Universidad de Buenos Aires
Facultad de Ingeniería
Año 2025 - 1er cuatrimestre

Procesos Estocásticos (86.09)

Trabajo Práctico N°1, Aplicaciones del método de Montecarlo

ESTUDIANTES:

Leroy, Joaquin jleroy@fi.uba.ar	110452
Cabrera, Santiago smcabrera@fi.uba.ar	110445
Ruggiero, Valentina Mora vruggiero@fi.uba.ar	109317

Índice

1. Introducción

En este trabajo práctico se aplicó el método de montecarlo al cálculo de probabilidades en el juego de Black Jack. Específicamente, se programó una función que simulara una ronda jugada por la banca recibiendo como parámetro una carta con la que iniciar. Esta función fue llamada muchas veces para estimar la probabilidad de que la banca se exceda dada una carta inicial, así como la probabilidad total de que la banca se exceda.

2. Ejercicios

2.1. Requerimientos y funciones auxiliares

Para llevar a cabo la simulación, se implementó la función `crear_mazo(carta)` que construye un mazo completo de 52 cartas, combinando los trece valores posibles con sus respectivos cuatro palos. Dicho mazo es un array de strings, por ejemplo: `Mazo = ['2','2','2','2','3','3','3','3', ..., 'A','A','A','A']`. La función recibe como parámetro la primera carta entregada a la banca, la cual es removida del mazo antes de proceder a mezclarlo, utilizando para ello la función `shuffle` de la biblioteca `random`.

Otra función auxiliar utilizada es `valor_carta(carta)`, encargada de asignar y devolver el valor numérico correspondiente a la carta recibida como parámetro. En particular, si la carta es un As, se optó por asignarle inicialmente un valor de 11, con la posibilidad de reducirlo a 1 a posteriori, en caso de que el valor total de la mano exceda 21.

2.2. Ejercicio 1

Para explicar como se implementó la función `ronda_banca(carta_inicial)` se utiliza el siguiente pseudocódigo:

Algorithm 1 Simulación de una ronda de la banca

```
function RONDA_BANCA(carta_inicial)
    mazo ← CREAR_MAZO(carta_inicial)              ▷ Crea el mazo excluyendo la carta inicial
    puntos ← VALOR_CARTA(carta_inicial)           ▷ Inicializa los puntos con la carta inicial
    ases_once ← 0                                 ▷ Contador de ases que valen 11
    while True do
        carta ← MAZO.POP()                        ▷ Extrae una carta del mazo
        if carta = 'A' then
            ases_once ← ases_once +1              ▷ Incrementa el contador de ases que valen 11
        end if
        puntos ← puntos + VALOR_CARTA(carta)      ▷ Suma el valor de la carta extraída
        if puntos > 21 and ases_once > 0 then
            ases_once ← ases_once −1
            puntos ← puntos −10                   ▷ Convierte un as que valía 11 a 1 si así conviniese
        end if
        if puntos > 21 then
            return True                           ▷ La banca se pasa
        else if puntos ≥ 17 then
            return False                          ▷ La banca se planta
        end if
    end while
end function
```

Nótese que la función propuesta en el algoritmo 1 devuelve un booleano: `True` si la banca se pasó y `False` si la banca se plantó. Además, se utiliza la funcion `.pop()`, lo que garantiza que se extraigan cartas del mazo sin reposición. El segundo algoritmo propuesto, `montecarlo_banca(M, carta_inicial)`, funciona de la siguiente manera:

Algorithm 2 Estimación por Monte Carlo de la probabilidad de que la banca se pase

```
function MONTECARLO_BANCA(M, carta_inicial)
    pasados ← 0                                  ▷ Contador de veces que la banca se pasa
    for i ← 1 to M do
        resultado ← RONDA_BANCA(carta_inicial)   ▷ Simula una ronda de la banca
        if resultado then
            pasados ← pasados +1                 ▷ Incrementa el contador si la banca se pasó en esa ronda
        end if
    end for
    return pasados / M                           ▷ Calcula la probabilidad de que la banca se pase
end function
```

La probabilidad de que la banca se pase sabiendo la primer carta tirada es fundamentalmente $\frac{\#\text{veces que se pasó}}{\#\text{realizaciones}}$, por lo que es lógico que a mayor cantidad de realizaciones, más fiel el valor de probabilidad arrojado.

2.3. Ejercicio 2

En este ejercicio se asumió que la carta inicial era un 6 y se simuló la probabilidad de que la banca se excediera variando el número de simulaciones M en potencias de 10 entre 1 y 10^6. Así, se obtuvo un gráfico de la probabiliad de excederse en función de la cantidad de realizaciones, $P_E(M)$. El gráfico obtenido fue el siguiente:

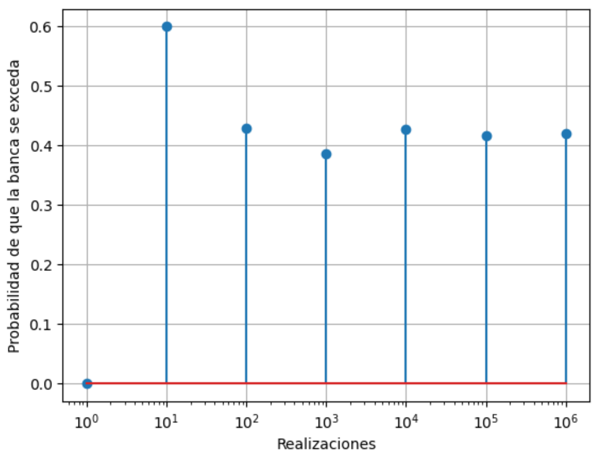

Figura 1: $P_E(M)$

Se observa que en un principio la probabilidad fluctua y que eventualmente converge a un valor algo superior a 0.4

2.4. Ejercicio 3

En este ejercicio se realizaron 10^5 simulaciones para cada valor posible de la carta inicial de la banca y luego se graficó la probabilidad de que la banca se exceda para cada una. El gráfico resultante es el siguiente:

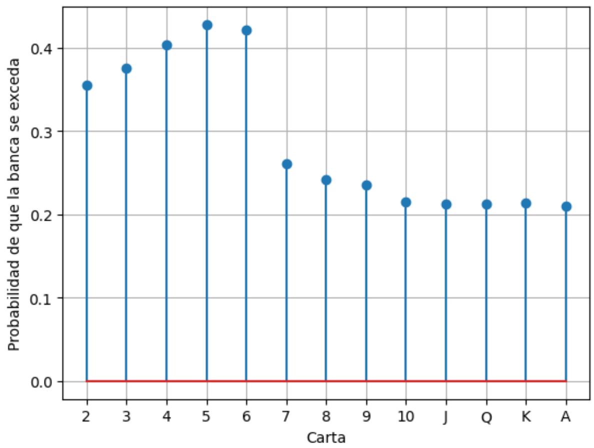

Figura 2: $P_E(C)$

Se observa de este gráfico una caída de probabilidad de que la banca se exceda cuando la carta inicial supera el 6. Esto se debe a que para todos los valores de carta menores o iguales a 6, la banca tendrá que robar tres cartas como mínimo para superar o igualar los 17 puntos (con la excepción del caso borde de un As y un 6 en la misma ronda).

Luego, con los valores obtenidos de las simulaciones, se calculó la probabilidad total de que la banca exceda los 21 puntos para cualquier valor posible de la carta inicial. Para esto, se utilizó la ley de la probabilidad total.

$P_E(X > 21) = P_E(X > 21|C = As).P_C(As) + P_E(X > 21|C = 2).P_C(2) + ... + P_E(X > 21|C = K).P_C(K)$

A su vez, como el mazo es un espacio muestral equiprobable, se plantea lo siguiente:

$P_E(X > 21) = P_E(X > 21|C = As) \cdot \frac{1}{13} + P_E(X > 21|C = 2) \cdot \frac{1}{13} + ... + P_E(X > 21|C = K) \cdot \frac{1}{13}$

Finalmente, tomando los valores obtenidos de la simulación obtenemos que la probabilidad de que la banca se exceda es:

$P_E(X > 21) = (0{,}355 + 0{,}376 + 0{,}404 + 0{,}428 + 0{,}422 + 0{,}261 + 0{,}241 + 0{,}236 + 0{,}216 + 0{,}213 + 0{,}212 + 0{,}214 + 0{,}211) \cdot \frac{1}{13} = 0{,}291$

Si bien este resultado es una aproximación de la probabilidad auténtica de que la banca se exceda, gracias al gran número de simulaciones que se realizaron se puede considerar este resultado como equivalente.

3. Conclusiones

Del ejercicio 2 se concluye que a medida que la cantidad de casos simulados aumenta, la probabilidad estimada se aproxima a la verdadera.

Del ejercicio 3 se concluye que la carta inicial que maximiza la probabilidad de que la banca se exceda es el 5. También se destaca que la probabilidad de que la banca se exceda es 0,29.

Por último, se concluye que el método de Montecarlo permite de manera eficaz estimar probabilidades que serían muy difíciles de obtener de forma analítica. En este caso, permitió obtener una función de probabilidad empírica para el exceso de la banca según la primer carta recibida, cuya forma no se asemeja a ninguna distribución analítica conocida, dado que presenta una caída irregular que responde más al comportamiento del juego que a un modelo matemático clásico.