Human Computation and Crowdsourcing: Works in Progress and Demonstration Abstracts
An Adjunct to the Proceedings of the Second AAAI Conference on Human Computation and Crowdsourcing

Post It or Not: Viewership Based Posting of Crowdsourced Tasks

Pallavi Manohar, Deepthi Chander
Koustuv Dasgupta, Sakyajit Bhattacharya
Xerox Research Centre India
{pallavi.manohar, deepthi.chander, koustuv.
dasgupta, sakyajit.bhattacharya}@xerox.com

L. Elisa Celis
École Polytechnique Fédérale de Lausanne (EPFL)
elisa.celis@epfl.ch

Abstract

We propose an online scheduling algorithm for posting crowdsourcing tasks which maximizes a novel metric called *task viewership*. This metric is computed using stochastic model based on coverage process and it measures the likelihood that a task is viewed by multiple crowd workers, which is correlated to the likelihood that it will be selected and completed.

Introduction

Crowdsourcing is now a prevalent way to recruit workers for a wide range of jobs, from micro-tasks (e.g., image tagging, event annotation, digitization) to complex tasks (e.g., translation, proof reading, programming, design). In many platforms, e.g., Amazon Mechanical Turk, requesters post task(s) and crowd workers accept and complete tasks of their choice as per their convenience. Thus, posting tasks to the right crowd at the right time is critical for timely completion and quality of work.

We propose a new metric, *task viewership*, which measures the number of available and attentive crowd workers at any point in time. It has been observed that workers tend to select recent jobs posted on the platform. Thus, posting tasks during high visibility periods not only ensures quick turnarounds, but also improves the completion rates by ensuring that tasks are more likely to be selected. If we have some knowledge about worker qualities, we can further refine this metric to measure the task viewership of *good* crowd workers, hence allowing us to improve the quality of our results. We model arrival of workers on a platform and the duration for which they are online (logged on a platform) as random processes. This allows us to give probabilistic guarantees on the number of workers online at a particular time. We present an algorithm for task posting which increases the task viewership.

Related Work

Task scheduling on a crowdsourcing platform has been widely studied in the literature. In (Rajan et al. 2013), the authors propose an online approach that coordinates the scheduling of crowd tasks across multiple platforms but limit themselves to learning from *externally observable characteristics* (e.g. response time, accuracy) of the platform. (Mao, Kamar, and Horvitz 2013) models the engagement of workers on platforms using machine learning techniques on real data, but does not propose a way to schedule accordingly. Similar to our work, (Faridani, Hartmann, and Ipeirotis 2011) models workers using a discrete choice model for each worker and a Non-Homogeneous Poisson Point process (NHPP) arrival process for workers, but focuses on finding the right pricing. However, due to budget constraints pricing may not be flexible in real-world scenarios, hence we wish to optimize similar properties at a fixed price by varying the time at which we post.

Task Posting Strategy using Viewership

We first model task viewership using the technique of coverage process from stochastic geometry. A coverage process is defined as follows (Hall 1988). Let $P = \{\zeta_1, \zeta_2, \}$ be a countable set of points in a d-dimensional Euclidean space and $\{C_1, C_2, \}$ a countable collection of non-empty d-dimensional sets. $C\{\zeta_i + C_i, i = 1, 2, \}$ is a coverage process where $\zeta_i + C_i$ denotes the Minkowski sum of ζ_i and C_i. We assume that workers arrive according to NHPP as in (Faridani, Hartmann, and Ipeirotis 2011) and λ_t is the time varying rate of worker arrivals to a platform. Further, we assume that each worker is online for a random length of time, R which is described by an independent random variable with finite support. Thus, worker arrival points along with R form a one-dimensional coverage process. This process essentially captures temporal dynamics of the workers on a platform using their log in (arrival) and log-out (departure) times.

Using (Manohar, Ram, and Manjunath 2009), we can analyse this coverage process to obtain task viewership statistics, e.g., at any point in time (i) what is the expected number of workers online on a platform (ii) what is the probability that there are at least k workers online?[1] Given worker statistics, we can also track the viewership of *good* workers (e.g., workers qualified for the task based on experience, reputation etc.) using thinning of Poisson arrival process and subsequently applying the coverage technique.

[1] Viewership by *at least k-workers* is important for majority voting methods used to ensure the accuracy of the tasks.

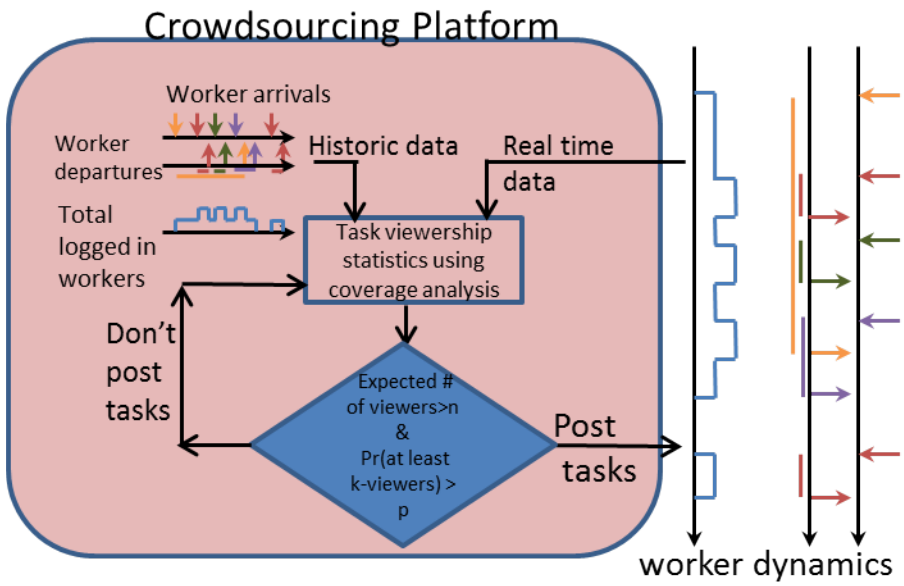

Figure 1: Workflow for the viewership based task posting strategy

We now present a task posting strategy based on task viewership determined by the stochastic model for crowd workers described above.

1. We first *estimate the parameters* for the worker arrival process ($\lambda(t)$) and distribution for online period of a worker ($f_R(r)$) using historical data of crowd workers on a platform. We propose online update of these parameters using real time data as follows. $\lambda(t)$ can be assumed to be linear over appropriate time sub-intervals. The slope and intercept of the linear model can be easily updated in an online manner when new data comes. The online duration can be assumed to be a Beta distribution, the parameters of the distribution can be updated using the observed duration of time.

2. Using these parameters for the coverage process, we obtain *task viewership statistics* which are used to make task posting decisions.

3. Our *posting strategy* works as follows. Given a burst of tasks, based on viewership statistics, we decide how to distribute them over time – posting them simultaneously may be counter-productive as the requester's tasks will be competing with each other for viewership. At the proposed time of posting, we make online decision on whether to post the tasks or not based on the updated viewership statistics – allowing to accommodate any real time variations in worker statistics. Fig. 1 describes the workflow of our strategy. Threshold parameters for decision making, (i) n for expected number of online workers and (ii) p for the probability of k-workers being online at the time of posting, are the tunable parameters. These are determined by the requester and are typically based on the task type and its requirements with respect to completion time etc. If the threshold parameters are met then tasks are posted else they wait until the next time epoch and check for the threshold condition again. How long to wait to meet the condition or to change the condition itself depends on requester's discretion.

Future Work

We are working towards the evaluation of our strategy with real world data from a crowdsourcing platform. Such evaluation would be critical for understanding the applicability of the proposed solution in real-world scenarios. We are further exploring how the proposed metric can be extended to multiple platforms, i.e. by using appropriate platform statistics to decide when to post on which platform to increase the task viewership.

References

Faridani, S.; Hartmann, B.; and Ipeirotis, P. 2011. Whats the right price? pricing tasks for finishing on time. In *AAAI*.

Hall, P. 1988. *Introduction to the Theory of Coverage Process*. John Wiley and Sons.

Manohar, P.; Ram, S.; and Manjunath, D. 2009. Path coverage by a sensor field :the non homogeneous case. *ACM Transactions On Sensor Networks (TOSN)* 5(2).

Mao, A.; Kamar, E.; and Horvitz, E. 2013. Why stop now? predicting worker engagement in online crowdsourcing. In *AAAI HCOMP*.

Rajan, V.; Bhattacharya, S.; Celis, E.; Chander, D.; Dasgupta, K.; and Karanam, S. 2013. Crowdcontrol: An online learning approach for optimal task scheduling in a dynamic crowd platform. In *ICML Workshop: Machine Learning Meets Crowdsourcing*.