Optimal Stopping Rules

Subhash Suri

January 9, 2020

1 Secretary Problem

- In TCS and math, toy problems or puzzles often serve as useful abstractions for understanding complex real-world problems. By stripping away unnecessary and irrelevant details, these simpler models can expose the underlying structure and complexity more clearly.

- The *secretary problem* is one such famous problem that models a number of situations where the goal is to decide which candidate (solution) to choose. More broadly, what rule to follow in deciding when to stop searching?

- The puzzle formulation and rules are as follows:

  1. We (an administrator) want to hire the best secretary.
  2. After the position is advertised, we receive a number of applications from the interested candidates. Suppose a total of n candidates have applied.
  3. We will interview each candidate in the *order* they applied.
  4. We assume a *total order* among these n candidates: *rank* 1 being the best, rank 2 second best etc.
  5. However, this ranking is *ordinal* (not cardinal), meaning that once we interview i and j, we can tell which one is better (relative ordering), but we do not assign any numerical (cardinal) values to their abilities.
  6. Unfortunately, it is an "seller's market" meaning that after we interview a candidate, we need to immediately decide whether to offer her the position (in which we stop the search), or reject her and continue the search. If we reject a candidate, she is no longer available.
  7. Our goal is to hire the *best* (top-ranked) candidate.

8. A reasonable mathematical model will be to assume that the order of candidates is a random one, and we want an algorithm that *maximizes* the probability that we hire the best secretary under these rules.

1.1 Small Examples

- To get some insight, suppose $n = 2$. Then, shall we accept or reject the first candidate? The best candidate is equally likely to be either the first or the second, so we have 50% chance if we accept the first one. We also have same prob. if we reject first, and hire second. So, the case of $n = 2$ is trivial, and either option gives us 50% success prob.
- What about $n = 3$?
  1. If we accept the first, we have only $1/3 = 33\%$ success prob because the best secy is among the last two with prob $2/3$.
  2. Suppose, instead, we always reject the first and hire the second one if she is better than first, otherwise dismiss her too. If we dismiss 2nd, we always accept 3rd.
  3. Following this strategy, we win in three $(2, 1, 3), (2, 3, 1), (3, 1, 2)$, and lose in three $(1, 2, 3), (1, 3, 2), (3, 2, 1)$.
  4. Thus, we improve our odds of winning from 33% to 50%.
- However, this type of case analysis quickly gets unwieldy, and we need a more sophisticated way to reason.
- The key point is that with n candidates, if we pick one at random, we have only $1/n$ chance of winning. So, with just 100 candidates, our odds are only 1%. Can we do better?

1.2 History of the Problem

- The problem is widely believed to have made its first appearance in Feb 1960 issue of Scientific American's Martin Gardner's recreational mathematics column. But its origins remain mysterious, and subject of debate among scholars.
- Harvard mathematician Fred Mosteller recalled hearing about in 1955 from Gleason, who heard it from someone else, and so on and so on. Many sources point to Merrill Flood as possible originator. Flood (U Michigan mathematician) famous for popularizing TSP problem (one of the most famous problems in CS), inventing the Prisoner's Dilemma (one of the most famous problems in Game Theory), etc. According to lore, Merrill M. Flood introduced the problem in 1949, calling it the fiancee problem in a lecture.

- The Secretary Problem is interesting because it models a fundamental dilemma inherent in a number of decision problems such as:
  1. Apartment rentals: in hot markets, grab it or lose it.
  2. Selling a house: accept an offer, or hold off for a better one, which may never materialize?
  3. Parking spots in cities: how greedy to be?
  4. How many partners to date before settling?
  5. Software release: when to stop debugging, and release?
  6. Similar issues in stock market investing, emergency planning, etc.
- Obviously, our toy problem has some unrealistic assumptions, but it gives a decent start. We will explore variants of the problem with more realistic assumptions and objectives later.

1.3 Look Then Leap and 37% Rule

- What makes Secretary Problem mathematically intriguing is the following dilemma:

  > *An algorithm for solving the problem can fail in two ways: stopping early or stopping late. If you pick one too early, you will miss the best; if you stop too late, you hold out for a better candidate who doesn't exist.*

- *Therefore, the optimal strategy must strike a balance.*
- One thing is clear: during the interview process, you should never hire someone who isn't the best seen so far!
- But being the "best so far" isn't good enough: the very first candidate, for instance, is best so far by definition, but may be a premature choice.
- Clearly, the rate at which we encounter best yet will decrease with time. The second applicant has a 50/50 chance, but the fifth candidate has only 1-in-5 chance of being best yet.
- Therefore, the best yet applicants get more and more impressive with time, but they also become more and more infrequent.
- A sensible strategy, therefore, would seem to be the following *Look-then-leap Rule*: we "look" at a *predetermined* fraction of the candidates, "gathering data," during which we will not choose anyone no matter how impressive. After that, we enter the "Leap"

phase, and commit instantly to anyone who outshines the best candidate encountered in the look phase.

- Let us revisit the $n = 3$ case. When we see the first candidate, we have *no information*, but when we reach the third candidate, we have *no choices*: we have to make an offer since we have dismissed all others!
- But when we see the 2nd candidate, we have a little bit of both information and choice: we know whether she is better than first, and we still have a choice of accepting or rejecting her.
- The natural strategy would appear to be this: hire her if she is better than first, and dismiss otherwise.
- This turns out to be the optimal strategy for $n = 3$ candidates.
- As the application pool grows to n candidates, the exact place to draw the line between Look and Leap phases turns out to be 37%: that is, look at the first 37% of the candidates, choosing none, and then leap at the first candidate who is better than anyone seen so far.
- What is even more remarkable is that this strategy also has 37% chance of success, namely, of finding the candidate! Weird, right?
- A strategy with 63% failure rate may not seem like something to boast—i.e. even our best strategy fails most of the time! But keep in mind that by choosing one at random would give a success prob. of *only* $1/n$, which worsens with the size of the applicant pool. By comparison, the 37% guarantee holds no matter how large the pool.

1.4 Mathematical Analysis of the Secretary Problem

- Let us formalize our model of the secretary problem.
    1. There is a single position to fill. There are n applicants for the position, and the value of n is known.
    2. The applicants are interviewed sequentially in random order, with each order being equally likely.
    3. The applicants can be ranked from best to worst unambiguously: specifically, after interviewing candidate X, the administrator knows whether X is the better than all previous candidates, but has no information about the future ones.
    4. Immediately after an interview, the applicant is either accepted or rejected, and the decision is irrevocable.

5. The decision to accept or reject an applicant can be based only on the relative ranks of the applicants interviewed so far.
6. The objective is to have the highest probability of selecting the best applicant of the whole group. *This is equivalent to maximizing the expected payoff, defined as* 1 *for the best applicant and* 0 *otherwise (i.e. choosing non-best candidate has* 0 *payoff).*
7. If the decision can be deferred to the end, this can be solved by the simple maximum selection algorithm of tracking the maximum and selecting the overall maximum at the end. The difficulty is that the decision must be made immediately.

- The secretary problem turns out to have an elegant solution, with an strikingly simple optimal stopping rule using the so-called *Look-and-then-Leap* approach.
- Specifically, we are going to choose a threshold number r, and interview the first $r-1$ applicants with no intention of making them an offer. *This phase is for calibration (learning from the data).*
- Starting from the rth applicant, we select the very first one whose rank is better than all the applicants we saw in the Looking phase.
- For an arbitrary cutoff r, the probability that the best applicant is selected is

$$
\begin{aligned}
P(r) &= \sum_{i=1}^{n} P((i \text{ is selected}) \cap (i \text{ is best applicant})) \\
&= \sum_{i=1}^{n} P(i \text{ is selected } \mid i \text{ is best applicant}) \times P(i \text{ is best}) \\
&= \sum_{i=1}^{r-1} 0 + \frac{1}{n} \times \left[\sum_{i=r}^{n} P(\text{best of first } i-1 \text{ in first } r-1 \mid i \text{ is best applicant}) \right] \\
&= \frac{1}{n} \times \sum_{i=r}^{n} \frac{r-1}{i-1} \\
&= \frac{r-1}{n} \times \sum_{i=r}^{n} \frac{1}{i-1} \\
&= \frac{r-1}{n} \times \ln \frac{n}{r-1}
\end{aligned}
$$

- The sum is not defined for $r=1$, but in that case it is easy to see that $P(1) = 1/n$.

- Explanation: when we do fail to hire the best secretary? This can only happen if between rth and $(i-1)$st applicants, we meet someone who is better than the top candidate among the first $r-1$.

- Thus, in order to avoid this failure scenario, we just have to ensure that the *best candidate among the first $i-1$ comes before rth applicant.* This is precisely what the summation accounts for!

- Probability $P(r)$ is maximized for $r-1 = \frac{n}{e}$, for which we get $P(r) = \frac{1}{e}$.

  1. Take derivative of the function $f = \frac{r}{n} \ln \frac{n}{r} = \frac{r}{n} \ln n - \frac{r}{n} \ln r$.
  2. $\frac{df}{dr} = \frac{\ln n}{n} - \frac{\ln r}{n} - \frac{1}{n}$.
  3. Since second derivative is negative, we find the best r by setting $df/dr = 0$.
  4. We get $\ln r = \ln n - 1$, which holds true for $r = n/e$.

- That is, in the looking phase, we pass over $1/e$ fraction of the candidates, and then choose the first one that is better than any candidate seen in the looking phase.

- In an interesting twist, the payoff for this stopping rule is $1/e$.

- The proof of optimal strategy is by Karlin and Robbins. It is known that all other strategies are dominated by a strategy of the form *"reject the first p unconditionally, then accept the next candidate who is better"*.

- **Homework Problem 1:** Think about why the optimal strategy must have look-then-leap format.

2 Other Variants of the Secretary Problem

2.1 Full Information Secretary Problem

- The classical version assumes as little as possible about the candidates. But there are many settings where each candidate's quality can be measured on a numerical scale: score on an exam such as SAT, GRE or LSAT.

- Then, a candidate with 75th percentile score is better than 3/4 of the pool (assuming the pool is a fair representation of the population at large).

- We can therefore consider a "full information" version of the secretary problem, where each candidate has a score, say, between 0 and 100, which we learn after interviewing the candidate.

- The problem is still complicated: if the first candidate is 95th percentile, there is still the possibility that other candidates with higher scores are in the pool.
- However, since we now have a numerical estimate of how likely that is, we can factor that into our decision making. The prob. that the next candidate will be in 96th or higher percentile is only 1/20.
- Full information means that we no longer need to look before leaping. Instead, we can use a *Threshold Rule* (which depends on how many candidates are left): we immediately accept a candidate if she is above a set percentile.
- For intuition, it's best to reason backwards. If we are at the last candidate, we always accept. But when considering the candidate $n-1$, the threshold is 50%. If she is above 50th percentile, we accept, otherwise roll the dice on the last candidate.
- Similarly, we will choose the third-to-last, if she is above 69th percentile.
- What is the prob. of winning (hiring the best secretary)? The answer is 58%.
- **Homework Problem 2:** Try to derive this win probability.

2.2 Pretty Good Secretary

- The single-minded focus on hiring just the best secretary may seem like an overkill, although in some financial decisions, not getting the best price is always regretful.
- Suppose instead of optimizing the probability of finding the best, we want to optimize the quality (rank) of the secretary hired: *the best secretary has rank* 1*, while the worst has rank* n*.*
- If we apply the previous algorithm, with the obvious modification that we reach the end and have not hired, we hire the last person, then
  the expected rank is $\Omega(n)$*, which is pretty bad!*
  This is because with probability $1/e$ we reach the end without hiring anyone.
- Surprisingly, there is a different algorithm that picks a person with averaged rank $O(1)$. This algorithm is rather complicated, and is based on computing a series of time steps $t_0 \leq t_1 \leq \cdots t_k$, and then proceed as follows: we reject the first t_0 applicants, then if we find someone in the first t_1 that is better than all previous, we hire. Otherwise, between (t_1+1)th and t_2 applicant, we are willing to hire someone who is either the best or the second best of those seen so far. And so on.
- Basically, with time we get increasingly more desperate.
- **Homework Problem 3:** Read or think about it.

2.3 Time Discounting

- Let us now consider a variant that puts us closer to the *real estate* realm.
- Imagine we are selling a house. You hire a realtor who will advertise and market the house. After you have made some additional investments into the house (painting, re-flooring, new appliances, landscaping), the house is put on the market.
- Buyers arrive one at a time, and for each offer you typically have to decide instantly whether to accept the offer, or reject it. (Rejected offers disappear.)
- So far, this is just like the secretary problem in that you want a strategy that maximizes the prob. of getting the highest $ offer, but with one small difference.
- Each day, or week, or month, the house does not sell, costs money (e.g. mortgage payment)! Therefore, the offer of $\$D$ today is much better than the same offer a month later.
- In addition, the house selling is similar to full information game, because the objective value of the offers is in dollars, which not only tells us which one is better, but also by how much.
- In house selling, our goal isn't just to get the max value offer, but rather to maximize

$$\text{profit} \;=\; \text{offer} \;-\; \text{cost}$$

- Read the book for more details.
- **Homework Problem 4.** Construct a simple, natural model of the carrying cost, and analyze it.

2.4 Secretary Problem with Offer Refusal

- A Secretary Problem with Uncertain Employment, M. H. Smith, J. of Applied Prob. 1975. Considers a model where each applicant can refuse an offer with a fixed prob. irrespective of its rank.
- The Secretary Problem and Its Extensions: A Review, by P. R. Freeman, in International Statistical Review, Aug. 1983.

2.5 Card Turning

- The secretary problem has been used in card tricks, where it also leads to a surprising and unexpected puzzle outcome.
- First a story. John Elton was an enlisted man in the US Air Force during Vietnam war, and later a math professor at Georgia Tech.
  During his time in Vietnam, he used the secretary problem for a moneymaking game in barracks.
- Elton asked his fellow airman to write down 100 different numbers, positive or negative, as large or as small as they wished, on 100 slips of paper, turn them face down on a table and mix them up.
- He would bet that he could turn the slips of paper over one at a time, and *stop with the higest number.*
- He convinced them that the chance of him winning was *obviously so small* they should pay him $10 if he won, and he will pay $1 each time he lost.
- There was no shortage of bettors, and even though Elton lost nearly 2/3 of the time, with 10 to 1 odds, he raked in a bundle by winning just 1/3 of the time.
- First note that there is a very simple strategy for winning 1/4 of the time, which already gives him a huge advantage in this game.
  1. Call an observed number a "record" if it is the highest number seen so far.
  2. Suppose you turn over *half* the cards, without stopping, and then stop with the first record you see.
  3. This has prob. 1/4 of success. Why? Because it succeeds whenever the largest number is in the second half, and the second largest in the first half. By randomness, each of these (independent) events occurs 50% of the time, for the combined prob. of $\frac{1}{2} \times \frac{1}{2}$.
- But using the secretary algorithm—turn up $100/e$ cards without stopping, and then stopping with the next record—gives Elton the $1/e \approx 0.36$ prob. of winning.

2.6 A Surprise at $N = 2$

- Suppose you must decide whether to stop and choose between two slips of paper.
- You turn over one, observe a number, and now must decide whether it is larger or smaller than the hidden number on the second.

- Common sense suggests this should be a 50-50 game: the larger number is equally likely to be either first or second, so one cannot do better than to toss a coin. This coin toss gives you 50% chance of winning.
- Rather suprisingly, David Blackwell of UC Berkeley figured out a way to win this game more than half the time!
- Most mathematicians also don't believe it upon hearing it the first time. (Similar to the Monty Hall 3-doors problem.)
- The solution works as follows.
    1. Generate a random number R, using a standard Gaussian or any other device. (Drawing from any distribution over the real line works.)
    2. Turn over first slip, and observe its number X.
    3. If $R > X$, then turn over the second card. Otherwise, stop with X.
- How can such a simple-minded strategy guarantee a win more than half the time?
- Consider the real number line, and place the two number under the slips N_1, N_2 on it. This partitions the line into three parts.
- Suppose p is the prob. that $R < N_1$, and q the prob. that $R > N_2$. This leaves prob. $1 - (p + q)$ of R falling between N_1 and N_2.
- If $R < N_1, N_2$, then you win 1/2 the time, depending on which number is on slip 1.
- Similarly, if $R > N_1, N_2$, you win 1/2 the time.
- However, if R is between N_1 and N_2, then you *always win*, with prob. $1 - (p + q)$.
- Since $N_1 \neq N_2$ are distinct integers, the prob. that R falls in between the two is strictly non-zero!
- If the writer is forced to choose integers in a fixed range, such as $[1, 100]$, he cannot make this prob. too small.

3 Prophet's Inequality and Optimal Stopping Rule

3.1 A Gambler's Stopping Game

- You are presented with a sequence of n treasure chests.
- Each chest contains a cash prize, but the chests are locked and you cannot see their contents. (Notationally, v_i denotes the prize of chest i.)
- However, each chest has a distribution over non-negative values printed on it, and you are told that the value of the prize in each chest was drawn independently from its distribution. (Notationally, F_i denotes the distribution of chest i.)
- For instance, each chest may be an urn, containing balls of n different colors. Each color represents a different $ prize value. The mixture of balls population represents the distribution F_i. We imagine that a ball from this urn has been chosen at random, but not yet disclosed to us.
- The host of the game will open the chests for you, one at a time. When a chest is opened, you can see the prize and must make a choice to either accept, terminating the game immediately, or reject, in which case the prize is lost but the game proceeds to the next chest.
- How should you play this game to maximize your expected winnings?
- How is this problem different from Secretary's problem?
  1. In Secretary problem, the order of arrival is assumed to be *random*, meaning all $n!$ orders are equally likely, while in prophet's inequality game, no assumption about the order is needed; as a result, the sequence can be *adversarial*.
  2. In the Secretary problem, where prize information is purely ordinal (rank ordering) and goal is to maximize the *probability* of selecting the max, in the prophet's game the prizes have real values, drawn from known distributions, and and the goal is to maximize the prize collected.
- **Prophet Inequality and Posted Prices.** In recent research, Prophet's inequality has been used to show that optimal but very complex and unimplmentable selling mechanisms can be replaced by very simple posted-price mechanisms with only a small loss in gain.

3.2 Optimal Strategy

One can compute the **optimal** strategy with a backward induction:

- The payoff on day n is $\mathbf{E}(v_n)$.
- On day $n-1$, the gambler stops if payoff v_{n-1} exceeds $\mathbf{E}(v_n)$, etc.

But this optimal strategy has several drawbacks:

- Complicated, with n different unrelated thresholds for each day.
- Does not explain the nature and properties of the game.
- Non-robust to changes in the order of the distributions.

3.3 A Threshold Strategy and Prophet Inequality

- Let us ask the following question: how does the expected value of the (aforementioned) optimal strategy compare with the expected maximum prize?
- In other words, how well does an optimal game contestant perform relative to an omniscient prophet, who can see inside the chests and therefore trivially win the best prize every time?
- (Note that the omniscient prophet's winning is also a random variable depending on the random draws within each chest.)
- We will show that (surprisingly?) that the gambler's expected winning is always at least 1/2 of the prophet's winning! (The ratio of 2 is also the best possible.)
- The strategy is both simple and non-adaptive: **We simply pick a fixed target value t, and accept the first prize $v_i \geq t$.**
- Specifically, we will choose the *median* of the max prize as our threshold, namely, t such that $P[\max_i v_i \geq t] = 1/2$.
- This strategy is clearly sub-optimal, because there is a chance we will accept nothing, and reject even the last day's payoff. But we can prove the following theorem.

Prophet Inequality: *If we choose t such that $\Pr[\textit{no prize}] = 1/2$, then*

$$\mathbf{E}[\text{prize for strategy } t] \;\geq\; \frac{1}{2}\mathbf{E}[\max_i v_i]$$

3.4 Proof of the Prophet Inequality.

The proof consists of 3 steps.

1. Derive an upper bound U on $\mathbf{E}[\max_i v_i]$.

2. Derive an lower bound L on $\mathbf{E}$[prize], with strategy t.

3. Show that $L \geq U/2$ for the choice of $x = 1/2$, where

$$x \;=\; \Pr[\text{rejects all prizes}] \;=\; \Pi q_i, \;\text{where } q_i = \Pr[v_i < t] \;\; (i.e.\text{chest } i \text{ rejected})$$

3.5 Upper Bound for Max

- The notation $x^+ = \max\{0, x\}$ discards the negative value cases.

- Clearly,

$$\mathbf{E}[\max_i v_i] \;\leq\; t + \mathbf{E}[\max_i (v_i - t)^+],$$

  where the first term is an upper bound for cases with payoff $\leq t$, and the second term accounts for *excess* for the cases with payoff $> t$. (The initial t amount in those cases is already accounted for by the first term.)

- The expected max is upper bounded by the expected sum, so we can write this as

$$\mathbf{E}[\max_i v_i] \;\leq\; t + \sum_i \mathbf{E}[(v_i - t)^+]$$

3.6 Lower Bound for Gambler's Payoff

- $\mathbf{E}[\text{prize}] \;\geq\; (1-x)t \;+\; \sum_i \mathbf{E}[(v_i - t)^+ | v_j < t, \;\text{for } j < i] \times \Pr[v_j < t, \;\text{for } j < i]$

- The first term accounts for the event that the gambler meets his threshold during the game, which occurs with prob. $(1 - x)$. The second term accounts for the payoff in excess of t.

- Observe that $\Pr[v_j < t, \;\text{for } j < i] \;=\; \Pi_{j<i}\, q_j \;\geq\; x$, by definition.

- Thus,

$$\mathbf{E}[\text{prize}] \;\geq\; (1-x)t \;+\; x \sum_i \mathbf{E}[(v_i - t)^+ |\text{ other } v_j < t]$$

- Because the conditioned variable is independent of the conditioning event, we get

$$\mathbf{E}[\text{prize}] \;\geq\; (1-x)t \;+\; x \sum_i \mathbf{E}[(v_i - t)^+]$$

- When $x = 1/2$, we get

$$\mathbf{E}[\text{prize}] \ \geq \ \frac{1}{2}\left(t \ + \ \sum_i \mathbf{E}[(v_i - t)^+]\right) \ \geq \ \frac{1}{2}\mathbf{E}[\max_i v_i]$$

- This shows that gambler's payoff is at least 1/2 of the Prophet's payoff.

3.7 Optimality of Approximation

- The following example shows that the approximation ratio of 2 is best possible.
- Suppose there are only two chests. The first deterministically contains a prize of 1.
- The second chest contains a prize of $1/\varepsilon$ with prob. ε, for an arbitrarily small $\varepsilon \in (0, 1]$, and otherwise contains nothing.
- What is the expected value of the maximum prize? The max prize is $1/\varepsilon$ with prob. ε, and with the remaining $(1 - \varepsilon)$ prob. the max prize is 1 (first chest). Thus, the expected value of the maximum prize is $\varepsilon\frac{1}{\varepsilon} + (1 - \varepsilon)1 \ = \ 2 - \varepsilon$.
- On the other hand, no strategy can guarantee an expected value more than 1. The first chest always contains a prize of 1, and so the contestant can either take it (and receive prize of 1) or leave it, in which case he must take the prize of second chest whose expected value is also 1.
- Therefore, no policy with approximation factor better than $2 - \varepsilon$ exists, for any $\varepsilon > 0$.

4 Burglar Problem: When to Quit

- Boris Berezovsky was a Russian billionaire, and one of the richest man in Russia. He used trading in an environment of hyperinflation, and became one of the new class of oligarchs. (Before that he was living on a mathematician's salary from the USSR Academy of Sciences.)
- After becoming rich, he became critical of the Russian president. One day Putin was asked about Berezovsky's criticisms, and replied:

> The state has a cudgel in its hands that you use to hit just once, but on the head. We haven't used this cudgel yet.... The day we get really angry, we won't hesitate.

- Berezovsky left Russia permanently the next month, taking up exile in England, where he continued to criticize Putin's regime.
- How did Berezovsky decide it was time to leave Russia? Is there a way, perhaps, to think mathematically about the advice to *quit while you're ahead*?
- Berezovsky might actually have considered this very question himself, since the topic he had worked on all those years ago as a mathematician was none other than optimal stopping. He even authored the first (and, so far, the only) book entirely devoted to the secretary problem.

4.1 The Burglar's Problem

- The problem of quitting while you are ahead has been analyzed under several different guises, but perhaps the most appropriate to Berezovsky's case is known as the *burglar problem.*

    1. A burglar has the opportunity to carry out a sequence of robberies. Each robbery provides some reward, and there's a chance of getting away with it each time.
    2. Each successful burglary increases the burglar's loot.
    3. But if the burglar is caught, he gets arrested and loses all his accumulated gains.
    4. What algorithm should he follow to maximize his expected take?

- The results are pretty intuitive: the number of robberies you should carry out is roughly equal to the chance you get away, divided by the chance you get caught.
- If you're a skilled burglar and have a 90% chance of pulling off each robbery (and a 10% chance of losing it all), then retire after $90/10 = 9$ robberies.
- **A mathematical model.**

    Consider a burglar each of whose attempted burglaries is successful with prob. p. If successful, the amount of loot earned is j with prob. p_j, for $j = 0, 1, 2, \ldots, m$. (That is, each loot is a random variable with distribution p_j.) If unsuccessful, the burglar is caught and loses everything accumulated so far, and the game ends. What's burglar's optimal policy?

- **Solution.**

    1. Suppose the current loot is i. If the burglar decides to stop, his reward is i, and the game ends.

2. On the other hand, if he continues, then if successful his revised loot will be $i + j$, with prob. p_j.
3. Thus, if we let $V(i)$ be the burglar's expected maximum loot in the current state, then the optimality condition is

$$V(i) \quad = \quad \max \left\{ i, \; p \sum_j p_j V(i+j) \right\}$$

4. Under the one-stage look-ahead policy, the burglar should stop in state i if $i \geq p \sum_j p_j (i+j)$.
5. That is, if $\mu = \sum_j j p_j$ denotes the expected return from a successful burglary, then the stopping condition is $i \;\geq\; p(i+\mu)$, which gives

$$i \;\geq\; \frac{p\mu}{1-p}$$

- That is, a ham-fisted amateur with a 50/50 chance of success? The first time you have nothing to lose, but don't push your luck more than once.

4.2 Lessons and Generalizations

- Despite his expertise in optimal stopping, Berezovsky's story ends sadly. He died in March 2013, found by a bodyguard in the locked bathroom of his house in Berkshire with a ligature around his neck. The official conclusion of a postmortem examination was that he had committed suicide, hanging himself after losing much of his wealth through a series of high-profile legal cases involving his enemies in Russia.

- Perhaps he should have stopped sooner—amassing just a few tens of millions of dollars, say, and not getting into politics.

- Surprisingly, not giving up *ever* also makes an appearance in the optimal stopping literature. It might not seem like it from the wide range of problems we have discussed, but there are sequential decision-making problems for which there is no optimal stopping rule.

- A simple example is the game of **"triple or nothing."** Imagine you have \$1, and can play the following game as many times as you want: *bet all your money, and have a 50% chance of receiving triple the amount and a 50% chance of losing your entire stake. How many times should you play?*

- There is no optimal stopping rule for this problem since each time you play your average gains are a little higher. Starting with \$1, you will get \$3 half the time and \$0 half the time, so on average you expect to end the first round with \$1.50 in your pocket. Then, if you were lucky in the first round, the two possibilities from the \$3 you've just won are \$9 and \$0, for an average return of \$4.50 from the second bet, and so on.

- The math shows that you should always keep playing. But if you follow this strategy, you will eventually lose everything. Some problems are better avoided than solved.