Odds Algorithm

An Online Algorithm

Group Fibonado

20. Dec 2016

Outline

1. Introduction
   - Online Algorithm
   - The Secretary Problem
2. Optimal Stopping
3. Odds Algorithm
   - Algorithm
   - Proof

Online Algorithm

- Page replacement algorithm (LRU, Marking algorithm)
- Insertion sort
- Perceptron
- Odds algorithm

The Secretary Problem on dog planet

Description Interview n candidates for a position one at a time. After each interview decide if the candidate is the best so far and hire him/her.

Goal Maximize the probability of choosing the best among all n candidates.

The Secretary Problem on dog planet

Description Interview n candidates for a position one at a time. After each interview decide if the candidate is the best so far and hire him/her.

Goal Maximize the probability of choosing the best among all n candidates.

Giant Schnauzer

Description Interview n candidates for a position one at a time. After each interview decide if the candidate is the best so far and hire him/her.

Goal Maximize the probability of choosing the best among all n candidates.

Giant Schnauzer

Fair

The Secretary Problem on dog planet

Description Interview n candidates for a position one at a time. After each interview decide if the candidate is the best so far and hire him/her.

Goal Maximize the probability of choosing the best among all n candidates.

Giant Schnauzer

Fair

English Springer Spaniel

The Secretary Problem on dog planet

Description Interview n candidates for a position one at a time. After each interview decide if the candidate is the best so far and hire him/her.

Goal Maximize the probability of choosing the best among all n candidates.

Giant Schnauzer

Fair

English Springer Spaniel

Nice

Husky

Description Interview n candidates for a position one at a time. After each interview decide if the candidate is the best so far and hire him/her.

Goal Maximize the probability of choosing the best among all n candidates.

Giant Schnauzer

Fair

English Springer Spaniel

Nice

Husky

No way

Description Interview n candidates for a position one at a time. After each interview decide if the candidate is the best so far and hire him/her.

Goal Maximize the probability of choosing the best among all n candidates.

Giant Schnauzer

Fair

English Springer Spaniel

Nice

Husky

No way

Miniature Schnauzer

The Secretary Problem on dog planet

Description Interview n candidates for a position one at a time. After each interview decide if the candidate is the best so far and hire him/her.

Goal Maximize the probability of choosing the best among all n candidates.

Giant Schnauzer

Fair

English Springer Spaniel

Nice

Husky

No way

Miniature Schnauzer

Great

Description Interview n candidates for a position one at a time. After each interview decide if the candidate is the best so far and hire him/her.

Goal Maximize the probability of choosing the best among all n candidates.

Giant Schnauzer

Fair

English Springer Spaniel

Nice

Husky

No way

Miniature Schnauzer

Great

Border collie

The Secretary Problem on dog planet

Description Interview n candidates for a position one at a time. After each interview decide if the candidate is the best so far and hire him/her.

Goal Maximize the probability of choosing the best among all n candidates.

Giant Schnauzer

Fair

English Springer Spaniel

Nice

Husky

No way

Miniature Schnauzer

Great

Border collie

...

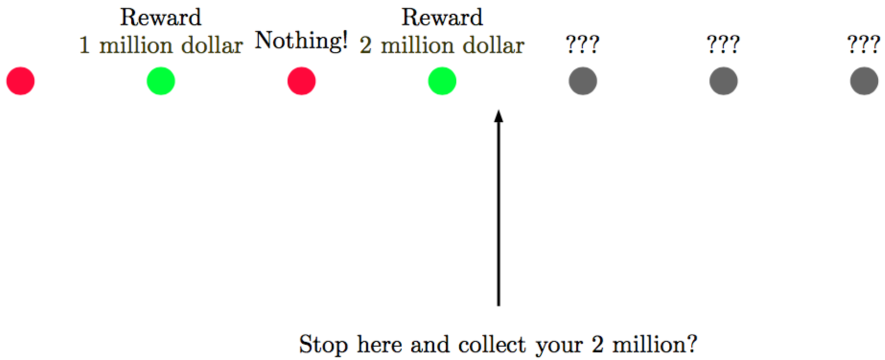
Outline

1 Introduction
- Online Algorithm
- The Secretary Problem

2 Optimal Stopping

3 Odds Algorithm
- Algorithm
- Proof

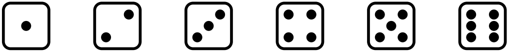
Optimal Stopping (Discrete time case)



The problem concerns with:

- When to stop??
- How to maximize the reward?

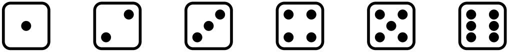
Dice Toss



Dice Toss

Consider a game that consists of throwing a fair, six-sided die n times, and whose aim is to stop at the last 6 obtained.

Dice Toss



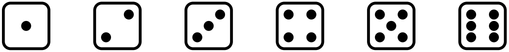
Dice Toss

Consider a game that consists of throwing a fair, six-sided die n times, and whose aim is to stop at the last 6 obtained.

After each toss, you can choose either stop or continue.

Reward. 0 if you didn't stop on the last ⚅.
Otherwise, you get $1 million$

Dice Toss



Dice Toss

Consider a game that consists of throwing a fair, six-sided die n times, and whose aim is to stop at the last 6 obtained.

After each toss, you can choose either stop or continue.

Reward. 0 if you didn't stop on the last ⚅.
Otherwise, you get 1 *million*

Let's play!

Dice Toss

Dice Toss

Dice Toss

⚄ ⚁

Dice Toss

Dice Toss

⚄ ⚁ ⚅ ⚃

Dice Toss

⚄ ⚁ ⚅ ⚃ ⚃

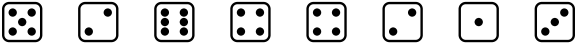
Dice Toss

⚄ ⚁ ⚅ ⚃ ⚃ ⚁

Dice Toss

⚄ ⚁ ⚅ ⚃ ⚃ ⚁ ⚀

Dice Toss



Dice Toss

⚄ ⚁ ⚅ ⚃ ⚃ ⚁ ⚀ ⚂ ⚅

Dice Toss

⚄ ⚁ ⚅ ⚃ ⚃ ⚁ ⚀ ⚂ ⚅ ⚀

Dice Toss

⚄ ⚁ ⚅ ⚃ ⚃ ⚁ ⚀ ⚂ ⚅ ⚀

Dice Toss

⚄ ⚁ ⚅ ⚃ ⚃ ⚁ ⚀ ⚂ ⚅ ⚀

How to maximize the probability that we stop at the last ⚅?

$$\mathbb{P}(\text{Obtaining } \textit{one} \text{ ⚅ in last } \ell \text{ throws }) = C_\ell^1 \frac{1}{6} \cdot (\frac{5}{6})^{\ell-1} = \frac{\ell}{6} \cdot (\frac{5}{6})^{\ell-1}$$

$$\mathbb{P}(\text{Obtaining } \textit{one} \text{ ⚅ in last } \ell \text{ throws }) = C_\ell^1 \frac{1}{6} \cdot (\frac{5}{6})^{\ell-1} = \frac{\ell}{6} \cdot (\frac{5}{6})^{\ell-1}$$

Differentiating this expression and setting it to 0, we find it is maximized at $\ell = 6(\textit{or } \ell = 5)$.

$$\mathbb{P}(\text{Obtaining } \textit{one} \text{ ⚅ in last } \ell \text{ throws }) = C_\ell^1 \frac{1}{6} \cdot (\frac{5}{6})^{\ell-1} = \frac{\ell}{6} \cdot (\frac{5}{6})^{\ell-1}$$

Differentiating this expression and setting it to 0, we find it is maximized at $\ell = 6(or\ \ell = 5)$.

Intuitively, the most sensible strategy therefore seems to wait till we only have 6 throws left, and then choose the first ⚅ that occurs after that.

$$\mathbb{P}(\text{Obtaining } \textit{one} \text{ ⚅ in last } \ell \text{ throws }) = C_\ell^1 \frac{1}{6} \cdot (\frac{5}{6})^{\ell-1} = \frac{\ell}{6} \cdot (\frac{5}{6})^{\ell-1}$$

Differentiating this expression and setting it to 0, we find it is maximized at $\ell = 6(\textit{or } \ell = 5)$.

Intuitively, the most sensible strategy therefore seems to wait till we only have 6 throws left, and then choose the first ⚅ that occurs after that.

$$\begin{aligned}\mathbb{P}(\text{Strategy leads to the last⚅}) &= \mathbb{P}(\text{Obtaining } \textit{one} \text{ ⚅ in last 6 throws }) \\ &= (\frac{5}{6})^5 = 0.4018\end{aligned}$$

$$\mathbb{P}(\text{Obtaining } \textit{one} \text{ ⚅ in last } \ell \text{ throws }) = C_\ell^1 \frac{1}{6} \cdot (\frac{5}{6})^{\ell-1} = \frac{\ell}{6} \cdot (\frac{5}{6})^{\ell-1}$$

Differentiating this expression and setting it to 0, we find it is maximized at $\ell = 6(\textit{or } \ell = 5)$.
Intuitively, the most sensible strategy therefore seems to wait till we only have 6 throws left, and then choose the first ⚅ that occurs after that.

$$\begin{aligned}\mathbb{P}(\text{Strategy leads to the last⚅}) &= \mathbb{P}(\text{Obtaining } \textit{one} \text{ ⚅ in last 6 throws }) \\ &= (\frac{5}{6})^5 = 0.4018\end{aligned}$$

What about general cases? The Secretary Problem?

Outline

1. Introduction
   - Online Algorithm
   - The Secretary Problem
2. Optimal Stopping
3. Odds Algorithm
   - Algorithm
   - Proof

Problem Restatement

Problem

You are observing a sequence of events, which may be a success or not, you are required to make the decision to stop or continue the observation.

Goal

Stop on the last success.

Theorem

Every kind of stop strategy can be reduced to the following rule:

ignore all successes before the k^{th} observation, then choose the first success encountered.

Alice and Bob go biking

Alice and Bob want to go biking someday in this week, but the later, the better.

Algorithm

Homework 99

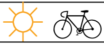
Alice and Bob go biking

	Mon	Tue	Wed	Thu	Fri	Sat	Sun
$\mathbb{P}$(☼)	0.1	0.5	0.3	0.1	0.3	0.2	0.1

Alice and Bob go biking

	Mon	Tue	Wed	Thu	Fri	Sat	Sun
$\mathbb{P}$(☼)	0.1	0.5	0.3	0.1	0.3	0.2	0.1
Outcome 1	☼	☼	☼ 🚲	////	////	☼	////

Alice and Bob go biking

	Mon	Tue	Wed	Thu	Fri	Sat	Sun
$\mathbb{P}($$)$	0.1	0.5	0.3	0.1	0.3	0.2	0.1
Outcome 1							
Outcome 2							

Alice and Bob go biking

	Mon	Tue	Wed	Thu	Fri	Sat	Sun
$\mathbb{P}$(☀)	0.1	0.5	0.3	0.1	0.3	0.2	0.1
Outcome 1	☀	☀	☀ 🚲	////	////	☀	////
Outcome 2	////	☀	////	☀	////	☀ 🚲	////
Outcome 3	☀	☀	☀	////	////	☀	////

Odds Algorithm

Let $I_1, \ldots, I_n$ be a sequence of independent indicators, s.t.

$$I_k = \begin{cases} 1 & \text{if ☼} \\ 0 & \text{otherwise ( )} \end{cases}$$

$$p_k = \mathbb{E}(I_k) = 1 - q_k = \mathbb{P}(\text{day } k \text{ is ☼})$$

$$r_k = p_k / q_k \text{ (the odds)}$$

Odds Algorithm

	Mon	Tue	Wed	Thu	Fri	Sat	Sun
p_k	0.1	0.5	0.3	0.1	0.3	0.2	0.1

Odds Algorithm

	Mon	Tue	Wed	Thu	Fri	Sat	Sun
p_k	0.1	0.5	0.3	0.1	0.3	0.2	0.1
r_k	1/9	1	3/7	1/9	3/7	1/4	1/9

Odds Algorithm

	Mon	Tue	Wed	Thu	Fri	Sat	Sun
p_k	0.1	0.5	0.3	0.1	0.3	0.2	0.1
r_k	1/9	1	3/7	1/9	3/7	1/4	1/9
$\sum_{j=k}^{n} r_j$	2.44	2.33	1.33	0.90	0.79	0.36	0.11

Odds Algorithm

	Mon	Tue	Wed	Thu	Fri	Sat	Sun
p_k	0.1	0.5	0.3	0.1	0.3	0.2	0.1
r_k	1/9	1	3/7	1/9	3/7	1/4	1/9
$\Sigma_{j=k}^{n} r_j$	2.44	2.33	1.33	0.90	0.79	0.36	0.11

↑ starts from this day, choose the first ☼ day.

Optimal Rule of the Odds Algorithm

Ignore all successes before k^{th} observation, then stop at the first success. Where k is the largest k s.t. $\Sigma_{j=k}^{n} r_j \geq 1$.

Proof of the Odds Theorem

Let $S_k = I_k + \cdots + I_n$.

Observation

$S_k = 1 \iff$ exactly 1 ☼ day after the $k - 1^{th}$ day.

Claim

let k^* be optimal rule, k^* maximize $\mathbb{P}(S_k = 1)$.

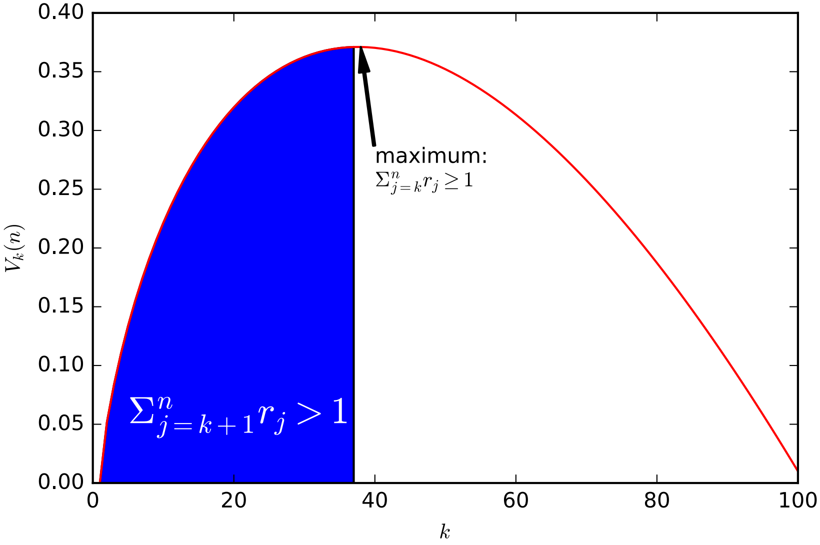
Actually, we can find the precise formula for $\mathbb{P}(S_k = 1)$ using basic probability theory:

$$\mathbb{P}(S_k = 1) = (\Sigma_{j=k}^{n} r_j)(\prod_{j=k}^{n} q_j) = V_k(n)$$

$$V_k(n) \text{ increasing in } k \Leftrightarrow V_k(n) < V_{k+1}(n) \Leftrightarrow^* \Sigma_{j=k+1}^{n} r_j > 1$$

Observation

1. $\Sigma_{j=k+1}^{n} r_j$ is monotonically decreasing as k increases.
2. $\mathbb{P}(S_k = 1)$ increases up to a certain value of k and then decrease
3. There is therefore a single maximum attained at the largest k for which $\Sigma_{j=\mathbf{k}}^{n} r_j \geq 1$.


0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
0
20
40
60
80
100
k
$V_k(n)$
maximum:
$\Sigma_{j=k}^{n} r_j \geq 1$
$\Sigma_{j=k+1}^{n} r_j > 1$

Secretary Problem Revisited

Let $I_1, \ldots, I_n$ be indicators s.t. $I_k = 1$ if secretary k is the best secretary seen so far and $I_k = 0$ otherwise. Clearly,

$$p_k = 1/k, r_k = \frac{1}{k-1}$$

$$R_s = 1/(n-1) + 1/(n-2) + \cdots + 1/(s-1), \text{stopped at 1.}$$

As $n \to \infty$,

$$s/n \to 1/e \approx 37\%$$

$$V(n) = ((s-1)/n)R_s \to 1/e.$$

Secretary Problem Revisited

Let $I_1, \dots, I_n$ be indicators s.t. $I_k = 1$ if secretary k is the best secretary seen so far and $I_k = 0$ otherwise. Clearly,

$$p_k = 1/k, r_k = \frac{1}{k-1}$$

$$R_s = 1/(n-1) + 1/(n-2) + \cdots + 1/(s-1), \text{stopped at } 1.$$

As $n \to \infty$,

$$s/n \to 1/e \approx 37\%$$

$$V(n) = ((s-1)/n)R_s \to 1/e.$$

37% Rule

reject the first n/e secretaries and to then accept the best secretary so far, if any.

Q&A

Thanks.