

# 압축 센싱 신호 복구를 위한 AMP(Approximate Message Passing) 알고리즘 소개 및 성능 분석

백형호\*, 강재욱\*, 김기선\*, 이흥노<sup>o</sup>

## Introduction and Performance Analysis of Approximate Message Passing (AMP) for Compressed Sensing Signal Recovery

Hyeong-ho Baek\*, Jae-wook Kang\*, Ki-sun Kim\*, Heung-no Lee<sup>o</sup>

### 요 약

CS(Compressed Sensing)는 오늘날 신호 처리 영역에서 많은 주목을 받고 있는 이론 중의 하나이다. 이 CS 분야에서 효과적인 복구 알고리즘을 설계하는 것은 가장 큰 도전적 연구 중의 하나로 인식되고 있다. 이에 따라 다양한 복구 알고리즘이 많은 문헌을 통해서 제안 되었으며 최근에 Maleki와 Donoho에 의해 제안된 AMP(Approximation Message Passing) 알고리즘은 기존에 제시된 알고리즘에 비해 간단한 구조를 가지고 있지만 좋은 성능을 보여줌으로써 상당한 주목을 받고 있다. 기존의 (BP) Belief Propagation 알고리즘은 오직 희소(Sparse) 센싱 행렬에서만 좋은 성능을 보여 준 것에 반해, AMP 알고리즘은 밀집(Dense) 센싱 행렬에 기초를 둔 Belief Propagation 알고리즘임에도 불구하고 이와 비슷한 성능을 보여준다. 본 논문은 다양한 영역에서 AMP 알고리즘이 적용되기 위하여 이에 대한 지침 및 기존의 고전적 Message Passing 알고리즘과의 관계에 대해 분석하였다. 또한 기존의 알고리즘과의 비교 분석을 통해 AMP 알고리즘의 우수성을 제시하였다.

**Key Words** : Compressive sensing, AMP, Compressed sensing, Approximate Message Passing, CS recovery algorithm

### ABSTRACT

We introduce Approximate Message Passing (AMP) algorithm which is one of the efficient recovery algorithms in Compressive Sensing (CS) area. Recently, AMP algorithm has gained a lot of attention due to its good performance and yet simple structure. This paper provides not only a understanding of the AMP algorithm but its relationship with a classical (Sum-Product) Message Passing (MP) algorithm. Numerical experiments show that the AMP algorithm outperforms the classical MP algorithms in terms of time and phase transition.

### 1. 서 론

아날로그 신호를 디지털 신호로 변환하는 첫

단계는 시간 축에 대하여 신호를 양자화 하는 샘플링이다. 그 획득한 샘플로부터 신호를 손실 없이 복원하기 위해서는 대상 신호의 최대 주파수의 두 배

※ 이 논문은 2013년도 정부 (교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임 (중견연구자-도약연구사업, NO. 2013-035295)

◆ First Author : 광주과학기술원 정보통신공학부 INFONET 연구실, baek@gist.ac.kr, 학생회원

° Corresponding Author : 광주과학기술원 정보통신공학부 INFONET 연구실, heungno@gist.ac.kr 정회원

\* 광주과학기술원 정보통신공학부 CSNL 연구실, jwkkang@gist.ac.kr, kskim@gist.ac.kr, 중신회원  
 논문번호 : KICS2013-08-391, 접수일자 : 2013년 8월 31일, 최종논문접수일자 : 2013년 11월 5일

이상의 표본율(Sampling rate)로 샘플링을 수행해야 한다는 사실이 Nyquist 샘플링 이론에 의해서 알려져 있다. 하지만 많은 연구자들은 Nyquist 이론이 최적화된 신호 샘플링 방법인가에 대한 의심을 품어왔고 다음과 같은 질문을 던졌다. “신호가 포함하고 있는 정보량에 따라서 표본율을 조절할 수 있지 않을까?” 이러한 의문에 대한 대답을 제시하는 이론이 요구 되었고, 2006년에 Candes와 Donoho에 의해서 제안된 CS(Compressive Sensing)<sup>[1]</sup>이론은 그에 대한 대답을 다음과 같이 하였다. “만약 대상으로 하고 있는 신호가 충분히 희소(Sparse)하다면, 그 신호의 모든 정보를 손실 없이 획득(Aquisition) 하기 위해서는 신호의 길이  $N$ 보다 작은  $M$ 개의 압축된 샘플로 충분하다.”

위에서 소개한 CS 이론과 고전적인 Nyquist 샘플링 이론은 신호획득 구조를 근본적으로 다르게 한다. Nyquist 신호 획득에서는 먼저 대상 신호에서 Nyquist 샘플링 이론을 기반으로 정보를 추출하고 신호처리를 위한 정보량을 줄이기 위해서 다시 신호를 압축한다. 즉 “sample and compress”를 수행한다. 이와 대조적으로 CS 이론에서는 대상 신호로부터의 정보 추출과 압축을 동시에 수행한다. 즉 “compression on the fly”를 수행한다. 따라서 CS 기반의 신호 획득은 신호의 정보량을 빠르고 압축적으로 측정하고 보존할 수 있다.

하지만 명백히 Nyquist 샘플링 이론에 대항하는 CS 이론은 두 가지의 특별한 전제 조건이 필요하다. 먼저 이전에 존재하던 정보추출과 압축의 두 단계의 과정이 결합함으로써 인해 더욱 복잡하고 획기적인 신호 처리 시스템의 설계가 요구된다. 또한 복구 하고자 하는 신호가 희소 신호이어야 한다는 제한이 있다. 하지만 모든 신호는 희소한 표현이 가능한 적절한 도메인이 존재하기 때문에 획득될 신호가 희소하지 않다면 적절한 도메인으로 변환 후 희소한 신호로 표현될 수 있다<sup>[2]</sup>. 예를 들어 어떤 신호는 시간 축에서는 뾰뾰해 보이지만 주파수 축에서는 몇 개의 주파수 성분으로 표현할 수 있다. 따라서 두 번째 전제 조건은 적절한 도메인 변환 (위의 예제에서는 푸리에 변환)을 찾는다면 해결된다.

CS에서의 “compression on the fly” 신호획득 과정은 대상 신호를 임의의 센싱 행렬로 선형 투영(linear projection)과정으로 묘사한다. 그리고 그것은 다음과 같은 수학적 표현으로 나타낸다.

$$y_{(M \times 1)} = A_{(M \times N)} s_{(N \times 1)} \quad (1)$$

여기서  $s$ 는 대상 희소 신호 벡터,  $A$ 는  $M \times N$  ( $M < N$ )차원의 행렬, 그리고  $y$ 는 대상 신호의 정보를 압축적으로 보존하고 있는 측정 벡터(measurements)이다. 따라서 CS 복구 문제의 목표는 낮은 차원의 측정 벡터  $y$ 에서 높은 차원의 희소 신호 벡터  $s$ 를 복구하는 것이다. 이 희소 신호  $s$ 를 복원하는 문제는 underdetermined system에서 해를 찾는 문제와 같고 일반적으로 무수히 많은 해를 가진다. 하지만 식 (1)의 해를 구하기 위해서 우리가 가지고 있는 대상 신호가 희소하다는 조건을 이용하고 센싱 행렬  $A$ 가 특정한 조건을 만족한다면 유일한 해를 구할 수 있다는 사실이 이론적으로 증명되었다<sup>[2]</sup>. 그리고 그 후에 그 유일한 희소 해  $\hat{s}$ 를 구하기 위한 실용적인 신호복구 알고리즘이 convex optimization<sup>[3]</sup>, linear programming<sup>[1]</sup>, bases decomposition<sup>[4]</sup> 등의 다양한 접근 방법을 통해 제안되었다. 그 중 최근에 제안된 graphical modeling<sup>[5]</sup> 기법에 기초를 둔 AMP(Approximate Message Passing) 알고리즘<sup>[3]</sup>은 간단한 구조에 비해 놀라운 성능을 보여주기에 학계에서 많은 주목을 받고 있다. 하지만 불행히도 이 AMP 알고리즘이 다양한 영역에 적용 및 응용될 수 있음에도 불구하고 국내에서 이를 소개한 논문이 미비하다. 따라서 AMP 알고리즘의 기본적인 이해를 돕고 이를 응용하기 위한 지침은 중요한 역할을 할 수 있다. 본 논문에서는 [6]에 추가적인 내용을 덧붙여 고전적 Sum-Product Message Passing<sup>[5]</sup> 알고리즘에서 CS 문제를 해결하기 위해 제안된 AMP 알고리즘을 단계별로 유도하여 폭넓은 이해를 돕는다. 그리고 BHT-BP<sup>[7]</sup>와 SuPrEM<sup>[8]</sup>과 같이 다른 방식으로 MP를 구현한 희소 신호 복구 알고리즘과 비교하여 AMP 알고리즘의 저 복잡도 특성을 실험적으로 보인다. 또한 희소 신호 복구를 대해서 가장 널리 알려진 알고리즘인 Lasso<sup>[9]</sup> 그리고 OMP<sup>[10]</sup>와의 성능 비교를 통해서 AMP 알고리즘의 우수성을 입증한다.

## II. 본 론

### 2.1. 고전적 Sum-Product Message Passing 알고리즘으로부터 AMP 알고리즘의 유도

본 장에서는 고전적 MP(Message Passing) 알고리즘으로부터 AMP 알고리즘으로의 심도있는 수학적 유도 과정을 단계별로 보여준다. 이에 앞서 본

논문에서 이용하는 표기법 및 MP 알고리즘에서 이용하는 factor 그래프와 메시지에 대해 정의한다.

Factor graph의 예는 그림 1과 같다.

- 1)  $i$ 와  $j$ 는 각각  $j \in [M] := \{1, 2, \dots, M\}$ ,  $i \in [N] := \{1, 2, \dots, N\}$ 인 인덱스를 의미한다.
- 2) 센싱 행렬의  $j$ 번째 행,  $i$ 번째 열의 요소는  $A_{ji}$ 로 표기한다.
- 3) 벡터  $\mathbf{s}, \mathbf{y}$ 의 요소는 각각  $s_i, y_j$ 로 표기한다.
- 4) factor 그래프  $G := (V, F, E)$ 는 변수 노드 집합(variable node)  $V := [N]$ , 측정 노드 집합(measurement node)  $F := [M]$ 와 edge  $E := V \times F = \{(i, j) : i \in [N], j \in [M]\}$ 를 가지는 완전 이분 그래프(complete bipartite graph)이다. 그리고 edge에 의해 연결된 노드 사이에서 주고받는 메시지는 각각 VM(Variable to Measurement) 메시지  $\{v_{i \rightarrow j}\}_{i \in V, j \in F}$  MV(Measurement to Variable) 메시지  $\{\hat{v}_{j \rightarrow i}\}_{j \in F, i \in V}$ 로 나타낸다.

2.1.1. MP 알고리즘을 CS 문제에 적용

이전의 고전적 MP 알고리즘<sup>5)</sup>으로부터 AMP 알

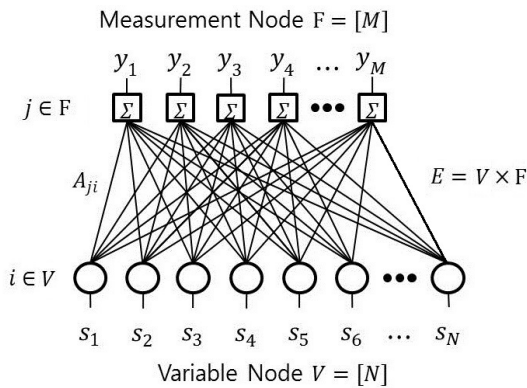


그림 1. Factor 그래프  
Fig. 1. Factor graph

고리즘을 유도하는 가장 첫 번째 단계는 CS 문제에 대한 회소 신호 벡터  $\mathbf{s}$ 에 대한 Sum-Product 알고리즘을 구성하는 것이다. 이를 위해서는 센싱 행렬  $A$ 와 측정 벡터  $\mathbf{y}$ 가 주어졌을 때, 모든 변수  $\{s_i\}_{i \in [N]}$ 의 joint pdf를 설계해야 한다. 여기서 우리는 신호 벡터  $\mathbf{s}$ 의 회소 특성과 측정값이  $\mathbf{y}$ 라는 점을 이용하여 joint pdf를 식 (2)와 같이 설계할 수 있다.

$$p(\mathbf{s}) = \frac{1}{C} \prod_{i=1}^N \exp(-\beta |s_i|) \prod_{j=1}^M \delta(y_j = A_j \mathbf{s}) \quad (2)$$

여기서  $C$ 는 확률 정규화 상수 이고,  $\delta(\cdot)$ 는 Dirac delta function을 의미한다. 따라서 식 (2)를 바탕으로 하여 CS 문제에 대한 고전적 MP 알고리즘은 표 1의 Algorithm 1과 같이 표현된다. 그런데 여기에서 AMP 알고리즘은 측정 노드 집합과 변수 노드 집합의 요소들이 edge에 의해서 뻗뻗히 연결되어 있는 Factor 그래프 즉, 밀집 그래프(dense graph) 환경을 토대로 하여 설계되었다. 이에 반해 고전적 MP 알고리즘과 BHT-BP<sup>7)</sup>와 SuPrEM<sup>8)</sup>과 같이 다른 방식으로 MP를 구현한 회소 신호 복구 알고리즘은 측정 노드 집합과 변수 노드 집합의 요소들이 edge에 의해서 드문하게 연결되어 있는 Factor 그래프 즉, 회소 그래프(sparse graph) 환경에서 설계 되었다. 따라서 AMP 알고리즘의 유도를 위하여 측정 노드 집합과 변수 노드 집합의 연결을 나타내는 센싱 행렬  $A$ 의 요소  $A_{ji}$ 의 분포를  $\mathcal{M}(0, 1/M)$ 로 가정한다.

2.1.2. 파라미터 형태로의 메시지 변환

표 1의 Algorithm 1에서 주어진 메시지의 형태는 실수 범위의 확률 분포이기 때문에 복잡한 계산을 요구한다. 이러한 문제를 해결하기 위하여 복잡한 확률 분포의 형태를 간단한 파라미터 메시지로 변환하도록 한다. 그런데 여기서 자연스럽게 어떠한 파라미터를 메시지로 정할 것인가 하는 의문이 생긴다. 이 의문은  $N$ 이 상당히 크고 밀집 그래프 환경에서 중심 극한 정리에 의해 메시지가 Gaussian 분포로 근사됨을 이용하면 답을 얻을 수 있다. 왜냐하면 Gaussian 분포는 우리가 아는 바와 같이 평균과 분산을 이용해서 정확히 표현할 수 있기 때문이다. 따라서 메시지로 이용할 파라미터는 평균과 분산이 된다<sup>11)</sup>.

그러면 위에서 언급한 내용을 바탕으로 하여 표 1의 Algorithm 1에서의 MV 메시지  $\hat{v}_{j \rightarrow i}$ 에 대해서 고려해 보자. 하나의 MV 메시지  $\hat{v}_{j \rightarrow i}$ 는 식 (3)의  $Z$ 와 같은 확률 변수를 가지는 확률 분포이다.

$$Z := y_j - \sum_{i' \neq i} A_{ji'} s_{i'} \quad (3)$$

따라서  $Z$ 의 확률 분포는 식 (3)에서 보이는 것처럼  $N-1$ 개의 확률 변수  $s_1, s_2, \dots, s_{i-1}, s_{i+1}, s_N$ 들에

의해서 결정된다. 그리고  $N$ 이 상당히 크고 밀집 그래프 환경에서 중심 극한 정리에 의해서  $Z$ 는 Gaussian 분포로 근사화 된다. 따라서 MV 메시지  $\hat{v}_{j \rightarrow i}$ 는  $Z$ 의 평균과 분산으로 대체할 수 있다. 그런데 확률 변수  $s_i$ 의 확률 분포는 측정 노드  $j$ 와 iteration 인덱스  $t$ 에 따라 달라진다. 이러한 사실을 적용하여  $s_i$ 의 평균과 분산을 각각  $x_{i \rightarrow j}^t$ 와  $\hat{\tau}_{i \rightarrow j}^t / \beta$ 라 하면 확률 변수  $Z$ 의 평균과 분산은 아래와 같이 표현된다. 여기에서  $x_{i \rightarrow j}$ 와  $\hat{\tau}_{i \rightarrow j}$ 는 위에서 정의한 VM 메시지와 비슷하게  $\{x_{i \rightarrow j}\}_{i \in V, j \in F}$ ,  $\{\hat{\tau}_{i \rightarrow j}\}_{i \in V, j \in F}$ 로 표기한다. 또한 벡터  $\mathbf{x}, \boldsymbol{\tau}$ 의 요소는 각각  $x_i, \tau_i$ 로 표기한다.

$$z_{j \rightarrow i}^t := E(Z) = y_j - \sum_{i' \neq i} A_{ji'} x_{i' \rightarrow j}^t \quad (4)$$

$$\text{Var}(Z) = \sum_{i' \neq i} (A_{ji'})^2 \frac{\hat{\tau}_{i' \rightarrow j}^t}{\beta} \quad (5)$$

$$\tau_{j \rightarrow i}^t := \sum_{i' \neq i} (A_{ji'})^2 \hat{\tau}_{i' \rightarrow j}^t \quad (6)$$

여기서  $t$ 는 알고리즘의 iteration 인덱스이다.

다음으로 VM 메시지  $v_{i \rightarrow j}$ 에 대해서 고려해 보자. VM 메시지  $v_{i \rightarrow j}$ 는 표 1의 Algorithm 1에 나타낸 바와 같이 첫 번째 항 Laplacian과 두 번째 항  $\prod_{j' \neq j} \hat{v}_{j' \rightarrow i}$ 의 곱이다. 여기서 MV 메시지  $\hat{v}_{j \rightarrow i}$ 는 위에 언급한 것과 같이 Gaussian 분포를 가진다. 따라서 product term  $\prod_{j' \neq j} \hat{v}_{j' \rightarrow i}$ 는 Gaussian 분포를 가지는  $M-1$ 개의 확률 변수의 곱이다. 여기서 우리가 아는 바와 같이 Gaussian 확률 변수의 곱의 분포는 또 다른 Gaussian 이 된다. 즉, VM 메시지  $v_{i \rightarrow j}$ 의 product term  $\prod_{j' \neq j} \hat{v}_{j' \rightarrow i}$ 은 Gaussian 분포를 가진다고 할 수 있다. 이러한 사실을 종합하면, VM 메시지  $v_{i \rightarrow j}$ 는 Laplacian과 Gaussian의 곱인 아래의 분포로 표현된다.

$$f_\beta(s; a, b) := 1/z_\beta(a, b) \times \exp\left\{-\beta|s| - \frac{\beta}{2b}(s-a)^2\right\} \quad (7)$$

여기에서  $z_\beta(a, b)$ 는 정규화 상수를 의미하고 파라미터  $a, b$ 에 의해서 변화한다. VM 메시지  $v_{i \rightarrow j}$ 의 정확한 분포를 구하기 위하여 식 (7)의 파라미터  $a$

와  $b$ 를 결정해야 한다.  $a$ 와  $b$ 는  $\prod_{j' \neq j} \hat{v}_{j' \rightarrow i}$ 을 계산함으로써  $a = \sum_{j' \neq j} A_{ji'} z_{j' \rightarrow i}^t$ ,  $b = \tau_{j \rightarrow i}^t$ 로 구해진다.<sup>[10]</sup> 따라서 VM 메시지  $v_{i \rightarrow j}$ 의 분포는 아래와 같다.

$$v_{i \rightarrow j}^{t+1} = f_\beta\left(s_i; \sum_{j' \neq j} A_{ji'} z_{j' \rightarrow i}^t, \tau_{j \rightarrow i}^t\right) \quad (8)$$

우리는 이제 파라미터 메시지로 이용할 VM 메시지  $v_{i \rightarrow j}$ 의 평균과 분산을 구하는데 집중한다. VM 메시지  $v_{i \rightarrow j}$ 의 평균과 분산을 간단하게 표현하기 위해  $f_\beta(s_i; a, b)$ 의 평균과 분산을 아래와 같이 정의한다.

$$F_\beta(a; b) := E_{f_\beta(s; a, b)}(Z) \quad (9)$$

$$G_\beta(a; b) := \text{Var}_{f_\beta(s; a, b)}(Z) \quad (10)$$

위에서 표기한 식 (9)와 (10)을 이용하여 VM 메시지  $v_{i \rightarrow j}$ 의 평균  $x_{i \rightarrow j}^{t+1}$ 와 분산  $\hat{\tau}_{i \rightarrow j}^t$ 을 나타내면 아래와 같다.

$$x_{i \rightarrow j}^{t+1} = F_\beta\left(\sum_{j' \neq j} A_{ji'} z_{j' \rightarrow i}^t; \tau_{j \rightarrow i}^t\right) \quad (11)$$

$$\hat{\tau}_{i \rightarrow j}^t = \beta G_\beta\left(\sum_{j' \neq j} A_{ji'} z_{j' \rightarrow i}^t; \tau_{j \rightarrow i}^t\right) \quad (12)$$

한편  $A_{ji}$ 의 각 요소의 분포가  $\mathcal{M}(0, 1/M)$ 라는 가정에 의해서 식 (6)의  $\tau_{j \rightarrow i}^t$ 는 변수 노드와 측정 노드 사이에서 메시지의 교환 없이 각 iteration 마다 갱신되는 값이 됨을 주목하자.  $A_{ji}$ 의 분포에 의해서  $A_{ji}^2 = O(1/M)$ 이기 때문에  $N, M$ 이 상당히 크고 밀집 그래프 환경에서는 식 (6)  $\sum_{i' \neq i} (A_{ji'})^2 \hat{\tau}_{i' \rightarrow j}^t$ 에서 제외된 하나의 메시지  $(A_{ji})^2 \hat{\tau}_{i \rightarrow j}^t$ 에 대한 영향은 미비하다. 다시 말하면, 식 (6)의  $\tau_{j \rightarrow i}^t$ 는 변수 노드와 측정 노드 모두에 대하여 독립적이라고 할 수 있다. 즉, 식 (13)과 같이 표현할 수 있다.

$$\tau_{j \rightarrow i}^t = \tau^t \quad \forall i, j. \quad (13)$$

식 (13)은 위에서 정의한 factor 그래프에서 메시지의 교환을 의미하는 edge가 없는 상태로 나타낼 수 있고 이를 간단하게 edge 독립이라고 표현한다. 즉, 변수 노드와 측정 노드 사이에서 메시지의 교환 없이 각 iteration 마다 갱신되는 값  $\tau^t$ 는 edge 독립

이고 그 값은 식 (12)를 식 (6)에 대입하여 유도된 아래의 식을 통해 구할 수 있다.

$$\tau^t = \sum_{i' \neq i} (A_{ji'})^2 \beta G_\beta \left( \sum_{j \neq j'} A_{ji'} z_{j \rightarrow i}^t ; \tau^{t-1} \right) \quad (14)$$

또한  $A_{ji}$ 의 분포를 이용하면  $E(A_{ji}^2) = 1/M$ 이다. 따라서 식 (14)에 이를 적용한 표현은 아래로 근사화 된다.

$$\tau^{t+1} = \frac{\beta}{M} \sum_i G_\beta \left( \sum_j A_{ji} z_{j \rightarrow i}^t ; \tau^t \right) \quad (15)$$

### 2.1.3. $\beta \rightarrow \infty$ 을 이용한 메시지의 간략화

식 (2)의  $\beta$  값을 무한대로 설정하면 식 (1)의 해 (solution) 주위로 확률 분포의 질량(mass)이 집중된다. 이것은 최대 최소 신호의 값을 찾아주는  $l_1$  norm 알고리즘과 일치한다. 따라서  $\beta \rightarrow \infty$ 의 설계는 최소 신호를 복구하는 현재의 문제에 적합하다. 또한 흥미롭게도  $\beta \rightarrow \infty$ 의 설계는 식 (9), (10)의  $F_\beta$ 와  $G_\beta$ 을 간략화할 수 있는 여지를 마련해 준다.

먼저  $\beta \rightarrow \infty$ 인 경우 식 (7)의 평균 값  $F_\beta$ 에 대해 고려해 보자.  $F_\beta$ 는 적분 영역에 포함된 피적분항 혹은 식 (7)의 지수의 최대값에 의해 주도 된다.

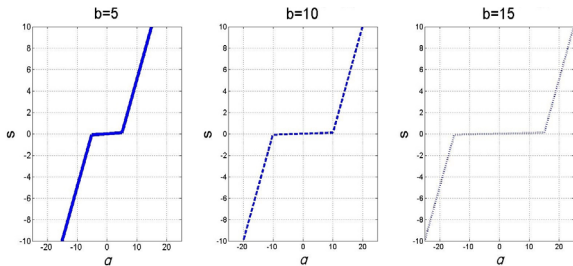


그림 2. 변수 a와 b에 따른  $\operatorname{argmin}_s \left\{ |s| + \frac{1}{2b}(s-a)^2 \right\}$   
 Fig. 2.  $\operatorname{argmin}_s \left\{ |s| + \frac{1}{2b}(s-a)^2 \right\}$  according to variable a and b

즉, 아래의 식 (16)으로 표현된다.

$$\begin{aligned} F_\beta(a,b) &= \int_{-\infty}^{\infty} s f_\beta(s;a,b) ds \\ &= \int_{-\infty}^{\infty} s \frac{1}{z_\beta(a,b)} \\ &\quad \times \exp \left\{ -\beta |s| - \frac{\beta}{2b}(s-a)^2 \right\} ds \quad (16) \\ &\approx \operatorname{argmin}_s \left\{ |s| + \frac{1}{2b}(s-a)^2 \right\} \end{aligned}$$

여기서  $F_\beta(a,b)$ 의 값인 식  $|s| + \frac{1}{2b}(s-a)^2$ 의 최소값은 s에 의해서 미분된 값이 0이 되는 지점이다. 이를 수식으로 표현하면 아래와 같이 나타난다.

$$\begin{aligned} \frac{d}{ds} \left\{ |s| + \frac{1}{2b}(s-a)^2 \right\} &= 0 \\ \frac{1}{b}(s-a) + \operatorname{sgn}(s) &= 0 \end{aligned} \quad (17)$$

이 식 (17)을 이용하여 변수 a와 b에 따른 식  $|s| + \frac{1}{2b}(s-a)^2$ 을 최소로 만드는 s는 그림 2와 같다. 이것은 아래에 정의된 soft threshold function의 파라미터  $x = a$ 이고  $\theta = b$ 인 경우와 일치한다.

$$\eta(x;\theta) \equiv \begin{cases} x-\theta & \text{if } x > \theta \\ 0 & \text{if } -\theta \leq x \leq \theta \\ x+\theta & \text{if } x < -\theta \end{cases} \quad (18)$$

다시 말하면  $\beta \rightarrow \infty$ 인 경우 식 (8)에서 정의한  $F_\beta$ 는 soft threshold function으로 대체할 수 있다. 이러한 사실을 이용하면 VM 메시지  $v_{i \rightarrow j}$ 의 평균값인 식 (11)은 soft threshold function으로 아래와 같이 간단하게 계산된다.

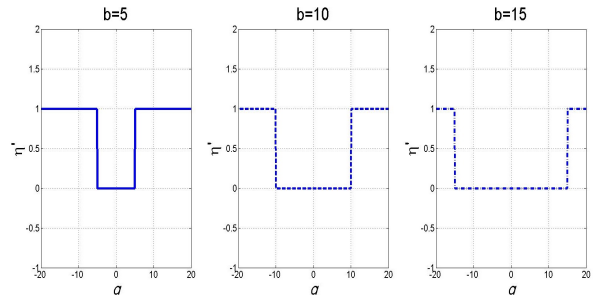


그림 3. 변수 a와 b에 따른  $\eta'(a;b)$  함수  
 Fig. 3.  $\eta'(a;b)$  function according to variable a and b

$$x_{i \rightarrow j}^{t+1} = \eta \left( \sum_{j' \neq j} A_{ji'} z_{j' \rightarrow i}^t ; \theta_t \right) \quad (19)$$

여기서  $\theta_t = \alpha \sqrt{\tau^t}$ 이고  $\alpha$ 는 soft threshold function으로 대체 하면서 생기는 오차를 조정하는 값이다.

다음으로  $\beta \rightarrow \infty$ 인 경우 식 (7)의 분산 값  $G_\beta$ 에 대해 고려해 보자. 분산 값  $G_\beta$ 는 VM 메시지  $v_{i \rightarrow j}$ 의 평균값에 따라서 두 가지의 경우가 발생한다. 만

약 식 (19)에서 계산된 평균값이 0이라면 대략적으로 Laplacian 분포가 우세하다 할 수 있기 때문에 분산은  $2/\beta^2$ 이 된다. 이는  $\beta \rightarrow \infty$ 인 경우, 근사적으로 0이 된다. 반면에 평균값이 0이 아니라면 대략적으로 Gaussian 분포가 우세하여 분산은  $\lim_{\beta \rightarrow \infty} \frac{b}{\beta}$ 가 된다. 따라서 식 (19)에서 계산된 평균값이 0이 되는 부분( $-b \leq a \leq b$ )에서는 식 (7)의 분산 값  $G_\beta$ 가 0이 되고, 평균값이 0이 아닌 부분( $-b > a, b < a$ )에서는  $\lim_{\beta \rightarrow \infty} \frac{b}{\beta}$ 가 된다. 이것은 그림 3에 나타낸 soft threshold function의 미분  $\eta'(a;b)$ 을 이용하여 아래와 같이 간략하게 표현할 수 있다.

$$\lim_{\beta \rightarrow \infty} G_\beta(a;b) = \lim_{\beta \rightarrow \infty} \frac{b}{\beta} \eta'(a;b) \quad (20)$$

따라서 식 (15)의  $G_\beta$ 를 식 (20)에서와 같이 soft threshold function을 미분한 함수  $\eta'(a;b)$ 로 대체하면 각 iteration마다 갱신되는  $\tau_{t+1}$ 를 아래와 같이 간단하게 계산할 수 있다.

$$\tau^{t+1} = \frac{1}{M} \sum_i^N \hat{\tau}^t \eta' \left( \sum_j^M A_{ji} z_{j \rightarrow i}^t ; \theta_t \right) \quad (21)$$

여기서  $\delta = M/N$ 이라 한다면, 식 (21)은 다음과 같다.

$$\tau^{t+1} = \frac{\hat{\tau}^t}{N\delta} \sum_i^N \eta' \left( \sum_j^M A_{ji} z_{j \rightarrow i}^t ; \theta_t \right) \quad (22)$$

따라서  $\beta \rightarrow \infty$ 일 경우 이 전 항에서 유도한 파라미터 passing 알고리즘은 표 1의 Algorithm 2와 같이 간략화 된다.

#### 2.1.4. MP 알고리즘에서 AMP로의 유도

지금까지 유도된 알고리즘은 처음보다 많이 간략화 되었지만, 아직까지 각 edge에서 독립된 총  $2MN$ 의 메시지를 주고받아야 하기 때문에  $N$ 이 크다면 각 iteration 마다 계산량이 상당히 많을 수밖에 없다. 본 장에서는 이러한 문제점을 해결하기 위하여 메시지 추정을 이용한 계산량 감소 방안을 제공한다.

이를 위하여 먼저 하나의 측정 노드에서  $N$ 개의

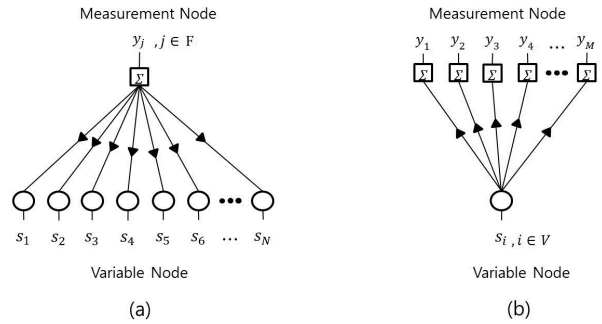


그림 4. (a) 하나의 측정 노드에서 이와 연결된 이웃 변수 노드들의 메시지를 묘사한 factor graph (b) 하나의 변수 노드에서 이와 연결된 이웃 측정 노드들의 메시지를 묘사한 factor 그래프

Fig. 4. (a) Factor graph describing Measurement to Variable (MV) Message (b) Factor graph describing Variable to Measurement (VM) Message

변수 노드들의 메시지 전달을 나타낸 그림 4(a)를 참조한다. 여기서 우리는 계산량을 감소시키기 위하여 각 변수 노드에 따라 다른 메시지를 보냈던 이전의 알고리즘과 달리 모든 변수 노드에 상관없이 똑같은 MV 메시지  $z_j^t$ 를 보낸다고 가정한다. 하지만 실제로 메시지는 각 변수 노드에 영향을 받기 때문에 오차가 발생한다. 그러므로 오차를 정정하는 항  $\Delta z_{j \rightarrow i}^t$ 을 추가하여 측정 노드에서 변수 노드로 보내는 MV 메시지  $z_{j \rightarrow i}^t$ 를 아래와 같이 가정할 수 있다.

$$z_{j \rightarrow i}^t = z_j^t + \Delta z_{j \rightarrow i}^t \quad (23)$$

그림 4 (b)의 변수 노드에서 측정 노드로 보내는 VM 메시지  $x_{i \rightarrow j}^t$ 도 마찬가지로의 방법을 이용할 수 있다. 따라서 VM 메시지  $x_{i \rightarrow j}^t$ 는 아래와 같이 가정한다.

$$x_{i \rightarrow j}^t = x_i^t + \Delta x_{i \rightarrow j}^t \quad (24)$$

그러면 이제 위에서 가정된 식을 이용하여 표 1의 Algorithm 2를 변형한다. 식 (24)  $x_{i \rightarrow j}^t$ 를 VM 메시지  $z_{j \rightarrow i}^t$ 에 적용하면 아래와 같은 결과를 얻는다.

$$z_{j \rightarrow i}^t = y_j - \underbrace{\sum_{i'=1}^N A_{ji'} x_{i' \rightarrow j}^t}_{z_j^t} - \sum_{i'=1}^N A_{ji'} \Delta x_{i' \rightarrow j}^t + \frac{A_{ji} x_i^t}{\Delta z_{j \rightarrow i}^t} \quad (25)$$

다음으로 추정된 식 (23)  $z_{j \rightarrow i}^t$ 을 표 1의 Algorithm 2의 VM 메시지  $x_{i \rightarrow j}^{t+1}$ 에 적용하면 다음과 같다.

$$x_{i \rightarrow j}^{t+1} = \eta \left( \sum_{j'=1}^M A_{j'i} z_{j' \rightarrow i}^t + \sum_{j'=1}^M A_{j'i} \Delta z_{j' \rightarrow i}^t - A_{ji} z_j^t ; \theta_t \right) \quad (26)$$

식 (26)을 더욱 간략히 하기 위해서 1차 Taylor 근사를 적용하면 아래와 같이 표현된다.

$$x_{i \rightarrow j}^{t+1} = \eta \left( \sum_{j'=1}^M A_{j'i} z_{j' \rightarrow i}^t + \sum_{j'=1}^M A_{j'i} \Delta z_{j' \rightarrow i}^t ; \theta_t \right) - A_{ji} z_j^t \eta' \left( \underbrace{\sum_{j'=1}^M A_{j'i} z_{j' \rightarrow i}^t + \sum_{j'=1}^M A_{j'i} \Delta z_{j' \rightarrow i}^t}_{\Delta x_{i \rightarrow j}^t} ; \theta_t \right) \quad (27)$$

여기서 이어서 식 (25)의  $\Delta z_{j \rightarrow i}^t = A_{ji} x_i^t$ 을 식 (27)의  $x_i^t$ 에 대입하면 다음과 같은 결과를 얻는다.

$$x_i^{t+1} = \eta \left( \sum_{j'=1}^M A_{j'i} z_{j' \rightarrow i}^t + \sum_{j'=1}^M A_{j'i} \Delta z_{j' \rightarrow i}^t ; \theta_t \right) = \eta \left( \sum_{j'=1}^M A_{j'i} z_{j' \rightarrow i}^t + x_i^t ; \theta_t \right) \quad (28)$$

같은 방법으로 식 (27)의  $\Delta x_{i \rightarrow j}^t$ 를 식 (25)의  $z_j^t$ 에 대입하면 다음과 같다.

$$z_j^t = y_j - \sum_{i'=1}^N A_{ji'} x_{i'}^t - \sum_{i'=1}^N A_{ji'} \Delta x_{i' \rightarrow j}^t = y_j - \sum_{i'=1}^N A_{ji'} x_{i'}^t + \frac{z_j^{t-1}}{\delta} \left\langle \eta' \left( \sum_{j'=1}^M A_{j'i'} z_{j' \rightarrow i}^{t-1} + x_i^{t-1} ; \theta_t \right) \right\rangle \quad (29)$$

Onsager Term

여기서  $\delta = M/N$ ,  $\langle \cdot \rangle = \sum_{i=1}^N (\cdot) / N$ 을 의미한다.

위 식에서  $\langle \eta'(\mathbf{x}; \theta) \rangle$ 는 임의의 임계값  $\theta$ 보다 크거나  $-\theta$ 보다 작은  $\mathbf{x}$  요소의 비율을 나타낸다. 예를 들어  $\theta$ 가 0.2이고  $\mathbf{x}$ 의 길이  $N=10$ 이라 하자. 그러면 soft threshold function을 미분한  $\eta'(\cdot)$ 가 그 10개의 값 중에서 0.2보다 크고 -0.2보다 작

은 값은 1로, 나머지는 0으로 만든다. 그리고  $\langle \cdot \rangle$  함수에 의해서 전체의 요소들 중에서 그 요소들의 비율이 구해진다. 여기서 -0.2보다 작고 0.2보다 큰 요소의 개수가 3개라고 한다면  $\langle \eta'(\mathbf{x}; \theta) \rangle$  값은 0.3이 된다. 즉,  $\mathbf{x}$ 의 요소 중에서 임계값 0.2보다 크고 -0.2보다 작은 요소는 전체  $\mathbf{x}$ 의 요소 중에서 30%라는 의미이다.

표 1. CS문제에 대한 고전적 Message Passing, Parameter Passing, AMP 알고리즘  
Table 1. Classical Message Passing, Parameter Passing, AMP Algorithm for CS problem.

#### Algorithm 1 Classical Message Passing iteration

1) Variable to Measurement Message

$$v_{i \rightarrow j}^{t+1}(s_i) = \exp(-\beta |s_i|) \prod_{j' \neq j} \hat{v}_{j' \rightarrow i}^t(s_i)$$

2) Measurement to Variable Message

$$\hat{v}_{j \rightarrow i}^t(s_i) = \int_{\mathbf{s} \setminus s_i} \delta(y_i = \mathbf{A} \mathbf{s}) \prod_{i' \neq i} v_{i' \rightarrow j}^t$$

#### Algorithm 2 Parameter Passing iteration

1) Variable to Measurement Message

$$x_{i \rightarrow j}^{t+1} = \eta \left( \sum_{j' \neq j} A_{j'i} z_{j' \rightarrow i}^t ; \theta_t \right)$$

2) Measurement to Variable Message

$$z_{j \rightarrow i}^t = y_j - \sum_{i' \neq i} A_{ji'} x_{i'}^t$$

3) Variance Update

$$t^{t+1} = \frac{t}{N\delta} \sum_i \eta' \left( \sum_{j' \neq j} A_{j'i} z_{j' \rightarrow i}^t ; \theta_t \right)$$

#### Algorithm 3 AMP iteration

1) Variable Node Update

$$\mathbf{x}^{t+1} = \eta(\mathbf{A}^T \mathbf{z}^t + \mathbf{x}^t ; \theta_t)$$

2) Measurement Node Update

$$\mathbf{z}^t = \mathbf{y} - \mathbf{A} \mathbf{x}^t + \frac{\mathbf{z}^{t-1}}{\delta} \underbrace{\langle \eta'(\mathbf{A}^T \mathbf{z}^{t-1} + \mathbf{x}^{t-1} ; \theta_t) \rangle}_{\text{Onsager Term}}$$

3) Variance Update

$$t^{t+1} = \frac{t}{\delta} \langle \eta'(\mathbf{A}^T \mathbf{z}^t + \mathbf{x}^{t+1} ; \theta_t) \rangle$$

결론적으로 매 iteration 마다 각 노드는 자신과 연결된 이웃 노드로 똑같은 메시지인 식 (28)과 (29)를 보낸다. 따라서 각 edge에서 독립된 총  $N+M$ 개의 메시지를 주고받기 때문에 계산량이 상당히 감소하였음을 확인할 수 있다.

마지막으로 모든 변수 노드와 측정 노드에 대하여 식 (28)과 (29)을 표현하기 위하여 벡터를 이용하면 최종적으로 유도된 AMP 알고리즘인 표 1의 Algorithm 3을 얻을 수 있다.

## 2.2. AMP의 특징

본 장에서는 AMP 알고리즘의 동작에 대해서 분

석한다. 특히 State evolution을 이용한 AMP 복구 성능에 대한 예측, Minimax 임계값 선택 그리고 Lasso와 AMP와의 동등성 대한 내용을 다룬다.

2.2.1. MP의 성능 추적을 위한 State Evolution

State evolution (SE)[12]은 AMP의 성능을 알고리즘 반복 횟수  $t$ 에 따라서 정확하게 묘사한다. 여기서 ‘state’이라는 용어는 AMP의 수렴이 하나의 파라미터 (또는 하나의 상태)로 추적된다는 것을 의미한다. 그리고 ‘evolution’이라는 용어는 그 AMP의 상태 (state), 즉  $AMP_t(x^t, z^t, \sigma_{SE}^t)$ 라 할 때, 반복  $t$ 에 따라 진화하며  $t \rightarrow \infty$ 에서 하나의 고정점 (fixed-point)으로 수렴한다는 것을 가리킨다. 여기서 고정점이란  $t \rightarrow \infty$ 로 인해서 알고리즘이 수렴하는 상태 즉

$$(\mathbf{x}^*, \mathbf{z}^*, \sigma_{SE}^*) := AMP_{t \rightarrow \infty}(\mathbf{x}^t, \mathbf{z}^t, \sigma_{SE}^t) \quad (30)$$

을 말한다. 본 AMP 알고리즘의 SE 분석에서는 (1)에서 제공된 선형 신호 모델에 가산 잡음  $w \sim N(0, \sigma_w^2)$ 을 추가한 다음과 같은 모델을 고려한다.

$$\mathbf{y}_{(M \times 1)} = \mathbf{A}_{(M \times N)} \mathbf{x}_{(N \times 1)} + \mathbf{w}_{(M \times 1)} \quad (31)$$

여기서  $\mathbf{x}_{(N \times 1)}$ 는  $x_i = E[s_i | \mathbf{y}]$  for  $i \in V$ 로 구성된 벡터로써 측정벡터  $\mathbf{y}$ 가 주어졌을 때 실제 회소 신호  $\mathbf{s}$ 의 조건부 평균값이다. SE를 이용한 AMP의 성능 추적이 수학적으로 견고하기 위해서는 다음과 같은 조건이 만족되어야 한다.

- 1) 센싱 행렬  $\mathbf{A}$ 는 Gaussian 행렬로 그 원소는 i.i.d. 하게  $\mathcal{N}(0, 1/M)$  분포를 따른다.
- 2) 목적 신호  $\mathbf{x}$ 의 경험적 2차 모멘트가  $E_x[X^2]$ 에 수렴한다. 즉  $\frac{1}{N} \sum_{i=1}^N x_i^2 \rightarrow E_x[X^2]$ .
- 3) 가산 잡음  $w$ 의 경험적 2차 모멘트가  $E_x[W^2]$ 에 수렴한다. 즉  $\frac{1}{M} \sum_{j=1}^M w_j^2 \rightarrow E_x[W^2]$ .

그러면 AMP의  $t$ 번째 반복에서 AMP의 MSE 복구 성능  $\tau_t$ 는 다음과 같이 재귀적으로 나타낼 수 있다<sup>[8]</sup>.

$$(\sigma_{SE}^{t+1})^2 = \frac{N}{M} E[(\eta(X_0 + \sigma_{SE}^t Z; \theta_t) - X_0)^2 + \sigma_w^2, \text{ for } t \geq 0 \quad (33)$$

여기서  $Z \sim \mathcal{N}(0, 1)$  이고,  $X_0 \sim p_x$ 은 신호벡터의 한 성분이며  $\sigma_w^2$ 은 잡음  $w$ 의 분산이다. 그리고  $E[\cdot]$ 은  $Z$ 와  $X_0$ 에 관련한다.  $\sigma_{SE}^t$ 의 초기값은  $\sigma_{SE}^{t=0} = \sigma_w^2 + (N/M)E[X_0^2]$ 으로 설정된다. 즉 AMP의 복구 성능은 하나의 파라미터  $(\sigma_{SE}^t)^2$ 에 대한 1차원 재귀식으로 표현된다.

이제, (33)의 SE 재귀식에 대한 경험적인 유도 (heuristic derivation)를 설명한다<sup>[12][13]</sup>. 본 설명을 위해서 Algorithm 3의 AMP표현에 다음과 같은 세 가지 수정을 가한다.

- 1) 수정 1: 센싱 행렬  $\mathbf{A}$ 을 반복  $t$ 마다 독립인  $\mathbf{A}(t)$ 로 대체한다. 즉  $t$  반복에서 사용되는  $\mathbf{A}(t=1), \mathbf{A}(t=2), \mathbf{A}(t=3) \dots$ 는 상호 독립이며, 각 성분은  $A_{ij}(t) \sim \mathcal{N}(0, 1/M)$ 을 따른다. 따라서  $\mathbf{A}(t)$ 와  $t$  반복에서 얻은 추정 신호  $\mathbf{x}^t$ 는 독립이다.
- 2) 수정 2: 측정 벡터  $\mathbf{y}$  (31)를  $t$ 반복에서  $\mathbf{y}^t = \mathbf{A}(t)\mathbf{x} + \mathbf{w}$ 로 놓는다.
- 3) 수정 3: 표1의 알고리즘 3에 표현된 AMP의 Measurement node update 실행 시 Onsager term을 제거한다.

위의 수정에 따라 본 유도는 실제 알고리즘에는 정확하게는 들어맞지는 않으나, 그럼에도 불구하고 SE의 동작을 직관적으로 이해하는데 유용하다. 그 수정된 표현은 아래와 같다.

$$\mathbf{x}^{t+1} = \eta(\mathbf{A}(t)^T \mathbf{z}^t + \mathbf{x}^t; \theta_t) \quad (34)$$

$$\mathbf{z}^t = \mathbf{y}^t - \mathbf{A}(t)\mathbf{x}^t \quad (35)$$

본 유도는 (34)에 (35)을 대입하면서 시작된다.

$$\mathbf{x}^{t+1} = \eta(\mathbf{x} + \mathbf{A}(t)^T \mathbf{w} + \mathbf{B}(t)(\mathbf{x}^t - \mathbf{x}); \theta_t) \quad (36)$$

여기서  $\mathbf{B}(t) := \mathbf{I} - \mathbf{A}(t)^T \mathbf{A}(t)$ 로 정의한다. 그리고  $\mathbf{B}(t)$ 의 대각 성분을 제외한 성분들은 중심 극한 정리에 의해서 근사적으로  $B_{ij}(t) \sim \mathcal{N}(0, 1/M)$ 으로 수렴한다. 신호의 한 성분  $x_i^t$ 의 MSE를 점근적으로

$$(\hat{\sigma}_{SE}^t)^2 = \lim_{N \rightarrow \infty} \frac{1}{N} \|\mathbf{x}^t - \mathbf{x}\|_2^2 \quad (37)$$

이라고 하자. 그러면 수정 1에 따라서  $\mathbf{A}(t)$ 와  $\mathbf{x}^t$ 는 독립이므로,  $\mathbf{B}(t)(\mathbf{x}^t - \mathbf{x})$ 의 각 성분은



$\mathcal{N}\left(0, \frac{N}{M}(\hat{\sigma}_{SE}^t)^2\right)$ 인 i.i.d. Gaussian 분포를 따른다.

그리고  $\mathbf{A}^T \mathbf{w} \sim \mathcal{N}(0, \mathcal{I}^2)$  임이  $N \rightarrow \infty$ 에서 사실임으로, (36)에서  $\eta(\cdot)$  함수 안의 벡터  $\mathbf{x} + \mathbf{A}(t)^T \mathbf{w} + \mathbf{B}(t)(\mathbf{x}^t - \mathbf{x})$ 의 한 성분의 MSE는

$$(\sigma_{SE}^{t+1})^2 = \sigma_w^2 + \frac{N}{M}(\hat{\sigma}_{SE}^t)^2 \quad (38)$$

으로 나타나며, 그 벡터 성분에 대한 확률 변수는  $X_0 + \tau_t Z$ 으로 표현 할 수 있다. 또한, 이에 따라 식 (37)는  $N \rightarrow \infty$ 인 경우,

$$\begin{aligned} (\hat{\sigma}_{SE}^t)^2 &= \lim_{N \rightarrow \infty} \frac{1}{N} \|\mathbf{x}^{t+1} - \mathbf{x}\|_2^2 \\ &= E[(\eta(X_0 + \sigma_{SE}^t Z; \theta_t) - X_0)^2] \end{aligned} \quad (39)$$

으로 다시 쓸 수 있다. 최종적으로 (39)와 (38)을 통해서 우리는 식 (33)에 주어진 SE 재귀식에 이를 수 있다.

위의 설명으로부터 센싱 행렬  $\mathbf{A}(t)$ 와 추정 신호  $\mathbf{x}^t$  사이의 독립성이 SE를 유도하는데 반드시 보장되어야 함을 보았다. 하지만 일반적인 회소 신호 복구 문제에서는 센싱 행렬  $\mathbf{A}$ 가  $t$ 반복에 걸쳐 상수이므로  $\mathbf{x}^t$ 와 독립성이 보장되지 않는다. AMP는 Onsager 항을 (35)에 추가하여 상수 센싱 행렬  $\mathbf{A}$ 와 신호  $\mathbf{x}$ 간의 상관성을  $N \rightarrow \infty$ 에서 무효화 시키고, 복구 간에 신호 성분 사이의 간섭 요소인  $\mathbf{B}(\mathbf{x}^t - \mathbf{x})$ 항을 최소화 시킨다. 따라서 센싱 행렬과 추정 신호 사이의 독립성이 보장되지 않는 경우에도 AMP는 SE (33)에서 제시하는 MSE 성능을 점근적으로 따라갈 수 있다. IST(Iterative soft thresholding)<sup>[14]</sup>은 Algorithm 3의 AMP 알고리즘에서 Onsager 항을 제외한 형태를 가진다. 즉

$$\mathbf{x}^{t+1} = \eta(\mathbf{A}^T \mathbf{z}^t + \mathbf{x}^t; \theta_t), \quad (40)$$

$$\mathbf{z}^t = \mathbf{y} - \mathbf{A} \mathbf{x}^t \quad (41)$$

이다. 따라서 AMP와 IST 두 가지 알고리즘의 복구 성능을 비교함으로써  $\mathbf{A}$ 와  $\mathbf{x}^t$ 사이의 상관성이 복구 성능에 얼마나 영향을 미치는가를 확인할 수 있다. 실제로 [13],[14]에서는 그러한 두 알고리즘 사이의 성능 차이를 실험적으로 입증하며 IST는 SE에서 제시하는 성능을 만족 시키지 못함을 확인하였다.

우리는 AMP의 SE 성능 추적으로부터 다음과 같은 사항을 주목한다.

- 1) Remark 1:  $\mathbf{A}^T \mathbf{z}^t + \mathbf{x}^t$ 는 점근적으로 Gaussian 분포  $\mathcal{N}(\mathbf{x}, \mathcal{I}(\sigma_{SE}^t)^2)$ 을 따른다[12].
- 2) Remark 2:  $(\sigma_{SE}^t)^2$ 는  $\eta(\cdot; \theta_t)$ 의 입력값의 MSE를 나타내고  $(\hat{\sigma}_{SE}^{t+1})^2$ 는  $\eta(\cdot; \theta_t)$ 의 출력값의 MSE를 나타낸다.
- 3) Remark 3: SE 재귀식은 단조 감소하고 concave 함수이다[15].

Remark 1은 함수  $\eta(\cdot; \theta_t)$ 의 입력값  $\mathbf{x}^t + \mathbf{A}^T \mathbf{z}^t$ 이 “신호 성분 + 잡음 같은 성분”의 형태를 가진다는 사실을 말한다. 여기서 “잡음 같은 성분”은 Gaussian 가산 잡음  $\mathbf{A}^T \mathbf{w}$ 에 의한 성분과 복구 과정에서 생성되는 신호성분 간의 간섭 성분  $\mathbf{B}(\mathbf{x}^t - \mathbf{x})$ 로 이루어져 있다. 그리고 그 두 성분의 합은  $\mathcal{N}(0, \mathcal{I}(\sigma_{SE}^t)^2)$ 을 따른다. Remark 2에 대한 해석은 다음과 같다.  $(\sigma_{SE}^t)^2$ 는 (36)~(38)에서 보이듯  $\eta(\cdot; \theta_t)$ 의 입력값 즉 “신호 성분 + 잡음 같은 성분”의 MSE 성분이다. 그리고  $(\hat{\sigma}_{SE}^{t+1})^2$ 는 특정한 임계값  $\theta_t$ 을 가지고 그 “잡음 같은 성분”을 Denoising 한 신호  $\mathbf{x}^t = \eta(\mathbf{A}^T \mathbf{z}^t + \mathbf{x}^t; \theta_t)$ 의 MSE의 성분이다. Remark 3는 SE 재귀식은 단조 감소하고 오목 (concave) 함수이기 때문에 임계값  $\theta^t$ 가 적절하게 선택된 경우 AMP는 지속적으로 빠르게 수렴하며, 그에 대응하는 SE는  $t \rightarrow \infty$ 에서 하나의 고정점 (Fixed-point)  $(\sigma_{SE}^*)^2 = \lim_{t \rightarrow \infty} (\sigma_{SE}^t)^2$ 에 수렴한다는 사실을 알려준다.

### 2.1.1. Minimax 규칙에 의한 임계값 $\theta_t$ 의 결정

식 (18)에 따라서, Soft threshold 함수  $\eta(\mathbf{x}; \theta)$ 는 입력 벡터 성분들  $x_i \in \mathbf{x}$  중 크기(magnitude)가 임계값  $\theta$  이상이 되는 성분을 잡음 성분으로 간주하고 그 크기  $|x_i|$ 를  $\theta$ 만큼 줄이는 Denoising을 한다. 그러므로 Remark 1에 근거하여, AMP의  $t$  반복에서 임계값  $\theta_t$ 을 아래와 같이 “잡음 같은 성분”의 MSE 값  $(\sigma_{SE}^t)^2$ 을 이용하여 설정한다.

$$\theta_t = \alpha \sigma_{SE}^t \quad (42)$$

여기서 자연스럽게 나오는 질문은 상수  $\alpha \geq 0$ 을

어떻게 정할 것인가에 대한 것이다. 이 질문에 대한 답변은 **Minimax** 추정기를 이용해서 설명한다<sup>[7]</sup>. 신호 추정 문제에서 해당 신호분포에 대한 정확한 정보가 주어지지 않고 일부 특징만 주어지는 경우가 일반적이다. 예를 들어 **CS** 문제에서 희소 신호는 **Laplace** 분포, **Spike-and-slab** 분포, **student-t** 분포 등 다양한 분포에서 생성될 수 있지만 복구 알고리즘은 이 신호를 단순히  $K$  희소 신호로 규정하고 문제를 푸는 경우가 많다. **Minimax** 추정기는 그러한 경우에 대해서 같은 특징을 공유하는 신호분포 집합  $F_X$  중 **MSE**를 최대화 하는 최악 분포(The worst case distribution)를 가정하고 그 **MSE**를 최소화 하는 방향으로 신호 추정을 수행한다. 식 (39)로부터 최악 분포에 대한 **MSE**를 정의하면 다음과 같다.

$$mse(\sigma_{SE}^t; \alpha) := \sup_{p_X \in F_X} E_X[(\eta(X_0 + \sigma_{SE}^t Z; \theta_t) - X_0)^2] \quad (43)$$

여기서  $p_X \in F_X$ 는  $X_0$ 에 대한 신호분포이다. 식 (43)의 표현을  $\alpha$ 에 대해서 최소화 하면 우리는 **Minimax** 추정을 통해서 얻을 수 있는 **Minimax MSE**값 (44)과 그에 대한 **Minimax** 최적의 임계값  $\theta_t^\# = \alpha^\# \sigma_{SE}^t$ 을 위한  $\alpha^\#$ 값 (45)을 아래와 같이 찾을 수 있다.

$$mse^\#(\sigma_{SE}^t) := \min_{\alpha \in \mathbb{R}_+} mse(\sigma_{SE}^t; \alpha) = (\sigma_{SE}^t)^2 \min_{\alpha \in \mathbb{R}_+} mse(\alpha) \quad (44)$$

$$\alpha^\#(\sigma_{SE}^t) \equiv \min_{\alpha \in \mathbb{R}_+} mse(\sigma_{SE}^t; \alpha) \quad (45)$$

(44)에서 등호가 성립하기 위해서 ‘Scale invariance’<sup>[15]</sup> 특성을 이용한다. 따라서 **Minimax** 추정기는 집합  $F_X$ 에 속하는 신호 분포에서 생성되는 모든 신호에 대해서 **soft threshold function**  $\eta(\mathbf{x}; \theta)$ 을 사용했을 때, 균등하게 좋은 성능을 가진다.

2.2.2. Lasso와 AMP와의 동등성

**Lasso**는 희소 신호 복구를 위한 가장 대표적인 최적화 문제설정이다<sup>[1,9]</sup>. 즉 **Lasso** 문제는 다음과 같이 정의된다.

$$(P_{Lasso}) : \min_{\mathbf{s}} \|\mathbf{y} - \mathbf{A}\mathbf{s}\|_2^2 + \lambda \|\mathbf{s}\|_1 \quad (44)$$

하지만 (44)의 **Lasso** 최적화 문제를 직접 푸는 것은 신호의 길이  $N$ 이 길어질수록 계산적으로 어렵다. 그렇기 때문에 다양한 저 복잡도 알고리즘들이 **Lasso** 문제로부터 파생되었다. **AMP**는  $N \rightarrow \infty$ 에서 **Lasso** 문제의 해를 정확하게 구하는 저 복잡도 알고리즘이다. 본 장에서는 **AMP** 알고리즘이 (44)의 **Lasso** 문제를 정확하게 풀기위한 조건에 대해서 설명한다. 식 (44)에서 **Lasso** 규칙 (Regularization) 변수  $\lambda$ 와 **AMP**의 임계값  $\theta$ 가 식 (44)과 같은 관계를 가졌을 때  $t \rightarrow \infty$ 에서 **AMP**의 고정점 (Fixed-point)  $\mathbf{x}^*$ 는 **Lasso** 최적해와 동등하다<sup>[15]</sup>.

$$\lambda = \theta(1 - \text{Onsager term}) \quad (45)$$

이러한 사실은 간단하게 보일 수 있다. Remark 3에 근거하여 **SE**에 의해서 추적되는 **MSE**  $t_t^2$ 는  $t$  반복에 걸쳐서 단조 감소 한다. 따라서 고정점  $t \rightarrow \infty$ 에서의  $(\sigma_{SE}^*)^2 = \lim_{t \rightarrow \infty} (\sigma_{SE}^t)^2$  값은 매우 작고 **soft threshold** 임계값  $\theta^* = \alpha \sigma_{SE}^*$  또한 매우 작다. 그러한 사실을 이용해서 고정점에서의  $(\mathbf{x}^*, \mathbf{z}^*)$ 의 표현은 근사적으로 아래와 같이 표현될 수 있다.

$$\mathbf{x}^* = \eta(\mathbf{A}\mathbf{z}^* + \mathbf{x}^*; \theta^*) \approx \mathbf{A}^T \mathbf{z}^* + \mathbf{x}^* - \mathbf{v}\theta^* \quad (46)$$

여기서  $\mathbf{v}$ 는  $\mathbf{x}^*$ 의  $l_1$ -norm의 하나의 subgradient 벡터이다. 즉,  $\mathbf{v} \in \partial \|\mathbf{x}^*\|_1$  이다. 이어서, Algorithm 3의 measurement node 업데이트로부터 얻은 식  $\mathbf{z}^* = \frac{\mathbf{y} - \mathbf{A}\mathbf{x}^*}{(1 - \text{Onsager term})}$ 을 식 (45)에 대입하면

$$\theta(1 - \text{Onsager term})\mathbf{v} = \mathbf{A}^T(\mathbf{y} - \mathbf{A}\mathbf{x}^*) \quad (47)$$

와 같은 결과를 얻고 이것은  $\lambda = \theta(1 - \text{Onsager term})$  일 때의 **Lasso**의 Karush-Kuhn-Tucker 조건의 하나인 stationarity 조건과 일치 한다. 추가적으로, 위에서 토의한 **SE** 재귀식에 식 (45)에서 얻은 관계식을 이용해서 얻은 임계값  $\theta = \frac{\lambda}{(1 - \text{Onsager term})}$ 을 적용한다면 **AMP**는  $N \rightarrow \infty$ 에서 **Lasso**의 해를 정확하게 예측 할 수 있다.

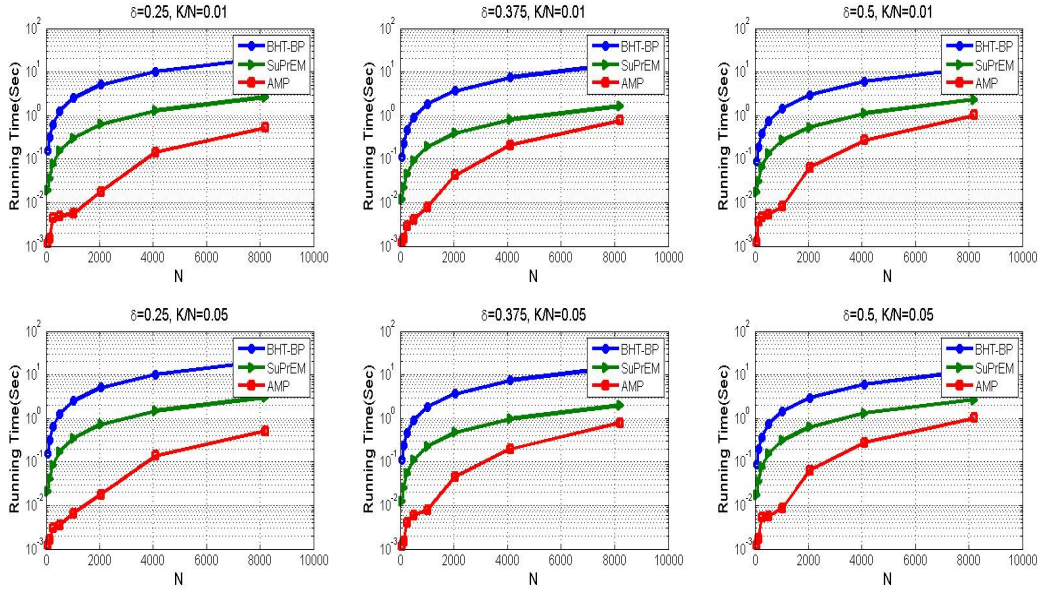


그림 5. AMP, SuPrEM, BHT-BP 알고리즘의 Running time  
Fig. 5. Running time for AMP, SuPrEM, BHT-BP

### III. 실험

#### 3.1. 신호 복구를 위한 실행 시간 비교

우리는 AMP 알고리즘의 신호 복구를 위한 계산 복잡도를 확인하기 위하여, AMP의 실행 시간을 MP 기반의 다른 알고리즘(BHT-BP<sup>[7]</sup>와 SuPrEM<sup>[8]</sup>)들과 비교하여 시뮬레이션을 통해 알아보았다. 먼저 BHT-BP와 SuPrEM에 대해서 간략하게 소개한다.

BHT-BP(Bayesian hypothesis test via belief propagation)<sup>[7]</sup>: BHT-BP는 nonparametric 방식으로 MP를 구현하여 적용한 희소 신호 복구 알고리즘이다. BHT-BP에서는 하나의 메시지가  $N_d$ 의 사이즈를 가지는 벡터로 표현되고, 따라서  $N_d$ 가 증가할수록 신호처리에 더 많은 계산이 소요된다. MP 메시지의 벡터 표현을 위해서 BHT-BP는 메시지를 균등하게 샘플링 한다. 즉 메시지 표현을 위해 어떠한 근사화도 사용하지 않는다. 이러한 이유로 BHT-BP는 복잡한 확률 분포로 가정되는 희소 신호도 정확하게 추론 가능하다는 장점을 가진다<sup>[7]</sup>. AMP와는 다르게 BHT-BP에서는 이진 희소 행렬을 센싱 행렬  $A \in \{0,1\}^{M \times N}$ 로 사용한다. 여기서  $A$ 의 하나의 열벡터는  $L$ 개의 영이 아닌 성분으로 이루어져 있다. BHT-BP는  $O(LN_d \log N_d M \log N + KM)$ 의 계산 복잡도를 가진다, 여기서  $K$ 는 신호  $x$ 의

영이 아닌 원소의 개수를 나타낸다.

SuPrEM(Sum-product with expectation maximization)<sup>[8]</sup>: SuPrEM은 parametric 방식으로 MP 알고리즘을 구현하여 적용한 희소 신호 복구 알고리즘이다. 구체적으로, SuPrEM에서는 희소 신호 복구를 위한 MAP 추정 문제를 설정하고, 그것을 EM 알고리즘 사용하여 해결한다. SuPrEM의 장점은 EM 알고리즘의 높은 계산 복잡도를 MP 알고리즘을 적용하여 개선한다는 점이다. BHT-BP와 마찬가지로 SuPrEM 또한 이진 희소 센싱 행렬을 기반으로 하고 있다. 하지만 그 이진 희소 센싱 행렬은 해당논문에서 저자가 설계한 Low-density frame (LDF)<sup>[13]</sup> 이라고 불리는 특수 행렬에 국한된다. SuPrEM은  $K$ -희소 신호 복구에 대해서 잡음에 강인한 특성을 가지며, 매우 낮은 계산 복잡도  $O(LM \log N)$ 를 가진다. 하지만 대상이 되는 신호 모델이  $K$ -희소 신호에 제한된다는 점과 센싱 행렬이 LDF로 제한된다는 한계를 가지고 있다<sup>[13]</sup>.

AMP의 주목할 만한 점은 Gaussian 센싱 행렬  $A_{ji} \sim N(0,1/M)$ 을 사용하면서도, 1차 Taylor 근사화를 이용해서 신호 복구를 위한 복잡도를 획기적으로  $O(NM \log N)$ 으로 낮추었다는 것이다.

AMP의 계산 복잡도의 실험적 성능 비교를 위해서 BHT-BP는  $N_d = 128$ 로 설정 하였고 모든 알고리즘의 최고 반복 횟수를 30번으로 제한하였다. 그

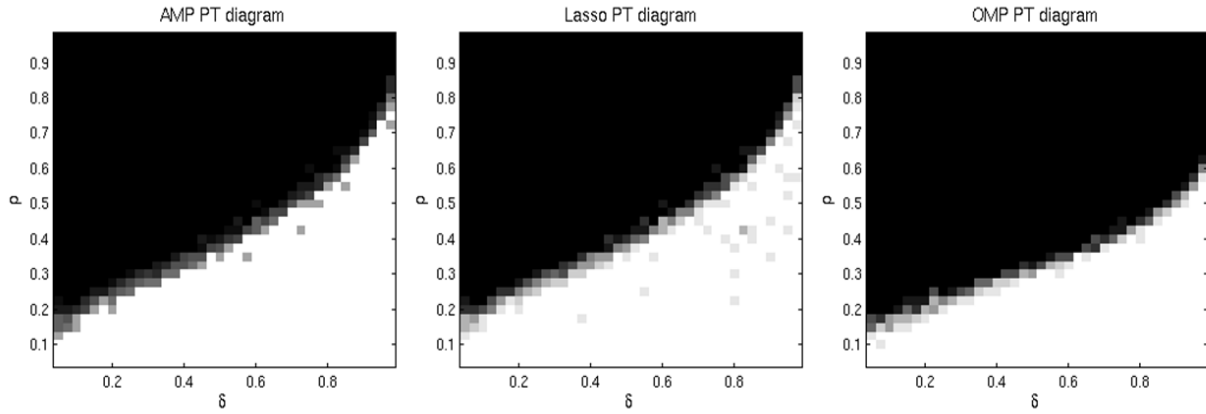


그림 6. CS 복구 알고리즘(왼쪽부터 AMP[13], Lasso[14], OMP[15])에 대한 실험적 Phase transition 도표의 비교  
 Fig. 6. Observed phase transition of AMP, Lasso, OMP

림 5는 위에서 언급된 세 가지 알고리즘의 실행 시간을 도시한다. (이 실험은 Intel Core i7-3820 CPU 3.60GHz, 16GB RAM, 32bit 운영체제 환경에서 이루어졌다.) 그림 5의 실험은  $N=2^4 \sim 2^{13}$ ,  $\delta=0.25, 0.375, 0.5$ ,  $K/N=0.01, 0.05$ 의 환경에서 수행되었다. 그림 5의 실험결과 알고리즘의 MSE는  $5 \times 10^{-5}$ 이하의 충분히 작은 값을 보이면서  $O(\log N)$  복잡도를 가지는 것이 확인 되었다. 그 가운데 실행 시간이 BHT-BP, SuPrEM, AMP 순으로 높은 것을 확인 하여, 위에서의 언급한 계산 복잡도에 대한 토의가 타당함을 뒷받침 하였다.

3.2. 신호 복구의 Phase Transition 성능 비교

MP 알고리즘의 CS 복구 성능을 실험적인 phase transition (PT)을 통해서 분석한다. PT은  $(\rho, \delta)$ 의 평면에서 신호 복구 성공확률을 기준으로 해당 알고리즘이 신호 복구에 성공하는 영역과 실패하는 영역을 도시한다. 다시 말해서 그림 6에서 백색으로 표시되는 영역은 알고리즘이 성공적으로 신호를 복구하는 영역이고, 검은색으로 표시되는 영역은 알고리즘이 신호 복구에 실패하는 영역이다. 여기서  $\rho = K/M$ 은 신호의 희소성을 나타내고,  $\delta = M/N$ 은 신호 획득 시의 undersampling 비율을 나타낸다. 그림 6은 AMP<sup>[13]</sup>, Lasso<sup>[9]</sup>, OMP<sup>[10]</sup>의 PT도표를 실험적으로 비교하여 보인다.

Lasso는 가장 대표적인 희소 신호 복구 알고리즘으로 섹션 II-다의 식 (44)에서 소개되었다. Lasso를 성능 비교 대상으로 선택한 이유는 Lasso의 해는 희소 신호 복구 문제에서 센싱 행렬  $A$ 이 특정조건만을 만족하면 L0-norm 방법과 동등하다고 알려져 있기 때문이다<sup>[1]</sup>. 여기서 L0-norm 방법은  $N$ 개의

길이를 가지고 있는 신호 중에서  $K$ 개의 요소가 0이 아닌 모든  $K$ -희소 신호를 만들어 그 중 문제와 맞는 것을 해로 결정하는 과도 검색(Exhaustive Search) 방법이다. 그리고 이 방법은 희소 신호 복구 문제에 대한 최적의 해를 제공한다. 또한, AMP 알고리즘은  $N \rightarrow \infty$ 에서 Lasso 문제의 해를 정확하게 구하는 것으로 알려져 있으므로, AMP와 Lasso의 성능을 비교해서 그러한 사실을 확인 하는 것은 의미가 있다.

OMP를 또 다른 비교 대상으로 선택한 이유는 OMP는 저 복잡도 희소 신호 복구 알고리즘으로 가장 널리 알려있기 때문이다. OMP에 대한 소개는 아래와 같이 한다.

OMP(Orthogonal matching pursuit)<sup>[10]</sup>: OMP 알고리즘은 단계적으로 희소 신호의 해를 구하는 알고리즘이다. 각  $k$ 단계에서 OMP는 residual 벡터 즉  $r^{(k)} = y - A\hat{s}^{(k)}$ 와 센싱 행렬의 모든 열벡터  $a_i, \forall i \in \{1, \dots, N\}$ 와의 상관도를 구한다. 그리고 가장 상관도가 큰 열벡터의 인덱스  $i$ 를 희소 신호의 서포트 집합 (영이 아닌 신호 원소의 인덱스 집합)에 포함 시킨다. 따라서  $K$  단계 후 모든 영이 아닌 원소를 찾게 되고, 그렇게 구한 서포트 집합 정보를 이용해서 최소 자승 방법으로 희소 신호를 복구 할 수 있다. OMP와 같이 문제 전체를 멀리 내다보는 것 보다 각 단계에서 당장의 상황만을 고려해서 신호를 복구 하는 알고리즘을 greedy 알고리즘이라고 한다. 하지만 greedy 철학에 의한 알고리즘임에서 불구하고 OMP는 저 복잡도 특성과  $O(MNK)$  동시에 비교적 우수한 복구 성능을 제공한다는 것이 알려져 있다. 따라서 가장 잘 알려진 저 복잡도 알고리즘인 OMP와 새로운 저 복잡도 알고리즘인

AMP를 비교하는 것은 의미가 있다.

본 실험에서는 신호 복구 성공을 복구된 신호와 실제 신호의 MSE가  $10^{-3}$  이하 인 경우로 정의 하였고, 평면위의 한 격자의 색이 밝을수록 신호 복구 성공 확률이 1에 가깝도록 설정하였다. 또한 본 실험은  $N=1024$ 을 가지고 영이 아닌 값이  $\{-1,+1\}$ 에 속하는 신호를 대상으로 하였다. AMP 복구실험에는 Minimax 규칙을 이용하여 찾은 최적의 임계값  $\theta$ 을 적용하여 수행하였다. 그림 6의 결과는 AMP가 OMP보다 더 넓은  $(\rho, \delta)$  평면 영역에서 성공적으로 신호를 복구 할 수 있음을 보여준다. 또한, AMP의 PT도표는 Lasso의 도표와 근사적으로 일치함을 보여준다. 따라서 그림 6로부터 우리는 AMP의 신호 복구 성능이 Lasso의 성능과 점근적으로 동등함을 실험적으로 확인 할 수 있다.

#### IV. 결 론

본 논문에서는 최근 학회에서 많은 주목을 받고 있는 AMP 알고리즘을 고전적 MP와 연결하여 심도 있는 수학적 유도와 AMP 알고리즘의 특징을 제시함으로써 이해하기 쉬운 지침을 제공하였다. 또한 MP에 기반을 둔 여러 알고리즘과의 실행 시간 비교, 그리고 대표적인 다른 희소 신호 복구 알고리즘인 Lasso와 OMP와의 PT 비교 시뮬레이션을 통해 AMP 알고리즘이 우수한 복구 성능에 비해 상당한 계산적 감소를 가진다는 사실을 확인 하였다. 이러한 사실을 바탕으로 AMP 알고리즘은 큰 차원을 가지는 문제에 응용하고 이를 해결하는데 많은 도움이 될 것으로 기대된다. 하지만 본문에서 소개한 AMP 알고리즘의 우수한 성능 결과를 보장받기 위해서는 본문에서 가정한 몇 가지의 전제 조건을 고려해야 한다. 먼저 AMP 알고리즘이 동작하기 위해서는, 대상 희소 신호 벡터의 차원  $N$ 이 상당히 크고 밀집 그래프 환경이어야 한다. AMP 알고리즘은 메시지를 Gaussian 분포로 근사화하기 때문에 대상 희소 신호 벡터가 작은 차원을 가지는 문제에서는 중심 극한 정리에 따라 근사화가 유효하지 않다. 다음으로 AMP 알고리즘이 (33)에 소개된 State Evolution에 의해서 보장되기 위해서는 센싱 행렬  $A$ 의 요소가  $N(0,1/M)$ 인 Gaussian 분포를 따라야 한다. 비록 [16]에서 센싱 행렬의 요소가  $\pm 1$  중에서 i.i.d하게 선택되는 경우와 partial Fourier 행렬인 경우도 다루었지만 오직 실험적인 성능 결과만을

제시했을 뿐이었다. 또한 AMP 알고리즘이 최적의 성능을 달성하기 위해서는 그에 대응하는 임계값  $\theta$ 을 구하는 것이 중요하다. 이 임계값  $\theta$ 를 구하기 위한 Minimax 접근이 (44),(45)에서 소개되었지만 그 과정은 익숙하지 않은 사용자에게는 어려울 수도 있다. 마지막으로 AMP알고리즘을 비롯한 많은 CS 복구 알고리즘은 대다수의 0과 몇 개의 0이 아닌 요소로 이루어진 완벽한 희소신호로 가정하여 설계 되었다. 하지만 실제 환경에서의 자연적 신호는 완벽한 희소 신호이기 보다는 대다수의 작은 요소와 몇 개의 큰 요소로 구성된 신호가 대부분이기 때문에 실제 환경과 이론적 환경 간의 이질감이 존재한다. 이와 같이 AMP 알고리즘을 실제 연구 문제에 적용하기 위해서는 위와 같이 몇 가지의 고려해야 할 사항이 있다. 따라서 이러한 사항들을 극복하고 많은 분야에서 일반적으로 적용될 수 있는 AMP 알고리즘의 개발과 실제 환경에서의 자연적 신호를 정확하게 복구할 수 있는 방법을 강구하는 것이 앞으로의 연구 방향으로 제시될 수 있을 것이다.

#### References

- [1] D. L. Donoho, "Compressed sensing," *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1289 - 1306, Sep. 2006.
- [2] E. Candes and M. Wakin, "An introduction to compressive sampling," *IEEE Signal Process. Mag.*, vol. 25, no. 2, pp. 21 - 30, Mar. 2008.
- [3] J. A. Tropp, "Just relax: convex programming methods for identifying sparse signals in noise," *IEEE Trans. Inf. Theory*, vol. 52, no. 3, pp. 1030-1051, Mar. 2006.
- [4] D. L. Donoho, Y. Tsaig, I. Drori, and J. L. Starck, "Sparse solution of underdetermined systems of linear equations by stagewise orthogonal matching pursuit," *IEEE Trans. Inf. Theory*, vol. 58, no. 2, pp. 1094-1121, Feb. 2012.
- [5] F. R. Kschischang, B. J. Frey, and H. A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 498-519, Feb. 2001.
- [6] H. H. Baek, S. J. Park, J. M. Ryu, and H. N. Lee, "Reanalysis of Approximate Message

Passing (AMP) for compressed sensing signal recovery,” in *Proc. KICS Int. Conf. Commun. 2013 (KICS ICC 2013)*, pp. 701-702, Jeju Island, Korea, June 2013.

[7] J. Kang, H.-N. Lee, and K. Kim, “Bayesian hypothesis test using nonparametric belief propagation for noisy sparse recovery,” *IEEE Trans. Signal Process.*, Submitted.

[8] M. Akcakaya, J. Park, and V. Tarokh, “A coding theory approach to noisy compressive sensing using low density frame,” *IEEE Trans. Signal Process.*, vol. 59, no. 12, pp. 5369-5379, Nov. 2011.

[9] R. Tibshirani, “Regression shrinkage and selection via the lasso,” *J. Royal Statistical Soc.: Series B*, vol. 58, no. 1, pp. 267-288, 1996.

[10] J. A. Tropp and A. C. Gilbert, “Signal recovery from random measurements via orthogonal matching pursuit,” *IEEE Trans. Inf. Theory*, vol. 53, no. 12, pp. 4655-4666, Dec. 2007.

[11] D. L. Donoho, A. Maleki, and A. Montanari, “Message passing algorithms for compressed sensing: I. motivation and construction,” in *Proc. IEEE Inform. Theory Workshop (ITW 2010)*, pp. 1-5, Cairo, Egypt, Jan. 2010.

[12] M. Bayati and A. Montanari, “The dynamics of message passing on dense graphs, with applications to compressed sensing,” *IEEE Trans. Inf. Theory*, vol. 57, no. 2, pp. 764 - 785, Feb. 2011.

[13] D. L. Donoho, A. Maleki, and A. Montanari, “Message passing algorithms for compressed sensing,” *Proc. National Academy Sci.*, vol. 106, no. 45, pp. 18914 - 18915, Sep. 2009.

[14] A. Maleki and D. L. Donoho, “Optimally tuned iterative reconstruction algorithms for compressed sensing,” *IEEE J. Sel. Topics Signal Process.*, vol. 4, no. 2, pp. 330-341, Apr. 2010.

[15] D. L. Donoho, A. Maleki, and A. Montanari, “The noise-sensitivity phase transition in compressed sensing,” *IEEE Trans. Inf. Theory*, vol. 57, no. 10, pp. 6920-6941, Oct. 2011.

[16] A. Maleki and D. L. Donoho, “Optimally tuned iterative Reconstruction Algorithms for compressed sensing,” *IEEE J. Sel. Topics Signal Process.*, vol. 4, no. 2, pp. 330-341, Apr. 2010.

**백 형 호 (Hyeong-ho Baek)**



2012년 2월 경북대학교 전자공학부 졸업  
 2012년 3월~현재 광주과학기술원 정보통신공학부 석사과정  
 <관심분야> 무선통신, 압축센싱, 디지털 신호 처리

**강 재 욱 (Jae-wook Kang)**



2009년 2월건국대학교 정보통신공학과 졸업  
 2010년 8월 광주과학기술원 정보통신공학부 석사 졸업  
 2010년 9월~현재 광주과학기술원 정보통신공학부 박사과정 재학  
 <관심분야> 통계적 신호처리, 압축센싱, 기계학습

**김 기 선 (Ki-seon Kim)**



1978년 서울대학교 전자공학과 졸업  
 1980년 서울대학교 전자공학과 석사 졸업  
 1987년 University of Southern California, Los Angeles 박사 졸업

1994~현재 광주과학기술원 정보통신공학부 정교수  
 <관심분야> 센서 네트워크, 무선통신, 수중통신, 생체 신호처리

이 흥 노 (Heung-no Lee)



1993년 University of  
California, Los Angeles  
전자공학과 졸업

1994년 University of  
California, Los Angeles  
전자공학과 석사 졸업

1999년 University of  
California, Los Angeles 전자 공학과 박사 졸업

1994 ~ 현재 광주과학기술원 정보통신공학부 정교수  
<관심분야> 압축센싱, 정보이론, 부호 이론, BCI