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 $\mu_i$. 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 $\mu^*$ 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

$$X_i + \sqrt{\frac{2 \cdot \ln n}{T_i(n)}}$$  

(1)

Where $X_i$ 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 bands
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

Copyright © 2013, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.
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 \( O(B \cdot \log(n)) \).

**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 \( \mu_i \) 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.

**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


Erratum: This research was partially supported by IDF.