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

CrowdUtility: A Recommendation System for Crowdsourcing Platforms

Deepthi Chander[1] Sakyajit Bhattacharya[1] Elisa Celis[2] Koustuv Dasgupta[1]
Saraschandra Karanam [1] Vaibhav Rajan[1] Avantika Gupta[1]
Xerox Research Centre India, Bangalore, India [1]
{deepthi.chander, sakyajit.bhattacharya, koustuv.dasgupta, karanam.saraschandra, vaibhav.rajan, avantika.gupta}@xerox.com
Ecole Polytechnique F´ed´erale de Lausanne (EPFL), Lausanne, Switzerland[2]
elisa.celis@epfl.ch

Abstract

Crowd workers exhibit varying work patterns, expertise, and quality leading to wide variability in the performance of crowdsourcing platforms. The onus of choosing a suitable platform to post tasks is mostly with the requester, often leading to poor guarantees and unmet requirements due to the dynamism in performance of crowd platforms. Towards this end, we demonstrate *CrowdUtility,* a statistical modelling based tool for evaluating multiple crowdsourcing platforms and recommending a platform that best suits the requirements of the requester. CrowdUtility uses an online Multi-Armed Bandit framework, to schedule tasks while optimizing platform performance. We demonstrate an end-to end system starting from requirements specification, to platform recommendation, to real-time monitoring.

Introduction

Recent years have witnessed the emergence of a plethora of crowdsourcing platforms, such as Crowdflower, MobileWorks, Amazon Mechanical Turk, which aim to leverage the collective intelligence of a largely distributed Internet workforce to solve a wide range of tasks. These tasks (such as image tagging, digitization, translation) are posted on crowdsourcing platforms by requesters and are picked by workers registered to these platforms. Workers get monetized on task completion if they meet the SLAs (performance expectations such as accuracy, task completion time) specified by the requester of the task. The challenge in crowdsourcing arises from the fact that crowdworker availability and performance is typically dynamic, unlike in an organizational setting. This dynamism translates to temporal variations in the performance of platforms in terms of parameters such as accuracy and task completion time. Previous work has reported these dynamic variations at both daily and timely scales (Dasgupta et al. 2013). Additionally requesters may themselves have diverse requirements in terms of the size, complexity and timings of the tasks, as well as performance expectations. This heterogeneity in requester requirements confronted with the temporally varying platform performance characteristics makes it difficult for the requester to choose from the abundance of crowdsourcing platforms in the market today. We demonstrate CrowdUtility, a first-of-a-kind statistical machine learning based tool, which models dynamic platform behaviors; uses these models to recommend the best platform for a task; and schedules large batches of tasks to meet user requirements.

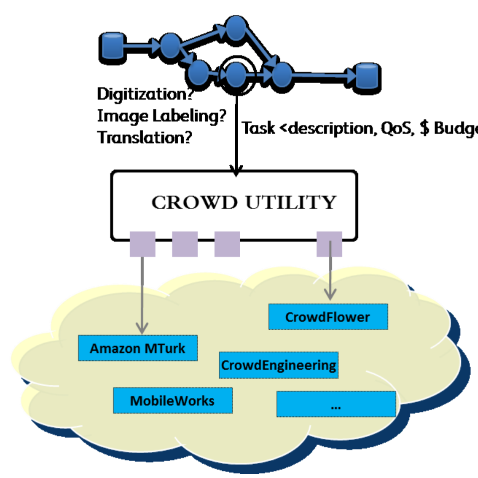

Figure 1: Architecture of CrowdUtility

System Overview

Assigning the right task to the right platform at the right time can have significant implications on how well task requirements (e.g. SLAs) are met. Poor assignments, especially in the case of enterprise tasks, can lead to SLA violations like larger completion times, increased costs or poor accuracy. Current research focuses on platform performance prediction based on knowledge of internals (eg. worker demographics, identities) of platforms. With the mushrooming of platforms and limited insight into platform behaviors, the current state of the art does not address the increasing demands of reliable, enterprise crowdsourcing scenarios. With this motivation, we demonstrate *CrowdUtility*, a novel system for evaluating multiple crowdsourcing applications and recommending a platform that best suits the requirements of the requester.

How it works

Each task is represented by a category, a task description, payment made for the task (dollars), and QoS parameters (e.g. response time, accuracy). Every time a task executes in a platform, CrowdUtility collects statistics (mentioned above) pertaining to the execution of the task. We focus on statistical parameters that are externally observable characteristics (EOCs), i.e. parameters that can be measured for each task without internal knowledge of the platforms. Thus each platform is treated as a "black-box" and historical characteristic data is maintained by CrowdUtility for each platform (Dasgupta et al. 2013). The EOCs in the system include number of fields, number of independent worker validations, payment per task, time and date of posting the task, task completion time, and task accuracy. EOC measurements are subsequently used by CrowdUtility as follows: (1) the platform statistics repository collects and maintains historical data from multiple platforms; (2) this data is used to create statistical models that characterize each platform's behavior over time; (3) recommendations are computed based on input task requirements and existing behavior models; (4) after a recommended platform is selected by the user, CrowdUtility dispatches the tasks to the platform for execution. On completion, data from the current execution is fed back to CrowdUtility for real-time updation of the performance models.

Recommendation System

CrowdUtility casts the problem of platform recommendation as a classification problem (Dasgupta et al. 2013). The classes are the platforms and the features are the EOC characteristics. A classifier learns the behavior of a platform using historical values of EOC parameters. When a new set of parameters is provided by the requester, CrowdUtility *classifies* the parameters to one particular platform. In our evaluation, we used 75% of the data from a four-week dataset for training and 25% of the data as test set. Tree based classifiers, K-Nearest Neighbors and Gradient Boosting Classifier achieved 90% classification accuracy. The Gini importance of various features in the classification revealed that Task Accuracy, Task-Completion Time, Day of Week and Hour of Day are the most important discriminatory features between the two platforms.

Task scheduling strategy

A multi-armed bandit based scheduling framework has been used to learn changing temporal patterns of platform performance and to adapt to these changes by varying the number of tasks submitted at different instants of time. Task scheduling progresses in rounds, where in each round, a subset of tasks (from a batch) is submitted to the crowd with a stipulated set of requirements (e.g., on cost/time/accuracy). Response parameters such as mean completion time and mean-to-variance ratio of accuracy (in a previous round) are used to refine the parameter settings for the next round, such that requirements are further optimized. Our experiments showed that adaptive learning algorithms significantly outperformed other alternatives. In particular, a Thompson Sampling (TS) - based strategy has been demonstrated to achieve higher performance while scheduling large batches of tasks (Rajan et al. 2013).

Demonstration

A demonstration of CrowdUtility can be viewed and downloaded at https://vimeo.com/101611708. CrowdUtility currently provides three functionalities – Browse, Analyze and Recommend. Browse allows a user to search for and explore detailed offerings of available platforms. Analyze allows a requester to compare historical performance across platform(s) for different task categories, sizes, time zones and metrics. The Recommend feature allows a requester to specify desired performance requirements and find a list of platforms in descending order of their likelihood of meeting requirements. The requester can also use the scheduling feature for optimizing the execution of large batches. When the requester chooses a particular platform, our system automatically distributes tasks to the chosen one via platform APIs. The requester can view the progress of task execution and metrics (cost, accuracy, completion rate etc.) in real-time. The tool is currently in pilot with the Xerox Healthcare Services who are interested in engaging crowd workers for digitizing large batches of insurance forms. We are further exploring how to schedule tasks across multiple platforms that are eligible for a specific type of a task.

References

Dasgupta, K.; Rajan, V.; Karanam, S.; Ponnavaikko, K.; Balamurugan, C.; and Piratla, N. M. 2013. Crowdutility: know the crowd that works for you. In CHI'13 Extended Abstracts on Human Factors in Computing Systems, 145–150. ACM.

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