Course

Business Intelligence and Performance Management

Year 2015-2016

Multiarmed bandits

Part 1

Prof. Nicola Ciaramella

Sommario

1 INTRODUCTION

1.1 Motivations

The multi-armed bandits (MABs) model is conceived to frame a lot of decision problems form many fields, with a lot of both theoretical and practical applications. In this course we are interested in providing a reference framework including both concepts and tools, staying at an intuitive informal level. It is easy to find references on this topic on the web to get a deeper insight. Mathematical developments will be kept at an elementary level, sufficient to give an insight about possible implementations in real systems.

Typical business applications for MABs are in recommendation systems, e-commerce and online advertising, which are critical for the whole online economy.

Very often, coping with problems of these kinds, we have to manage a problem of *online learning with uncertainty, partial information and limited resources*.

Online learning means we learn in progress, while making decisions and taking actions: at each step we select an action out of a range of options, we get a certain outcome, more or less good, we collect some new information and we repeat the cycle.

Uncertainty denotes situations where the phenomenon we cope with is probabilistic in nature and/or we know it only by means of a probabilistic view. For example, in advertising online we try to submit readers advertisements they are likely to click on. We have no deterministic law telling us if a certain (individual) person will click or not on a certain advertisement; we neither have a deterministic collective-level law like "3 out of 100 readers will click". We have to reason with probabilistic and statistical statements like "the probability that 3 or more people out of 100 will click on this advertisement is 10%".

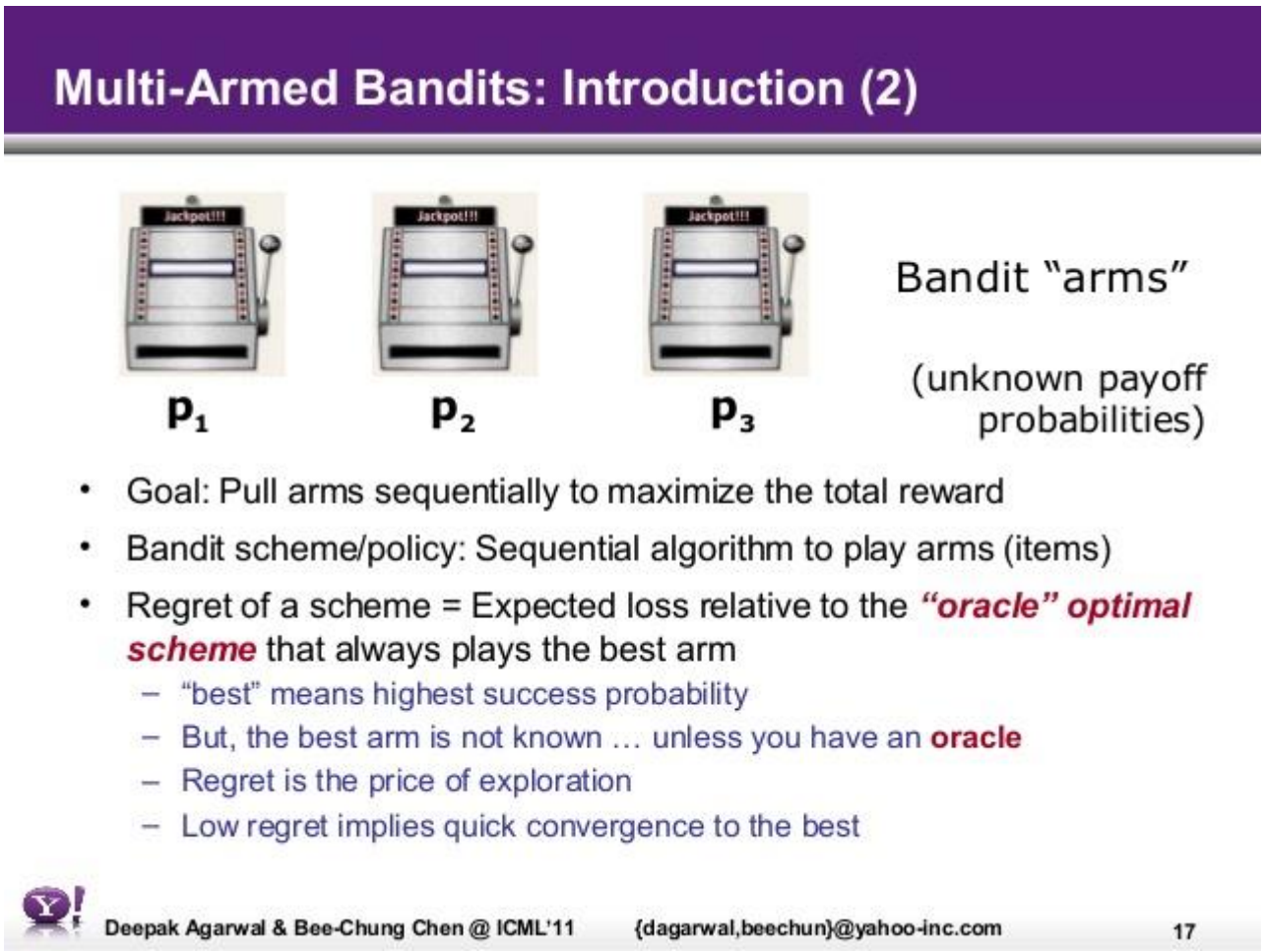
Partial information, in this context, means that once we have selected a certain action, we get information about the outcome of that action (how many clicks we got) but not about the possible outcome of actions we could have chosen (how many clicks another advertisement would have got, if we had submitted it to readers).

Limited resources says we can make only a limited number of decisions and take a limited number of actions, because we have a certain "budget" and/or we have a limited time horizon. For example, we have to show three advertisements to readers of our site for a total of one million times, each time choosing the advertisement we prefer. After one million rounds the game is over, and we cannot continue data analysis. This means we have to understand which is the best policy for

advertisements delivering during the time we play the game of advertising. We are not interested very much in understanding in the long run or asymptotically: we have to make timely decisions, so we need timely understanding of the game rules (which are only partially known, because of uncertainty).

Readers are invited to keep in their minds that we are reasoning about *economic problems*: our goal is maximizing utility (money or something else). For us knowledge and information are precious, but they are not the ultimate goals, instead they are resources to exploit to gain utility.

1.2 Problem formulation



We have a collection of one-armed bandits (jargon for slot-machines) and are endowed with a certain amount of tokens. At each stage of the game we select a bandit, insert a token into it and win an amount of money. Bandits are randomized: at each stage they can give a different amount of money. We assume each bandit has an inner probabilistic law, so it is random but governed by a probabilistic law. We do not assume every bandit shares the same inner law; indeed, in practice we will face bandits with different, sometimes very different, laws. This means there is some hidden

logic in this game: we do not initially know it, but we are able to gain some information about it while playing. If bandit A systematically delivers more money than bandit B for many rounds, we can reasonably assume it will continue to do this in the future (reasonably, not certainly).

Our goal is to maximize the overall amount of money we will have gained when we finish our tokens.

Remembering that the amount of tokens is finite, the intuition is that:

- We have to discover which bandit is the best.
- So, we need to sometimes choose a bandit we think is not the best in order to *explore* it.
- Once we are confident to have discovered the best bandit, we *exploit* it choosing it repeatedly.
- We need discover the best bandit as soon as possible.
- Then, we have to balance exploration and exploitation, choosing the "right" number of exploratory choices at the "right" moments on the "right" bandits.

The above considerations are far from rigorous and univocal: they are only a first guideline to start reasoning.

Take note that not only do we assume each bandit has an inner program governing the probabilistic outcomes, but we also assume that these inner programs are invariant over time. This assumption is realistic in some real-life contexts, not in others. We use it to simplify the problem.

2 COMMON POLICIES

In this section we will examine some policies commonly used to manage the exploitation-exploration trade-off.

2.1 Introductory example

Let us take only two bandits and assume we have 100 tokens to use.

At the first round, we obviously have no preference, so we choose random: say we choose bandit A. We get 3 dollars as *reward*. This does not mean that bandit A will give a reward of 3 in future rounds.

The situation is:

A	3								
B									

One could possibly say “I am interested in gaining at least 200 dollars over the whole game, that is an average reward of 2 per round. Therefore, the first outcome of bandit A is satisfactory for me and I insist on it”. This statement is meaningful, but in this context we are interested in maximizing the outcome, not in satisfying a fixed goal, so we do not follow this policy.

Intuitively, now we have to choose B to make possible comparing the bandits. We do it and get a reward of 2 dollars.

A	3								
B	2								

In the table we do not distinguish the temporal order of outcomes (A at round 1, B at round 2) because of the assumption on time-invariance of probabilistic distributions of bandits reward laws.

Now we compare the bandits and say that A is better *until now*. It is perfectly reasonable to choose A at the third round.

Imagine we get 0 dollars: the situation becomes

A	3	0							
B	2								

Now B is better *in average until now*, because it has an average value of 2 versus 1.5 for A. This suggest we have to choose B at round 4. This drives us to a first draft of policy: choose the bandit which is best in average until now.

More interestingly, imagine that at round 3 we choose A and get a reward of 2:

A	3	2							
B	2								

Bandit A remains preferable, having average value of 2.5. Now do we have to *exploit* information we have collected so far and choose A, or to *explore* B to give it equal chances? (better said: to have the same amount of information we already have about A). Let us choose A again , getting reward 2 again:

A	3	2	2						
B	2								

The criterion of *the best bandit in average until now* favorites A because of its average 2.33 versus 2 for B. Though, we probably feel a growing impulse to choose B, because it is getting reasonable the hypothesis that it could turn off to be better than A. Possibly, A has simply been luckier until now. The intuition is that choosing B instead of A at this point we expect to sacrifice 0.33 dollars (2.33 of A minus 2 of B) in order to get fresh information, which could be useful to improve our future rewards (if B is really better). This impulse to exploration instead of exploration get stronger

if the difference in average values gets smaller. If we have chosen A 30 times with average reward 2.10 and B 3 times with average reward 2.05 then the impulse to explore B becomes compelling.

We are trading-off between money and information. This is not the classical apple vs. oranges comparison. Indeed, *information is money* in some sense. It is a source of profit. We see that our ultimate goal is gaining profit (overall reward) though we have a penultimate goal: gaining information. *To maximize expected profit we have to invest in acquiring information, paying a certain price for it*. This intuition plays a central role in a vast domain of algorithms used in common online business (advertising, e-commerce, news recommendation and so on).

Now we are going to examine some policies designed to balance the *exploitation-exploration trade-off*.

2.2 Exploration First policy

The first policy we define is

Exploration First

Control parameters: E = number of initial exploratory trials per bandit

Algorithm:

1. Try each bandit E times.
2. Select the bandit with the maximum average reward in phase 1.
2. Select that same bandit for each of the remaining rounds.

This policy is meaningful and interesting, but far from simple and easy to use.

The idea is devoting a first phase of the game to exploration, learning which bandit is the best with reasonable confidence then in a second phase exploit information gained in the first one. Perfectly reasonable.

The trouble is in *parameter tuning*, i.e. in choosing a good value for parameter E.

Let N = 100 and K = 2. If we choose E = 10, then we spend the first 20 rounds in exploration and the next 80 in exploitation. The expected reward of the whole game is:

$$0.20 \times \text{avgAB} + 0.80 \times \text{probA} \times \text{avgA} + 0.80 \times \text{probB} \times \text{avgB}$$

where:

- avgAB is the mean of mean value of A and mean value of B
- avgA and avgB are the mean values of A and B
- probA and probB are the probabilities of selecting A or B as winner of the first phase

Pay attention: all these values are unknown, we do not know reward probability distributions of A and B, they are indeed what we are trying to guess. We know that avgA > avgB implies probA >

probB, i.e. the bandit with greater expected value has a greater probability to win the first phase contest, but we do not know how much greater. Double greater in expected value does not imply double more probable to win.

The same is true if we choose E = 20 or any other value. So, we are not able to select E on a rational basis, *unless we have some prior knowledge before starting the game*. If we know, or confidently assume, that bandits have a probability distribution of the Normal Gaussian shape with mean and variance in certain ranges then we can use probability theory to choose a good value for E. This is a complex topic, requiring a background in probability calculus and we will not treat it. For our present purposes, it is enough to say that in business practice the Exploration First policy is commonly used, but often in very empirical style, with outcomes strongly depending on luck.

2.3 Greedy policy

Now a very intuitive policy we have already met:

Greedy

Control parameters: none

Algorithm:

1. Try each bandit once.
2. For each remaining round select the bandit with the maximum average reward at the time.

At each step we choose the leader insofar. This is different from Exploration First because the leader can change at each step. For example, A gets reward 3 at round 1 and B gets 2 at round 2. At this point A is the leader and we choose it for round 3. If it gets reward greater than 1 then we choose it again at round 4, because its average stays above 2. Otherwise, B becomes the new leader.

The algorithm is extremely simple, without parameters to be tuned. It can work well, but is also prone to very bad performance if unlucky. Imagine A has true distribution mean 5 dollars and B has 3 dollars. At round 1 it gets only 1 dollar because bad luck. At round 2 B is selected and gets 2. It is chosen again and stays above average reward 1 for the remaining 98 rounds. The final outcome will be around 3 dollars, while A was capable of 5 dollars, if only recognized as the better bandit, but it never got a second chance.

A possible cure for this risk is to initially try each bandit more than once, which means to have a first phase dedicated to exploration, as in Exploration First policy, and a second in which bandits can alternate as leader.

Exploration First Then Greedy

Control parameters: E = number of initial exploratory trials per bandit

Algorithm:

1. Try each bandit E times.

2. For each remaining round select the bandit with the maximum average reward at the time.

Indeed, Greedy is a particular instance of this policy where we choose parameter E = 1.

Another option, much more common, is the Epsilon Greedy policy we will see in a short.

2.4 Epsilon Greedy policy

First we introduce a strange policy:

Random

Control parameters: none

Algorithm:

1. At each round chose a bandit random.

Certainly not a complex algorithm: it performs exploration at each round, never exploiting what it has learnt. This behavior sounds foolish, but this policy in conceived not to work alone. Indeed it is a component of the most famous exploration-exploitation policy:

Epsilon Greedy

Input: none

Control parameters: ε (pronounce Epsilon) = exploration rate

Algorithm:

1. At each round chose follow Greedy with probability 1 - ε or Random with probability ε.

It is a mix of two policies, with a parameter balancing them. It is more understandable if stated in this way: at each round with probability 1 - ε you choose the leader insofar, with probability ε you choose a bandit random (maybe the leader itself).

Note that there are two random components: one to decide whether to choose greedily or random, a second to choose a bandit random (if so decided in the previous one).

It is usual to choose values for epsilon among 0.01, 0.05, 0.10 and 0.15. There is no rational basis for this, simply use.

Strangely enough, it is far from easy to design a policy capable to systematically beat Epsilon Greedy. Considering its simplicity, this can be very frustrating for algorithm designers.

Its drawbacks are evident:

- Even if you have a reasonable confidence to have found the best bandit, you cannot exploit this knowledge as intensively as you would like, because you are bounded to continue investing an epsilon fraction of your rounds in exploration.
- In general, the intensity of exploration is independent from particular features of the bandits you are playing with.
- It is difficult to choose epsilon, and the overall performance of the policy heavily depends on the value you choose initially.

Nonetheless, Epsilon Greedy is the first choice for every time one meets a multi armed bandit problem and it is used as a benchmark in scientific literature: first of all, experiment if your new algorithm systematically beats Epsilon Greedy.

2.5 Epsilon Greedy variants

The basic epsilon Greedy policy can be modified in many ways.

A first variant is the Epsilon Decreasing family of policies. It works like Epsilon Greedy at each round, but ε changes over time. Now the parameter to tune is not a number but a function that links time and ε. Examples of functions used to compute ε(t), the value of ε at time t, are:

- ε(t) = 1 / t
- ε(t) = 1 / square root of t
- ε(t) = 1 / logarithm of t

The choice is dictated by peculiarities of the concrete problem at hand, first of all by the total number of rounds.

More sophisticated schemes try to measure the uncertainty at time t and assign to ε a value increasing on it (the greater the uncertainty, the greater ε). The intuition is that if many bandits are close each other in value insofar then is reasonable to make more exploration choices (they have a limited cost and good chance to discover something new). The same when some bandit have been explored rarely but are not so bad insofar.

Another reasonable idea is to select non-leaders with different probabilities: it is not natural to select the second bandit or the 50^{th} in ranking insofar with the same probability.