IUI 2018

Two Tools are Better Than One: Tool Diversity as a Means of Improving Aggregate Crowd Performance

Jeann Y. Song, Raymond Fok, Alan Lundgard, Fan Yang, Juho Kim, Walter S. Lasecki
Crowdsourcing Platforms

- clickworker
- microtask
- upwork
- spare 5
- Amazon Mechanical Turk
- Wikipedia
- Playment
- Mighty Ai
- appen
- CrowdFlower
- Lab in the Wild
Crowdsourcing for Human Computation

https://playment.io/

https://www.crowdguru.de/en/
Crowdsourcing Strategy: Microtasking
Crowdsourcing Strategy: Aggregation

Task

Divide

Microtasks

Aggregate multiple answers

× 2
× 2
× 2
× 2
× 2
Crowdsourcing Strategy: Using Single Tool

Task

Divide

Microtasks
Aggregate multiple answers
Same tool or interface

× 2
× 2
× 2
× 2
Problem with using a single tool:

Systematic bias can be accumulated, resulting in inaccurate aggregated result.
Q. What is Systematic Bias?

A. Reliable, but not valid performance

1. Reliable, not valid
2. Not Reliable, But Valid
3. Not Reliable, not Valid
4. Reliable, Valid
Example of Systematic (Error) Bias

Tool 1: Opensurfaces (TOG 2013)

Tool 2: Click’n’Cut (CrowdMM 2014)
Example of Systematic (Error) Bias

**Tool 1: Opensurfaces (TOG 2013)**


**Tool 2: Click’n’Cut (CrowdMM 2014)**

Example of Systematic (Error) Bias

Tool 1: Opensurfaces (TOG 2013)

Tool 2: Click’n’Cut (CrowdMM 2014)
Proposed Approach:

Use tool diversity as a means of improving aggregate crowd performance
What is **Tool Diversity**?

A property that measures how **different** tools can be built in terms of their induced biases.
Ensemble learning constructs a combination of two alternative hypotheses $h_1$ and $h_2$ with proper weights ($w_1$ and $w_2$), and approximates the best hypothesis $f$ by averaging the two.
Proposed Method: Leverage Tool Diversity

Task

Divide

Microtasks

Diff tools

× 2

× 2

× 2

× 2

× 2
Proposed Method: Leverage Tool Diversity

Task

Divide

Microtasks

Diff tools

Semantic image segmentation task

worker tools

crowd workers

worker answers

answer aggregator
Choosing the Tools

Q. How to diversify errors produced by different tool types?
Choosing the Tools

Q. How to diversify errors produced by different tool types?

Q. What are different types of objects?

A. General objects, Fuzzy materials, plants, furry objects, transparent objects, reflective surfaces (intuitive, deformability)
Instructions and Worker Interface

Worker Interface:

Basic Trace ($T_1$)

1. Click to see instructions.
2. Please find a **bowl** in the scene below.
3. Draw an outline of it by holding and dragging your left mouse button.
4. You can clear the outline by clicking **Space Bar** in your keyboard.
5. You have **30 seconds** to outline each object.

Click below to see the result:
- Check the Result
- Submit HIT
Instructions:

**ATTENTION!** Please carefully read the instructions before you ACCEPT HIT.

(Please don't accept HIT if you've previously worked on this task. You will not be paid more than once even though you complete the HIT multiple times.)

Welcome to our object aligning system. The task is to align all objects in the list with icon images. You'll have 30 seconds to align each object.

**Object Highlighting - mode 3 instruction**

1. Click the flashlight icon to highlight the flashlight on the screen.
2. Drag the flashlight to its correct position on the scene.

**Good example of aligning a flashlight**

**Bad example of aligning a flashlight**

Payments and Expected Time:
- Task (c. 5 minutes) tasks; $0.35 for successful completion.
- IMPORTANT: You must finish all HITs to get paid.

Please ACCEPT HIT to start task!
Experiment Settings

- 12 different visual scenes
- Total 51 objects
- Six unique workers for each tool-scene pair (total 288+ workers)
- Total 1224 object segmentations
- Platform: Amazon Mechanical Turk

Each worker was paid between $0.35 and $0.60 per task, depending on the number of objects they had to segment or on the level of difficulty of given tool (a pay rate of ~$10/hr).
Performance of Individual Tools

**Accuracy Metrics**

\[
\text{Precision (P)} = \frac{TP}{TP + FP} \\
\text{Recall (R)} = \frac{TP}{TP + FN} \\
\text{F}_1 \text{ Score} = \frac{2 \times P \times R}{(P + R)}
\]

- TP = True Positive
- FP = False Positive
- FN = False Negative

**Different tools have different error patterns**
Performance of Individual Tools

Accuracy Metrics

Precision (P) = \(\frac{TP}{TP + FP}\)
Recall (R) = \(\frac{TP}{TP + FN}\)
F₁ Score = \(\frac{2 \times P \times R}{P + R}\)

TP = True Positive
FP = False Positive
FN = False Negative

Different tools have different error patterns
What we observed

Accuracy Metrics

Precision (P) = \frac{TP}{TP + FP}

Recall (R) = \frac{TP}{TP + FN}

F_1 \text{ Score} = \frac{2 \times P \times R}{(P + R)}

TP = \text{True Positive}
FP = \text{False Positive}
FN = \text{False Negative}

Different tools have different error patterns
Some of the Answers from Workers
How can we see the effect of leveraging tool diversity?
Comparison of Aggregation Methods

Method 1. Single tool aggregation (Uniform majority voting): Baseline

\[ T_1 \rightarrow \text{Aggregate} \]

\[ T_2 \rightarrow \text{Aggregate} \]
Comparison of Aggregation Methods

Method 2. **Multiple** tool aggregation (Uniform majority voting)

\[ T_1 \times T_2 \rightarrow \text{Aggregate} \]

Method 3. **Multiple** tool aggregation (Expectation maximization)

\[ T_1 \times T_2 \rightarrow \text{Aggregate} \]
Comparison of Aggregation Methods

* significant at $p < .05$, ** significant at $p < .01$ compared to Multiple (EM)
Comparison of Aggregation Methods

High recall + high precision pairs gave the highest performance improvement.

* significant at $p < .05$, ** significant at $p < .01$ compared to Multiple (EM)
Generalization
Generalizability: Expected Human Error is Diverse
Generalizability: Aggregation Improves Quality

Quality Improves
Generalizability: Objective Correct Answer Exists

Tasks with objective answers:

- Image segmentation
- Live captioning
- Text annotation
- Handwriting recognition

Task with subjective answers:

Creative writing

This paper presents Soylent, a word processing interface that uses crowd workers to help with proofreading, document shortening, editing and commenting tasks. Soylent is an example of a new kind of interactive user interface in which the end user has direct access to a crowd of workers for assistance with tasks that require human attention and common sense. Implementing these kinds of interfaces requires new software programming patterns for interface software, since crowds behave differently than computer systems. We have introduced one important pattern, Find-Fix-Verify, which splits complex editing tasks into a series of identification, generation, and verification stages that use independent agreement and voting to produce reliable results. We evaluated Soylent with a range of editing tasks, finding and correcting 82% of grammar errors when combined with automatic checking, shortening text to approximately 85% of original length per iteration, and executing a variety of human macros successfully.
Generalizability: Tolerates Imperfections

Example: Scribe (UIST 2012)


```
1:   learn g is such a suitcase word though right so has a lot of there s a lot
2:   o learning is such learning is a lot
3:   learning ss such a suitcase word though learning has is a lot
4:   learning is su h a right so learning a lot
5:   so learning is such a suitcase though learning has lot
6:   learning is such a suitcase word though right this in a lot
F:   so learning is such a suitcase word though right so learning has a lot of there is a lot
```
Possible Future Applications

Application 1: Tagging Long Videos
- Context
- Granularity

Application 2: Multichannel NLP
- Text
- Audio

Application 3: Complex/Diverse Annotation
- Higher level
- Lower level

Application 4: Computer-Human Integration
- Precision
- Recall
Thank you!

Authors:
Jean Y. Song (jyskwon@umich.edu / jyskwon.github.io), Raymond Fok, Alan Lundgard, Fan Yang, Juho Kim, Walter S. Lasecki

Funding:
Denso Corporation
Toyota Research Institute
MCity at the University of Michigan
National Research Foundation of Korea
Backup Slides
Tool 1

Basic Trace ($T_1$)

1. Please find a **bowl** in Scene below.
2. Draw an outline of it by holding and dragging your left mouse button.
3. You can clear the outline by clicking **Space Bar** in your keyboard.
4. You have **30 seconds** to outline each object.
5. **Countdown:** 20 sec

Click below to see the result:
- **Check the Result**
- **Submit HIT**
Drag-and-Drop ($T_2$)

1. Please find a **bowl** in Scene below.
2. Find the most similar icon image from the Icon List and click.
   (Only shape matters. Colors do not have to match.)
3. **Drag/scale/rotate** the icon to **overlap** the bowl in Scene.
4. You have **30 seconds** to align each object.

Scene

Icon List

Click below to see the result:
- Check the Result
- Submit HIT

(click to select; scroll right to see more options)
Tool 3

Pin-Placing ($T_3$)

1. Please find a **bowl** in Scene below.
2. Find the most similar icon image from the Icon List and click.
   (Only shape matters. Colors do not have to match.)
3. Place **4 markers** on corresponding locations by **Left Click**.
4. You can deselect a marker by **Right Clicking** on it.
5. You have **30 seconds** to align each object.

Scene

Icon Image

Click below to see the result:

Check the Result

(Plase put all 4 markers on both Scene and Icon Image before you click)

Submit HIT
Tool 4

Floodfill ($T_4$)

1. Please highlight a bowl.
2. You are highlighting a bowl.

Please note that you cannot cancel highlight once you click the green button.

Go back to click different position   That's great!
Pixel-Level Majority Voting (50% agreement)

Worker 1

Worker 2

Worker 3

Worker 4

Aggregate

Final answer
In an image, label a pixel as 1 if it belongs to a target object, and 0 if background.

Assume:

- image \( A \) having \( N \) total pixels
- \( M \) crowd workers
- The label a worker \( m \) assigns to each pixel is denoted as \( z_{mn} \)
- all labels from worker \( m \) as a vector \( Z_m \)
- the true labels of \( A \) to be estimated are denoted as a vector \( Y \)
- \( \theta \) is the confusion matrices set to be estimated.

We can estimate the true labels \( Y \) by maximizing the marginal likelihood of the observed worker labels:

\[
I(\theta) := \log \left( \sum_{Y \in \{0,1\}^n} L(\theta; Y, Z) \right)
\]

The EM algorithm works iteratively by applying the 1) expectation step and the 2) maximization step.