Task Search in a Human Computation Market

Lydia B. Chilton
University of Washington
hmslydia@cs.washington.edu

John J. Horton
Harvard University
horton@fas.havard.edu

Robert C. Miller
MIT CSAIL
rcm@mit.edu

Shiri Azenkot
University of Washington
shiri@cs.washington.edu

ABSTRACT

In order to understand how a labor market for human computation functions, it is important to know how workers search for tasks. This paper uses two complementary methods to gain insight into how workers search for tasks on Mechanical Turk. First, we perform a high frequency scrape of 36 pages of search results and analyze it by looking at the rate of disappearance of tasks across key ways Mechanical Turk allows workers to sort tasks. Second, we present the results of a survey in which we paid workers for self-reported information about how they search for tasks. Our main findings are that on a large scale, workers sort by which tasks are most recently posted and which have the largest number of tasks available. Furthermore, we find that workers look mostly at the first page of the most recently posted tasks and the first two pages of the tasks with the most available instances but in both categories the position on the result page is unimportant to workers. We observe that at least some employers try to manipulate the position of their task in the search results to exploit the tendency to search for recently posted tasks. On an individual level, we observed workers searching by almost all the possible categories and looking more than 10 pages deep. For a task we posted to Mechanical Turk, we confirmed that a favorable position in the search results do matter: our task with favorable positioning was completed 30 times faster and for less money than when its position was unfavorable.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval; J.4 [**Social and Behavioral Sciences**]: Economics

General Terms

Economics, Experimentation, Human Factors, Measurement

KDD-HCOMP'10, July 25th, 2010, Washington,DC, USA.

Keywords

Search, Amazon Mechanical Turk, Human Computation, Crowdsourcing, Experimentation

1. INTRODUCTION

In every labor market, information plays a critical role in determining efficiency: buyers and sellers cannot make good choices unless they know their options. According to a rational economic model of labor supply, workers use information about the availability and the nature of tasks to make decisions about which tasks to accept. If workers lack full information about the tasks, they are likely to make suboptimal decisions, such as accepting inferior offers or exiting the market when they would have stayed, had they known about some other task. In large markets with many buyers and sellers, lack of knowledge about all available options is a key source of friction. This lack of knowledge stems from the fact that searching is neither perfect nor costless.

The "search problem" is particularly challenging in labor markets because both jobs and workers are unique, which means that there is no single prevailing price for a unit of labor, nevermind a commodified unit of labor. In many labor markets, organizations and structures exist to help improve the quality and quantity of information and reduce these "search frictions." We have already seen conventional labor markets augmented with informational technology, such as job listing web sites like Monster.com [2], and yet a great deal of of information is privately held or distributed through social networks and word of mouth.

In online labor markets, there is relatively little informal sharing of information as in traditional markets. Given that online labor markets face no inherent geographic constraints, these markets can be very large, further exacerbating search frictions. These frictions can and are eased by making the data searchable through a variety of search features. If agents can sort and filter tasks based on characteristics that determine the desirability of a task, they are able to make better choices, thereby improving market efficiency. Of course, this hypothetical gain in efficiency depends upon the search technology provided and the responses of buyers to that technology.

Amazon's Mechanical Turk (MTurk) is an online labor market for human computation. It offers several search features that allow workers to find Human Intelligence Tasks (HITs). HITs can be sorted by fields such as most available, highest reward, and title. HITs are also searchable by keyword, minimum reward, whether they require qualifications (and combinations thereof). In this paper, we seek to provide insight into how workers search for tasks.

We present the results of two methods for investigating search behavior on MTurk. First, we scrape pages of available HITs at a very high rate and determine the rate at which a type of HIT is being taken by workers. We use a statistical model to compare the rate of HIT disappearance across key categories of HITs that MTurk allows workers to search by. Our premise for observing search behavior is that search methods which return HITs with higher rates of disappearance are the search methods which workers use more. This method relies only on publicly available data—the list of HITs available on www.mturk.com. Second, we issued a survey on MTurk asking over 200 workers how they searched for tasks. We posted the survey with carefully chosen parameters so as to position it so that it can be found more easily by some search methods and not others. This way, we can target search behaviors that are not picked up by the analysis of the scraped data.

In this paper, we first motivate studying search behavior in an online human computation labor market by reviewing related work. We next describe MTurk's search features and our method of data collection based on these search features. We then present a statistical analysis of our high-frequency scraped data followed by the results of a survey of MTurk workers.

2. RELATED WORK

In the field of human computation, it is important to ask "Why are people contributing to this effort?" What are the motivations behind labeling an image [17], uploading videos to YouTube [11], or identifying the genre of a song [13]? On MTurk, the main motivation seems to be money [12, 10]. However, financial motivations do not imply that workers are only motivated to find HITs offering the highest rewards: workers may choose from among thousands of HITs that differ in required qualifications, level of difficulty, amount of reward, and amount of risk. With all of these factors at play, workers face significant decision problems. Our task was to model this decision problem.

Because MTurk is a labor market, labor economics—which models workers as rational agents trying to maximize net benefits over time—may provide a suitable framework. Research on the labor economics of MTurk has shown that workers do respond positively to prices but that price is not indicative of work quality [14]. Previous work [9] shows that not all workers follow a traditional rational model of labor supply when choosing their level of output. Instead of maximizing their wage rate, many workers create target earnings—they focus on achieving a target such as $.25 or $1.00 and ignore the wage rate. Although price is important to task choice, it is clearly not the only factor; research on search behavior is necessary in order to more fully understand how workers interact with the labor market.

In other domains where users must cope with a large information space, progress and innovation in search have had enormous positive impacts. Developments such as PageRank [4] have made it possible to search the long tail of large, linked document collections by keyword. Innovations such as faceted browsing [8] and collaborative filtering [7] have made it faster and easier to search within more structured domains such as shopping catalogues. Web queries are sensitive to changes in content over time, and searchers have an increasing supply of tools to help them find updates on webpages they revisit [1]. In labor and human computation markets, search tools are still evolving and it remains to be determined what features will prove most useful in overcoming search frictions.

Web search analysis has shown that the first result on the first page of search results has by far the highest click-through rate [16]. In a market model where parties have full information, there would be no need for search technologies—no purely "positional effects"—all parties would instantly and costlessly find their best options. In reality, search technologies are likely to have strong effects on the buyer-seller matches that are made. This point is evident by the fact that companies are willing to spend billions of dollars to have advertisements placed in prime areas of the screen [5].

While there are similarities between many kinds of online search activities, we must be careful in generalizing. Search behavior in one domain does not necessarily carry over to other domains. In this paper we focus solely on the domain of workers searching for HITs on MTurk, using web search behavior as a guide to what features may be important.

3. MTURK SEARCH FEATURES

At any given time, MTurk will have on the order of 100, 000 HITs available. HITs are arranged in HIT groups. All HITs posted by the same person with the same featues (title, reward, description, etc) are listed together in a HIT group to avoid repetition in the list of available HITs. HIT groups are presented much like traditional web search engine results, with 10 HIT groups listed on each page of search results. By default, the list is sorted by "HITs Available (most first)." The default view gives seven pieces of information:

1. Title (e.g., "Choose the best category for this product")
2. Requester (e.g., "Dolores Labs")
3. HIT Expiration Date (e.g., "Jan 23, 2011 (38 weeks)")
4. Time Allotted (e.g., "60 minutes")
5. Reward (e.g., "$0.02")
6. HITs Available (e.g., 17110)
7. Required qualifications (if any)

By clicking on a HIT title, the display expands to show three additional fields:

1. Description (e.g., "Assign a product category to this product")
2. Keywords (e.g., categorize, product, kids)
3. Qualifications. (e.g., "Location is US")

The MTurk interface also offers several search features prominently positioned at the top of the results page, including a keyword search and a minimum reward search. Additionally, HITs can be sorted in either ascending or descending order by six categories:

1. HIT Creation Date (***newest*** or ***oldest***)
2. HITs Available (***most*** or ***fewest***) (i.e., how many subtasks may be performed)

3. Reward Amount (***highest*** or ***lowest***)

4. Expiration Date (***soonest*** or ***latest***)

5. Title (***a-z*** or ***z-a***)

6. Time Allotted (***shortest*** or ***longest***)

Some sort results change more quickly than others. Sorting by most recent creation date will produce a very dynamic set of HIT groups. However, sorting by most HITs Available will produce a fairly static list because it takes a long time for a HIT group that contains, say, $17,000$ HITs to fall off the list.

On each worker's account summary page, MTurk suggests 10 particular HITs, usually with high rewards but no other apparent common characteristic. Other than this, we are unaware of any services to intelligently recommend HITs to workers. We presume that nearly all searching is done using the MTurk search interface. A Firefox extension called Turkopticon [15] helps workers avoid tasks posted by requesters that were reported by other users as being unfair in their view. This could affect workers' choices of which tasks to accept, but probably has little effect on the category workers choose to sort by and does not reorder or remove HITs from the results pages.

Requesters post tasks on MTurk and pay workers for acceptable work. There are two basic types of HITs that encompass most postings: (1) tasks such as image labeling, where workers are invited to perform as many various tasks as are available to be completed, and (2) tasks such as surveys, which require many workers but allow each worker to do the HIT only once. Type (2) HIT groups appear to workers as single HIT because there is only one HIT available to each worker. If a worker does not accept the HIT, it will remain on his list of available HITs until the total number of performances desired by the requester are completed. It often takes a substantial amount of time to achieve the throughput of workers necessary to complete type (2) HITs where the task only appears as a single available HIT to each worker. Because it is a small wonder they those HITs get done at all, we call them "1-HIT Wonders."

For HIT groups that allow individual workers to perform a single task multiple times (type (1)), the number of "HITs Available" is constantly updated. The number can increase in only two cases: first, when a HIT is returned by a worker unperformed and, second, in the rare event that more tasks are added to the HIT group before it is finished. Fortunately, this second case is easy to detect.

In this paper, we use HIT disappearance rates to perform a statistical analysis of search behavior for type (1) HIT groups containing multiple tasks available to all workers. We complement our analysis with the results of a survey where workers were asked about how they search for tasks such as 1-HIT Wonders that the scraping does not address.

4. METHOD A: INFERENCES FROM OBSERVED DATA

We want to determine whether the sort category, the page on which a HIT appears (page placement), and the position it occupies on that page (page position) affect the disappearance rate of a HIT. Our premise is that if any of these factors affect the disappearance rate of a HIT, then workers are using that search feature to find HITs. This approach makes use of what economists call a "natural experiment".

To understand it, consider the following thought experiment: Imagine that we could randomly move HITs to different pages and page positions within the various search categories without changing the attributes of the HITs. With this ability, we could determine the causal effects of HIT page placement and page position by observing how quickly a HIT is completed. Of course, we do not actually possess the ability to manipulate search results in this way for HITs we do not post, but we are able to observe the functioning of the market and potentially make a similar inference based on the scraped data.[1]

The problem with this approach, however, is that HIT characteristics that are likely to affect the popularity of a HIT should also affect its page placement and page position within the search results of the various sorting categories. In fact, the relationship between attributes and search results page placement and page position is the whole reason MTurk offers search features—if search results page placement and page position were unrelated to HIT characteristics that workers cared about, then any search category based on those characteristics would offer no value.

To deal with the problem of correlation between attributes and page placement and page position, we needed to make several assumptions and tailor our empirical approach to the nature of the domain. Before discussing our actual model, it will be helpful to introduce a conceptual model. We assume that the disappearance of a particular HIT during some interval of time is an unknown function of that HIT's attributes and its page placement and page position (together, its position in the list of search results):

$$\text{Disappearance of HIT } i = F(\text{position of } i, X_i)$$

where X_i are all the HIT attributes, market level variables, and time-of-day effects that might also affect disappearance rate. Our key research goal was to observe how manipulations of the position of i—while keeping X_i constant—affect our outcome measure, the disappearance of HITs (i.e., work getting done). We were tempted to think that we could include a sufficient number of HIT co-variates (e.g., reward, keywords, etc.) and control for the effects of X_i, but this is problematic because we cannot control for all factors.

Our goal of untangling causality was complicated by the fact that we cannot exogenously manipulate search results and the fact that there are HIT attributes for which we cannot control. Our approach was to acknowledge the existence of unobserved, HIT-specific idiosyncratic effects but to assume that those effects are constant over time for a particular HIT group. With this assumption, we relied upon movement in HIT position within a sort category as a "natural experiment." When we observed a HIT moving from one position to another, and then observed that the HIT disappeared more quickly in the new position, we attributed the difference in apparent popularity to the new position, and not to some underlying change in the nature of the HIT.

4.1 Data Collection

There are 12 ways in which MTurk allows workers to sort HITs—six categories, each of which can be sorted in ascend-

[1] In Method B, we will actually manipulate our HITs position in the search results (albeit by modifying the HIT attributes).

ing or descending order. We scraped the first three pages of search results for each of the 12 sorting methods. Each page of results lists 10 hits, which meant that we scraped 360 HITs in each iteration. We scraped each of the 36 pages approximately every 30 seconds—just under the throttling rate. Although we ran the scraper for four days, due to our own data processing constraints, in this analysis we used data collected in a 32-hour window beginning on Thursday, April 29, 2010, 20:37:05 GMT and ending on Saturday, May 1, 2010, 04:45:45 GMT. The data contains $997,322$ observations. Because of the high frequency of our scraping, each posted HIT is observed many times in the data: although the data contains nearly $1,000,000$ observations, we only observed $2,040$ unique HITs, with each HIT observed, on average, a little more than 480 times.

For each HIT, we recorded the 10 pieces of information that the MTurk interface offers (seven default and three additional features—see Section 3), as well as the sort category used to find the HIT, the page placement of the HIT (1, 2 or 3), and the page position of the HIT (first through tenth).

4.1.1 Measuring HIT disappearance

Formally, let s index the sort category, let g index the groups of observations of the same HIT (which occur because each HIT is observed many times), and let i index time-ordered observations within a HIT group. Let HITs be ordered from oldest to most recently scraped, such that $i+1$ was scraped after i. Let the number of HITs available for observation i be y_{igs}. The change is simply $\Delta y_{igs} = y_{(i+1)gs} - y_{igs}$.

Unfortunately, there are several challenges in treating this outcome measure as a direct measure of uptake. First, requesters can add and delete tasks in a HIT group over time. If a requester deletes a large number of tasks, then a regression might incorrectly lead us to believe that there was something very attractive to workers about the page placement and/or page position of the HIT. Second, for 1-HIT Wonders, the HITs-available measure might not change even though the HIT is very popular. Finally, the absolute change is "censored" in that the number of HITs available cannot go below zero, which means that HIT groups with many tasks available are able to show greater changes than HIT groups with few tasks available. For these reasons, we make our outcome variable an indicator for a drop in HITs available, regardless of the magnitude of the change:

$$Y_{igs} = 1 \cdot \{\Delta y_{ijs} < 0\}$$

4.2 Econometric set-up

To implement our conceptual model, we assumed a linear regression model in which the expected outcome is a linear function of a series of variables. Obviously, many factors determine whether or not a HIT disappears from search results. Page placement, page position, and sort category certainly matter, but so do variables that are hard to observe and which we cannot easily include in a regression. For example, a comparatively high-paying, short task that looks fun will probably be accepted more quickly than one that is low-paying, long, and tedious. In creating our model, we sought to separate the desirability of a task from the page placement and page position it occupies in the search results of a particular sorting category.

4.2.1 Choosing a model

Rather than simply introduce a number of controls that will invariably miss factors that are unmeasured, we included a HIT group-specific control in the regressions called a "group random effect." Each HIT group was modeled as having its own individual level of attractiveness that stays constant over time. In this way, all factors that have a static effect on HIT attractiveness were controlled for, regardless of whether or not we are able to observe the factors that determine a HIT groups attractiveness.

With a group-specific random effect, we eliminate one of they key problems that would arise from simply using a pooled model. For example, a fun HIT that always stays on the third page of results sorted by highest reward might cause us to erroneously believe that the third page is very important, when the case is simply that one HIT group is disappearing quickly.

One necessary word on the nomenclature of these two models: when dealing with data that has a natural group structure, a model with a group specific-effect is called a "random effects" model or "multilevel" model; a model that ignores the group structure and does not include any group-specific effects is called a "pooled" model. Both types are linear regression models. While our group-specific effect model "controls" for group factors, we do not actually include a dummy variable for each group (which is called the "fixed effects" model). Rather, we start with a prior assumption that each effect is a random draw from a normal distribution (the random effects model) [6]. As the number of observations increases, this fixed effects/random effects distinction becomes less important.

4.2.2 Model

The group-specific random effects model is as follows: for an observation i of group g, we estimate

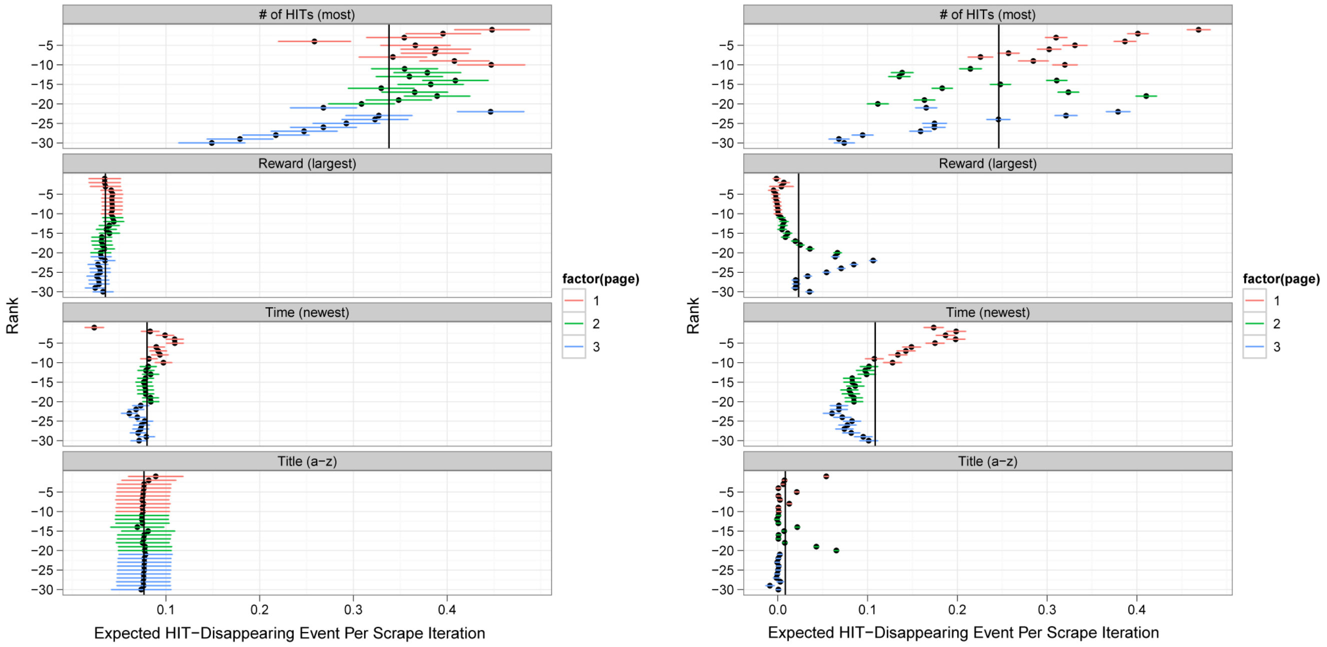
$$Y_{igs} = \sum_{r=1}^{30} \beta_s^r x_{igs}^r + \gamma_g + \tau_{H(i,g)} + \epsilon \qquad (1)$$

where $x_{ig}^r = 1$ if the HIT i is at position r (and $x_{ig}^r = 0$ otherwise) and where $\gamma_g \sim N(0, \sigma_\gamma^2)$ is the group-specific random effect and $\tau_{H(i)} \sim N(0, \sigma_\tau^2)$ is a time-of-day random effect. The pooled model ignores this grouped structure and imposes the constraint that there is no group effect and $\gamma_g = \gamma = 0$.

4.3 Results

We applied our models to four of MTurk's 12 sorting options: *newest* HITs, *most available* HITs, *highest reward*, and *title a-z*. The others—including *shortest* time allotted and *latest expiration*—tend to produce 1-HIT Wonders and don't seem like natural ways to sort HITs.

In Figure 1, the collection of coefficients, β_s^r, from Equation 1 are plotted for both the group-specific random effects model (panel (a)) and the pooled model (panel (b)), with error bands two standard errors wide. The four sorting options are listed at the top of the figure: *newest*, *most*, *highest reward*, *a-z*. The points are the coefficients (with standard error bars) for each page and position on that page, ranging from one to 30 (positions are displayed as negative numbers to preserve the spatial ordering). The coefficients $\hat{\beta}_s^r$ are interpretable as the probability that a HIT occupying position r decremented by one or more HITs during the scraping



(a) Group-specific random effects model

(b) Pooled model

Figure 1: Effects of position in sort category on HIT disappearance. Pages are illustrated with color. Horizonal lines represent one standard error on either side.

interval. Our premise was that sort categories, page placements, and page positions with higher probability of the HIT decrement are those that workers search by.

4.4 Group random effects model

Under the assumptions of the group-specific random effects model, we interpret the position coefficients as the probability that a HIT occupying a particular page position will have a disappearance event during the scraping interval. The only sort category that shows strong positional effects is *most*, with similar rates for the first and second results pages, and then a strong drop-off on the third page.

From this, we conclude that workers actively sort by the *most* available HITs and look at results on the first two pages of search results. On the third page of *most* available HITs, the average coefficient is high, but position effects seem to matter. We conjecture this to mean that the further down the third page the HIT is located, the more likely workers are to abandon this search strategy. However, more investigation is needed in order to confirm this interpretation.

The page placements and page positions generated by the *highest reward* sorting category have no apparent effect on search behavior. The overall levels of disappearance for HITs among the search results are low, which is to be expected given the relative unpopularity of high-reward HITs. Also unsurprisingly, the page placements and page positions generated by the *title a-z* sorting category have no apparent effect on search behavior.

4.5 Pooled model

The group-specific random effects results shown in panel (a) of Figure 1 are more credible than those of the pooled model in panel (b). Nevertheless, panel (b) coefficients are far more precise. This is because panel (a) coefficients are determined solely by within-group movements in search result position. For sorting categories that do not show much movement—such as *title*—the panel (a) estimates are comparatively imprecise. However, the panel (b) estimates only capture the innate attractiveness of whatever HIT was occupying a particular position. For example, at $r = 20$ in panel (b), *title*, we see a strong statistically significant effect, but this is presumably only a feature of whatever HIT was occupying that position.

This "stationary HIT" bias issue also arises in panel (b), *reward*: on the first page of search results for *highest*, there is almost zero probability of the HIT disappearing. It may seem surprising that high-reward tasks aren't being taken, but this fact can be readily explained. Most MTurk HITs have a reward between \$0.02 and \$1.00. However, there are some HITs that get posted for larger sums, on the order of \$10.00. There are very few such HITs and they tend to be unpopular, so they stay on the *highest reward* page for a long time without disappearing. We do see that on the second and third results pages, the probability of non-zero disappearance rate goes up.

4.5.1 Searching by newest HITs?

According to panel (a), the search results position within *newest* HITs is generally either irrelevant, or, in the case of position 1, page 1, is actually very harmful! This *is* surprising. Sorting by the *newest* HITs seems to be a very good way to find fresh and interesting tasks. Further, as our survey results (described below) show, most workers report searching for HITs based *newest*.

Interestingly, *newest* is a sort category where we would expect the inclusion of group effects to be irrelevant, since position in the *newest* sort category should be essentially mechanical—a HIT group is posted, starts off at the first position, and then moves down the list as newer HIT groups are posted. Yet, in comparing *newest* across the two panels, the effects of position are radically different: in panel (b), we see that position has a strong effect on uptake, with the predicted pattern, while in panel (a), there appears to be little of no effect. To summarize, the pooled model suggests strong effects, while the random effects model suggests no effects.

4.5.2 *Discrepancy*

We hypothesized that there were no positional effects in the group-specific random effects model because certain requesters actively game the system by automatically re-posting their HIT groups in order to keep them near the top of *newest*. This hypothesis reconciles two surprising findings.

First, gaming explains the absence of *newest* effects in panel (a): the benefits to being in a certain position are captured as part of the group effect. To see why, consider a requester that can always keep his HIT at position 1. No matter how valuable position 1 is in attracting workers, all of this effect will be (wrongly) attributed to that particular HIT's group-specific effect.

Second, gaming explains why position 1 appears to offer far worse performance in panel (a) (and why there is a generally increasing trend in coefficient size until position 4). A very large share of the HITs observed at position 1 in the data came from gaming requesters. A still large but relatively smaller share of the HITs observed at position 2 came from gaming requesters who were trying to get their HIT groups back into position 1, and so on. Because all the purely positional benefits get loaded onto the group effects of gaming requestors, positions more likely to be occupied by gaming requestors will appear to offer smaller positional benefits.

Gaming not only reconciles the data—it is also directly observable. We identified the HITs that had 75 or more observations in position 1 of *newest*. In Figure 2, the position $(1-30)$ is plotted over time with the HIT group ID and title posted above. It was immediately clear that some HITs are updated automatically so that they occupy a position near the top of the search results: until day 0.5, the HIT groups in the second and third positions were "competing" for the top spot; around day 0.9, they faced competition from a third HIT group (first panel) that managed to push them down the list.

Under these gaming circumstances, the group-specific random effects model is inappropriate, as the positions are not randomly assigned. For this reason, the pooled model probably offers a clearer picture of the positional effects, though the coefficient estimates are certainly biased because they incorporate the nature of HITs occupying position one, which we know is from gaming.

We conclude that the pooled regression results are the correct analysis for the *newest* HITs category. The pooled regression results show that workers are actively searching by *newest* HITs and focus on the first page of results.

5. METHOD B: WORKER SURVEY

Another way to study how workers search for HITs is to ask the workers directly, by conducting a survey on MTurk. Although this approach is more expensive, more obtrusive, and smaller-scale than scraping the MTurk web site, it has two advantages. First, surveying complements the scraping technique. Scraping is better suited to tracking postings of large numbers of type (1) HITs whose tasks can be performed multiple times by any worker, so an anonymous scraper can observe their acceptance. Surveys, by contrast, are necessarily 1-HIT Wonders, offered at most once to every worker, and MTurk offers no way for a scraper to observe how many workers have accepted the offer. Since a substantial use of MTurk involves 1-HIT Wonders for online experimentation and surveys of other kinds, we need a method for exploring search behavior on this side of the market. Second, surveys can provide more detailed and subjective explanations for worker behavior than aggregate statistics.

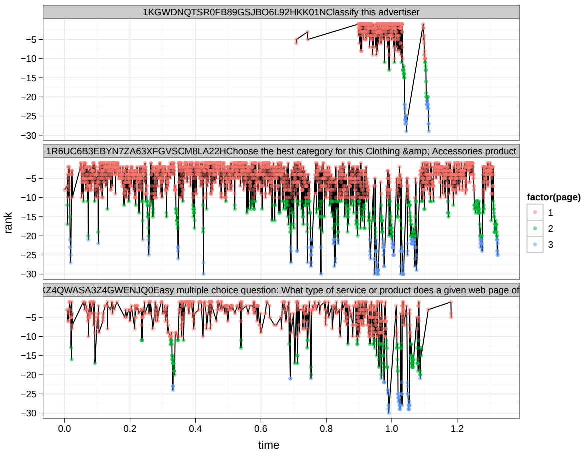


Figure 2: Position of three HITs over time sorted by *newest*. (Time measured in days)

This section presents the results of a survey of roughly 250 workers about how they sort and filter HITs. Since MTurk was used to conduct the survey, the characteristics of the HIT used to solicit survey respondents have an effect on the results. We explored this effect by posting the survey in several different ways in order to sample workers who exhibited different search behaviors. In the process, we observed how certain choices of HIT characteristics can make a HIT very hard to find, substantially reducing the response rate. Finally, the survey also collected free-form comments from workers about how they find good and bad HITs.

5.1 Procedure

The survey was designed to probe respondents' immediate search behavior—specifically, how respondents used MTurk's search interface to discover the survey HIT itself. The survey asked several questions: (1) which of the 12 sort categories they were presently using; (2) whether they were filtering by a keyword search or minimum price; and (3) what page number of the search results they found the survey HIT on. A final question asked the worker for free-form comments about how easy or hard it is to find HITs.

The survey was posted on MTurk in four ways, using different HIT characteristics for each posting to give it high and low positions in the various sorted search results. Aside from these manipulated parameters, the four postings were identical (e.g. in title, description, preview, and actual task).

Best-case posting. In one posting, the parameters of the HIT were chosen to place it among the top search results under each of the six primary sort categories. Thus the HIT would appear on the first page, or as close as possible, when HITs were sorted by *any* attribute, though only in one direction (ascending or descending). The goal of this posting was to capture as many workers as possible from each of the six chosen sort orders that the posting optimized.

The HIT was automatically one of the *newest*, at least at first, and had the *fewest* possible HITs available because it was only offered once to each worker. For reward, we chose the *least* amount: $0.01. For the *soonest* expiration date and the *shortest* time allotted, we chose 5 hours and 5 minutes, respectively. Finally, we optimized for alphabetical title order, *a-z*, by starting the title with a space character.

The best-case posting was also labeled with the keyword "survey", which we knew from pilot studies was frequently used by turkers to find desirable survey HITs. The other three postings had the same title and description as the best-case posting, but no "survey" keyword.

Worst-case posting. To test the impact of especially poor position on a HIT's reachability, we also posted the survey in a way that was *hard to find* as possible for workers using any sorting category. This was done by choosing parameter values near the median value of existing HITs on MTurk, so that the survey HIT would appear in the middle of the pack, as far as possible from the first page in both ascending and descending order by that attribute.

For example, at the time the survey was posted, the median number of HITs available was two. Postings with two HITs covered pages 55–65 (out of 100 pages of HITs requiring no qualifications) when the list was sorted by *fewest* HITs available, and pages 35–45 when sorted by *most* HITs. As a result, a worker sorting in either direction would have to click through at least 35 pages to reach any 2-HIT posting, which is highly unlikely. We therefore posted the survey with two independent HITs.

For the remaining attributes, we chose a reward amount of $0.05 and an expiration period of one week and allotted 60 minutes in order to position the HIT at least 20 pages deep among search results sorted by reward amount, creation date, and time allotted, in both ascending and descending order. The median title of all current HITs started with "Q,", so we started the survey title with the word "Question" to make it hard to find whether sorting by *a-z* or *z-a*.

Creation date, however, cannot be chosen directly. In particular, a HIT cannot be posted with a creation date in the past. Freshly posted HITs generally appear on the first page of the *newest* sort. In order to artificially age our survey HIT before making it available to workers, we initially posted a nonfunctional HIT, which presented a blank page when workers previewed it or tried to accept it. After six hours, when the HIT had fallen more than 30 pages deep in the *newest* sort order, the blank page was replaced by the actual survey, and workers who discovered it from that point on were able to complete it.

Newest-favored and a-z-favored posting. We performed two additional postings of the survey, which were intended to favor users of one sort order while discouraging other sort orders. The *newest-favored* posting was like the worst-case posting in all parameters except creation date. It appeared functional immediately, so users of the *newest* sort order would be most likely to find it. Similarly, the *a-z-favored* posting used worst-case choices for all its parameters except its title, which started with a space character so that it would appear first in *a-z* order.

All four postings were started on a weekday in May 2010, with their creation times staggered by 2 hours to reduce conflict between them. The best-case posting expired in 5 hours, while the other three postings continued to recruit workers for 7 days.

5.2 Results

Altogether, the four postings recruited 257 unique workers to the survey: 70 by the best-case posting (in only 5 hours), 58 by the worst-case posting, 59 by newest-favored, and 70 by a-z-favored (all over 7 days). Roughly 50 workers answered more than one survey posting, detected by comparing MTurk worker IDs. Only the first answer was kept for each worker.

Figure 3 shows the rate at which each posting recruited workers to the survey over the initial 24-hour period. The best-case posting shows the highest response rate, as expected, and the worst-case posting has the lowest. The newest-favored posting shows a pronounced knee, slowing down after roughly 1 hour, which we believe is due to being pushed off the early pages of the *newest* sort order by other HITs. Because the posting's parameters make it very hard to find by other sort orders, it subsequently grows similarly to worst-case. The a-z-favored posting, by contrast, recruits workers steadily, because it remains on the first page of the a-z sort order throughout its lifetime. These results show that HIT parameters that affect sorting and filtering have a strong impact on the response rate to a HIT. All four postings were identical tasks with identical descriptions posted very close in time, and the best-case posting actually offered the lowest reward ($0.01 compared to $0.05 for the others), but its favorable sort position recruited workers roughly 30 times faster than the worst-case posting.

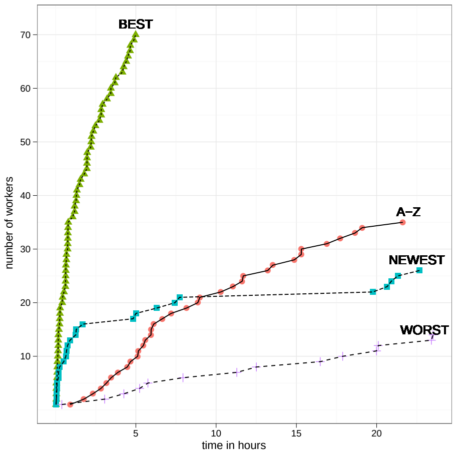


Figure 3: Number of workers responding to each survey posting within the first 24 hours of posting (time measured in days)

Figure 4 shows the responses to the survey question about sorting category, grouped by the posting that elicited the response. The best-case posting recruited most of its workers from the *newest* sort order, reinforcing the importance of this sort order for 1-HIT wonders like a survey. The *newest* sorting category also dominated the responses to the newest-favored posting, as expected, and *a-z* appeared stongest in the a-z-favored posting. Strong appearance of *most* HITs was a surprise, however, since each posting had only 1 or 2 HITs.

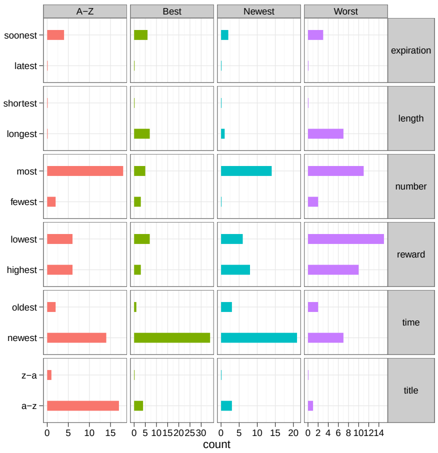


Figure 4: Number of workers who reported using each sort order to find each of the four postings.

For the survey question about reward amount filtering, roughly 65% of the survey respondents reported that they had not set any minimum amount when they found the survey HIT. Surprisingly, however, 9% reported that they had used a reward filter that was *greater* than the reward offered by the survey HIT. For example, even though the best-case posting offered only $0.01, ten respondents reported that they found the survey while searching for HITs that paid $0.05 or more. We followed up by sending email (through MTurk) to several of these workers, and found that the problem can be traced to a usability bug in MTurk: *"Once we submit a hit MTurk takes us back to all hits available, 1st page with no $ criteria."* In other words, a filter is discarded as soon as the worker performs a single task. Several other workers mentioned similar problems in their free-form comments.

Similarly, roughly 20% of the workers reported using a keyword filter when they found the survey, but half of these reported using keywords that were not actually included in the survey HIT, suggesting that the filter they thought they were using was no longer in effect. 21 workers reported using the ”survey” keyword, of whom 16 were responding to the best-case posting, which actually did include that keyword.

Figure 5 shows the distribution of result page numbers on which workers reported finding the survey HIT. Roughly half found it on the first or second page of results, as might be expected. Yet a substantial fraction (25%) reported finding the survey beyond page 10. Because of the way the survey was posted, particularly the worst-case posting, workers would not have been likely to find it anywhere else. Yet it is still surprising that some workers are willing to drill dozens of pages deep in order to find tasks.

The free-form comment question generated a variety of feedback and ideas. Many respondents reported that they would like to be able to search for or filter out postings from particular requesters. Other suggestions included the ability to group HITs by type, and to offer worker-provided ratings of HITs or requesters. Several workers specifically mentioned the burden of clicking through multiple pages to find good HITs:

"Scrolling through all 8 pages to find HITs can be a little tiring."

"I find it easier to find hits on mechanical turk if I search for the newest tasks created first. If I don't find anything up until page 10 then I refresh the page and start over otherwise it becomes too hard to find tasks."

Several workers pointed out a usability bug in MTurk that makes this problem even worse:

"It is difficult to move through the pages to find a good HIT because as soon as you finish a HIT, it automatically sends you back to page 1. If you are on page 25, it takes so much time to get back to page 25 (and past there)."

"Please keep an option to jump to a page directly without opening previous pages."

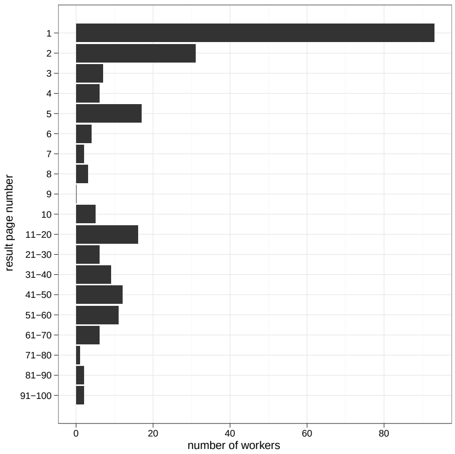


Figure 5: Histogram of result page numbers on which workers found the survey HIT. Most found it on page 1, but many were willing to drill beyond page 10.

6. CONCLUSION

In this paper, we studied the search behavior of MTurk workers using two different methods. In our first method, we scraped pages of available tasks at a very high rate and used the HIT disappearance rates to see if certain sorting methods resulted in greater task acceptance.

Because this method does not accurately measure all types of HITs, we also posted a survey on MTurk asking workers how they search for our survey task. We posted the task with specially chosen posting parameters complementary to those of the HITs we gathered information on using the scraper.

Both methods show that workers tend to sort by *newest* HITs. The scraping revealed that workers also sort by *most* HITs and focus mainly on the first two pages of search results but ignore the positions of the HITs on the page. The survey data confirms that even though *newest* and *most* are the most popular search strategies, the other sort categories are also used—even relatively obscure ones such as *title a-z* and *oldest* HITs, but to a much lesser extent. The survey confirms that the first two pages of the search results are most important, but also shows evidence that some workers are willing to wade fairly deep into search results pages and to periodically return to the *newest* HITs results.

7. FUTURE WORK

Method A does not account for keyword searching or sorting by required qualifications. The Method A empirical results provide insight into how workers search, but it would useful to develop a structural model of search behavior that incorporated the quantity of work being done could make predictions.

8. ACKNOWLEDGMENTS

John thanks the NSF-IGERT Multidisciplinary Program in Inequality & Social Policy for generous financial support (Grant No. 0333403). Thanks to Marc Grimson for help running the survey and to Greg Little and Jaime Teevan for feedback. Thanks to Robin Y. Horton, Carolyn Yerkes and three anonymous reviewers for very helpful comments. All plots were made using `ggplot2` [18] and all multilevel models were fit using `lme4`[3].

9. REFERENCES

[1] E. Adar, M. Dontcheva, J. Fogarty, and D. Weld. Zoetrope: interacting with the ephemeral web. In *Proceedings of the 21st annual ACM symposium on User interface software and technology*, pages 239–248. ACM, 2008.

[2] D. Autor. Wiring the labor market. *Journal of Economic Perspectives*, 15(1):25–40, 2001.

[3] D. Bates and D. Sarkar. lme4: Linear mixed-effects models using S4 classes. *URL http://CRAN. R-project. org/package= lme4, R package version 0.999375-28*, 2008.

[4] S. Brin, L. Page, R. Motwami, and T. Winograd. The PageRank citation ranking: bringing order to the web. In *Proceedings of ASIS'98*, pages 161–172, 1998.

[5] B. Edelman, M. Ostrovsky, and M. Schwarz. Internet advertising and the generalized second-price auction: Selling billions of dollars worth of keywords. *The American Economic Review*, pages 242–259, 2007.

[6] A. Gelman and J. Hill. *Data analysis using regression and multilevel/hierarchical models*. Cambridge University Press Cambridge, 2007.

[7] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry. Using collaborative filtering to weave an information tapestry. *Commun. ACM*, 35(12):61–70, 1992.

[8] M. Hearst, A. Elliott, J. English, R. Sinha, K. Swearingen, and K. Yee. Finding the flow in web site search. *Communications of the ACM*, 45(9):49, 2002.

[9] J. Horton and L. Chilton. The labor economics of paid crowdsourcing. *Proceedings of the 11th ACM Conference on Electronic Commerce 2010 (forthcoming)*, 2010.

[10] J. J. Horton, D. Rand, and R. J. Zeckhauser. The Online Laboratory: Conducting Experiments in a Real Labor Market. *NBER Working Paper w15961*, 2010.

[11] B. A. Huberman, D. Romero, and F. Wu. Crowdsourcing, attention and productivity. *Journal of Information Science (in press)*, 2009.

[12] P. Ipeirotis. Demographics of mechanical turk. *New York University Working Paper*, 2010.

[13] E. Law, L. Von Ahn, R. Dannenberg, and M. Crawford. Tagatune: A game for music and sound annotation. In *International Conference on Music Information Retrieval (ISMIR'07)*, pages 361–364. Citeseer, 2003.

[14] W. Mason and D. J. Watts. Financial incentives and the 'performance of crowds'. In *Proc. ACM SIGKDD Workshop on Human Computation (HCOMP)*, 2009.

[15] M. Silberman, J. Ross, L. Irani, and B. Tomlinson. Sellers' problems in human computation markets. 2010.

[16] A. Spink and J. Xu. Selected results from a large study of Web searching: the Excite study. *Inform. Resear.—Int. Electron. J*, 6(1), 2000.

[17] L. Von Ahn and L. Dabbish. Labeling images with a computer game. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 319–326. ACM, 2004.

[18] H. Wickham. ggplot2: An implementation of the grammar of graphics. *R package version 0.7, URL: http://CRAN.R-project.org/package=ggplot2*, 2008.