Adaptive Incentive Selection for Crowdsourcing Contests

Extended Abstract

Nhat V.Q. Truong
University of Southampton
Southampton, UK
nvqt1r14@ecs.soton.ac.uk

Sebastian Stein
University of Southampton
Southampton, UK
ss2@ecs.soton.ac.uk

Long Tran-Thanh
University of Southampton
Southampton, UK
l.tran-thanh@soton.ac.uk

Nicholas R. Jennings
Imperial College London
London, UK
n.jennings@imperial.ac.uk

KEYWORDS

incentive; crowdsourcing contest; budgeted MAB

ACM Reference Format:
Nhat V.Q. Truong, Sebastian Stein, Long Tran-Thanh, and Nicholas R. Jennings. 2018. Adaptive Incentive Selection for Crowdsourcing Contests. In *Proc. of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2018), Stockholm, Sweden, July 10–15, 2018,* IFAAMAS, 3 pages.

1 INTRODUCTION

The success of crowdsourcing projects relies critically on motivating the crowd to contribute [2, 10]. Given this, contests[1] have been shown to be an effective approach in these projects, as they are effective and cheap. Actually, by rewarding participants in a contest, task requesters do not have to pay for every task completed as in other types of financial rewarding such as paying for performance [8] or using bonuses [4, 13]. Indeed, they have to pay only for a certain number of participants, e.g., the top two who have completed the most tasks in a day. 99designs (www.99designs.com), TopCoder (www.topcoder.com), and Taskcn (www.taskcn.com) are some well-known crowdsourcing platforms using contests for attracting participants. Nevertheless, the effectiveness of the contests (henceforth, *incentives*[2]) might be different between crowdsourcing projects based on specific properties of those projects, such as the project purpose (e.g., building data for scientific studies or collecting data for a company), the task nature (e.g., interesting or boring), or the participant community (e.g., the extent to which they are in contact with each other), as the participants might have different motivations [3, 5, 14]. Therefore, finding an appropriate way for an autonomous agent to choose an effective incentive in a crowdsourcing project (which is referred to as the *incentive selection problem*, ISP) is necessary.

In general, the effectiveness of the incentives in a specific crowdsourcing project is unknown in advance. Thus, in order to identify the most effective one to provide (i.e., exploitation), the agent has to try each incentive several times to evaluate its respective effectiveness (i.e., exploration). Given this need to balance exploitation and exploration, budgeted multi-armed bandits (MABs), e.g., [11] and [12], are a good approach for this problem. Specifically, they model the problem as a machine with k arms (corresponding to k incentives), pulling an arm (providing an incentive to a group of participants) incurs a fixed cost (attached to the arm) and delivers a random reward (i.e., the utility) drawn from an unknown distribution. The objective in a MAB problem is to find a pulling policy that maximises the expected total reward within a given budget (e.g., £500) before a deadline (e.g., in two weeks). A number of algorithms have been proposed to solve the budgeted MAB problem [1, 9, 11, 12]. However, these algorithms are not designed to work with the time budget (i.e., the deadline) of the ISP and they do not consider the group-based nature of the incentives (i.e., contests), that is, the outcome of pulling an arm is the total aggregated outcome of the individuals in the corresponding contest group. Thus, as we will show in Section 4, they are not efficient when dealing with the ISP. To illustrate the importance of the group-based nature, consider the two cases when the group size is 5 (i.e., 5 participants per contest) and 20 respectively. Current MAB algorithms would not treat these cases significantly differently. However, the latter clearly provides us with more information on each pull (as it has more samples, i.e., participants). As a result, the second case requires fewer rounds of exploration in order to achieve the same level of understanding of the participants' performance (e.g., after 5 pulls of an arm in the first case, we have effectively sampled the performance of 100 individuals, but would require 20 pulls of an arm in the second case to reach that sample size). Hence it is necessary to consider the group-based nature, in order to determine appropriate numbers of pulls for the arms.

[1] We use the term "contest" in a broad sense to refer to any situation in which participants exert effort to submit tasks for prizes, which are provided based on relative performance. The prizes can be tangible rewards, points, or positions on a leaderboard. Thus, all-pay auctions, lotteries, and leaderboards can be considered as contests.
[2] The incentives focused on in this paper are contests. However, the problem stated and the algorithms discussed can be used with any other types of incentive, such as paying for performance or using bonus payment. Hence, to keep the problem general, we use "incentives" instead of "contests".

Proc. of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2018), M. Dastani, G. Sukthankar, E. André, S. Koenig (eds.), July 10–15, 2018, Stockholm, Sweden.

In order to address this gap, we introduce two algorithms to deal with the 2d-budgeted MABs (MABs constrained by an overall financial budget and a time budget), which take into consideration both the time limit and the group-based nature of the problem.

2 PROBLEM OVERVIEW

Suppose we are going to run a crowdsourcing project. We want to complete as many tasks as possible with a given budget $\mathcal{B}$ before a given time. To do that, we spend that budget on providing contests to encourage participants (referred to as *users*) to perform tasks. The contests can be designed in different ways (which vary in performance evaluation method or prize distribution). However, their effectiveness (i.e., average number of tasks completed per cost unit in each incentive[3]) is unknown in advance. Thus, we are interested in finding an efficient algorithm for selecting the incentives (i.e., exploring the effectiveness of contest structures and then exploiting the most effective one) to maximise the total number of tasks completed. The algorithm needs to be run without manual tuning its situation-specific parameters, so that it can be used by an autonomous agent.

3 ALGORITHMS

We introduce two algorithms for the ISP: *HAIS* (our new approach, stands for Hoeffding-based Adaptive Incentive Selection) and *Stepped ϵ-first* (a modification of ϵ-first [11]). Although the algorithms are designed for incentives in the form of contests, they can be used with many other types of incentives, such as pay for performance or bonuses (where the group size is 1, i.e., there are no contests).

3.1 HAIS

HAIS spends two periods to explore, and the rest to exploit. Specifically, in the first period, it pulls the arms so that each arm has at least a certain number of user (e.g., 20) in order to have *initial estimates* of the arms' performance. In the second period, it applies *Hoeffding*'s inequality [6] to identify a target number of users (i.e., the minimum number of users to have in each arm after the second period) to obtain a certain confidence level (e.g., 50%) that the current best arm (based on the estimates) is the real best arm. Before spending the last period for *pure exploiting* (i.e., pulling the best arm with the residual budget), it conducts *stepped exploiting*, that is to spread the residual budget over the periods so that it can switch the arm if it finds that the current best arm is not the best any more.

3.2 Stepped ϵ-first

This algorithm is a modified version of ϵ-first [11] that is designed to run more effectively under a time limit. Specifically, it spends $\epsilon\mathcal{B}$ (where ϵ is specified in advance, e.g., 0.2) in the first period to explore by pulling the arms evenly until exceeding this budget. Its exploration is almost the same as HAIS'.

[3] To keep the presentation clear, we stick with this simple metric on the effectiveness. However, in practice, it is possible to use more complicated metrics by combining multiple aspects, such as task quantity, task quality, and completion time.

4 EMPIRICAL EVALUATION

To measure the performance of the algorithms, we run simulations with synthetic data. We compare HAIS and Stepped ϵ-first with the following benchmarking algorithms with some minor modifications (as they are not designed to deal with the time constraint of the ISP): *ϵ-first*, *fKUBE* [12], *Stepped fKUBE* (fKUBE with stepped exploiting), *Survival of the Above Average (SOAAv)* [9], and *Optimal Solution* (unachievable in reality, that pulls the real best arm all the time).

4.1 Results and Discussion

In general, both HAIS and Stepped ϵ-first are effective (Figure 1). Stepped ϵ-first performs well but requires its parameter (i.e., ϵ) to be chosen appropriately (that might be difficult when lack of prior knowledge). Besides, HAIS performs best in most cases without depending significantly on the parameter choosing. In contrast, the others (Stepped fKUBE and SOAAv) are not suitable for the ISP as they do not have efficient ways to balance exploration-exploitation take advantage of the time budget in the exploitation phase.

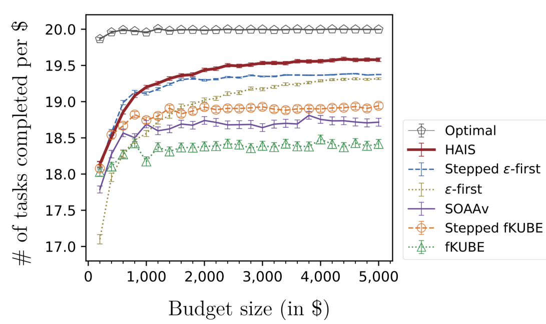

Figure 1: Performance of the algorithms in the simulations on synthetic data with 5 arms. *Mean values and 99% confidence intervals are plotted.*

5 FUTURE WORK

As our algorithms make several assumptions, we will address these in future work in order to deal with more general settings. To illustrate this with an example, we plan to develop algorithms that work efficiently in the following case with three incentives (or significantly more complex settings with many more incentives): the first incentive might pay for every bulk of tasks [7], the second one may be bonuses [13], and the last one may use contests with group size of 5. In this case, the incentives' group sizes are different (1, 1, and 5, respectively). Additionally, the corresponding periods might be different in length (e.g., only 2 hours with the first two incentives and 1 day with the last one).

REFERENCES

[1] Ashwinkumar Badanidiyuru, Robert Kleinberg, and Aleksandrs Slivkins. 2013. Bandits with Knapsacks. In *FOCS '13 IEEE 54th Annual Symposium on Foundations of Computer Science*. Institute of Electrical and Electronics Engineers (IEEE), Berkeley, California, USA, 207–216.

[2] Anhai Doan, Raghu Ramakrishnan, and Alon Y. Halevy. 2011. Crowdsourcing Systems on the World-Wide Web. *Commun. ACM* 54, 4 (April 2011), 86–96.

[3] Bruno S. Frey and Reto Jegen. 2001. Motivation Crowding Theory. *Journal of Economic Surveys* 15, 5 (Dec. 2001), 589–611.

[4] Christopher Harris. 2011. You're Hired! An Examination of Crowdsourcing Incentive Models in Human Resource Tasks. In *CSDM '11 Proceedings of the Workshop on Crowdsourcing for Search and Data Mining at the Fourth ACM International Conference on Web Search and Data Mining (WSDM)*. Association for Computing Machinery (ACM), Hong Kong, China, 15–18.

[5] James Heyman and Dan Ariely. 2004. Effort for Payment a Tale of Two Markets. *Psychological Science* 15, 11 (2004), 787–793.

[6] Wassily Hoeffding. 1963. Probability Inequalities for Sums of Bounded Random Variables. *J. Amer. Statist. Assoc.* 58, 301 (March 1963), 13.

[7] Kazushi Ikeda and Michael S. Bernstein. 2016. Pay It Backward: Per-Task Payments on Crowdsourcing Platforms Reduce Productivity. In *CHI '16 Proceedings of the 2016 Conference on Human Factors in Computing Systems*. ACM, San Jose, California, USA, 4111–4121.

[8] Winter Mason and Duncan J. Watts. 2010. Financial Incentives and the "performance of Crowds". *KDD-HCOMP '09 ACM SigKDD Explorations Newsletter* 11, 2 (2010), 100–108.

[9] Sandip Sen, Anton Ridgway, and Michael Ripley. 2015. Adaptive Budgeted Bandit Algorithms for Trust Development in a Supply-Chain. In *AAMAS '15 Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*. International Foundation for Autonomous Agents and Multiagent Systems (IFAAMAS), Istanbul, Turkey, 137–144.

[10] Henri Simula. 2013. The Rise and Fall of Crowdsourcing?. In *HICSS '13 Proceeding of the 46th Hawaii International Conference on System Sciences*. Institute of Electrical and Electronics Engineers (IEEE), Wailea, Hawaii, 2783–2791.

[11] Long Tran-Thanh, Archie Chapman, Jose Enrique Munoz De Cote Flores Luna, Alex Rogers, and Nicholas R. Jennings. 2010. Epsilon–first Policies for Budget–limited Multi-Armed Bandits. In *AAAI '10 Proceedings of the 24th AAAI Conference on Artificial Intelligence*. Association for the Advancement of Artificial Intelligence (AAAI), Atlanta, Georgia, USA, 1211–1216.

[12] Long Tran-Thanh, Archie C. Chapman, Alex Rogers, and Nicholas R. Jennings. 2012. Knapsack Based Optimal Policies for Budget-Limited Multi-Armed Bandits. In *AAAI '12 Proceedings of the 26th AAAI Conference on Artificial Intelligence*. Association for the Advancement of Artificial Intelligence (AAAI), Toronto, Ontario, Canada, 1134–1140.

[13] Ming Yin and Yiling Chen. 2015. Bonus or Not? Learn to Reward in Crowdsourcing. In *IJCAI '15 Proceedings of the 24th International Joint Conference on Artificial Intelligence*. Association for the Advancement of Artificial Intelligence (AAAI), Buenos Aires, Argentina, 201–207.

[14] Haichao Zheng, Dahui Li, and Wenhua Hou. 2011. Task Design, Motivation, and Participation in Crowdsourcing Contests. *International Journal of Electronic Commerce* 15, 4 (July 2011), 57–88.