Message-passing Algorithms for Approximate Singular Vector Computation

Sewoong Oh

Massachusetts Institute of Technology
joint work with David R. Karger(MIT) and Devavrat Shah(MIT)
Outline

1. Inference Problem in Crowdsourcing

2. Analysis of the Message-passing Algorithm
## Crowdsourcing

<table>
<thead>
<tr>
<th></th>
<th>Speed/Cost</th>
<th>Reliability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Undergrad intern:</td>
<td>300 images/hr, cost: $15/hr</td>
<td>95%</td>
</tr>
<tr>
<td>MTurk (single label):</td>
<td>3000 images/hr, cost: $15/hr</td>
<td>65%</td>
</tr>
</tbody>
</table>
Crowdsourcing

<table>
<thead>
<tr>
<th></th>
<th>Speed/Cost</th>
<th>Reliability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Undergrad intern</td>
<td>300 images/hr, cost: $15/hr</td>
<td>95%</td>
</tr>
<tr>
<td>MTurk (single label)</td>
<td>3000 images/hr, cost: $15/hr</td>
<td>65%</td>
</tr>
<tr>
<td>MTurk (multiple labels)</td>
<td>600 images/hr, cost: $15/hr</td>
<td>95%</td>
</tr>
</tbody>
</table>
Crowdsourcing system

Goal: Reliably estimate the tasks at minimum cost

Challenges:
1. Task Assignment
2. Inference Problem
Crowdsourcing system

Goal: Reliably estimate the tasks at minimum cost

Challenges:
1. Task Assignment
2. Inference Problem
Crowdsourcing system

Goal: Reliably estimate the tasks at minimum cost

Challenges:
1. Task Assignment
2. Inference Problem
Crowdsourcing system

Goal: Reliably estimate the tasks at minimum cost

Challenges:
1. Task Assignment
2. Inference Problem
Crowdsourcing system

Goal: Reliably estimate the tasks at minimum cost

Challenges:
1. Task Assignment
2. Inference Problem
Crowdsourcing system

Goal: Reliably estimate the tasks at minimum cost

Challenges:
1. Task Assignment
2. Inference Problem
Crowdsourcing system

Goal: Reliably estimate the tasks at minimum cost

Challenges:
1. Task Assignment
2. Inference Problem
Crowdsourcing system

Goal: Reliably estimate the tasks at minimum cost

Challenges:
1. Task Assignment
2. Inference Problem
Goal: Reliably estimate the tasks at minimum cost

Challenges:
1. Task Assignment
2. Inference Problem
Previous approaches neglect the role of task assignment

1. Task Assignment Which task should be assigned to which batch?
2. Inference Problem What is the correct answer to the tasks?

Previous work

- **Do not** address the role of task assignment
- Focuses on inference problem given model
  - [Dawid, Skene '79] [Smyth et al. '95] [Sheng, Provost, Ipeirotis '08]
  - Expectation Maximization (EM) based heuristics
    - converges to local minimum
    - sensitive to initialization
    - no performance guarantees
Addressing both challenges together

1. Task Assignment  Which task should be assigned to which batch?
2. Inference Problem  What is the correct answer to the tasks?

Our approach:

1. Task Assignment  $\rightarrow$ Random Graph
2. Inference Problem  $\rightarrow$ Low-rank Matrix Approximation

Extremely efficient and asymptotically order-optimal!
Preview of main result

Probability of Error vs. Labels per image

- 1 label/image (35% Error)
- 15 labels/image (1% Error)

- Majority Voting
- EM Algorithm
- Our Approach
- Lower Bound
Random $(\ell, r)$-regular bipartite graphs
Random ($\ell, r$)-regular bipartite graphs have good properties:

- **Locally Tree-like**
- **Good Expander**

→ Sharpen Analysis

→ High Signal-to-Noise Ratio
Modeling the crowd

Binary tasks: \( t_i \in \{+1, -1\} \)

Worker reliability: \( p_j \in [0, 1] \)

\[
A_{ij} = \begin{cases} 
  t_i & \text{with probability } p_j \\
  -t_i & \text{with probability } 1 - p_j
\end{cases}
\]

\( A_{ij} \)'s are independent conditioned on \( t_i \) and \( p_j \)

\( p_j \) drawn i.i.d. from \( \mathcal{F}(p_j) \)

Necessary to assume we know whether \( \mathbb{E}_{\mathcal{F}}[p_j] > 0.5 \) or not
Modeling the crowd

Tasks

Underlying assumptions

▶ Heterogeneous workers
▶ Homogeneous tasks
▶ Unbiased workers
Inference problem

- Given the responses \( \{A_{ij}\} \)
- Infer the answers \( \{t_i\} \)

\[
\hat{t}_i = \text{sign} \left( \sum_j A_{ij} \right)
\]

Majority Voting:

Oracle Estimator who knows \( p_j \)’s:

\[
\hat{t}_i = \text{sign} \left( \sum_j \log \left( p_{j1} - p_{j2} \right) A_{ij} \right)
\]

Our Approach:

Singular vectors

<table>
<thead>
<tr>
<th>Probability of error</th>
<th>Cost = Number of assignments per task</th>
</tr>
</thead>
<tbody>
<tr>
<td>1e-05</td>
<td>1</td>
</tr>
<tr>
<td>0.0001</td>
<td>5</td>
</tr>
<tr>
<td>0.001</td>
<td>10</td>
</tr>
<tr>
<td>0.01</td>
<td>15</td>
</tr>
<tr>
<td>0.1</td>
<td>20</td>
</tr>
<tr>
<td>1</td>
<td>25</td>
</tr>
<tr>
<td>5</td>
<td>30</td>
</tr>
</tbody>
</table>

\[
1e-05
0.0001
0.001
0.01
0.1
1
5  10  15  20  25  30
Probability of error
Cost = Number of assignments per task

Majority Voting
Oracle Estimator

9 / 29
Inference problem

- Given the responses \( \{A_{ij}\} \)
- Infer the answers \( \{t_i\} \)

Majority Voting:
\[
\hat{t}_i = \text{sign} \left( \sum_j A_{ij} \right)
\]
Inference problem

- Given the responses \(\{A_{ij}\}\)
- Infer the answers \(\{t_i\}\)

- Majority Voting:
  \[ \hat{t}_i = \text{sign} \left( \sum_j A_{ij} \right) \]

- Oracle Estimator who knows \(p_j\)'s:
  \[ \hat{t}_i = \text{sign} \left( \sum_j \log \left( \frac{p_j}{1 - p_j} \right) A_{ij} \right) \]
Inference problem

- Given the responses \{A_{ij}\}
- Infer the answers \{t_i\}

Majority Voting:
\[ \hat{t}_i = \text{sign} \left( \sum_j A_{ij} \right) \]

Oracle Estimator who knows \(p_j\)'s:
\[ \hat{t}_i = \text{sign} \left( \sum_j \log \left( \frac{p_j}{1 - p_j} \right) A_{ij} \right) \]
Inference problem

- Given the responses \( \{A_{ij}\} \)
- Infer the answers \( \{t_i\} \)

\[
A_{ij}
\]

\[
t_i
\]

\[
p_{j1}
\]

\[
p_{j2}
\]

\[
p_{j3}
\]

\[
t_i
\]

\[
p_{j1}
\]

\[
p_{j2}
\]

\[
p_{j3}
\]

\[
+t_1 - p_2 + p_3
\]

Majority Voting:
\[
\hat{t}_i = \text{sign}\left(\sum_j A_{ij}\right)
\]

Oracle Estimator who knows \(p_j\)’s:
\[
\hat{t}_i = \text{sign}\left(\sum_j \log \left(\frac{p_j}{1 - p_j}\right) A_{ij}\right)
\]

Our Approach:
- Singular vectors

Cost = Number of assignments per task

- Majority Voting
- Oracle Estimator
- Iterative Algorithm
Inference using singular vectors

$$\mathbb{E}[A_{ij}|t_i, p_j] = t_i(2p_j - 1)$$
Inference using singular vectors

\[ \mathbb{E}[A_{ij} | t_i, p_j] = t_i(2p_j - 1) \]

- **Algorithm**: 
  1. Compute the first singular vectors \((u, v)\) of \(A\)
  2. Output estimate: \(\hat{t}_i = \text{sign}(u_i)\)

\begin{align*}
\text{data} & \quad \mathbb{E}[A | \{t_i\}, \{p_j\}] & \quad \text{Random Perturbation} \\
\begin{array}{cccccccc}
- & + & - & - \\
+ & - & + & - \\
+ & - & + & - \\
- & - & - & - \\
\end{array} & \begin{array}{cccccccc}
- & + & - & - \\
+ & - & + & - \\
+ & - & + & - \\
- & - & - & - \\
\end{array} & \begin{array}{cccccccc}
\text{low-rank signal} & + & \text{noise} \\
\end{array}
\end{align*}
Inference using singular vectors

\[ \mathbb{E}[A_{ij} | t_i, p_j] = t_i(2p_j - 1) \]

Algorithm:
1. Compute the first singular vectors \((u, v)\) of \(A\)
2. Output estimate: \(\hat{t}_i = \text{sign}(u_i)\)

Problem
- [Achlioptas, McSherry 02], [Keshavan, Montanari, Oh 10]
- Current analysis techniques do not give tight bounds
Message-passing approach gives sharper analysis

- **Power Iteration:**

  Iterate:
  \[
  \tilde{T}_i = \sum_{j \in \partial i} \tilde{W}_j A_{ij}, \quad \tilde{W}_j = \sum_{i \in \partial j} \tilde{T}_i A_{ij}
  \]
Message-passing approach gives sharper analysis

- **Power Iteration:**
  
  Iterate: 
  \[
  \tilde{T}_i = \sum_{j \in \partial i} \tilde{W}_j A_{ij}, \quad \tilde{W}_j = \sum_{i \in \partial j} \tilde{T}_i A_{ij}
  \]

- **Message-passing Power Iteration:**
  Iteratively update messages \( \{T_{i \rightarrow j}\} \) and \( \{W_{j \rightarrow i}\} \)

  **Task-likelihood update**
  
  \[
  T_{i \rightarrow j} = \sum_{j' \in \partial i \setminus j} W_{j' \rightarrow i} A_{ij'}
  \]

  **Worker-reliability update**
  
  \[
  W_{j \rightarrow i} = \sum_{i' \in \partial j \setminus i} T_{i' \rightarrow j} A_{i'j}
  \]
Message-passing approach gives sharper analysis

- **Message-passing Power Iteration:**

  **Task-likelihood update**

  
  \[ T_{i \rightarrow j} = \sum_{j' \in \partial i \setminus j} W_{j' \rightarrow i} A_{ij'} \]

  **Worker-reliability update**

  
  \[ W_{j \rightarrow i} = \sum_{i' \in \partial j \setminus i} T_{i' \rightarrow j} A_{i'j} \]

  **Decision update**

  
  \[ T_i = \sum_{j \in \partial i} W_{j \rightarrow i} A_{ij} \]

  \[ \hat{t}_i = \text{sign}(T_i) \]
Message-passing approach gives sharper analysis

- **Message-passing Power Iteration:**
  - Task-likelihood update
    \[ T_{i \rightarrow j} = \sum_{j' \in \partial i \setminus j} W_{j' \rightarrow i} A_{ij'} \]
  - Worker-reliability update
    \[ W_{j \rightarrow i} = \sum_{i' \in \partial j \setminus i} T_{i' \rightarrow j} A_{i'j} \]

- **Decision update**
  \[ T_i = \sum_{j \in \partial i} W_{j \rightarrow i} A_{ij} \]
  \[ \hat{t}_i = \text{sign}(T_i) \]

---

**Our Inference Algorithm**

- Requires no knowledge of \( \mathcal{F} \)
- Provide sharp bound on the probability of error
Performance Analysis
Performance depends on collective quality of the crowd

Probability of error depends on $\mathcal{F}(\cdot)$ through Crowd Quality Parameter:

$$q \equiv \mathbb{E}_{\mathcal{F}}[(2p_j - 1)^2]$$

- Phase transition at $q^2(\ell - 1)(r - 1) = 1$
  - $q^2(\ell - 1)(r - 1) < 1$: Iteration Hurts
  - $q^2(\ell - 1)(r - 1) > 1$: Iteration Helps

---

Majority Voting
EM Algorithm
Message-passing
Oracle Estimator
Sharp bound on the probability of error

Theorem. [Karger, O., Shah ’11]

In the large system limit, and for \((\ell - 1)(r - 1)q^2 > 1\), using a random graph and \(k\) iterations of the message-passing algorithm achieves

\[
\lim_{m \to \infty} \frac{1}{m} \sum_{i=1}^{m} \mathbb{P}(t_i \neq \hat{t}_i) \leq \exp \left\{ -\frac{q\ell}{2\sigma_k^2} \right\},
\]

for \(\sigma_k^2 \equiv \left(3 + \frac{1}{qr}\right) \frac{q^2(\ell-1)(r-1)}{q^2(\ell-1)(r-1)-1} + \frac{2}{q}(q^2(\ell - 1)(r - 1))^{-k} \).
Upper bound matches the minimax lower bound

- **Message-passing algorithm** $(\ell \leq r, q(\ell - 1) > 1, k \to \infty)$:
  \[
P_{\text{error}} \leq e^{-\frac{1}{8}(q\ell-1)}
  \]

- **Matching minimax lower bound**:
  \[
  \inf_{\text{Alg}, G(\ell)} \sup \{ t_i \}, \{ \mathcal{F} | \mathbb{E}[(2p-1)^2] = q \} \quad P_{\text{error}} \geq \frac{1}{2} e^{-(q\ell+O(q^2\ell))}
  \]

- **Majority Voting**:
  \[
  \inf_{G(\ell)} \sup \{ t_i \}, \{ \mathcal{F} | \mathbb{E}[(2p-1)^2] = q \} \quad P_{\text{error}} \geq e^{-C(q^2\ell+1)}
  \]

---

**Graph**

- **Majority Voting**
- **EM Algorithm**
- **Message-passing**
- **Oracle Estimator**
Our approach is asymptotically order-optimal

\[ P_{\text{error}} \leq e^{-\frac{1}{8}(q\ell - 1)} \]

**How much do we need to spend to achieve** \( P_{\text{error}} \leq \epsilon \)?

- **Sufficient** to choose \( \ell = O\left(\frac{1}{q} \log\left(\frac{1}{\epsilon}\right)\right) \)
- **Necessary** to have \( \ell = \Omega\left(\frac{1}{q} \log\left(\frac{1}{\epsilon}\right)\right) \)
Proof Sketch
Recap

- **Graphical model**
  - task variables: $t_i \in \{\pm 1\}$
  - worker variables: i.i.d. $p_j \sim \mathcal{F}(\cdot)$

\[
A_{ij} = \begin{cases} 
  t_i & \text{with probability } p_j \\
  -t_i & \text{with probability } 1 - p_j
\end{cases}
\]
Recap

- **Graphical model**
  - task variables: \( t_i \in \{ \pm 1 \} \)
  - worker variables: i.i.d. \( p_j \sim \mathcal{F}(\cdot) \)

\[
A_{ij} = \begin{cases} 
  t_i \text{ with probability } p_j \\
  -t_i \text{ with probability } 1 - p_j 
\end{cases}
\]

- **Message-passing Power Iteration:**

\[
\begin{align*}
T_{i \rightarrow j}^{(k)} &= \sum_{j' \in \partial i \setminus j} W_{j' \rightarrow i}^{(k-1)} A_{ij'} \\
W_{j \rightarrow i}^{(k)} &= \sum_{i' \in \partial j \setminus i} T_{i' \rightarrow j}^{(k)} A_{i'j}
\end{align*}
\]

\[
T_i^{(k)} = \sum_{j \in \partial i} W_{j \rightarrow i}^{(k-1)} A_{ij} \\
\hat{t}_i^{(k)} = \text{sign}(T_i^{(k)})
\]

![Graphical model diagram](image)

**Performance metric:** (by symmetry, assume all \( t_i = +1 \))

\[
\lim_{m \to \infty} \frac{1}{m} \sum_{i=1}^{m} P(\hat{t}_i^{(k)} = -1) = \lim_{m \to \infty} \frac{1}{m} \sum_{i=1}^{m} P(T_i^{(k)} \leq 0)
\]
Recap

- **Graphical model**
  - task variables: \( t_i \in \{ \pm 1 \} \)
  - worker variables: i.i.d. \( p_j \sim \mathcal{F}(\cdot) \)

\[
A_{ij} = \begin{cases} 
  t_i & \text{with probability } p_j \\
  -t_i & \text{with probability } 1 - p_j 
\end{cases}
\]

- **Message-passing Power Iteration:**

\[
\mathcal{T}_{i \rightarrow j}^{(k)} = \sum_{j' \in \partial i \setminus j} W_{j' \rightarrow i}^{(k-1)} A_{ij'}
\]
\[
W_{j \rightarrow i}^{(k)} = \sum_{i' \in \partial j \setminus i} \mathcal{T}_{i' \rightarrow j}^{(k)} A_{i'j}
\]

- **Final Estimate**

\[
\hat{t}_i^{(k)} = \text{sign}(\mathcal{T}_{i}^{(k)})
\]

\[
\mathcal{T}_{i}^{(k)} = \sum_{j \in \partial i} W_{j \rightarrow i}^{(k-1)} A_{ij}
\]

- **Performance metric:** (by symmetry, assume all \( t_i = +1 \))

\[
\lim_{m \rightarrow \infty} \frac{1}{m} \sum_{i=1}^{m} \mathbb{P}(\hat{t}_i^{(k)} = -1) = \lim_{m \rightarrow \infty} \frac{1}{m} \sum_{i=1}^{m} \mathbb{P}(\mathcal{T}_{i}^{(k)} \leq 0)
\]
Message Update

Initialize: 
\[ W_{j \rightarrow i}^{(0)} \sim \mathcal{N}(1, 1) \]

Density Evolution

\[ W^{(0)} \sim \mathcal{N}(1, 1) \]
Message Update

Initialize:

\[ W_{j \rightarrow i}^{(0)} \sim \mathcal{N}(1, 1) \]

1st step:

\[ T_{i \rightarrow j}^{(1)} = \sum_{j' \in \partial i \setminus j} W_{j' \rightarrow i}^{(0)} A_{ij'} \]
\[ W_{j \rightarrow i}^{(1)} = \sum_{i' \in \partial j \setminus i} T_{i' \rightarrow j}^{(1)} A_{i'j} \]

Density Evolution

\[ W^{(0)} \sim \mathcal{N}(1, 1) \]

\[ T^{(1)} \overset{d}{=} \sum_{j=1}^{\ell-1} W_{j}^{(0)} A_{p_{j}, j} \]
\[ W_{p}^{(1)} \overset{d}{=} \sum_{i=1}^{r-1} T_{i}^{(1)} A_{p, i} \]

\[ A_{p} = \begin{cases} +1 \text{ w.p. } p \\ -1 \text{ w.p. } 1 - p \end{cases} \]
\[ p \sim \mathcal{F} \]
Message Update

Initialize:

\[ W_{j \rightarrow i}^{(0)} \sim \mathcal{N}(1, 1) \]

\[ W^{(0)} \sim \mathcal{N}(1, 1) \]

\[ k \text{-th step:} \]

\[ T_{i \rightarrow j}^{(k)} = \sum_{j' \in \partial i \setminus j} W_{j' \rightarrow i}^{(k-1)} A_{ij'} \]

\[ T^{(k)} = d \sum_{j=1}^{\ell-1} W_{p_j,j}^{(k-1)} A_{pj,j} \]

\[ W_{j \rightarrow i}^{(k)} = \sum_{i' \in \partial j \setminus i} T_{i' \rightarrow j}^{(k)} A_{i'j} \]

\[ W_{p}^{(k)} = d \sum_{i=1}^{r-1} T_{i}^{(k)} A_{p,i} \]

\[ T_{i}^{(k)} = \sum_{j' \in \partial i} W_{j' \rightarrow i}^{(k-1)} A_{ij'} \]

\[ \hat{T}^{(k)} = d \sum_{j=1}^{\ell} W_{p_j,j}^{(k-1)} A_{pj,j} \]

Density Evolution

Challenges:

- Random variables take real values
- Cannot describe/track the probability density functions
- Cannot evaluate

\[ P_{\text{error}} = P(\hat{T}^{(k)} \leq 0) \]
Message Update

Initialize:

\[ W_{j\rightarrow i}^{(0)} \sim \mathcal{N}(1, 1) \]

\[ W^{(0)} \sim \mathcal{N}(1, 1) \]

\[ k \)-th step:

\[ T_{i\rightarrow j}^{(k)} = \sum_{j' \in \partial i \setminus j} W_{j'\rightarrow i}^{(k-1)} A_{ij'} \]

\[ W_{j\rightarrow i}^{(k)} = \sum_{i' \in \partial j \setminus i} T_{i'\rightarrow j}^{(k)} A_{i'j} \]

\[ T_{i}^{(k)} = \sum_{j' \in \partial i} W_{j'\rightarrow i}^{(k-1)} A_{ij'} \]

\[ \hat{T}^{(k)} \equiv \sum_{j=1}^{\ell-1} W_{p,j}^{(k-1)} A_{p,j} \]

\[ T^{(k)} \equiv \sum_{j=1}^{r-1} T_{i}^{(k)} A_{p,i} \]

\[ \hat{T}^{(k)} \equiv \sum_{j=1}^{\ell} W_{p,j}^{(k-1)} A_{p,j} \]

Density Evolution

Challenges

- Random variables take real values
- Cannot describe/track the probability density functions
- Cannot evaluate \( P_{\text{error}} = \mathbb{P}(\hat{T}^{(k)} \leq 0) \)
Find a simple structure that describes the messages

- Histogram of $T_{i \rightarrow j}$'s

1st iteration
Find a simple structure that describes the messages

- Histogram of $T_{i\rightarrow j}$

2nd iteration
Find a simple structure that describes the messages

- Histogram of $T_{i \rightarrow j}$'s

3rd iteration
Find a simple structure that describes the messages

- Histogram of $T_{i \rightarrow j}$’s

4th iteration
Find a simple structure that describes the messages

- Histogram of $T_{i\rightarrow j}$’s

5th iteration
Find a simple structure that describes the messages

- Histogram of $T_{i \rightarrow j}$’s

6th iteration
Find a simple structure that describes the messages

- Histogram of $T_{i\rightarrow j}$’s

7th iteration
Find a simple structure that describes the messages

- Histogram of $T_{i \rightarrow j}$’s

9th iteration
Find a simple structure that describes the messages

- Histogram of $T_{i \rightarrow j}$’s

15th iteration
Find a simple structure that describes the messages

- Histogram of $T_{i \rightarrow j}$’s

20th iteration

Strategy: Find simple structure to describe the messages

- $T^{(k)}$’s are sub-Gaussian
- Two parameters: mean $m_k$, scale factor $\sigma_k^2$
- Can be solved in a closed form
Sub-Gaussian random variables

- $X$ is sub-Gaussian with mean $m$ and scale factor $\sigma^2$ if
  \[ E[e^{\lambda X}] \leq e^{m\lambda + \frac{1}{2}\sigma^2\lambda^2} \]

- Applying Chernoff’s inequality
  \[ P(X \leq m - a\sigma) \leq E[e^{X\lambda - m\lambda + a\sigma\lambda}] \leq e^{\frac{1}{2}\sigma^2\lambda^2 + a\sigma\lambda} \leq e^{-\frac{1}{2}a^2} \]
Sub-Gaussian random variables

- $X$ is sub-Gaussian with mean $m$ and scale factor $\sigma^2$ if
  \[ \mathbb{E}[e^{\lambda X}] \leq e^{m\lambda + \frac{1}{2} \sigma^2 \lambda^2} \]

- Applying Chernoff’s inequality
  \[ P(X \leq m - a\sigma) \leq \mathbb{E}[e^{X\lambda - m\lambda + a\sigma \lambda}] \leq e^{\frac{1}{2} \sigma^2 \lambda^2 + a\sigma \lambda} \leq e^{-\frac{1}{2} a^2} \]

- $\hat{T}^{(k)}$ is sub-Gaussian with mean $m_k$ and scale factor $\sigma_k^2$
  \[ P(\hat{T}^{(k)} \leq 0) \leq e^{-\frac{m_k^2}{2\sigma_k^2}} \]
Sub-Gaussian random variables

- \( \mathbf{X} \) is sub-Gaussian with mean \( m \) and scale factor \( \sigma^2 \) if
  \[
  \mathbb{E}[e^{\lambda \mathbf{X}}] \leq e^{m \lambda + \frac{1}{2} \sigma^2 \lambda^2}
  \]

- Applying Chernoff’s inequality
  \[
  \mathbb{P}(\mathbf{X} \leq m - a \sigma) \leq \mathbb{E}[e^{\mathbf{X} \lambda - m \lambda + a \sigma \lambda}] \leq e^{\frac{1}{2} \sigma^2 \lambda^2 + a \sigma \lambda} \leq e^{-\frac{1}{2} a^2}
  \]

- \( \hat{\mathbf{T}}(k) \) is sub-Gaussian with mean \( m_k \) and scale factor \( \sigma_k^2 \)
  \[
  \mathbb{P}^{\hat{\mathbf{T}}(k) \leq 0} \leq e^{-\frac{m_k^2}{2 \sigma_k^2}}
  \]

Density evolution on real-values messages can be solved by establishing sub-Gaussianity
Experiments with real crowdsourcing tasks

- Learning similarities using Crowdsourcing
  - recommendations
  - searching

You wanted to get the tie on the top, but it was not available. Which one would be a better substitute?

- [ ] the tie the left
- [x] the tie on the right
Experiments with real crowdsourcing tasks

- Crowdsourcing can handle much more complex tasks

Which tie matches my shirt better?
- [ ] the tie on the left
- [x] the tie on the right
Message-passing algorithm improves over existing algorithms

Which colors...  

left  
right

Number of responses per task

Probability of error

0.14  EM Algorithm
0.12  Majority voting
0.06  Message-passing
Message-passing algorithm improves over existing algorithms

![Graph showing average probability of error vs. number of responses per task](image)

- **EM Algorithm**
- **Majority voting**
- **Message-passing**

Which colors... (indicate left and right with check marks)

- **left**
- **right**
Gain due to random task assignment is significant

- How does the graph change the performance?

**Average probability of error**

![Graph showing the performance of EM Algorithm and Message-passing algorithms with varying number of responses per task.](image)

- EM Algorithm
- Message-passing
Gain due to random task assignment is significant

- How does the graph change the performance?

### Graphs with small spectral gap

- EM Algorithm
- Message-passing

- Deterministic graphs with block-diagonal adjacency matrices
- Random graphs are good guidelines
  - Possible to generate deterministic graphs that are locally tree-like and good expanders
Under adaptive/sequential scenario

- Create batches **adaptively**
- Cannot explore/exploit reliable workers
Under adaptive/sequential scenario

- Create batches adaptively
- Cannot explore/exploit reliable workers
Under adaptive/sequential scenario

- Create batches adaptively
- Cannot explore/exploit reliable workers
Under adaptive/sequential scenario

- Create batches adaptively
- Cannot explore/exploit reliable workers
Under adaptive/sequential scenario, our approach is still asymptotically order-optimal.

**Theorem. [Karger, O., Shah ’11]**

Let $\Delta$ be the budget necessary to achieve a target accuracy $P_{error} \leq \epsilon$. Then, for any $Alg$ that can adaptively assign tasks,

$$\inf_{Alg} \sup_{\{t_i\}, \{p_j\} \in \mathcal{F}(q)} \mathbb{E}[\Delta] = \Omega \left( \frac{1}{q} \log \left( \frac{1}{\epsilon} \right) \right)$$

- Still **Necessary** to have budget $\geq C \frac{1}{q} \log(\frac{1}{\epsilon})$
- No significant gain in using an **adaptive** task allocation
When workers are biased

- Binary tasks: $t_i \in \{+1, -1\}$
- Worker reliability: $p_j^+, p_j^- \in [0, 1]$

If $t_i = +1$, then

$$A_{ij} = \begin{cases} 
  t_i & \text{w.p. } p_j^+ \\
  -t_i & \text{w.p. } 1 - p_j^+ 
\end{cases}$$

If $t_i = -1$, then

$$A_{ij} = \begin{cases} 
  t_i & \text{w.p. } p_j^- \\
  -t_i & \text{w.p. } 1 - p_j^- 
\end{cases}$$
When workers are biased

- Binary tasks: \( t_i \in \{+1, -1\} \)
- Worker reliability: \( p_j^+, p_j^- \in [0, 1] \)

If \( t_i = +1 \), then

\[
A_{ij} = \begin{cases} 
  t_i & \text{w.p. } p_j^+ \\
  -t_i & \text{w.p. } 1 - p_j^+
\end{cases}
\]

If \( t_i = -1 \), then

\[
A_{ij} = \begin{cases} 
  t_i & \text{w.p. } p_j^- \\
  -t_i & \text{w.p. } 1 - p_j^-
\end{cases}
\]
**Conclusion**

**Goal**  Design reliable and cost-efficient crowdsourcing systems

- **Contributions**
  - First to address both task assignment and inference problem
  - Message-passing Power Iteration Algorithm
  - Establish sub-Gaussianity to prove sharp bounds