severely sharp divergence of GAMP and AMP comes from the phase transition property of compressed sensing. Although BGP-GAMP also begins to diverge at  $\gamma = 5$ , the deterioration speed is slower than AMP. Furthermore, the RMSE of BGP-GAMP is always less than that of AMP. When  $\gamma < 4$ , the RMSE of BGP-GAMP is a little less than that of OMP, and when  $\gamma < 6$ , the RMSE of BGP-GAMP is obviously less than that of BP. Figure 1b shows that AMP violently diverges at  $\gamma = 1.95$ , GAMP violently diverges at  $\gamma = 3$ , but BGP-GAMP just begins to diverge at  $\gamma = 8$ . The RMSE of BGP-GAMP is always less than that of BP, and less than that of OMP until  $\gamma = 8$ . Moreover, Figure 1a,b shows that BGP-GAMP outperforms OMP slightly in performance over a parameter range  $\gamma < 2$ . In summary, our proposed algorithm obtains better robustness compared with the standard AMP/GAMP algorithms, at the same time our proposed algorithm achieves performance advantage over a suitable parameter range compared with the standard OMP and BP algorithms.

The second experiment, termed  $\alpha$  test, investigates the highly correlated columns sensing matrix **A** case. The elements of **A** are neither normal nor i.i.d distributed. Create two types of sensing matrix: type 1 means  $\mathbf{A} = (1/N)\mathbf{P}\mathbf{Q}$ ,  $p_{mr}, q_{rn} \sim \mathcal{N}(0, 1)$ , where  $\mathbf{P} \in \mathbb{R}^{M \times R}$ ,  $\mathbf{Q} \in \mathbb{R}^{R \times N}$ , and  $R \triangleq \alpha N$ ; type 2 means  $\mathbf{A} = \mathbf{P}\mathbf{Q}$  like type 1, but normalizes each column of **A**. We know that **A** is *low rank* for  $\alpha < M/N$ . Set the range of  $\alpha$  to  $[0.2, 0.3, \dots, 1.0]$ . Since GAMP totally diverges at every  $\alpha$  with type 1 and type 2 cases, it is not visible in Figure 2a,b. AMP also totally diverges at every  $\alpha$  with type 2 case, therefore it

is also not visible in Figure 2b. The two figures show that BGP-GAMP remains stable even in a low rank scenario  $\alpha \approx 0.4$ , and its performance is obviously better than AMP and BP, and a little better than OMP.



**Figure 1.** Reconstruction performance comparison of the Non-zero mean sensing matrix. (a) Type-1 sensing matrix; (b) Type-2 sensing matrix.

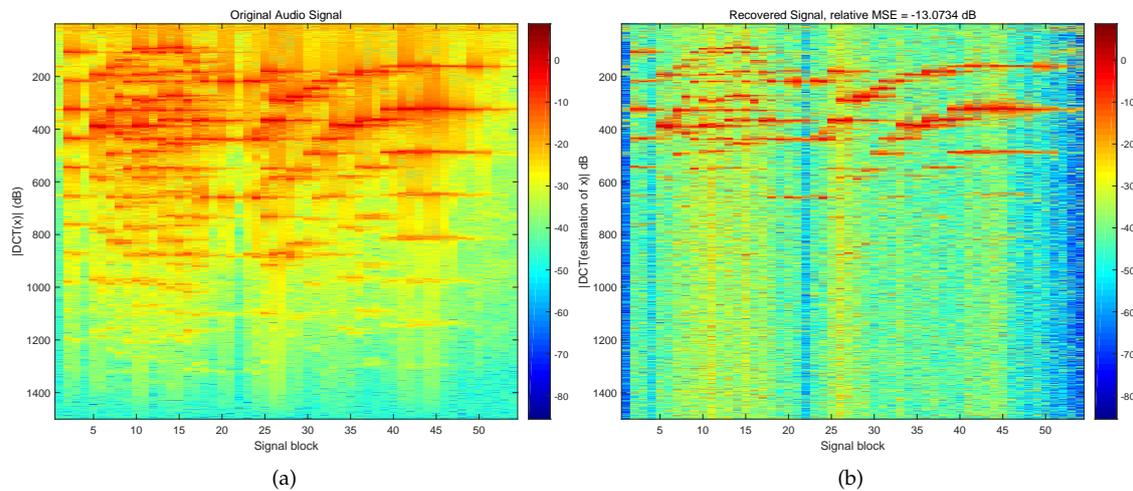


**Figure 2.** Reconstruction performance comparison of the highly column correlations sensing matrix. (a) Type-1 sensing matrix; (b) Type-2 sensing matrix.

It is worth noting that the threshold 0.1 in Figures 1 and 2 is the relative mean square error, not absolute mean square error, which means the average error amplitude is just 1/100 of the original signal amplitude. These figures show, for instance, that the standard AMP algorithm and BP algorithm are approaching this threshold. Therefore, it is not a poor performance. We do not consider MAP-GAMP since MAP estimation is not general as MMSE estimation, please see [11] Section 3.6.3 for more discussion.

The last experiment uses real world data, namely, a segment of music by Mozart. The data has 81,000 sample points, and it is segmented to 54 blocks, each block has  $N = 1500$  sample points. The compressed sampling rate is set to 1/3, therefore, the measurement number is  $M = 500$ . We create the sensing matrix by DCT transform:  $\mathbf{A}(i,:) = (1/N)(\text{DCT}(\Phi(i,:)))^T$ , where every element of  $\Phi$  is drawn from  $\mathcal{N}(0,1)$ , and then multiplied a scale factor  $1/\sqrt{M}$  to it. Figure 3a shows the modulus

of DCT coefficients of the original signal, and Figure 3b shows the modulus of DCT coefficients of the recovered signal. We can see that the major energy has been recovered from the compressed measurements.



**Figure 3.** Recover the audio signal by Bernoulli Gaussian Pursuit GAMP. (a) Spectra of the original signal; (b) Spectra of the recovered signal, major signal energy has been restored but lost some detail.

## 6. Conclusions

In this paper we propose a new compressed sensing algorithm, the two-stage method. This algorithm pursues the support of a sparse vector sequentially, and meanwhile estimates the amplitudes of the sparse vector by the FS-GAMP process. Numerical experiments with simulation and real world data confirm that our new algorithm performs very well in correctness and robustness. In future work we plan to analyze the variation rate of the marginal posterior probability  $\pi$  to obtain better results.

**Acknowledgments:** The author would like to thank the anonymous reviews for their comments that helped to improve the quality of the paper. This research was supported by the National Natural Science Foundation of China (NSFC) under Grant U1533125, and the Scientific Research Program of the Education Department of Sichuan under Grant 16ZB0032.

**Author Contributions:** Qingshan You, Yongjie Luo and Qun Wan conceived and designed the experiments; Yongjie Luo performed the experiments; Qingshan You wrote the paper.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Eldar, Y.C.; Kutyniok, G. *Compressed Sensing: Theory and Applications*; Cambridge University Press: Cambridge, UK, 2012.
2. Donoho, D.L.; Maleki, A.; Motanari, A. Message passing algorithms for compressed sensing. *Proc. Natl. Acad. Sci. USA* **2009**, *106*, 18914–18919.
3. Rangan, S. Generalized approximate message passing for estimation with random linear mixing. In Proceedings of the IEEE International Symposium on Information Theory (ISIT), St. Petersburg, Russia, 31 July–5 August 2011; pp. 2168–2172.
4. Vila, J.; Schniter, P.; Rangan, S.; Krzakala, F.; Zdeborová, L. Adaptive damping and mean removal for the generalized approximate message passing algorithm. In Proceedings of the 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brisbane, Australia, 19–24 April 2015; pp. 2021–2025.
5. Luo, Y.J.; Gui, G.; Cong, X.C.; Wan, Q. Sparse Estimation Based on a New Random Regularized Matching Pursuit Generalized Approximate Message Passing Algorithm. *Entropy* **2016**, *18*, 207.

6. Rodger, J.A. A fuzzy nearest neighbor neural network statistical model for predicting demand for natural gas and energy cost savings in public buildings. *Expert Syst. Appl.* **2014**, *41*, 1813–1829.
7. Vila, J.; Schniter, P. Expectation-Maximization Bernoulli-Gaussian Approximate Message Passing. In Proceedings of the Asilomar Conference on Signals, Systems, and Computers (SS&C), Pacific Grove, CA, USA, 6–9 November 2011; pp. 799–803.
8. Wainwright, M.J.; Jordan, M.I. Graphical models, exponential families, and variational inference. *Found. Trends Mach. Learn.* 2008, *1*, 1–305.
9. Pati, Y.C.; Rezaifar, R.; Krishnaprasad, P.S. Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. In Proceedings of the Asilomar Conference on Signals, Systems, and Computers, Pacific Grove, CA, USA, 1–3 November 1993; Volume 1, pp. 40–44.
10. Candès, E.; Romberg, J.; Tao, T. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Trans. Inf. Theory* **2006**, *52*, 489–509.
11. Barbier, J. Statistical Physics and Approximate Message Passing Algorithms for Sparse Linear Estimation Problems in Signal Processing and Coding Theory. Ph.D. Thesis, Université Paris Diderot–Paris VII, Paris, France.



© 2016 by the authors; licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC-BY) license (<http://creativecommons.org/licenses/by/4.0/>).