

Modeling and Exploration of Crowdsourcing Micro-Tasks Execution

Pavel Kucherbaev, Florian Daniel*, Stefano Tranquillini, Maurizio Marchese

University of Trento, DISI, Via Sommarive 9, Povo, Italy, 38123
Tomsk Polytechnic University, Lenin Avenue 30, Tomsk, Russia, 634050
{lastname}@disi.unitn.it

Introduction

There are two key issues which are critical for a person crowdsourcing work: quality control and timeliness. While the first topic is extensively covered in the literature (Al-lahbakhsh *et al.* 2013; Ipeirotis *et al.* 2010), the timeliness is still not very well analyzed. In our past experiments on CrowdFlower we noticed that 90% of task instances were completed in only 10 minutes, the rest 10% could take hours to be completed. There were also extra peaks in performance on the 30th and 60th minutes since the task publication. From the observation of these patterns we wanted to understand better how the crowd workers execute tasks, to try to describe this behavior mathematically and to identify ways to speed up the tasks execution.

A *requester* can crowdsource work by publishing a *task*. The requester can upload a *dataset* for the task and ask one or more crowd *workers* who satisfy an optional *preselection* strategy to process each *data unit* from the dataset. For each data unit corresponding *assignments* are created that bind together the task and the data units, as well as the workers that accept the assignment and their eventual results. Each assignment can be in one of 3 possible states: *to be assigned* – no worker joined the assignment yet, *started* – a worker is working on the assignment, *finished* – the worker submitted results for the assignment. The assignment *duration* is the time from when the assignment is started to when it is finished. If an assignment is started but not finished within a given timeout, the assignment *expires* and becomes again available to other workers. The worker of the expired assignment is no longer able to submit results and is not rewarded.

Exploration

In order to explore the possible reasons for the execution patterns we observed on CrowdFlower we conducted a task where workers need to transcribe and fill up four textual fields (the company name, the address, the date of the purchase, and the total amount) for a given receipt photo.

Qualitative exploration

For the qualitative exploration we varied the reward amount and the workers preselection in the next 3 conditions: 1) re-

ward = \$0.10, preselection = anybody, 2) reward = \$0.10, preselection = masters only (on CrowdFlower these workers are called level 3 and have accuracy level above 85%, on MTURK these workers are called masters and have high accuracy level which is not disclosed) and 3) reward = \$0.01, preselection = anybody. The experiment was conducted in parallel on CrowdFlower and MTURK, where different conditions were launched sequentially.

The cumulative executions are shown in Figure 1, where the blue lines represent assignments started and the black lines represent assignments completed. The detailed assignment executions are presented in Figure 2. Here the *y*-axis shows different data units with receipt images and *x*-axis shows the time since the task launch in minutes. To generate these graphs we have implemented an R-script¹, which is publicly available along with the experiments results data for the benefit of the crowdsourcing research community.

Findings We identified, that on CrowdFlower there is a strong parallelism in tasks execution and the number of workers involved is close to the number of units in this task (e.g. 39/40). The workers here start their assignments in the first several seconds (e.g. 9.50 seconds) since the task is published. The execution parallelism on MTURK is weak as the number of workers here is smaller (e.g. 7/40). While it is not technically possible for a single worker to start several assignments at the same time on CrowdFlower, on MTURK some workers do reserve several data units for themselves and work on them sequentially. On CrowdFlower when a worker abandons an assignment, it is released to other workers only in 30 minutes (assignment duration time limit defined by CrowdFlower), which results in assignment start time outliers with 30 minutes period. Sometimes it can take a worker a significant amount of time to finish an assignment, which could be caused by some distraction (e.g. a phone call).

Quantitative exploration

We found that on CrowdFlower in condition 1 (Figure 1), where we pay \$0.10, workers pick the units faster, than in condition 3, where we pay \$0.01. Keeping this in mind and the prior work of other researchers we further investigate

Copyright © 2015, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

¹<http://bit.ly/1Jpjed>

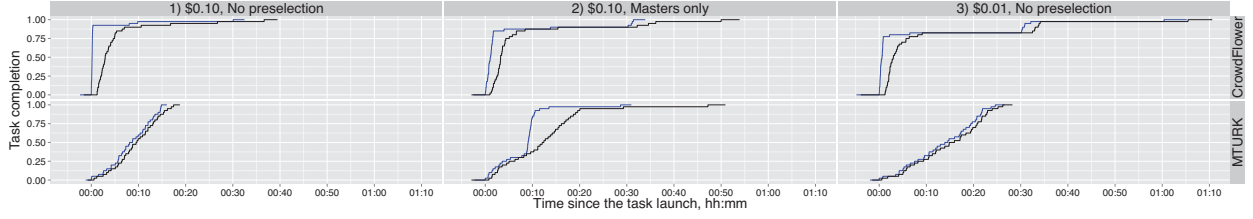


Figure 1: Cumulative assignments start (blue) and completion (black)



Figure 2: Detailed assignments start and completion

how and if the assignments duration time and assignments start time depend on reward amount and units amount.

We conducted the same task we used in the qualitative exploration phase, but now varying 2 tasks properties: the reward amount and the units amount. From the tasks listing page on CrowdFlower we found that the offered rewards lay in a range from \$0.01 to \$0.20 and units amount are in a range from 1 to 70. In our experiment we independently varied the reward amount from \$0.01 to \$0.25 with a step of \$0.03 (9 conditions with fixed 20 units), and the units amount from 10 to 100 with a step of 10 (10 conditions with a fixed reward amount of \$0.01). The condition with a reward \$0.01 and units amount 20 is common and was run only once, so in total we had 18 conditions. In order to have more representative data we ran our experiment 3 times. We spent around \$130 for all the conditions.

Findings We unexpectedly identified that on CrowdFlower higher rewards do not cause lower assignments start time or lower assignments duration. The fact that for workers on CrowdFlower “easy to complete” factor is more important than the reward amount (Gadiraju *et al.* 2014), could describe our finding. There is a weak correlation between the data units amount and the assignments start time. We also identified a linear correlation between the start time and the number of units of the task.

Future Directions

In future we want to conduct similar experiments but with a wide range of task types on CrowdFlower, MTURK and other platforms where it is possible to collect assignments start and end times data. This could help us to finally have a complete view on how various task properties (e.g. reward amount) affect the performance. Based on the findings of our study, next we intend to develop a runtime task monitoring and a management component that will allow us to identify assignment start time outliers and to solve them. Such an approach potentially should help to decrease the overall task execution time and to make it more predictable.

References

- Mohammad Allahbakhsh, Boualem Benatallah, Aleksandar Ignjatovic, Hamid Reza Motahari-Nezhad, Elisa Bertino, and Shahram Dustdar. Quality control in crowdsourcing systems: issues and directions. *IEEE IC*, 17(2):76–81, 2013.
- Ujwal Gadiraju, Ricardo Kawase, and Stefan Dietze. A taxonomy of microtasks on the web. In *HT '14*, pages 218–223. ACM, 2014.
- Panagiotis G. Ipeirotis, Foster Provost, and Jing Wang. Quality management on amazon mechanical turk. In *HCOMP '10*, pages 64–67. ACM, 2010.