Łańcuchy Markowa Rewolucja Monte Carlo

Kacper Wnęk

March 2023

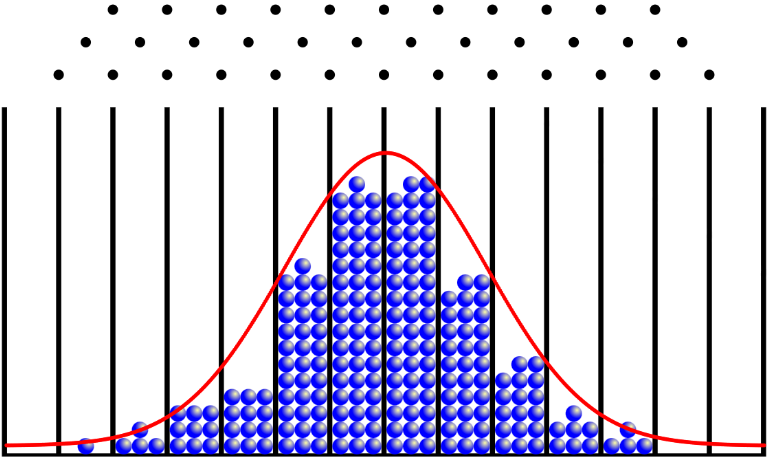
Plan Prezentacji

- Przypomnienie RP
- Łańcuchy Markowa
- Metoda Monte Carlo
- MCMC-Łańcuchy Markowa Monte Carlo

Koncepcja wartości oczekiwanej Bernouliego

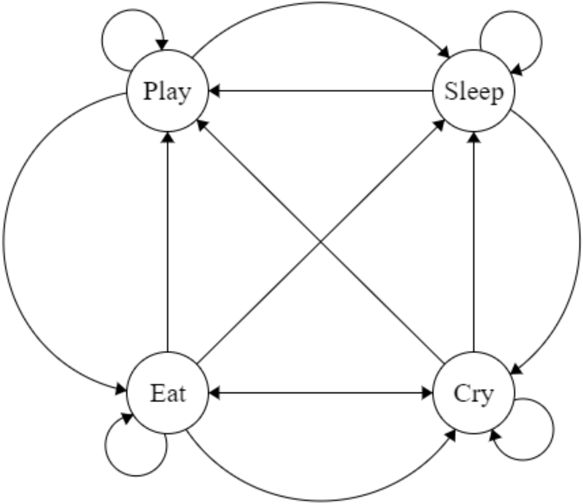
Wniosek: Jeśli obserwacje zdarzeń będą kontynuowane w nieskończoność to okaże się, że wszystkim na świecie rządzą precyzyjne stosunki i zmiany podglegające stałym prawom

Deska Galtona



Nekrasov vs Markov

Łańcuch Markowa



Łańcuch Markowa

Łańcuch Markowa to rodzaj procesu stochastycznego, w którym przyszłe stany zależą tylko od obecnego stanu i są niezależne od przeszłych stanów. Formalnie, łańcuch Markowa to ciąg zmiennych losowych $X_1, X_2, \ldots,$ z wartościami w skończonym zbiorze stanów $S = \{s_1, s_2, \ldots, s_n\}$, który spełnia warunek Markowa:

$$P(X_{n+1} = s_j \mid X_1 = s_{i_1}, X_2 = s_{i_2}, \ldots, X_n = s_{i_n}) = P(X_{n+1} = s_j \mid X_n = s_{i_n})$$

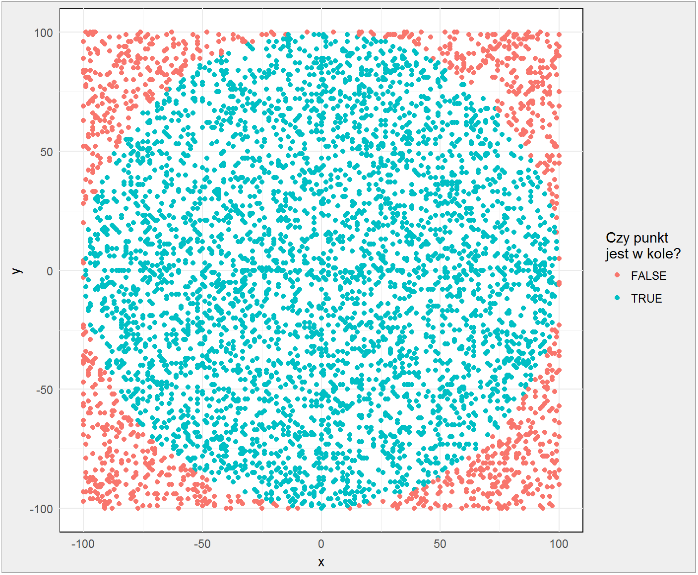
dla $1 \leq i_1, i_2, \ldots, i_n, j \leq n$.

Łańcuch Markowa

Łańcuch Markowa posiada własność memoryless, co oznacza, że zależność przyszłych stanów od obecnego stanu nie zależy od ilości przeszłych stanów, które przeszły się pomiędzy nimi. Innymi słowy, przyszłe stany zależą tylko od ostatniego stanu.
Łańcuch Markowa ma wiele zastosowań, w tym:

- Modelowanie procesów losowych, takich jak ruch uliczny, prognozowanie pogody, procesy ekonomiczne, itp.
- Analiza i modelowanie danych, takich jak sekwencje DNA, sekwencje językowe, itp.
- Zastosowania w teorii gier i sztucznej inteligencji, takie jak proces decyzyjny Markowa (MDP)

Metoda Monte Carlo



Łańcuchy Markowa Monte Carlo

W przypadku, gdy S jest przestrzenią skończoną stanów dyskretnych

$$S = \{x_1, x_2, \ldots, x_n\}$$

możemy prawdopodobieństwo przejścia sformułować w postaci macierzy losowej P:

$$\begin{bmatrix} T(x_1,x_1) & T(x_1,x_2) & ... & T(x_1,x_n) \\ T(x_2,x_1) & T(x_2,x_2) & ... & T(x_2,x_n) \\ ... & ... & ... & ... \\ T(x_n,x_1) & T(x_n,x_2) & ... & TT(x_n,x_n) \end{bmatrix}$$

warunek normalizacji

$$\sum_{j=1}^{n} T(x_i, x_j) = 1$$

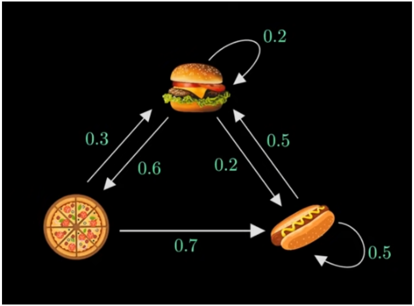
Łańcuchy Markowa Monte Carlo

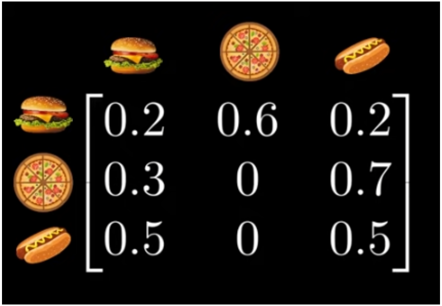
Własności macierzy losowych:

- suma elementów w wierszu jest równa 1 (warunek normalizacji)
- iloczyn macierzy losowych też jest macierzą losową
- przynajmniej jedna wartość własna macierzy równa jest 1

Łańcuchy Markowa Monte Carlo

Jak to działa?





Niech P będzie macierzą prawdopodobieństwa z poprzedniego slajdu

$$P = \begin{bmatrix} 0.2 & 0.6 & 0.2 \\ 0.3 & 0 & 0.7 \\ 0.5 & 0 & 0.5 \end{bmatrix}$$

To co chcemy obliczyć to rozkład, czyli prawdopodobieństwo każdego ze stanów, nazwijmy je π.Na poczatku wybierzmy dowolny ze stanów, od którego zaczniemy. Niech to będzie pizza. Oznaczmy zatem $\pi_0 = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}$

Łańcuchy Markowa Monte Carlo

Zobaczmy co sie stanie jeśli pomnożymy nasz wektor przez macierz prawdopodobieństwa

$$\pi_0 P = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0.2 & 0.6 & 0.2 \\ 0.3 & 0 & 0.7 \\ 0.5 & 0 & 0.5 \end{bmatrix} = \begin{bmatrix} 0.3 & 0 & 0.7 \end{bmatrix}$$

Otrzymaliśmy drugi wiersz macierzy, czyli inaczej mówiąc prawdopodobieństwo pozostałych stanów jeśli pierwszym stanem była pizza.Weźmy ten wynik i wstawmy go w miejsce π_0
i wykonajmy działanie ponownie

Łańcuchy Markowa Monte Carlo

$$\pi_1 P = \begin{bmatrix} 0.3 & 0 & 0.7 \end{bmatrix} \begin{bmatrix} 0.2 & 0.6 & 0.2 \\ 0.3 & 0 & 0.7 \\ 0.5 & 0 & 0.5 \end{bmatrix} = \begin{bmatrix} 0.41 & 0.18 & 0.41 \end{bmatrix}$$

$$\pi_2 P = \begin{bmatrix} 0.41 & 0.18 & 0.41 \end{bmatrix} \begin{bmatrix} 0.2 & 0.6 & 0.2 \\ 0.3 & 0 & 0.7 \\ 0.5 & 0 & 0.5 \end{bmatrix} = \begin{bmatrix} 0.34 & 0.25 & 0.41 \end{bmatrix}$$

Łańcuchy Markowa Monte Carlo

Wykonując takie działania wielokrotnie finalnie otrzymamy πP=π
Jeśli to równanie przypomina wam to wektor własny to się nie mylicie :)) jednak rozczarowując miłośników algebry liniowej nie będę się zagłębiał w szczegóły tego podobieństwa. Najbardziej istotny jest fakt, że dane π jest szukanym rozkładem.

$$\pi = \begin{bmatrix} 0.35211 & 0.21127 & 0.43662 \end{bmatrix}$$

Gdzie w tym wszystkim metoda Monte Carlo?
Łańcuchy Markowa działają świetnie gdy mówimy o niezbyt wielkich macierzach. Co jeśli nasza macierz jest rozmiarów 10!x10!? Naturalnie nie będziemy takich macierzy liczyć, więc tu z pomocą przychodzi metoda Monte Carlo. Zamiast całej macierzy weźmiemy tylko próbkę.

Łańcuchy Markowa Monte Carlo

Ideą Markov Chain Monte Carlo (MCMC) jest wykorzystanie Łańcuchów Markowa do próbkowania zgodnie z niejednolitymi rozkładami. Metoda ta, choć obliczeniowo ekspansywna w niższych wymiarach, jest niezwykle wydajna w wyższych wymiarach, ponieważ jej stopień zbieżności jest niezależny od wymiaru przestrzeni.[2]

Łańcuchy Markowa Monte Carlo w zastosowaniu

Fantastycznym przykładem działania MCMC i wykorzystania w kryptografi jest ten czerpany z pracy studentów Stanfordu Marc Coram oraz Phil Beineke.[1]
W jednym z więzień udało się przechwycić zaszyfrowaną wiadomość

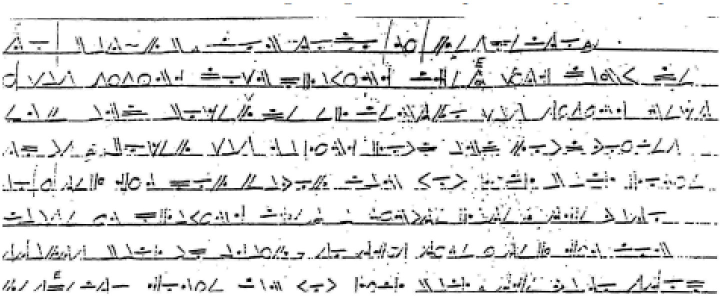


Figure: Obrazek pochodzi z [1]

Studenci szybko ustalili, że szyfr polegał na zwykłym zastąpieniu każdej litery pewnym symbolem, zatem istniał jakiś nieznany schemat, nazwijmy go funkcją, która działała w następujący sposób:

$$f : \{\text{ zaszyfrowana wiadomość}\} \to \{\text{zwykły alfabet}\}$$

Łańcuchy Markowa Monte Carlo w zastosowaniu

W celu złamania kodu studenci wykorzystali standardowy text i utworzyli macierz prawdopodobienstwa każdego z symboli oraz opracowali wzór wiarygodności funkcji

$$Pl(f) = \prod_i M(f(s_i), f(s_{i+1}))$$

Następnie algorytm działał w następujący sposób:

- Wybranie losowego startu, nazwijmy je f.
- Obliczanie Pl(f)
- Zmiana f na f_* dokonując losowej transpozycji 2 wartości przypisanych do f
- Obliczenie Pl(f_*) i jeśli jest większe od Pl(f) to zaakceptowanie f_*
- W przeciwnym wypadku "rzucenie monetą" o prawdopodobieństwie sukcesu Pl(f_*)/Pl(f) i jeśli wypadnie orzeł to zaakceptowanie f_*, jeśli nie to zostanie przy f

Łańcuchy Markowa Monte Carlo w zastosowaniu

Algorytm miał działać do momentu aż zwrócony tekst będzie miał jakiś sens. Okazało się, że już po około 2 tysiącach kroków rozszywrowana wiadomość jest zrozumiała i w nie ulega zmianie w wyniku dalszego działąnia algorytmu. Algorytm działa na tyle dobrze, ze można rozszyfrować wiadomość zaledwie ze skrawka tekstu a co dopiero z całej notatki jaką udało się przejąć w więzieniu.

to bat-rb. con todo mi respeto. i was sitting down playing chess with danny de emf and boxer de el centro was sitting next to us. boxer was making loud and loud voices so i tell him por favor can you kick back homie cause im playing chess a minute later the vato starts back up again so this time i tell him con respecto homie can you kick back. the vato stop for a minute and he starts up again so i tell him check this out shut the f**k up cause im tired of your voice and if you got a problem with it we can go to celda and handle it. i really felt disrespected thats why i told him. anyways after i tell him that the next thing I know that vato slashes me and leaves. dy the time i figure im hit i try to get away but the c.o. is walking in my direction and he gets me right dy a celda. so i go to the hole. when im in the hole my home boys hit doxer so now "b" is also in the hole. while im in the hole im getting schoold wrong and

Figure: Obrazek pochodzi z [1]

Łańcuchy Markowa Monte Carlo w zastosowaniu

Zastosowania symulacyjnych metod MCMC obejmują bardzo szeroki zakres dziedzin i aplikacji praktycznych.
Przykłady zastosowania MCMC

- Obliczanie całek wielowymiarowych
- Problemy typu "problem plecaka"
- Fizyka komputerowa
- Statystyka Bayesowska
- Biologia obliczeniowej
- Lingwistyka komputerowa

Dziękuję za uwagę

Bibliografia

Persi Diaconis.
The markov chain monte carlo revolution.
Bulletin of the American Mathematical Society, 46(2):179–205, 2009.

Matthew Richey.
The evolution of markov chain monte carlo methods.
The American Mathematical Monthly, 117(5):383–413, 2010.