Quality Control for Comparison Microtasks

Petros Venetis    Hector Garcia-Molina

Stanford University

August 12, 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)

Our work
- 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)

Our work

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

Issues

- User Interfaces
- Machine Learning
- Algorithms
- Quality Control
- Systems
- Spammer Detection

Applications

- Max item retrieval (example next)
- Sorting (get restaurants sorted by rating)
- Top-k (retrieve 10 best LinkedIn profiles for a job)
Crowdsourcing

Issues

User Interfaces

Machine Learning

Quality Control

Systems

Spammer Detection

Algorithms
## Crowdsourcing

### Issues
- User Interfaces
- Machine Learning
- **Algorithms**
- Quality Control
- Systems
- Spammer Detection

### Applications
- Max item retrieval (example next)
- Sorting (get restaurants sorted by rating)
- Top-\( k \) (retrieve 10 best LinkedIn profiles for a job)
Example: Tournament Max Algorithm

Tournament Algorithm

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

Tournament Algorithm

\[ r_1 = 5 \]

\[ e_1 \neq e_2 \neq e_3 \neq e_4 \neq e_5 \neq e_6 \]
Example: Tournament Max Algorithm

Tournament Algorithm

\[ r_1 = 5 \]

\[ e_1 \rightarrow e_1 \]

\[ e_3 \rightarrow e_4 \]

\[ e_5 \rightarrow e_6 \]
Example: Tournament Max Algorithm

Tournament Algorithm

$r_1 = 5$

$e_1$

$e_2$

$e_3$

$e_4$

$r_2 = 3$

$e_1$

$e_4$

$e_6$
Example: Tournament Max Algorithm

Tournament Algorithm

$r_1 = 5$

$e_1 \neq e_2 \neq e_3 \neq e_4 \neq e_5 \neq e_6$

$r_2 = 3$

$e_1 \neq e_4 \neq e_6$

$e_1$
Example: Tournament Max Algorithm (cont’d)

Example: Finding Peak Hours
Example: Tournament Max Algorithm (cont’d)

Example: Finding Peak Hours

[Images of a library with shelves and people]
Example: Finding Peak Hours
Example: Tournament Max Algorithm (cont’d)

Comparisons

$r = 3$

HIT
Example: Tournament Max Algorithm (cont’d)

Comparisons

$r = 3$

HIT

1

2
Quality Control for Comparison Microtasks

Issues

User Interfaces

Algorithms

Systems

Machine Learning

Quality Control

Spammer Detection
Quality Control for Comparison Microtasks

### Issues

<table>
<thead>
<tr>
<th>User Interfaces</th>
<th>Machine Learning</th>
<th>Quality Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>Algorithms</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Systems</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Spammer Detection</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Setting: Experimental

- Amazon’s Mechanical Turk
- Comparisons of various difficulties
- Dataset with ground truth
Quality Control Techniques

Many!

- Masking: Asking multiple workers to perform each task
- Detection: Ignore bad worker answers
- Evicting bad workers
- Retaining good workers
- Different pay rates according to worker quality
- Train before tasks
- ...
Quality Control Techniques

Many!

- **Masking**: Asking multiple workers to perform each task
- **Detection**: Ignore bad worker answers
- Evicting bad workers
- Retaining good workers
- Different pay rates according to worker quality
- Train before tasks
- ...

...
Which image has more dots?

Dataset

\( q(e_1) = 90 \)
\( q(e_2) = 100 \)
Dataset

Which image has more dots?

$q(e_1) = 90$

$q(e_2) = 100$
Experiments

Find image with most dots

- 1, 2, . . . , 1000 dots per image
- $0.01$ per HIT
- 4 comparisons per HIT
- 4 images per comparison

Statistics

- $\sim 28,500$ distinct comparisons
- $r \in \{1, 2, 3, 4, 5\}$
- $\sim 54,000$ worker responses
- $\sim 1,100$ distinct worker IDs

For good coverage: No more than 50 HITs per hour
Experiments

Find image with most dots

- 1, 2, . . . , 1000 dots per image
- $0.01 per HIT
- 4 comparisons per HIT
- 4 images per comparison

Statistics

- \( \sim 28,500 \) distinct comparisons
- \( r \in \{1, 2, 3, 4, 5\} \)
- \( \sim 54,000 \) worker responses
- \( \sim 1,100 \) distinct worker IDs
- For good coverage: No more than 50 HITs per hour
Comparison Difficulty

**Definition**

When comparing items in \( S = \{e_1, e_2, \ldots, e_s\} \), difficulty is

\[
diff(S) = \frac{q_2(S)}{q_1(S)}
\]

**Characteristics**

- Values in \([0, 1]\)
- Takes into account only top-2 values
Comparison Difficulty Effectiveness

Very effective Metric

Similar correctness for different $q_1(h)$ but the same $\text{diff}(S)$
Why is Difficulty important?

Tournament Algorithms

\[
\begin{align*}
    r_1 &= 5 \\
    &e_1 \\
    &e_2 \\
    &e_3 \\
    &e_4 \\
    &e_5 \\
    &e_6 \\
    r_2 &= 3 \\
    &e_1 \\
    &e_4 \\
    &e_6 \\
\end{align*}
\]

Easier comparisons initially

Harder towards the end

We need to take into account various difficulty values
Why is Difficulty important?

Difficulty in Tournament Algorithms

- Easier comparisons initially
- Harder towards the end
Why is Difficulty important?

Difficulty in Tournament Algorithms

- Easier comparisons initially
- Harder towards the end
- We need to take into account various difficulty values
Effect on Comparison Accuracy

Accuracy increases as we ask more workers. It reaches a plateau after a while. It is higher for easy comparisons.
Masking: Choosing the Plurality Vote

Effect on Comparison Accuracy

- Accuracy increases as we ask more workers
- It reaches a plateau after a while
- It is higher for easy comparisons

![Graph showing accuracy increase with number of workers](image)
Can we do better than Masking?

### Detection

<table>
<thead>
<tr>
<th>Worker</th>
<th>{e_1, e_2}</th>
<th>{e_3, e_4}</th>
<th>{e_5, e_6}</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>e_1</td>
<td>e_3</td>
<td>e_5</td>
</tr>
<tr>
<td>B</td>
<td>e_1</td>
<td>e_4</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>e_1</td>
<td></td>
<td>e_5</td>
</tr>
<tr>
<td>D</td>
<td></td>
<td>e_4</td>
<td>e_6</td>
</tr>
<tr>
<td>Plurality</td>
<td>e_1</td>
<td>e_4</td>
<td>e_5</td>
</tr>
<tr>
<td>Max</td>
<td>e_1</td>
<td>e_3</td>
<td>e_5</td>
</tr>
</tbody>
</table>
Can we do better than Masking?

### Detection

<table>
<thead>
<tr>
<th>Worker</th>
<th>{e_1, e_2}</th>
<th>{e_3, e_4}</th>
<th>{e_5, e_6}</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>e_1</td>
<td>e_3</td>
<td>e_5</td>
</tr>
<tr>
<td>B</td>
<td>e_1</td>
<td>e_4</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>e_1</td>
<td></td>
<td>e_5</td>
</tr>
<tr>
<td>D</td>
<td></td>
<td>e_4</td>
<td>e_6</td>
</tr>
<tr>
<td>Plurality</td>
<td>e_1</td>
<td>e_4</td>
<td>e_5</td>
</tr>
<tr>
<td>Max</td>
<td>e_1</td>
<td>e_3</td>
<td>e_5</td>
</tr>
</tbody>
</table>

### Scores Considered

- **Gold Standard** \( s_{GS}(A) = 1 \)
- **Plurality Agreement** \( s_P(A) = \frac{2}{3} \)
- **Work time** \( s_T \)
How good are these Scores?

Very!

- For worker with at least 10 comparisons done
- Actual score = fraction of correct answers
- Very high correlation!
Is Detection helpful?

It increases Accuracy for each Assignment

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Worker \{e_1, e_2\} \{e_3, e_4\} \{e_5, e_6\}

A e_1 e_3 e_5
B e_1 e_4 
C e_1 e_5 
D e_4 e_6 

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

17 / 20
Is Detection helpful?

It increases Accuracy for each Assignment

\[
\begin{align*}
|\mathcal{A}'| & = 20,000 \\
\text{Worker} & \{e_1, e_2\} \quad \{e_3, e_4\} \quad \{e_5, e_6\} \\
A & e_1 \quad e_3 \quad e_5 \\
B & e_1 \quad e_4 \\
C & e_1 \quad e_5 \\
D & e_4 \quad e_6 \\
\vdots & \vdots \quad \vdots \quad \vdots \\
\end{align*}
\]
Is Detection helpful?

It increases Accuracy for each Assignment

| \(|A'|\) | 10,000 | 20,000 |
|---|---|---|
| \(|A'|\) | \(8,000\) |

<table>
<thead>
<tr>
<th>Worker</th>
<th>({e_1, e_2})</th>
<th>({e_3, e_4})</th>
<th>({e_5, e_6})</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>(e_1)</td>
<td>(e_3)</td>
<td>(e_5)</td>
</tr>
<tr>
<td>B</td>
<td>(e_1)</td>
<td>(e_4)</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>(e_1)</td>
<td></td>
<td>(e_5)</td>
</tr>
<tr>
<td>D</td>
<td></td>
<td>(e_4)</td>
<td>(e_6)</td>
</tr>
<tr>
<td>(\ldots)</td>
<td>(\ldots)</td>
<td>(\ldots)</td>
<td>(\ldots)</td>
</tr>
</tbody>
</table>

\(|A'| = 8,000\)
Is Detection helpful? (cont’d)

It increases Accuracy for each Comparison

<table>
<thead>
<tr>
<th>Worker</th>
<th>{e_1, e_2}</th>
<th>{e_3, e_4}</th>
<th>{e_5, e_6}</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>e_1</td>
<td>e_3</td>
<td>e_5</td>
</tr>
<tr>
<td>B</td>
<td>e_1</td>
<td>e_4</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>e_1</td>
<td></td>
<td>e_5</td>
</tr>
<tr>
<td>D</td>
<td></td>
<td>e_4</td>
<td>e_6</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>Plurality</td>
<td>e_1</td>
<td>e_4</td>
<td>e_5</td>
</tr>
</tbody>
</table>
Is Detection helpful? (cont’d)

It increases Accuracy for each Comparison

![Graph showing accuracy over different comparison sets.](image)

<table>
<thead>
<tr>
<th>Worker</th>
<th>{e_1, e_2}</th>
<th>{e_3, e_4}</th>
<th>{e_5, e_6}</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>e_1</td>
<td>e_3</td>
<td>e_5</td>
</tr>
<tr>
<td>B</td>
<td>e_1</td>
<td>e_4</td>
<td>e_5</td>
</tr>
<tr>
<td>C</td>
<td>e_1</td>
<td>e_4</td>
<td>e_5</td>
</tr>
<tr>
<td>D</td>
<td>e_4</td>
<td>e_6</td>
<td></td>
</tr>
<tr>
<td>\ldots</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Plurality</td>
<td>e_1</td>
<td>e_4</td>
<td>e_5</td>
</tr>
</tbody>
</table>
But at what cost?

Cost per benefit study

For a set of comparisons:
- Benefit = # correct plurality responses after detection
- Cost = # questions posted

Answer: High

![Graph showing cost per benefit study](image)
Conclusions

Summary

- Microtask difficulty has to be considered in crowdsourced algorithms.
- We can assess a worker’s quality accurately.
- After detecting bad workers, we can improve comparison accuracy.
- The cost/benefit is minimum without detection.

Contact
venetis@cs.stanford.edu
Conclusions

Summary
- Microtask difficulty has to be considered in crowdsourced algorithms
- We can assess a worker’s quality accurately
- After detecting bad workers, we can improve comparison accuracy
- The cost/benefit is minimum without detection.

Current Work
- Building worker models that will match experimental data
- Dynamic adjustments to account for comparison difficulty in crowdsourced algorithms
Conclusions

Summary

- Microtask difficulty has to be considered in crowdsourced algorithms
- We can assess a worker’s quality accurately
- After detecting bad workers, we can improve comparison accuracy
- The cost/benefit is minimum without detection.

Current Work

- Building worker models that will match experimental data
- Dynamic adjustments to account for comparison difficulty in crowdsourced algorithms

Contact

venetis@cs.stanford.edu