A Platform for Task Recommendation in Human Computation

Debarshi Basak
Delft, The Netherlands
d.basak@student.tudelft.nl

Babak Loni
Delft University of Technology
b.loni@tudelft.nl

Alessandro Bozzon
Delft University of Technology
a.bozzon@tudelft.nl

ABSTRACT

Crowdsourcing and Human computation have enabled industry and scientists to create innovative solutions by harnessing organised collective human effort. In human computation platforms, it is observed that workers spend large amount of time searching for appropriate tasks due to lack of effective task discovery mechanism. This loss of time translates to loss of incentives for worker and affects motivation to solve more tasks. Task recommendation in human computation can not only help mitigating this problem, but it can also result in high quality answers from motivated workers. While few works empirically proved the benefits of task recommendation in human computation platforms, we advocate for a better scientific understanding of how worker- and task-modelling can help to systematically achieve faster and higher-quality task executions. To this end, it is fundamental the availability of tools able to "open the box" of human computation, and offer direct control on worker and task properties, to be later used for recommendation. This paper presents `BruteForce`, a framework that simplifies experiments with commercial human computation platforms, while offering task recommendation features based on rich (and extensible) set of worker and task properties. We describe the characteristics of `BruteForce`, and we report on a set of preliminary experiments with three user profiling techniques namely *feature-independent, feature-based* and *Composite*.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous; D.2.8 [**Software Engineering**]: Metrics—*complexity measures, performance measures*

Keywords

Crowdsourcing, human computation, Recommender systems

1. INTRODUCTION

Crowdsourcing can be defined as a phenomena that takes place when a firm sources an online community (or crowd) for service either for free or certain incentive. When computation is crowdsourced to human, we call this phenomenon as human computation. Although research in human computation is on a rise in both research community and industry, few works addressed the study of how recommendation techniques could be applied in order to improved efficiency and predictability of marketplaces like Amazon Mechanical Turk [3].

We can distinguish two classes of recommendation techniques, namely *Task-oriented* and *Performer-oriented* recommendation techniques. Yuen et al. [5] formulated a task-oriented technique named *TaskRank*, in which the ratings of the tasks are predicted based on the explicit rating provided by workers. In a later work [6], authors propose a mechanism to implicitly compute ratings for a given task. More recently, other works explored performer-oriented techniques. Sarasua et al. [4] introduce the idea of creating a worker profile based on explicit and implicit data available in social Web and across various human computation platforms. In *OpenTurk* [2], authors have designed and implemented a framework for pushing tasks to workers on Facebook based on social profiles analysis. They propose category-based, text-based and graph-based approaches to push task to workers. These works empirically demonstrated the benefits of task recommendation in human computation platforms, and clearly show the benefits deriving from the availability of rich task and worker models. The majority of human computation marketplaces, however, operate as "black boxes", providing their users with little or no tools to profile workers, and rarely offer recommendation mechanisms.

2. THE BRUTEFORCE PLATFORM

To further our understanding of task recommendation for human computation, there is the need for tools able to "open the box" of human computation, and offer direct control on worker and task properties. To this end we developed `BruteForce`[1], a framework that simplifies experiments with commercial human computation platforms, while offering task recommendation features based on rich (and extensible) set of worker and task properties.

`BruteForce` design has been inspired by previous works in the field[1], and provides an abstraction layer on top of existing human computation tools and platforms where researchers and practitioners can easily design, execute, and monitor tasks. W.r.t. existing works, we focused on the problem of task assignment and recommendation by incorporating a rich set of worker and task properties. In addition to traditional metadata like title, description, etc, a task is described by: 1) task *media type*, e.g. text, image, video, audio, and surveys; 2) task *operation type*, e.g. tagging, summarization, classification, ranking and comparison; and 3) task *topic*, selected among one of the 10 high level topics from Open Calais

CrowdRec 2014, 6 Oct. 2014, Foster City, CA, USA

[1] `http://bruteforce.in/`

document categorization service[2]. Workers can be modelled according to three dimensions: 1) explicit information coming from external sources (e.g. social networks), gathered with the consent of the worker during registration to the platform; 2) implicit information logged during the workers' activities on the platforms; and 3) feedbacks provided by task requesters.

In addition to standard components (e.g. task creation and management interface), `BruteForce`'s architecture consists of additional modules devoted to the support of task recommendation, namely: a *worker-modeling* module, a *task-modeling* module, and a *task recommendation* module that is used both to suggest tasks to workers (*pull* mode) and to pre-assigns tasks (*push* mode). All modules are designed with extensibility in mind. The *task recommendation* module, together with the logging and feedback functionalities of `BruteForce`, supports several recommendation techniques (e.g. item-based nearest-neighbour and user-based nearest-neighbour).

The pluggable and customisable nature of `BruteForce` allows implementing a recommendation system native to the platform that would be handy for online and user-study-based experiments, as the one reported in the next section.

3. EXPERIMENT

To empirically demonstrate the capabilities of `BruteForce`, we performed an experiment on task recommendation using a *feature-independent*, *feature-based* and a *composite* profiling-based algorithms. In the *feature-independent* approach we create a user-task matrix with values as number of executions done by the user. In the *feature-based* approach we create a user-feature matrix in which users are represented based on their features. In the third approach which is *composite* profiling, an extended matrix is created which is the combination of user-task and user-feature matrices.

`BruteForce` was also used to create a curated dataset of 70 different tasks, spreading across 10 categories, five operations and three classes. In total, 24 workers have participated in our experiment. Workers were invited from two social web platforms Facebook and LinkedIn as well as our locally created platform. A clear set of instruction where given to performers to choose tasks that they really liked. For truly understanding performer preferences, we did not provide incentives to performers in our platform. This ensured that performers would choose tasks that motivate them based on the content of the tasks. To avoid bias due to the availability of tasks, the experiment were conducted on our platform using a non-competitive setup wherein users did not compete for tasks. In other words, if we have n tasks and m users then total number of assignments are $n * m$.

We defined 18 different features for user. The features are defined based on the same 10 categories, five operations and three classes with which a task can be defined. For evaluation, we trained our algorithms using 60% of the dataset[3] as training and used rest of the data as test. The three recommendation approaches recommend an un-ordered set of tasks based on training set. We evaluated the performance of the three approaches, as reported in Table 1.

In addition to traditional set-based measures like precision, recall, and F1-measure, we also consider *User Miss Rate*, that is the number of performers who do not have any matched recommended items in the test dataset. Such a metric is really important in the context of task recommendation in human computation, as all the platform workers should be provided with tasks to be executed.

User Profile Type	Prec.	Rec.	F1	UMR
Random	0.025	0.011	0.015	0.920
Feature Independent	0.277	0.399	0.328	0.670
Feature Based	**0.303**	0.331	0.316	0.670
Composite	0.285	**0.408**	**0.336**	**0.330**

Table 1: Performance (Precvision, Recall, F1-measure , and User Miss Rate) of the three evaluated recommendation strategies.

We can observe how each technique provide good performance with a single measures. Such a diversity is an interesting property to discuss, as it shows how different task recommendation scenario call for different recommendation strategies. When quality is important, a recommender with high precision (in our case *feature based*) is suitable, whereas a scenario where speed or task execution coverage is important, then a recommender with high recall is preferred (e.g, the *feature independent*). The *composite* approach outperforms both the *feature independent* and the *feature based* profile techniques, especially considering the *User Miss Rate*. Such a result shows the potential of combining different techniques for task recommendation, where the employment of workers is a fundamental requirements.

4. CONCLUSION AND FUTURE WORK

In this paper, we presented `BruteForce`, and discuss an example of on-going research carried out with it. We evaluated a performer-oriented recommendation approach, where tasks are modelled based on topic, operation and class. Using implicit positive-only feedback data collected from workers, we evaluate these recommendation strategies, and we elaborate on their application in different task recommendation scenario. Future work includes the extension of the platform with new recommendation algorithms, and their test with the created dataset.

Acknowledgment

This work was supported by the Dutch national program COMMIT.

5. REFERENCES

[1] A. Bozzon, M. Brambilla, S. Ceri, and A. Mauri. Reactive crowdsourcing. In *Proceedings of the 22nd international conference on World Wide Web*, pages 153–164. International World Wide Web Conferences Steering Committee, 2013.

[2] D. E. Difallah, G. Demartini, and P. Cudré-Mauroux. Pick-a-crowd: Tell me what you like, and i'll tell you what to do. In *Proceedings of the 22nd international conference on World Wide Web*, pages 367–374. International World Wide Web Conferences Steering Committee, 2013.

[3] P. G. Ipeirotis. Analyzing the amazon mechanical turk marketplace. *XRDS: Crossroads, The ACM Magazine for Students*, 17(2):16–21, 2010.

[4] C. Sarasua and M. Thimm. Microtask available, send us your cv! In *Cloud and Green Computing (CGC), 2013 Third International Conference on*, pages 521–524. IEEE, 2013.

[5] M.-C. Yuen, I. King, and K.-S. Leung. Task matching in crowdsourcing. In *Internet of Things (iThings/CPSCom), 2011 International Conference on and 4th International Conference on Cyber, Physical and Social Computing*, pages 409–412. IEEE, 2011.

[6] M.-C. Yuen, I. King, and K.-S. Leung. Taskrec: probabilistic matrix factorization in task recommendation in crowdsourcing systems. In *Neural Information Processing*, pages 516–525. Springer, 2012.

[2] `http://www.opencalais.com/documentation/calais-web-service-api/api-metadata/document-categorization`

[3] `http://bruteforce.in/response.csv`