Advanced topics in ML and AGT **Fall Semester, 2017/8**

Lecture 13: Bayesian approach to MAB - Gittins index

Lecturer: Yishay Mansour *Scribe: Matan Hasson ,Tomer Ben Moshe, Alon Kafri*

13.1 Introduction

In the Bayesian approach to the *Multi Armed Bandit* problem we assume a statistical model governing the rewards (or costs) observed upon sequentially choosing one of n possible *arms*. We consider a γ-*discounted* setting in which the value of a reward r at time t is $r\gamma^t$. We will see that although searching for an optimal policy (a rule for choosing the next arm, based on history, such that expected rewards are maximal) may be infeasible, the structure of an optimal policy is based on an *index* value that may be computed for each arm independently. The optimal policy will just choose next the arm of highest index, and update the index value (of the chosen arm only) based on the observed result, thereby breaking down the optimization problem to a small set of independent computations.

13.2 Bayesian MAB and Index Policy

We will assume that each arm is a *Markovian* process and that the values are discounted. In time t the reward value is r_t and the total value is $V = \sum_{t=0}^{\infty} r_t\gamma^t$ where $\gamma \in [0, 1)$. Also, we will assume that the model is a *Semi-Markovian Process* with S states, i.e.,we may spend more than one time unit in a state between two consecutive transitions. Each state s has $R(s), T(s)$ which are random variables that define the reward value and time spent in state s, respectively. The transition function $P(s_i|s_j)$ is the probability of switching to state s_i from state s_j.

Index Policy gives an index value for each arm based on its state, and chooses the one with the highest value. Thus, the computation is done separately for each arm (combined with finding the maximum), so is not exponential w.r.t. the number of arms.

13.2.1 Example: Single Machine Scheduling

In this section we show two settings of jobs scheduling. Only one of them has an index policy which is optimal.

Index Policy Existence

There are n jobs, each job i has its cost $C(i)$ and execution time $T(i)$ which are random variables (independent between i and j, but might be dependent for a given i). A solution to the problem is a permutation π on the jobs order. Let $cost(\pi)$ be the cost of a permutation.

$$C(\pi) = \sum_{i=1}^{n} C(i)(\sum_{j;\pi_j \leq \pi_i} T(j))$$

The goal is to find $\pi^* = argmin_\pi E[C(\pi)]$.

The following claim shows an index policy which is optimal.

Claim 13.1 *The optimal ordering of jobs in the single machine scheduling setting is by decreasing* $\frac{E[C(i)]}{E[T(i)]}$.

Proof: Let π^* be an optimal policy. Consider j_1 and j_2, two of the n jobs to be performed sequentially by π^*. Since the costs related to the rest of the jobs are the same regardless of the order in which j_1 and j_2 are performed, we can assume that j_1 and j_2 are the only jobs (i.e.,$n = 2$, $j_1 = 1$, $j_2 = 2$). Now, the total costs if performing first j_1 and then j_2 are $E[C(1)T(1) + C(2)(T(1) + T(2))]$, and the total costs if performing the jobs in the reversed order are $E[C(2)T(2) + C(1)(T(1) + T(2))]$. Therefore, an optimal policy will perform j_1 before j_2 if and only if

$$E[C(1)T(1) + C(2)(T(1) + T(2))] \leq E[C(2)T(2) + C(1)(T(1) + T(2))] \tag{13.1}$$

Using the linearity of the expectation and the independence of the jobs, we get:

$$\begin{aligned} E[C(1)T(1) + C(2)(T(1) + T(2))] &= E[C(1)T(1)] + E[C(2)(T(2)] + E[C(2)]E[T(1)] \\ E[C(2)T(2) + C(1)(T(2) + T(1))] &= E[C(2)T(2)] + E[C(1)(T(1)] + E[C(1)]E[T(2)] \end{aligned}$$

Using that in (13.1) we get $E[C(2)]E[T(1)] \leq E[C(1)]E[T(2)]$, which implies $\frac{E[C(2)]}{E[T(2)]} \leq \frac{E[C(1)]}{E[T(1)]}$

□

In this first example, we see that the optimal policy is an *index policy*, that is, a policy that is based on an index value function (that may be evaluated independently for each possible option) and at each decision time selects the option having highest index. In the single machine scheduling setting the options at each decision time are the jobs to be handled next by the machine and the index function of job i is $\frac{E[C(i)]}{E[T(i)]}$. Also note the simple interchange argument in the proof - we will use similar interchange arguments throughout.

No Index Policy

Note that in the proof we used the fact that the jobs are independent processes. If this is not the case, an index policy does not necessarily exist. Assume the following two distributions

Probability	C(1)	C(2)	T(1)	T(2)
1/2	100	10	100	100
1/2	100	100	10	100

Probability	C(1)	C(2)	T(1)	T(2)
1/2	100	10	10	100
1/2	100	100	100	100

Note that the jobs are dependent. Furthermore, note that in the first distribution an optimal policy would prefer to schedule job 1 before 2 as opposed to the second distribution. Recall that an index depends on its arm only. The two distributions are identical from each arm's perspective, so the index must be the same on both distributions. Therefore there is no index policy which is optimal.

13.2.2 Example: Sponsored Search

Assume there are n advertisements where each advertisement i has a value v_i, click probability c_i and a termination probability p_i. When a user clicks on advertisement i, he terminates and we get v_i. He may also terminate without clicking at any advertisement. We would like to order the advertisements for achieving a maximal value before the user terminates. Define the value of a permutation π as

$$Value(\pi) = v_{\pi(1)} \cdot c_{\pi(1)} + (1 - c_{\pi(1)}) \cdot (1 - p_{\pi(1)}) \cdot Value(\pi(2, \cdots, n))$$

That is, the first advertisement can be chosen, and therefore its value is multiplied by the click probability, or the first one is not chosen and then we calculate the value of the permutation without 1. Our goal is to find $\pi^* = argmax_\pi E[Value(\pi)]$

Claim 13.2 *The Index Policy* $\frac{v_i}{1-p_i+\frac{p_i}{c_i}}$ *descending is* π^*

Proof: Let π^* be the optimal solution. Again, we use an interchange argument. A similar reasoning as in the previous examples indicates that we may restrict our attention to two advertisements only, which without loss of generality we assume are advertisements 1 and 2. The value of performing 1 before 2 is $c_1v_1 + (1-c_1)(1-p_1)(c_2v_2)$ whereas the value

of performing 2 before 1 is $c_2v_2 + (1-c_2)(1-p_2)(c_1v_1)$. Therefore, an optimal policy will perform 1 before 2 if and only if

$$\begin{aligned} c_1v_1 + (1-c_1)(1-p_1)(c_2v_2) \geq c_2v_2 + (1-c_2)(1-p_2)(c_1v_1) &\Rightarrow \\ c_1v_1 + c_2v_2 - c_1c_2v_2 - p_1c_2v_2 + c_1c_2p_1v_2 \geq c_2v_2 + c_1v_1 - c_2c_1v_1 - p_2c_1v_1 + c_2c_1p_2v_1 &\Rightarrow \\ v_1(1 + \frac{p_2}{c_2} - p_2) \geq v_2(1 + \frac{p_1}{c_1} - p_1) &\Rightarrow \\ \frac{v_1}{1 - p_1 + \frac{p_1}{c_1}} \geq \frac{v_2}{1 - p_2 + \frac{p_2}{c_2}} & \end{aligned}$$

□

13.2.3 Example: Object Search

Assume we search for an object which is placed in one of n places. The probability it is in place i is p_i. If we look in place i and its there, the probability we will find it is q_i. The cost of looking in place i is c_i. We are looking for a policy that sequentially choose places to be searched such that the average cost of finding the object is minimal.

Note that $p_1...p_n$ can be thought as our prior belief, and if upon searching in place i the item is not found, the probability p_i is updated according to Bayes rule:

$$p_i^{new} = Pr[\textit{item in place } i | \textit{item not found upon searching place } i] = \frac{(1-q_i)p_i}{1-q_ip_i}$$

and

$$p_j^{new} = Pr[\textit{item in place } j | \textit{item not found upon searching place } i] = \frac{q_j}{1-q_ip_i}$$

Claim 13.3 *The permutation according to $\frac{c_i}{p_iq_i}$ ascending is optimal.*

Proof: Let π^* be the optimal solution. By interchange argument we can restrict ourselves to considering two consecutive places in π^*, assume places 1 and 2. The cost of searching place 1 before 2 is $c_1 + (1-p_1q_1)c_2$ whereas the cost of searching place 2 before 1 is $c_2 + (1-p_2q_2)c_1$. Therefore, an optimal policy will perform 1 before 2 if and only if $p_2q_2c_1 \leq p_1q_1c_2$, that is $\frac{c_1}{p_1q_1} \leq \frac{c_2}{p_2q_2}$ □

Notice that since the index depends on p_i which is updated after each round, the places should be resorted after each step.

13.2.4 Example: Bernoulli Multi Armed Bandit

We are given n arms $B_1 \dots B_n$. Each arm B_i when selected has an (unknown) probability of success θ_i (Bernoulli random variable $\text{Ber}(\theta_i)$). At a sequence of decision times $t =$

$0, 1, 2 \ldots$ we select an arm i, and (if successful) earn a γ-discounted reward γ^t. Given a prior probability distribution on the values $\{\theta_i\}_{i=1}^n$, our goal is to find an optimal rule for the sequence of arms chosen such that the average of the $\gamma-$discounted sum of rewards over time is maximal. As before, the probability distribution of θ_i is updated according to Bayes' rule after observing the result of every selection. For example, if the prior distribution of θ_i is Beta$(1, 1)$ (i.e.,uniform over $[0, 1]$) then after observing a_i successes and b_i failures in $a_i + b_i$ selections of arm i, the posterior probability distribution for θ_i is Beta$(1 + a_i, 1 + b_i)$. Note that if the probability distributions for θ_i are Beta(α_i, β_i) then the obvious greedy policy that at each step chooses the arm of highest index $\frac{\alpha_i}{\alpha_i+\beta_i}$ is not optimal. This is because given two arms of the same index value $\frac{\alpha_i}{\alpha_i+\beta_i} = \frac{\alpha_j}{\alpha_j+\beta_j}$ but different times used (e.g.,$\alpha_i + \beta_i << \alpha_j + \beta_j$) an optimal policy will prefer arm i over j since the substantially larger information gain in observing B_i (which has much higher variance at this point) may be later used to achieve higher expected rewards.

One-Armed Bandit

To see how the expected total reward under the optimal policy may be calculated, consider the simple setting $n = 2$ where arm 1 having an unknown success probability (our "one arm"), and arm 2 having a fixed known success probability p. Now, denote by $R(\alpha, \beta, p)$ the expected total reward under an optimal policy, when the probability of success of arm 1 is $\theta \sim Beta(\alpha, \beta)$. $R(\alpha, \beta, p)$ satisfies the following recursion:

$$R(\alpha, \beta, p) = \max\{\frac{p}{1-\gamma}, \frac{\alpha}{\alpha+\beta}[1 + \gamma R(\alpha + 1, \beta, p)] + \frac{\beta}{\alpha+\beta}\gamma R(\alpha, \beta + 1, p)\} \qquad (13.2)$$

where $\frac{p}{1-\gamma}$ is the expected reward when choosing arm 2 indefinitely[1], and the other term sums two summands which are the optimal expected rewards when choosing arm 1 and observing a success, or a failure, respectively. We may therefore solve for $R(\alpha, \beta, p)$ iteratively, starting with an approximation[2] for all values of α and β such that $\alpha + \beta = N$. Since the influence of this step is multiplied by γ^N, we can take N such that $\gamma^N = \epsilon$. We continue by calculating iteratively for all values of α and β such that $\alpha + \beta = N - 1$ and so on. It can be shown that that the approximation error exponentially[3] decreases with N. An index value for arm 1 given a Beta(α, β) probability of success may be the value of p for which the max in (13.2) is over two expressions of the same value. In what follows we formalize this notion and prove the existence of the Gittins index and its form. We start with the formal model.

[1] if it is optimal to choose arm 2 once, then it remains optimal thereafter since the information before choosing arm 2 is the same as the information after observing the result

[2] larger values of $\alpha + \beta$ imply higher concentration around the true success probability θ, and therefore we are able to provide increasingly good approximations of R as we increase the initial $\alpha + \beta$

[3] an ϵ-approximation to R for $\alpha + \beta = N$ results in an $\epsilon\gamma$-approximation to R for $\alpha + \beta = N - 1$

13.3 Bayesian MAB Model

Let $B_1 \dots B_n$ be n arms, and $S_1, ..., S_n$ be their states sets. At time t, each arm B_i is in a specific state $x_i(t) \in S_i$. When an arm B_i is selected at time t, it changes its state to $y \in S_i$, with probability $p_i(y|x_i(t))$, and observes a reward $r(x_i(t))$. Unselected arms stay in their current states. We wait $T(x_i(t))$, until the next time we can select another arm. $T(x_i(t))$ is set according to a probability distribution.

Our goal is to find a policy (a rule that given the history and the problem parameters chooses which arm to select at every decision time) that maximizes the average (over realizations[4]) of the $\gamma-$discounted sum of rewards over time. The contribution of B_i, that has been selected in $t_1, ..., t_l, ...$ to that average is:

$$E[\sum_{t_l} \gamma^{t_l} r(x_i(t_l))] \tag{13.3}$$

It will be convenient to consider the observed reward $r(s)$ (where s is the state of the selected arm at decision time t) as being 'spread' over the time interval ending in the subsequent decition time $t + T(s)$. We therefore define the reward *rate* $\overline{r}(s)$ as follows:

$$\overline{r}(s) = \frac{E[r(s)]}{E[\int_0^T \gamma^t dt | x(0) = s]}$$

Note that $E[\int_0^T \gamma^t \overline{r}(s) dt | x(0) = s] = E[r(s)]$ and therefore the two reward methods are equivalent with respect to the target (13.3). It will also be convenient to refer to the arm choice process as being continuous between decision times - i.e.,the arm is being chosen throughout the time period (resulting in $\overline{r}(s)$ reward per unit of time) until the next decision time, like the regular Markovian model. Now, we define for a fixed time interval $[0, T)$ the affective

$$w(T) = \int_0^T \gamma^t dt = \frac{1 - \gamma^T}{\ln \frac{1}{\gamma}} \tag{13.4}$$

And note that for such a fixed T we have

$$E[\int_0^T \gamma^t \overline{r}(s) dt | x(0) = s] = E[r(s)] = E[w(T)]\overline{r}(s) \tag{13.5}$$

It is assumed that at every decision time t all the states $x(t) = (x_1(t), \dots, x_n(t))$ and problem parameters (e.g.,the discount factor γ, the transition distributions p_i and reward function r) are known to the policy. Therefore, optimizing (13.3) is possible by state space evaluation methods such as dynamic programming. Such methods however are computationally infeasible due to the exponential size of the state space.

[4]all expectations are over realizations, unless explicitly indicated otherwise

In what follows we will see that the optimal policy for (13.3) is an *index* policy - a policy that assigns to each arm an index value that only depends on its state regardless of the states of the other arms . At each decision time the policy selects the arm of the highest index value. In doing so, we replace a problem of evaluating values of $\prod_i |S_i|$ states (exponential in n) with n independent computations of the values of $|S_i|$ states for each arm.

13.4 Gittins Index: Finite number of states

The set of states S_i is finite for every B_i, and is disjoit with S_j for every $j \neq i$ (i.e.,$S_i \cap S_j = \phi$). Thus $S = \bigcup S_i$ is finite, let us denote $|S| = N$.

We first show that at any decision time it is optimal to choose the arm of maximal reward rate, and then we use this to prove (by induction on the number of states $|S|$) that an optimal index policy exists. Furthermore, the construction in the proof will serve to define the index.

Claim 13.4 *It is optimal to choose an arm which is in state* $s_N = \arg\max_{s \in S} \overline{r}(s)$

Proof: Note that it is not necessarily the case that there *is* an arm in state s_N, the claim is that *if* there is then any optimal policy will choose it right away. The proof is stems from the static nature of the unselected arms states.

Assume that arm B_1 is in state s_N at time 0 ($x_1(0) = s_N$). We use a simple interchange argument: assume there is an optimal policy that chooses a sequence of arms in states different than s_N, and eventually chooses B_1; By choosing B_1 in time 0, we can get another optimal policy.

Let π be an optimal policy that after a period of length τ, collecting an accumulated reward R, chooses B_1 until the next decision time $\tau + T$. The reward observed by π during the interval $[0, \tau + T)$ is $V(\pi) = R + \gamma^\tau r(s_N) = R + \gamma^\tau w(T)\overline{r}(s_N)$.

We will compare the accumulated reward of π with that of a policy π' that chooses B_1 at time 0 for a period of length T and then chooses the same sequence as π during a period of length τ and is identical to π thereafter. Note that the states of the arms at time $T+\tau$ is the same for both policy realizations. The reward observed by π' during the interval $[0, \tau + T)$ is $V(\pi') = r(s_N) + \gamma^T R = w(T)\overline{r}(s_N) + \gamma^T R$. We consider the difference between the reward of π' and the reward of π:

$$V(\pi') - V(\pi) = (w(T)\overline{r}(s_N) + \gamma^T R) - (R + \gamma^\tau w(T)\overline{r}(s_N)) = w(T)\overline{r}(s_N) - R(1-\gamma^T) - \gamma^\tau w(T)\overline{r}(s_N)$$

Now, by the definition of s_N we have that $R \leq w(\tau)\overline{r}(s_N)$ and therefore the above difference is at least

$$V(\pi') - V(\pi) \geq \overline{r}(s_N)[w(T) - w(\tau)(1-\gamma^T) - \gamma^\tau w(T)] = \overline{r}(s_N)[w(T)(1-\gamma^\tau) - w(\tau)(1-\gamma^T)] = 0$$

where the last equality is by (13.4). We conclude that choosing the state of global maximum reward rate is optimal. □

We now use the claim to constructively prove that an optimal index policy exists:

Theorem 13.5 *If the number of states S is finite ($|S| = N$), then there exists an optimal index policy. Furthermore, the index values may be iteratively computed as follows:*

$$v(s_j) = \frac{E[\sum_{t_l<\tau} r(x(t_l))\gamma^{t_l} dt | x(0) = s_j]}{E[\int_0^\tau \gamma^t dt | x(0) = s_j]}, \qquad j = N, N-1, \ldots, 1 \tag{13.6}$$

Where the expectations above are over realizations that start with an arm at state $x(0) = s_j$ and continue (arm chosen again and again at decision times t_l) until a decision time τ in which the state of the arm is no longer in the set of already computed 'higher priority' values $\{s_N, \ldots, s_{j+1}\}$

Proof: First we prove by induction on the number of states that there is an optimal index policy (i.e.,that there is an ordering of the states such that it is optimal to choose the state of highest order). When there is a single state this is trivial. Now, assume the existence of such an ordering for a problem of $N-1$ states. We can now consider a modification of the given problem to a problem of $N-1$ states such that the rewards and decision times of an optimal policy for the original setting are the same as the rewards and decision times of an optimal index policy for the modified setting:

We eliminate the state of highest reward rate (s_N) by modifying the probabilities of transitions $p(y|s)$, reward rates $\overline{r}(s)$, and decision times $T(s)$ such that whenever an arm reaches state s_N at a decision time it is automatically selected (therefore the actual decision times in the modified setting are until no arm is in state s_N). By the inductive assumption, there is an optimal index policy for the modified setting (implying an ordering of the $N-1$ states at every decision time, that only depends on the state). By the claim above, any optimal policy for the original setting of N states selects an arm at state s_N when available. Therefore, the combination of the selection rule of state s_N with the optimal index policy for the other $N-1$ states forms an optimal index policy for the original setting.

We now turn to explicitly formulate the index value based on the above construction. First note that $\overline{r}(s_N) \geq \overline{r}_1(s_{N-1})$ where $\overline{r}_1(s_{N-1})$ is the maximal reward rate of the best arm s_{N-1} in the modified setting not including s_N. Therefore the list of non-increasing, iteratively computed values

$$v(s_j) = \overline{r}_{N-j}(s_j), \quad j = N, N-1, \ldots, 1$$

may serve as the index values of the states in S, where $\overline{r}_{N-j}(s_j)$ is the maximal reward rate of the best arm s_j in the modified setting not including $\{s_N, \ldots, s_{j+1}\}$. By the construction of the modified settings we have (13.6). □

13.5 Gittins Index: Infinite Number of States

In this section we will explore the general form of an optimal index policy assuming that it exists. Two additional existence proofs (not assuming finite state space) are given in

subsequent sections. To simplify notation we assume from now on that the decision times are fixed at times $t = 0, 1, 2....$ The results apply and are easy to generalize to the case of random decision times.

13.5.1 One Armed Bandit

We start by observing that the infinite horizon accumulated rewards of a single state fixed $\lambda-$reward arm is $\frac{\lambda}{1-\gamma}$, denote such an arm by $B(\lambda)$. Consider a setting of two arms, B and $B(\lambda)$, where the expectation reward of B (our one arm) is unknown. An optimal policy that switches from arm B to arm $B(\lambda)$ at some decision time $\tau > 0$, will never switch back to B. The information regarding B in future decision times is the same as the information that was available at time τ and resulted in choosing $B(\lambda)$. We conclude that the maximal average reward is the optimal choice of the *stopping time* τ:

$$\sup_{\tau>0} E[\sum_{t=0}^{\tau-1} \gamma^t r(x(t)) + \gamma^\tau \frac{\lambda}{1-\gamma} | x(0) = s_0] \tag{13.7}$$

where s_0 is the starting state of arm B, the average is over all realizations of the state transitions and rewards of arm B, and the supremum is over all functions τ that associate a stopping time in $\{1, 2, \ldots\}$ to a realized states history[5]. We are looking for the fixed reward λ^* that makes the two arms equivalent, e.g., equally optimal to switch to $B(\lambda^*)$ initially, or wait for the optimal switch time. Therefore $\lambda^*(s_0)$ may serve as the index value of arm B. $\lambda^*(s_0)$ is satisfying:

$$\sup_{\tau>0} E[\sum_{t=0}^{\tau-1} \gamma^t r(x(t)) + \gamma^\tau \frac{\lambda^*}{1-\gamma} | x(0) = s_0] = \frac{\lambda^*}{1-\gamma}$$

or equivalently

$$\sup_{\tau>0} E[\sum_{t=0}^{\tau-1} \gamma^t r(x(t)) - (1-\gamma^\tau)\frac{\lambda^*}{1-\gamma} | x(0) = s_0] = 0$$

The left hand side of the above equation, the supremum of a decreasing linear function of λ^* is convex and decreasing in λ. Therefore, the above equation has a single root that may also be expressed as follows (since $\frac{1-\gamma^\tau}{1-\gamma} = \sum_{t=1}^{\tau-1} \gamma^t$):

$$\lambda^*(s_0) = \sup\{\lambda | \sup_{\tau>0} E[\sum_{t=0}^{\tau-1} \gamma^t [r(x(t)) - \lambda] | x(0) = s_0] \geq 0\} \tag{13.8}$$

The above provides an economic interpretation of λ^* as the highest rent (per period) someone (who has an optimal stopping policy τ) may be willing to pay for receiving the rewards of

[5] A stopping time is a mapping from histories to a decision of either to continue or to stop

B. From (13.8) we get that λ^*, the index value of arm B at state s_0, is of the following form:

$$v(B, s_0) = \lambda^*(s_0) = \sup_{\tau>0} \frac{E[\sum_{t=0}^{\tau-1} \gamma^t r(x(t))|x(0) = s_0]}{E[\sum_{t=0}^{\tau-1} \gamma^t | x(0) = s_0]} \tag{13.9}$$

Note that it is a legitimate index since it only depends on the state and parameters of B. Note also that (13.9) coincides with (13.6) since the optimal stopping time τ is inherent in the construction described in the proof of Theorem 13.5.

Finally, consider the optimal stopping time τ in (13.8), which is characterized by the set of stopping states $\Theta(s_0)$. It can be shown that any state s having index value $v(B, s) < v(B, s_0)$ must be a stopping state, and any stopping state s must satisfy $v(B, s) \leq v(B, s_0)$:

$$\{s|v(B, s) < v(B, s_0)\} \subseteq \Theta(s_0) \subseteq \{s|v(B, s) \leq v(B, s_0)\}$$

This implies that an optimal policy will not stop at a state having higher index value than the index value of the initial state. Switches will always occur by reaching a state of lower index value than that of the initial state. The following example illustrates the power of using the index.

13.5.2 Multi Armed Bandit

In this section we extend the one armed value index to the multi armed case.

Theorem 13.6 *An Index policy with respect to*

$$v(B_i, s) = \sup_{\tau>0} \frac{E[\sum_{t=0}^{\tau-1} \gamma^t r(x_i(t))|x_i(0) = s]}{E[\sum_{t=0}^{\tau-1} \gamma^t | x_i(0) = s]}$$

is optimal.

Let us denote the numerator and denominator of the index defined in Theorem 13.6 by $R_\tau(B_i, s)$ and $W_\tau(B_i, s)$ respectively, so we have $v(B_i, s) = \lambda_i(s) = \sup_{\tau>0} \frac{R_\tau(B_i,s)}{W_\tau(B_i,s)}$. We first prove the following interchange claim:

Claim 13.7 *For two arms B_1 and B_2 at states x_1 and x_2 respectively at time t, if*

1. $\lambda_1(x_1) > \lambda_2(x_2)$
2. $\tau = \tau(x_1)$ *the optimal stopping time of* B_1 *at state* x_1
3. σ *an arbitrary stopping time for* B_2 *at time state* x_2

then the expected reward is higher when selecting B_1 for a period τ and then selecting B_2 for a period σ than the expected reward when the order is reversed.

Proof: $\lambda_1(x_1) > \lambda_2(x_2) \Rightarrow \frac{R_\tau(B_1,x_1)}{W_\tau(B_1,x_1)} > \frac{R_\sigma(B_2,x_2)}{W_\sigma(B_2,x_2)}$. Now, since for any $\sigma > 0$ we have $W_\sigma(B_i,s) = \frac{1-E[\gamma^\sigma|x_0=s]}{1-\gamma}$, the last inequality is equivalent to $\frac{R_\tau(B_1,x_1)}{1-E[\gamma^\tau|x_1]} > \frac{R_\sigma(B_2,x_2)}{1-E[\gamma^\sigma|x_2]}$ which in turn is equivalent to $R_\tau(B_1,x_1) + E[\gamma^\tau|x_1]R_\sigma(B_2,x_2) > R_\tau(B_2,x_2) + E[\gamma^\sigma|x_2]R_\tau(B_1,x_1)$. The left side of the last inequality is the expected reward when selecting B_1 for a period τ and then selecting B_2 for a period σ, while the right side is the expected reward when the order is reversed. □

We are now ready to prove Theorem 13.6:

Proof: For a given setting and the index (13.9) define a parameterized class of policies Π_k. A policy π is in Π_k if it makes at most k arm selections that are not the arm of highest index value (at decision time). We will show by induction on k that an optimal policy belongs to Π_0. First, consider $\pi \in \Pi_1$. We use the interchange claim 13.7 to show that π is not optimal. Indeed, consider the time t_0 in which π deviates and selects arm B_2 (having index λ_{2,t_0}) instead of arm B_1 of maximal index[6] $\lambda_{1,t_0} > \lambda_{2,t_0}$ (without loss of generality we may assume $t_0 = 0$ so $\lambda_{1,0} > \lambda_{2,0}$). Since π may not deviate again, arm B_1 will get selected as soon as $\lambda_{2,\sigma} < \lambda_{1,0}$, and remain selected for the optimal period τ. By the interchange claim 13.7, the reward of π during time $\sigma + \tau$ is less than the reward of a policy π' that reverses the arms order and selects arm B_1 first for a period of length τ followed by arm B_2 for a period of length σ (and is identical to π thereafter). Note that the states of B_1 and B_2 at time $\tau + \sigma$ do not depend on which policy was used. We conclude that π is not optimal and that optimal policies restricted to Π_1 should never exercise the (single) option to deviate. Therefore, optimal policies restricted to Π_k should never exercise their last option to deviate, and (inductively restricting attention to $\Pi_{k-1}, \Pi_{k-2}, \ldots$) we conclude that the Gittins index policy is optimal in Π_k. We are not done since there might be a better policy in Π_∞, which is not accounted for in the induction. Assume that the optimal policy π^* is in Π_∞ and not in Π_0. Given any $\epsilon > 0$, for a sufficiently large k, there exists an ϵ-optimal policy in Π_k (since ϵ determines a time horizon after which the discounted rewards are of negligible influence) which, by the above reasoning belongs to Π_0. Since Π_0 holds an optimal policy for any $\epsilon > 0$, it also holds the optimal policy. □

[6]Note that if multiple arms have maximal index (i.e.,in case B_1 is not unique) it does not matter which arm of maximal index is selected first, and therefore without loss of generality we may assume that B_1 is selected.