Q2

William HU ZIHAO

2024-09-24

Q2 Secretary Problem

1) make_choice

```r
make_choice <- function(N, split_number) {
  input_list <- sample(1:N, N, replace = FALSE)

  eval_group <- input_list[1:split_number]
  select_group <- input_list[(split_number+1):N]

  best_in_eval <- max(eval_group)

  for (candidate in select_group) {
    if (candidate >= best_in_eval) {
      return(candidate)  # Return the first match
    }
  }

  return(tail(select_group, 1))
}
```

```r
simulate_make_choice <- function(N, split_number, num_simulations = 1000) {
  success_count <- 0

  for (i in 1:num_simulations) {
    selected_candidate <- make_choice(N, split_number)

    if (selected_candidate == N) {
      success_count <- success_count + 1
    }
  }

  prob_success <- success_count / num_simulations
  return(prob_success)
}
```

```r
simulate_make_choice(100,10)
```

```
## [1] 0.26
```

```r
simulate_make_choice(100,30)
```

```
## [1] 0.368
```

```
simulate_make_choice(100,50)
```

```
## [1] 0.381
```

```
simulate_make_choice(100,70)
```

```
## [1] 0.238
```

2) find_optimal

```
find_optimal <- function(N, num_simulations = 1000) {
  optimal_prob <- 0
  optimal_split <- 0

  for (split_number in 1:(N/2)) {
    success_count <- 0

    for (i in 1:num_simulations) {
      chosen_candidate <- make_choice(N, split_number)

      if (chosen_candidate == N) {
        success_count <- success_count + 1
      }
    }

    prob <- success_count / num_simulations

    if (prob > optimal_prob) {
      optimal_prob <- prob
      optimal_split <- split_number
    }
  }

  return(list(optimal_split = optimal_split, optimal_prob = optimal_prob))
}
```

3) simulation of make_choice and find_optimal

```
set.seed(789)
result_3 <- find_optimal(N = 3)
print(result_3)
```

a. N = 3

```
## $optimal_split
## [1] 1
##
## $optimal_prob
## [1] 0.51
```

```
set.seed(789)
result_10 <- find_optimal(N = 10)
print(result_10)
```

b. N = 10

```
## $optimal_split
## [1] 4
##
## $optimal_prob
## [1] 0.408
```

```r
set.seed(789)
result_100 <- find_optimal(N = 100)
print(result_100)
```

c. N = 100

```
## $optimal_split
## [1] 39
##
## $optimal_prob
## [1] 0.391
```

```r
set.seed(789)
result_500 <- find_optimal(N = 500)
print(result_500)
```

d. N = 500

```
## $optimal_split
## [1] 168
##
## $optimal_prob
## [1] 0.404
```

```r
set.seed(789)
result_1000 <- find_optimal(N = 1000)
print(result_1000)
```

e. N = 1000

```
## $optimal_split
## [1] 430
##
## $optimal_prob
## [1] 0.41
```

In conclusion, the results are consistent with theoretical expectations. As N increases, the optimal split approaches $\frac{N}{e}$ (approximately 37% of N candidates), and the probability of selecting the best candidate drifts around $\frac{1}{e}$ (approximately 37%).

While the simulation results did not strictly converge to theoretical results, they are very close and align with the theoretical behavior.