

Wykład 9: Markov Chain Monte Carlo

Wykładowca: Andrzej Ruciński Pisarz: Ewelina Rychlińska i Wojciech Wawrzyniak

Wstęp

W tej części wykładu zajmiemy się zastosowaniami łańcuchów Markowa w informatyce. Naszym celem będzie bowiem przedstawienie na kilku przykładach tzw. metody MCMC (ang. Markov Chain Monte Carlo), czyli metody Monte Carlo z wykorzystaniem łańcuchów Markowa.

Ogólnie metoda Monte Carlo MC jest zaliczana do klasy metod symulacyjnych, gdyż jest stosowana do modelowania matematycznego procesów zbyt złożonych, aby można było przewidzieć ich wyniki za pomocą prostego modelu zastępczego, który można zaimplementować na komputerze, a następnie przeprowadzić symulacje tegoż modelu. Istotną rolę w metodzie MC odgrywa losowanie wielkości charakteryzujących proces, przy czym losowanie dokonywane jest zgodnie ze znanym nam rozkładem.

Metody MC powstały w odpowiedzi na postawione problemy przy projektowaniu eksperymentów fizycznych, w fizyce statystycznej czy chemii fizycznej, stąd są one bardzo rozpowszechnione w tych naukach.

Warto wspomnieć, że twórcami metody Monte Carlo była grupa wybitnych matematyków - Nicholas Metropolis, John von Neumann oraz Stanisław Ulam, pracujących w ośrodku badań jądrowych w Los Alamos w latach 40-tych XX wieku.

Idea MCMC

Niech Ω będzie skończoną przestrzenią probabilistyczną o rozbudowanej strukturze ze skomplikowanym rozkładem π . Metoda MCMC polega na konstrukcji nieredukowalnego nieokresowego łańcucha Markowa o danej przestrzeni stanów $S = \Omega$, macierzy przejść P o nieskomplikowanej budowie oraz rozkładzie stacjonarnym π . Celem bowiem będzie skorzystanie z twierdzenia ergodycznego, które gwarantuje, że obrany rozkład początkowy dla takiego łańcucha po wystarczająco długim czasie dąży do rozkładu stacjonarnego.

1 Przykłady

Przykład 1. *The hard-core model*

Niech dany będzie graf $G = (V, E)$, gdzie $|V| = k$ jest jego liczbą wierzchołków oraz $|E| = l$ liczbą krawędzi. Wierzchołki grafu są ponumerowane w sposób losowy wartościami 0 oraz 1. Oczywiście każde takie przyporządkowanie wartości 0, 1 jest funkcją, tutaj zwaną **konfiguracją**, jedną z $\{0, 1\}^V$ możliwych. **Konfiguracją dopuszczalną** nazywamy taką, w której żadne dwie „1” nie są połączone krawędzią (w teorii grafów konfiguracją dopuszczalną jest tzw. zbiór niezależny).

Niech Z_G będzie liczbą dopuszczalnych konfiguracji w grafie G .
Dla dowolnej konfiguracji $\xi \in \{0, 1\}^V$ bierzemy miarę probabilistyczną

$$\mu_G(\xi) = \begin{cases} \frac{1}{Z_G} & , \text{ jeżeli } \xi \text{ jest dopuszczalna} \\ 0 & , \text{ w przeciwnym przypadku} \end{cases}$$

tzn. losowanie dowolnej konfiguracji sprowadza się do wybrania każdej z dopuszczalnych z równym prawdopodobieństwem.

Naturalnym pytaniem, jakie się pojawia jest:

Jaka jest średnia liczba „1” w losowej konfiguracji wybranej zgodnie z rozkładem $\mu_G(\xi)$?

Niech $n(\xi)$ będzie liczbą „1” w ξ (inaczej wielkość zbioru niezależnego) oraz X losową konfiguracją wybraną zgodnie z $\mu_G(\xi)$. Wówczas średnia wielkość konfiguracji dopuszczalnej w G (średnia wielkość zbioru niezależnego) jest równa

$$E(n(X)) = \sum_{\xi \in \{0,1\}^V} n(\xi) \cdot \mu_G(\xi) = \frac{1}{Z_G} \sum_{\xi \in \{0,1\}^V} n(\xi) \cdot I_{\{\xi \text{ dopuszczalna}\}}$$

Zauważmy, że powyższa suma zawiera bardzo dużo składników, co znacznie zniechęca do jej obliczenia. A gdyby do postawionego problemu użyć symulacji? Pojawia się jednak problem:

Ile wykonać symulacji, aby otrzymać satysfakcjonujący wynik nieodbiegający zbyt od powyższego, oraz w jaki sposób wygenerować losowe ξ będące dopuszczalnym?

Pomysł polega na konstrukcji nieredukowalnego i nieokresowego łańcucha Markowa $X = (X_0, X_1, \dots)$, na przestrzeni stanów $S = \{\xi \in \{0, 1\}^V : \xi \text{ dopuszczalna}\}$, mającego $P_{\xi, \xi'} = \mathbb{P}(X_{n+1} = \xi' | X_n = \xi)$ oraz rozkład stacjonarny μ_G .

Łańcuch Markowa spełniający powyższe wymagania tworzymy na podstawie algorytmu:

1. Wylosuj $v \in V$ (z równym prawdopodobieństwem)
2. Rzuć symetryczną monetą
3. Jeśli „Orzeł oraz $\forall w \in N_G(v) : X_n(w) = 0$ ”, to $X_{n+1}(v) = 1$;
w przeciwnym przypadku $X_{n+1}(v) = 0$
4. $\forall w \neq v : X_{n+1}(w) = X_n(w)$

Powstały w ten sposób łańcuch Markowa jest nieredukowalny i nieokresowy (zad.dom). Pozostaje jeszcze pokazać, że μ_G jest jego rozkładem stacjonarnym, co sprowadza się do sprawdzenia jego odwracalności.

Zatem μ_G będzie odwracalny, gdy:

$$\forall \xi, \xi' \text{ dopuszczalnych: } \mu_G(\xi) \cdot P_{\xi, \xi'} = \mu_G(\xi') \cdot P_{\xi', \xi}$$

Niech $d = d(\xi, \xi')$ będzie odległością Hamminga, będącą liczbą wierzchołków różniących się wartościami w ξ i ξ' .

Należy rozpatrzeć przypadki:

1. $d = 0$, gdy $\xi = \xi'$
2. $d \geq 2$, to $P_{\xi', \xi} = 0$
3. $d = 1$, to $\exists v: \xi(v) = 1 \wedge \xi'(v) = 0$

Zatem otrzymujemy: $\mu_G(\xi) \cdot P_{\xi, \xi'} = \frac{1}{Z_G} \frac{1}{2k} = \mu_G(\xi') \cdot P_{\xi', \xi}$, czyli μ_G jest żądanym rozkładem stacjonarnym.

Podany powyżej algorytm jest szczególnym przypadkiem algorytmu zwanego **próbkowaniem (probierzem) Gibbsa** (ang. Gibbs sampler, Glauber dynamics), który jest używany do symulacji rozkładu prawdopodobieństwa π na przestrzeni stanów D^V , gdzie D jest zbiorem możliwych wartości każdego wierzchołka pochodzącego ze zbioru V .

Próbkowanie Gibbsa:

1. Wylosuj $v \in V$ (z równym prawdopodobieństwem)
2. Wylosuj $X_{n+1}(v)$ zgodnie z warunkowym rozkładem π względem wszystkich wartości $X_n(w)$ dla $w \neq v$
3. $\forall w \neq v: X_{n+1}(w) = X_n(w)$

Przykład 2. Losowe q -kolorowanie

Niech dany będzie graf $G = (V, E)$ oraz $q \geq 2$ kolorów. Przez **kolorowanie grafu** będziemy rozumieć jedno konkretne przypisanie kolorów wierzchołkom (jedno z $[q]^V$ możliwych). Interesować nas będą **q -kolorowania grafu**, czyli kolorowania, w których przypisano wierzchołkom q różnych kolorów w taki sposób, aby żadne dwa sąsiednie nie były tego samego koloru.

Losowym q -kolorowaniem jest q -kolorowanie wybrane z jednakowym prawdopodobieństwem ze zbioru wszystkich możliwych kolorowań, czyli zgodnie z rozkładem jednostajnym $\rho_{G,q}$ na przestrzeni $[q]^V$.

Próbkowanie Gibbsa dla losowego q -kolorowania:

1. Wylosuj $v \in V$ (z równym prawdopodobieństwem)
2. Wylosuj $X_{n+1}(v)$ zgodnie z warunkowym rozkładem $\rho_{G,q}$ pod warunkiem $\{X_n(w) : w \neq v\}$
3. $\forall w \neq v: X_{n+1}(w) = X_n(w)$

Powstały w ten sposób łańcuch Markowa jest nieredukowalny i nieokresowy z rozkładem stacjonarnym $\rho_{G,q}$, przy czym nie jest łatwym zadaniem pokazanie nieredukowalności (zad.dom).

Inny algorytm dla losowego q -kolorowania tzw. wersja Metropolis:

1. Wylosuj $v \in V$ oraz $c \in [q]$
2. Jeżeli „ c jest OK dla v ”, to $X_{n+1}(v) = c$;
w przeciwnym przypadku $X_{n+1}(v) = X_n(v)$
3. $\forall w \neq v: X_{n+1}(w) = X_n(w)$

2 Przybliżone przeliczanie

Nawiązując do Przykładu 2 spróbujemy wejść w zagadnienie przeliczania interesujących nas obiektów za pomocą losowych algorytmów - szybkich i dających dobre przybliżenie z bardzo dużym prawdopodobieństwem.

Niech dany będzie graf $G = (V, E)$ oraz $q \geq 2$ kolorów - jak w Przykładzie 2. Naturalnym pytaniem, jakie się pojawia jest: *Ile jest różnych q -kolorowań danego grafu G ?*

Rozwiązanie naiwne sprowadza się do wygenerowania wszystkich q -kolorowań dla danego grafu G i ich zliczenia. Jest to jednak podejście niepraktyczne, zwłaszcza w przypadku „dużych” grafów, ze względu na bardzo dużą liczbę możliwości kolorowań.

Celem jest zatem znalezienie losowego algorytmu o następujących własnościach:

- (a) o wielomianowym czasie działania względem k
- (b) dla każdego $\epsilon > 0$ zwracającym wynik z spełniający nierówność:

$$(1 - \epsilon) \cdot Z_{G,q} \leq z \leq (1 + \epsilon) \cdot Z_{G,q}$$

- (c) z prawdopodobieństwem sukcesu co najmniej $\frac{2}{3}$ (czyli spełnienia dwóch pierwszych podpunktów).

Algorytmy losowe o powyższych własnościach nazywane są skrótem **RPTAS - Randomized polynomial time approximation scheme**. Natomiast algorytmy, w których do podpunktu (a) dołożymy założenie, iż czas działania algorytmu powinien być wielomianowy względem $\frac{1}{\epsilon}$ nazywamy **FPRAS - Fully polynomial randomized approximation scheme**.

Naiwne podejście:

Niech dany będzie graf $G = (V, E)$, $|V| = k$ oraz $Z_{G,q}$ oznacza liczbę q -kolorowań w G . Przypuśćmy, że każdemu wierzchołkowi został przypisany kolor ze zbioru $[q]$ z jednakowym prawdopodobieństwem. Wówczas każda konfiguracja $\xi \in [q]^V$ powstaje z prawdopodobieństwem $\frac{1}{q^k}$. Zatem prawdopodobieństwo, że podczas tej procedury powstanie q -kolorowanie, jest równe $\frac{Z_{G,q}}{q^k}$.

Rozważmy powtórzenie n razy powyższej procedury. W takim przypadku otrzymamy schemat Bernoulliego n prób z prawdopodobieństwem sukcesu $p = \frac{Z_{G,q}}{q^k}$. Niech Y_n będzie

liczbą właściwych q -kolorowań otrzymanych w n próbach. Wówczas Y_n ma rozkład dwumianowy z $E(Y_n) = n \cdot \frac{Z_{G,q}}{q^k}$, co implikuje, że $E\left(\frac{q^k}{n} Y_n\right) = Z_{G,q}$, czyli $\frac{q^k}{n} Y_n$ jest nieobciążonym estymatorem dla $Z_{G,q}$. Zauważmy, że na podstawie nierówności Czebyszewa

$$\mathbb{P}(|X - E(X)| \geq \epsilon) \leq \frac{Var(X)}{\epsilon^2}$$

mamy

$$\mathbb{P}\left(\left|\frac{q^k}{n} Y_n - Z_{G,q}\right| \geq \epsilon \cdot Z_{G,q}\right) \leq \frac{q^{2k} \cdot Var(Y_n)}{n^2 \epsilon^2 Z_{G,q}^2} \leq \frac{q^k}{n \epsilon^2 Z_{G,q}}$$

gdzie $Var(Y_n) = n \frac{Z_{G,q}}{q^k} \left(1 - \frac{Z_{G,q}}{q^k}\right)$ jest wariancją zmiennej losowej Y_n , dla której zastosowano oszacowanie $1 - \frac{Z_{G,q}}{q^k} \leq 1$.

Aby były spełnione podpunkty (b) i (c) założeń algorytmu *RPTAS*, tzn., aby aproksymacja z błędem ϵ zachodziła z prawdopodobieństwem co najmniej $2/3$, trzeba jednak założyć, że $n \geq \frac{3}{2} \frac{q^k}{\epsilon^2 Z_{G,q}}$, co, na podstawie ZD 8/3, jest co najmniej wykładniczą funkcją k (o podstawie $(q-1)/q$) i nasz algorytm nie spełnia punktu (a) założeń algorytmu *RPTAS*.

Zatem okazuje się, że zastosowanie zaproponowanej naiwnej metody nie jest dobrym rozwiązaniem problemu i trzeba zastosować bardziej wyrafinowany sposób, który zostanie przedstawiony na kolejnym wykładzie.