18.310 lecture notes September 2, 2013

Sequential Choice

Lecturer: Michel Goemans

1 A game

Consider the following game. I have 100 blank cards. I write down 100 different numbers on the cards; I can choose any numbers I want, and you don't get to see what numbers I choose. So e.g., I might choose

$$1, 2, 10000000, \pi, -10000000, 10^{10^{1}0}, 0.01, \ldots$$

The deck of cards is then shuffled well, so that the cards are now in uniformly random order. The deck is placed before you face down, and the top card is turned over to show a number (maybe it's 1000000). You have two choices: you can say "stop", and choose this card, or skip it. If you skip it, the next card is turned over, so you can see the number, and again you may choose to stop and choose the card, or go onto the next.

The game ends once you pick a card; you win if the card you pick was the largest of all the cards. Note that the game would be very easy if you knew what numbers I wrote down; you'd just keep skipping until you see the card with the largest number. But since you know nothing about the numbers, it looks pretty hard...

It costs \$10 to play; if you win, I'll give you \$$P$. How big should P be before you're willing to play?

Surprisingly, you should play even if $P = 37$! There is a strategy that wins with probability at least $1/e = .367\ldots$. Let's first see a simpler strategy and argument that wins with probability at least $1/4$.

Here's the strategy: for the first 50 cards, always skip, no matter what the numbers are. After that, stop whenever you see a card that is largest of all the cards you've seen so far.

Claim 1. *If the largest card is in the 2nd half of the deck, and the second largest card is in the first half of the deck, then you win.*

Proof. After the first 50 cards are overturned, we have seen the second largest card. Thus we will not stop until we see the largest card (which is in the second half of the deck, and so hasn't been skipped). □

Thus

$$\begin{aligned}
\Pr(\text{win}) &\geq \Pr(\text{largest card in 2nd half}) \Pr(\text{2nd largest card in 1st half} \mid \text{largest card in 2nd half}) \\
&= \frac{50}{100} \cdot \frac{50}{99} \\
&> 1/4.
\end{aligned}$$

Note that our analysis was not completely tight; it's possible to win even if the 2nd largest card is not in the first half. For example, if the third largest card is in the first half, and the largest card appears in the 2nd half but before the second largest card.

Now lets try to use the same idea, but optimize things. Let n be the number of cards (previously $n = 100$, but no reason to restrict ourselves). Our strategy will be similar, but we will skip over the first t cards (previously, $t = 50$), and from then on pick a card if it's the largest we've seen so far. Our goal will be to analyze this strategy precisely for a given k, and then determine the best choice of k. Our primary interest will be for n large.

Theorem 1. *The probability of winning with this strategy is*

$$\Pr(win) = \frac{t}{n} \sum_{k=t+1}^{n} \frac{1}{k-1};$$

as $n \to \infty$,

$$\Pr(win) \to -\frac{t}{n} \ln(t/n).$$

Proof. □

Let's split the event that we win based on the position of the largest card:

$$\begin{aligned} \Pr(\text{win}) &= \sum_{k=t+1}^{n} \Pr(\text{pick } k\text{'th card, and } k\text{'th card is the largest}) \\ &= \sum_{k=t+1}^{n} \Pr(\text{pick } k\text{'th card} \mid k\text{'th card is the largest}) \Pr(k\text{'th card is the largest}). \end{aligned}$$

Now $\Pr(k$'th card is the largest$) = 1/n$, for any k. On the other hand, for $k > t$, the probability that we pick the k'th card, conditioned on it being the largest, is simply the probability

$$\begin{aligned} &\Pr(\text{pick } k\text{'th card} \mid k\text{'th card is the largest}) \\ &\qquad TODO \end{aligned}$$

The probability that the largest card rank 1 choice being the j'th one we look at is $1/n$. Given that it is the j'th possibility we look at, the probability that we successfully choose it is simply $k/(j-1)$, since we will choose it only if the best possibility among the first $j-1$ is in the first k. Summing on j, we get the success probability, given that we set our threshold to be k, is

$$P_k = \sum_{j=k+1}^{n} \frac{1}{n} \frac{k}{j-1} = \frac{k}{n} \sum_{j=k}^{n-1} \frac{1}{j}$$

Now, we can use the fact that $\sum_{j=1}^{n} \frac{1}{j} \approx \ln n$ to approximate this sum

$$P_k \approx \frac{k}{n} \ln(k/n).$$

2 Introduction

There are times in life when you will be faced with the problem of choosing among a large number alternatives, but you will be forced to decide which choice to make before seeing many or most of them.

A typical situation is when you are looking for an apartment to rent in a market in which really good apartments at reasonable prices are hard to find. Then, you typically examine potential places one by one. And if the one you currently see is just right, unless you decide right away to take it, somebody else will, and you will no longer have this choice available. This problem has traditionally been called the *secretary problem*, and the story attached to it was of an executive interviewing secretaries.

We give a mathematical model of this kind of situation by imagining that there are N alternatives, which at the beginning are entirely unknown to you; you view them one at a time, and must accept or reject each one immediately, without seeing any more.

In the first case, we consider the goal of finding the very best among these alternatives. We could also seek to choose one of the best two, or to find a strategy to minimize the expected rank of your choice among the N alternatives (where best is rank 1, next best rank 2, and so on,) or make a choice that ranks among the top Q percent of the possible choices.

We are going to assume that you examine the choices in an entirely random order, so that you have no reason to believe that the unseen choices are better or worse than those you have already seen. Thus, if you have seen and passed k candidates, we assume that the next one to come along has an equal chance, or probability $1/(k+1)$, or lying in one of the $k+1$ intervals defined by the k choices you have already seen. In practical situations, this assumption can be false, so you should be cautious in applying the analysis below. For example, a classical strategy on the part of some real estate agents (or nasty trick, from the buyer's point of view) is to first show the buyer five or six really bad houses, and then show them a reasonable house that (possibly) one of her friends has listed. The buyers get an unrealistic sense of the quality of the houses on the market, and are likely to be fooled and pick the last (and still average quality) house.

3 Seeking the very best out of n possibilities

In situations in which we must make a choice like this, many of us tend to use one of two quite poor strategies. Either we loath the process of choosing so muech that we take the first possible alternative, or we procrastinate at choosing until there is only one possibility left. With either of these strategies, our chance of getting the best out of the n choices we could have seen is $1/n$. We can do much better.

How? Obviously, to do better we must examine the first possibility, and perhaps the next few, merely to learn about available choices. At some point, or threshold, we must decide to accept the next choice that is the best so far. If our goal is to accept only the best alternative, we must certainly reject any that has worst rank than any we have seen before.

We can analyze this problem based on the assumption that the choices are randomly ordered by merit, as follows. Suppose our threshold is the kth possibility, which means that we definitely reject the first k choices we see and pick the next one that is the best so far. Then we definitely lose if the rank 1 choice is among the first k seen. Suppose that the rank 1 choice is among the last $n-k$ we look at. Then, we will reject it if and only if the best one we saw before we looked at the rank 1 choice is among the first k. If it is, then we would have passed on it, and since everything we saw between it and the rank 1 choice is inferior to the next best, we would have waited for our rank one choice. On the other hand, if the best choice before the rank 1 choice is after our threshold k, we certainly would have taken it, and never reached the rank 1 choice.

We can write down a formula for our chance of success, given that our threshold is k. The probability of our rank 1 choice being the j'th one we look at is $1/n$. Given that it is the j'th possibility we look at, the probability that we successfully choose it is simply $k/(j-1)$, since we will choose it only if the best possibility among the first $j-1$ is in the first k. Summing on j, we get the success probability, given that we set our threshold to be k, is

$$P_k = \sum_{j=k+1}^{n} \frac{1}{n} \frac{k}{j-1} = \frac{k}{n} \sum_{j=k}^{n-1} \frac{1}{j}$$

Now, we can use the fact that $\sum_{j=1}^{n} \frac{1}{j} \approx \ln n$ to approximate this sum

$$P_k \approx \frac{k}{n} \ln(k/n).$$

If we differentiate this expression, we find that the maximum occurs when $\ln k/n = 1$, or $k = n/e$. The value of the probability is then $k/n = \frac{1}{e}$, so if these approximations are reasonable, our probability of getting the very best choice is around $1/e$, which is roughly .37.

The results are: with $n = 5$, the best k value is 2, which means you let the first two candidates go by and pick the next better one; k increases to 3 when $n = 8$, to 4 when $n = 11$, to 5 when $n = 13$, to 6 when $n = 16$, and so on. The best value for k is usually the one nearest to n/e.

The probability of successfully choosing the best candidate is bigger than $1/e$ for small values of n. Thus, for $n = 8$, choosing $k = 3$ gives a probability .4098 of success.

4 Calculating probabilities for choosing the very best on a spreadsheet

Now, let's look at how we might efficiently calculate the optimum value of the threshold on a spreadsheet; the point after which we might possibly choose a candidate. If we are trying to choose the very best candidate, then the formula in the previous section is fairly efficient to calculate in a straightforward manner: for each of the n possible values of the threshold, we have a nice formula for the probability of success. In fact, we don't even need to check all n possible thresholds; just the ones near n/e.

However, suppose we were trying to maximize the probability of choosing one of the best 10 candidates. The optimum strategy would then look like this: choose 10 thresholds k_1, k_2, …, k_{10}, and if you are past the i'th threshold, choose the current candidate if it is one of the best i you have seen so far. Now, there are roughly $nchoose10$ possible values for the 10 thresholds, and this would make calculations quite inefficient if you had to test even a small fraction of these possible values.

There is a clever way to get around this. This way works fairly well for any number of thresholds, and also works for quite a few variations on this problem. What I will explain in these notes is how to do it for the problem where your goal is to choosing the very best possibility, and I will assign some variation on this for homework. The idea is to work backwords from the last choice, and at every point compute the probability of success, given that you haven't chosen a candidate before this step. We will let P_k be the probability of success, given that we have let the first k candidates go by. We will use the fact that this probability P_k is independent of the candidates we have seen so far.

If we reach the last possibility without having chosen earlier, we must choose the last, and the probability that we win is $\frac{1}{n}$. Thus, $P_{n-1} = \frac{1}{n}$. Now, let's calculate P_{n-2}. suppose that we reach the second to last possible choice. If this choice is the best one we've seen so far, we should choose it. The probability that the next to last is the best one we've seen so far is $\frac{1}{n-1}$, and the probability that we win if we choose it is the probability that the last choice is not the optimal one, which is $\frac{n-1}{n}$. Now, if it's not best so far (with probability $\frac{n-2}{n-1}$, we let it go by, and our chance of winning is $P_{n-1} = \frac{1}{n}$. We have to let it go by, and if we do, our probability of winning is $P_{n-1} = \frac{1}{n}$. Thus, we get

$$\begin{aligned} P_{n-2} &= \Pr(\text{next} - \text{to} - \text{last is best so far}) \cdot \Pr(\text{next} - \text{to} - \text{last is optimal} \mid \text{best so far}) \\ &\quad + \Pr(\text{next to last is not best so far}) \cdot P_{n-1} \\ &= \frac{1}{n-1}\frac{n-1}{n} + \frac{n-2}{n-1}\frac{1}{n} \\ &= \frac{2n-3}{n(n-1)} \end{aligned}$$

Now, suppose that we know P_k, and want to calculate P_{k-1}. First, let's assume that we are past the threshold. Then we have

$$\begin{aligned} P_{k-1} &= \Pr(k\text{th is best so far}) \cdot \Pr(k\text{th is optimal} \mid \text{best so far}) \\ &\quad + \Pr(k\text{th is not best so far}) \cdot P_k. \end{aligned}$$

We know the probability that the kth guy is the best so far is just $\frac{1}{k}$, and the probability that it's not best so far is $\frac{k-1}{k}$, so the only probability we have to calculate to make this recursion work is the probability that the kth guy is optimal, given that it's the best so far. I think the easiest way to to think about this is to realize that the two events E_A and E_B are independent, where E_A is the event that the the kth candidate is the best of the first k candidates, and E_B is the event that the optimal candidate is among the first k. Then, it's easy to check that the probability that we want to calculate is $\Pr(E_B|E_A)$. Since these two events are independent, this probability is just $\Pr(E_B) = \frac{k}{n}$. I've left out a few details here, but you should have learned enough about probability earlier in this course to fill them in fairly easily.

Now, we can compute the probability of winning, given we choose k as the threshold recursively, as follows

$$\begin{aligned} P_{k-1} &= \Pr(k\text{th is best so far}) \cdot \Pr(k\text{th is optimal} \mid \text{best so far}) \\ &\quad + \Pr(k\text{th is not best so far}) \cdot P_k \\ &= \frac{1}{k}\frac{k}{n} + \frac{k-1}{k}P_k. \end{aligned}$$

It's fairly easy to check that this gives the same formula for P_k we saw earlier. But why is this more efficient? One reason is that this lets us compute all the P_k's at once, and then we merely need take the maximum of them. In fact, it becomes very easy to find the maximum because, starting with $k = n$, we can show that P_k increases at first, and then decreases, obtaining a maximum around $k \approx n/e$.

However, there's another thing we can do with this formula which generalizes more easily to other problems. Let's change the formula slightly. Let Q_k be the probability of winning with the optimal strategy, given that we have already let the first k choices go by. Then we have, by the same reasoning that gave us the formula for P_k,

$$\begin{aligned} Q_{k-1} &= \Pr(k\text{th is best so far}) \cdot \Pr(\text{optimal strategy} \mid k\text{th is best so far}) \\ &\quad + \Pr(k\text{th is not best so far}) \cdot Q_k \end{aligned}$$

But what is the optimal strategy if the kth item is best so far? It is either to choose the kth item or pass on it, depending on which gives you a larger probability of success. The probability of success if we choose the kth item is just $\frac{k}{n}$, and the probability of success if we pass on it is Q_k. Thus, we have the formula

$$Q_{k-1} = \frac{1}{k} \max\left(\frac{k}{n}, Q_k\right) + \frac{k-1}{k} Q_k.$$

Now, how do we generalize this to the problem where we want to find one of the top L items. We can write down a similar formula for Q_{k-1}. For each k, and for i going from 1 to L, we have to calculate the probability of success if we choose the k'th item, given that the k'th item is the i'th best so far. Then, if the k'th item is the i'th best so far, we choose it only if this gives us a higher probability of winning than Q_k.

5 Seeking one of the top two (or top L) choices.

The problem of finding the best strategy when you lower your standards and consider yourself successful if you choose on of the top two choices or one of the top L choices can be addressed just as the previous problem, but we have to consider the possibility of having more than one threshold.

Suppose we seek the best or second best. After the first threshold we accept the best candidate so far, and after the second threshold we take either the best or second best that comes along. When we seek to get one of the top L, we similarly want to consider L different thresholds.

The calculations are somewhat more complicated, and we leave the details for you. The results are: for $L = 2$ and small values of n, we can achieve a probability of success above 0.6. As n increases, our chance of success dwindles down to around 0.57.

6 Seeking the Best Expected Rank

This problem can also be handled by the same general approach. If there are n possible choices all together, we can compute a threshold for each stage, where we accept a candidate in some stage if its rank is less than than or equal to the threshold for that stage. Call this threshold $T(s)$, where s is the number of candidates still unseen.

When $s = 1$, there is only one candidate left after the one you are currently examining. We will be stuck with the rank of the final candidate if we pass on the current choice. This last choice has expected rank $(n + 1)/2$. Suppose the candidate we are looking at at this point has rank r out of the $n - 1$ we have seen so far. We can deduce that it therefore has expected rank $r(n + 1)/n$

among all n cndidates, so that we should accept it if $r(n+1)/n$ is better than $(n+1)/2$. we get this by noticing that with probability r/n the last candidate raises our rank by 1 above its present value of r.

Let us assume that goodness means low rank, so that rank 1 is best. Then our expected rank, before the $(n-1)$st cgiuce us $(n+1)/4$ if we choose the $n-1$st, and $(n+1)/2$ if we don't do so. We have a roughly 50-50 chance of doing either, so our threshold for the $(n-2)$nd choice is roughly $3(n+1)/8$. A similar computation can be made to deduce the expected rank before seeing the $(n-2)$nd if we have passed over the previous choices. This determines a threshold for our $(n-3)$rd choice, and we can continue back to the beginning.

Though this sounces somewhat complicated, it isn't, and it is very easy for a machine to determine the expected rank at the beginning for any reasonable n. The result is quite surprising. It turns out that the expected rank at the beginning is always less than 4, independent of n.

The best strategy appears to be roughly as follows. We let the first quarter of the choices go by. Then we take only the best until the next quarter of what is left, i.e., choosing the best in the next $(1/4)(3/4)n$ candidates, and so on. (Don't count on this. Figure it out for yourself.) So if you want to get an expected rank better than the 4th best, you can do so even if there are a billion candidates. Of course, you have to examine almost a billion of them to do this, which is probably ridiculously impractical.