Quality Control for Comparison Microtasks

Petros Venetis Hector Garcia-Molina

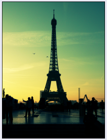
Stanford University

August 12, 2012

Crowdsourcing: Getting Tasks done by People

Why?

- Humans are better than computers in certain tasks

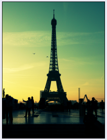


- Human opinions are desired (product and ad design)

Crowdsourcing: Getting Tasks done by People

Why?

- Humans are better than computers in certain tasks



- Human opinions are desired (product and ad design)

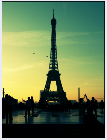
Our work

- 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

Crowdsourcing

Issues

User Interfaces

Machine Learning

Algorithms

Quality Control

Systems

Spammer Detection

Crowdsourcing

Issues

User Interfaces

Machine Learning

Algorithms

Quality Control

Systems

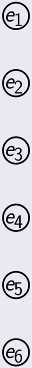
Spammer Detection

Applications

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

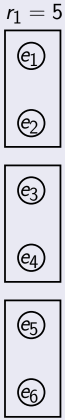
Example: Tournament Max Algorithm

Tournament Algorithm



Example: Tournament Max Algorithm

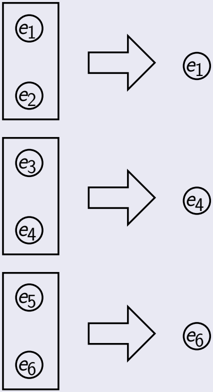
Tournament Algorithm



Example: Tournament Max Algorithm

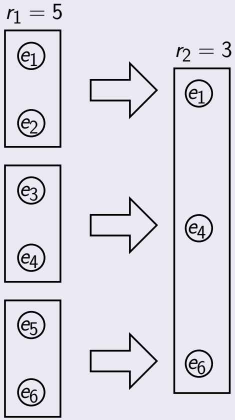
Tournament Algorithm

$r_1 = 5$



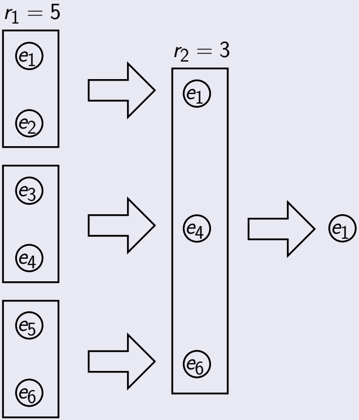
Example: Tournament Max Algorithm

Tournament Algorithm



Example: Tournament Max Algorithm

Tournament Algorithm



Example: Tournament Max Algorithm (cont'd)

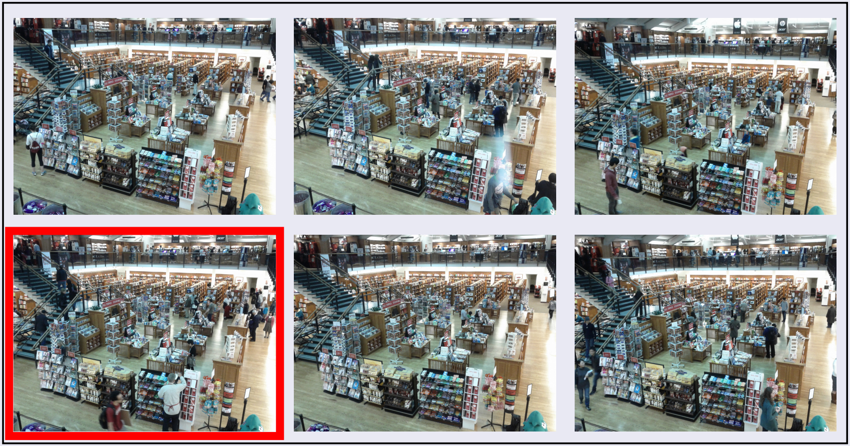
Example: Finding Peak Hours

Example: Tournament Max Algorithm (cont'd)

Example: Finding Peak Hours

Example: Tournament Max Algorithm (cont'd)

Example: Finding Peak Hours



Example: Tournament Max Algorithm (cont'd)

Comparisons

Example: Tournament Max Algorithm (cont'd)

Comparisons

Quality Control for Comparison Microtasks

Issues

User Interfaces

Machine Learning

Algorithms

Quality Control

Systems

Spammer Detection

Quality Control for Comparison Microtasks

Issues

User Interfaces

Machine Learning

Algorithms

Quality Control

Systems

Spammer Detection

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
- . . .

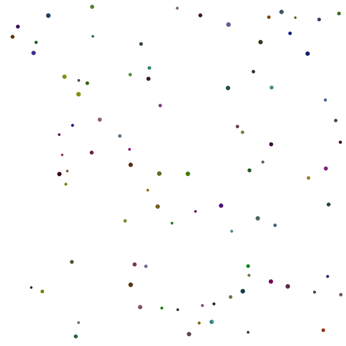
Quality Control Techniques

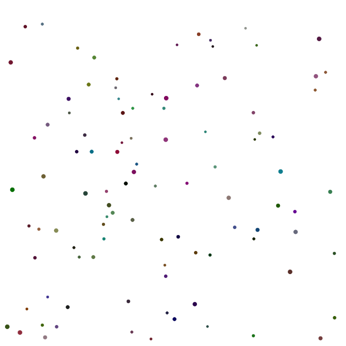
Many!

- 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
- ...

Dataset

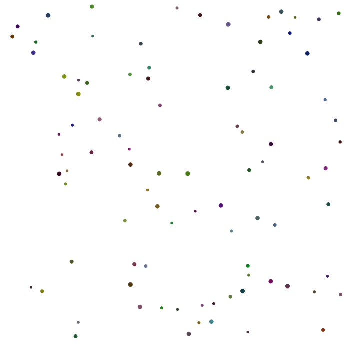
Which image has more dots?



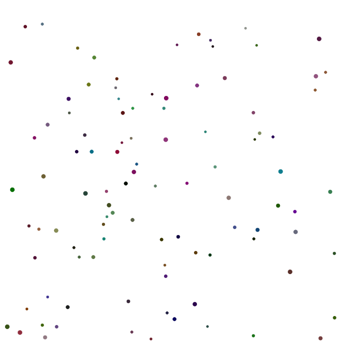


Dataset

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

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

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

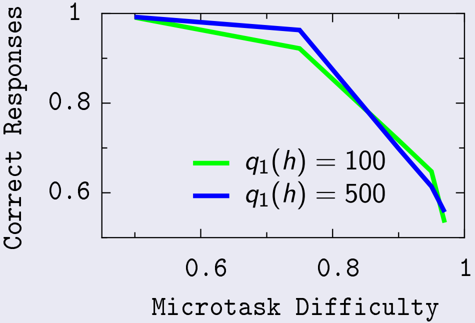
$$\mathrm{diff}(S) = \frac{q_2(S)}{q_1(S)}$$

Characteristics

- Values in $[0, 1]$
- Takes into account only top-2 values

Comparison Difficulty Effectiveness

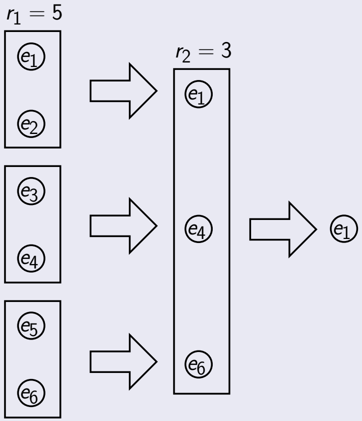
Very effective Metric



- Similar correctness for different $q_1(h)$ but the same $\mathrm{diff}(S)$

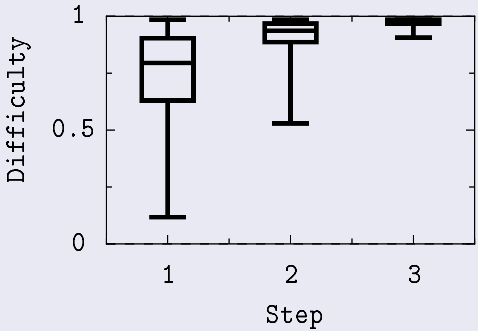
Why is Difficulty important?

Tournament Algorithms



Why is Difficulty important?

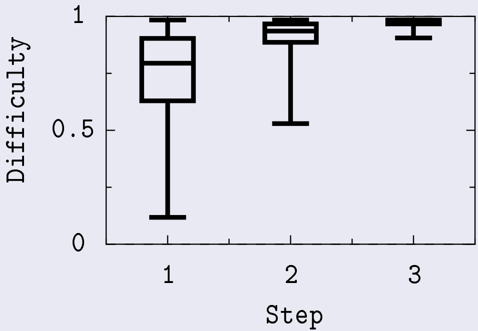
Difficulty in Tournament Algorithms



- Easier comparisons initially
- Harder towards the end

Why is Difficulty important?

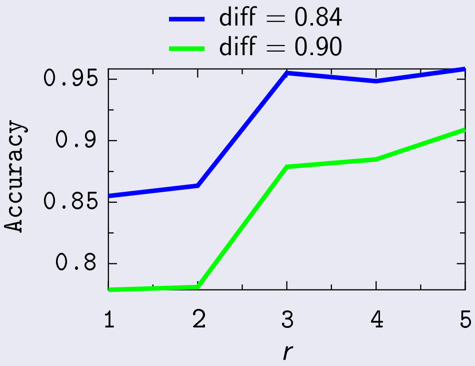
Difficulty in Tournament Algorithms



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

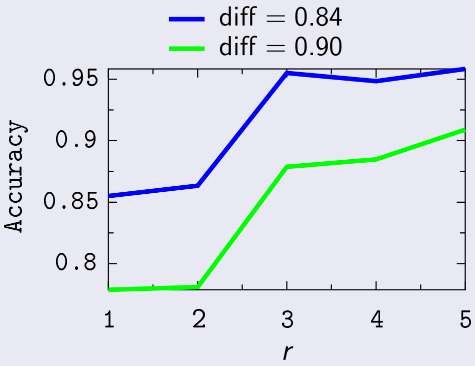
Masking: Choosing the Plurality Vote

Effect on Comparison Accuracy



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

Can we do better than Masking?

Detection

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
Plurality	e_1	e_4	e_5
Max	e_1	e_3	e_5

Can we do better than Masking?

Detection

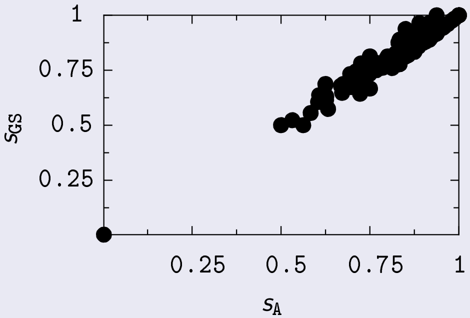
Worker	$\{e_1, e_2\}$	$\{e_3, e_4\}$	$\{e_5, e_6\}$
A	e_1	e_3	e_5
B	e_1	e_4	
C	e_1		e_5
D		e_4	e_6
Plurality	e_1	e_4	e_5
Max	e_1	e_3	e_5

Scores Considered

- Gold Standard $s_{\mathrm{GS}}(\mathrm{A}) = 1$
- Plurality Agreement $s_{\mathrm{P}}(\mathrm{A}) = \frac{2}{3}$
- Work time s_{T}

How good are these Scores?

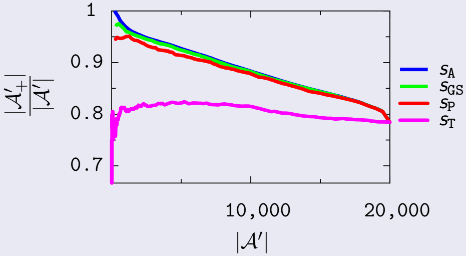
Very!



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

Is Detection helpful?

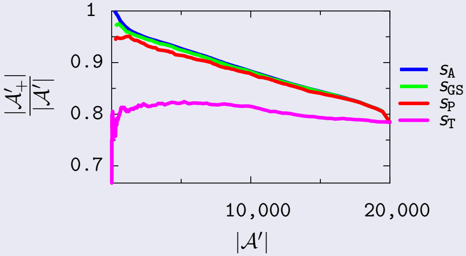
It increases Accuracy for each Assignment



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
⋮	⋮	⋮	⋮

Is Detection helpful?

It increases Accuracy for each Assignment



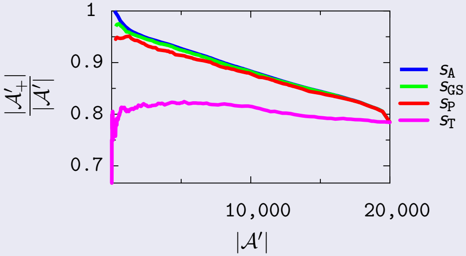
Worker	$\{e_1, e_2\}$	$\{e_3, e_4\}$	$\{e_5, e_6\}$
A	e_1	e_3	e_5
B	e_1	e_4	
C	e_1		e_5
D		e_4	e_6
⋮	⋮	⋮	⋮

$|\mathcal{A}'| = 20{,}000$

All responses

Is Detection helpful?

It increases Accuracy for each Assignment



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~~	$\cancel{e_1}$		$\cancel{e_5}$
~~D~~		$\cancel{e_4}$	$\cancel{e_6}$
⋮	⋮	⋮	⋮

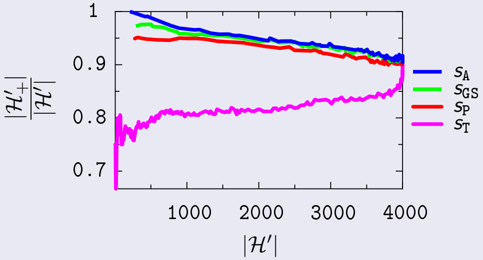
$$|\mathcal{A}'| = 8{,}000$$

40% responses

All responses

Is Detection helpful? (cont'd)

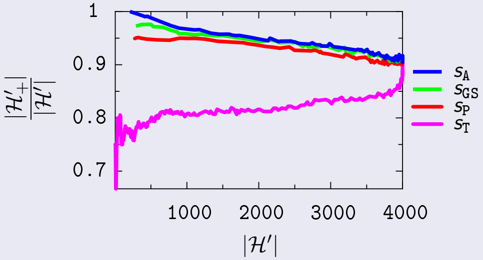
It increases Accuracy for each Comparison



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
⋮	⋮	⋮	⋮
Plurality	e_1	e_4	e_5

Is Detection helpful? (cont'd)

It increases Accuracy for each Comparison



Worker	$\{e_1, e_2\}$	$\{e_3, e_4\}$	$\{e_5, e_6\}$
~~A~~	$\cancel{e_1}$	$\cancel{e_3}$	$\cancel{e_5}$
~~B~~	$\cancel{e_1}$	$\cancel{e_4}$	
~~C~~	$\cancel{e_1}$		$\cancel{e_5}$
D		e_4	e_6
$\vdots$	$\vdots$	$\vdots$	$\vdots$
Plurality	$\cancel{e_1}$	e_4	e_5

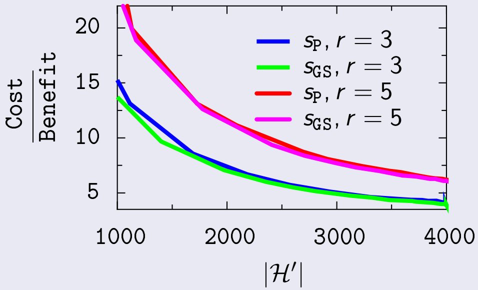
But at what cost?

Cost per benefit study

For a set of comparisons:

- Benefit = # correct plurality responses after detection
- Cost = # questions posted

Answer: High



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.

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`