Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)

A Finite Memory Automaton for Two-Armed Bernoulli Bandit Problems

Ariel Rao
Carnegie Mellon University
5032 Forbes Avenue, SMC 3662
Pittsburgh, PA 15289-3662
(650) 996-4637
https://raoariel.github.io
arielrao@cmu.edu

Introduction

Existing approaches to the multi-armed bandit (MAB) primarily rely on perfect recall of past actions to generate estimates for arm payoff probabilities; it is further assumed that the decision maker knows whether arm payoff probabilities can change. To capture the computational limitations many decision making systems face, we explore performance under bounded resources in the form of imperfect recall of past information. We present a finite memory automaton (FMA) designed to solve static and dynamic MAB problems. The FMA demonstrates that an agent can learn a low regret strategy without knowing whether arm payoff probabilities are static or dynamic and without having perfect recall of past actions. Roughly speaking, the automaton works by maintaining a relative ranking of arms rather than estimating precise payoff probabilities.

Two-Armed Bernoulli Bandits

For simplicity, we restrict our analysis of the FMA's performance to the class of two-armed Bernoulli bandit (TABB) problems, where there are exactly two arms with probabilistic win-loss payoffs of $\{0, 1\}$ following Bernoulli distributions. We consider static environments where the true payoff probabilities remain constant as well as dynamic environments with either small frequent changes to payoff probabilities or unbounded infrequent changes to payoff probabilities. Static TABB (Granmo 2008) and dynamic TABB with small frequent changes (Gupta, Granmo, and Agrawala 2011) have been studied previously. We include TABB with unbounded changes because these capture a distinct feature of real world decision problems.

Motivation

Many approaches to solving bandit problems work by estimating payoff probabilities of the arms; ϵ-greedy strategies sample all arms to get a rough estimate for each arm, upper confidence bound (UCB) strategies estimate a range that the payoff probabilities are likely to lie within (Mayo-Wilson, Zollman, and Danks 2012), and the Bayesian learning automaton (BLA) uses a beta distribution to estimate payoff probabilities (Granmo 2008). However, the MAB is not fundamentally an estimation problem; the MAB is a ranking problem. The FMA bypasses probability estimation to directly produce a relative ranking of the arms.

Further, most existing approaches to MAB rely on the full history of received signals for perfect Bayesian updating of beliefs about the arm payoff probabilities. There have been a number of techniques using finite automata (Wilson 2014) and Turing machines (Halpern and Pass 2015) to represent computationally limited agents for other problems within game and decision theory. The FMA borrows constructs for bounded decision making from automata theory while leveraging a payoff probability tracking mechanism similar to that presented in the BLA. By encoding beliefs about the relative ranking of the arms within a finite state space, the FMA is essentially restricted to a finite set of possible beliefs. Moreover, the fact that the FMA maintains rankings rather than point-estimates makes it a flexible enough framework to perform well in both static and dynamic environments.

Finite Memory Automaton

We model a decision maker (DM) as a stochastic finite state automaton where each state encodes discrete beliefs about the payoff probabilities of the arms in the decision problem.

Formally, a DM for the TABB is a tuple $DM = (\Omega, \omega_0, a, t)$ where

- $\Omega = \{1, ..., m\}^2$ is the state space;
- $\omega_0 \in \Omega$ is the starting state;
- $a : \Omega \to \Delta(\{1, 2\})$ where $a(\omega)$ specifies probabilistically which arm the decision maker should play at state ω;
- $t : \Omega \times \{1, 2\} \times \{0, 1\} \to \Delta(\Omega)$ where $t(\omega, i, s)$ determines the decision maker's new state as updated by the last signal s from arm i.

State Space Each state $\omega = (r_1, r_2)$ encodes beliefs about the rankings of the two arms. r_i represents the current rank of arm i and is constrained by $1 \leq r_i \leq m$.

Starting State Prior beliefs the decision maker may have can be captured in the initial rankings as the starting state.

Action For each state, the action function a assigns a probability for selecting the next arm. A balance of exploration and exploitation can be achieved when a DM's tendency to explore is inversely proportional to its confidence in the arms' relative rankings. [1]

Transition A probabilistic transition between states is determined by the previous arm selected and its corresponding output signal. Since the DM has a finite state space and can only retain a sliding window of past signals, the DM becomes more sensitive to recent signals. The notion of *inertia* is developed to mitigate this recency bias. Each edge in the state space is assigned a temporal score tracking the frequency of traversal, where higher traversal scores translate to lower inertia along that edge.

Experiments

To show that the FMA framework performs well in multiple TABB environments, each DM is run with the same settings against other algorithms specialized for that environment. [2] We take random arm selection as the baseline strategy. Figure 1 compares performance in the standard static environment. This experiment consists of 5000 randomly generated arm pairs for horizons of 500. Figure 2 compares performance in a mixed environment where arm payoff probabilities exhibit randomized stages of static and dynamic behavior. This experiment contains 5000 sequences with horizons of 10000.

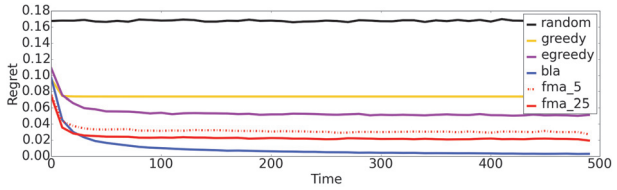

Figure 1: Regret over time in a static environment.

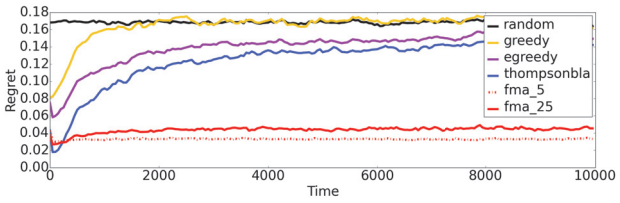

Figure 2: Regret over time in a mixed environment.

Discussion

From our experimental results across TABB conditions, we make the following observations about the FMA framework.

Relative Ranking Unlike high performing algorithms which focus on the precision of arm payoff probability estimation, the FMA is concerned with the accuracy of the ranking of arm payoff probability. From experimental results on static TABB, we see that such a ranking is sufficient for learning a strategy with low regret for horizons of over ten thousand stages and without relying on perfect recall of past signals. When we further consider dynamic TABB, we see that a ranking is more suitable for recognizing and responding to changes in arm payoff probability.

Fast Learning By sampling measures of regret over time, we can examine how well each algorithm learns its strategy. This learning behavior is especially valuable in applied systems where there is not an expectation of infinite data points and fast learning of a high performance strategy has an impact.

Flexible Framework An assumption that most algorithms for TABB make is that the agent knows what type of behavior is expected of the bandit arms, such as whether arm payoff probabilities are static or dynamic. However, an agent may not know this information a priori, so this assumption is not always appropriate. An even more compelling reason to relax this assumption is that many applications of bandit arms exhibit all three types of behavior at various stages in the decision problem and it may not be clear to the agent which type of behavior an arm is currently exhibiting. We evaluate the FMA with a single fixed setting and show that the FMA can perform well in each environment without requiring specialized tuning or additional mechanisms.

Additional Results An extended discussion of the experimental setup and results can be found at https://raoariel.github.io/finite-memory-automaton/ .

Acknowledgements

I would like to thank my advisor Adam Bjorndahl for his guidance and the Game Theory reading group for their insights on bounded rationality and bandit problems.

References

Granmo, O. C. 2008. A bayesian learning automaton for solving two-armed bernoulli bandit problems. *Seventh International Conference on Machine Learning and Applications*.

Gupta, N.; Granmo, O. C.; and Agrawala, A. 2011. Thompson sampling for dynamic multi-armed bandits. *10th International Conference on Machine Learning and Applications and Workshops*.

Halpern, J. Y., and Pass, R. 2015. Algorithmic rationality: Game theory with costly computation. *Journal of Economic Theory*.

Mayo-Wilson, C.; Zollman, K.; and Danks, D. 2012. Wisdom of crowds versus groupthink: Learning in groups and in isolation. *International Journal of Game Theory*.

Wilson, A. 2014. Bounded memory and biases in information processing. *Econometrica*.

[1] A special case is when an arm has the highest ranking, m. It is advantageous to eliminate exploration in these states and play only that highest ranking arm. If that arm has perfect payoff, the DM can take full advantage and exploit that arm in all future stages.

[2] For ϵ-greedy, we take ϵ=0.3. Specifications for the BLA are taken from Granmo (2008), specifications for the BLA with Thompson samping are taken from Gupta, Granmo, and Agrawala (2011), and specifications for the FMA are in Additional Results.