Max Algorithms in Crowdsourcing Environments

Petros Venetis\textsuperscript{1}  
Kerui Huang\textsuperscript{2}  
Hector Garcia-Molina\textsuperscript{1}  
Neoklis Polyzotis\textsuperscript{2}

\textsuperscript{1}Stanford University  
\textsuperscript{2}UC Santa Cruz

April 20, 2012
Crowdsourcing: Getting Tasks done by People

Why?

- Humans are better than computers in certain tasks
- Human opinions are desired (product and ad design)
Crowdsourcing: Getting Tasks done by People

**Why?**
- Humans are better than computers in certain tasks
- Human opinions are desired (product and ad design)

**Positioning**
- Worker motivation
- Skills required
- Time for tasks
Crowdsourcing: Getting Tasks done by People

Why?

- Humans are better than computers in certain tasks
- Human opinions are desired (product and ad design)

Positioning

- Worker motivation: payment
- Skills required: no qualifications
- Time for tasks: microtasks/seconds
Crowdsourcing

**Issues**

- User Interfaces
- **Algorithms**
- Machine Learning
- Quality Control
- Systems
- Spammer Detection
Max Item Problem: Example

Finding Peak Hours
Max Item Problem: Example

Finding Peak Hours
Max Item Problem: Example

Finding Peak Hours
Crowdsourcing Marketplaces

Example: Amazon’s Mechanical Turk Marketplace

Requester
Crowdsourcing Marketplaces

Example: Amazon’s Mechanical Turk Marketplace

- Requester
- HITs
Crowdsourcing Marketplaces

Example: Amazon’s Mechanical Turk Marketplace

Requester

HITs

Workers
Crowdsourcing Marketplaces

Example: Amazon’s Mechanical Turk Marketplace

[Diagram showing the interaction between Requester, HITs, and Workers]
Crowdsourcing Marketplaces

Example: Amazon’s Mechanical Turk Marketplace

Requester → HITs → Workers
Crowdsourcing Algorithms

Notion of steps

Step 1

Step 2

Step 3

HITs
Items have inherent quality values

Max item $e^* \in E$:

$e \leq e^* \forall e \in E \setminus \{e^*\}$
Max Algorithms

Model

\[ \mathcal{E} \]

- Items have inherent quality values
Max Algorithms

Model

\[ E \]

- Items have inherent quality values
- Max item \( e^* \in E \):
  \[ e \leq e^* \ \forall e \in E \setminus \{e^*\} \]
Worker Tasks

HITs Used: Comparisons

\[ r = 3 \]

\[ s = 4 \]
Worker Tasks

HITs Used: Comparisons

$r = 3$

$s = 4$

$C_0/C_1/CC$

/BE

/BD

$8/21$
Structured Algorithms

How to break up the problem to retrieve max item:

- Bubble
- Tournament
Crowdsourced Max Algorithms

**Structured Algorithms**

How to break up the problem to retrieve max item:
- Bubble
- Tournament

**Unstructured Algorithms (not here)**

Which comparison to perform next?
- Which item is the max?
Max Algorithms: Bubble

Example

$e_1$

$e_2$

$e_3$

$e_4$

$e_5$

$e_6$
Max Algorithms: Bubble

Example

\[ s_1 = 3 \]
\[ r_1 = 5 \]
Max Algorithms: Bubble

Example

\begin{itemize}
  \item $s_1 = 3$
  \item $r_1 = 5$
\end{itemize}
Max Algorithms: Bubble

Example

\[ s_1 = 3 \]
\[ r_1 = 5 \]

\[ s_2 = 2 \]
\[ r_2 = 5 \]
Max Algorithms: Bubble

Example

\[ s_1 = 3 \]
\[ r_1 = 5 \]

\[ s_2 = 2 \]
\[ r_2 = 5 \]
Max Algorithms: Bubble

Example

$s_1 = 3$
$r_1 = 5$

$s_2 = 2$
$r_2 = 5$

$s_3 = 3$
$r_3 = 1$
Max Algorithms: Bubble

Example

\[ s_1 = 3 \]
\[ r_1 = 5 \]

\[ s_2 = 2 \]
\[ r_2 = 5 \]

\[ s_3 = 3 \]
\[ r_3 = 1 \]
Max Algorithms: Bubble

Example

\[ s_1 = 3 \]
\[ r_1 = 5 \]

Steps = 3
Questions = 5 + 5 + 1 = 11

\[ s_2 = 2 \]
\[ r_2 = 5 \]

\[ s_3 = 3 \]
\[ r_3 = 1 \]
Max Algorithms: Tournament

Example

<table>
<thead>
<tr>
<th>e_1</th>
</tr>
</thead>
<tbody>
<tr>
<td>e_2</td>
</tr>
<tr>
<td>e_3</td>
</tr>
<tr>
<td>e_4</td>
</tr>
<tr>
<td>e_5</td>
</tr>
<tr>
<td>e_6</td>
</tr>
</tbody>
</table>
Max Algorithms: Tournament

Example

\[ s_1 = 2 \]
\[ r_1 = 5 \]

\[ e_1 \]
\[ e_2 \]
\[ e_3 \]
\[ e_4 \]
\[ e_5 \]
\[ e_6 \]
Max Algorithms: Tournament

**Example**

\[ s_1 = 2 \]
\[ r_1 = 5 \]

\[
\begin{array}{c}
\text{e}_1 \\
\text{e}_2 \\
\hline
\text{e}_3 \\
\text{e}_4 \\
\hline
\text{e}_5 \\
\text{e}_6
\end{array}
\]

\[ \Rightarrow \text{e}_1 \]
\[ \Rightarrow \text{e}_4 \]
\[ \Rightarrow \text{e}_6 \]
Max Algorithms: Tournament

Example

\[ s_1 = 2 \]
\[ r_1 = 5 \]
\[ e_1 \]
\[ e_2 \]
\[ e_3 \]
\[ e_4 \]
\[ e_5 \]
\[ e_6 \]

\[ s_2 = 3 \]
\[ r_2 = 3 \]
\[ e_1 \]
\[ e_4 \]
\[ e_6 \]
Max Algorithms: Tournament

Example

\begin{align*}
  s_1 &= 2 \\
  r_1 &= 5 \\
  e_1 &\rightarrow e_2 \\
  e_3 &\rightarrow e_4 \\
  e_5 &\rightarrow e_6 \\
  s_2 &= 3 \\
  r_2 &= 3 \\
  e_1 &\rightarrow e_4 \\
  e_4 &\rightarrow e_6 \\
  e_6 &\rightarrow e_1
\end{align*}
Max Algorithms: Tournament

Example

\[ s_1 = 2 \]
\[ r_1 = 5 \]

\[ e_1 \rightarrow e_2 \rightarrow e_3 \rightarrow e_4 \rightarrow e_5 \rightarrow e_6 \]

\[ s_2 = 3 \]
\[ r_2 = 3 \]

Steps = 2
Questions = \[ 3 \cdot 5 + 1 \cdot 3 = 18 \]
How to select \( \{r_i\} \) and \( \{s_i\} \)

**Problem Statement**

Maximize \( A = A(\{r_i\}, \{s_i\}) \) subject to:

- \( \Pr[A \text{ returns max item from } \mathcal{E}] \)
- \( \text{Cost}(A, \mathcal{E}) \leq B \)
- \( \text{Steps}(A, \mathcal{E}) \leq T \)

Tuning Strategies (based on hill climbing):

- Constant \( r_i, s_i \)
- Varying \( r_i, \) constant \( s_i \)
- Varying \( r_i, s_i \)
How to select \{r_i\} and \{s_i\}

Problem Statement

maximize \[ A = A(\{r_i\}, \{s_i\}) \]

subject to

\[ \Pr[A \text{ returns max item from } \mathcal{E}] \]

\[ \text{Cost}(A, \mathcal{E}) \leq B \]

\[ \text{Steps}(A, \mathcal{E}) \leq T \]

Tuning Strategies (based on hill climbing)

- Constant \( r_i, s_i \)
- Varying \( r_i \), constant \( s_i \)
- Varying \( r_i, s_i \)
Models Considered

- Comparison Input $= \{e_1, e_2, \ldots, e_s\}$
- $e_s < \ldots < e_2 < e_1$
- $p_i$: probability $e_i$ is returned by worker
Models Considered

- Comparison Input = \{e_1, e_2, \ldots, e_s\}
- \(e_s < \ldots < e_2 < e_1\)
- \(p_i\): probability \(e_i\) is returned by worker

Worker Error Models

<table>
<thead>
<tr>
<th>Model</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constant</td>
<td>(p_1 = p, \ p_2 = p_3 = \ldots)</td>
</tr>
<tr>
<td>Linear</td>
<td>(p_1) decreases on (s)</td>
</tr>
<tr>
<td>Order-based</td>
<td>(p_1 &gt; p_2 &gt; \ldots &gt; p_s)</td>
</tr>
<tr>
<td>Distance-based</td>
<td>(p_i)’s depend on value differences of items</td>
</tr>
</tbody>
</table>
Models Considered

- **Comparison Input** = \{e_1, e_2, \ldots, e_s\}
- e_s < \ldots < e_2 < e_1
- \(p_i\): probability \(e_i\) is returned by worker

### Worker Error Models

<table>
<thead>
<tr>
<th>Model</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constant</td>
<td>(p_1 = p, \ p_2 = p_3 = \ldots)</td>
</tr>
<tr>
<td>Linear</td>
<td>(p_1) decreases on (s)</td>
</tr>
<tr>
<td>Order-based</td>
<td>(p_1 &gt; p_2 &gt; \ldots &gt; p_s)</td>
</tr>
<tr>
<td>Distance-based</td>
<td>(p_i)'s depend on value differences of items</td>
</tr>
</tbody>
</table>

### Worker Compensation Models

<table>
<thead>
<tr>
<th>Model</th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constant</td>
<td>(c)</td>
</tr>
<tr>
<td>Linear</td>
<td>(c + \lambda \times s)</td>
</tr>
</tbody>
</table>
### Why Not Analysis?

- Analysis only for simple models and still is expensive
Experiments

Why Not Analysis?

- Analysis only for simple models and still is expensive

Single HIT Accuracy \((s, r; \vec{p}) = \)

\[
\sum_{l=1}^{s} \frac{1}{l} \cdot \sum_{n=1}^{r} \sum_{L \in \mathcal{L}} \sum_{0 \leq k_i \leq n-1, i \in \bar{L}} \sum_{\sum_{i \in \bar{L}} k_i + l \cdot n = r} \frac{r!}{(n!)^l \cdot \prod_{j \in \bar{L}} k_j !} \cdot \prod_{z \in L} p_z^n \cdot \prod_{w \in \bar{L}} p_w^{k_w}
\]

Simulations

Used linear error and compensation model

100,000 simulations per data point
Experiments

Why Not Analysis?

- Analysis only for simple models and still is expensive

Single HIT Accuracy \( s, r; \bar{p} \) =

\[
\sum_{l=1}^{s} \frac{1}{l} \cdot \sum_{n=1}^{r} \sum_{L \in \mathcal{L}} \sum_{0 \leq k_{i} \leq n-1, i \in \bar{L}} \sum_{i \in \bar{L}} k_{i} + l \cdot n = r \left[ \frac{r!}{(n!)^{l} \cdot \prod_{j \in \bar{L}} k_{j}!} \cdot \prod_{z \in \mathcal{L}} p_{z}^{n} \cdot \prod_{w \in \bar{L}} p_{w}^{k_{w}} \right]
\]

Simulations

- Used linear error and compensation model
- 100,000 simulations per data point
It pays off to vary $r_i, s_i$
Tournament is better

![Graph showing the probability of max item as a function of budget. The graph compares Tournament and Bubble strategies.](image-url)
It pays off to have more responses towards the end

![Bar chart showing tournament outcomes with different values of B (1500, 4000, 5500)].
MTurk Experiment

Dataset

80 and 100 dots
MTurk Experiment

Dataset

80 and 100 dots

Setting

- Learned distance-based model \((s_i = 4)\)
- Images with 5, 10, \ldots, 320 dots
- Tournaments

- \(B\) enough for 105 comparisons
- 200 runs
- $0.01 per comparison
### MTurk Experiment (cont’d)

<table>
<thead>
<tr>
<th>Constant $r_i$</th>
<th>Varying $r_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$r_1 = r_2 = r_3 = 5$</td>
<td>$r_1 = 3$</td>
</tr>
<tr>
<td></td>
<td>$r_2 = 5$</td>
</tr>
<tr>
<td></td>
<td>$r_3 = 37$</td>
</tr>
</tbody>
</table>

Observations:
- Experiments match predictions
- Varying repetitions ($r_i$) improves results significantly
MTurk Experiment (cont’d)

Constant $r_i$
- $r_1 = r_2 = r_3 = 5$
Predicted Pr[max item]: 0.67

Varying $r_i$
- $r_1 = 3$
- $r_2 = 5$
- $r_3 = 37$
Predicted Pr[max item]: 0.84
MTurk Experiment (cont’d)

<table>
<thead>
<tr>
<th>Constant $r_i$</th>
<th>Varying $r_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$r_1 = r_2 = r_3 = 5$</td>
<td>$r_1 = 3$</td>
</tr>
<tr>
<td>Predicted $\Pr[\text{max item}]$: 0.67</td>
<td>$r_2 = 5$</td>
</tr>
<tr>
<td>Measured $\Pr[\text{max item}]$: 0.69</td>
<td>$r_3 = 37$</td>
</tr>
<tr>
<td></td>
<td>Predicted $\Pr[\text{max item}]$: 0.84</td>
</tr>
<tr>
<td></td>
<td>Measured $\Pr[\text{max item}]$: 0.80</td>
</tr>
</tbody>
</table>
MTurk Experiment (cont’d)

Constant $r_i$
- $r_1 = r_2 = r_3 = 5$

Predicted Pr[max item]: 0.67
Measured Pr[max item]: 0.69

Varying $r_i$
- $r_1 = 3$
- $r_2 = 5$
- $r_3 = 37$

Predicted Pr[max item]: 0.84
Measured Pr[max item]: 0.80

Observations
- Experiments match predictions
- Varying repetitions ($r_i$) improves results significantly
Experimental

- Algorithms are robust
- Repetitive algorithms not helpful
- Relaxing step bound increases accuracy
- Finding the top-1 item is usually hard, but top-$k$% is easy

Analysis

- How to analyze tournament and bubble algorithms (for some models)
Conclusions

Summary

- It pays off to vary the size of a task \((s_i)\)
- It pays off to optimize the number of repetitions \((r_i)\)
- Tournament performs significantly better than bubble
- Tuning tournaments improves results in practice
Conclusions

Summary

- It pays off to vary the size of a task \((s_i)\)
- It pays off to optimize the number of repetitions \((r_i)\)
- Tournament performs significantly better than bubble
- Tuning tournaments improves results in practice

Current Work

- Spammer detection and appropriate actions
- Dynamic adjustments to account for comparison difficulty