

Volatile Multi-Armed Bandits for Guaranteed Targeted Social Crawling

Zahy Bnaya*, Rami Puzis*, Roni Stern†, Ariel Felner*

* Information Systems Engineering, Ben-Gurion University, Beer-Sheva, Israel

[†] SEAS, Harvard University, Cambridge MA, USA

Abstract

We introduce a new variant of the multi-armed bandit problem, called *Volatile Multi-Arm Bandit* (VMAB). A general policy for VMAB is given with proven regret bounds. The problem of collecting intelligence on profiles in social networks is then modeled as a VMAB and experimental results show the superiority of our proposed policy.

Introduction

The multi-arm bandit problem (MAB) (Robbins 1985) assumes a set of K independent gambling machines (or arms). At each turn (n), a player pulls one of the arms (a_i) and receives a reward ($X_{i,n}$) drawn from some unknown distribution with a mean value of μ_i . A policy for MAB chooses the next arm to pull based on previously observed rewards.

MAB policies are often designed to minimize their *accumulated regret* (R_n), which is the accumulated expected loss of rewards by not pulling the optimal arm at all turns up to n . Formally, $R_n = n \cdot \mu^* - \sum_{i=1}^k \mu_i \cdot E[T_i(n)]$ where μ^* is the expected reward of the best arm and $E[T_i(n)]$ is the expected number of pulls of arm i in the first n turns. UCB1 (Auer, Cesa-Bianchi, and Fischer 2002) is a benchmark policy ensuring that $R_n = O(\log(n))$. Initially UCB1 pulls each arm once. Then, on each turn n UCB1 pulls an arm a_i that maximizes

$$\overline{X}_i + \sqrt{\frac{2 \cdot \ln n}{T_i(n)}} \quad (1)$$

Where \bar{X}_i is the average reward observed so far by pulling arm a_i . MAB applications spread on many areas such as clinical trials, web search, Internet advertising and multi-agent systems.

Volatile multi-arm bandits

Standard MAB problems assume a constant set of K arms which are available indefinitely. We propose an extension of MAB, where new arms can *appear* or *disappear* on each turn. We call this MAB variant *Volatile-multi-Arm bandit problem* (VMAB). In VMAB, every arm a_i is associated

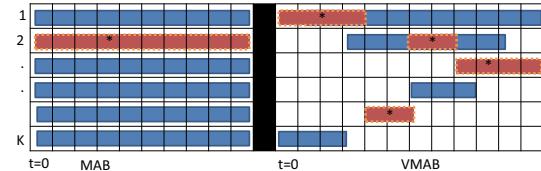


Figure 1: Regular MAB (left) and Volatile MAB (right).

with a pair of turns (s_i, t_i) during which this arm is available, referred to as the arm's *lifespan*. Figure 1 illustrates the difference between MAB and VMAB. The standard MAB (left) has a fixed set of K arms, one of them is optimal (colored red). In VMAB arms appear and disappear and the optimal arm at each time may change (right). We assume that the arms' *lifespans* are unknown in advance.

A policy for VMAB chooses on each turn n an arm a_i with a lifespan $s_i \leq n \leq t_i$. The expected regret of a VMAB policy (labeled R^v) is:

$$R_n^v = \sum_{t=1}^n \mu^*(t) - \mu(I(t))$$

Where $\mu^*(n)$ is the expected reward of the optimal arm at turn n and $I(t)$ is the arm selected at time t .

Obviously, the regret of VMAB depends on the number of available arms at each time step. Consider a set of K arms with mutual-exclusive lifespans. In this case, the accumulated regret is 0 since there is only one arm to pull at each time. By contrast, if all arms have the same lifespans, the problem is identical to standard MAB.

VMAB is a reminiscent of *Mortal Multi-Armed Bandits* (Chakrabarti et al. 2008) designed for online-advertising. They also assume that arms can expire or become available. However, the number of arms is fixed, i.e., whenever one arm disappears, a new arm appears. Their analysis focuses on when to stay with the previous best arm and when to select a new arm, a problem which is inherently different than ours. *Restless-bandits* (Whittle 1988) assumes a fixed set of arms, from which only a subset is available at each turn. Several other similar variants exists (Whittle 1981; Kleinberg, Niculescu-Mizil, and Sharma 2010) that use a different set of assumptions than ours.

VMAB Policy

For solving a VMAB problem, we propose the VUCB1 policy. VUCB1 uses the same formula as UCB1, but only on the

available arms. If no new arm appears, VUCB1 selects the arm that maximizes the UCB1 formula given in Equation 1. When a new arm appears we do not consult the UCB1 formula but three steps are taken:

- 1) Immediately pull the new arm.
- 2) Reset the $T_i(n)$ (the number of times arm i was pulled) of all other existing arms a_i to 1 but keeping their reward averages.
- 3) The total number of pulls (labeled by the *turn* variable n) is set to the current number of available arms.

Theorem 1 *The expected regret R_n of VUCB1 is $O(B \cdot \log(n))$ where B is the number of times a new arm has appeared during turns $[1..n]$.*

Proof Sketch: While no new arm appears and no arm disappears, VUCB1 behaves exactly like UCB1. Now, if there are t turns until an arm has appeared, then the regret for these t turns is bounded by $O(\log(t)) \leq O(\log(n))$. Removing an arm does not affect the regret. Thus, the accumulative regret after a new arm appears B times is $O(B \cdot \log(n))$. \square

TONIC

To demonstrate the applicability of VMAB, consider the Target Oriented Network Intelligence Collection (TONIC) problem (Stern et al. 2013). In TONIC we are interested in retrieving information about a given person of interest, denoted as the *target*, from social network (SN). Due to privacy issues, the target SN profile is inaccessible to third parties. However, other SN profiles contain information about the target and are accessible, having more relaxed privacy settings. Such profiles are called *leads*. The TONIC problem is to find as many leads as possible while minimizing the number of analyzed profiles.

Solving TONIC consists of analyzing SN profiles, an action referred to as “acquiring” a profile. If a profile is a lead then acquiring it, e.g., with information extraction methods (Chang et al. 2006; Tang et al. 2010; Pawlas, Domanski, and Domanska 2012, *inter alia*), reveals information about the target, and may provide pointers to additional *potential leads*. Several TONIC heuristics were proposed to decide which *potential lead* to acquire next in order to find more leads (Stern et al. 2013). The best performing TONIC heuristic, called *BysP*, associates each lead with a *promising rate* (PR), which is intended to represent how likely it is for a random profile connected to this lead to also be a lead. PR of a lead l would ideally be the percentage of leads out of the total profiles connected to l , and is estimated by considering only the previously acquired profiles. BysP then prioritized potential leads by aggregating the PR values of the leads they are connected to. For further details on TONIC and BysP, see (Stern et al. 2013).

The PR values used by BysP may be misleading, as they only depend on the previously acquired profiles. This raises an *exploration vs. exploitation* tradeoff that can be modeled naturally as a VMAB, as follows. We partition all leads and potential leads into equivalence classes, such that profiles that are connected to the same set of leads are in the same equivalence class (this is similar to the the notion of *structurally equivalence* (Sailer 1978)). Each equivalence class is

a single bandit arm, and arms appear and disappear as equivalence classes are created and removed when new leads are found. The expected reward μ_i of each arm is the proportion of leads in that class. Rewards of $[0, 1]$ are received by analyzing a random profile of a certain equivalence class and determining whether it is a lead or not. Pulling a bandit arm correspond in TONIC to acquiring a profile from a selected equivalence class. Thus, modeling TONIC as VMAB allows using the VUCB1 policy to balance exploration and exploitation more informedly than BysP, when deciding which profile to acquire next. The experimental results next demonstrate the effectiveness VUCB1 in solving TONIC.

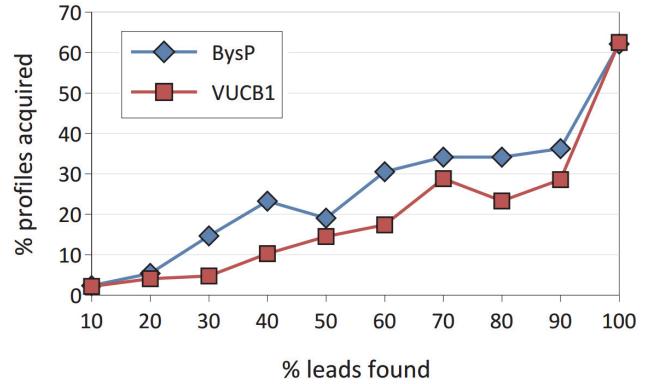


Figure 2: VUCB1 dominates BysP, finding more leads faster.

Empirical evaluation

The data set we used for our experiments was obtained from the Google+ network and included 211K profiles with 1.5M links between them. This data set was collected by (Fire et al. 2011) and made available. From this data set we randomly selected a set of 100 profiles having at least 30 friends. These profiles were used as the targets in our experiments. Figure 2 demonstrates the average percentage of profile acquisitions (Y axis) required to find each percentage of leads (X axis). Results show that VUCB1 clearly dominates BysP, demonstrating the merit of modeling TONIC as a VMAB problem and solving it with VUCB1.

Conclusions

We defined the Volatile Multi-Arm Bandits problem where arms can appear/disappear and presented a policy for VMAB that guarantees logarithmic bounds on accumulated regrets for this problem. We applied this policy to the TONIC problem achieving state-of-the-art performance. We believe that VMAB is relevant to many other applications and that we have only touched the surface of this direction. In the future we aim to further analyze VMAB and find tighter bounds for variants of VMAB. In addition, we aim to explore other real-world applications that can be modeled as VMAB.

References

- Auer, P.; Cesa-Bianchi, N.; and Fischer, P. 2002. Finite-time analysis of the multiarmed bandit problem. *Mach. Learn.* 47(2-3):235–256.
- Chakrabarti, D.; Kumar, R.; Radlinski, F.; and Upfal, E. 2008. Mortal multi-armed bandits. In *Neural Information Processing Systems*, 273–280.
- Chang, C.; Kayed, M.; Girgis, M.; Shaalan, K.; et al. 2006. A survey of web information extraction systems. *IEEE transactions on knowledge and data engineering* 18(10):1411.
- Fire, M.; Tenenboim, L.; Lesser, O.; Puzis, R.; Rokach, L.; and Elovici, Y. 2011. Link prediction in social networks using computationally efficient topological features. In *Privacy, security, risk and trust (passat), 2011 ieee third international conference on and 2011 ieee third international conference on social computing (socialcom)*, 73–80. IEEE.
- Kleinberg, R.; Niculescu-Mizil, A.; and Sharma, Y. 2010. Regret bounds for sleeping experts and bandits. *Machine learning* 80(2-3):245–272.
- Pawlas, P.; Domanski, A.; and Domanska, J. 2012. Universal web pages content parser. In *Computer Networks*, volume 291 of *Communications in Computer and Information Science*. Springer Berlin Heidelberg. 130–138.
- Robbins, H. 1985. Some aspects of the sequential design of experiments. In *Herbert Robbins Selected Papers*. Springer. 169–177.
- Sailer, L. D. 1978. Structural equivalence: Meaning and definition, computation and application. *Social Networks* 1(1):73 – 90.
- Stern, R.; Samama, L.; Puzis, R.; Felner, A.; Bnaya, Z.; and Beja, T. 2013. Target oriented network intelligence collection for the social web. In *AAAI AIW*.
- Tang, J.; Yao, L.; Zhang, D.; and Zhang, J. 2010. A combination approach to web user profiling. *ACM Trans. Knowl. Discov. Data* 5(1):2:1–2:44.
- Whittle, P. 1981. Arm-acquiring bandits. *The Annals of Probability* 284–292.
- Whittle, P. 1988. Restless bandits: Activity allocation in a changing world. *Journal of applied probability* 287–298.

Erratum: This research was partially supported by IDF.