

# Trabajo práctico 1

## Aplicaciones del método Monte Carlo

### 1. Introducción

#### 1.1. Objetivo

El Black Jack es uno de los juegos de cartas más populares en los casinos, donde los jugadores deben sacar cartas hasta alcanzar una suma de 21 puntos sin excederla. El objetivo del trabajo práctico es implementar un programa que calcule la probabilidad de que la banca se exceda, es decir, que se pase de los 21 puntos, teniendo como dato la primer carta con la que inicia la banca. Sin embargo, debido a las diferentes reglas del juego, se torna muy complejo el cálculo de dicha probabilidad de forma analítica. Por esta razón, se propone utilizar el método Monte Carlo para simular un gran número de realizaciones que permitan estimar la probabilidad de que la banca pierda la ronda en función su primer carta.



#### 1.2. Reglas del Juego

##### ■ Valor de las cartas:

- El mazo de Black Jack posee 52 cartas. Cada palo ( $\heartsuit$ ,  $\spadesuit$ ,  $\diamondsuit$  y  $\clubsuit$ ) consta de 13 cartas: un **As**, las numeradas del 2 al 10 y las figuras **J**, **Q** y **K**.
- El valor de las cartas numéricas (2 a 10) es su valor nominal.
- El valor de las figuras (**J**, **Q**, **K**) es de 10 puntos cada una.
- El **As** puede valer 1 u 11, según lo que sea más beneficioso. El **As** se cuenta como 11 siempre que su total de puntos no supere 21, de lo contrario, se cuenta como 1.

##### ■ Reglas para la Banca:

- La banca siempre debe pedir cartas hasta que su total sea **17 puntos o más**. Si la banca alcanza o supera los 17 puntos acumulados, no puede pedir más cartas.
- Si la banca supera 21 puntos, pierde la ronda, ya que se “pasa” del puntaje máximo permitido.

## 2. Ejercicios

### Requerimientos para las simulaciones

1. Suponer que se utiliza un único mazo estándar de 52 cartas.
2. Suponer que en cada nueva ronda, las cartas se mezclan aleatoriamente con una distribución uniforme. Sugerencia: en MATLAB puede utilizar `randperm()` para obtener índices permutados. En Python puede usar el método `.shuffle()` del paquete `random` o `numpy` para permutar los elementos.
3. Suponer que cada vez que la banca saca una nueva carta, no hay reposición en el mazo, hasta que se vuelva a barajar en una nueva ronda.

### Ejercicio 1

- (a) **Simulación de una ronda:** Implementar una función `ronda_banca()` que simule una ronda jugada por la banca. La función debe recibir como dato la primer carta con la que inicia la banca y retornar una variable que indique si se pasó o no, de acuerdo a las reglas mencionadas.
- (b) **Simulación de múltiples rondas:** Definir la función `montecarlo_banca()` que debe recibir dos parámetros: el número  $M$  de realizaciones (cantidad de rondas a simular) y el valor de la primer carta con la que iniciaría la banca. La función debe retornar la probabilidad de que la banca se pase dado el valor de la carta inicial. Describa en palabras (puede ser un pseudocódigo) el algoritmo utilizado para estimar la probabilidad pedida.

¿?  
No debería devolver la carta sacada por la banca?

La idea sería esta: la función `ronda_banca()` recibe una carta y simula el juego entero de la banca. Indica si, dada una carta inicial, la banca llega a un puntaje entre 17 y 21 o si pierde.

### Ejercicio 2

Para una dada carta inicial, suponer un 6, simular la probabilidad de que la banca se exceda ( $P_E$ ) en función de distinta cantidad de realizaciones:  $M = \{1, 10, 10^2, 10^3, 10^4, 10^5, 10^6\}$ . Haga un gráfico de la probabilidad de excederse en función del número de realizaciones,  $P_E(M)$  (con el eje  $M$  logarítmico).

$$P(\text{de q' la banca se pase}) \approx \frac{\text{# veces q' se pasó}}{M}$$

### Ejercicio 3

- (a) Dejando fijo el número de realizaciones en  $M = 10^5$ , simular y graficar la probabilidad de que la banca se exceda en función de la carta inicial,  $P_E(C)$ . Sugerencia: puede utilizar un valor entero que identifique cada carta, por ejemplo  $C = 1, 2, \dots, 13$  (ya que son 13 tipos distintos de cartas).
- (b) Con las probabilidades estimadas para cada tipo de carta, calcule la probabilidad total de que la banca exceda los 21 puntos para cualquier valor posible de la carta inicial. Ayuda: considere la probabilidad de ocurrencia de cada carta y la ley de probabilidad total.

$$P(B > 21) = P(B > 21 | C=1) \cdot P(C=1) + P(B > 21 | C=2) \cdot P(C=2) + \dots \rightarrow \text{¿?} \cdot \text{¿?}$$

## 3. Conclusiones

Como conclusiones, elabore un resumen breve y conciso comentando características que considere relevantes del método propuesto en este trabajo y los resultados obtenidos, así como dificultades encontradas (si fuera el caso) y cómo fueron abordadas.

#### 4. Normas y material entregable

- **Informe:** El informe debe entregarse en formato PDF (**no se aceptarán otros formatos**) y con nombre: `TP1_GXX.PDF` (donde **XX** es el número de grupo). El código fuente (de MATLAB, OCTAVE o PYTHON) debe presentarse en un archivo aparte.
- **Código:** Los archivos de código fuente desarrollados deben estar en formato `.m` (si se usa MATLAB/OCTAVE) o formato `.py` (si usa PYTHON). El código debe incluirse junto al informe en un archivo ZIP (con mismo nombre que el informe) que deberá subirse al campus.

¡APLICAR AL TRUCCO?