Crowd-Workers: Aggregating Information Across Turkers To Help Them Find Higher Paying Work

Chris Callison-Burch
University of Pennsylvania
ccb@upenn.edu

Abstract

The Mechanical Turk crowdsourcing platform currently fails to provide the most basic piece of information to enable workers to make informed decisions about which tasks to undertake: what is the expected hourly pay? Mechanical Turk advertises a reward amount per assignment, but does not give any indication of how long each assignment will take. We have developed a browser plugin that tracks the length of time it takes to complete a task, and a web service that aggregates the information across many workers. Crowd-Workers.com allows workers to discovery higher paying work by sorting tasks by estimated hourly rate.

Extended Abstract

Crowdsourcing has been enabled by online labor markets like Amazon Mechanical Turk (MTurk), CrowdFlower, and oDesk. The most prominent of these is MTurk, where Requesters advertise work in the form of Human Intelligence Tasks (HITs), and workers (often referred to as Turkers), can browse the listings choose which tasks they would like to work on. MTurk is ostensibly a marketplace, where generous, high paying Requesters successfully compete for Turkers' attention over unreliable or lower paying Requesters. However, several design choices in the MTurk platform means that it is hard for workers to identify good Requesters, and to find high paying work (Silberman, Irani, and Ross 2010; Silberman et al. 2010). In particular:

- MTurk allows worker to search for HITs based on reward amount on a price-per-item basis, with no way of estimating an hourly wage.[1]

- MTurk's reputation system is asymmetric. Workers are ranked and filtered based on their past performance, but it's not possible to sort Requesters based on how quickly they pay or filter them based on their rejection rates.

- When a Requester rejects work a Turker's work, the worker is not paid, and there is no mechanism for appeal or redress.

[1]MTurk does compute and display an effective hourly rate, but only displays it to the Requester who posted the task.
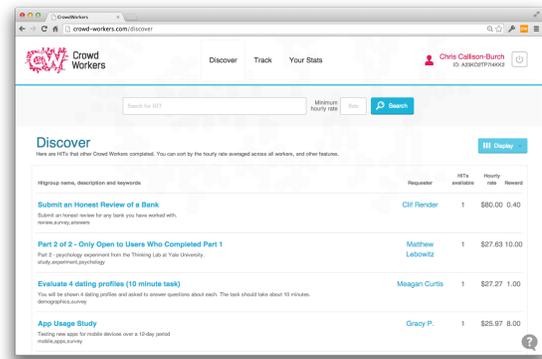

Figure 1: The crowd-workers.com web service allows users to sort HITs based on hourly rate. The hourly rate is estimated by tracking how long it took other workers to complete the task (using our browser extension), along with the reward amount (which MTurk makes available).

- Since Turkers' ability to continue working is based on their past approval rating, they are often quite conservative about taking on new tasks.

Since the MTurk platform falls short, Turkers often turn to external resources to improve their experience on MTurk and to optimize their productivity (Martin et al. 2014). They use forums like Turker Nation to discuss Requesters and work habits, tools like Reddit to share HITs worth working on, and browser plugins like TurkOpticon to share star ratings of Requesters (Irani and Silberman 2013).

In this work, we adopt one of the design principles that has been suggested to make online labor fairer and more transparent: disclosing the effective hourly rates of tasks (Bederson and Quinn 2011). We have designed a Chrome plugin that tracks when a works accepts and submits a HIT. This information is combined with the reward amount, and aggregated over many workers. Our web service, crowd-workers.com, lists allows workers to discover higher paying work by sorting tasks by estimated hourly rate. (Figure 1).

The Crowd-Workers browser extension currently computes aggregate information about hourly rates. We plan to

TurkOpticon's qualitative attributes	CrowdWorker's quantitative equivalents
Generosity: How well has this requester paid for the amount of time their HITs take?	**Average hourly rate:** What is the average hourly rate that other Turker make when they do this requester's HITs?
Promptness: How promptly has this requester approved your work and paid?	**Expected time to payment:** On average, how much time elapses between submitting work to this Requester and receiving payment?
Fairness: How fair has this requester been in approving or rejecting your work?	**Approval/rejection rates:** What percent of assignments does this Requester approve?
Communicativity: How responsive has this requester been to communications or concerns you have raised?	**Reasons for rejection:** Archive of all of the reasons for Workers being rejected or blocked by this Requester.

Table 1: TurkOpticon provides a mechanism for workers to give start ratings requesters along four qualitative dimensions. Our CrowdWorkers plugin aims to replace these qualitative ratings with quantitative equivalents. Currently it computes average hourly rates.

extend it to track additional information about requesters, including speed of payment, and approval/rejection rates. The goal is to replace TurkOpticon's subjective ratings with more quantitative information. (Table 1).

We have conducted a pilot study that tracked more than 77,000 HIT submissions from 110 Turkers. 50 of the subjects were paid participants who were recruited on mturk.com. Up to 26 of the subjects were undergraduates from the University of Pennsylvania's "Crowdsourcing and Human Computation" course (http://crowdsourcing-class.org). The remainder of the subjects found our tool on their own through word of mouth. 65 of the subjects completed more than 100 HITs. Figure 2 plots the average hourly earnings for each of those workers. The average hourly earning across all workers was \$9.56 per hour, and the median hourly earning was \$7.85 per hour.

In the future, we plan to analyze whether our web service helps workers improve their productivity. We believe that the data that we gather will also provide insight into the Mechanical Turk marketplace (Ipeirotis 2010a), how task pricing and financial incentives affect completion time (Mason and Watts 2009; Singer and Mittal 2011), and how earnings and work patterns vary by worker demographics (Ipeirotis 2010b).
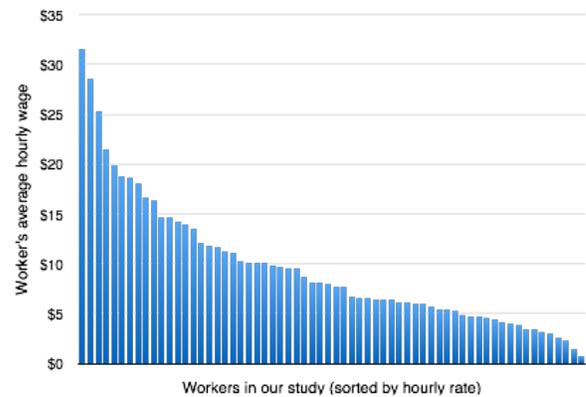
Acknowledgements

Figure 2: The hourly earnings of the 65 Turkers in our pilot study who completed more than 100 HITs using the Crowd-Workers browser plugin.

References

Bederson, B. B., and Quinn, A. J. 2011. Web workers unite! addressing challenges of online laborers. In *CHI '11 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '11, 97–106.

Ipeirotis, P. G. 2010a. Analyzing the Mechanical Turk marketplace. In *ACM XRDS*.

Ipeirotis, P. G. 2010b. Demographics of Mechanical Turk. Technical Report Working paper CeDER-10-01, New York University, Stern School of Business.

Irani, L. C., and Silberman, M. S. 2013. Turkopticon: Interrupting worker invisibility in Amazon Mechanical Turk. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, 611–620.

Martin, D.; Hanrahan, B. V.; O'Neill, J.; and Gupta, N. 2014. Being a Turker. In *Proceedings of The 17th ACM Conference on Computer Supported Cooperative Work and Social Computing (CSCW)*.

Mason, W., and Watts, D. J. 2009. Financial incentives and the "performance of crowds". In *Proceedings of the Workshop on Human Computation at the International Conference on Knowledge Discovery and Data Mining (KDD-HCOMP '09)*.

Silberman, M. S.; Ross, J.; Irani, L.; and Tomlinson, B. 2010. Sellers' problems in human computation markets. In *Proceedings of the ACM SIGKDD Workshop on Human Computation*, HCOMP '10, 18–21.

Silberman, M. S.; Irani, L.; and Ross, J. 2010. Ethics and tactics of professional crowdwork. *XRDS* 17(2):39–43.

Singer, Y., and Mittal, M. 2011. Pricing mechanisms for online labor markets. In *Third AAAI Human Computation Workshop (HCOMP'11)*.