The Budgeted Multi-Armed Bandit Problem

Omid Madani[1], Daniel J. Lizotte[2], and Russell Greiner[2]

[1]Yahoo! Research Labs
74 N. Pasadena Ave,
Pasadena, CA 91101
omid.madani@overture.com

[2]Dept. of Computing Science
University of Alberta
Edmonton, T6J 2E8
{dlizotte|greiner}@cs.ualberta.ca

The following *coins problem* is a version of a multi-armed bandit problem where one has to select from among a set of objects, say classifiers, after an experimentation phase that is constrained by a time or cost budget. The question is how to spend the budget. The problem involves pure exploration only, differentiating it from typical multi-armed bandit problems involving an exploration/exploitation tradeoff [BF85]. It is an abstraction of the following scenarios: choosing from among a set of alternative treatments after a fixed number of clinical trials, determining the best parameter settings for a program given a deadline that only allows a fixed number of runs; or choosing a life partner in the bachelor/bachelorette TV show where time is limited. We are interested in the computational complexity of the coins problem and/or efficient algorithms with approximation guarantees.

1 The Coins Problem

We are given:

- A collection of n independent coins, indexed by the set $\mathcal{I}$, where each coin is specified by a probability density function (prior) over its head probability. The priors of the different coins are independent, and they can be different for different coins.
- A budget b on the total number of coin flips.

We assume the tail and the head outcomes correspond to receiving no reward and a fixed reward (1 unit) respectively. We are allowed a trial/learning period, constrained by the budget, for the sole purpose of experimenting with the coins, *i.e.*, we do not collect rewards in this period. At the end of the period, we are allowed to pick only a single coin for all our future flips (reward collection).

Let the actual head probability of coin i be θ_i. We define the *regret* from picking coin i to be $\theta^* - \theta_i$, where $\theta^* = \max_{j \in \mathcal{I}} \theta_j$. As we have the densities only, we basically seek to make coin flip decisions and a final choice that lead to minimizing our *expected* regret. It is easy to verify that when the budget is 0, the choice of coin that minimizes expected regret is one with maximum expected head probability over all the coins, *i.e.*, $\max_i E(\Theta_i)$, where Θ_i denotes the random variable corresponding to head probability of coin i, and the expectation $E(\Theta_i)$ is taken over the density for coin i.

A strategy is a prescription of which coin to flip given all the coins' flip outcomes so far. A strategy may be viewed as a finite directed rooted tree, where each node indicates a coin to flip, each edge indicates an outcome (heads or tails), and the leaves indicate the coin to choose [MLG04]. No path length from root to leaf exceeds the budget. Thus the set S of such strategies is finite. Associated with each leaf node j is the (expected) regret r_j, computed using the densities (one for each coin) at that node. Let p_j be the probability of "reaching" leaf j: p_j is the product of the probabilities of coin flip outcomes along the path from root to that leaf. We define the *regret* of a strategy to be the expected regret, where the expectation is taken over the coins' densities and the possible flip outcomes: $Regret(s) = \sum_{j \in \text{Tree Leafs of } s} p_j r_j$. The optimal regret r^* is then the minimum achievable (expected) regret and an *optimal strategy* s^* is one achieving it[1]

$$r^* = \min_{s \in S} Regret(s), \quad s^* = \arg\min_{s \in S} Regret(s). \tag{1}$$

We assume the budget is no larger than a polynomial in n, and that we can represent the densities and update them (when the corresponding coin yields a heads or tails outcome), and compute their expectation efficiently (*e.g.*, the family of beta densities). With these assumptions, the problem is in PSPACE [MLG04].

Open Problem 1 *Is computing the first action of an optimal strategy NP-hard?*

2 Discussion and Related Work

We explore *budgeted learning* in [MLG04,LMG03]. We show that the coins problem is NP-hard under non-identical coin flip costs and non-identical priors, by reduction from the Knapsack problem. We present some evidence that the problem remains difficult even under identical costs. We explore constant-ratio approximability for strategies and algorithms[2]: an algorithm is a constant ratio approximation algorithm if its regret does not go above a constant multiple of the minimum regret. We show that a number of algorithms such as round-robin and greedy cannot be approximation algorithms. In the special case of identical priors (and coin costs), we observe empirically that a simple algorithm we refer to as *biased-robin* beats the other algorithms tested, and furthermore, its regret is very close to the optimal regret on the limited range of problems for which we could compute the optimal. Biased-robin sets $i = 1$, and continues flipping coin i until the outcome is tails, at which time it sets i to $(i \bmod n) + 1$, and repeats until the budget is exhausted. Note that biased-robin doesn't take the budget into account except for stopping! An interesting open problem is then:

Open Problem 2 *Is biased-robin a constant-ratio approximation algorithm, for identical priors and budget of* $b = O(n)$*?*

References

[BF85] D. Berry and B. Fristedt. *Bandit Problems: Sequential Allocation of Experiments*. Chapman and Hall, New York, NY, 1985.

[1] No randomized strategy has regret lower than the optimal deterministic strategy [MLG04].

[2] An algorithm defines a strategy (for each problem instance) implicitly, by indicating the next coin to flip [MLG04].

[LMG03] D. Lizotte, O. Madani, and R. Greiner. Budgeted learning of Naive Bayes classifiers. In *UAI-2003*, 2003.

[MLG04] O. Madani, D. Lizotte, and R. Greiner. Active model selection (submitted). Technical report, University of Alberta and AICML, 2004. http://www.cs.ualberta.ca/∼madani/budget.html.