Mechanism Design for Online Crowdsourcing Problem

Hui Huang
UNI: hh2924

Wenqian Xing
UNI: wx2261

Xinghan Gong
UNI: xg2356

Yufei Wu
UNI: yw3613

Abstract

With the proliferation of online users on the Internet, numerous opportunities for crowdsourcing to perform various of tasks have been created. However, the development of online mechanisms for the crowdsourcing platform is relatively slow compared with the rapid growth of the demand. In this paper, we review the current two state-of-the-art online mechanisms for the crowdsourcing problem in posted-price model and auction model respectively. We implement both mechanisms and conduct numerical experiments to test their performance on the same dataset. In addition, we further improve the posted-price mechanisms based on our own understanding and use a new method in proving truthfulness in the auction-based mechanism. By conducting this project, we have gained a full understanding on the online crowdsourcing problem and the corresponding online and offline mechanisms. The code of our project can be found on `https://github.com/vgerous/IEOR-4530-Final-Project`.

1 Introduction

Crowdsourcing is a type of participative online activity in which an entity proposes to a group of individuals via an open platform, the voluntary undertaking of a task. The participants will receive the satisfaction of a given type of need, be it economic, social recognition, self-esteem, or the development of individual skills, while the crowdsourcer will obtain and utilize to their advantage that what the user has brought to the venture, whose form will depend on the type of activity undertaken [2]. Typical applications include Amazon Mechanical Turk, a commercial platform which matches micro-tasks submitted by requesters to willingly workers, and non-profit organization, Wikipedia, has used crowdsourcing to develop common goods [9].

Meanwhile, mobile devices have been equipped with various types of sensors, including accelerometer sensor, gyroscope and GPS, which could be assigned to perform perception tasks under an open platform within crowdsourcing paradigm. Therefore, a new spatial data collection mode, crowdsensing, which combines the concept of crowdsourcing and potential perception ability in mobile devices, has gradually become one of the research highlights [1]. Taking advantage of the ubiquitous presence of powerful mobile computing devices (especially smartphones) in recent years, it has also become an appealing method to businesses that wish to collect data without making large-scale investments.

The typical system architecture of crowdsensing is shown in Figure 1. The system architecture includes three parts: server platform, data requester and task participants (data providers). The server in the cloud accepts service requests from data requesters, allocate perception tasks to the participants and process the uploaded data by performing various management functions. As for the data participants, after receiving the perception tasks, they will collect the required data, and then return it to the server, which will further process the data and give to the data requester. Crowdsensing is a distributed, mobile and autonomous service model. It implements functions such as data perception, data collection and information service provision through the whole process.

Crowdsensing can collect massive multi-dimensional heterogeneous data from various places, solve large-scale data demand problems, and provide high-quality and reliable data services. However,

Preprint. Under review.

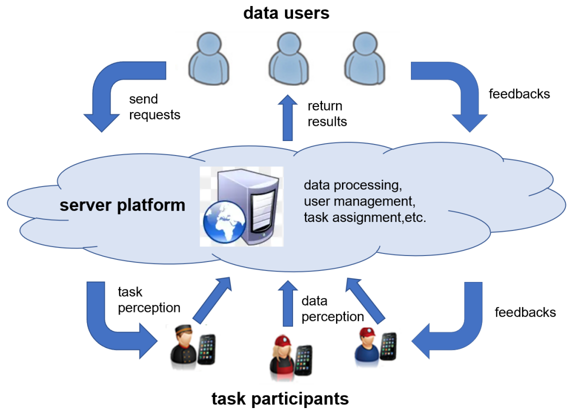


Figure 1: A typical architecture of a crowdsourcing system

with the popularity of crowdsensing, new problems and challenges have gradually emerged. Among them, the main reason for restricting its development is participants' low enthusiasm, and also the server platform cannot recruit enough participants to obtain high-quality and reliable perception data. Therefore, how to motivate more participants to join in providing perceptual data with high reliability is one of the key issues in crowdsensing.

In this paper, we review two different mechanisms for the online crowdsensing problem. The first one is a posted-price mechanism which was introduced by Singla *et al.*in 2013 [7]. The second one is an auction-based mechanism which was introduced by Zhao *et al.*in 2014 [12]. Both these two different mechanisms aim to solve the online crowdsensing problem with stochastic arrivals and perform pretty well on their own dataset. However, we are interested in directly comparing these two different approaches and observing their performance on the same dataset. In this sense, we implement both mechanisms in the same environment. Furthermore, we add some improvements on them based on our own understanding of the online crowdsensing problem.

The main contributions of this article are summarized as follows.

- review the online mechanisms of the crowdsensing problem;
- implement various mechanisms and conduct numerical experiments;
- improve the posted-price mechanism in combining budget and price information.
- improve the proof process regarding the cost-truthfulness in auction-based mechanism.

The remainder of this article is organized as follows. The literature review is presented in Section 2. In Section 3, we formulate the problem statement in detail. In Section 4, the posted-price mechanism is first described. In Section 5, the auction mechanism is then described and provide the new proof process. In Section 6, the numerical results of our experiments are shown. The conclusion part will be in Section 7.

2 Literature Review

In this section, we conduct a thorough literature review on both online and offline mechanisms of crowsourcing problem. Most recent literature related to incentive mechanism design can be categorized into offline [3, 10, 6, 8, 4] and online [7, 12, 11, 5] cases. In the offline scenario, the platform need to know information of all users before making decisions on whether or not accepting users' bids. While in the online scenario, the platform makes decisions only based on the information of users arrived so far, without future information.

2.1 Offline Mechanisms

In our project, reviewed past studies in designing offline incentive mechanisms include [3, 10, 6, 8, 4]. Generally, two different kinds of models, platform-centric and user-centric models, are considered in these researches. In the platform-centric model, the platform provides an unchanged reward to participants and the model focuses on solving Nash or Walrasian equilibrium with Stackelberg game or exchange economy theory. While in the user-centric model, the reward for the sensing tasks can be determined by participants, and direct mechanism is applied to achieve truthfulness and individual rationality. Kong *et al.*[4] analyzed design principles of different incentive mechanisms, including auctions, lotteries, and trust and reputation systems. Peng *et al.*[6] incorporated the consideration of data quality into the design of incentive mechanism, and the payment is related to quality of sensed data provided by user. Song *et al.*[8] also considered the quality of data and design a mechanism to maximize the valuation of the performed tasks with a limited budget. Wang *et al.*[10] and Gan *et al.*[3] further considered the privacy protection problem of the users when design the incentive mechanism.

2.2 Online Mechanisms

Online incentive mechanism in crowdsourcing has also been explored in [7, 12, 11, 5]. Singla *et al.*[7] designed a new posted price-model, where workers are offered a take-it-or-leave-it price offer, using regret minimization in online learning. They further proved the truthfulness, budget feasibility and near-optimal utility of the mechanism, and extensive experiments based on real data were carried out to show its effectiveness. Zhao *et al.*[12] proposed two online mechanisms OMZ and OMG which satisfy truthfulness, individual rationality, budget feasibility, polynomial computation time, consumer sovereignty and constant competitiveness, to maximize the total value services. The mechanisms considered an multiple-stage sampling accepting process in online auction, and dynamically increases the sample size as well as learns a density threshold used for future decision. Xie *et al.*[11] considered a critical property that the values of users contributions decrease as time goes by in online crowdsourcing. They proposed a new strategyproof mechanism to select users based on a time-related threshold, and proved that the mechanism can also achieve computational efficiency, budget feasibility, and a constant competitive ratio. Another important study in the area (Li *et al.*[5]) emphasized that instead of participants arriving at the platform in an online manner, they should arrive online in a random order in realistic mobile crowdsourcing applications, and their interests and status may frequently change over time.

3 Problem Statement

In this section, we describe the problem formulation and the offline benchmarks of the online crowdsourcing problem we aim to address.

Task, crowdsourcer and users. We consider the various types of platforms that post the crowdsourcing tasks as the 'crowdsourcer'. A task posted by the crowdsourcer can be performed by an individual 'user'. To simplify the problem, we assume that the crowdsourcer has unit utility to each task and a fixed budget constraint $B > 0$. The objective for the crowdsourcer is to maximize its total utility A over the whole period $[0, T]$ given the fixed budget constraint. On the other hand, there is a finite pool of users U over the whole period. Each individual user $u_i \in U$ has a private cost c_i to perform the task posted by the crowdsourcer. In a truthful mechanism, the user will report their true cost $b_i = c_i$ in a bidding model. We further assume that the cost of all the users follow an independent and identically distribution f with a lower bound c_{min} and an upper bound c_{max}. The user u_i will perform the task if the payment is larger than or equal to his private cost $p_i \geq c_i$.

Online arrival mechanism. In the online crowdsourcing problem, we consider all the users have stochastic arrivals and arrive one at a time. Consider the maximum time length is T, the users' arrivals will follows a Poisson distribution. The size of the finite pool of users is $|U|$, then the corresponding arrival times are uniformly distributed in the time range of $[0, T]$. In this sense, each time a user arrives at a time $t_i \in [0, T]$, the corresponding cost will be sampled from $c_i \sim f$.

Offline benchmarks. We consider two offline benchmarks in the crowdsourcing problem. The first one is the variable price offline policy (OPT-VAR), where the crowdsourcer can pick different prices for different users knowing all the information about them. In this sense, the crowdsourcer will sort the users by their private cost and pick the ones with lower cost until all the budget is exhausted. The second one is the fixed price offline policy (OPT-FIX), where the crowdsourcer can pick a single best price offer to all the users knowing all the information about them.

4 Posted-price Mechanism

We first review the posted-price mechanism introduced by Singla *et al.*in 2013 [7]. In the paper, the authors present a novel posted-price online learning mechanism called BP-UCB under the environment of crowdsourcing tasks. This BP-UCB mechanism is based on the existing mechanisms using multi-armed bandits for online auctions with some further improvements to support the feature of budget control. Then, we further improve the part of combining the remaining budget and price information into the original BP-UCB mechanism.

4.1 Multi-armed Bandits in crowdsourcing

In crowdsourcing tasks, the posted-price can be modelled discretely as the arms of a multi-armed bandit. Using a multiplicative factor $(1 + \alpha)$, we could discretize the price from $[c_{min}, c_{max}]$ to a set of K arms of prices.

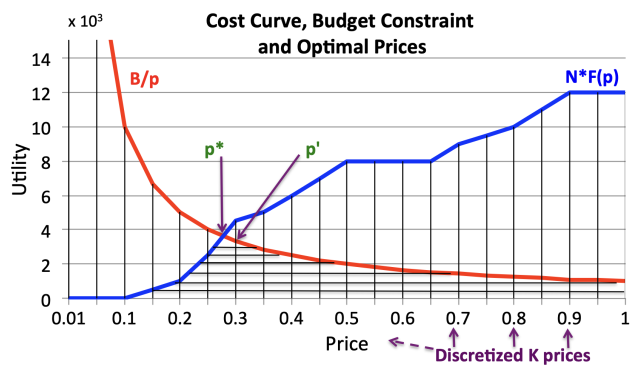


Figure 2: Optimal price p^* over budget constraint and a finite pool of users

In Figure 2, the red line indicates the utility of choosing a fix price if the size of the pool of the users $|U|$ is infinite. On the other hand, the blue line indicates the utility of choosing a fix price in the case that the budget B is unlimited. The crossing point of those two lines is the optimal price p^* in expectation to choose. However, since the price is discrete, the arm p' is the optimal arm in expectation to play in the MAB model.

4.2 Improved BP-UCB Mechanism

In this section, we introduce the improved BP-UCB by combining the remaining budget and price information.

Algorithm 1 is the high-level framework of our improved BP-UCB mechanism, where the improvements are marked in red. We keep the main structure of this algorithm the same as the original BP-UCB mechanism. We start the algorithm by discretizing the prices into K arms in an MAB model. Then, we initialize the time t, budget B^t, current utility A, the upper confidence bounds F_i^t and number of plays N_i^t of each arm in the MAB model.

After the initialization phase, we enter into the online arrival part of the algorithm. If the current budget B^t is positive and time is not reaching to the end, we update the approximate upper confidence bounds F_i^t and estimate the values V_i^t. Compared with the original BP-UCB mechanism, we include the price information p_i into the process of updating the upper confidence bounds. In this sense, the arms with lower prices will have a larger variance compared with those with higher prices. Then we pick the price arm p_i that maximize the current values to offer to the current user u^t. If $p_i \geq c_t$, the user will accept the offer and perform the task. Otherwise, the user will leave without taking the offer. After observing the acceptance or rejection action from the current user, we update total utility A, the remaining budget B^t as well as the related parameters of the UCB. Finally, the algorithm will terminate and output the total utility if the budget is exhausted or the time is reaching to the end.

Algorithm 1: Improved BP-UCB Mechanism

Input: budget constraint B; information of users at each arrival time $u^t,\ c_{min},\ c_{max},\ \alpha$
Output: total utility A

Discretize the prices into K arms. $p_0 \leftarrow c_{min},\ p_{k+1} \leftarrow (1+\alpha)p_k,\ p_K \leftarrow c_{max}$
Initialize time, budget and utility. $t \leftarrow 0,\ B^t \leftarrow B,\ A \leftarrow 0$
Initialize UCB. $N_i^t \leftarrow 0,\ F_i^t \leftarrow 0$
while $B^t > 0$ **and** $t < T$ **do**
  $F_i^t \leftarrow F_i^t + \sqrt{\frac{2 \cdot \ln(t)}{N_i^t \cdot p_i}}$
  $V_i^t \leftarrow \min\left\{F_i^t, \frac{B^t}{N_i^t}\right\}$
  $i^t \leftarrow \operatorname*{argmax}_i V_i^t$, s.t. $p_i \leq B^t$
  Offer price $p^t \leftarrow p_i$ to user u^t
  Observe acceptance or rejection decision y^t from user u^t
  Update $F_i^t \leftarrow F_i^t + \frac{(y^t - F_i^t)}{(N_i^t + 1)}$; $N_i^t \leftarrow N_i^t + 1$
  Update $A \leftarrow A + y^t; B^{t+1} \leftarrow B^t - p^t \cdot y^t; t \leftarrow t + 1$
return A

5 Auction Mechanism

As for the auction model, we look deep into the classic online crowdsourcing model proposed in [12]. The whole process of auctions in crowdsourcing can be described as follows:

1. The platform first provide a task description to users, and some interested users will submit their bid values to the platform.
2. The platform will then decide whether or not accept the bid, and if so, the platform will then give a payment plan to the users.
3. Those who accept the payment will then upload the collected data as required in the task to the platform.

In this project, we only consider the case where the arrival time equals departure time for each user, i.e. users have to immediately decide whether or not to take the payment offer upon arrival. And if we assume each user has unit utility and could only complete exact one task, then our objective is to maximize the number of hired users or completed tasks with a fix budget B.

5.1 Online Mechanism under Zero Arrival-departure Interval Case (OMZ)

The second paper [12] proposed an multi-stage mechanism called OMZ. In Algorithm 2, we first divide all T stages and budget B into $\lfloor \log_2 T \rfloor$ parts for each stage. Then for each arrived users, if his marginal density is not less than threshold ρ^* and total payment at time t is smaller than the stage budget B', the algorithm will allocate the task to him and give a payment equal to the threshold value. In the algorithm, the users with task allocated will be added into the selected set of user pool S.

At the end of each stage, the algorithm will learn a new density threshold used to make future decisions from the past performed auctions, and dynamically increase the sample size and stage budget for the next stage by Algorithm 3. Finally, the algorithm will return the size of selected set of users $|S|$ as the total utility.

5.2 Update Density Threshold

Algorithm 3 implements a greedy method to get the new density threshold. It first sorts users by their marginal densities. If the marginal density of user i is larger than the total average value of selected users including i, we will then add it to a new user set J, which will be later used to compute the new density threshold.

Algorithm 2: OMZ Mechanism

Input : Budget constraint B, deadline T

Initialize stage-step, time-step and density threshold: $t \leftarrow 1$, $T' \leftarrow \frac{T}{2^{\lfloor \log_2 T \rfloor}}$, $\rho^* \leftarrow \epsilon$

Initialize selected, unselected user set (used for update density threshold at the end of each stage) and stage-budget: $S \leftarrow \varnothing$, $S' \leftarrow \frac{T}{2^{\lfloor \log_2 T \rfloor}}$, $B' \leftarrow \frac{B}{2^{\lfloor \log_2 T \rfloor}}$

while $t < T$ **do**

  /* Decide selected users and payment p_i */

  if there is a user i arriving at time t **then**

    if $b_i \leq V_i(S)/\rho^* \leq B' - \sum_{j \in S} p_j$ **then**

      $p_i \leftarrow V_i(S)/\rho^*$; $S \leftarrow U \sqcup \{i\}$

    else

      $p_i \leftarrow 0$

    $S' \leftarrow S' \sqcup \{i\}$

  /* Update stage-budget at the end of each stage */

  if $t = \lfloor T' \rfloor$ **then**

    $\rho^* \leftarrow GetDensityThreshold(B', S')$

    $T' \leftarrow 2T'$; $B' \leftarrow 2B'$

  $t \leftarrow t + 1$

return $|S|$

Algorithm 3: GetDensityThreshold

Input : Stage-budget B', sample set S'

Initialize $J \leftarrow \varnothing$; $i \leftarrow \underset{j \in S'}{\text{argmax}}(V_j(J)/b_j)$

/* Greedy-based threshold updation */

while $b_i \leq \frac{V_i(J)B'}{V(J \sqcup \{i\})}$ **do**

  $J \leftarrow J \sqcup \{i\}$

  $i \leftarrow \underset{j \in S' \setminus J}{\text{argmax}}(V_j(J)/b_j)$

$\rho \leftarrow V(J)/B'$

return ρ/δ

An example of the Algorithm 2 and 3 is given in Figure 3 with total time $T = 8$. The mechanism first divides the total time T into four stages. And in each stage, there is an upper bound for the total payment, which will dynamically increase from $B/8$ to B. And we can also observe that the threshold value will increase from $\frac{1}{2}$ to $\frac{1}{4}$.

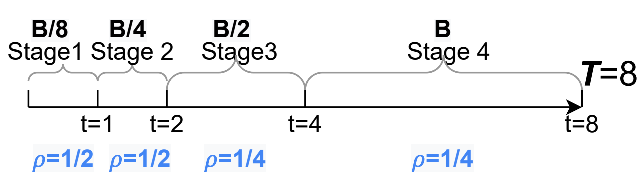


Figure 3: Execution example of OMZ when T=8

5.3 Truthfulness in OMZ

Time-truthfulness. It is trivial to achieve time-truthfulness in this simplified case since each users will not obtain a sensing job or a payment by reporting a later arrival-time or earlier departure-time respectively. Therefore, rational users would truthfully report their arrival and departure time.

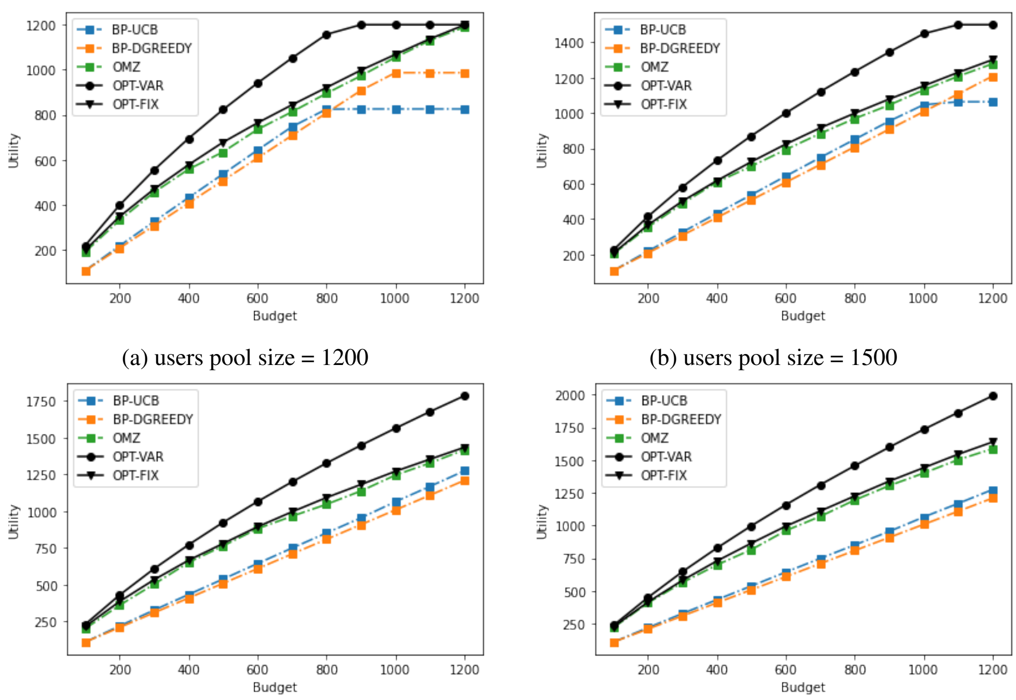
Cost-truthfulness. To show the cost-truthfulness of OMZ, we use a simpler method than the proof process in [12] as follows. Consider a scenario where a user i arrives at some stage with density threshold ρ^*. Then we have,

- *Case 1: No remaining budget left.* Since the user's cost will not affect the allocation result, the user will not obtain higher utility by misreporting cost.
- *Case 2: There exists some remaining budget.* In OMZ, each winning user is paid with a critical value $V_i(S)/\rho^*$. And if user i wins, then $c_i \leq V_i(S)/\rho^*$ and we know that any user with bid $c_i^{'}$, where $c_i^{'} = c_i - \delta$ with $\delta > 0$, will also win the auction. Therefore, OMZ is monotonic in this case and according to Myerson's Lemma, we can show that OMZ is cost-truthful.

6 Numerical Results

We conduct numerical experiments to test the performance of various online mechanisms on the same simulated dataset. We assume the cost of the users follows an uniform distribution with the lower bound $c_{min} = 0.2$ and the upper bound $c_{max} = 1$. In the BP-UCB mechanism, we set the discretized factor $\alpha = 0.01$ and number of arms $K = 200$. We also implement the BP-DGREEDY mechanism purposed in the original paper, which is a simple version of BP-UCB mechanism with truthful bidding information to update the upper confidence bounds.

6.1 Posted-price v.s. Auction



(a) users pool size = 1200

(b) users pool size = 1500

(c) users pool size = 2000

(d) users pool size = 3000

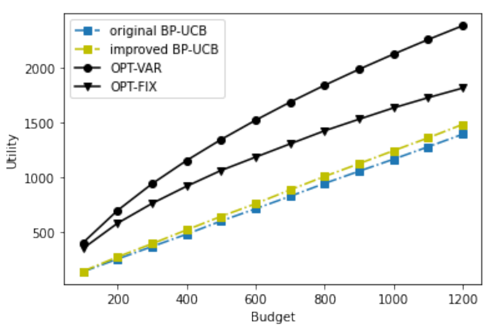
Figure 4: numerical results of various online mechanisms

Figure 4 shows the performance of different online mechanisms for crowdsourcing problem with the increase of the initial budget on the same dataset. We select the users pool size to be varied from 1200 to 3000, which is in a reasonable range respect to the budgets. In Figure 4a and Figure 4b, we observe that both BP-UCB and BP-DGREEDY mechanisms will converge to some fix utility with the increase of the initial budget. This is due to the users pool is exhausted before the crowdsourcer using all the budget. In this sense, the crowdsourcer is too 'conservative' in choosing the offered price. In contrast, in Figure 4c and Figure 4d, since the users pool size is large enough, all the online mechanisms can use their budget up before the users pool is exhausted. In these cases, we observe that all the online mechanisms have a stable increase in utility with the increase in the initial budget.

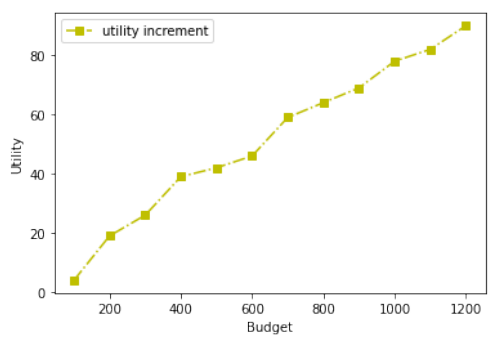
In comparison, we observe that the auction mechanism has a better performance over the posted-price mechanism. This result is in our expectation since the auction mechanism does require the truthful bids from the users. We can also observe that the auction mechanism approximate the offline fixed price benchmark (OPT-FIX) very closely, which is the optimal performance for the posted-price mechanism that can be achieved.

6.2 Improved BP-UCB v.s. Original BP-UCB

We also compare our improved BP-UCB with the original BP-UCB introduced by Singla. We keep the same experiment environment with the previous section and we set the users pool size to be 3000.



(a) comparison with original BP-UCB



(b) increased utility over original BP-UCB

Figure 5: numerical results of improved BP-UCB

Figure 5a and Figure 5b indicates the numerical results of our improved BP-UCB. We observe that our improved BP-UCB mechanism has a minor utility increment over the original BP-UCB mechanism. This result is reasonable since we keep the main structure of our mechanism the same as the original version. We only change the way of including the price and budget information in the process of updating UCB. Nevertheless, these experiments help us fully understand the BP-UCB mechanism and further improve it.

7 Conclusion

In this project, we review two different mechanisms for the online crowdsourcing problem in both posted-price model and auction model. We implement those online mechanisms with the corresponding offline benchmarks and conduct massive numerical experiments to compare their performance on the same dataset. Based on our understanding, we further improve the posted-price mechanisms and test their performance. In addition, we use a new method in proving truthfulness in the auction mechanism. By conducting this project, we have gained a full understanding on the current related research of online crowdsourcing and the corresponding mechanisms.

Acknowledgement

We are very grateful to Prof. Christian Kroer for his valuable suggestions on our research direction and patient guidance during the whole project.

References

[1] Jeffrey A Burke, Deborah Estrin, Mark Hansen, Andrew Parker, Nithya Ramanathan, Sasank Reddy, and Mani B Srivastava. Participatory sensing. 2006.

[2] Enrique Estellés-Arolas and Fernando González-Ladrón-de Guevara. Towards an integrated crowdsourcing definition. *Journal of Information science*, 38(2):189–200, 2012.

[3] Xiaoying Gan, Yuqing Li, Yixuan Huang, Luoyi Fu, and Xinbing Wang. When crowdsourcing meets social iot: An efficient privacy-preserving incentive mechanism. *IEEE Internet of Things Journal*, 6(6):9707–9721, 2019.

[4] Linghe Kong, Kui Ren, Muhammad Khurram Khan, Qi Li, Ammar Rayes, Merouane Debbah, and Yuichi Nakamura. Sustainable incentive mechanisms for mobile crowdsensing. *IEEE Communications Magazine*, 55(3):60–61, 2017.

[5] Gang Li and Jun Cai. An online incentive mechanism for crowdsensing with random task arrivals. *IEEE Internet of Things Journal*, 7(4):2982–2995, 2020.

[6] Dan Peng, Fan Wu, and Guihai Chen. Data quality guided incentive mechanism design for crowdsensing. *IEEE transactions on mobile computing*, 17(2):307–319, 2017.

[7] Adish Singla and Andreas Krause. Truthful incentives in crowdsourcing tasks using regret minimization mechanisms. In *Proceedings of the 22nd international conference on World Wide Web*, pages 1167–1178, 2013.

[8] Boya Song, Hamed Shah-Mansouri, and Vincent WS Wong. Quality of sensing aware budget feasible mechanism for mobile crowdsensing. *IEEE Transactions on Wireless Communications*, 16(6):3619–3631, 2017.

[9] Araz Taeihagh. Crowdsourcing, sharing economies and development. *Journal of Developing Societies*, 33(2):191–222, 2017.

[10] Xiong Wang, Zhe Liu, Xiaohua Tian, Xiaoying Gan, Yunfeng Guan, and Xinbing Wang. Incentivizing crowdsensing with location-privacy preserving. *IEEE Transactions on Wireless Communications*, 16(10):6940–6952, 2017.

[11] Jiapeng Xie, Shuo Yang, Fan Wu, Xiaofeng Gao, and Guihai Chen. A strategy-proof budget feasible online mechanism for crowdsensing with time-discounting values. In *2016 IEEE Global Communications Conference (GLOBECOM)*, pages 1–6. IEEE, 2016.

[12] Dong Zhao, Xiang-Yang Li, and Huadong Ma. Budget-feasible online incentive mechanisms for crowdsourcing tasks truthfully. *IEEE/ACM Transactions on Networking*, 24(2):647–661, 2014.