Reconstruction algorithm in compressed sensing based on maximum a posteriori estimation

Koujin Takeda[1] and Yoshiyuki Kabashima[2]

Department of Intelligent Systems Engineering, Ibaraki University[1]
Department of Computational Intelligence and Systems Science, Tokyo Institute of Technology[2]

E-mail: ktakeda@mx.ibaraki.ac.jp

Abstract. We propose a systematic method for constructing a sparse data reconstruction algorithm in compressed sensing at a relatively low computational cost for general observation matrix. It is known that the cost of ℓ_1-norm minimization using a standard linear programming algorithm is $O(N^3)$. We show that this cost can be reduced to $O(N^2)$ by applying the approach of posterior maximization. Furthermore, in principle, the algorithm from our approach is expected to achieve the widest successful reconstruction region, which is evaluated from theoretical argument. We also discuss the relation between the belief propagation-based reconstruction algorithm introduced in preceding works and our approach.

1. Introduction

Nowadays, use of the compressed sensing (CS) approach [1, 2, 3] is rapidly spreading to various fields in information technology, where the sparsity of the original data plays a crucial role [4].

In this article, we present our study on a very basic problem of CS. Let us consider a linear observation process expressed as

$$\boldsymbol{y} = \boldsymbol{F}\boldsymbol{x}, \tag{1}$$

where $\boldsymbol{y} \in \mathbb{R}^M$ is the observed data and $\boldsymbol{x} \in \mathbb{R}^N$ is the original sparse data with many zero entries. The (averaged) number of nonzero entries is denoted by K. The $M \times N$ matrix $\boldsymbol{F}$ describes the process of observation, and the limit of large M, N with compression rate $\alpha := M/N < 1$ is taken into account. (Throughout this paper, bold symbols denote a vector/matrix.)

We investigate the basic ℓ_1-norm minimization

$$\min_{\boldsymbol{x}} \|\boldsymbol{x}\|_1 \ \text{ subject to } \ \boldsymbol{y} = \boldsymbol{F}\boldsymbol{x}. \tag{2}$$

This problem is written as linear programming and can be solved at a computational cost of $O(N^3)$ using a standard algorithm, such as the interior point method. However, many alternative algorithms have been proposed for reducing the computational cost [5, 6]. Among these, Approximate Message Passing (AMP) [7], which is a thresholding algorithm that is based on the message passing approach [8], is noteworthy. In this algorithm, each entry of the matrix $\boldsymbol{F}$ is assumed to be random with identical Gaussian distribution. Then, it is shown that the original data can be reconstructed at a computational cost of $O(N^2)$. Furthermore, this algorithm has theoretical significance: for the AMP, the successful reconstruction threshold in terms of compression rate α and data sparsity $\rho := K/N$ is analytically shown to be the same as for ℓ_1-norm minimization [9] using a state evolution technique [7, 10, 11, 12]. However, it is not intuitively evident from their argument why the threshold of the AMP agrees with that of ℓ_1-norm minimization.

In this article, we propose an approach for constructing a sparse data reconstruction algorithm by using maximum *a posteriori* probability (MAP) for general matrix $\boldsymbol{F}$ with as small an M as possible. Using this approach, we obtain an algorithm whose computational cost is $O(N^2)$. (For a special case, when the matrix $\boldsymbol{F}$ is sparse and has only $O(1)$ nonzero entries in each column/row, the computational cost is reduced to $O(N)$.) In addition, our approach explains from another perspective why the reconstruction thresholds of the AMP and ℓ_1-norm minimization are analytically equivalent.

2. Reconstruction algorithm using the MAP

To construct our MAP algorithm, first, we prepare the posterior probability. We need to infer the original data $\boldsymbol{x}$ from the observed data $\boldsymbol{y}$ and the matrix $\boldsymbol{F}$; therefore, we define the posterior probability $P(\boldsymbol{x}|\boldsymbol{y},\boldsymbol{F})$ for $\boldsymbol{x}$ in accordance with the ℓ_1-norm minimization problem as (2)

$$P(\boldsymbol{x}|\boldsymbol{y},\boldsymbol{F}) \quad := \quad \lim_{\beta\to\infty}\frac{1}{Z(\beta)}\exp\left(-\beta\sum_i|x_i|\right)\prod_\mu\delta\left(y_\mu-\sum_i F_{\mu i}x_i\right), \tag{3}$$

where $Z(\beta)$ is the normalization,

$$\begin{aligned} Z(\beta) &:= \exp(-\beta C), \\ C &:= \min_{\boldsymbol{x}}\|\boldsymbol{x}\|_1 \ \text{ subject to } \ \boldsymbol{y}=\boldsymbol{F}\boldsymbol{x}. \end{aligned} \tag{4}$$

(Roman subscripts run from 1 to N, and Greek from 1 to M). This probability is zero unless the constraint $\boldsymbol{y}=\boldsymbol{F}\boldsymbol{x}$ is satisfied. In addition, it is unity only if ℓ_1-norm of $\boldsymbol{x}$ is minimum; otherwise zero. Then, it is found that the MAP solution is equivalent to the ℓ_1-norm minimum. Note that we do not need to take the limit of $\beta\to\infty$ for a MAP solution; however, we take it for clarifying the relation between the MAP algorithm and the reconstruction threshold analysis using a statistical mechanical replica method [13] that gives the exact expression of the threshold [9], where this posterior probability is defined as the Boltzmann weight, and zero temperature limit $\beta\to\infty$ is taken for technical reasons related to the analysis.

An obstacle to dealing with posterior probability is the singular delta function; we regularize this by the quadratic term

$$P(\boldsymbol{x}|\boldsymbol{y},\boldsymbol{F}) \quad = \quad \lim_{\beta\to\infty}\frac{1}{Z(\beta)}\exp\left(-\beta\frac{\sum_\mu(y_\mu-\sum_i F_{\mu i}x_i)^2}{2}-\beta k\sum_i|x_i|\right). \tag{5}$$

This posterior reproduces the original ℓ_1 minimization problem by taking the limit $\beta\to\infty$. (The readers can find a similar form of the probability density in [10] through the discussion of the AMP algorithm.) In this formulation, we need to introduce a constant $k(>0)$ that represents the “relative significance” of constraint and minimization. For ℓ_1-norm minimum solution, we must consider the limit $k\to0$ because we are attempting to find the minimum solution *under* the constraint at present. This framework is essentially the same as that used in the lasso [14]; however, the crucial point here is that we need to take the limit $k\to0$ for the MAP solution appropriately.

For the MAP solution, we differentiate the term in the exponential with respect to x_i, and write the stationary condition

$$\sum_\mu F_{\mu i}y_\mu-\sum_{j(\neq i)}\sum_\mu F_{\mu i}F_{\mu j}x_j-\sum_\mu F_{\mu i}^2x_i-k\Theta(x_i)=0 \tag{6}$$

Journal of Physics: Conference Series **473** (2013) 012003 doi:10.1088/1742-6596/473/1/012003

(where $\Theta(x)$ is the Heaviside function), which can be rewritten as

$$\begin{aligned} x_i &= \frac{1}{\sum_\mu F_{\mu i}^2} \; \eta\left(\sum_\mu F_{\mu i}\left(z_\mu + F_{\mu i}x_i\right); k\right), \\ z_\mu &:= y_\mu - \sum_i F_{\mu i}x_i. \end{aligned} \tag{7}$$

Here, for convenience, we define the function for thresholding

$$\eta(x;k) := \begin{cases} x-k & k<x \\ 0 & -k \le x \le k \\ x+k & x<-k \end{cases}, \tag{8}$$

and also introduce the variable z_μ, which represents the residual error of the constraint $\boldsymbol{y} = \boldsymbol{F}\boldsymbol{x}$. Basically, we can obtain the ℓ_1-norm minimum solution by solving this stationary condition. We construct an iterative algorithm by adding the iteration step superscript (t):

$$\begin{aligned} x_i^{(t)} &= \frac{1}{\sum_\mu F_{\mu i}^2} \; \eta\left(\sum_\mu F_{\mu i}\left(z_\mu^{(t)} + F_{\mu i}x_i^{(t-1)}\right); k\right), \\ z_\mu^{(t)} &= y_\mu - \sum_i F_{\mu i}x_i^{(t-1)}. \end{aligned} \tag{9}$$

The MAP solution is obtained by finding the fixed point of $x_i^{(t)}$. Remember that the original ℓ_1-norm minimization is a convex optimization, and accordingly there is no fixed point of local minimum.

We now give some remarks about this algorithm (9).

- Computational cost
  In algorithmic equations (9), only a single summation appears, and the numbers of the equations for $x_i^{(t)}$ and $z_\mu^{(t)}$ are N, M, respectively. Then, the computational cost is $O(N^2)$ when the compression rate M/N is $O(1)$. In addition, if the matrix $\boldsymbol{F}$ is sparse (having only $O(1)$ nonzero entries in each column/row, which is discussed in [15, 16, 17]), the computational cost is reduced to $O(N)$. Note that the number of iteration steps for convergence is assumed to be much smaller than the orders of M and N. However, in general, it does not hold near the region of the reconstruction threshold, where the speed of $k \to 0$ limit must be slow for convergence and the number of iteration steps for $k \to 0$ becomes dominant in comparison with the orders of M and N.
- Applicability to general $\boldsymbol{F}$
  We do not make any assumption for $\boldsymbol{F}$. Therefore, in principle, this algorithm can be used for general matrix $\boldsymbol{F}$, whereas in the original AMP [7] or in the statistical mechanical analysis [13] i.i.d. random entries are assumed.
- $k \to 0$ limit
  We must take the $k \to 0$ limit to the final iteration step. Empirically, k should be decreased exponentially with the step as $k^{(t)} \propto \exp(-t/\mathrm{const.})$. The decay constant in the exponential is significant because too small a constant leads to a wrong solution. This constant is found to be the threshold parameter in the AMP algorithm equation as will be elucidated later, which is significant for convergence as stated in [7]. In the original work of the AMP [7] the authors discussed the appropriate limit of $k \to 0$ by choosing k as the mean squared error, and, in conjunction with the state evolution technique, they arrive at the

reconstruction threshold for convergence, which is represented by the relation between α and ρ with auxiliary variable z,

$$\rho = \alpha \max_{z\geq 0}\left(\frac{1-\frac{2}{\alpha}\left\{(1+z^2)H(z) - z\frac{e^{-\frac{z^2}{2}}}{\sqrt{2\pi}}\right\}}{1+z^2-2\left\{(1+z^2)H(z) - z\frac{e^{-\frac{z^2}{2}}}{\sqrt{2\pi}}\right\}}\right), \tag{10}$$

where $H(z) := \int_z^{+\infty} dx e^{-x^2/2}/\sqrt{2\pi}$. This expression is *analytically* equivalent to that of ℓ_1-norm minimization evaluated by combinatorial geometry [9, 18]. Replica method can also give the threshold equations [13] as

$$2(1-\rho)\left(H(z) - \frac{1}{z}\frac{e^{-\frac{z^2}{2}}}{\sqrt{2\pi}}\right) + \rho = 0,$$
$$\alpha = 2(1-\rho)H(z) + \rho, \tag{11}$$

which is also analytically equivalent. (Readers can check the equivalence between (10) and (11) after some algebra.)
As elucidated above, our initial step for the algorithm is the probability distribution in (5), which appears as Boltzmann weight in replica analysis for computation of the threshold. In the AMP, the convergence condition (11) is discussed using the thresholding algorithm *without* Onsager term (namely, from the naïve algorithm (9)), which also originates from the distribution (5). This relation explains why we arrive at the same threshold equation.

- Partial update
  As is widely recognized, the naïve thresholding algorithm (9) still presents a problem. We cannot always obtain a correct ℓ_1-norm minimum solution using (9), even in successful reconstruction region from (10, 11) and even when the limit $k \to 0$ is suitably designed. Such a failure occurs near the reconstruction threshold of ℓ_1-norm minimization. (Readers will find the details below.) The AMP successfully solves this problem by the introduction of Onsager reaction term, which is reduced from perturbative analysis in the original work [7]. We attempt to address this problem from another point of view, which is applicable for general matrix $\boldsymbol{F}$.
  The strategy is as follows. In the original algorithm, we fully update the variable $x_i^{(t)}$ in each step. Now, we introduce a partition ratio in the algorithm, and modify the algorithm to include the partial update rule as

$$\begin{aligned} x_i^{(t)} &= \frac{\gamma^{(t)}}{1+\gamma^{(t)}}x_i^{(t)} + \frac{1}{1+\gamma^{(t)}}\frac{1}{\sum_\mu F_{\mu i}^2}\,\eta\left(\sum_\mu F_{\mu i}\left(z_\mu^{(t)} + F_{\mu i}x_i^{(t-1)}\right); k\right), \\ z_\mu^{(t)} &= y_\mu - \sum_i F_{\mu i}x_i^{(t-1)}, \end{aligned} \tag{12}$$

  where the partition ratio $\gamma^{(t)}$ is dependent on the step in general. Obviously, the fixed point of the algorithm is the same as that of the original; nevertheless, the convergence to a correct solution is improved by choosing $\gamma^{(t)}$ properly.
  Here $\gamma^{(t)}$ can be chosen arbitrarily. We can discuss how to design $\gamma^{(t)}$ to achieve better convergence by the stability analysis around the fixed point of the algorithm. However, at present we do not have the best prescription of how to choose $\gamma^{(t)}$. We omit the details of the discussion here due to limited space.

3. Discussion on the MAP algorithm

Let us move on to the properties of the proposed algorithm. Here, we use the matrix $\boldsymbol{F}$ with i.i.d random entries drawn from Gaussian distribution with zero mean and variance $1/M$, which is the same as in the AMP [7] and statistical mechanical analysis [13]. For large M, N, we have $\lim_{M\to\infty}\sum_\mu F_{\mu i}^2 = 1$ and the algorithm equation (9) is simplified to

$$\begin{aligned} x_i^{(t)} &= \eta\left(\sum_\mu F_{\mu i} z_\mu^{(t)} + x_i^{(t-1)}; k\right), \\ z_\mu^{(t)} &= y_\mu - \sum_i F_{\mu i} x_i^{(t-1)}, \end{aligned} \tag{13}$$

which can also be found in the introductory part of [7], where it is mentioned that the performance of this naïve thresholding algorithm is worse than that of ℓ_1-norm minimization. We introduce a partition ratio in the first line of (13) (here γ is not dependent on the step t) to improve it:

$$x_i^{(t)} = \frac{\gamma}{1+\gamma} x_i^{(t-1)} + \frac{1}{1+\gamma}\eta\left(\sum_\mu F_{\mu i} z_\mu^{(t)} + x_i^{(t-1)}; k\right). \tag{14}$$

We rewrite this equation as

$$x_i^{(t)} = \frac{\gamma}{1+\gamma} x_i^{(t-1)} + \eta\left(\frac{1}{1+\gamma}\left\{\sum_\mu F_{\mu i} z_\mu^{(t)} + x_i^{(t-1)}\right\}; k\right), \tag{15}$$

using the property of η. (We rescale the parameter k, which is irrelevant to the fixed point because k should be taken to zero, finally.) Instead of (15), we now introduce slightly modified update rule

$$x_i^{(t)} = \eta\left(\frac{1}{1+\gamma}\sum_\mu F_{\mu i} z_\mu^{(t)} + x_i^{(t-1)}; k\right) \tag{16}$$

by moving the first term on r.h.s. into the argument of η. This gives the same fixed point as (15), which can be verified by eliminating the step superscript (t) from both update rules (15, 16) and solving them with respect to x. With the rescaled residual error $\widehat{z}_\mu^{(t)} = z_\mu^{(t)}/(1+\gamma)$, we obtain

$$\begin{aligned} x_i^{(t)} &= \eta\left(\sum_\mu F_{\mu i} \widehat{z}_\mu^{(t)} + x_i^{(t-1)}; k\right), \\ \widehat{z}_\mu^{(t)} &= \frac{1}{1+\gamma}\left\{y_\mu - \sum_i F_{\mu i} x_i^{(t-1)}\right\}. \end{aligned} \tag{17}$$

This expression of the algorithm indicates that the partition ratio is associated with the scaling of the residual error z_μ.

Next, we consider the step-dependent $\gamma^{(t)}$. In particular, we choose

$$\gamma^{(t)} = \frac{1}{M}\sum_i \eta'\left(\sum_\mu F_{\mu i} z_\mu^{(t)} + x_i^{(t-1)}; k\right), \tag{18}$$

where $\eta'(x;k) := \partial_x \eta(x;k)$. This $\gamma^{(t)}$ is chosen in order to achieve faster convergence according to the second order stability analysis around the solution, however the details are omitted. The point is that $\gamma^{(t)}$ is expressed by the function for thresholding, namely equation (8), which also appears in the algorithm. Note that $\gamma^{(t)}$ approaches the value $K/M = \rho/\alpha$ for $t \to \infty$ when the reconstruction is successful. For this step-dependent $\gamma^{(t)}$, we can also reach another expression of the algorithm in a similar manner as constant γ, by using that $\gamma^{(t)}$ is represented by the function (8),

$$\begin{aligned}
x_i^{(t)} &= \eta\left(\sum_\mu F_{\mu i}\widehat{z}_\mu^{(t)} + x_i^{(t-1)}; k\right), \\
\widehat{z}_\mu^{(t)} &= \frac{1}{1 + \frac{1}{M}\sum_j \eta'\left(\sum_\mu F_{\mu j} z_\mu^{(t)} + x_j^{(t-1)}; k\right)} z_\mu^{(t)}, \\
z_\mu^{(t)} &= y_\mu - \sum_i F_{\mu i} x_i^{(t-1)}.
\end{aligned} \tag{19}$$

This expression indicates that the scaling of the residual error z_μ varies as a function of the step t.

4. Relation to the AMP

Based on the observation above, we choose the step-dependent partition ratio $\gamma^{(t)}$ with negative sign

$$\gamma^{(t)} = -\frac{1}{M}\sum_i \eta'\left(\sum_\mu F_{\mu i} z_\mu^{(t)} + x_i^{(t-1)}; k\right), \tag{20}$$

which leads to partial update rule by external division. In this case we have almost the same equations as in (19); the only difference is the negative sign in the denominator in the second equation. Here, we also introduce the partition ratio $\widehat{\gamma}^{(t)}$ for the update rule of the rescaled residual error $\widehat{z}_\mu^{(t)}$,

$$\widehat{z}_\mu^{(t)} = \widehat{\gamma}^{(t-1)}\widehat{z}_\mu^{(t-1)} + \frac{1 - \widehat{\gamma}^{(t-1)}}{1 - \frac{1}{M}\sum_j \eta'\left(\sum_\mu F_{\mu j} z_\mu^{(t)} + x_j^{(t-1)}; k\right)} z_\mu^{(t)}. \tag{21}$$

By choosing $\widehat{\gamma}^{(t)} = -\gamma^{(t)} = \sum_j \eta'(\sum_\mu F_{\mu j} z_\mu^{(t)} + x_j^{(t-1)})/M$, the algorithm is changed to

$$\begin{aligned}
x_i^{(t)} &= \eta\left(\sum_\mu F_{\mu i}\widehat{z}_\mu^{(t)} + x_i^{(x-1)}; k\right), \\
\widehat{z}_\mu^{(t)} &= z_\mu^{(t)} + \frac{1}{M}\widehat{z}_\mu^{(t-1)}\sum_j \eta'\left(\sum_\mu F_{\mu j} z_\mu^{(t)} + x_j^{(t-1)}; k\right), \\
z_\mu &= y_\mu - \sum_i F_{\mu i} x_i.
\end{aligned} \tag{22}$$

We arrive at this expression just by introducing the partition ratios and rewriting the update rule. When we replace $z_\mu^{(t)}$ with $\widehat{z}_\mu^{(t-1)}$ in the second equation above (approximation by $z \to \widehat{z}$, which does not change the fixed point because z and $\widehat{z}$ finally approach zero), and shift the step number of $x_i^{(t-1)}$ in η' to $t-2$ (which does not change the fixed point either), we finally obtain

$$\begin{aligned} x_i^{(t)} &= \eta\left(\sum_\mu F_{\mu i}\widehat{z}_\mu^{(t)} + x_i^{(t-1)}; k\right), \\ \widehat{z}_\mu^{(t)} &= y_\mu - \sum_i F_{\mu i}x_i^{(t-1)} + \frac{1}{M}\widehat{z}_\mu^{(t-1)}\sum_j \eta'\left(\sum_\mu F_{\mu j}\widehat{z}_\mu^{(t-1)} + x_j^{(t-2)}; k\right), \end{aligned} \tag{23}$$

which is nothing but the update rule of the variables in the AMP [7], where the parameter k controls the thresholding. Thus, we can find the relation between the AMP and the partial update MAP algorithm. In the AMP, the last term in the second equation in (23) is derived from the discussion of message passing, and is interpreted as the Onsager reaction term in statistical mechanics [7, 12]. Using the logic here, we can provide a viewpoint that this term is introduced by way of a partition ratio in order to achieve better convergence of the algorithm.

5. Numerical experiment

5.1. Reconstruction threshold

First, we evaluated the reconstruction threshold of our algorithm. In the experiment, the original data dimension was fixed as $N = 10^3$. The numbers of observations M and (averaged) nonzero data components K were varied. Each entry of $\boldsymbol{F}$ was drawn randomly and independently from Gaussian distribution with zero mean and variance $1/M$ (in Figures 1-3). Each nonzero element of $\boldsymbol{x}^0$ was similarly drawn from Gaussian distribution with zero mean and unit variance. The initial value of $x_i^{(t)}$ was set to be zero. The parameter k approached zero in each step exponentially by multiplying the factor 0.999 in each update, that is, a very slow approach to the limit of $k \to 0$ was used to guarantee convergence. We took the value $x_i^{(t)}$ as the reconstruction result after 5×10^3 steps in Figures 1 and 2, and 1×10^4 in Figures 3 and 4, respectively. For evaluating the reconstruction threshold, we conducted the experiment 50 times and computed the success rate for fixed M, K. In each trial, the reconstruction was judged to be a success if the mean squared error per data was less than 10^{-3}. Then, we computed the success rate for every pair of M, K by changing their values. (After the experiment for one pair of M, K, we increased/decreased M or K by 25.)

The results are shown in Figures 1-4. The success rate obtained in the experiment is expressed by gray-scale in each figure. In Figure 1, the result by the algorithm without partition ratio is shown, which exhibits narrower success region than that of ℓ_1-norm minimization. (This threshold was also evaluated in [7].) The result with the partition ratio $\gamma^{(t)} = 1$ is shown in Figure 2, where a reconstruction threshold close to ℓ_1-norm threshold curve is observed. Then, for improvement we used the step-dependent partition ratio in (18). With this modification, we can obtain much closer threshold to ℓ_1-norm curve as in Figure 3. In lower compression rate (=large α) region, the performance seems slightly worse than that of ℓ_1-norm minimization. However, we expect that larger number of iteration steps, appropriate choice of $k \to 0$ limit, and suitable partition ratio $\gamma^{(t)}$ will improve the performance. Next, for checking the applicability to general matrix $\boldsymbol{F}$, we study the case where the random matrix $\boldsymbol{F}$ is generated by a different rule: we first generate a dense random matrix $\boldsymbol{F}$ using the same rule as before, and then randomly eliminate 90% of the entries, which are set to be zero. In this experiment we use the algorithm (12) with the partition ratio $\gamma^{(t)} = 1$. The result in Figure 4 exhibits a success region similar to that of ℓ_1-norm, which can be understood from reconstruction threshold universality for wide class of random $\boldsymbol{F}$, as indicated in [19, 7] and theoretically discussed in [13, 20, 21].

From these results, we confirm that the MAP algorithm is essentially equivalent to ℓ_1-norm minimization, and through a suitable design of the algorithm, we can expect a performance that

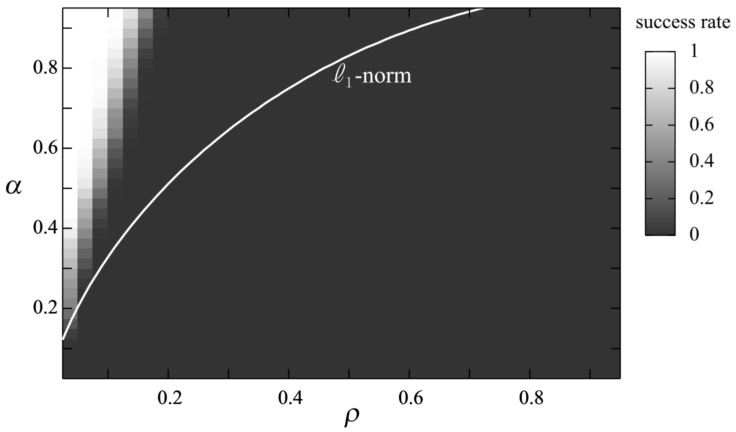

Figure 1. Profile of the success rate using the algorithm without partition ratio (9). Success rate is displayed by gray-scale. As seen, the success region is not as wide as with ℓ_1-norm minimization.

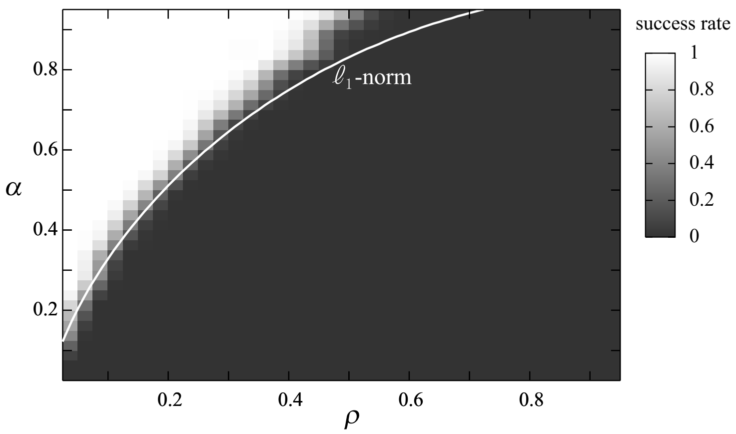

Figure 2. Profile of the success rate using the algorithm (12) with the partition ratio $\gamma^{(t)} = 1$. The area of the success region is almost the same as that of ℓ_1-norm minimization.

is almost the same as that of ℓ_1-norm. We stress that the algorithm proposed here will also be successful under the general matrix $\boldsymbol{F}$, as observed in Figure 4.

5.2. Convergence of the algorithm

Next, we examined the speed of convergence to the solution. We compared two algorithms: the MAP algorithm with step-dependent partition ratio (18), and the AMP (23). We set the parameters $N = 2000, M = 1000$ (compression ratio $\alpha = 0.5$) and $K = 200$ (fraction of nonzero entries $\rho = 0.1$). The rules for generating the matrix $\boldsymbol{F}$ and the original data $\boldsymbol{x}^0$ were the same as in Figures 1-3. For the parameter k, the exponential decay limit to zero was taken by multiplying 0.95 in each step for both algorithms. (Note that the original AMP uses the mean squared error for the update of k as elucidated before, which shows much better convergence. Here we adopt slow exponential update rule for both algorithms in the experiment for simpler experimental condition.) We conducted the reconstruction experiment 100 times and observed the behavior of the mean square error per data. The results are shown in Figure 5. It can be seen that the AMP exhibits a better performance than the MAP algorithm. From the relation between the MAP and the AMP algorithms, as discussed in the previous section, this difference

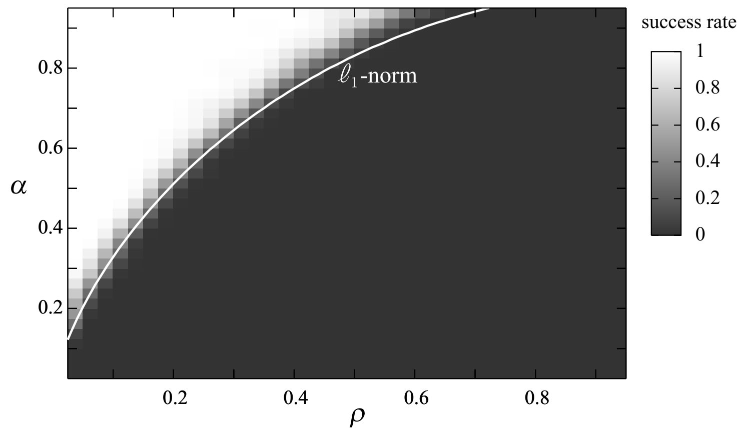

Figure 3. Profile of the success rate using the algorithm with step-dependent partition ratio $\gamma^{(t)}$ in (18). We can obtain a reconstruction threshold that is much closer to that of ℓ_1-norm minimization.

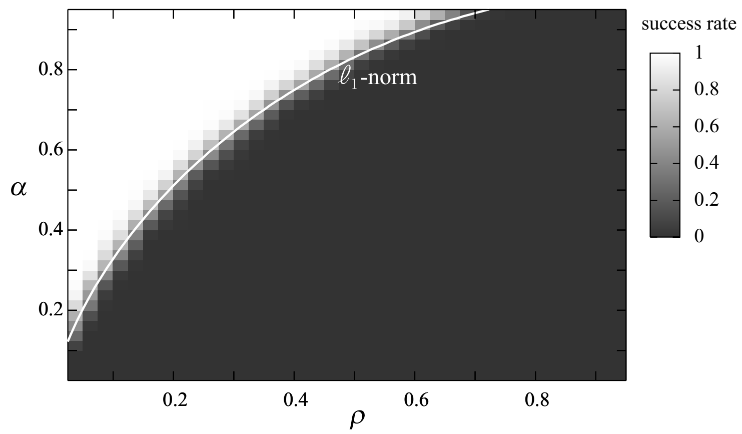

Figure 4. Profile of the success rate for a random matrix with a 10% nonzero matrix element. We use the algorithm with the constant partition ratio $\gamma^{(t)} = 1$. We can obtain a success region similar to that of ℓ_1-norm minimization also in this case.

in convergence speed might be understood from the introduction of the partition ratio to the residual error z_μ to improve the convergence in the AMP (as in (21)), whereas in the MAP algorithm in (12) this point is not taken into consideration. Therefore, it will be necessary to consider faster convergence of the residual error z_μ to improve the MAP algorithm.

6. Conclusion and perspective

In this article, we presented a methodology for constructing an algorithm using the MAP approach, discussed its relation with a known thresholding algorithm, and evaluated its performance through numerical experiments. We verified that, by designing the algorithm appropriately, almost the same reconstruction threshold as that of ℓ_1-norm minimization can be achieved. In the case of the i.i.d. random matrix $\boldsymbol{F}$, we clearly presented a viewpoint on the reason why the AMP has the same analytical expression of the reconstruction threshold as ℓ_1-norm minimization. The significant point is that we do not make any assumption for the matrix $\boldsymbol{F}$, and accordingly our algorithm's construction, whose computational cost is relatively low, is applicable to a general matrix.

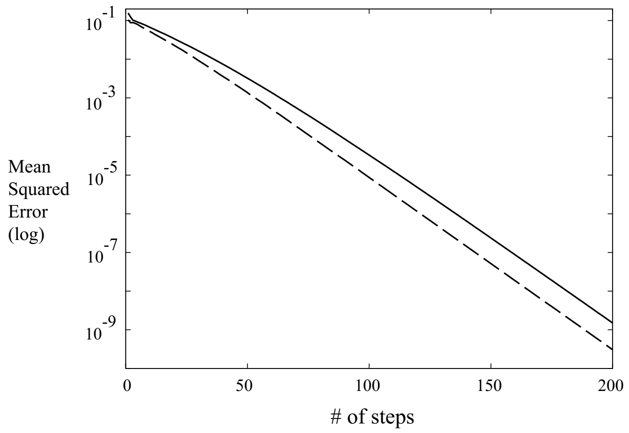

Figure 5. Convergence speed of the algorithm. We compare the convergence speed of the algorithms, the MAP with a step-dependent partition ratio (solid) and the AMP (broken). The behavior of the mean squared error per single data point is depicted. The AMP exhibits a better performance than the MAP in terms of convergence speed.

We also emphasized that it is still possible that faster convergence to the correct solution can be achieved by a more suitable design of the partition ratio. In the case of the random i.i.d. Gaussian matrix, the optimality of the algorithm is discussed in [12]. For a general matrix such as sparse or structured matrix, for which feasible algorithm is proposed and discussed in [15, 16, 17], future work will address a systematic method for designing an algorithm that will achieve faster convergence using the discussion presented in this paper.

Acknowledgments
This work is supported by KAKENHI Nos. 24700007 (KT), 22300003 and 25120013 (YK). YK also acknowledges the ELC project (Grant-in-Aid for Scientific Research on Innovative Areas MEXT Japan) for encouraging the research presented in this paper.

References

[1] Donoho D L 2006 *IEEE Trans. Inform. Theory* **52** 1289
[2] Candès E J and Tao T 2005 *IEEE Trans. Inform. Theory* **51** 4203
[3] Candès E J, Romberg J and Tao T 2006 *IEEE Trans. Inform. Theory* **52** 489
[4] Candès E J and Wakin M B 2008 *IEEE Signal Process. Mag.* **25** 21
[5] Elad M 2010 *Sparse and redundant representations* (New York: Springer)
[6] Tropp J A and Wright S J 2010 *Proc. of IEEE* **98** 948
[7] Donoho D L, Maleki A and Montanari A 2009 *Proc. Natl. Acad. Sci.* **106** 18914
[8] Pearl J 1988 *Probabilistic reasoning in intelligent systems: networks of plausible inference* (San Francisco: Morgan Kaufmann)
[9] Donoho D L and Tanner J 2005 *Proc. Natl. Acad. Soc.* **102** 9452
[10] Donoho D L, Maleki A and Montanari A 2010 *Proc. of Information Theory Workshop, Cairo* (*Preprint* arXiv:0911.4219)
[11] Donoho D L, Maleki A and Montanari A 2010 *Proc. of Information Theory Workshop, Cairo* (*Preprint* arXiv:0911.4222)
[12] Bayati M and Montanari A 2011 *IEEE Trans. Inform. Theory* **57** 764
[13] Kabashima Y, Wadayama T and Tanaka T 2009 *J. Stat. Mech.* L09003
[14] Tibshirani R 1996 *J. Royal Statist. Soc.* B **58** 267
[15] Kabashima Y and Wadayama T 2011 *Preprint* arXiv:1102.3220
[16] Krzakala F, Mézard M, Sausset F, Sun Y and Zdeborová L 2012 *Phys. Rev.* X **2** 021005
[17] Krzakala F, Mézard M, Sausset F, Sun Y and Zdeborová L 2012 *J. Stat Mech.* P08009
[18] Donoho D L and Tanner J 2009 *J. Amer. Math. Soc.* **22** 1
[19] Donoho D L and Tanner J 2009 *Phil. Trans. R. Soc.* A **367** 4273

[20] Bayati M, Lelarge M and Montanari A 2012 *Proc. of IEEE International Symposium on Information Theory* 1643 (*Preprint* arXiv:1207.7321)
[21] Takeda K and Kabashima Y 2012 *Proc. of 46th Annual Conference on Information Sciences and Systems* (*Preprint* arXiv:1203.6246)