Quality Control for Comparison Microtasks

Petros Venetis
Stanford University
Stanford, CA 94305
venetis@cs.stanford.edu

Hector Garcia-Molina
Stanford University
Stanford, CA 94305
hector@cs.stanford.edu

ABSTRACT

We study quality control mechanisms for a crowdsourcing system where workers perform object comparison tasks. We study error masking techniques (e.g., voting) and detection of bad workers. For the latter, we consider using gold-standard questions, as well as disagreement with the plurality answer. We perform experiments on Mechanical Turk that yield insights as to the role of task difficulty in quality control, and the effectiveness of the schemes.

1. INTRODUCTION

Crowdsourcing is becoming popular for tasks that humans can perform well, such as image labeling, text translation, and categorization. One common problem observed in various crowdsourcing marketplaces (e.g., Amazon's Mechanical Turk [1] and oDesk [5]) is the variance in worker quality. Some workers may be extremely good in performing a particular task and some others may be very bad. The bad workers may be performing badly either because they do not have the necessary training or because they are spammers (i.e., workers that only attempt to maximize their income without paying attention on the tasks they perform).

In this workshop paper we present some preliminary experimental results that shed some light on how to manage worker quality. In particular, we focus on two common approaches: masking versus detection. With masking, the same task is performed by multiple workers, and some type of voting is used to select the final output. With detection, the system tries to identify bad workers and somehow discount their results. For detection, we focus on two common approaches: using gold standard tasks (for which the correct answer is known) to evaluate workers, versus plurality agreement, where disagreeing with the plurality is counted as evidence that a worker is not good.

For our experiments we use Amazon's Mechanical Turk and focused on comparison tasks. (Comparison tasks are useful, for instance, for finding the best Facebook profile that matches a target person, for finding the most relevant URL for a given user query, or for detecting peak hours at a public place from a set of photographs.) Our tasks ask human workers to compare synthetically generated images, and to tell us which one has more dots. (Our tasks are inspired by the jelly-beans-in-a-jar experiment [8].) The advantage of our synthetic tasks is that: (a) there is a precise notion of correctness, (b) we know in advance the correct answer, (c) we can quantify the notion of "task difficulty", and (d) we can easily generate tasks with different difficulties. (That is, we can generate images with fewer or more dots, and we can compare images with similar or dissimilar numbers of dots.) We believe that our type of tasks are ideally suited for an exploratory study such as ours, where we wish to precisely quantify differences in approaches and we wish to study a wide variety of difficulty scenarios.

There are two types of questions we wish to address with our experiments:

- How can task difficulty be quantified and what role does it play in worker and task result quality? In particular, we consider an algorithm that finds the best object (in our case, the image with the maximum number of dots) in a set of objects, and we show that task difficulty must be considered when designing such an algorithm.
- How does the effectiveness of (a) masking versus detection techniques and also (b) plurality versus gold-standard evaluation techniques compare? The answer is not simply which one provides the most accurate answers, but also at what cost (i.e., number of worker tasks that must be paid).

2. PRELIMINARIES

Our work considers *comparison microtasks*: The requester posts a comparison microtask h that presents a set of items $\mathcal{E}(h)$ and wants to determine the maximum item from $\mathcal{E}(h)$ based on some *quality function*. (Set $\mathcal{E}(h)$ is a subset of a larger repository of items $\mathcal{E}$.) A worker observes $\mathcal{E}(h)$ and his response is an item $e \in \mathcal{E}(h)$ that he believes is the maximum in $\mathcal{E}(h)$. Workers are paid for each microtask performed. For error masking or bad worker detection, the requester specifies how many worker responses $r(h)$ he seeks from distinct workers, for each microtask h.

Each item $e \in \mathcal{E}$ has an inherent quality value $q(e) \in [0, \infty)$ and no other item $f \in \mathcal{E} \setminus \{e\}$ has the same quality value, i.e., for any $e, f \in \mathcal{E}$ with $e \neq f$, we have $q(e) \neq q(f)$. Quality metric $q : \mathcal{E} \to [0, \infty)$ forms a strict order relation "$<$" (less than) between the items in $\mathcal{E}$: For any $e, f \in \mathcal{E}$ with $e \neq f$, either $e < f$ (when $q(e) < q(f)$) or $f < e$ (when $q(e) > q(f)$).

CrowdKDD '12, August 12, 2012, Beijing, China

For a subset $S \subseteq \mathcal{E}$, the item $e^*(S)$ for which $e < e^*(S)$ for all $e \neq e^*(S)$ ($e \in S$) is called the *maximum item* in S. (For example, $e^*(\mathcal{E})$ is the maximum item in $\mathcal{E}$.)

2.1 Comparison Microtask Difficulty

Intuitively, the difficulty of a microtask plays a critical role in quality control, since harder microtasks are more error prone. To study the impact of difficulty, it is important to have a concrete microtask difficulty metric. While there are many ways to define difficulty, even in the context of the comparison microtasks we are considering here, in our experiments we have found the following definition to be quite useful.

We define the difficulty of comparison microtask h as:

$$\text{diff}(h) = \frac{q_2(h)}{q_1(h)} \tag{1}$$

where $q_i(h)$ is the i^{th} highest quality value that an item in $\mathcal{E}(h)$ has. This definition is surprisingly simple: It only takes into account the ratio of the top two qualities, not the absolute quality numbers and not the quality of other objects in the comparison set. For instance our metric says that a microtask with $q_1(h) = 500$ and $q_2(h) = 400$ is just as difficult as one with $q_1(h) = 5$ and $q_2(h) = 4$. We will validate the usefulness of this simple metric in our experiments.

3. QUALITY CONTROL MECHANISMS

There are many ways to improve the quality of results from a crowdsourcing system, everything to recruiting good workers, to incentivizing them to produce good results, to disregarding or downplaying the results from bad workers. For concreteness in our experiments, we focus on the last category: We wish to map the microtask responses given by a set of workers into the final responses for the microtasks. For instance, if for the same microtask two workers gave a response A and two other workers gave response B, what should the system produce as the microtask answer? If say one of the A workers is thought to be a bad worker, we may want to disregard his answer, and use B as our final answer. Or say we have estimated the accuracy of the two A workers to be 0.9, and the accuracy of the two B workers to be 0.8, then we may go with the A answer.

As discussed in the introduction, here we consider two common techniques for producing microtask answers:

Masking: Repeat each microtask h a total of $r(h)$ times and use the result given by the plurality of workers.

Detection: Assign a *score* to workers, based on their results (see below). Then, based on the scores, eliminate the results from workers with a score below some threshold. Then use masking as before.

3.1 Score Assignment Methods

A scoring method assigns a score $s(w) \in [0, 1]$ to a worker w or a worker-microtask score $s(w, h) \in [0, 1]$ to the response a worker w gave for microtask h. In this paper we consider three scoring methods.

3.1.1 Gold Standard Performance

One simple way to detect a worker's quality is for the requester to ask the worker to perform some microtasks for which the requester knows the correct answer beforehand; these microtasks are known as *gold standard microtasks*. The worker should not be able to tell the difference between gold standard microtasks and the rest.

Assuming that a worker w has performed G gold standard microtasks and he has responded correctly in $g \in \{0, 1, \ldots, G\}$ of them, we assign a score

$$s_{\text{GS}}(w) = \frac{g}{G} \tag{2}$$

Since $0 \leq g \leq G$, score $s_{\text{GS}}(w)$ lies in $[0, 1]$ in all cases.

3.1.2 Plurality Answer Agreement

If we assume that most workers are good, then workers producing a minority result are likely to be bad. Thus we define a score

$$s_{\text{P}}(w) = \frac{f}{F} \tag{3}$$

to worker w, where F is the number of microtasks w has performed and f is the number of microtasks for which w agreed with the plurality answer. Since $0 \leq f \leq F$, $s_{\text{P}}(w) \in [0, 1]$ in all cases.

3.1.3 Microtask Work Time

Typical crowdsourcing microtasks do not require special skills and can typically be performed by workers in a few seconds. Thus, an important factor to evaluate worker quality is the time spent on a microtask. The reasoning in that spammers and other bad workers probably are not spending enough time on their microtasks. To capture this behavior, we assign to the answer worker w gave for microtask h a score

$$s_{\text{T}}(w, h) = \frac{t(w, h)}{T(h)} \tag{4}$$

where $T(h)$ is the maximum time a worker can spend on h and $t(w, h)$ is the actual time w spent while performing h. Since $0 \leq t(w, h) \leq T(h)$, $s_{\text{T}}(w, h) \in [0, 1]$ in all cases. The higher $s_{\text{T}}(w, h)$ is, the more time w spent on h and thus probably he is not a spammer.

4. EXPERIMENTS

We now describe a set of experiments performed on Amazon's Mechanical Turk (AMT in the following) that:

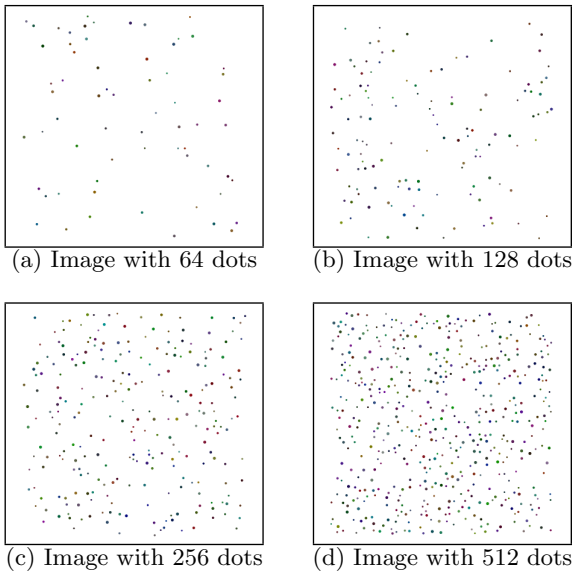
- Explores the connection between comparison microtask difficulty and worker accuracy (Section 4.3),
- Observes comparison microtask difficulty over time for tournament max algorithms (Section 4.4), and
- Studies and compares various score assignment methods (Section 4.5).

Before we delve into the experiments, we describe the dataset we have constructed (Section 4.1) and some key statistics about the experiments (Section 4.2).

4.1 Dataset

To be able to reason about microtask difficulty, we need a dataset $\mathcal{E}$ for which each item $e \in \mathcal{E}$ has a known quality value. Inspired by the jelly-beans-in-a-jar experiment [8], we created a dataset of images each containing a number $d(e)$ of colored dots, where $d(e) \in \{1, 2, \ldots D\}$. We define the quality value of image e as $q(e) = \frac{d(e)}{D}$. A small sample of our dataset is presented in Figure 1. We have generated 20 images (items) with 1 dot, 20 images with 2 dots, ..., 20

images with $D = 1{,}000$ dots. Given a set of items $S \subseteq \mathcal{E}$, a comparison microtask asks a worker to determine the image with the maximum number of dots in it.



(a) Image with 64 dots (b) Image with 128 dots

(c) Image with 256 dots (d) Image with 512 dots

Figure 1: Four images obtained from our dataset, each with different number of dots.

4.2 Statistics

For the experiments we describe in the following sections, we have posted several comparison microtasks to AMT and many different workers have worked on our microtasks. This section describes several key statistics about all posted microtasks (without bad worker detection and elimination).

In total, we posted ∼28,500 distinct comparison microtasks; each comparison asked workers to compare 4 images. The number of worker responses desired by a microtask h ($r(h)$) varied between 1 and 5 for different comparison microtasks. In total, we received ∼54,000 worker responses from ∼1,100 distinct workers. Figure 2 presents the distribution of workers (y-axis) that have performed particular numbers of comparisons (x-axis). We see that there are some tens of workers that have performed at most 10 microtasks (top left part of the graph), but there are also a few workers that each has performed some hundreds of microtasks (bottom right part of the graph). This distribution of workers across performed microtasks is typical in realistic crowdsourcing scenarios.

Crowdsourcing environment workers come from many different countries and cultures, they have different skills and behaviors, and each expects different pay rates from the requesters [2]. To get a good geographical coverage (and thus a relatively balanced sample of different worker characteristics), we posted microtasks to AMT on different times of the day and different days of the week. For example, we never posted more than 50 HITs per hour for the experiments described in the following sections.

4.3 Microtask Difficulty Effects

This section describes an experiment that sheds light in understanding how comparison microtask difficulty affects

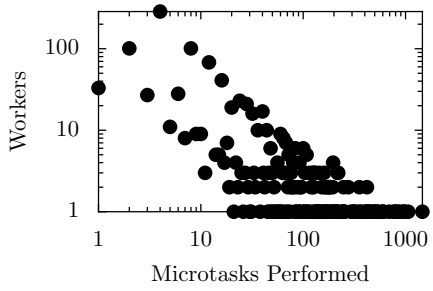


Figure 2: Distribution of workers across different numbers of performed comparison microtasks.

the (average) worker accuracy. For this experiment, we do not filter any bad worker responses.

Setting For this experiment, we constructed comparison microtasks for which a worker compares 4 images. We controlled the two images with the highest numbers of dots ($q_1(h) > q_2(h)$). The other two images had $q_3(h)$ and $q_4(h)$ selected at random with the restriction that $q_4(h) \leq q_3(h) < q_2(h)$. We used many different combinations for values $q_1(h)$ and $q_2(h)$ (seen in Table 1). For each combination of $q_1(h)$ and $q_2(h)$ we posted 500 HITs on AMT; each HIT contained 4 comparison microtasks (thus, in total we posted 2,000 comparison microtasks per combination of $q_1(h)$ and $q_2(h)$ values) with the particular $q_1(h)$ and $q_2(h)$ values. After the HITs for one combination of $q_1(h)$ and $q_2(h)$ values were completed, we moved on to a different combination of $q_1(h)$ and $q_2(h)$ values. Each HIT was compensated with $0.01.

$q_1(h)$	$q_2(h)$	diff(h)	$q_1(h)$	$q_2(h)$	diff(h)
100	50	0.5	500		
	75	0.75		375	0.75
	90	0.9		450	0.9
	95	0.95		475	0.95
	97	0.97		485	0.97

Table 1: The combinations of quality values used and the corresponding difficulty of the microtask.

Observations We present the results in Figure 3. The x-axis is the microtask difficulty for the microtasks in some combination of values from Table 1. The y-axis is the fraction of worker responses that were correct. We present two lines: One for the cases where $q_1(h) = 100$ and one to the cases where $q_1(h) = 500$. There are some minor differences between the two lines in Figure 3, but these differences are never higher than 0.04. The small difference between the two lines, suggests that the actual values of $q_1(h)$ and $q_2(h)$ are not very important to define a microtask's difficulty. Comparison difficulty can be captured by the relative values of $q_2(h)$ and $q_1(h)$ and our definition of comparison microtask difficulty (Section 2.1) captures exactly that.

We observe that for relatively easy microtasks (difficulty at 0.5), workers performed exceptionally, by answering correctly 99.1% of the microtasks. The fraction of correct responses is reduced significantly as microtask difficulty increases: We only get ∼55% correct responses when microtask difficulty is 0.97. The range of worker accuracy values is significantly large across different microtask difficul-

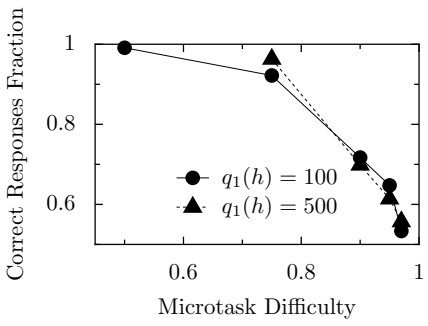


Figure 3: Exploring the probability of a worker giving the correct response for a microtask versus the microtask difficulty.

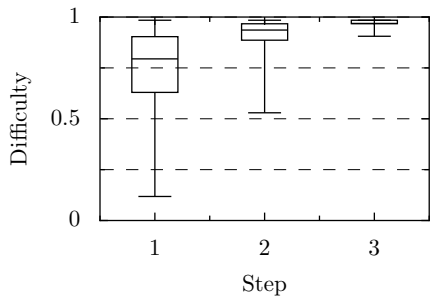


Figure 4: The microtask difficulties across steps for the tournament algorithm runs.

ties. Thus, real crowdsourcing algorithms have to consider (worker) models with significantly different worker accuracy for easy and hard microtasks. We delve more into this in the next section, where we discuss how particular max algorithms (tournament algorithms) should be designed in practice to take this effect into account.

Of course, there are many other factors that can impact worker accuracy apart from microtask difficulty. Some of them are: (1) the number of images shown in the comparison microtask, (2) the number of comparison microtasks per HIT, and (3) whether the comparison microtask asks workers to compare dots or images with real people. While we have not (yet) studied these factors in detail, microtask difficulty should be factored into an algorithm like max.

4.4 Difficulty in Tournament Algorithms

This section describes an experiment performed to understand how the comparison microtask difficulty evolves for standard max algorithms. We did not apply any quality control mechanisms (Section 3) for the experiments presented in this section.

Setting We use tournament algorithms [9] that attempt to retrieve the item e with the highest quality value $q(e)$ from a set of items $\mathcal{E}$, i.e., $e^*(\mathcal{E})$. A tournament algorithm operates in steps. Some microtasks are posted during step 1. When all worker responses have been given for step 1, microtasks for step 2 are posted and so on. A tournament algorithm operates as follows. Set $\mathcal{E}_i$ contains the items that are compared at step i; $\mathcal{E}_1 = \mathcal{E}$. At step i, $\mathcal{E}_i$ is (randomly) partitioned in non-overlapping sets (S_j, for $j = 1, 2, \ldots, \lceil \frac{|\mathcal{E}_i|}{s_i} \rceil$) of size s_i ($|S_j| = s_i$). Sets S_j are given to r_i workers each to be compared and the winners of each comparison form set $\mathcal{E}_{i+1}$. If $\mathcal{E}_{i+1}$ has exactly one item, this item is the output of the tournament algorithms. Otherwise, the process is repeated for one more step.

For our experiment, we fix $r_i = 5$ (i.e., 5 workers respond to each comparison microtask) and $s_i = 4$ (i.e., each microtask shows 4 images) for every step i. We run 20 tournament algorithms that attempt to find the maximum item from a set of 64 images: One with 5 dots, one with 10 dots, ..., and one with 320 dots. We include two comparison microtasks per HIT and compensate workers with \$0.01 per HIT they complete. Since we show 4 images per comparison microtask, the tournament algorithms are completed in 3 steps.

Observations Figure 4 presents the comparison microtask difficulties observed (in the 20 tournaments we run) across steps using box/whiskers plots. Each box/whiskers plot shows the range of difficulty values observed for a step. The top bar represents the maximum value observed, the bottom bar is the lowest value. The bottom of the box is the first quartile (minimum value above the 25% percent of the observed values), the top of the box is the third quartile (minimum value above 75% percent of the observed values), and the line in the middle of the box is the median value observed.

From Figure 4 we make two observations. We first observe that the range of microtask difficulty values decreases as the step increases. For example, during step 1, we observe difficulty values between 0.15 and 0.95, but during step 3, we observed difficulty values between 0.85 and 1. The range of difficulty values during step 1 is high, because items are assigned to random comparisons before they are compared by workers. Thus, there is a chance that the top-4 items are included in the same microtask comparison (with a difficulty close to 1), but there is also a chance that the top-1 item and the bottom-3 items are included in the same microtask comparison (with a difficulty close to 0).

We also observe that on average the difficulty values increase as the step increases. At step 1 the median difficulty value is 0.8, at step 2 the median value is 0.95, and at step 3 the median value is 0.98. The difficulty value increase is expected, since the items that are considered in final steps are the "winners" of previous steps and thus closer (on average) to the maximum item and to each other. Our observations lead to the conclusion that more worker responses $r(h)$ are required towards the end of a tournament algorithm to obtain the correct answer (i.e., the actual maximum item). This coincides with previous work: [9] concludes (via simulations) that more worker responses should be requested as we get closer to the end of a tournament to improve result accuracy.

4.5 Worker Scores, Masking, and Detection

This section studies different score assignment methods, masking, and bad worker detection. First, we study the impact of $r(h)$ on the comparison microtask final answer quality (Section 4.5.1). Then, we study the correlation of various score assignment methods (Section 4.5.2) and compare score assignment methods performance (Section 3.1). Finally, we describe a cost analysis that compares masking and detection (Section 4.5.4).

4.5.1 $r(h)$ Impact on Microtask Accuracy

As we have seen for masking, a comparison microtask h is given to $r(h)$ distinct workers, their responses are aggregated, and the final answer is the response with the highest number of votes (plurality vote). We now study how changing $r(h)$ affects the probability of getting a correct final answer, without eliminating any worker answers.

Setting We posted 1,000 HITs, each containing 3 comparison microtasks with difficulty equal to 0.84 and another 1,000 HITs, each containing 3 comparison microtasks with difficulty equal to 0.9. We asked $r(h) = 5$ workers to perform each HIT. We then used the first $r = 1, 2, \ldots, 5$ of the $r(h)$ responses per HIT and measured the fraction of comparisons for which the plurality vote of the r answers was correct. Our experiment is essentially the same as having 5 separate experiments: One with $r(h) = 1$, one with $r(h) = 2$, ..., and one with $r(h) = 5$. To save money, we used $r(h) = 5$ and prefixed the responses appropriately.

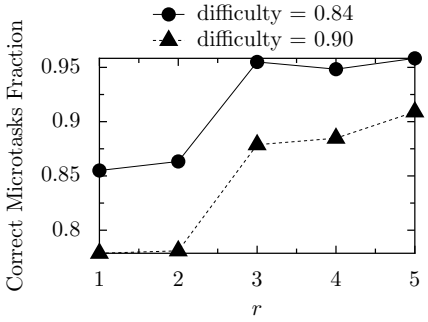


Figure 5: Impact of r (or $r(h)$) on microtasks' final answer accuracy.

Observations Figure 5 presents the results: r is presented in the x-axis and the fraction of correct comparisons is presented in the y-axis. There are two lines in the graph: One for difficulty equal to 0.84 and one for difficulty equal to 0.90. We observe that the fraction of correct comparisons is higher for the comparisons with lower difficulty. We also observe that there is no significant improvement when r is increased from 1 to 2; the reason is that the probability of the plurality vote when $r = 2$ is the correct when either both responses are correct (or when one worker made the correct choice and the tie is broken correctly). There is a significant improvement when r increases from 2 to 3. Then, for the low difficulty comparisons we reach a plateau; for the high difficulty comparisons there is a (slight) improvement while r increases up to 5.

We should mention that Figure 5 is consistent with Figure 3. That is, given the microtask difficulty, Figure 3 can predict the correctness rate. Given the correctness rate, one can compute the probability that the plurality is correct [9] and this is similar to the probability retrieved by Figure 5.

Although we did not eliminate any worker answers (we only used masking) for this experiment, we can see that changing the r (or $r(h)$) value has a severe impact on a microtask's final answer. Also, there is a clear tradeoff between the requester's budget (directly related to $r(h)$) and the quality achieved in the final answer. For example, when the difficulty is 0.9, one can get the fraction of correct comparisons to 0.78 with enough budget for one worker response ($r(h) = 1$); with enough budget for 5 worker responses, the fraction of correct comparisons is increased to 0.96. The requester can choose the desired accuracy according to his application needs and, thus, he can choose appropriately his monetary budget.

4.5.2 Score Correlations

We now explore how well our proposed score assignment methods characterize a worker.

Setting For our experiments we fixed the microtask difficulty at 0.9 ($q_1(h) = 500$ and $q_2(h) = 450$) and posted 1,000 HITs to AMT. Each HIT contained 4 comparison microtasks and we asked 5 workers to perform each HIT. Worker compensation per HIT was \$0.05.

We explore the correlation of our two proposed worker scores ($s_{\text{GS}}(w)$ and $s_{\text{P}}(w)$) with the worker actual score. For s_{GS} we assumed for our experiments that the first of each $K = 5$ microtasks performed by each worker belonged to the gold standard. (We have tried various values for K and obtained similar results as the ones we present here.) In the extreme case that $K = 1$, $s_{\text{GS}}(w)$ becomes the worker's *actual score*, i.e., the fraction of the comparisons w has performed for which he responded correctly; we denote this score by $s_{\text{A}}(w)$.[1]

Observations The Pearson correlation value for s_{A} and s_{GS} is 0.90; for s_{A} and s_{P} it is 0.75. Since s_{GS} is more correlated with s_{A} than s_{P} is, the gold standard performance is a better proxy for a worker's quality than the plurality answer agreement. Although both s_{GS} and s_{P} have similar performance, they both have their drawbacks. For example, s_{GS} needs the generation of gold standard microtasks, the requester has to pose them in a way that the worker does not know they are gold standard microtasks, and finally the requester has to pay workers for the gold standard microtasks and thus he spends money with no direct return. On the other hand, s_{P} costs a lot to compute (one needs $r(h)$ to be greater than 1 for s_{P} to be computed).

If we restrict ourselves to workers that have performed 10 or more comparisons, then the correlation values increase significantly: The correlation of s_{A} and s_{GS} increases to 0.97 and the correlation of s_{A} and s_{P} increases to 0.92. The increase suggests that the more comparisons a worker has performed (and thus the more information we have for a worker), the more accurately we can create a score that describes his quality. For the workers that have performed 10 or more comparisons, we plot the corresponding values for $s_{\text{A}}(w)$ and $s_{\text{GS}}(w)$ in Figure 6 and we observe that there is a clear linear relationship between the two scores.

4.5.3 Score Assignment Comparison

We now compare the effectiveness of the score assignment methods we described in Section 3.1 when using bad worker detection.

Setting We used the same HITs as in Section 4.5.2: 1,000 HITs each containing 4 comparison microtasks of fixed difficulty 0.9 ($q_1(h) = 500$, $q_2(h) = 450$, $q_3(h)$ and $q_4(h)$ were selected at random such that $q_3(h), q_4(h) < 450$) and we

[1]We note that s_{T} is not a workers score, but a worker-microtask score. That is, s_{T} characterizes one worker response for a specific microtask, not the worker's behavior in multiple microtasks.

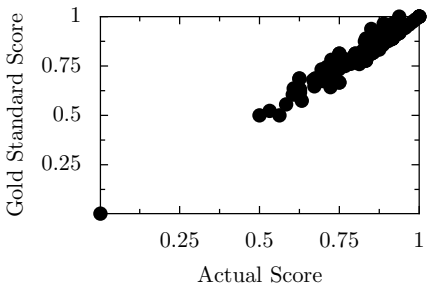


Figure 6: Correlation between $s_{\mathbf{GS}}$ (gold standard score) and $s_{\mathbf{A}}$ (actual score).

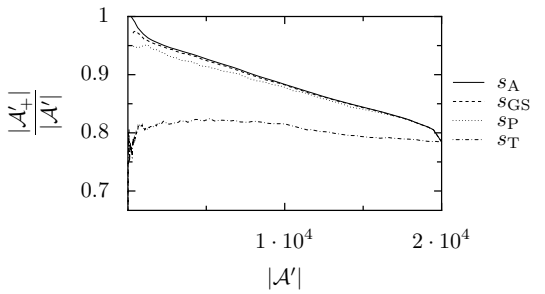
paid each worker $0.05 for each HIT he completed. We asked $r(h) = 5$ distinct workers to perform each comparison and we denote the responses for comparison microtask h as $a_1(h)$, $a_2(h)$, ..., $a_5(h)$. (In total, we had 4,000 distinct comparison microtasks and 5 worker responses for each comparison.) We call the set of all 4,000 comparisons $\mathcal{H}$ and the set of all $4{,}000 \times 5 = 20{,}000$ answers $\mathcal{A} = \{a_i(h) : i \in \{1, 2, \ldots, 5\}, \forall h \in \mathcal{H}\}$.

Observations After obtaining the worker answers, we used various threshold values t and ignored answers $a_i(h)$ for each microtask h provided by a worker w with $s_{\mathrm{X}}(w) < t$ or $s_{\mathrm{X}}(w, h) < t$ for X = A, GS, P, T. After this filtering, we are left with a set of answers $\mathcal{A}' \subseteq \mathcal{A}$; from $\mathcal{A}'$ only some answers $\mathcal{A}'_+$ are correct ($\mathcal{A}'_+ \subseteq \mathcal{A}'$). We plot the fraction of correct answers from $\mathcal{A}'$ $\frac{|\mathcal{A}'_+|}{|\mathcal{A}'|}$ (y-axis) versus the size of the "remaining" answers $|\mathcal{A}'|$ (x-axis) in Figure 7(a).

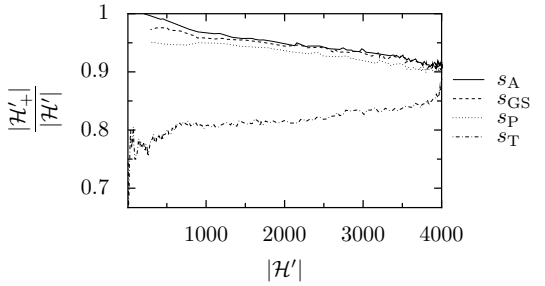
Because of the answer filtering, some comparison $h \in \mathcal{H}$ may or may not have any remaining answers in $\mathcal{A}'$. (For example, for a comparison microtask h, all responses may have been performed by workers with $s_{\mathrm{X}}(w) < t$ and thus will be filtered out. In this case, comparison h is left with no answers.) We denote the set of comparisons that have at least one remaining answer $\mathcal{H}'$. (For the previous example, the microtask $h \notin \mathcal{H}'$ since it is left with no answers after the filtering.) Once the remaining answers for a comparison h are collected, the plurality vote can be retrieved and compared to the (actual) correct answer for h; the comparisons $h \in \mathcal{H}'$ with a correct plurality answer form set $\mathcal{H}'_+$. In Figure 7(b) we plot the fraction of comparisons from $\mathcal{H}'$ for which the plurality vote from the remaining answers is correct $\frac{|\mathcal{H}'_+|}{|\mathcal{H}'|}$ (y-axis) versus the number of comparisons with at least one remaining answer $|\mathcal{H}'|$ (x-axis).

For both graphs, the x-axis corresponds to various t values for the score methods we have described. Low t values correspond to the right part of the x-axis (only a few very bad workers are eliminated), while high t values correspond to the left part of the x-axis (only a few very good workers remain unfiltered, and we have answers for fewer tasks). The lines corresponding to s_{A} for both graphs express ideal case for which we know the actual quality of each worker.

We observe that s_{GS} and s_{P} perform almost as well as the actual worker score (s_{A}), as expected from the high correlation between both s_{GS} and s_{P} with s_{A}. There is a small difference between the performance of s_{GS} and s_{P}: s_{GS} performs slightly better. In the following, we present a cost analysis of these two score methods to understand their dynamics more.



(a) Correct answers fraction versus remaining answer after response elimination.



(b) Correct comparison microtasks fraction versus remaining comparison microtasks with answers after response elimination.

Figure 7: Comparing the effectiveness of the three score assignments methods: $s_{\mathbf{GS}}$, $s_{\mathbf{P}}$, and $s_{\mathbf{T}}$.

We conclude with two key observations. First, we note that although one can improve the fraction of correct comparisons by eliminating bad workers, the improvements are not very high (moving from the right to the left part of the graph in Figure 7(b)). We provide more intuition in the next experiment. Second, we see in both graphs that $s_{\mathrm{T}}(w, h)$ is not a very good indicator of the quality of a worker w's work on h (neither for $\mathcal{A}'$ nor for $\mathcal{H}'$), but potentially it can be combined in a richer signal that takes more information into account.

4.5.4 Cost Analysis: Masking versus Detection

We now explore the cost of masking and detection to achieve a particular benefit. We use the same type HITs and the same notation ($\mathcal{A}$, $\mathcal{A}'$, $\mathcal{A}'_+$, $\mathcal{H}$, $\mathcal{H}'$, and $\mathcal{H}'_+$) as in Section 4.5.3.

Setting After we filter some worker answers based on worker scores, each correct plurality answer for a microtask $h \in \mathcal{H}'$ (also, $h \in \mathcal{H}'_+$ since the plurality answer is correct) gives a positive benefit $b_p > 0$. Each incorrect plurality answer for $h \in \mathcal{H}'$ (i.e., $h \in \mathcal{H}' \setminus \mathcal{H}'_+$) gives a negative benefit $b_n \leq 0$. For the comparison microtasks in $\mathcal{H}'$, we can calculate the total benefit using $\text{Benefit} = \sum_{h \in \mathcal{H}'_+} b_p + \sum_{h \in \mathcal{H}' \setminus \mathcal{H}'_+} b_n$. To achieve this value of benefit, the requester issued a number of HITs and paid the workers $\text{Cost} = \sum_{h \in \mathcal{H}} r(h)$ for their answers.

Observations Figure 8 presents the $\frac{\text{Cost}}{\text{Benefit}}$ on the y-axis and the various values of $|\mathcal{H}'|$ in the x-axis (for various score thresholds t). We used either $r = 5$ answers per comparison microtask or $r = 3$ and the two worker scores for the detection: s_{GS} and s_{P}. The right part of the x-axis corresponds to

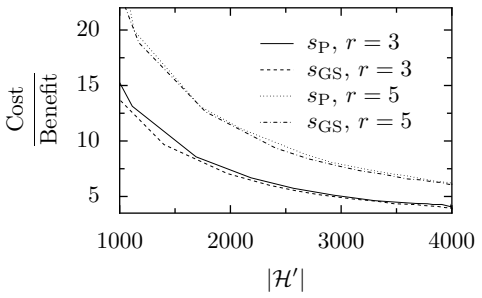


Figure 8: Exploring the cost associated with the benefit that we get for comparisons.

very few worker eliminations and the left part of the x-axis corresponds to many worker eliminations. For this graph we used $b_p = 1$ and $b_n = -1$; we have also experimented with different values of b_p and b_n and obtained very similar trends.

We first observe that $\frac{\text{Cost}}{\text{Benefit}}$ takes its lowest value at the very right of the graph (i.e., when no workers are eliminated and we only use masking) both when $r = 3$ and when $r = 5$. Since masking (without worker elimination) gives the minimum cost per benefit value, it should be preferred in practice comparing to worker elimination with the gold standard performance and the plurality agreement metric. In any case, between the two score methods, we observe that s_{GS} gives a lower cost per benefit than s_{P} in all cases and thus gold standard performance is a (slightly) better alternative than plurality agreement.

Finally, we see that the same values for $\frac{\text{Cost}}{\text{Benefit}}$ can be achieved by different values of r. For instance, we can achieve $\frac{\text{Cost}}{\text{Benefit}} = 10$ with both $r = 3$ and with $r = 5$, but with different score threshold values: For $r = 3$, set $\mathcal{H}'$ contains ∼1,500 microtasks, while for $r = 5$, set $\mathcal{H}'$ contains ∼2,300 microtasks. The requester, according to his application needs (quality and desired number of completed microtasks), can select how aggressive (if at all) he will be with worker elimination.

5. RELATED WORK

The notion of easy and hard microtasks is used in [3], where each microtask asks workers to click back and forth between two narrow vertical bars separated by some number of pixels (the distance between the bars determines the difficulty of the task). The focus of [3] is not the study of microtask difficulty or quality control mechanisms, but the estimation of the reservation wage (i.e., the minimum wage a worker is willing to accept for a task). To estimate the reservation wage, [3] uses a labor supply model and concludes that the median for AMT is $1.38 per hour.

There has been a lot of work on quality control mechanisms. In [10], various mechanisms (based on worker response patterns, gold sets, and plurality voting) are compared via simulations. Also, in [4] a mechanism for microtasks that have two possible responses is developed. Another mechanism for aggregating multiple responses for subjective details is described in [6]. These works lack experiments that compare various quality control mechanisms with each other on a real crowdsourcing marketplace.

One important aspect of crowdsourcing marketplaces explores how to motivate AMT workers. In [7], the effectiveness of various social and financial incentive schemes is observed. (An example of a social incentive scheme is the addition on the microtask description of the phrase "Before you complete the questions, I just wanted to thank you again for doing this work. My name is Aaron." An example of a financial incentive scheme is the promise that if a worker's response agrees with the plurality response, he will be rewarded with a monetary bonus.) Our work focuses on different strategies for refining the result of an algorithm (or a microtask) and we do not explore how to incentivize workers. The quality control techniques we describe can be applied orthogonally to the incentives provided to the workers.

6. CONCLUSION AND FUTURE WORK

We have experimentally studied worker quality control mechanisms for a crowdsourcing system. Because we know the correct answers for our synthetic microtasks, we could precisely determine the effectiveness of the various mechanisms. In particular, we studied error masking and detection of bad workers using different scoring functions, evaluating the impact on task accuracy, the number of completed microtasks, and on the cost/benefit ratio. We also showed that task difficulty is an important factor for worker accuracy and must be taken into account in the design of crowdsourced algorithms (such as the tournament algorithm we studied). The results we presented are preliminary, but we believe that similar experiments can be used to study other quality control mechanisms and additional effectiveness metrics.

Acknowledgments

We thank Neoklis Polyzotis for useful discussions on this work.

7. REFERENCES

[1] Amazon's Mechanical Turk. www.mturk.com, June 2012.

[2] Paul Heymann and Hector Garcia-Molina. Turkalytics: Analytics for Human Computation. WWW, pages 477–486, 2011.

[3] John Joseph Horton and Lydia B. Chilton. The Labor Economics of Paid Crowdsourcing. EC, pages 209–218, 2010.

[4] David R. Karger, Sewoong Oh, and Devavrat Shah. Iterative Learning for Reliable Crowdsourcing Systems. In *NIPS*, pages 1953–1961, 2011.

[5] oDesk. www.odesk.com, June 2012.

[6] Drazen Prelec. A Bayesian Truth Serum for Subjective Data. *Science*, 306(5695):462–466, 2004.

[7] Aaron D. Shaw, John J. Horton, and Daniel L. Chen. Designing Incentives for Inexpert Human Raters. CSCW, pages 275–284. ACM, 2011.

[8] James Surowiecki. *The Wisdom of Crowds*. Anchor, 2005.

[9] Petros Venetis, Hector Garcia-Molina, Kerui Huang, and Neoklis Polyzotis. Max Algorithms in Crowdsourcing Environments. In *WWW*, pages 989–998, 2012.

[10] Jeroen Vuurens, Arjen P. de Vries, and Carsten Eickhoff. How Much Spam Can You Take? An Analysis of Crowdsourcing Results to Increase Accuracy. CIR, 2011.