Simulare Monte Carlo - Blackjack

1. Ce este algoritmul Monte Carlo?

Algoritmul Monte Carlo este o metodă de simulare stocastică bazată pe simulări aleatorii repetate, utilizată pentru a estima soluții pentru probleme complexe. În cazul jocului Blackjack, simulările determină probabilitatea ca jucătorul să câștige atunci când alege să tragă o carte ("hit") sau să stea ("stand"), în funcție de următorii parametri:

- Scorul jucătorului înainte de a lua o decizie.
- Cărțile vizibile ale dealer-ului (numite "upcard").
- Acțiunea jucătorului ("hit" sau "stand").

2. Cum funcționează algoritmul Monte Carlo în acest caz?

a. Inițializarea simulărilor

Sunt definite structuri de date care vor stoca numărul de câștiguri și numărul total de simulări pentru fiecare combinație posibilă de:

- Sumă a cărților jucătorului: între 2 și 21.
- Upcard-ul dealer-ului: între 1 și 10.
- Acțiunea: "hit" sau "stand".

b. Simularea unui joc de Blackjack (simulate_blackjack)

Implementare: https://github.com/dariadragomir/Blackjack Monte Carlo Simulation/blob/mai n/blackjack Monte Carlo Simulation/blob/mai n/blackjack Monte Carlo Simulation/blob/mai

Această funcție modelează un singur joc:

1. Generarea mâinilor inițiale:

- Se selectează două cărți pentru jucător și două pentru dealer dintr-un pachet
- Se calculează scorurile inițiale pentru jucător și dealer.

2. Determinarea acțiunii jucătorului:

 Jucătorul decide aleatoriu dacă face "hit" (mai trage o carte) sau "stand" (se oprește).

3. Rezolvarea jocului:

- Dacă jucătorul face "hit", trage o carte şi verifică dacă a depăşit 21 sau a ajuns la 21.
- Dealer-ul joacă conform regulilor standard (trebuie să continue să tragă până ajunge la cel puțin 17).

4. Determinarea câștigătorului:

• Dacă jucătorul depășește 21, pierde automat.

- o Dacă dealer-ul depășește 21, jucătorul câștigă.
- În alte cazuri, câștigătorul este cel care are scorul mai mare.

c. Simularea pe scară largă (monte_carlo_blackjack)

Această funcție rulează simulările de Blackjack de un număr foarte mare de ori (N_SIMULATIONS = 10**8):

1. Actualizarea statisticilor:

 După fiecare joc, rezultatul este înregistrat în probabilites și number_simulations, în funcție de scorul inițial, upcard-ul dealer-ului și actiunea luată.

2. Calcularea probabilităților:

 Pentru fiecare combinație posibilă de parametri, probabilitatea este calculată ca raportul dintre numărul de câștiguri și numărul total de simulări:
 P=Numărul de câștiguri/Numărul total de simulări

De exemplu:

```
Player's Score: 14
```

Dealer Upcard: 1 | Hit Probability: 16.1% | Stand Probability: 8.3% Dealer Upcard: 2 | Hit Probability: 19.5% | Stand Probability: 11.9%

Metoda Monte Carlo utilizează un principiu de bază din teoria probabilităților și statistica inferențială: dacă avem o serie de observații independente ale unei variabile aleatoare, atunci media acestor observații converge în mod probabilistic către valoarea așteptată a variabilei aleatoare (2. Teorema Limită Centrală).

1. Fundamentul teoretic al metodei Monte Carlo

Presupunem că vrem să aproximăm probabilitatea unui eveniment P(A). Aceasta este valoarea teoretică pe care încercăm să o estimăm. Prin simulare, generăm un eșantion aleator de N observații, fiecare având aceeași probabilitate P(A) de succes. Fie variabila aleatoare Xi, care ia valoarea 1 dacă evenimentul A apare și 0 altfel. Observațiile sunt **independente și identic distribuite (i.i.d.)**.

Estimăm P(A) folosind media eșantionului:

$$\hat{P}(A) = rac{1}{N} \sum_{i=1}^N X_i.$$

1. Valoarea așteptată:

$$\mathbb{E}[\hat{P}(A)] = P(A).$$

$$\operatorname{Var}(\hat{P}(A)) = rac{\operatorname{Var}(X_i)}{N} = rac{P(A)(1-P(A))}{N}.$$

2. Varianța:

Varianța scade invers proporțional cu N, ceea ce înseamnă că estimarea devine mai precisă pe măsură ce numărul de simulări crește.

2. Teorema Limită Centrală

Conform Teoremei Limita Centrala, pentru un număr mare de simulări (N mare), distribuția

estimatorului $\hat{P}(A)$ devine aproximativ normală, indiferent de distribuția lui Xi.

$$\hat{P}(A) \sim \mathcal{N}\left(P(A), rac{P(A)(1-P(A))}{N}
ight).$$

Adică:

4. Inegalitatea Cebâșev

O altă metodă de a cuantifica eroarea estimării folosește **Inegalitatea Cebâșev**, care oferă o limită pentru probabilitatea ca media eșantionului să se abată de la valoarea așteptată cu mai mult de o toleranță $\epsilon > 0$:

$$P(|\hat{P}(A) - P(A)| \geq \epsilon) \leq rac{ ext{Var}(\hat{P}(A))}{\epsilon^2}.$$

Aceasta înseamnă că probabilitatea ca estimarea să fie departe de valoarea reală P(A) scade proporțional cu creșterea numărului de simulări N. Cu alte cuvinte, creșterea N reduce incertitudinea în estimare.

5. Inegalitatea Chernoff-Hoeffding

$$P(|\hat{P}(A) - P(A)| \geq \epsilon) \leq 2 \exp\left(-2N\epsilon^2
ight).$$

Această inegalitate arată că probabilitatea unei deviații mari scade **exponențial** cu numărul de simulări N. Este mai precisă decât Cebâșev pentru variabile limitate, cum ar fi cazul lui Xi, care este binară (0 sau 1).

Inegalitatea Chernoff - Hoeffling:

Tie X₁,..., X_n noarabile aleatoare independente is identic

distribute, i.e. $a \le \#_i \le b$, $\forall i = 1, m$.

Atunci $\forall \in >0$, $P(\overline{S_n} - \mu) \ge E \le 2e^{-\frac{2n}{2n-2}}$.

In casul nostru, putem modela resultatul unei

maini de Blackjock după executarea unei arțiuni

(de "hit" sau "stand") folosind o noariabilă alsatoare

Bernoulli de probabilitate p, desarece resultatul poate fi

"aistig" sau "pierdere"

X ~ Bernoulli (p) = (0 1) 0 = "pierdere"

X ~ Bernoulli (p) = (n-pp) 1 = "aistig"
0 = % i ≤ 1

Determinăm numărul de simulări necesare pentru

a presice resultatul cu o marjà de eroare E = 0,05 cu

un nivel de încredere L = 0,98 (98%) pentru o anumită mână:

 $P(|S_{m}-\mu| < \varepsilon) \ge 1-2e^{-2m\varepsilon^{2}} \ge C$ $(=) e^{-2m\varepsilon^{2}} \le \frac{1-L}{2} (=) 2m\varepsilon^{2} \ge -lm(\frac{1-L}{2})$ $(=) m \ge \frac{1}{2\varepsilon^{2}} lm(\frac{2}{1-L}) (=) m \ge \frac{1}{2\cdot(0.05)^{2}} \cdot lm(\frac{2}{1-0.98})$ $(=) m \ge \frac{1}{2\cdot\frac{25}{10^{2}}} \cdot lm(\frac{2}{0.02}) (=) m \ge \frac{10^{4}}{5\cdot10} \cdot lm(10^{2}) (=) m \ge 921$

Sunt necesare cel putin 921 simulari pentru fierare mana, pentru $\varepsilon=0,05$ si $\mathcal{L}=0,98$.

Bibliografie:

https://ro.wikipedia.org/wiki/Legea_numerelor_mari http://math.etc.tuiasi.ro/rstrugariu/cursuri/SPD2018/c5.pdf https://en.wikipedia.org/wiki/Monte_Carlo_method

					חוג רוטג	apilities					
2 -	0.2	0.26	0.25	0.25	0.24	0.23	0.26	0.24	0.2	0.17	
m -	0.2	0.25	0.25	0.24	0.23	0.23	0.26	0.23	0.22	0.17	
4 -	0.16	0.23	0.2	0.21	0.2	0.2	0.21	0.2	0.18	0.14	
٠ - 2	0.13	0.19	0.18	0.17	0.17	0.16	0.17	0.16	0.14	0.1	
9 -	0.12	0.17	0.17	0.17	0.14	0.15	0.15	0.14	0.12	0.084	
- 1	0.12	0.17	0.17	0.16	0.15	0.14	0.16	0.14	0.12	0.085	
œ -	0.14	0.21	0.2	0.2	0.19	0.19	0.27	0.2	0.17	0.12	
E 6-	0.18	0.27	0.26	0.25	0.24	0.23	0.33	0.32	0.22	0.16	
s St 10 -	0.26	0.35	0.34	0.33	0.32	0.31	0.4	0.39	0.36	0.24	
Player's Sum 12 11 10 9	0.42	0.47	0.46	0.46	0.45	0.45	0.48	0.46	0.43	0.38	
- 12 ay	0.17	0.21	0.21	0.2	0.2	0.2	0.23	0.21	0.18	0.15	
т П	0.17	0.2	0.2	0.2	0.19	0.18	0.22	0.2	0.19	0.15	
14 -	0.16	0.2	0.19	0.19	0.18	0.17	0.2	0.21	0.19	0.15	
15.1	0.15	0.18	0.17	0.17	0.17	0.15	0.22	0.2	0.18	0.14	
- 19	0.15	0.18	0.17	0.17	0.15	0.17	0.22	0.21	0.18	0.15	
17.1	0.15	0.17	0.16	0.15	0.17	0.16	0.22	0.2	0.18	0.14	
- 18	0.13	0.15	0.13	0.15	0.15	0.14	0.18	0.18	0.16	0.13	
19]	0.1	0.12	0.11	0.13	0.13	0.12	0.16	0.16	0.13	0.11	
	i	2	3	4	Stand Pro	babilities	7	8	9	10	
- 2	0.079	0.12	0.11	0.11	0.1	Uncard 0.11	0.08	0.075	0.067	0.039	
m -	0.08	0.12	0.11	0.12	0.11	0.11	0.081	0.077	0.061	0.036	
4 -	0.082	0.12	0.12	0.12	0.11	0.11	0.081	0.073	0.066	0.035	
ري - دي	0.078	0.12	0.12	0.12	0.11	0.11	0.08	0.075	0.068	0.036	
9 -	0.082	0.12	0.12	0.12	0.11	0.1	0.079	0.075	0.067	0.035	
- -	0.08	0.12	0.12	0.12	0.1	0.1	0.079	0.076	0.066	0.034	
ω -	0.079	0.12	0.12	0.11	0.1	0.1	0.081	0.077	0.066	0.035	
ے ہ -	0.08	0.12	0.11	0.11	0.1	0.1	0.08	0.076	0.067	0.035	
n 01 -	0.08	0.12	0.11	0.11	0.11	0.1	0.081	0.077	0.068	0.035	
£ = -	0.63	0.72	0.72	0.72	0.71	0.71	0.71	0.71	0.7	0.67	
Player's Sum	0.084	0.12	0.12	0.12	0.11	0.11	0.082	0.078	0.067	0.036	
-	0.083	0.12	0.12	0.12	0.11	0.11	0.082	0.077	0.07	0.036	
4 -	0.083	0.12	0.12	0.12	0.11	0.11	0.08	0.081	0.069	0.036	
15	0.083	0.12	0.11	0.11	0.1	0.1	0.082	0.079	0.068	0.035	
16	0.083	0.12	0.11	0.11	0.099	0.11	0.083	0.08	0.07	0.036	
17	0.083	0.12	0.11	0.11	0.11	0.11	0.084	0.079	0.069	0.037	
- 18	0.19 0.29	0.26 0.39	0.24	0.25 0.37	0.23 0.35	0.23 0.35	0.46 0.69	0.22 0.34	0.19	0.16 0.24	
19	0.29	0.39	0.37	0.37	0.33	0.35	0.89	0.34	0.29	0.24	
			-	1						1	
	1	2	3	4	5	6	7	8	9	10	

Proiect realizat de : Dragomir Daria Nicoleta, grupa 252 Jilavu Izabela Maria, grupa 251 Soare Alex Antonio, grupa 251