Distributed Approximate Message Passing with Summation Propagation

Ryo Hayakawa (Kyoto University, Japan)
Ayano Nakai (Kyoto University, Japan)
Kazunori Hayashi (Osaka City University, Japan)
Outline

1. Introduction

2. Preliminaries
   i. AMP Algorithm
   ii. Consensus Propagation

3. Proposed Method: Distributed AMP Algorithm

4. Simulation Result

5. Conclusion
Outline

1. **Introduction**

2. Preliminaries
   i. AMP Algorithm
   ii. Consensus Propagation

3. Proposed Method: Distributed AMP Algorithm

4. Simulation Result

5. Conclusion
Compressed Sensing [1]

reconstruct a **sparse** vector \( x \in \mathbb{R}^N \)
from its **underdetermined** linear measurement \( y = Ax + v \in \mathbb{R}^M (M < N) \)

\( x \in \mathbb{R}^N \): unknown sparse vector (most elements are zero)

\( A \in \mathbb{R}^{M \times N} \): measurement matrix \((M < N)\)

\( y = Ax + v \in \mathbb{R}^M \): measurement vector

noise vector

Application

✦ magnetic resonance imaging (MRI) [2]
✦ wireless channel estimation [3]

---


unknown sparse vector $\mathbf{x} \in \mathbb{R}^N$

measurement matrix: $A_k \in \mathbb{R}^{M_k \times N}$
measurement vector: $y_k = A_k x + v_k \in \mathbb{R}^{M_k}$
reconstruct $\mathbf{x}$ from $y_k, A_k \ (k = 1, \ldots, K)$

Application [4]
- sensor network
- video coding
- image fusion

Conventional Methods (1/2)

- **D-LASSO** [5]
  - (Distributed-Least Absolute Shrinkage and Selection Operator)
- **D-ADMM** [6]
  - (Distributed-Alternating Direction Method of Multipliers)
  - The computational complexity might be large
- **D-IHT** [7]
  - (Distributed-Iterative Hard Thresholding)
  - Each node performs simple calculations
  - The sparsity level is required

---


Distributed AMP [8], Multi-processor AMP [9] (Approximate Message Passing)

- Each node performs simple calculations
- The sparsity level is **not** required
- A **fusion node** communicating with all nodes is required

---


propose a **fully distributed** AMP algorithm, which does not require any fusion node

1. obtain update equations of the AMP algorithm for distributed measurements
   - **local** computation at each node
2. propose **summation propagation** for the global computation
   - **global** computation using communications
3. show the validity of the proposed algorithm via computer simulation
Outline

1. Introduction

2. Preliminaries
   i. AMP Algorithm
   ii. Consensus Propagation

3. Proposed Method: Distributed AMP Algorithm

4. Simulation Result

5. Conclusion
AMP Algorithm (1/2)

\[ x \in \mathbb{R}^N \] : unknown sparse vector

\[ A \in \mathbb{R}^{M \times N} \] : measurement matrix \((M < N)\)

i.i.d. elements with zero mean and unit variance

\[ y = Ax + v \in \mathbb{R}^M \] : measurement vector

\[ \ell_1 \text{ optimization} \]

\[ \hat{x} = \arg \min_{z \in \mathbb{R}^N} \|z\|_1 \text{ subject to } y = Az \]

AMP Algorithm \([10, 11]\)

(Approximate Message Passing)

✓ low complexity (no matrix inversion, \(O(MN)\))

✓ asymptotic analysis


**AMP Algorithm (2/2)**

\( \mathbf{x} \in \mathbb{R}^N \): unknown sparse vector

\( \mathbf{y} = A\mathbf{x} + \mathbf{v} \in \mathbb{R}^M \): measurement vector

1. Initialization: \( t = 1, \hat{x}(1) = 0, s(0) = 0, r(0) = 0, \hat{\sigma}^2(0) = 0 \)

2. \( s(t) = y - A\hat{x}(t) + \frac{1}{\Delta} s(t-1) \langle \eta' (r(t-1); \hat{\sigma}^2(t-1)) \rangle \)

3. \( r(t) = \hat{x}(t) + \frac{1}{M} A^T s(t) \)

4. \( \hat{\sigma}^2(t) = \frac{\|s(t)\|_2^2}{MN} \)

5. \( \hat{x}(t+1) = \eta (r(t); \hat{\sigma}^2(t)) \)

\( \Delta = M/N \): measurement ratio

\( \langle \cdot \rangle \): mean

**example of \( \eta(\cdot;\cdot) \): soft thresholding**

\[ \eta(u; \sigma^2) = \begin{cases} 0 & \text{if } u < 0 \\ u & \text{if } u \geq 0 \end{cases} \]
A distributed algorithm for **average consensus** on undirected graphs

All nodes obtain the mean $\mu = \frac{1}{K} \sum_{k=1}^{K} c_k$

communicate with neighbor nodes

The graph is a tree
# of iterations $\geq$ graph diameter

average consensus is achieved

Outline

1. Introduction

2. Preliminaries
   i. AMP Algorithm
   ii. Consensus Propagation

3. Proposed Method: Distributed AMP Algorithm

4. Simulation Result

5. Conclusion
unknown sparse vector \( \mathbf{x} \in \mathbb{R}^N \)

measurement matrix: \( A_k \in \mathbb{R}^{M_k \times N} \)
measurement vector: \( y_k = A_k \mathbf{x} + \mathbf{v}_k \in \mathbb{R}^{M_k} \)

reconstruct \( \mathbf{x} \) from \( y_k, A_k \) \((k = 1, \ldots, K)\)

All measurements \( y_1, \ldots, y_K \) can be combined as

\[
\begin{bmatrix}
  y_1 \\
  \vdots \\
  y_K
\end{bmatrix} =
\begin{bmatrix}
  A_1 \\
  \vdots \\
  A_K
\end{bmatrix} \mathbf{x} +
\begin{bmatrix}
  v_1 \\
  \vdots \\
  v_K
\end{bmatrix}
\]
Proposed Method: Distributed AMP Algorithm

AMP Algorithm for Distributed Model (1/2)

### Centralized Model

\[ y = A\hat{x} + v \]

### AMP Algorithm

\[
\begin{align*}
    s(t) &= y - A\hat{x}(t) \\
    &\quad + \frac{1}{\Delta} s(t-1) \langle \eta' (r(t-1); \hat{\sigma}^2(t-1)) \rangle \\
    r(t) &= \hat{x}(t) + \frac{1}{M} A^T s(t) \\
    \hat{\sigma}^2(t) &= \frac{\|s(t)\|_2^2}{MN} \\
    \hat{x}(t+1) &= \eta(r(t); \hat{\sigma}^2(t))
\end{align*}
\]

### Distributed Model

\[
\begin{bmatrix}
    y_1 \\
    \vdots \\
    y_K
\end{bmatrix} =
\begin{bmatrix}
    A_1 \\
    \vdots \\
    A_K
\end{bmatrix}
\begin{bmatrix}
    x \\
    \vdots \\
    v_K
\end{bmatrix}
\]

### AMP Algorithm

\[
\begin{align*}
    s_k(t) &= y_k - A_k\hat{x}(t) \\
    &\quad + \frac{1}{\Delta} s_k(t-1) \langle \eta' (r(t-1); \hat{\sigma}^2(t-1)) \rangle \\
    r(t) &= \sum_{k=1}^{K} \left( \frac{1}{K} \hat{x}(t) + \frac{1}{M} A_k^T s_k(t) \right) \\
    \hat{\sigma}^2(t) &= \sum_{k=1}^{K} \frac{\|s_k(t)\|_2^2}{MN} \\
    \hat{x}(t+1) &= \eta(r(t); \hat{\sigma}^2(t))
\end{align*}
\]
**Proposed Method: Distributed AMP Algorithm**

**AMP Algorithm for Distributed Model (1/2)**

### Centralized Model

\[ y = Ax + \nu \]

**AMP Algorithm**

\[
s(t) = y - A\hat{x}(t) + \frac{1}{\Delta} s(t-1) \langle \eta'(r(t-1); \hat{\sigma}^2(t-1)) \rangle
\]

\[
r(t) = \hat{x}(t) + \frac{1}{M} A^T s(t)
\]

\[
\hat{\sigma}^2(t) = \frac{\|s(t)\|_2^2}{MN}
\]

\[
\hat{x}(t+1) = \eta(r(t); \hat{\sigma}^2(t))
\]

### Distributed Model

\[
\begin{bmatrix} y_1 \\ \vdots \\ y_K \end{bmatrix} = \begin{bmatrix} A_1 \\ \vdots \\ A_K \end{bmatrix} x + \begin{bmatrix} \nu_1 \\ \vdots \\ \nu_K \end{bmatrix}
\]

**AMP Algorithm**

\[
s_k(t) = y_k - A_k\hat{x}(t) + \frac{1}{\Delta} s_k(t-1) \langle \eta'(r(t-1); \hat{\sigma}^2(t-1)) \rangle
\]

\[
r(t) = \sum_{k=1}^{K} \left( \frac{1}{K} \hat{x}(t) + \frac{1}{M} A_k^T s_k(t) \right)
\]

\[
\hat{\sigma}^2(t) = \sum_{k=1}^{K} \frac{\|s_k(t)\|_2^2}{MN}
\]

\[
\hat{x}(t+1) = \eta(r(t); \hat{\sigma}^2(t))
\]
Proposed Method: Distributed AMP Algorithm

AMP Algorithm for Distributed Model (1/2)

**Centralized Model**

\[ y = Ax + \nu \]

**AMP Algorithm**

\[ s(t) = y - A\hat{x}(t) + \frac{1}{\Delta} s(t-1) \langle \eta' (r(t-1); \hat{\sigma}^2(t-1)) \rangle \]

\[ r(t) = \hat{x}(t) + \frac{1}{M} A^T s(t) \]

\[ \hat{\sigma}^2(t) = \frac{\|s(t)\|^2}{2MN} \]

\[ \hat{x}(t+1) = \eta (r(t); \hat{\sigma}^2(t)) \]

**Distributed Model**

\[
\begin{bmatrix}
    y_1 \\
    \vdots \\
    y_K
\end{bmatrix} =
\begin{bmatrix}
    A_1 \\
    \vdots \\
    A_K
\end{bmatrix} x +
\begin{bmatrix}
    v_1 \\
    \vdots \\
    v_K
\end{bmatrix}
\]

**AMP Algorithm**

\[ s_k(t) = y_k - A_k\hat{x}(t) + \frac{1}{\Delta} s_k(t-1) \langle \eta' (r(t-1); \hat{\sigma}^2(t-1)) \rangle \]

\[ r(t) = \sum_{k=1}^{K} \left( \frac{1}{K} \hat{x}(t) + \frac{1}{M} A_k^T s_k(t) \right) \]

\[ \hat{\sigma}^2(t) = \sum_{k=1}^{K} \frac{\|s_k(t)\|^2}{2MN} \]

\[ \hat{x}(t+1) = \eta (r(t); \hat{\sigma}^2(t)) \]
Proposed Method: Distributed AMP Algorithm

AMP Algorithm for Distributed Model (1/2)

**centralized model**

\[ y = Ax + \nu \]

**AMP algorithm**

\[
\begin{align*}
    s(t) &= y - A \hat{x}(t) \\
    &\quad + \frac{1}{\Delta} s(t-1) \langle \eta' (r(t-1); \hat{\sigma}^2(t-1)) \rangle \\
    r(t) &= \hat{x}(t) + \frac{1}{M} A^T s(t) \\
    \hat{\sigma}^2(t) &= \frac{\|s(t)\|^2}{MN} \\
    \hat{x}(t+1) &= \eta (r(t); \hat{\sigma}^2(t))
\end{align*}
\]

**distributed model**

\[
\begin{bmatrix}
    y_1 \\
    \vdots \\
    y_K
\end{bmatrix} =
\begin{bmatrix}
    A_1 \\
    \vdots \\
    A_K
\end{bmatrix}
\begin{bmatrix}
    x \\
    \vdots \\
    \nu_K
\end{bmatrix}
\]

**AMP Algorithm**

\[
\begin{align*}
    s_k(t) &= y_k - A_k \hat{x}(t) \\
    &\quad + \frac{1}{\Delta} s_k(t-1) \langle \eta' (r(t-1); \hat{\sigma}^2(t-1)) \rangle \\
    r(t) &= \sum_{k=1}^{K} \left( \frac{1}{K} \hat{x}(t) + \frac{1}{M} A_k^T s_k(t) \right) \\
    \hat{\sigma}^2(t) &= \sum_{k=1}^{K} \frac{\|s_k(t)\|^2}{MN} \\
    \hat{x}(t+1) &= \eta (r(t); \hat{\sigma}^2(t))
\end{align*}
\]
Proposed Method: Distributed AMP Algorithm

AMP Algorithm for Distributed Model (2/2)

**Centralized Model**

\[ y = Ax + \nu \]

**AMP Algorithm**

\[
\begin{align*}
    s(t) &= y - A\hat{x}(t) \\
          &+ \frac{1}{\Delta} s(t-1) \langle \eta' (r(t-1); \hat{\sigma}^2(t-1)) \rangle \\
    r(t) &= \hat{x}(t) + \frac{1}{A^T} s(t) \\
    \hat{\sigma}^2(t) &= \frac{\|s(t)\|_2^2}{MN} \\
    \hat{x}(t+1) &= \eta (r(t); \hat{\sigma}^2(t))
\end{align*}
\]

**Distributed Model**

\[
\begin{bmatrix}
    y_1 \\
    \vdots \\
    y_K
\end{bmatrix} =
\begin{bmatrix}
    A_1 \\
    \vdots \\
    A_K
\end{bmatrix} x +
\begin{bmatrix}
    v_1 \\
    \vdots \\
    v_K
\end{bmatrix}
\]

**AMP Algorithm**

\[
\begin{align*}
    s_k(t) &= y_k - A_k\hat{x}(t) \\
          &+ \frac{1}{\Delta} s_k(t-1) \langle \eta' (r(t-1); \hat{\sigma}^2(t-1)) \rangle \\
    r(t) &= \sum_{k=1}^{K} \left( \frac{1}{K} \hat{x}(t) + \frac{1}{M} A_k^T s_k(t) \right) \\
    \hat{\sigma}^2(t) &= \sum_{k=1}^{K} \frac{\|s_k(t)\|_2^2}{MN} \\
    \hat{x}(t+1) &= \eta (r(t); \hat{\sigma}^2(t))
\end{align*}
\]

*cannot be computed locally*
Proposed Method: Distributed AMP Algorithm

**Summation Propagation**

We propose *summation propagation* to compute

\[ r(t) = \sum_{k=1}^{K} \left( \frac{1}{K} \hat{x}(t) + \frac{1}{M} A_k^T s_k(t) \right) \]

\[ \hat{\sigma}^2(t) = \sum_{k=1}^{K} \frac{\|s_k(t)\|^2}{MN} \]

by using the idea of consensus propagation

- The graph is a tree
- # of iterations \( \geq \) graph diameter

All nodes obtain the summation

\[ \sum_{k=1}^{K} c_k \]
Outline

1. Introduction

2. Preliminaries
   i. AMP Algorithm
   ii. Consensus Propagation

3. Proposed Method: Distributed AMP Algorithm

4. Simulation Result

5. Conclusion
Simulation Result

Graph Structure

number of nodes: $K = 50$

graph diameter: 6
Problem Settings (Sparse Vector Reconstruction)

unknown sparse vector \( \mathbf{x} \in \mathbb{R}^N \)

probability distribution (unknown):
\[
p(x_n) = q \delta(x_n) + (1 - q) \phi(x_n)
\]

probability density function of standard Gaussian distribution

\( \mathbf{x} \) is sparse when \( q \) is large

simulation result

communication link

node \( k \)

measurement matrix: \( A_k \in \mathbb{R}^{M_k \times N} \)

measurement vector: \( y_k = A_k \mathbf{x} + \mathbf{v}_k \in \mathbb{R}^{M_k} \)
Simulation Result

MSE for Sparse Vector Reconstruction
(Mean-Square-Error)

- Centralized AMP
- Distributed AMP ($T' = 6$)
- Distributed AMP ($T' = 5$)
- Distributed AMP ($T' = 4$)

$N = 1000$
$M_k = 6$
$\sigma_v^2 = 0.1$
$q = 0.95$

# of iterations in summation propagation

MSEs at $K = 50$ nodes when $T' = 4$
MSEs at $K = 50$ nodes when $T' = 5$
When $T' = 6$ (graph diameter), the performance is the same as the centralized AMP algorithm.

Simulation Result

MSE for Sparse Vector Reconstruction

(Mean-Square-Error)

- $N = 1000$
- $M_k = 6$
- $\sigma_v^2 = 0.1$
- $q = 0.95$

# of iterations in summation propagation

MSEs at $K = 50$ nodes when $T' = 4$

MSEs at $K = 50$ nodes when $T' = 5$

When $T' = 6$ (graph diameter), the performance is the same as the centralized AMP algorithm.
Outline

1. Introduction

2. Preliminaries
   i. AMP Algorithm
   ii. Consensus Propagation

3. Proposed Method: Distributed AMP Algorithm

4. Simulation Result

5. Conclusion
Conclusion

Purpose of This Study

propose a **fully distributed** AMP algorithm, which does not require any fusion node

1. obtain update equations of the AMP algorithm for distributed measurements

    ![local computation at each node](image)

    ![global computation using communications](image)

2. propose **summation propagation** for the global computation

3. show the validity of the proposed algorithm via computer simulation

Future Work

- extension for generalized AMP algorithm
- comparison with conventional methods
We can apply the AMP algorithm for binary vector reconstruction by using another function as $\eta(\cdot;\cdot)$
Appendix

Success Rate for Binary Vector Reconstruction

$N = 1000$
$M_1 = \cdots = M_K = \frac{\Delta N}{K}$
$\sigma_v^2 = 1$
$T = 50$

$p_1 = 0.6$
$p_1 = 0.9$

# of iterations in summation propagation

$\Delta$ (measurement ratio)