ECSE 506 Project Report:

Multi-armed Bandit Problems

Ayman Elkasrawy

260458854

Hussam Nosair

260454183

April 16, 2012

Contents

1 Introduction

It is quite common, in real-world situations, that a decision maker is faced with the problem of allocating limited available resources to a number of competing projects. For example, machine job scheduling, project management, and clinical trials are all concerned with the reallocation of limited available resources to competing jobs, projects, or trials. These examples belong to a class of sequential resource allocation problems known as Multi-armed bandit (MAB) problems. From a paradigmatic point of view, MAB problems highlight the fundamental conflict between exploitation (choosing the best decision to maximize the immediate expected payoff) and exploration (trying other decisions to better understand their expected payoff). In fact, the name, multi-armed bandit, was traditionally motivated by the single-arm bandit (the slot machine). In the MAB case, the slot machine has several arms and the gambler is faced with the decision of whether to pull the best arm, based on current knowledge, or try other arms in hope to maximize future payoffs. The report will focus on the classical MAB problem and its optimal solution.

2 The Classical MAB Problem

First, we define the trivial case of a single-armed bandit process and then extend it to the multi-armed bandit process. Next, the MAB problem is stated as a maximization problem.

2.1 A Bandit Process

A bandit process refers to a single-armed bandit problem. The arm is selected (pulled) repeatedly generating two random sequences,

$$(X(0),\, X(1),\, X(2),\, ...\,)$$
$$(R(X(0)),\, R(X(1)),\, R(X(2)),\, ...\,)$$

where $X(n) \in \mathcal{R}$ refers to the state of the bandit process after being selected n times, and $R(X(n)) \in \mathcal{R}_+$ refers to the reward obtained as a result of being in state $X(n)$ after the n^{th} selection. We will assume the bandit process is a time homogeneous Markov process and thus, the state transition is modeled as

$$X(n+1) = f(X(n), W(n)) \qquad \forall n = 0,\, 1,\, 2,\, ...$$

where $f(\cdot)$ is given and $W(n) \in \mathcal{R}$ is a sequence of independent random variables that are also independent of $X(0)$.

2.2 Extension to MAB process

A k-armed bandit process consists of k independent single arms where only one arm is played at every decision time. This generates k independent single-armed bandit processes as described earlier. When the decision maker selects an arm, the other arms are frozen and their corresponding processes are frozen as a result. Only the process of the selected arm is continued, leading to a state change and a reward being obtained. The whole system evolves according to the following model:

$$X_i(n+1) = \begin{cases} f(X_i(n), W_i(n)) & \text{if } U_i(n) = 1 \\ X_i(n) & \text{if } U_i(n) = 0 \end{cases} \tag{1}$$

$$R_i(n) = \begin{cases} R_i(X_i(n)) & \text{if } U_i(n) = 1 \\ 0 & \text{if } U_i(n) = 0 \end{cases} \tag{2}$$

where $i \in \{1, 2, ..., k\}$ denotes the i^{th} arm and U_i is the decision to select $(= 1)$ or freeze $(= 0)$ the i^{th} arm. Therefore, the sequence $\{U(0), U(1), U(2), ...\}$, where $U(n) := (U_1(n), U_2(n), ..., U_k(n))$, is a time homogenous Markov decision policy of the form:

$$U(n) = g(X(n))$$

where $X(n) := ((X(n), X_2(n), ..., X_k(n))$. Here, it is sufficient to confine our search within the set of Markov decision policies, as a result of assuming a Markov MAB process. In other words, the optimal policy is going to be a Markov decision policy. The sequence $\{U(0), U(1), U(2), ...\}$ is referred to as a scheduling policy.

2.3 The MAB Problem Statement

The problem can be stated as the following:

Determine a scheduling policy, g, that maximizes

$$J^g := \mathbb{E}\left[\sum_{t=0}^{\infty} \beta^t \sum_{i=1}^{k} R_i(X_i(t), U_i(t)) \middle| X(0)\right] \tag{3}$$

subject to 1 and 2. $B \in (1,1)$ is a discount factor which guarantees convergence of the summation over infinite time horizon.

3 Solving The MAB Problem

The MAB problem involves sequential decision making and therefore, it can be solved via stochastic dynamic programming (backward induction). Albeit being intractable, stochastic dynamic programming was the only known approach and presented a limited understanding of the structure of the optimal solution, until Gittins was able to formulate and prove the optimal solution to be simple and of an index type via forward induction. We will show the intractability of backward induction formulation and proceed to forward induction formulation in deriving the index-type optimal solution.

3.1 Stochastic Dynamic Programming Solution (Backward Induction)

The standard infinite horizon dynamic programming formulation can be applied to the MAB problem. The value function can be written as following:

$$V(x) = \sup_u \mathbb{E}\left[R(X,U) + \beta V(f(X,W,U)) | X = x, U = u\right] \tag{4}$$

where $x \in \mathcal{R}^k$, and $u_i \in \{0,1\}$.

We've assumed earlier that the MAB process is a time homogenous Markov process. We add a second assumption that the reward function, $R(n)$, is bounded. The result is a controlled Markov decision problem for which the infinite horizon dynamic program converges to a unique fixed point solution that is optimal. The resulting optimal policy is also a Markov decision policy as mentioned earlier.

Unfortunately, the dynamic program suffers from the curse of dimensionality, in which the size of the program grows exponentially with number of states of each arm. So, if we have k

arms and each arm can occupy N number of states, then the size of the dynamic program is k^Npossible states. The curse of dimensionality makes the dynamic program computationally infeasible and hence, offers little information about the structure of the optimal solution. Next, we present the approach of forward induction to tackle this problem.

3.2 Gittins Index Theorem (Forward Induction)

Stochastic dynamic programming solution proceeds backward on induction and hence, leads to no loss of optimality at the expense of computational complexity. On the other hand, forward induction reduces the computational complexity at the expense of possible loss of optimality. To begin with, we consider a myopic policy which maximizes the conditional expected reward over the next stage (i.e. *one*-step-look-ahead). Although myopic policies are much simpler to compute, they are generally not optimal.

Next, we can improve upon myopic policies by maximizing the conditional expected total reward over the next T stages. In other words, we consider T-step-look-ahead policies where T is a fixed number. As T increases, the optimality of the T-step-look-ahead policies improves at the expense of computational simplicity. Nonetheless, such policies are generally suboptimal.

Furthermore, the notion of T-step-look-ahead policies can be extended by varying number of look-ahead stages; call it τ. In this case, it does not make sense to maximize the conditional expected total reward, because τ will grow arbitrarily large and make the comparison among the decision rules meaningless; in addition to worsening computation feasibility. Instead, we maximize over τ the conditional expected total reward *rate*. Then, we pick the decision rule with the best reward rate and select it to runτ times. The maximization is repeated at the end of τ runs. Policies generated by this procedure are called forward induction policies, and they are generally suboptimal, except for certain stochastic decision problems to which our defined classical MAB problem belongs to.

In order to have an intuitive understanding of why forward induction policies may be suboptimal, consider the example of a car traveling in one direction, and there are several intersecting routes along that direction. Each route has a certain speed limit, and we are interested in maximizing the total discounted distance traveled over an infinite time horizon. Rationally,

we would want to maximize the immediate distance traveled by picking the fastest road before the discount factor becomes smaller and smaller with time. So, a forward induction policy would pick the route with the highest distance rate (speed limit) as long as we do not intersect a route with a higher speed limit. It is possible to run into a situation where we prefer our route over a slower route that leads later on to a route much faster than ours. Moreover, it is also possible that our route ends later on at an intersection which provides alternative routes that are much slower than our route and the previous routes we rejected. Overall, it is possible that picking the fastest route via forward induction policy lead to accumulating less discounted distance, especially when the discount factor is closer to unity. Hence, the forward induction policy may be suboptimal.

On the other hand, there is one situation where we are guaranteed to always accumulate the most traveled distance by always picking the fastest route. We would require to always have access to the slower routes we rejected earlier at each intersection. If we compare this example to the MAB bandit and equate the intersecting routes to the arms of the bandit and traveling speed to reward rate, then the forward induction policy is optimal for the classical MAB problem, due to the following assumptions we made:

1. Only one arm is played at each decision time
2. Frozen arms that we rejected are always available for continuation at next decision time
3. The freezing time does not affect the state and reward sequence after continuing a frozen arm
4. Frozen arms contribute no rewards while frozen

Therefore, a forward induction policy is optimal for the MAB problem and can be enumerated as follows:

1. At time t, for each arm $i = 1, ..., k$, maximize over τ the conditional expected reward rate

$$v_i(x_i(t)) = \max_{\tau} \frac{\mathbb{E}\left[\sum_{s=t}^{t+\tau-1} \beta^s R_i(X_i(t+s)) \middle| x_i(t)\right]}{\mathbb{E}\left[\sum_{s=t}^{t+\tau-1} \beta^s \middle| x_i(t)\right]} \tag{5}$$

2. Pick the arm with the highest reward rate

$$i^* = \max_i v_i(x_i(t)) \tag{6}$$

3. Run the process for the duration of τ^* by repeatedly playing arm i^*

4. Repeat 1-3 at next decision time $t + \tau^*$

The value $v_i(x_i(t))$ is called *dynamic allocation index* or *Gittins index* of arm i at state $x_i(t)$. The above result can be restated by what is known as *Gittins Index Theorem*:

Reward is maximized by always continuing the bandit having greatest value of dynamic allocation index.'

3.3 Advantage of Gittins Index Theorem

Gittins index theorem, based on forward induction, simplifies the computational complexity significantly when compared to backward induction. Recall that under backward induction, the computational complexity grows exponentially and the size of the problem is exponential in N. Under Gittins index theorem, the problem reduces to a size that is linear in N (i.e. $k \cdot N$), where k is the number of arms and N is the number of states. Moreover, Gittins index theorem exposes the nature of the optimal policy to be of an index-type.

4 Proof of Gittins Index Theorem

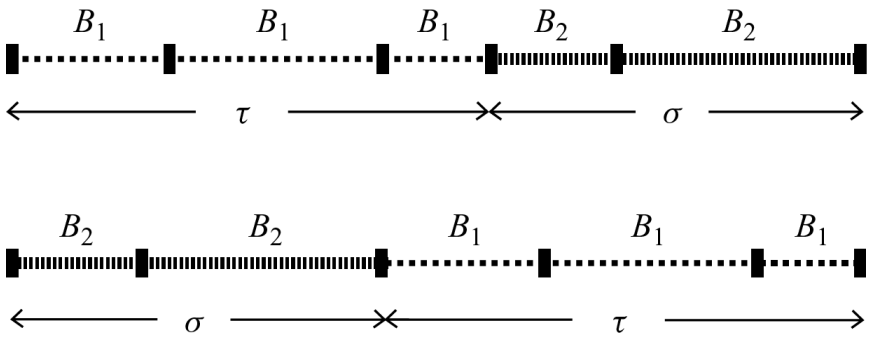
Over the last 40 years, Gittins index theorem had been proven and reproved several times. These proofs vary in difficulty and interpretation of the multi-armed bandit problem. Here, we present a simple intuitive proof based on an interchange argument [2], and then proceed to a more rigorous proof by Tsitsiklis [3].

4.1 An Intuitive Proof via Interchange Argument

The intuition behind the interchange argument is that it is optimal to select earlier the arm with greatest Gittins index (conditional expected reward rate), because this allows us to accumulate more rewards as soon as possible before the discount factor increases geometrically with time. The proof goes as follows:

Let there be two arms which belong to a 2-armed bandit process. At a decision time t, arm B_1 has a Gittins index $v(B_1)$ and an optimal stopping time τ, and arm B_2 has a Gittins index $v(B_2)$ and an optimal stopping time σ. Suppose Gittins index of arm $B_1$is greater than Gittins

index of arm B_2 (i.e. $v(B_1) > v(B_2)$). There are two possible decision rules: give priority to arm B_1 followed by playing arm B_2, or give priority to arm B_2 followed by playing arm B_1. This is illustrated by the following diagram where each arm is run for the duration of its optimal stopping time.



Then the conditional expected total reward over the duration of $\tau + \sigma$ for the policy that gives priority to arm B_1 is $R_\tau(B_1) + \mathbb{E}\beta^\tau R_\sigma(B_2)$, whereas the conditional expected total reward for the policy that gives priority to arm B_2 is $R_\sigma(B_2) + \mathbb{E}\beta^\sigma R_\tau(B_1)$. $R_\tau(B_1)$ and $R_\sigma(B_2)$ are the numerators of the Gittins index of arm $B_1$and arm B_2 respectively. We have assumed that arm B_1 has a greater Gittins index than arm B_2. Therefore,

$$\begin{aligned} v(B_1) > v(B_2) \quad &\Longrightarrow \quad \frac{\mathbb{E}\left[\sum_{s=0}^{\tau-1} \beta^s R_{B_1}(X_{B_1}(s)) \middle| x_{B_1}(0)\right]}{\mathbb{E}\left[\sum_{s=0}^{\tau-1} \beta^s \middle| x_{B_1}(0)\right]} > \frac{\mathbb{E}\left[\sum_{s=0}^{\sigma-1} \beta^s R_{B_2}(X_{B_2}(s)) \middle| x_{B_2}(0)\right]}{\mathbb{E}\left[\sum_{s=0}^{\sigma-1} \beta^s \middle| x_{B_2}(0)\right]} \\ &\Longrightarrow \quad \frac{R_\tau(B_1)}{\frac{1-\mathbb{E}\beta^\tau}{1-\beta}} > \frac{R_\sigma(B_2)}{\frac{1-\mathbb{E}\beta^\sigma}{1-\beta}} \\ &\Longrightarrow \quad R_\tau(B_1) + \mathbb{E}\beta^\tau R_\sigma(B_2) > R_\sigma(B_2) + \mathbb{E}\beta^\sigma R_\tau(B_1) \end{aligned} \tag{7}$$

This implies that giving priority to the arm with the greater Gittins index (i.e. arm B_1) yields higher conditional expected total reward.

4.2 A Rigorous Proof by Tsitsiklis

The following proof by Tsitsiklis is complete as it addresses the bandit problem for multiple arms and compares policies in terms of the conditional expected total reward over an infinite horizon. In order to simplify the calculations, Tsitsiklis assumes that the MAB process is a semi-Markov process in continuous time with an exponential discount rate. He also assumes that the arms have disjoint finite state spaces. The process is time homogenous and the arms

are independent. All these assumptions conform to out definition of the classical multi-arm bandit problem.

We are interested in maximizing the conditional expected total reward:

$$\mathbb{E}\left[\sum_{i=0}^{\infty} e^{-\alpha t_i} R_i \middle| X(0)\right] \tag{8}$$

where $\alpha > 1$ and $e^{-\alpha t_i}$ is the discount factor as a function of continuous time. R_i is the reward received as a result of the i^{th} play. The discrete time expected discounted reward resulting from the i^{th} play is

$$e^{-\alpha t_i}\mathbb{E}\left[R(x_i)|x_i\right] \tag{9}$$

where x_i is the state of the arm played at the i^{th} play. Tsitsiklis introduced a reward structure in continuous time which yields that same expected discounted reward of the i^{th} play as 9. The structure relies on defining a continuous reward rate over interval between the $i^{th}$and $(i+1)^{th}$play. Call this interval $T(x_i)$ which is random and uncontrolled by the decision maker. The continuous time reward rate is

$$r(x) = \frac{\mathbb{E}\left[R(x)\right]}{\mathbb{E}\left[\int_0^{T(x)} e^{-\alpha t} dt\right]} \tag{10}$$

Under the new reward structure, the expected discounted reward of the i^{th}play is

$$\mathbb{E}\left[\int_{t_i}^{t_i+T(x_i)} e^{-\alpha t} r(x_i) dt \middle| x_i\right] = e^{-\alpha t_i}\mathbb{E}\left[R(x_i)|x_i\right] \tag{11}$$

by substituting 10.

Tsitsiklis states the following theorem:

If the discrete state space of each arm is finite, then there exists a priority rule which is optimal.

The proof goes by induction on the joint state space of all the arms. Let N be the cardinality of the joint state space of all arms. If $N = 1$, then there exists trivially a priority rule as the only available arm will be played repeatedly. Assume there exists a priority rule for $N = K$, we will consider the case of $N = K+1$ and show that there exists a priority rule. The following

interchange argument shows that such priority rule exists and selects the arm to which belongs the state $s^*$with the highest reward rate, $r(s^*)$; call this arm $i*$.

Consider two policies, π and π'. Let policy π choose to play arm i^*at timeτ (i.e. not a priority policy). Let policy π'choose to play arm i^*at $t = 0$ (i.e. a priority policy) and thereafter mimic policy π except at $t = \tau$. In other words, we are only interchanging arm i^*at $t = \tau$ with a non optimal arm at $t = 0$. Let the reward rate be $\bar{r}(t) = r(x(t))$ a function of time, then $\bar{r}(t) \leq r(s^*)$ for all t. The expected discounted reward $J(\pi)$ under policy π is

$$J(\pi) = \mathbb{E}\left[\int_0^{\tau} \bar{r}(t)\, e^{-\alpha t} dt + \int_{\tau}^{\tau+T(s^*)} r(s^*)\, e^{-\alpha t} dt + \int_{\tau+T(s^*)}^{\infty} \bar{r}(t)\, e^{-\alpha t} dt\right] \tag{12}$$

and the expected discounted reward $J(\pi')$ under policy π' is

$$J(\pi') = \mathbb{E}\left[\int_0^{T(s^*)} r(s^*)\, e^{-\alpha t} dt + \int_{T(s^*)}^{\tau+T(s^*)} \bar{r}(t)\, e^{-\alpha t} dt + \int_{\tau+T(s^*)}^{\infty} \bar{r}(t)\, e^{-\alpha t} dt\right] \tag{13}$$

The exponential discount factor and integrating in continuous time simplifies the above two expressions such that showing $J(\pi') \geq J(\pi)$ is equivalent to showing that

$$\mathbb{E}\left[(1 - e^{-\alpha\tau}) \int_0^{T(s^*)} r(s^*)\, e^{-\alpha t} dt\right] \geq \mathbb{E}\left[(1 - e^{-\alpha T(s^*)}) \int_0^{\tau} \bar{r}(t)\, e^{-\alpha t} dt\right] \tag{14}$$

which is true because $\bar{r}(t) \leq r(s^*)$. Therefore, the priority rule under policy π' is optimal at $t = 0$ and any later decision time because of the stationarity of the problem.

The above argument assumes that arm $i^*$is at state $s^*$and therefore the priority rule is optimal. Now assume that arm $i^*$is at a state $x \neq s^*$and is played. If this play causes a transition to state s^*, arm i^*will be played repeatedly until eventually the arm transitions to some state different from s^*; say y. This succession of plays can be viewed as a single play which cannot be interrupted due to applying a priority policy that selects s^* repeatedly. This single play has a random duration $\hat{T}(x)$ equal to the sum of random durations of the repeated plays. We can define an equivalent reward rate for the composite play under the new reward structure 10 as follows

$$\hat{r}(t) = \frac{\mathbb{E}\left[\int_0^{\hat{T}(x)} e^{-\alpha t} r(t) dt\right]}{\mathbb{E}\left[\int_0^{\hat{T}(x)} e^{-\alpha t} dt\right]} \tag{15}$$

which will be received throughout the duration of the composite play, $\hat{T}(x)$. This composite play allows us to redefine arm $i^*$by removing $s^*$and replacing $T(x)$ and $r(x)$ with $\hat{T}(x)$ and $\hat{r}(x)$. We also modify the state transition probabilities to transition from state s to state y directly. The modified problem is now a new MAB problem with one less state (i.e. $N = K$) which we have already assumed to have an optimal priority policy, say $\hat{\pi}$, in the induction process. Thus, by induction, the unmodified problem (i.e. $N = K + 1$) also has an optimal priority policy by solving the modified MAB problem; i.e. give top priority to state $s^*$and follow the priority rule $\hat{\pi}$ for the remaining states. In order to solve for $\hat{\pi}$, we can reapply the above logic and remove the second best state after $s^*$and solve the reduce problem with $N = K - 1$, which also has an optimal priority rule by induction. This entails ordering the states by their reward rates in a decreasing order which is equivalent to Gittins index theorem.

5 References

1. A. Mahajan and D. Teneketzis, “Multi-armed bandit problems”, in Foundations and Applications of Sensor Management, Springer-Verlag.

2. John Gittins, Kevin Glazebrook, Richard Weber, “Multi-armed Bandit Allocation Indices”,Wiley; 2 edition (May 10, 2011)

3. J. N. Tsitsiklis, "A Short Proof of the Gittins Index Theorem", Annals of Applied Probability, Vol. 4, No. 1, 1994, pp. 194-199.