Iterative Learning for Reliable Crowdsourcing Systems

Bin Bi, Chen Liu, Yuchen Liu

Agenda

- Introduction
- Crowdsourcing Model
- Proposed Algoirthms
- Performance Guarantee and Optimality
- Density Evolution Analysis Technique (Proof of Thm. 2.1)

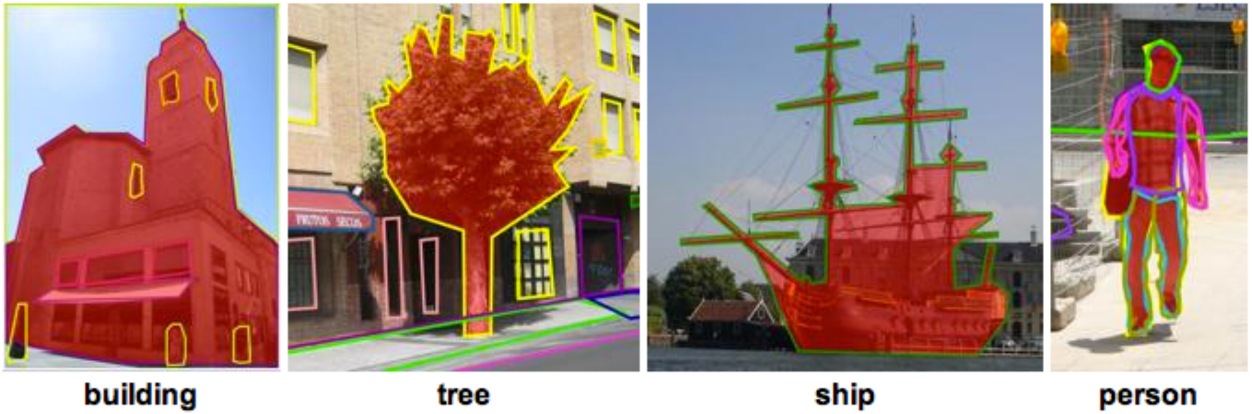
Agenda

- Introduction
- Crowdsourcing Model
- Proposed Algorithms
- Performance Guarantee and Optimality
- Density Evolution Analysis Technique (Proof of Thm. 2.1)

What’s Crowdsourcing

- Image classification
- Transcription
- Proof reading

- Large number of small and simple tasks
- Difficult for computers
- Easy for human



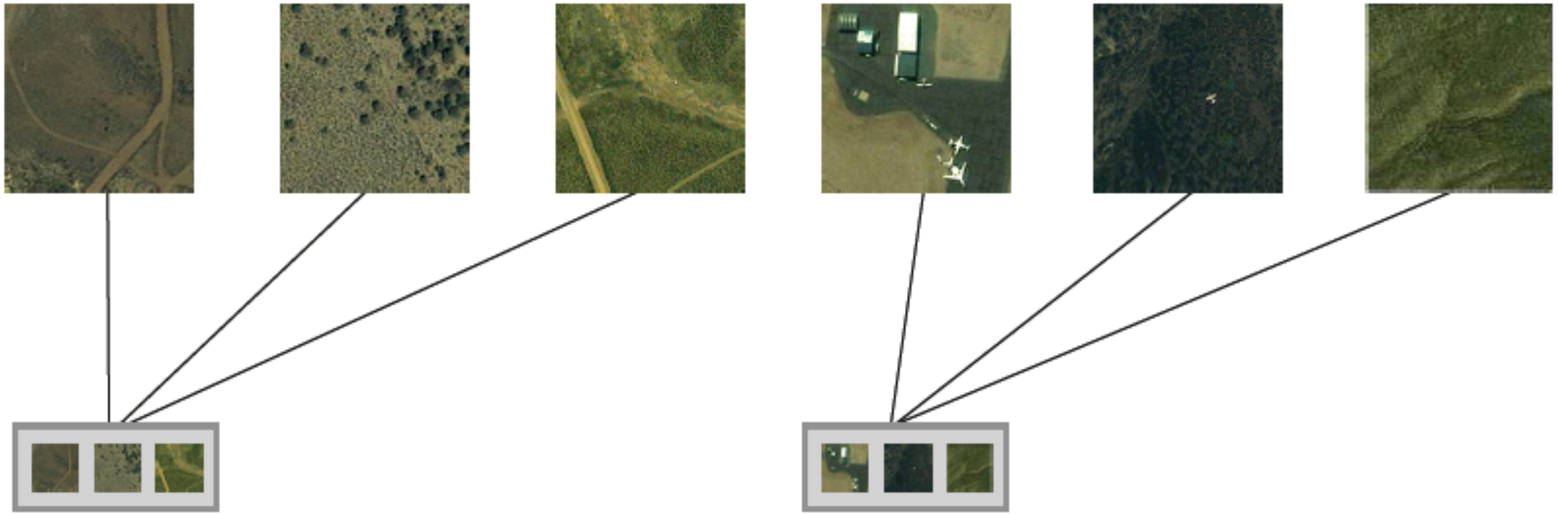
http://labelme.csail.mit.edu/mt_instructions.html

Characteristics of Crowdsourcing System

- Errors are common
  - Some workers are not reliable
- Workers are unidentifiable
  - Worker crowd is large
  - No prior knowledge of the worker's reliability
  - Tasks are distributed through open call
- No gold standard
  - Cannot condition payment on correctness of responses

Crowdsourcing Systems

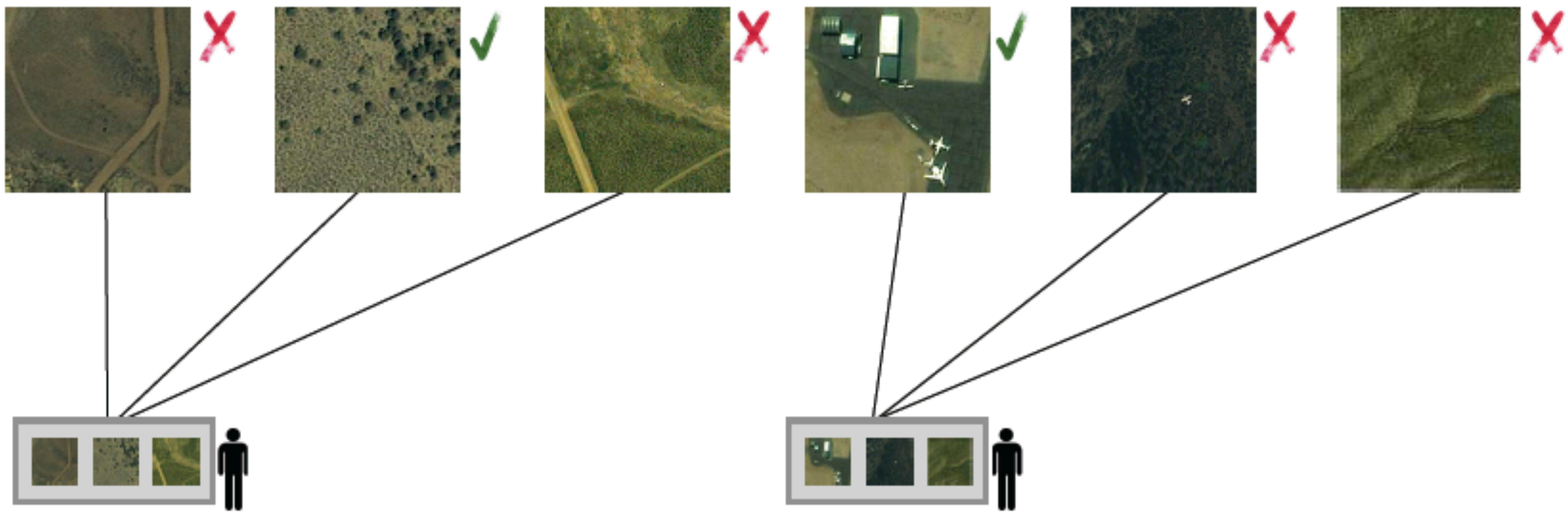
Batches of tasks are distributed to **unidentified** group of people through **open call**.



Source: S. Oh "Iterative Learning for Reliable Crowdsourcing Systems" NIPS 2011

Crowdsourcing Systems

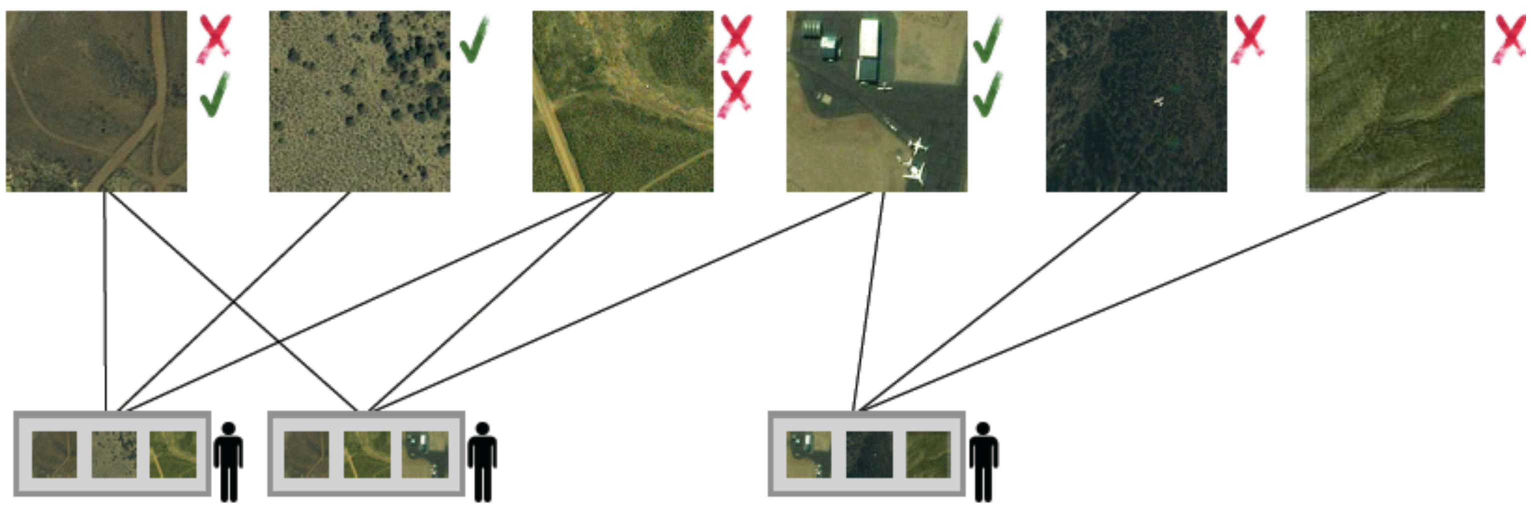
User give their **possibly inaccurate** answers.



Source: S. Oh "Iterative Learning for Reliable Crowdsourcing Systems" NIPS 2011

Crowdsourcing Systems

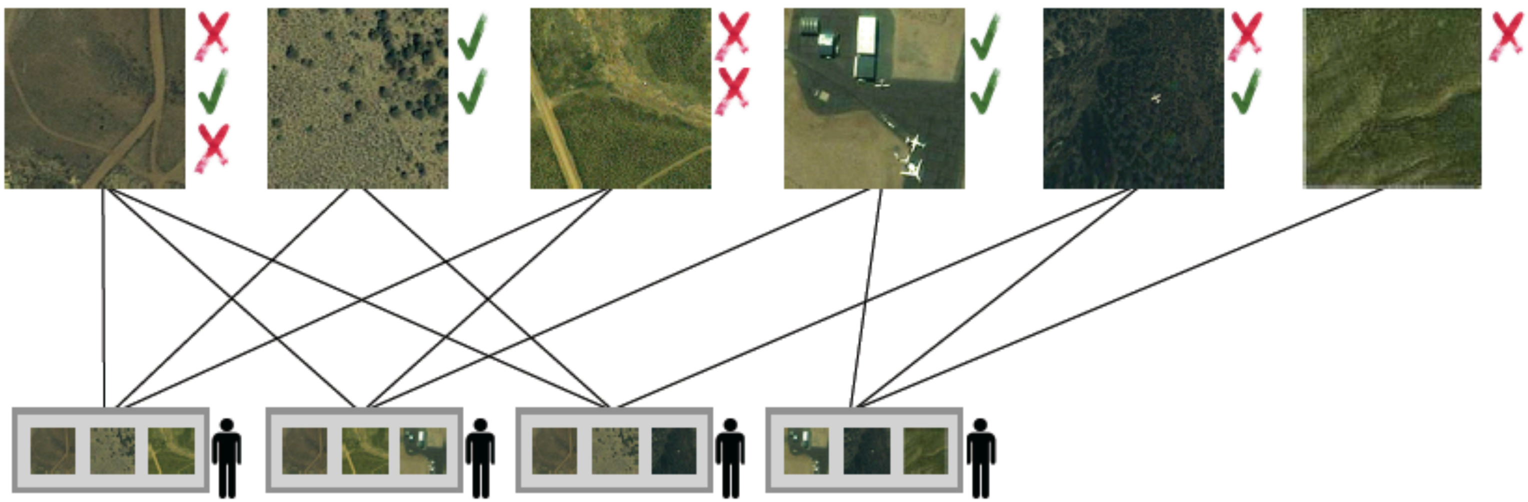
A task may be assigned to multiple workers to overcome the possible errors.



Source: S. Oh "Iterative Learning for Reliable Crowdsourcing Systems" NIPS 2011

Crowdsourcing Systems

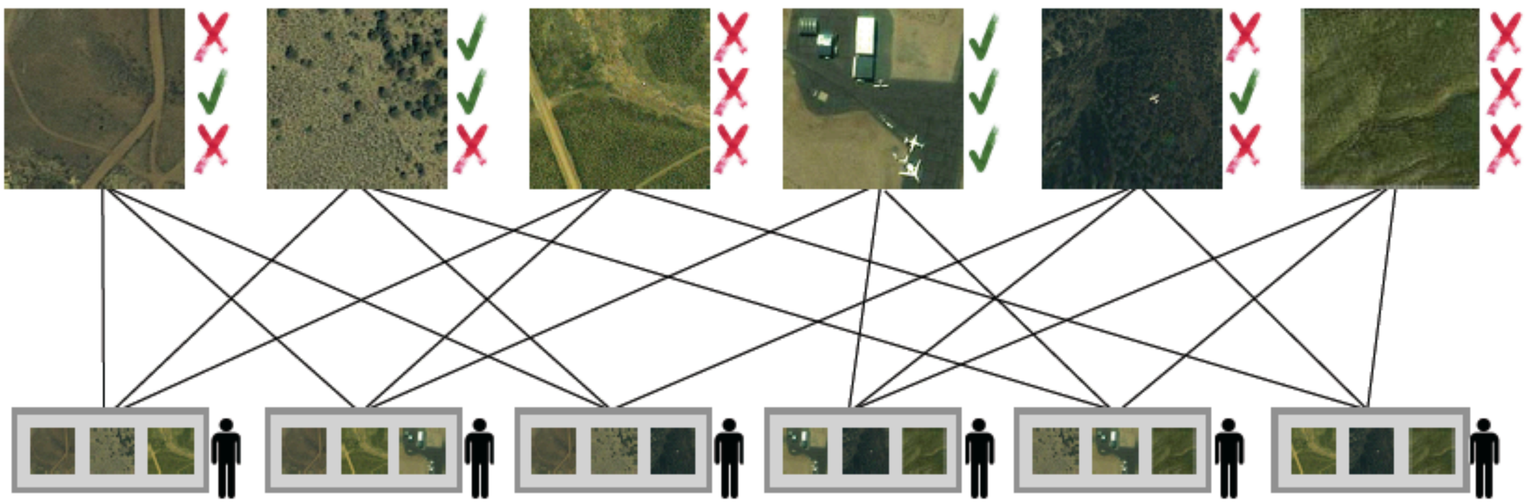
Users make random error based on their own quality.



Source: S. Oh "Iterative Learning for Reliable Crowdsourcing Systems" NIPS 2011

Crowdsourcing Systems

Final results are **aggregation** of multiple workers' response for each task. **Estimation is performed after all the answers are obtained**. Amount of the payment is according to the number of responses.



Source: S. Oh "Iterative Learning for Reliable Crowdsourcing Systems" NIPS 2011

Core Optimization Problem

- Achieve **a certain reliability** in answers with **minimum cost** (i.e. asking fewest possible questions)

Core Optimization Problem

- Achieve a certain reliability in answers with minimum cost (i.e. asking fewest possible questions)
- Challenges
  - Task assignment
  - Inference problem
- Solutions proposed by the paper
  - Task assignment: Random regular bipartite graph
  - Inference problem: Iterative inference algorithm
  - Proved to be optimal given certain amount of budget

Previous Related Work

- Focus on inference problem
- Learning from multiple responses
  - Majority Voting
    - Vulnerable to spammers
  - EM approach to learn reliability
    - Local optimal
    - No theoretical performance guarantee

Agenda

- Introduction
- Crowdsourcing Model
- Proposed Algorithms
- Performance Guarantee and Optimality
- Density Evolution Analysis Technique (Proof of Thm. 2.1)

Crowd Sourcing Model

- A set of m Tasks $\{t_i\}, i = 1, 2, \ldots, m$
- Each task associated with an unobserved 'correct' answer $s_i \in \{\pm 1\}$
- Tasks are assigned to n workers $\{w_j\}, j = 1, \ldots n$
- Answer on task t_i from worker w_j : $A_{ij} \in \{\pm 1\}$

Crowd Sourcing Model (cont.)

- Each worker has a reliability $p_j \in [0,1]$
  - The worker j randomly make errors according to
  - p_j does not depends on specific task p_j
  - $\{p_j\}, j = 1 \ldots n$ are i.i.d. random variables of a given distribution

- For task i answered by user j, the answer is defined as

$$A_{ij} = \begin{cases} s_i & w.p. \quad p_j \\ -s_i & w.p. \quad 1 - p_j \end{cases}$$

  - If i is not assigned to j $A_{ij} = 0$

- Crowd quality

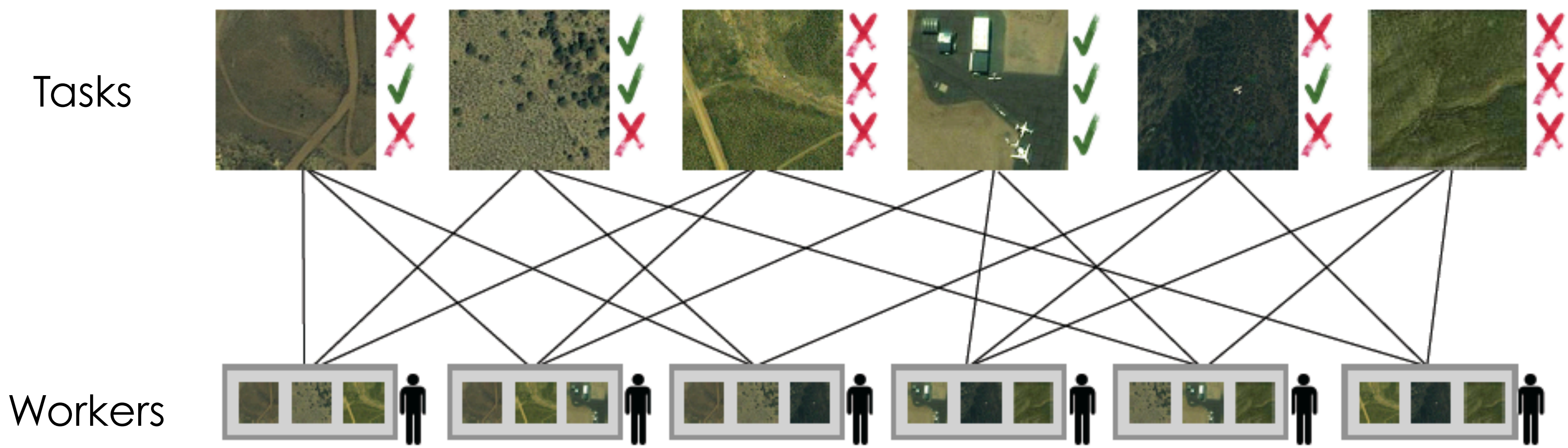
$$q \equiv \mathrm{E}[(2p_j - 1)^2]$$

Agenda

- Introduction
- Crowdsourcing Model
- Proposed Algorithms
- Performance Guarantee and Optimality
- Density Evolution Analysis Technique (Proof of Thm. 2.1)

Task Allocation Scheme

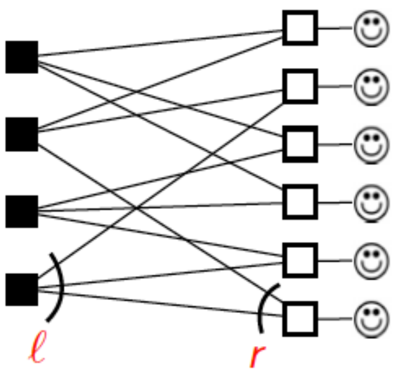
- Task allocation = Designing a bipartite graph



Source: S. Oh "Iterative Learning for Reliable Crowdsourcing Systems" NIPS 2011

Random (l,r)-regular bipartite graph

- Generate an (l,r)-regular random bipartite graph



- Random bipartite graph has good properties
  - Proved to be sufficient to achieve order-optimal performance

Iterative Inference Algorithm

Iterative Algorithm

Input: E, $\{A_{ij}\}_{(i,j)\in E}$, $k_{\max}$

Output: Estimation $\hat{s}(\{A_{ij}\})$

1: **For all** $(i,j) \in E$ **do**

Initialize $y^{(0)}_{j\to i}$ with random $Z_{ij} \sim \mathcal{N}(1,1)$;

2: **For** $k = 1, \ldots, k_{\max}$ **do**

For all $(i,j) \in E$ **do** $x^{(k)}_{i\to j} \leftarrow \sum_{j' \in \partial i \setminus j} A_{ij'} y^{(k-1)}_{j'\to i}$;

For all $(i,j) \in E$ **do** $y^{(k)}_{j\to i} \leftarrow \sum_{i' \in \partial j \setminus i} A_{i'j} x^{(k)}_{i'\to j}$;

3: **For all** $i \in [m]$ **do** $x_i \leftarrow \sum_{j \in \partial i} A_{ij} y^{(k_{\max}-1)}_{j\to i}$;

4: Output estimate vector $\hat{s}(\{A_{ij}\}) = [\mathrm{sign}(x_i)]$.

Iterative Inference Algorithm

- Message passing
  - Task message: $\{x_{i \to j}\}_{(i,j) \in E}$
  - Worker message: $\{y_{j \to i}\}_{(i,j) \in E}$

Iterative Inference Algorithm

- Message passing
  - Task message: $\{x_{i \to j}\}_{(i,j) \in E}$
  - Worker message: $\{y_{j \to i}\}_{(i,j) \in E}$

Worker j's reliability on item i

- Final estimate

$$\hat{s}_i = sign(\sum_{j \in \partial_i} A_{ij} y_{j \to i})$$

Neiborhood of i

A weighted sum of answers weighted by each worker's reliability.

Iterative algorithm for inference

- Update Process
  - Compute the item likelihood to be positive

$$x_{i\to j}^{(k)} \leftarrow \sum_{j'\in\partial i\backslash j} A_{ij'} y_{j'\to i}^{(k-1)}$$

Tasks are more likely to be positive if reliable workers say it is positive

  - Compute the reliability of users

$$y_{j\to i}^{(k)} \leftarrow \sum_{i'\in\partial j\backslash i} A_{i'j} x_{i'\to j}^{(k)}$$

Workers are reliable if their labels consistent with the likelihood of tasks

Agenda

- Introduction
- Crowdsourcing Model
- Proposed Algorithms
- Performance Guarantee and Optimality
- Density Evolution Analysis Technique (Proof of Thm. 2.1)

Performance Guarantee

- Define:

$$\mu \equiv \mathbb{E}[2\mathrm{p}_j - 1] \qquad q = \mathbb{E}[(2\mathrm{p}_j - 1)^2].$$

let $\hat{l} \equiv l - 1$ and $\hat{r} \equiv r - 1$.

Performance Guarantee

- Define:

$$\mu \equiv \mathbb{E}[2\mathbf{p}_j - 1] \qquad q = \mathbb{E}[(2\mathbf{p}_j - 1)^2].$$

let $\hat{l} \equiv l - 1$ and $\hat{r} \equiv r - 1$.

$$\rho_k^2 \equiv \frac{2q}{\mu^2 (q^2 \hat{l}\hat{r})^{k-1}} + \left(3 + \frac{1}{q\hat{r}}\right) \frac{1 - (1/q^2\hat{l}\hat{r})^{k-1}}{1 - (1/q^2\hat{l}\hat{r})} .$$

Performance Guarantee

- Define:

$$\mu \equiv \mathbb{E}[2\mathbf{p}_j - 1] \qquad q = \mathbb{E}[(2\mathbf{p}_j - 1)^2].$$

let $\hat{l} \equiv l - 1$ and $\hat{r} \equiv r - 1$.

$$\rho_k^2 \;\equiv\; \frac{2q}{\mu^2(q^2\hat{l}\hat{r})^{k-1}} + \left(3 + \frac{1}{q\hat{r}}\right)\frac{1 - (1/q^2\hat{l}\hat{r})^{k-1}}{1 - (1/q^2\hat{l}\hat{r})}\;.$$

$$\rho_\infty^2 \;=\; \left(3 + \frac{1}{q\hat{r}}\right)\frac{q^2\hat{l}\hat{r}}{q^2\hat{l}\hat{r} - 1}\;.$$

Performance Guarantee

- Define:

$$\mu \equiv \mathbb{E}[2\mathbf{p}_j - 1] \qquad q = \mathbb{E}[(2\mathbf{p}_j - 1)^2].$$

let $\hat{l} \equiv l - 1$ and $\hat{r} \equiv r - 1$.

$$\rho_k^2 \equiv \frac{2q}{\mu^2 (q^2 \hat{l}\hat{r})^{k-1}} + \left(3 + \frac{1}{q\hat{r}}\right) \frac{1 - (1/q^2\hat{l}\hat{r})^{k-1}}{1 - (1/q^2\hat{l}\hat{r})} .$$

For $q^2\hat{l}\hat{r} > 1$, let $\rho_\infty^2 \equiv \lim_{k\to\infty} \rho_k^2$ such that

$$\rho_\infty^2 = \left(3 + \frac{1}{q\hat{r}}\right) \frac{q^2\hat{l}\hat{r}}{q^2\hat{l}\hat{r} - 1} .$$

Performance Guarantee

Theorem 2.1. *For fixed $l > 1$ and $r > 1$, assume that m tasks are assigned to $n = ml/r$ workers according to a random (l, r)-regular graph drawn from the configuration model. If the distribution of the worker reliability satisfy $\mu \equiv \mathbb{E}[2\mathbf{p}_j - 1] > 0$ and $q^2 > 1/(\hat{l}\hat{r})$, then for any $s \in \{\pm 1\}^m$, the estimates from k iterations of the iterative algorithm achieve*

$$\lim_{m\to\infty} \frac{1}{m} \sum_{i=1}^{m} \mathbb{P}\Big(s_i \neq \hat{s}_i\big(\{\mathbf{A}_{ij}\}_{(i,j)\in E}\big)\Big) \quad \leq \quad e^{-lq/(2\rho_k^2)} \; . \tag{1}$$

Performance Guarantee

Theorem 2.1. *For fixed $l > 1$ and $r > 1$, assume that m tasks are assigned to $n = ml/r$ workers according to a random (l, r)-regular graph drawn from the configuration model. If the distribution of the worker reliability satisfy $\mu \equiv \mathbb{E}[2\mathbf{p}_j - 1] > 0$ and $q^2 > 1/(\hat{l}\hat{r})$, then for any $s \in \{\pm 1\}^m$, the estimates from k iterations of the iterative algorithm achieve*

$$\lim_{m\to\infty} \frac{1}{m} \sum_{i=1}^{m} \mathbb{P}\Big(s_i \neq \hat{s}_i\big(\{\mathbf{A}_{ij}\}_{(i,j)\in E}\big)\Big) \quad \leq \quad e^{-lq/(2\rho_k^2)} \ . \tag{1}$$

Corollary 2.2. *Under the hypotheses of Theorem 2.1,*

$$\lim_{k\to\infty} \lim_{m\to\infty} \frac{1}{m} \sum_{i=1}^{m} \mathbb{P}\Big(s_i \neq \hat{s}_i\big(\{\mathbf{A}_{ij}\}_{(i,j)\in E}\big)\Big) \quad \leq \quad e^{-lq/(2\rho_\infty^2)} \ . \tag{2}$$

Remarks on the Performance

- This iterative algorithm could converge quickly.
  - computationally efficient as majority voting.

Lemma 2.3. *Under the hypotheses of Theorem 2.1, the total* computational cost *sufficient to achieve the bound in Corollary 2.2 up to any constant factor in the exponent is* $O(ml\log(q/\mu^2)/\log(q^2\hat{l}\hat{r}))$.

Remarks on the Performance

- It's necessary to assume $\mu > 0$
  - Knowing the overall quality of the crowd
  - either most of people make a correct label
  - or most of people make a wrong label (flip the results)

Remarks on the Performance

- Do not require more information on the reliability distribution Pj.
  - EM algorithm is sensitive to initialization.
  - This iterative algorithm does not depend on initialization and could get to converge to the solution.

Remarks on the Performance

- Transition phase at $\hat{l}\hat{r}q^2 = 1$.
  - when $\hat{l}\hat{r}q^2 > 1$, we show that this algorithm is order-optimal and significantly improve majority voting.
  - when $\hat{l}\hat{r}q^2 < 1$, we observe from experiments that the error rate increases with k increases. It's better to select k =1, which essentially become the majority voting.
  - recall that if $\hat{l}\hat{r}q^2 < 1$, ρ_k^2 does not have a limitation.
- Same transition phase observed in EM.

Experimental Evaluation

- Majority Voting
  - decide the result on what the majority of workers agree on.
  - in formula:

$$\hat{s}_i = \mathrm{sign}\left(\sum_{j \in \partial i} A_{ij}\right),$$

Experimental Evaluation

- Oracle Estimator
  - Assume "oracle" know the exact reliability of each worker.

$$\hat{s}_i = \mathrm{sign}\Big(\sum_j \underbrace{W_{ij}}_{reliability} \underbrace{A_{ij}}_{response}\Big)$$

  - where

$$W_{ij} = \log\left(\frac{p_j}{1-p_j}\right)$$

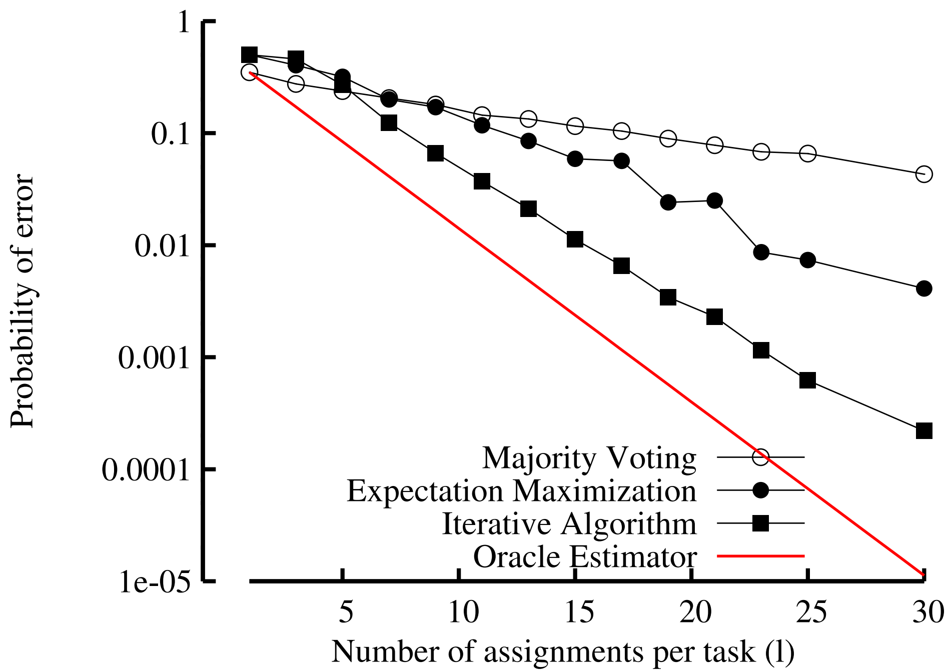
  - oracle fully trust workers with reliability 1, throw out answers from workers with reliability 0.

Experimental Evaluation

- Evaluation Metrics
  - Error rate P_error = $\frac{1}{m}\sum_{i=1}^{m}\mathbb{P}(s_i \neq \hat{s}_i)$.

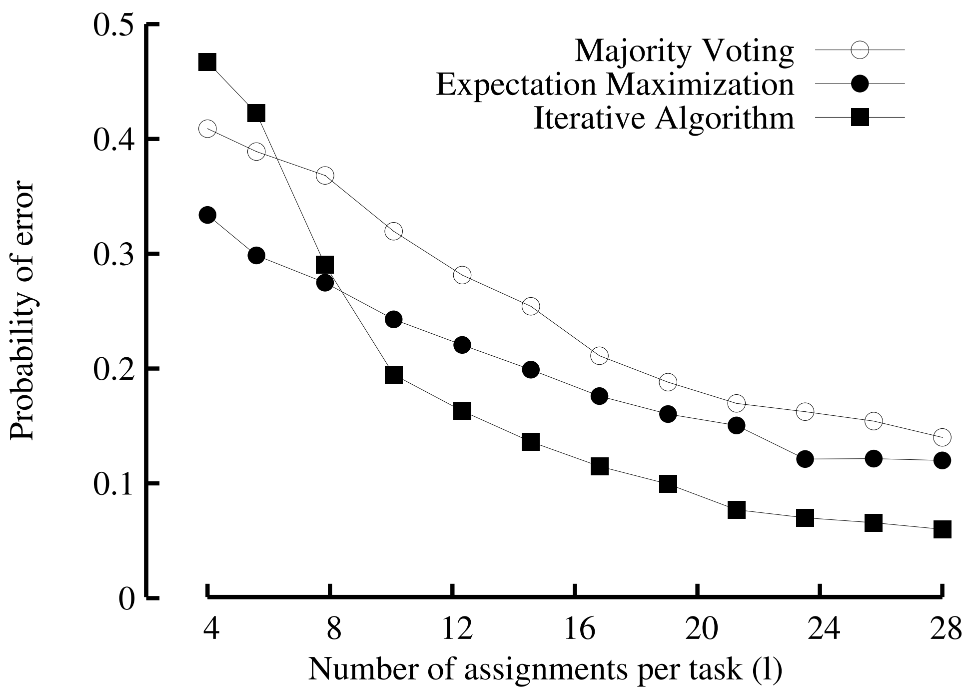
Experimental Evaluation

- Simulation Data (10 iteration, l = r, q = 0.3)
  - transition phase: l = 1 + 1/ 0.3 = 4.33



Experimental Evaluation

- Real crowd Amazon Mechanical Turk
  - color comparison, 50 tasks, 28 workers.
  - ground truth = color space distance
  - q = 0.175, transition phase = 5



Optimality

- One-shot scenario
  - all task assignments are done simultaneously.
  - then an estimation is performed after all the answers are obtained.
- Given a target accuracy $\epsilon \in (0, 1/2)$, how many assignments per task we need to achieve this goal?
- Order-optimal
  - better than majority voting
  - unavoidable

Optimality

- Notations:
  - distribution f is chosen from a set of all distributions on [0,1] which satisfy $\mathbb{E}_f[(2\mathbf{p}_j - 1)^2] = q$. we use $\mathcal{F}(q)$ to donate this set.
  - let $\mathcal{G}(m, l)$ denote the set of all bipartite graphs, including irregular graphs, having m task nodes and ml total edges.

  - Δ_{LB} is minimum cost per task necessary to achieve a target accuracy ϵ using any graph and the best possible algorithm on that graph.

Optimality

- Min. error rate for all possible assignments and all possible inference algorithm: (using oracle estimator)

Lemma 2.4. *The minimax error rate achieved by the best possible graph $G \in \mathcal{G}(m,l)$ using the best possible inference algorithm is at least*

$$\inf_{\text{Algo}, G \in \mathcal{G}(m,l)} \sup_{s, f \in \mathcal{F}(q)} d_m(s, \hat{s}_{G,\text{Algo}}) \geq \frac{1}{2}(1-q)^l ,$$

- when $q \leq C_1$ for some numerical constant $C_1 < 1$.

$$\inf_{\text{Algo}, G \in \mathcal{G}(m,l)} \sup_{s, f \in \mathcal{F}(q)} d_m(s, \hat{s}_{G,\text{Algo}}) \geq \frac{1}{2} e^{-(lq + C_2 l q^2)} ,$$

$$\Delta_{\text{LB}} = \Theta\Big(\frac{1}{q} \log\Big(\frac{1}{\epsilon}\Big)\Big) .$$

Optimality

- Majority voting

Lemma 2.5. *In the regime where where $q \leq C_2 < 1$, there exists a numerical constant C_3 such that*

$$\inf_{G \in \mathcal{G}(m,l)} \sup_{s, f \in \mathcal{F}(q)} d_m(s, \hat{s}_{\mathrm{G,Majority}}) \quad \geq \quad e^{-C_3(lq^2+1)} \ .$$

$$\Delta_{\mathrm{Majority}} = \Omega\Big(\frac{1}{q^2}\log\Big(\frac{1}{\epsilon}\Big)\Big) \ .$$

Optimality

- Iterative inference algorithm with random regular (l,r)-bipartite graph.

For $\hat{l}q \geq C_4$, $\hat{r}q \geq C_5$ and $C_4 C_5 > 1$

$$\lim_{m\to\infty} \sup_{s,f\in\mathcal{F}(q)} d_m(s, \hat{s}_{\mathrm{Iter}}) \quad \leq \quad e^{-C_6 lq} \; .$$

- Then minimum coast per task **sufficient** to achieve target error rate is:

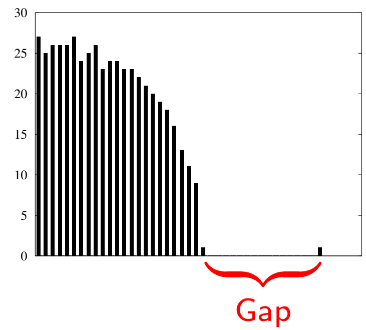
$$\Delta_{\mathrm{Iter}} = \Theta\Big(\frac{1}{q}\log\Big(\frac{1}{\epsilon}\Big)\Big) \; .$$

Agenda

- Introduction
- Crowdsourcing Model
- Proposed Algorithms
- Performance Guarantee and Optimality
- Density Evolution Analysis Technique (Proof of Thm. 2.1)

More Discussion

- Relations to low-rank matrix approximation
  - Gap between first singular value and the second

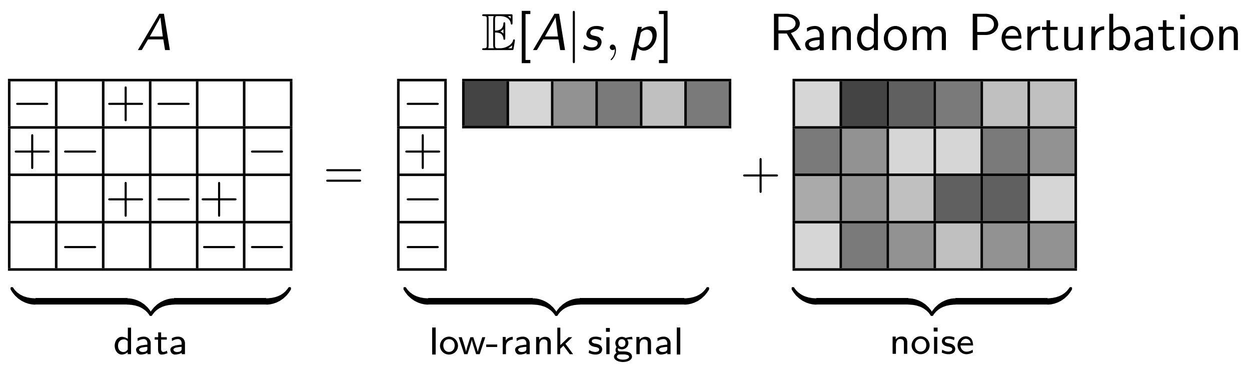


$\rightarrow$ High Signal-to-Noise Ratio

  - Use SVD to approximate the adjacency matrix

More Discussion

- Low-rank matrix approximation



- Power iteration

$$\text{for all } i, \quad u_i = \sum_{j \in \partial i} A_{ij} v_j, \ \text{ and for all } j, \quad v_j = \sum_{i \in \partial j} A_{ij} u_i$$

More Discussion

- Not strong enough bound:

Theorem II.1. *For fixed l and r which are independent of m, assume that m tasks are assigned to $n = ml/r$ workers under the* spammer-hammer *model according to a random (l, r)-regular graph drawn from the configuration model. Then, for any $s \in \{\pm 1\}^m$, with probability $1 - m^{-\Omega(\sqrt{l})}$, the low-rank approximation algorithm achieves*

$$d(s, \hat{s}(\mathbf{A})) \quad \leq \quad \frac{C(\rho)}{lq} \,, \tag{4}$$

where q is the probability that a randomly chosen worker is a hammer and $C(\rho)$ is a constant that only depends on $\rho \equiv l/r$.

source: D.R.Karger, S. Oh, and D. Shah, Budget-optimal crowdsourcing using low-rank matrix approximation, 2011

More Discussion

- Assessing quality of the workers (gold standard units)
  - Using ‘seed gold units’?
    - no help due to order-optimal
  - Using ‘pilot gold unites’?
    - only change the distribution of participating workers.

More Discussion

- Workers with different reliabilities and their prices
  - K classes of workers: 1, ..., k
  - reliability distribution parameter q_k and payment c_k

  - if only one class of workers is used:
    - per-task cost scales as $c_k / q_k \log(1/\epsilon)$
  - if a mix of workers from different classes
    - α_k is the fraction with $\sum_k \alpha_k = 1$
    - optimal per-task cost scales as $(\sum_k \alpha_k c_k)/(\sum_k \alpha_k q_k) \log(1/\epsilon)$
    - implies only select workers with minimal ratio of ck/qk

More Discussion

- What if we don't know about q in selecting the degree?

$$l = \Theta(1/q \log(1/\epsilon))$$

  - incremental design in which at step a the system is designed assuming $q = 2^{-a}$.
  - design two replicas of the task allocation.
  - compare the estimates obtained by these two replicas for all m tasks, if they agree amongst $m(1-2\epsilon)$ tasks, then stop and declare the final answer. Or else, increase a to a+1.

References

- D. R. Karger, S. Oh, D. Shah. Iterative Learning for Reliable Crowdsourcing Systems. NIPS 2011
- D. R. Karger, S. Oh, D. Shah. Budget-Optimal Task Allocation for Reliable Crowdsourcing Systems. CoRR, abs/1110.3564, 2011
- D.R. Karger, S. Oh, D. Shah. Budget-optimal crowdsourcing using low-rank matrix optimizations. Proc. of Allerton Conf. on Commun., Control and Computing, 2011