Examen 1er Parcial Robert P. Dobrow

Ojeda Contreras Braulio Melquisedec

2022-11-25

Se extraen cartas de una baraja estándar, con reemplazo, hasta que aparece un as. Simule la media y la varianza del número de cartas extraídas.

Definimos la función que simula agarrar un as de una baraja estándar (52 cartas).

```
tomar_as_baraja_est <- function() {
    # Consideraremos la siguiente notación
    # p: picas
    # d: diamantes
    # c: corazones
    # s: espadas
  # Y además que
  # a: as
  # j: joto
  # q: reina
  # k: rey
  baraja <- c("a_p", "2_p", "3_p", "4_p", "5_p", "6_p", "7_p", "8_p", "9_p", "10_p",
              "j_p", "q_p", "k_p",
              "a_d", "2_d", "3_d", "4_d", "5_d", "6_d", "7_d", "8_d", "9_d", "10_d",
              "j_d", "q_d", "k_d",
              "a_c", "2_c", "3_c", "4_c", "5_c", "6_c", "7_c", "8_c", "9_c", "10_c",
              "j_c", "q_c", "k_c",
              "a_s", "2_s", "3_s", "4_s", "5_s", "6_s", "7_s", "8_s", "9_s", "10_s",
              "j_s", "q_s", "k_s")
  total_cartas <- length(baraja)
  cartas_tomadas <- character()
  as <- FALSE
  i <- 1

  while (as == FALSE) {
    cartas_tomadas[i] <- sample(baraja, 1, prob = rep(c(1 / total_cartas), total_cartas))
    if (substr(cartas_tomadas[i], 1, 1) == "a")
      as <- TRUE
    else
      i <- i + 1
  }

  return(i)
}
```

Ahora, una vez hemos definido nuestra función, procedemos a ejecutarla para 1 millón de intentos y con ello calcular la media y la varianza de nuestras simulaciones.

```r
intentos <- 1000000
res <- replicate(intentos, tomar_as_baraja_est())
media_cartas_requeridas_as <- mean(res)
varianza_cartas_requeridas_as <- var(res)

cat("Media =", media_cartas_requeridas_as, "\tVarianza =", varianza_cartas_requeridas_as)
```

```
## Media = 12.98567     Varianza = 155.8182
```

En las aplicaciones de seguridad inform´atica, un honeypot es una trampa colocada en una red para detectar y contrarrestar a los piratas informáticos. Los datos de los honeypots se estudian en Kimou et al. (2010) utilizando cadenas de Markov. Los autores obtienen los datos de los honeypots de una base de datos central y observan los ataques contra cuatro puertos inform´aticos, 80, 135 139 y 445 durante un año. Los puertos son los estados de una cadena de Markov, junto con un estado que corresponde a que ningún puerto es atacado. Se observan los datos semanales, y se registra el puerto más atacado durante la semana. La matriz de transición de Markov estimada para los ataques semanales es:

```r
mat_puertos_atacados <- matrix(c(0, 0, 0, 0, 1,
                          0, 8/13, 3/13, 1/13, 1/13,
                          1/16, 3/16, 3/8, 1/4, 1/8,
                          0, 1/11, 4/11, 5/11, 1/11,
                          0, 1/8, 1/2, 1/8, 1/4), ncol = 5, byrow = TRUE)

alpha = c(0, 0, 0, 0, 1)
```

(a) Puertos menos y más probablemente atacados al cabo de 2 semanas

```r
n <- 2
mat_puertos_atacados_2 = mat_puertos_atacados %*% mat_puertos_atacados
dist_2 = as.vector(alpha %*% mat_puertos_atacados_2)
labels <- c('Port 80', 'Port 135', 'Port 139', 'Port 445', 'No attack')
res <- data.frame(labels, dist_2)
print(res)
```

```
##      labels    dist_2
## 1   Port 80 0.0312500
## 2  Port 135 0.2132867
## 3  Port 139 0.3868007
## 4  Port 445 0.2226836
## 5 No attack 0.1459790
```

Así que después de dos semanas, los puertos más y menos probablemente atacados son el puerto 139 y el 80 respectivamente.

(b) Distribución a largo plazo

Obtendremos la matriz de n pasos considerando un error de 10^{-8}. De tal modo, que si todas las diferencias absolutas entre cada entrada de la matriz de n-1 pasos y la de n pasos son menores que el error, pararemos en la multiplicación de matrices. Luego obtendremos la distribución en el largo plazo al multiplicar la distribución inicial por la matriz de n pasos obtenida.

```
error <- 10 ** -8
i <- 1
A_n_minus_1 <- mat_puertos_atacados
number_entries <- dim(mat_puertos_atacados)[1] * dim(mat_puertos_atacados)[2]
flag <- TRUE
while (flag) {
  A_n <- A_n_minus_1 %*% mat_puertos_atacados
  aux_mat <- A_n - A_n_minus_1
  aux_count <- length(aux_mat[abs(aux_mat) < error])
  if (aux_count == number_entries) {
    flag <- FALSE
  }
  else {
    i <- i + 1
    A_n_minus_1 <- A_n
  }
}
cat("En", i, "iteraciones, la matriz de transición converge a la matriz:\n")
```

```
## En 25 iteraciones, la matriz de transición converge a la matriz:
```

```
print(A_n)
```

```
##            [,1]      [,2]      [,3]      [,4]   [,5]
## [1,] 0.02146667 0.2669333 0.3434667 0.2273333 0.1408
## [2,] 0.02146667 0.2669333 0.3434667 0.2273333 0.1408
## [3,] 0.02146667 0.2669333 0.3434667 0.2273333 0.1408
## [4,] 0.02146667 0.2669333 0.3434667 0.2273333 0.1408
## [5,] 0.02146667 0.2669333 0.3434667 0.2273333 0.1408
```

```
dist_n = as.vector(alpha %*% A_n)
labels <- c('Port 80', 'Port 135', 'Port 139', 'Port 445', 'No attack')
res <- data.frame(labels, dist_n)
print(res)
```

```
##      labels     dist_n
## 1   Port 80 0.02146667
## 2  Port 135 0.26693333
## 3  Port 139 0.34346667
## 4  Port 445 0.22733333
## 5 No attack 0.14080000
```