Multi-Armed Bandits and Crowdsourcing

Submitted as part of Honours Project (Semester 3)

Debojit Das
Machine Learning Lab

Sujit P Gujar
Machine Learning Lab

Abstract

In this report, we talk about Multi-Armed Bandits (MABs) and their applications in Crowdsourcing. We summarise a few papers on these topics and give a brief overview of these papers.

Contents

1

Introduction

A crowdsourcing platform comprises of a lot of workers. Given a task, every worker has an expected return on the task. The return may vary from task to task (depending on factors like area of expertise, hardness of task, etc.) for every person. Also, the return may vary from person to person for any given task, depending on factors like how often a person makes mistakes and what their area of expertise is. Interestingly, even the same worker doing a particular task may give varying returns due to the stochastic nature of humans. This is why it is justified to think of a crowdsourcing system as a MAB where every worker is an arm of the bandit.

We will now study some useful terms that will be useful in the course of the report.

1.1 Multi-Armed Bandits (MAB)

The MABs we consider consist of arms, with unknown *stochastic rewards*, and every turn we consider which arm to play so as to maximise the total reward at the end. We can choose to play an arm that has the highest expected reward at that point of time, thereby exploiting it; or we can choose to explore the other arms in hopes for finding their expected reward to be higher.

In its most basic formulation, a K-armed bandit problem is defined by random variables $X_{i,n}$ for $1 \leq i \leq K$ and $n \geq 1$, where each i is the index of a gambling machine (i.e., the arm of a bandit). In the n-th round, if the i-th arm is pulled, a reward of $X_{i,n}$ is generated. These $X_{i,n}$ are independently and identically distributed (over all rounds) according to an unknown law with unknown expectation μ_i . Independence also holds for rewards across machines; i.e., $X_{i,s}$ and $X_{j,t}$ are independent (and usually not identically distributed) $\forall 1 \leq i < j \leq K$ $\forall s, t \geq 1$.[2]

Given a MAB formulation, the goal is to come up with an algorithm that decides the order in which the arms should be pulled in order to maximise the total expected reward.

We define *regret* as the expected loss in reward on following our algorithm as compared to the reward we would have obtained by following some optimal (in terms of total expected reward) strategy. More formally, for a stochastic MAB, the regret $R(A)$ after n rounds on following algorithm A is given by

$$R_A(n) = \mu^* n - \sum_{i=1}^{K} \mu_i T_i(n)$$

, where $\mu^* = \max_{1 \leq i \leq K} \mu_i$, μ_i is the expected reward on pulling arm i, and $T_i(n)$ is the number of times arm i was pulled in n rounds. Our objective is essentially to minimise the expected regret after T rounds, i.e., choose A that minimises $R_A(T)$.

At any given point of time, we can decide to either play the arm which has the highest average reward till that time hence *exploiting* our current knowledge, or choose to *explore* the other arms to get a better estimate since on updating, other arms may turn out to have higher expected reward.

1.2 Crowdsourcing

Crowdsourcing is a task allocation concept wherein a task can be outsourced to workers who are chosen from a crowd instead of being performed by a designated agent, which is often suitable for tasks that are trivial for humans but difficult for computers, such as classification tasks. Compared with traditional task allocation, the advantages of crowdsourcing include the following: faster completion speed, lower costs, higher accuracy, and completion of tasks that computers cannot perform.

1.3 MABs in Crowdsourcing

There are a lot of parallels between MABs and Crowdsourcing. Every arm of the bandit being pulled can be thought of as an agent performing a task. Every agent performs differently, similar to how every arm has a unique expected reward associated with it. This is why MABs find a lot of use in crowdsourcing settings.

1.4 Heterogeneous Tasks

Often the assumption that a task is identical to every other task will not be a fair assumption. In some settings, tasks can be of varying difficulty, or may require different skill-sets. We say that the tasks are heterogeneous.

1.5 Heterogeneous Workers

Similarly, often the assumption that a worker is identical to every other worker will not be a fair assumption. Workers may have different skills, and some may be more efficient than others at a given task. Then we say that the workers are heterogeneous.

Having discussed few basics, now we present summaries of few selected recent papers which use MAB techniques in crowdsourcing.[1]

MAB-based Papers

2

Thompson Sampling for Complex Bandit Problems

[Read3]

2.1 Basic Details

This paper, authored by Aditya Gopalan, Shie Mannor, Yishay Mansour, was published in ICML in 2014.

2.2 Problem Addressed

The paper considers a stochastic multi-armed bandit problems with complex actions over a set of basic arms, where the decision maker plays a complex action rather than a basic arm in each round. The reward of the complex action is some function of the basic arms' rewards, and the feedback observed may not necessarily be the reward per-arm. The objective of the paper is to prove and provide regret bounds in certain selected settings.

2.3 Solution Proposed

- The authors prove a general regret bound for Thompson sampling in complex bandit settings involving parameter, action and observation spaces and a likelihood function over them
- They provide a better regret bound based on marginal KL-divergences
- They consider the particular setting where we can only observe the MAX individual value of reward of chosen arms and propose and prove a regret bound

2.4 Claims

- The regret bound for the general setting scales logarithmically with time and the pre-constant for this logarithmic scaling can be explicitly characterized in terms of the bandit's KL divergence geometry and represents the information complexity of the bandit problem.
- A better regret bound for the same based on marginal KL-divergences
- A logarithmic regret bound for the setting where we can only observe the MAX individual value of reward of chosen arms with a significantly lower constant

2.5 Previous Work

Bayesian ideas for the multi-armed bandit date back nearly 80 years ago to the work of W. R. Thompson [10], who introduced an elegant algorithm based on posterior sampling. Recent work has shown frequentist-style regret bounds for Thompson sampling in the standard bandit model [1] and Bayes risk bounds in the purely Bayesian setting. Regarding bandit problems with actions/rewards more complex than the basic MAB, a related line of work is that of linear bandit optimization [5]

2.6 Novel Ideas

The authors focus on the performance of Thompson setting in a general action/feedback model (like the MAX reward function), and show novel frequentist regret bounds that account for the structure of complex actions.

3

Budget-Constrained Multi-Armed Bandits with Multiple Plays

[Read7]

3.1 Basic Details

This paper, authored by Datong P. Zhou and Claire J. Tomlin, was published in AAAI in 2018.

3.2 Problem Addressed

The paper studies the multi-armed bandit problem with multiple plays and a budget constraint for both the stochastic and the adversarial setting. At each round, exactly K out of N possible arms have to be played. In addition to observing the individual rewards for each arm played[2], the player also learns a vector of costs which has to be covered with an a-priori defined budget B. The objective is to derive algorithms that minimize regret for these settings, given that the reward and cost distribution of the arms are unknown.

3.3 Solution Proposed

- The authors provide an algorithm (UCB-MB) for the stochastic setting and prove its regret bound

[1] Note that these summaries represent our understanding of the papers; if any errors found, we are open to revisit.

[2] The paper does not address how the individual rewards combine. However, the setting is semi-bandit feedback

- They also provide an algorithm (Exp3.M.B) for the adversarial setting and prove its regret bound
- They adjust Exp3.M.B to handle budget constraints and thereby provide algorithm Exp3.1.M.B, and prove its regret bound

3.4 The main algorithms

3.4.1 UCB-MB

Given a bandit with N distinct arms, each arm indexed by $i \in [N]$ is associated with an unknown reward and cost distribution with unknown means $0 < \mu_r^i \leq 1$ and $0 < c_{min} \leq \mu_c^i \leq 1$, respectively. Realizations of costs $c_{i,t} \in [c_{min}, 1]$ and rewards $r_{i,t} \in [0, 1]$ are independently and identically distributed. At each round t, the decision maker plays exactly K arms (the K arms associated with the K largest confidence bounds) and subsequently observes the individual costs and rewards only for the played arms, which corresponds to the semi-bandit setting. Before the game starts, the player is given a budget $0 < B \in \mathbb{R}_+$ to pay for the materialized costs $\{c_{i,t} | i \in a_t\}$, where a_t denotes the indexes of the K arms played at time t. The game terminates as soon as the sum of costs at round t, namely $\sum_{j \in a_t} c_{j,t}$ exceeds the remaining budget.

3.4.2 Exp3.M.B

The adversarial case makes no assumptions on the reward and cost distributions whatsoever. They consider the extension of the classic setting as in (Uchiya, Nakamura, and Kudo 2010), where the decision maker has to play exactly $1 \leq K \leq N$ arms. For each arm i played at round t, the player observes the reward $r_i(t) \in [0, 1]$ and, unlike in previous settings, additionally the cost $0 < c_{min} < c_i(t) < 1$. The player is given a budget $B > 0$ to pay for the costs incurred, and the algorithm terminates after $\tau_{\mathcal{A}}(B)$ rounds when the sum of materialized costs in round $\tau_{\mathcal{A}}(B)$ exceeds the remaining budget.

Algorithm Exp3.M.B maintains a set of time-varying weights $\{w_i(t)\}_{i=1}^N$ for all arms, from which the probabilities for each arm being played at time t are calculated. The probabilities $\{p_i(t)\}_{i=1}^N$ sum to K (because exactly K arms need to be played), which requires the weights to be capped at a value $v_t > 0$ such that the probabilities $\{p_i(t)\}_{i=1}^N$ are kept in the range [0, 1]. In each round, the player draws a set of distinct arms at of cardinality $|a_t| = K$, where each arm has probability $p_i(t)$ of being included in a_t. At the end of each round, the observed rewards and costs for the played arms are turned into estimates $\hat{r}_i(t)$ and $\hat{c}_i(t)$. Arms with $w_i(t) < v_t$ are updated according to $(\hat{r}_i(t) - \hat{c}_i(t))$, which assigns larger weights as $\hat{r}_i(t)$ increases and $\hat{c}_i(t)$ decreases. Regret achieved by Exp3.M.B depends on the upper bound for G_max, where G_max is the optimal cumulative reward under the optimal set.

3.4.3 Exp3.1.M.B

If no upper bound on G_{max} exists, Algorithm Exp3.M.B. needs to be modified. The weights have to be updated differently. As in Algorithm Exp3.1 in (Auer et al. 2002), the authors use an adaptation of Algorithm Exp3.M.B, which they call Exp3.1.M.B. Algorithm Exp3.1.M.B utilizes Algorithm Exp3.M.B as a subroutine in each epoch until termination.

3.5 Claims

- UCB-MB achieves a regret of O($NK^4 logB$)
- Exp3.M.B achieves a regret of O($\sqrt{BNlog(N/K)}$)
- Exp3.1.M.B achieves a regret of O($\sqrt{NKTlog(N/K)} + \frac{N-K}{N-1} log(\frac{NT}{\delta}) + K^2 \sqrt{NT \frac{N-K}{N-1} log(\frac{NT}{\delta})}$) with a probability of $1 - \delta$, where T is the total number of rounds

3.6 Previous Work

In the extant literature, attempts to make sense of a cost component in MAB problems occur in (Tran-Thanh et al. 2010) and (Tran-Thanh et al. 2012), who assume time-invariant costs and cast the setting as a knapsack problem with only the rewards being stochastic. The papers that are closest to the addressed setting are [3] and [Read7]. The former investigates the stochastic case with a resource consumption. The latter paper focuses on the stochastic case, but does not address the adversarial setting at all.

3.7 Novel Ideas

Even though [3] investigates the stochastic case with a resource consumption, it obtains a regret bound of O($\sqrt{B}$) and this paper obtains a regret of O($logB$). This is also the first paper that addresses the adversarial budget constrained case.

4

Combinatorial Multi-Armed Bandit Based Unknown Worker Recruitment in Heterogeneous Crowdsensing

[Read2]

4.1 Basic Details

This paper, authored by Guoju Gao, Jie Wu, Mingjun Xiao, and Guoliang Chen, was published in INFOCOM in 2020.

4.2 Problem Addressed

In this paper, the authors focus on the unknown worker recruitment problem in mobile crowdsensing, where workers' sensing qualities are unknown a-priori. They consider the scenario of recruiting workers to complete some continuous sensing tasks. The whole process is divided into multiple rounds. In each round, every task may be covered by more than one recruited workers, but its completion quality only depends on these workers' maximum sensing quality. Each recruited worker will incur a cost, and each task is attached a weight to indicate its importance. The objective is to determine a recruiting strategy to maximize the total weighted completion quality under a limited budget. They extend the problem to the case where the workers' costs are also unknown.

4.3 Solution Proposed

- They turn the unknown worker recruitment problem for heterogeneous Mobile CrowdSensing (MCS) systems into a K-arm CMAB problem
- They propose an extended UCB based arm-pulling strategy to solve the CMAB problem and design the corresponding unknown worker recruitment online algorithm (UWR)
- They also study an extended case where both the sensing qualities and the costs of workers are unknown, and devise another algorithm (EUWR)
- They conduct extensive simulations on real-world traces to evaluate the significant performance of the algorithms

4.4 Claims

- UWR has a regret bound of O($NLK^3ln(B)$), where B, N, and L are the budget, the number of workers, and the number of options of each of the K workers, respectively.
- EUWR has a regret bound of O($NLK^3ln(NMB)$), where M is the number of tasks

4.5 The main algorithms

4.5.1 UWR

The platform selects the first options of all workers with the minimum cost to initialize several parameters, such as $n_i(t)$ (number of times worker i has been selected until the t-th round) and $\bar{q}_i(t)$ (average learned quality value for i until the t-th round). Then, the platform selects K workers according to the UCB-based qualities and the proposed selection criterion. To meet the constraint that at most one option of a worker can be selected in a round, let $\mathcal{P}^{t'}$ denote the set of not satisfying options. Then, the option with the largest ratio of the marginal UCB-based quality function value and cost is selected from the set $\mathcal{P} \setminus \mathcal{P}^{t'}$. The platform decides whether to terminate the algorithm based on the remaining budget. If the remaining budget is enough, the recruited workers perform the corresponding tasks, and send the sensing results to the platform. The platform updates the worker profiles. The remaining budget and total achieved weighted quality are accordingly updated.

4.5.2 EUWR

EUWR is very similar to UWR. The key difference between UWR and EUWR is that in EUWR, the selection criterion takes the obtained quality and cost values (as a ratio of quality to cost) into consideration simultaneously (whereas in UWR, only the quality was under consideration).

4.6 Previous Work

So far, there have been lots of researches on the worker recruitment problem in MCS. Only a few of them consider the unknown sensing qualities or costs in MCS systems. [8] studies to maximize the total sensing revenue for the budgeted robust MCS. [11] investigates how to select the most informative contributors with unknown costs for budgeted MCS. The most related works are [7], [Read7], in which they study the top K bandit selection problem. [7] proposes an algorithm which can achieve a good regret bound and only requires linear storage and polynomial computation. [Read7] designs a UCB-based algorithm with a O(NK^4logB) regret bound.

4.7 Novel Ideas

Although a lot of work has been done in the homogeneous setting, this paper studies the unknown worker recruitment problem for the heterogeneous MCS system. Especially, it involves a budget-limited maximum weighted coverage problem.

5

Understanding Workers, Developing Effective Tasks, and Enhancing Marketplace Dynamics

[Read4]

5.1 Basic Details

This paper, authored by Ayush Jain, Akash Das Sarma, Aditya Parameswaran and Jennifer Widom, was published in ACM VLDB in 2017.

5.2 Problem Addressed

This paper presents an experimental analysis of a dataset comprising over 27 million microtasks performed by over 70,000 workers issued to a large crowdsourcing marketplace [3] between 2012-2016. They shed light on three crucial aspects of crowdsourcing: task design, marketplace dynamics and worker behavior.

5.3 Preliminaries

- *Task Design* refers to helping requesters understand what constitutes an effective task, and how to go about designing one
- *Marketplace Dynamics* refers to helping marketplace administrators and designers understand the interaction between tasks and workers, and the corresponding marketplace load
- *Worker Behavior* refers to understanding worker attention spans, lifetimes, and general behavior, for the improvement of the crowdsourcing ecosystem as a whole

5.4 Solution Proposed

- The authors had access to a dataset (provided at batch-level) consisting of tasks issued on the marketplace from 2012–16
- They enriched the available data by generating three additional types of task attribute data: manual labels, design parameters and performance metrics
- To understand the worker supply and task demand interactions, they examined some basic statistics of the marketplace, and looked specifically at *task instance arrival distribution* and *worker availability*

5.5 Previous Work

The only paper that has performed an extensive analysis of crowdsourcing marketplace data is the recent paper by Difallah et al. [6]. This paper analyzed the data obtained via crawling a public crowdsourcing marketplace (in this case Mechanical Turk). Unfortunately, this publicly visible data provides a restricted view of how the marketplace is functioning, since the worker responses, demographics and characteristics of the workers, and the speed at which these responses are provided are all unavailable.

5.6 Novel Ideas

Their dataset has information about individual worker responses, hence they are able to study the question of task "effectiveness" unlike the papers in the past. Also, their crowdsourcing marketplace recruits workers from a collection of labor sources, making it a crowdsourcing "intermediary" or "aggregator", and allows for a number of interesting additional analyses.

6

What Prize is Right? How to Learn the Optimal Structure for Crowdsourcing Contests

[Read5]

6.1 Basic Details

This paper, authored by Nhat Van-Quoc Truong, Sebastian Stein, Long Tran-Thanh and Nicholas R. Jennings, was published in PRICAI in 2019.

6.2 Problem Addressed

In crowdsourcing, one effective method for encouraging participants to perform tasks is to run contests where participants compete against each other for rewards. These contests vary in their structure and parameters. With a given budget and a time limit, choosing incentives (i.e., contest structures with specific parameter values) that maximise the overall utility is not trivial, as their respective effectiveness in a specific project is usually unknown a priori. The objective is to find an appropriate way for an autonomous agent (i.e., a computer programme) to select an effective incentive in such a crowdsourcing project. They refer to this as the incentive selection problem (ISP)

6.3 Solution Proposed

The paper models the problem as a machine with N arms (corresponding to N incentives), pulling an arm (offering the corresponding incentive to a group of participants) incurs a fixed cost (attached to the arm) and delivers a random utility (e.g., the number of tasks completed) drawn from an unknown distribution. The objective in an MAB problem is to find a policy that maximises the total utility within a given budget before a deadline.

The authors combine the two (online learning with MABs and tuning with Bayesian Optimization) to deal with the ISP. By so doing, they decouple a complicated problem (with both learning the best structure and tuning its parameters) into two simple problems and deal with these in a learning process. They propose an algorithm, BOIS, to solve the ISP effectively by combining an MAB approach to learn the contest structures and Bayesian Optimization (BO) to tune the parameters of the structures.

[3] the name of the marketplace is not available in the paper

6.4 The BOIS algorithm

The idea of BOIS is using an MAB approach to deal with the learning problem (i.e., identifying the best cluster) and using BO with Gaussian processes to tackle the tuning problem (i.e., finding the optimal values of the parameters of a cluster). In each period (except the first one), it selects the incentive whose value of the acquisition function corresponding to this incentive is the largest compared to those of the other incentives in all clusters.

The general idea of tuning parameter values of a contest structure (i.e., finding the real best incentive in a cluster) using BO with Gaussian processes is the following. In each period, based on the incentives sampled in the previous periods, BOIS estimates the mean utilities of the incentives in the cluster using Gaussian process regression (GPR). Then, it calculates the UCBs of the incentives. After that, the incentive with the highest UCB will be the candidate to be applied next in the cluster. BOIS will then choose the candidate incentive in the cluster which has the highest UCB to be applied in that period. In order for the algorithm to use BO, it must have initial estimates of the incentives in each cluster. Therefore, in the first period (i.e., period 1), it samples several incentives, in order to obtain good estimates of the incentives. This step is referred to as the *sampling step*. Then, in each of the next periods (except the last one), it applies the most promising incentive, i.e., the incentive with the largest UCB. After that, it updates the UCBs of the incentives in the same cluster (i.e., C_i if $a \in C_i$). This step is referred to as the *stepped exploitation step*. Finally, in the last period it applies the best incentive with the remaining budget. This step is called the *pure exploitation step*, as it simply exploits the best incentive after exploring in the previous periods.

BOIS only uses $\epsilon_1 B$ for sampling. This amount might not be enough to sample all the candidate incentives. Therefore, BOIS simply iterates over the clusters and at each cluster it chooses a random (without repetition) incentive from this set. BOIS stops sampling when the budget for sampling is exceeded. Then, BOIS sets the budget for stepped exploitation, a specific portion of the residual budget which is identified by ϵ_2. us, This step can be considered as both exploiting (choosing the incentives whose estimates are high) and exploring (choosing the incentives whose potential to be the real best one are high). The remaining budget is used for the pure exploitation step.

6.5 Claims

- BOIS solves the ISP problem effectively
- BOIS is generally more effective compared to the state-of-the-art approaches

6.6 Previous Work

Much work has taken a game-theoretic approach to investigate the optimal (or efficient) design of contests in general and crowdsourcing contests in particular. It tries to answer the questions of how to distribute the prizes (number of prizes and their values) in contests (Luo et al., 2015; Cavallo and Jain, 2012; Moldovanu and Sela, 2001). A number of studies about budgeted MABs have been conducted, such as Badanidiyuru et al. (2018), Ho et al. (2016), and Tran-Thanh et al. (2010).

6.7 Novel Ideas

Existing studies do not consider factors related to the participants' intrinsic motivation that might affect their behaviour such as the project purpose or the task nature. None of them consider all important characteristics of the ISP, such as the budget constraints, multidimensional structure of the incentives, correlations between the arms, and the group-based nature of the arm.

7

Efficient Budget Allocation with Accuracy Guarantees for Crowdsourcing Classification Tasks

[Read6]

7.1 Basic Details

This paper, authored by Long Tran-Thanh, Matteo Venanzi, Alex Rogers and Nicholas R. Jennings, was published in AAMAS in 2013.

7.2 Problem Addressed

In this paper, the authors address the problem of budget allocation for redundantly crowdsourcing a set of classification tasks where a key challenge is to find a trade–off between the total cost and the accuracy of estimation.

7.3 Solution Proposed

- They introduce the problem of budget allocation for crowdsourcing classification tasks, in which the goal is to minimise the error of the estimated answers for a finite number of tasks, with respect to a budget limit
- They develop CrowdBudget, an algorithm that, combining with a fusion method, proveably achieves an efficient bound on the estimation error, which significantly advances the best known results

- They compare the performance of CrowdBudget with existing algorithms through extensive numerical evaluations on real–world data taken from a prominent crowdsourcing system

7.4 The CrowdBudget Algorithm

n_k denotes the number of users the agent aims to assign to task k. It first pre–sets to $n_k = \left\lfloor \frac{B}{c_k^2 \sum_{j=1}^{K} \frac{1}{c_j}} \right\rfloor$. The agent also maintains B^r that denotes the residual budget, which is initially set to be B. After each pre–set of n_k, B^r is decreased by $n_k c_k$. where c_k is the cost that user k has to be paid. Next, if $B^r > 0$, the agent sequentially increases the number of allocated users for each task k by 1, until the original budget is exceeded. This phase guarantees that the budget is fully used. Following this, the agent redundantly submits the tasks to the system, and once it receives the responses from the users, it uses an MV–efficient fusion method to estimate the answers to each of the tasks.

7.5 Claims

- CrowdBudget can achieve at most $\max\{0, \frac{K}{2} - O(\sqrt{B})\}$ estimation error with high probability, where K is the number of tasks and B is the budget size

- They demonstrate that their algorithm typically outperforms the state–of–the–art by achieving up to 40% lower estimation error.

7.6 Previous Work

Wellinder et al. proposed a multidimensional model of users in order to estimate the accuracy of a particular user's answer, and thus, to improve the estimation of the ground truth. A number of works used Bayesian learning techniques to predict the users' responses, such as the work of Kamar et al. and the IBCC algorithm. Bachrach et al. relied on a machine learning based aggregator to derive an efficient estimation of the correct answer. More related to this work is CrowdScreen, an algorithm proposed by Parameswaran et al. [9], that aims to find an optimal dynamic control policy with respect to both total cost and total estimation error over a finite set of tasks.

7.7 Novel Ideas

None of the previous works address the challenge of having different costs for different classification tasks. In addition, as per the other aforementioned approaches, there are no guarantee on performance.

8

Combinatorial Multi Armed Bandit with General Reward Functions

[Read1]

8.1 Basic Details

This paper, authored by Wei Chen, Wei Hu, Fu Li, Jian Li, Yu Liu and Pinyan Lu, was published in NIPS in 2016.

8.2 Problem Addressed

This paper studies the stochastic combinatorial multi-armed bandit (CMAB) framework that allows a general nonlinear reward function, whose expected value may not depend only on the means of the input random variables but possibly on the entire distributions of these variables.

8.3 Solution Proposed

- They generalize the CMAB framework to allow a general reward function whose expectation may depend on the entire distributions of the input random variables

- The propose the stochastically dominant confidence bound (SDCB) algorithm to achieve efficient learning in this framework with near-optimal regret bounds, even for arbitrary outcome distributions

- They give the first polynomial time approximation scheme (PTAS) for the offline K-MAX problem

8.4 The SDCB algorithm

Algorithm 1 SDCB (Stochastically dominant confidence bound)

```
Throughout the algorithm, for each arm $i \in [m]$, maintain: (i) a counter $T_i$ which stores the number of times arm $i$ has been played so far, and (ii) the empirical distribution $\hat{D}_i$ of the observed outcomes from arm $i$ so far, which is represented by its CDF $\hat{F}_i$

// Initialization
for $i$ in 1 to $m$ do
    // Action in the $i$-th round
    Play a super arm $S_i$ that contains arm $i$
    Update $T_j$ and $\hat{F}_j$ for each $j \in S_i$
end
for $t$ in $m+1, m+2, \dots$ do
    // Action in the $t$-th round
    For each $i \in [m]$, let $\underline{D}_i$ be a distribution whose CDF $F_i$ is
        $F_i(x) = \begin{cases} \max \underline{\hat{F}}_i(x) - \sqrt{\frac{3lnt}{2T_i}}, 0 & 0 \leq x < 1 \\ 1, & x = 1 \end{cases}$
    Play the super arm $S_t \leftarrow \text{Oracle}(D)$, where $\underline{D} = \underline{D}_1 \times \underline{D}_2 \times \cdots \times \underline{D}_m$
    Update $T_j$ and $\hat{F}_j$ for each $j \in S_t$
end
```

The algorithm starts with m initialization rounds in which each arm is played at least once. In the t-th round ($t > m$), the algorithm consists of three steps. First, it calculates for each $i \in [m]$ a distribution $\underline{D}_i$ whose CDF $\underline{F}_i$ is obtained by lowering the CDF $\hat{F}_i$. The second step is to call the -approximation oracle with the newly constructed distribution $\underline{D} = \underline{D}_1 \times \underline{D}_2 \times \cdots \times \underline{D}_m$ as input, and thus the super arm S_t output by the oracle satisfies $r_{\underline{D}}(S_t) \geq \alpha \max_{S \in F}\{r_{\underline{D}}(S)\}$. Finally, the algorithm chooses the super arm S_t to play, observes the outcomes from all arms in St, and updates T_j's and $\hat{F}_j$'s accordingly for each $j \in S_t$.

8.5 Claims

- SDCB estimates the distributions of underlying random variables and their stochastically dominant confidence bounds
- SDCB can achieve O($logT$) distribution-dependent regret and $\tilde{O}(\sqrt{T})$ distribution-independent regret, where T is the time horizon
- For K-MAX, they provide the first PTAS for its offline problem, and give the first $\tilde{O}(\sqrt{T})$ bound on the $(1-\epsilon)$- approximation regret of its online problem, for any $\epsilon > 0$

8.6 Previous Work

Most relevant to our work are studies on CMAB frameworks, among which some focus on linear reward functions while some do look into nonlinear reward functions. Chen et al. [4] look at general non-linear reward functions and Kveton et al. consider specific non-linear reward functions in a conjunctive or disjunctive form, but both papers require that the expected reward of playing a super arm is determined by the expected outcomes from base arms.

8.7 Novel Ideas

They generalize the existing CMAB framework with semi-bandit feedbacks to handle general reward functions. They also provide a result on the K-MAX problem where they consider general distribution from base arms (unlike Bernoulli outcomes, which has been done before).

9

Brief Overview

Paper	Appeared in	Problem Statement	Solution Proposed
Thompson Sampling for Complex Bandit Problems	ICML, 2014	To prove and provide regret bounds in certain selected complex-action stochastic multi-armed bandit settings	They prove the said regret bounds
Budget-Constrained Multi-Armed Bandits with Multiple Plays	AAAI, 2018	To derive algorithms that minimize regret for multi-armed bandit problem with multiple plays and a budget constraint for both the stochastic and the adversarial setting	They propose algorithms UCB-MB, Exp3.M.B and Exp3.1.M.B
Combinatorial Multi-Armed Bandit Based Unknown Worker Recruitment in Heterogeneous Crowdsensing	INFOCOM, 2020	To determine a recruiting strategy to maximize the total weighted completion quality under a limited budget in the case of unknown worker recruitment in heterogeneous crowdsensing	They propose algorithms UWR and EUWR
Understanding Workers, Developing Effective Tasks, and Enhancing Marketplace Dynamics	ACM VLDB, 2017	This is an experimental analysis to shed light on three crucial aspects of crowdsourcing: task design, marketplace dynamics and worker behavior	They examined some basic statistics of the marketplace, and looked specifically at task instance arrival distribution and worker availability
What Prize is Right? How to Learn the Optimal Structure for Crowdsourcing Contests	PRICAI, 2019	The objective is to find an appropriate way for an autonomous agent (i.e., a computer programme) to select an effective incentive in a crowdsourcing project.	They propose the algorithm BOIS
Efficient Budget Allocation with Accuracy Guarantees for Crowdsourcing Classification Tasks	AAMAS, 2013	They address the problem of budget allocation for redundantly crowdsourcing a set of classification tasks where a key challenge is to find a trade–off between the total cost and the accuracy of estimation	They propose CrowdBudget
Combinatorial Multi Armed Bandit with General Reward Functions	NIPS, 2016	They study the stochastic CMAB framework that allows a general nonlinear reward function, whose expected value may depend on the entire distributions of input random variables	They propose SDCB

Table 1: **Brief Description of MAB Papers**

Paper	**Combinatorial**	**Contextual**	**Budgeted**	**Task**	**Worker**	**Other Key Features**
Thompson Sampling for Complex Bandit Problems	Non-linear combinatorial function (MAX) with bandit feedback	No	No	Homogeneous	Heterogeneous	TS, not UCB. For MAX, they consider a set of M arms
Budget-Constrained Multi-Armed Bandits with Multiple Plays	Exactly K out of N arms	No	Total budget B	Homogeneous	Homogeneous	Every round also learn a vector of costs which has to be covered
Combinatorial Multi-Armed Bandit Based Unknown Worker Recruitment in Heterogeneous Crowdsensing	Exactly K out of N arms	No	Total budget B	Heterogeneous	Heterogeneous	Each worker can do at most L tasks. If a worker is selected, he will do every task he can perform. Offline task allocation
Understanding Workers, Developing Effective Tasks, and Enhancing Marketplace Dynamics	-	-	-	-	-	Experimental Paper
What Prize is Right? How to Learn the Optimal Structure for Crowdsourcing Contests	No	Yes, based on incentive parameters	Total budget B	Heterogeneous	Heterogeneous	Uses MAB as well as BO. Three phase budget allocation: sampling, stepped exploitation, pure exploitation
Efficient Budget Allocation with Accuracy Guarantees for Crowdsourcing Classification Tasks	No	No	Total budget B	Heterogeneous	Homogeneous	For each user u and binary task k, there is an unknown Bernoulli distribution from which u samples his answer to task k
Combinatorial Multi Armed Bandit with General Reward Functions	Select at most K arms	No	No	Homogeneous	Homogeneous	Semi-bandit feedback. Treats offline stochastic optimization algorithm as an oracle, and integrates it into the online learning framework

Table 2: **Comparing MAB papers**

Acknowledgements

We acknowledge Mr. Kumar Abhishek (Machine Learning Lab) for discussions on MAB and crowdsourcing which greatly helped with the pacing of our work.

List of Papers Summarized

[Read1] Wei Chen, Wei Hu, Fu Li, Jian Li, Yu Liu, and Pinyan Lu. Combinatorial multi-armed bandit with general reward functions, 2018.

[Read2] Guoju Gao, Jie Wu, Mingjun Xiao, and Guoliang Chen. Combinatorial multi-armed bandit based unknown worker recruitment in heterogeneous crowdsensing. In *39th IEEE Conference on Computer Communications, INFOCOM 2020, Toronto, ON, Canada, July 6-9, 2020*, pages 179–188. IEEE, 2020.

[Read3] Aditya Gopalan, Shie Mannor, and Yishay Mansour. Thompson sampling for complex bandit problems, 2013.

[Read4] Ayush Jain, Akash Das Sarma, Aditya Parameswaran, and Jennifer Widom. Understanding workers, developing effective tasks, and enhancing marketplace dynamics: A study of a large crowdsourcing marketplace, 2017.

[Read5] Van Quoc Truong Nhat, Sebastian Stein, Long Tran-Thanh, and Nick Jennings. What prize is right? how to learn the optimal structure for crowdsourcing contests. In Abhaya Nayak and Alok Sharma, editors, *PRICAI 2019: Trends in Artificial Intelligence*, volume 1160, pages 85–97. Springer, August 2019.

[Read6] Long Tran-Thanh, Matteo Venanzi, Alex Rogers, and Nicholas R. Jennings. Efficient budget allocation with accuracy guarantees for crowdsourcing classification tasks. In *AAMAS '13 Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems (01/05/13)*, pages 901–908, May 2013.

[Read7] Datong P. Zhou and Claire J. Tomlin. Budget-constrained multi-armed bandits with multiple plays. *CoRR*, abs/1711.05928, 2017.

References

[1] Shipra Agrawal and Navin Goyal. Analysis of thompson sampling for the multi-armed bandit problem, 2012.

[2] Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Mach. Learn.*, 47(2–3):235–256, May 2002.

[3] Ashwinkumar Badanidiyuru, Robert D. Kleinberg, and Aleksandrs Slivkins. Bandits with knapsacks. *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, pages 207–216, 2013.

[4] Wei Chen, Yajun Wang, and Yang Yuan. Combinatorial multi-armed bandit and its extension to probabilistically triggered arms. *CoRR*, abs/1407.8339, 2014.

[5] Varsha Dani, Thomas Hayes, and Sham Kakade. Stochastic linear optimization under bandit feedback. pages 355–366, 01 2008.

[6] Djellel Eddine Difallah, Michele Catasta, Gianluca Demartini, Panagiotis G. Ipeirotis, and Philippe Cudré-Mauroux. The dynamics of micro-task crowdsourcing: The case of amazon mturk. In *Proceedings of the 24th International Conference on World Wide Web*, WWW '15, page 238–247, Republic and Canton of Geneva, CHE, 2015. International World Wide Web Conferences Steering Committee.

[7] Yi Gai, Bhaskar Krishnamachari, and Rahul Jain. Combinatorial network optimization with unknown variables: Multi-armed bandits with linear rewards, 2010.

[8] Kai Han, Chi Zhang, and Jun Luo. Taming the uncertainty: Budget limited robust crowdsensing through online learning. *IEEE/ACM Trans. Netw.*, 24(3):1462–1475, June 2016.

[9] Edwin Simpson, Stephen Roberts, Ioannis Psorakis, and Arfon Smith. *Dynamic Bayesian Combination of Multiple Imperfect Classifiers*, pages 1–35. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[10] WILLIAM R THOMPSON. ON THE LIKELIHOOD THAT ONE UNKNOWN PROBABILITY EXCEEDS ANOTHER IN VIEW OF THE EVIDENCE OF TWO SAMPLES. *Biometrika*, 25(3-4):285–294, 12 1933.

[11] Shuo Yang, Fan Wu, Shaojie Tang, Tie Luo, Xiaofeng Gao, Linghe Kong, and Guihai Chen. Selecting most informative contributors with unknown costs for budgeted crowdsensing. In *24th IEEE/ACM International Symposium on Quality of Service, IWQoS 2016, Beijing, China, June 20-21, 2016*, pages 1–6. IEEE, 2016.