

Markov Chain Monte Carlo

Jan Kracík

jan.kratick@vsb.cz

Cílem je (přibližný) výpočet integrálu

$$I(g) = E_f[g(X)] = \int g(x)f(x)dx. \quad (1)$$

Umíme-li generovat nezávislé vzorky $x^{(1)}, x^{(2)}, \dots$ z rozdělení s hustotou pravděpodobnosti $f(x)$, lze (1) aproximovat

$$I_N(g) = \frac{1}{N} \sum_{i=1}^N g(x^{(i)}).$$

Podle silného zákona velkých čísel platí

$$I_N(g) \xrightarrow{\text{a.s.}} I(g).$$

Předpokládejme, že hustotu $f(x)$ lze shora omezit integrovatelnou funkcí $m(x) = Mg(x)$, kde $M \geq 1$ a $g(x)$ je hustota pravděpodobnosti.

Algoritmus (generování 1 vzorku)

Vstup: $f(x)$, $g(x)$, M

- 1 Generuj $Y \sim g$.
- 2 Generuj $U \sim \mathcal{U}(\langle 0, 1 \rangle)$.
- 3 Jestliže $U \leq \frac{f(Y)}{Mg(Y)}$, polož $X = Y$. Jinak jdi na. 1

Výstup: X

- Je obtížné najít $g(x)$ splňující požadavky:
 - snadné vzorkování z $g(x)$,
 - vysoká pravděpodobnost přijetí vzorku, tj. malé M , pro něž $f(x) \leq Mg(x)$.
- S rostoucí dimenzí veličiny x , klesá efektivita (pravděpodobnost přijetí vzorku).
Př: vícerozměrné normální rozdělení s omezeným nosičem

Pro větší dimenze je zamítací metoda prakticky nepoužitelná.

Řešením jsou metody MCMC.

Markov Chain Monte Carlo (MCMC) metody jsou simulační metody generující vzorky z ergodického Markovovského řetězce s daným stacionárním rozdělením (cílové rozdělení).

- Markovovský řetězec lze generovat mnohem efektivněji než nezávislé vzorky.
- Standardní nástroj pro bayesovské metody (nelze-li řešit analyticky).
- Lze modifikovat na optimalizační algoritmy.

Metropolisův Hastingsův algoritmus

Metropolisův Hastingsův algoritmus je obecný MCMC algoritmus. Mnoho v praxi využívaných MCMC algoritmů lze chápat jako jeho speciální případy.

Vstupy:

- Cílová hustota $f(x)$
- Návrhová hustota (proposal density) $q(x^*|x)$

Algoritmus je iterační, postupně generuje hodnoty Markovovského řetězce $x^{(1)}, x^{(2)}, \dots$

Tento řetězec označujeme jako *Metropolisův*.

Algorithm 1 Metropolisův Hastingsův algoritmus

- 1: zvol $x^{(0)}$ tak, aby $f(x^{(0)}) > 0$
 - 2: **for** $i = 1$ to N **do**
 - 3: vygeneruj $u \sim \mathcal{U}(\langle 0, 1 \rangle)$
 - 4: vygeneruj $x^* \sim q(x^* | x^{(i-1)})$
 - 5: $\mathcal{A}(x^{(i-1)}, x^*) := \min \left\{ \frac{f(x^*)q(x^{(i-1)} | x^*)}{f(x^{(i-1)})q(x^* | x^{(i-1)})}, 1 \right\}$
 - 6: **if** $u < \mathcal{A}(x^{(i-1)}, x^*)$ **then**
 - 7: $x^{(i)} := x^*$
 - 8: **else**
 - 9: $x^{(i)} := x^{(i-1)}$
 - 10: **end if**
 - 11: **end for**
-

Metropolisův Hastingsův algoritmus

Poznámky:

- z $f(x^{(0)}) > 0$ plyne $f(x^{(i)}) > 0$ pro všechna i
- vzorky x^* s

$$\frac{f(x^*)}{q(x^*|x^{(i-1)})} \geq \frac{f(x^{(i-1)})}{q(x^{(i-1)}|x^*)}$$

jsou vždy přijaty

- pro symetrickou $q(x^*|x^{(i-1)})$, tj.

$$q(x^*|x^{(i)}) = q(x^{(i)}|x^*),$$

je pravděpodobnost přijetí určena poměrem

$$\frac{f(x^*)}{f(x^{(i)})}$$

Podmínky praktického použití:

- $f(x)$ určená až na konstantu
- lze snadno vzorkovat z $q(x^*|x^{(i)})$
- $q(x^*|x^{(i)})$ určená až na konstantu nezávislou na $x^{(i)}$ nebo symetrická
- pro efektivitu algoritmu je zásadní volba vhodné návrhové hustoty $q(x^*|x^{(i)})$

Aby $f(x)$ mohlo být limitním rozdělením Metropolisova řetězce, musí $q(x^*|x^{(i)})$ splňovat nutnou podmínku

$$\text{supp } f \subset \bigcup_{x \in \text{supp } f} \text{supp } q(\cdot|x)$$

Je-li tato podmínka splněna, platí, že $f(x)$ je stacionárním rozdělením Metropolisova řetězce

Tvrzení

Nechť $(X^{(i)})$ je Metropolisův řetězec a pro návrhovou hustotu $q(x^*|x^{(i)})$ a cílovou hustotu $f(x)$ z Metropolisova Hastingsova algoritmu 4 platí

$$\text{supp } f \subset \bigcup_{x \in \text{supp } f} \text{supp } q(\cdot|x).$$

Pak $f(x)$ je stacionárním rozdělením $(X^{(i)})$.

Důkaz

Označme

$$r(x^{(i-1)}) = \int \mathcal{A}(x^{(i-1)}, x) q(x|x^{(i-1)}) dx.$$

Pro přechodovou hustotu pravděpodobnosti Metropolisova řetězce pak platí

$$\begin{aligned} K_{MH}(x^{(i)}|x^{(i-1)}) \\ = \mathcal{A}(x^{(i-1)}, x^{(i)}) q(x^{(i)}|x^{(i-1)}) + (1 - r(x^{(i-1)})) \delta(x^{(i)} - x^{(i-1)}). \end{aligned}$$

Metropolisův Hastingsův algoritmus

Pro $f(x)$ platí

$$\begin{aligned} & \mathcal{A}(x^{(i-1)}, x^{(i)})q(x^{(i)}|x^{(i-1)})f(x^{(i-1)}) \\ &= \min \left\{ \frac{f(x^{(i)})q(x^{(i-1)}|x^{(i)})}{f(x^{(i-1)})q(x^{(i)}|x^{(i-1)})}, 1 \right\} q(x^{(i)}|x^{(i-1)})f(x^{(i-1)}) \\ &= \min \left\{ f(x^{(i)})q(x^{(i-1)}|x^{(i)}), f(x^{(i-1)})q(x^{(i)}|x^{(i-1)}) \right\}. \end{aligned}$$

Podobně

$$\begin{aligned} & \mathcal{A}(x^{(i)}, x^{(i-1)})q(x^{(i-1)}|x^{(i)})f(x^{(i)}) \\ &= \min \left\{ f(x^{(i-1)})q(x^{(i)}|x^{(i-1)}), f(x^{(i)})q(x^{(i-1)}|x^{(i)}) \right\} \\ &= \mathcal{A}(x^{(i-1)}, x^{(i)})q(x^{(i)}|x^{(i-1)})f(x^{(i-1)}) \end{aligned} \tag{2}$$

Metropolisův Hastingsův algoritmus

Dále z vlastností Diracovy δ -funkce plyne

$$\begin{aligned}(1 - r(x^{(i-1)}))\delta(x^{(i)} - x^{(i-1)})f(x^{(i-1)}) \\ = (1 - r(x^{(i)}))\delta(x^{(i-1)} - x^{(i)})f(x^{(i)}).\end{aligned}\quad (3)$$

Z (2) a (3) plyne

$$K(x^{(i)}|x^{(i-1)})f(x^{(i-1)}) = K(x^{(i-1)}|x^{(i)})f(x^{(i)}).$$

Odtud dostáváme

$$\int K(x^{(i)}|x^{(i-1)})f(x^{(i-1)})dx^{(i-1)} = f(x^{(i)}),$$

z čehož plyne, že $f(x)$ je hustotou stacionárního rozdělení Metropolisova řetězce. □

K tomu, aby podobně jako u klasických MC metod platilo

$$I_N(g) = \frac{1}{N} \sum_{i=1}^N g(x^{(i)}) \xrightarrow{a.s.} \int g(x)f(x)dx \quad (4)$$

nezávisle navolbě $x^{(0)}$, musí být splněny další podmínky. Postačující podmínkou je

- 1 aperiodicita a
- 2 tzv. Harrisovská rekurence

Metropolisova řetězce.

Aperiodicita bude zaručena, pokud

$$P\left(f(x^*)q(x^{(i-1)}|x^*) \geq f(x^{(i-1)})q(x^*|x^{(i-1)})\right) < 1,$$

tj. pokud v MH algoritmu může dojít k zamítnutí navrženého vzorku x^* .

Harrisovská rekurence zjednodušeně znamená, že každá trajektorie Markovovského řetězce projde libovolnou množinou nekonečněkrát s pravděpodobností 1. Toto bude splněno, např. tehdy, pokud

$$\forall x^*, x \in \text{supp} : q(x^*|x) > 0.$$

Další postačující podmínkou pro platnost (4) je, aby $f(x)$ byla omezená a kladná na každé kompaktní podmnožině $\text{supp } f$ a aby existovaly $\varepsilon, \delta > 0$ takové, že

$$\forall x^*, x \in \text{supp } f : |x - x^*| < \delta \Rightarrow q(x^*|x) > \varepsilon.$$

Poznámka: Podle obou kritérií konvergence zaručí např. návrhové rozdělení

$$x^*|x \sim \mathcal{N}(x, \sigma^2).$$

Metropolisův Hastingsův algoritmus

V praxi se často využívají speciální případy Metropolisova Hastingsova algoritmu a odvozené varianty.

- Metropolisův algoritmus
- independent sampler
- Gibbsův sampler
- Langevinův algoritmus (Langevinova difúze)
- adaptivní MCMC
- směsi a cykly MH jader
- reversible jump (pro úlohy s volbou modelu)
- simulované žíhání (optimalizační metoda)
- ...

Independent sampler

Speciální případ Metropolisova Hastingsova algoritmu, návrhová hustota nezávisí na předchozích vzorcích.

$$q(x^* | x^{(i-1)}) = q(x^*)$$

Pak

$$\mathcal{A}(x^{(i-1)}, x^*) := \min \left\{ \frac{f(x^*)q(x^{(i-1)})}{f(x^{(i-1)})q(x^*)}, 1 \right\}.$$

Independent sampler bude efektivní jen když návrhová hustota $q(x)$ alespoň přibližně kopíruje cílovou hustotu $f(x)$.

Independent sampler se hodí např. tehdy, chceme-li v Metropolisově řetězci zajistit dlouhé skoky a usnadnit tak procházení celé množiny hodnot (při mnoha lokálních extrémech apod.).

Algorithm 2 Independent sampler

- 1: zvol $x^{(0)}$ tak, aby $f(x^{(0)}) > 0$
 - 2: **for** $i = 1$ to N **do**
 - 3: vygeneruj $u \sim \mathcal{U}(\langle 0, 1 \rangle)$
 - 4: vygeneruj $x^* \sim q(x^* | x^{(i-1)})$
 - 5: $\mathcal{A}(x^{(i-1)}, x^*) := \min \left\{ \frac{f(x^*)q(x^{(i-1)})}{f(x^{(i-1)})q(x^*)}, 1 \right\}$
 - 6: **if** $u < \mathcal{A}(x^{(i-1)}, x^*)$ **then**
 - 7: $x^{(i)} := x^*$
 - 8: **else**
 - 9: $x^{(i)} := x^{(i-1)}$
 - 10: **end if**
 - 11: **end for**
-

Metropolisův algoritmus

Speciální případ Metropolisova Hastingsova algoritmu, návrhová hustota je symetrická.

$$q(x^*|x^{(i-1)}) = q(x^{(i-1)}|x^*)$$

Pak

$$\mathcal{A}(x^{(i-1)}, x^*) := \min \left\{ \frac{f(x^*)}{f(x^{(i-1)})}, 1 \right\}.$$

V praxi často používaný algoritmus, jednoduchá implementace.

Častá volba $q(x^*|x^{(i-1)})$ normální se středem v $x^{(i-1)}$.

Algorithm 3 Metropolisův algoritmus

- 1: zvol $x^{(0)}$ tak, aby $f(x^{(0)}) > 0$
 - 2: **for** $i = 1$ to N **do**
 - 3: vygeneruj $u \sim \mathcal{U}(\langle 0, 1 \rangle)$
 - 4: vygeneruj $x^* \sim q(x^* | x^{(i-1)})$
 - 5: $\mathcal{A}(x^{(i-1)}, x^*) := \min \left\{ \frac{f(x^*)}{f(x^{(i-1)})}, 1 \right\}$
 - 6: **if** $u < \mathcal{A}(x^{(i-1)}, x^*)$ **then**
 - 7: $x^{(i)} := x^*$
 - 8: **else**
 - 9: $x^{(i)} := x^{(i-1)}$
 - 10: **end if**
 - 11: **end for**
-

Simulované žíhání

Simulované žíhání (simulated annealing) je metoda pro globální optimalizaci založená na MCMC algoritmech.
Hledáme

$$\hat{x} = \operatorname{argmax} f(x).$$

Časté využití: MLE, MAP

není omezeno na hustoty ($f(x)$ nemusí být normalizovaná)

Idea:

$$\hat{x} \doteq \operatorname{argmax}_{i=1, \dots, N} f(x^{(i)})$$

Vzorků z okolí glob. extrému bude většinou velmi málo.

Proto vzorkujeme z nehomogenního Markovovského řetězce se stacionárním rozdělením

$$f_i(x) \propto f^{\frac{1}{T_i}}(x).$$

T_i je klesající posloupnost s $\lim_{i \rightarrow \infty} T_i = 0$.

$f^{\frac{1}{T_i}}(x)$ se s rostoucím i koncentruje v okolí globálního maxima.

Úspěch závisí na volbě posloupnosti T_i .

Častá volba

$$T_i = (C \ln(i + T_0))^{-1},$$

kde C, T_0 jsou vhodně zvolené konstanty (závisí na konkrétním problému).

Algorithm 4 Simulované žíhání

- 1: zvol $x^{(0)}$ tak, aby $f(x^{(0)}) > 0$
 - 2: **for** $i = 1$ to N **do**
 - 3: vygeneruj $u \sim \mathcal{U}(\langle 0, 1 \rangle)$
 - 4: vygeneruj $x^* \sim q(x^* | x^{(i-1)})$
 - 5: nastav T_i podle zvoleného schématu
 - 6: $\mathcal{A}(x^{(i-1)}, x^*) := \min \left\{ \frac{f^{\frac{1}{T_i}}(x^*)q(x^{(i-1)} | x^*)}{f^{\frac{1}{T_i}}(x^{(i-1)})q(x^* | x^{(i-1)})}, 1 \right\}$
 - 7: **if** $u < \mathcal{A}(x^{(i-1)}, x^*)$ **then**
 - 8: $x^{(i)} := x^*$
 - 9: **else**
 - 10: $x^{(i)} := x^{(i-1)}$
 - 11: **end if**
 - 12: **end for**
-

Směsi a cykly MCMC jader

Důležitou vlastností MCMC metod je možnost kombinovat více MCMC generátorů.

Mají-li přechodová jádra K_1, K_2 stacionární rozdělení $f(x)$, pak stejné stacionární rozdělení má i směšové jádro

$$\alpha K_1 + (1 - \alpha) K_2,$$

pro $\alpha \in (0, 1)$. Totéž platí pro cyklické jádro

$$K_1 K_2.$$

Směšová jádra

$$\alpha K_1 + (1 - \alpha) K_2$$

mohou např. využívat

- jádro K_1 pro hrubé procházení celého prostoru (Independent sampler),
- jádro K_2 generující malé kroky pro důkladné lokální procházení (Metropolisův algoritmus).

Tento postup je vhodný např. pro hustoty s mnoha vzdálenými lokálními extrémy.

Algorithm 5 MCMC se směšovým jádrem

- 1: zvol $x^{(0)}$ tak, aby $f(x^{(0)}) > 0$
 - 2: **for** $i = 1$ to N **do**
 - 3: vygeneruj $z \sim \mathcal{U}(\langle 0, 1 \rangle)$
 - 4: **if** $z < \alpha$ **then**
 - 5: proved' krok MH algoritmu s jádrem K_1
 - 6: **else**
 - 7: proved' krok MH algoritmu s jádrem K_2
 - 8: **end if**
 - 9: **end for**
-

Cyklická jádra

$$K = K_1 K_2 \dots K_r$$

Ize využít pro generování vzorků náhodných vektorů $x^{(i)}$ po částech. Každé jádro generuje jen některé složky vektoru $x^{(i)}$ (blok). Využívá se např. tehdy, lze-li vektor x rozdělit na bloky, uvnitř kterých jsou složky silně korelované.

Speciálním případem MCMC algoritmu s cyklickým jádrem je *Gibbsův sampler*

Předpokládejme, že $x = (x_1, \dots, x_n)$ je náhodný vektor a pro cílovou hustotu $f(x)$ lze snadno najít podmíněné hustoty

$$f(x_j | x_1, \dots, x_{j-1}, x_{j+1}, \dots, x_n). \quad (5)$$

Gibbsův sampler je MCMC algoritmus, který cyklicky generuje všechny složky vektoru x MH algoritmem s návrhovou hustotou (5).

Algorithm 6 Gibbsův sampler

- 1: zvol $x^{(0)}$ tak, aby $f(x^{(0)}) > 0$
 - 2: **for** $i = 1$ to N **do**
 - 3: vygeneruj $x_1^{(i)} \sim f(x_1 | x_2^{(i-1)}, x_3^{(i-1)}, \dots, x_n^{(i-1)})$
 - 4: vygeneruj $x_2^{(i)} \sim f(x_2 | x_1^{(i)}, x_3^{(i-1)}, \dots, x_n^{(i-1)})$
 - 5: ...
 - 6: vygeneruj $x_j^{(i)} \sim f(x_j | x_1^{(i)}, \dots, x_{j-1}^{(i)}, x_{j+1}^{(i-1)}, \dots, x_n^{(i-1)})$
 - 7: ...
 - 8: vygeneruj $x_n^{(i)} \sim f(x_n | x_1^{(i)}, \dots, x_{n-1}^{(i)})$
 - 9: **end for**
-

Každý dílčí krok 1 iterace Gibbsova sampleru lze chápat jako krok MH algoritmu s návrhovou hustotou

$$q(x^* | x^{(i-1)}) = \begin{cases} f(x_j^* | x_{-j}^{(i-1)}) & \text{pro } x_{-j}^* = x_{-j}^{(i-1)} \\ 0 & \text{jinak,} \end{cases}$$

kde

$$x_{-j} = (x_1, \dots, x_{j-1}, x_{j+1}, \dots, x_n).$$

Pak platí, že

$$\mathcal{A}(x^{(i-1)}, x^*) = 1,$$

tj. k přijetí navrženého vzorku dochází vždy.

Gibbsův sampler se často používá pro Markovovská náhodná pole, protože

$$f(x_j | x_1, \dots, x_{j-1}, x_{j+1}, \dots, x_n) = f(x_j | x_{N_j}),$$

jejíž výpočet bývá relativně snadný a jeho složitost nezávisí na velikosti pole.