A Multi-Arm Bandit Approach To Subset Selection Under Constraints

Extended Abstract

Ayush Deva
IIIT Hyderabad
ayushdeva97@gmail.com

Kumar Abhishek
IIIT Hyderabad
kumar.abhishek@research.iiit.ac.in

Sujit Gujar
IIIT Hyderabad
sujit.gujar@iiit.ac.in

ABSTRACT

We explore the class of problems where a central planner needs to select a subset of agents, each with different quality and cost. The planner wants to maximize its utility while ensuring that the average quality of the selected agents is above a certain threshold. When the agents' quality is known, we formulate our problem as an integer linear program (ILP) and propose a deterministic algorithm, namely DPSS that provides an exact solution to our ILP.

We then consider the setting when the qualities of the agents are unknown. We model this as a Multi-Arm Bandit (MAB) problem and propose DPSS-UCB to learn the qualities over multiple rounds. We show that after a certain number of rounds, τ, DPSS-UCB outputs a subset of agents that satisfy the average quality constraint with a high probability. Next, we provide bounds on τ and prove that after τ rounds, the algorithm incurs a regret of $O(\ln T)$, where T is the total number of rounds. We further illustrate the efficacy of DPSS-UCB through simulations. To overcome the computational limitations of DPSS, we propose a polynomial-time greedy algorithm, namely GSS, that provides an approximate solution to our ILP. We also compare the performance of DPSS and GSS through experiments.

ACM Reference Format:
Ayush Deva, Kumar Abhishek, and Sujit Gujar. 2021. A Multi-Arm Bandit Approach To Subset Selection Under Constraints: Extended Abstract. In *Proc. of the 20th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2021), Online, May 3–7, 2021*, IFAAMAS, 3 pages.

1 MODEL AND SOLUTION APPROACH

Consider a fixed set of $N = \{1, 2, \ldots, n\}$ agents producing a particular product. Each agent, i, has a cost of production, c_i, and capacity, k_i. The quality of the j^{th} unit of produce by agent i is denoted by Q_{ij}, which we model as a Bernoulli random variable, with mean q_i, where q_i is referred to as the quality of the agent. Upon procuring a unit from agent i, a central planner, C, receives a utility given by $r_i = Rq_i - c_i$, where R is the proportionality constant. C needs to procure units from these agents so as maximize its total utility, z, whilst ensuring that the expected average quality of the units produced, $q_{av} = \frac{\sum_{i \in N} x_i q_i}{\sum_{i \in N} x_i}$ is above a certain threshold $\alpha \in [0, 1]$. Here, x_i refers to the number of units procured from the i^{th} agent.

We model our problem as an Integer Linear Program (ILP) ([10]: Equation 1) and propose a dynamic-programming (DP) based algorithm, *DPSS*, to solve for it. For ease of exposition, we consider $k_i = 1$, since each unit of produce can be considered a separate agent and the proofs and discussion follows. Under *DPSS*, we categorize agents into four categories: i) S_1: Agents with $q_i \geq \alpha$ and $r_i \geq 0$, ii) S_2: Agents with $q_i < \alpha$ and $r_i \geq 0$, iii) S_3: Agents with $q_i \geq \alpha$ and $r_i < 0$, iv) S_4: Agents with $q_i < \alpha$ and $r_i < 0$. The agents are then selected as shown in Algorithm 1.

Algorithm 1 DPSS

Inputs: N, α, R, costs $\boldsymbol{c} = \{c_i\}_{i \in N}$, qualities $\boldsymbol{q} = \{q_i\}_{i \in N}$
Output: Quantities procured $\boldsymbol{x} = (x_1, \ldots, x_n)$
Initialization: $\forall i \in N$, $r_i = Rq_i - c_i$, $z = 0$
Segregate S_1,S_2,S_3,S_4 as described in Section 1
$\forall i \in S_1$, $x_i = 1$; $z = z + r_i$; $d = \sum_{i \in S_1} (q_i - \alpha)$
$\forall i \in S_4$, $x_i = 0$
$G = S_2 \cup S_3$; $\forall i \in G, d_i = q_i - \alpha$
function DP($i, d^{te}, x^{te}, x^\star, z^{te}, z^\star$)
    if $i == |G|$ and $d^{te} < 0$ **then return** $x^\star, z^\star$
    if $i == |G|$ and $d^{te} \geq 0$ **then**
        if $z^{te} > z^\star$ **then**
            $z^\star = z^{te}$; $x^\star = x^{te}$
        return $x^\star, z^\star$
    $x^\star, z^\star = DP(i + 1, d^{te}, [x^{te}, 0], x^\star, z^{te}, z^\star)$
    $x^\star, z^\star = DP(i + 1, d^{te} + d_i, [x^{te}, 1], x^\star, z^{te} + r_i, z^\star)$
    return $x^\star, z^\star$
x^G, z^G = DP(0,d,[ ],[ ],0,0)
$\forall i \in G, x_i = x_i^G$
return x

Unknown Qualities. We now consider a setting when q_i are *unknown* beforehand and can only be learned by selecting the agents. When online learning is involved, the stochastic multi-armed bandit (MAB) problem captures the exploration vs. exploitation trade-off effectively [1–5, 12–18]. Since we select multiple agents in a round, we model it as a Combinatorial MAB (CMAB) problem [6–9, 11] with semi-bandit feedback and quality constraint (QC). In our setting, QC depends on the qualities of the agents that are unknown, which makes it different from similar works like Jain et al. [14] where the authors present a bandit framework where the constraint depends on the cost of the agents which are known beforehand.

We propose an abstract framework, SS-UCB (Algorithm 2), for subset selection problem with QC. SS-UCB assumes that there exist an offline subset selection algorithm, SSA, (e.g., DPSS), which returns a super-arm that satisfies the QC corresponding to the qualities, cost and threshold provided as input to it. An algorithm under the SS-UCB framework proceeds in discrete rounds, $t = 1, \ldots, T$. In round t, C selects a super arm, $S^t = \{i \in N | x_i^t = 1\}$, where x_i^t refers to if arm i is selected at round t. The expected average quality of S^t is given by $q_{av}^t = \frac{1}{s^t} \sum_{i \in S^t} q_i$. Let $s^t = |S^t|$

Proc. of the 20th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2021), U. Endriss, A. Nowé, F. Dignum, A. Lomuscio (eds.), May 3–7, 2021, Online.

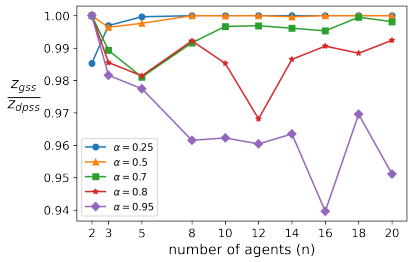

Figure 1: Relative Performance of GSS

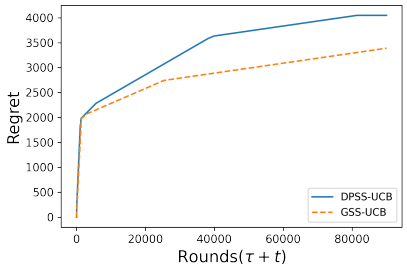

Figure 2: Regret incurred for $t > \tau$

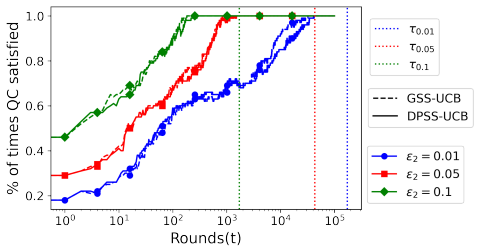

Figure 3: Constraint Satisfaction at each round

Algorithm 2 SS-UCB

1: **Inputs:** N, α, ϵ_2, R, costs $\boldsymbol{c} = \{c_i\}_{i \in N}$
2: For each agent i, maintain: w_i^t, q_i^t, $(\hat{q}_i^t)^+$
3: $\tau \leftarrow \frac{3 \ln T}{2\epsilon_2^2}$; $t = 0$
4: **while** $t \leq \tau$ **(Explore Phase) do**
5: Play a super-arm $S^t = N$
6: Observe qualities $X_i^j, \forall i \in S^t$ and update w_i^t, $\hat{q}_i^t$
7: $t \leftarrow t + 1$
8: **while** $t \leq T$ **(Explore-Exploit Phase) do**
9: For each agent i, set $(\hat{q}_i^t)^+ = \hat{q}_i^t + \sqrt{\frac{3 \ln t}{2 w_i^t}}$
10: S^t = SSA ($\{(\hat{q}_i^t)^+\}_{i \in N}, c, \alpha + \epsilon_2$,R)
11: Observe qualities $X_i^j, \forall i \in S^t$ and update w_i^t, $\hat{q}_i^t$
12: $t \leftarrow t + 1$

and w_i^t denote the number of rounds an agent i has been selected until round t, i.e., $w_i^t = \sum_{y \leq t} x_i^y$. For $i \in S^t$, C observes its realized quality X_i^j, where $j = w_i^t$ and $\mathrm{E}[X_i^j] = q_i$ and obtains a utility of $r(S^t) = \sum_{i \in S^t} Rq_i - c_i$. The empirical mean estimate of q_i at round t, is denoted by $\hat{q}_i^t = \frac{1}{w_i^t} \sum_{j=1}^{w_i^t} X_i^j$. The upper confidence bound (UCB) estimate is denoted by $(\hat{q}_i^t)^+$) (Algorithm 2, Line 9)

We refer to the algorithm as DPSS-UCB when we use DPSS (Algorithm 1) as SSA in the SS-UCB framework. We prove that DPSS-UCB outputs the super-arm that satisfies the QC with high probability after a certain threshold number of rounds, τ, and incurs a regret of $O(\ln T)$. Additionally, we supplement our proofs by evaluating our framework on simulated data (Figure 2 and 3).

Theorem 1. *For $\tau = \frac{3 \ln T}{2\epsilon_2^2}$, if each agent is explored τ number of rounds, then if we invoke DPSS with target threshold $\alpha + \epsilon_2$ and $\{(\hat{q}_i^t)^+\}_{i \in N}$ as the input, the QC is approximately met with high probability.*

$$\mathcal{P}\left(q_{av}^t < \alpha - \epsilon_1 \mid \frac{1}{s^t} \sum_{i \in S^t} (\hat{q}_i^t)^+ \geq \alpha + \epsilon_2, t > \tau\right) \leq exp(-\epsilon_1^2 t).$$

where ϵ_1 is the tolerance parameter and refers to the planner's ability to tolerate a slighty lower average quality than required.

Definition 1. *We say $\boldsymbol{q} = (q_1, q_2, \ldots, q_n)$ satisfies ϵ-seperatedness if $\forall S \subseteq N$, $U(S) = \frac{1}{s} \sum_{i \in S} q_i$ s.t. $U(S) \notin (\alpha - \epsilon, \alpha)$.*

This suggests that there is no super-arm $S \in \chi$, such that $\alpha - \epsilon \leq \frac{1}{|S|} \sum_{i \in S} q_i^t \leq \alpha$. It is important for DPSS-UCB to satisfy ϵ_1-seperatedness because if there exists such a super-arm, for which the average quality is between $(\alpha - \epsilon_1, \alpha)$, DPSS-UCB will include it in χ due to tolerance parameter ϵ_1 while it would violate the QC.

Theorem 2. *If qualities of the agents satisfy ϵ_1-seperatedness, then regret incurred for $t > \tau$ is bounded by $O(\ln T)$.*

GSS. : To overcome the computational limitations of DPSS in ($O(2^n)$ time complexity), we propose GSS that runs $O(n \log n)$ and provides an approximate solution to our ILP. GSS solves for the linearly relaxed variant of our ILP; however, due to the nature of our QC, we don't always drop the fractional part. The details of the algorithm can be found in [10]. While the approximation ratio of the utilities obtained through GSS and DPSS can be arbitrarily small, we show that in practice (Fig. 1), GSS gives close to optimal solutions at a huge computational benefit that allows us to scale our framework for a large number of agents. We then use GSS as our SSA in the SS-UCB framework and propose GSS-UCB as an alternate algorithm to DPSS-UCB when qualities are unknown. Through experiments on simulated data (Fig. 2 and 3), we show that GSS-UCB achieves a comparable regret and constraint satisfaction similar to DPSS-UCB.

2 CONCLUSION

In this paper, we addressed the class of problems where a central planner had to select a subset of agents that maximized its utility while ensuring a quality constraint. Our motivation towards this setting was consumer-oriented cooperative societies such as those of artisans where production is decentralized. Each producer has a different quality and cost of produce depending on its workmanship and the scale at which it operates. In such settings, the qualities of the individual units of produce are stochastic and usually difficult to quantify until the products are procured and sold in the market, which justifies the need for a MAB framework. Towards this, we propose a generalized SS-UCB framework that can be used to design and compare other approaches to this class of problems. The framework also allows solving for other interesting variants of the problem such as (i) where the pool of agents is dynamic (ii) where an agent selected in a particular round is not available for the next few rounds (sleeping bandits) (iii) where the planner needs to design a mechanism to elicit a strategic agents' cost of production truthfully. Kindly refer to [10] for the full version of the paper.

REFERENCES

[1] Kumar Abhishek, Shweta Jain, and Sujit Gujar. 2020. *Designing Truthful Contextual Multi-Armed Bandits Based Sponsored Search Auctions.* International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 1732–1734.

[2] Shipra Agrawal and Nikhil R Devanur. 2014. Bandits with concave rewards and convex knapsacks. In *Proceedings of the fifteenth ACM conference on Economics and computation*. 989–1006.

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

[4] Ashwinkumar Badanidiyuru, John Langford, and Aleksandrs Slivkins. 2014. Resourceful contextual bandits. In *Conference on Learning Theory*. 1109–1134.

[5] Arpita Biswas, Shweta Jain, Debmalya Mandal, and Y. Narahari. 2015. A Truthful Budget Feasible Multi-Armed Bandit Mechanism for Crowdsourcing Time Critical Tasks. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems* (Istanbul, Turkey) *(AAMAS '15)*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 1101–1109. http://dl.acm.org/citation.cfm?id=2772879.2773291

[6] Shouyuan Chen, Tian Lin, Irwin King, Michael R Lyu, and Wei Chen. 2014. Combinatorial Pure Exploration of Multi-Armed Bandits. In *Advances in Neural Information Processing Systems 27*. 379–387.

[7] Wei Chen, Wei Hu, Fu Li, Jian Li, Yu Liu, and Pinyan Lu. 2016. Combinatorial multi-armed bandit with general reward functions. In *Advances in Neural Information Processing Systems*. 1659–1667.

[8] Wei Chen, Yajun Wang, and Yang Yuan. 2013. Combinatorial Multi-Armed Bandit: General Framework and Applications. In *Proceedings of the 30th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 28)*, Sanjoy Dasgupta and David McAllester (Eds.). PMLR, Atlanta, Georgia, USA, 151–159. http://proceedings.mlr.press/v28/chen13a.html

[9] Richard Combes, Mohammad Sadegh Talebi Mazraeh Shahi, Alexandre Proutiere, and marc lelarge. 2015. Combinatorial Bandits Revisited. In *Advances in Neural Information Processing Systems 28*. 2116–2124.

[10] Ayush Deva, Kumar Abhishek, and Sujit Gujar. 2021. A Multi-Arm Bandit Approach To Subset Selection Under Constraints. arXiv:2102.04824 [cs.LG]

[11] Yi Gai, Bhaskar Krishnamachari, and Rahul Jain. 2010. Learning multiuser channel allocations in cognitive radio networks: A combinatorial multi-armed bandit formulation. In *IEEE Symposium on New Frontiers in Dynamic Spectrum*. 1–9.

[12] Chien-Ju Ho, Shahin Jabbari, and Jennifer Wortman Vaughan. 2013. Adaptive task assignment for crowdsourced classification. In *International Conference on Machine Learning*. 534–542.

[13] Shweta Jain, Satyanath Bhat, Ganesh Ghalme, Divya Padmanabhan, and Y. Narahari. 2016. Mechanisms with learning for stochastic multi-armed bandit problems. *Indian Journal of Pure and Applied Mathematics* 47, 2 (01 Jun 2016), 229–272. https://doi.org/10.1007/s13226-016-0186-3

[14] Shweta Jain, Sujit Gujar, Satyanath Bhat, Onno Zoeter, and Y Narahari. 2018. A quality assuring, cost optimal multi-armed bandit mechanism for expertsourcing. *Artificial Intelligence* 254 (2018), 44–63.

[15] David R Karger, Sewoong Oh, and Devavrat Shah. 2011. Iterative learning for reliable crowdsourcing systems. In *Advances in neural information processing systems*. 1953–1961.

[16] Aleksandrs Slivkins. 2019. Introduction to multi-armed bandits. *Foundations and Trends® in Machine Learning* (2019).

[17] Long Tran-Thanh, Sebastian Stein, Alex Rogers, and Nicholas R Jennings. 2014. Efficient crowdsourcing of unknown experts using bounded multi-armed bandits. *Artificial Intelligence* 214 (2014), 89–111.

[18] Long Tran-Thanh, Matteo Venanzi, Alex Rogers, and Nicholas R Jennings. 2013. Efficient budget allocation with accuracy guarantees for crowdsourcing classification tasks. In *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*. International Foundation for Autonomous Agents and Multiagent Systems, 901–908.