Verjetnost 2
Osmo poglavje
Monte Carlo Markovske verige — MCMC

Matjaž Omladič

Oktober 2010 – Januar 2011

Vsebina

Monte Carlo — 1

Metodo Monte Carlo so prvič uporabili pri računanju kritične mase urana pri izdelavi prve atomske bombe (Ulam). Danes nam pri Monte Carlo simulacijah pomagajo močni računalniki. Če poznamo (kumulativno) porazdelitveno funkcijo F slučajne spremenljivke X, če je funkcija F strogo naraščajoča in je slučajna spremenljivka U porazdeljena enakomerno zvezno, potem je $F^{-1}(U) \sim F$.

Kot primer uporabe si poglejmo izračun naslednjega integrala:

$$\int_0^1 x\,dx. \tag{1}$$

Na ta integral lahko pogledamo kot na matematično upanje slučajne spremenljivke U, ki je porazdeljena enakomerno zvezno na inervalu $[0,1]$. Imejmo zaporedje (psevdo)slučajnih števil

$$u_1, u_2, u_3, \ldots \tag{2}$$

neodvisnih in porazdeljenih enako kot U.

Monte Carlo — 1

Metodo Monte Carlo so prvič uporabili pri računanju kritične mase urana pri izdelavi prve atomske bombe (Ulam). Danes nam pri Monte Carlo simulacijah pomagajo močni računalniki.
Če poznamo (kumulativno) porazdelitveno funkcijo F slučajne spremenljivke X, če je funkcija F strogo naraščajoča in je slučajna spremenljivka U porazdeljena enakomerno zvezno, potem je $F^{-1}(U) \sim F$.
Kot primer uporabe si poglejmo izračun naslednjega integrala:

$$\int_0^1 x\,dx. \tag{1}$$

Na ta integral lahko pogledamo kot na matematično upanje slučajne spremenljivke U, ki je porazdeljena enakomerno zvezno na inervalu $[0,1]$. Imejmo zaporedje (psevdo)slučajnih števil

$$u_1, u_2, u_3, \ldots \tag{2}$$

neodvisnih in porazdeljenih enako kot U.

Monte Carlo — 2

Nanj lahko gledamo kot na realizacijo zaporedja n.e.p. slučajnih spremenljivk $U_1, U_2, U_3, \ldots$, porazdeljenih enako kot U. Vemo, da za to zaporedje velja Krepki zakon velikih števil, to pomeni, da zaporedje delnih vsot konvergira skoraj gotovo proti matematičnemu upanju spremenljivke U, torej proti integralu (1). Tvorimo zaporedje povprečij

$$\overline{u}_n = \frac{1}{n} \sum_{k=1}^{n} u_k$$

števil (2). To je torej realizacija zaporedja slučajnih spremenljivk, ki po KZVŠ konvergirajo skoraj gotovo proti integralu (1). To je osnovna varianta numerične metode Monte Carlo za računanje tega integrala. Kaj pa, če računamo kak bolj zapleten integral? Na primer $E(X)$, kjer je X slučajna spremenljivka, porazdeljena po (strogo naraščajoči) porazdelitveni funkciji F.

Monte Carlo — 2

Nanj lahko gledamo kot na realizacijo zaporedja n.e.p. slučajnih spremenljivk $U_1, U_2, U_3, \ldots$, porazdeljenih enako kot U. Vemo, da za to zaporedje velja Krepki zakon velikih števil, to pomeni, da zaporedje delnih vsot konvergira skoraj gotovo proti matematičnemu upanju spremenljivke U, torej proti integralu (1). Tvorimo zaporedje povprečij

$$\overline{u}_n = \frac{1}{n} \sum_{k=1}^{n} u_k$$

števil (2). To je torej realizacija zaporedja slučajnih spremenljivk, ki po KZVŠ konvergirajo skoraj gotovo proti integralu (1). To je osnovna varianta numerične metode Monte Carlo za računanje tega integrala. Kaj pa, če računamo kak bolj zapleten integral? Na primer $E(X)$, kjer je X slučajna spremenljivka, porazdeljena po (strogo naraščajoči) porazdelitveni funkciji F.

Monte Carlo — 3

Tedaj je (do porazdelitve natančno) $X = F^{-1}(U)$, kjer je U porazdeljen enakomerno zvezno na intervalu $[0,1]$. Če ima X matematično upanje, potem konvergira zaporedje povprečij

$$\overline{x}_n = \frac{1}{n}\sum_{k=1}^{n} F^{-1}(u_k),$$

kjer so u_k števila (2), po KZVŠ skoraj gotovo proti $E(X)$. Izberemo dovolj dolgo (npr 50000) tako zaporedje. Rezultat je približek za iskano matematično upanje.

No, v praksi je porazdelitveni zakon F odvisen še od parametrov in drugih spremenljivk. To odvisnost nam pomaga modelirati statistika v ti. Bayesovem pristopu.

Postopek začnemo tako: Izberemo začetne vrednosti teh parametrov in spremenljivk. Iz njih z uporabo Bayesovih metod in (psevdo)slučajnih generatorjev določimo prvi približek x_1 slučajne spremenljivke X.

Monte Carlo — 3

Tedaj je (do porazdelitve natančno) $X = F^{-1}(U)$, kjer je U porazdeljen enakomerno zvezno na intervalu $[0, 1]$. Če ima X matematično upanje, potem konvergira zaporedje povprečij

$$\overline{x}_n = \frac{1}{n}\sum_{k=1}^{n} F^{-1}(u_k),$$

kjer so u_k števila (2), po KZVŠ skoraj gotovo proti $E(X)$. Izberemo dovolj dolgo (npr 50000) tako zaporedje. Rezultat je približek za iskano matematično upanje.
No, v praksi je porazdelitveni zakon F odvisen še od parametrov in drugih spremenljivk. To odvisnost nam pomaga modelirati statistika v ti. Bayesovem pristopu.
Postopek začnemo tako: Izberemo začetne vrednosti teh parametrov in spremenljivk. Iz njih z uporabo Bayesovih metod in (psevdo)slučajnih generatorjev določimo prvi približek x_1 slučajne spremenljivke X.

MCMV

Na tej podlagi izračunamo nove vrednosti parametrov in pomožnih spremenlivk ter iz njih po istem postopku kot prej vrednost x_2. Postopek ponavljamo. Dobljeno zaporedje

$$x_1, x_2, x_3, \ldots$$

ni več realizacija nekega zaporedja n.e.p., temveč je realizacija neke markovske verige.

Če je veriga ergodijska, zaporedje konvergira. V praksi bomo morali zaporedje nekje končati. Če ga bomo končali dovolj pozno (npr po 50000 korakih), bomo dobili eno točko, porazdeljeno po istem zakonu kot iskana spremenljivka X. Ko najdemo celo množico (npr 50000) takih točk, dobimo precej dober vtis o iskani porazdelitvi.

MCMV

Na tej podlagi izračunamo nove vrednosti parametrov in pomožnih spremenlivk ter iz njih po istem postopku kot prej vrednost x_2. Postopek ponavljamo. Dobljeno zaporedje

$$x_1, x_2, x_3, \dots$$

ni več realizacija nekega zaporedja n.e.p., temveč je realizacija neke markovske verige.

Če je veriga ergodijska, zaporedje konvergira. V praksi bomo morali zaporedje nekje končati. Če ga bomo končali dovolj pozno (npr po 50000 korakih), bomo dobili eno točko, porazdeljeno po istem zakonu kot iskana spremenljivka X. Ko najdemo celo množico (npr 50000) takih točk, dobimo precej dober vtis o iskani porazdelitvi.

Tu opisana metoda se imenuje metoda "Monte Carlo markovskih verig", na kratko MCMV oz. po angleško Monte Carlo Markov Chains , kar krajšamo v MCMC.

MCMV

Na tej podlagi izračunamo nove vrednosti parametrov in pomožnih spremenlivk ter iz njih po istem postopku kot prej vrednost x_2. Postopek ponavljamo. Dobljeno zaporedje

$$x_1, x_2, x_3, \dots$$

ni več realizacija nekega zaporedja n.e.p., temveč je realizacija neke markovske verige.
Če je veriga ergodijska, zaporedje konvergira. V praksi bomo morali zaporedje nekje končati. Če ga bomo končali dovolj pozno (npr po 50000 korakih), bomo dobili eno točko, porazdeljeno po istem zakonu kot iskana spremenljivka X. Ko najdemo celo množico (npr 50000) takih točk, dobimo precej dober vtis o iskani porazdelitvi.
Tu opisana metoda se imenuje metoda "Monte Carlo markovskih verig", na kratko MCMV oz. po angleško Monte Carlo Markov Chains , kar krajšamo v MCMC.

Zgled

Oglejmo si preprost primer uporabe MC metode za kreiranje niza (psevdo)slučajnih parov števil, porazdeljenih po skupni gostoti

$$f(x,y) = kx^2 e^{-xy^2-y^2+2y-4x}.$$

Težko bomo našli paket, v katerem bi bila ta gostota že zakodirana. Vendar opazimo, da sta pogojni gostoti običajni: $X/y \sim \Gamma(3, (y^2+4)^{-1})$ in $Y/x \sim N((x+1)^{-1}, 2(x+1)^{-1}$. Začnemo s poljubnima približkoma x_0 in y_0. Generiramo x_1 porazdeljen po $\Gamma(3, (y_0^2+4)^{-1})$ ter y_1 porazdeljen po $N((x_0+1)^{-1}, 2(x_0+1)^{-1}$. Postopek ponavljamo. Prvih K parov odvržemo, ker ocenimo, da še niso prav porazdeljeni. Obdržimo naslednjih N, kolikor jih potrebujemo za nadaljevanje simulacij. Tule je koda tega (Gibbsovega) algoritma v R.

$$\begin{aligned} \text{gibbs} &= \text{function}(N, K, x_0, y_0) \\ \text{rezultat} &= \text{matrix}(0, \text{ncol} = 3, \text{nrow} = N) \end{aligned}$$

Zgled

Oglejmo si preprost primer uporabe MC metode za kreiranje niza (psevdo)slučajnih parov števil, porazdeljenih po skupni gostoti

$$f(x,y) = kx^2 e^{-xy^2-y^2+2y-4x}.$$

Težko bomo našli paket, v katerem bi bila ta gostota že zakodirana. Vendar opazimo, da sta pogojni gostoti običajni: $X/y \sim \Gamma(3,(y^2+4)^{-1})$ in $Y/x \sim N((x+1)^{-1}, 2(x+1)^{-1}$. Začnemo s poljubnima približkoma x_0 in y_0. Generiramo x_1 porazdeljen po $\Gamma(3,(y_0^2+4)^{-1})$ ter y_1 porazdeljen po $N((x_0+1)^{-1}, 2(x_0+1)^{-1}$. Postopek ponavljamo. Prvih K parov odvržemo, ker ocenimo, da še niso prav porazdeljeni. Obdržimo naslednjih N, kolikor jih potrebujemo za nadaljevanje simulacij. Tule je koda tega (Gibbsovega) algoritma v R.

$$\begin{aligned} \text{gibbs} &= \text{function}(N, K, x_0, y_0) \\ \text{rezultat} &= \text{matrix}(0, \text{ncol} = 3, \text{nrow} = N) \end{aligned}$$

Zgled — nadaljevanje

$$
\begin{aligned}
\text{rezultat}[,1] &= 1:N \\
& \quad \text{for}(i \ \ \text{in} \ \ 1:(N+K)) \\
\{x &= \text{rgamma}(1,3,1/(y_0 * y_0 + 4)) \\
y &= \text{rnorm}(1,1/(x_0+1),\text{sqrt}(2/(x_0+1))) \\
x_0 &= x \\
y_0 &= y \\
& \quad \text{if}(i > K) \\
\{\text{rezultat}[i-K,2:3] &= c(x,y)\}\} \\
& \quad \text{rezultat}
\end{aligned}
$$

Odprto ostane vptašanje, koliko začetnih približkov naj odvržemo, predno se bo porazdelitev markovske verige stabilizirala pri iskani. Ta primer uporabe Gibbsovega algoritma ima že nekatere Bayesovske elemente.

Zgled — nadaljevanje

$$
\begin{aligned}
\text{rezultat}[,1] &= 1:N \\
& \text{for}(i \ \ \text{in} \ \ 1:(N+K)) \\
\{x &= \text{rgamma}(1,3,1/(y_0 * y_0 + 4)) \\
y &= \text{rnorm}(1,1/(x_0+1),\text{sqrt}(2/(x_0+1))) \\
x_0 &= x \\
y_0 &= y \\
& \text{if}(i > K) \\
\{\text{rezultat}[i-K,2:3] &= c(x,y)\}\} \\
& \text{rezultat}
\end{aligned}
$$

Odprto ostane vptašanje, koliko začetnih približkov naj odvržemo, predno se bo porazdelitev markovske verige stabilizirala pri iskani. Ta primer uporabe Gibbsovega algoritma ima že nekatere Bayesovske elemente.

Bayesovo statistično sklepanje

Delovanje Bayesove statistike lahko opišemo v treh korakih:

- Postavimo verjetnostni model s *skupno verjetnostno porazdelitvijo* vseh *opaznih* in *neopaznih* količin na podlagi našega vedenja o problemu ter vseh doslej zbranih podatkov.
- Na podlagi *opaženih podatkov* in iz modela dobljenih *pogojnih verjetnosti* izračunamo in interpretiramo *aposteriorne porazdelitve* neopaznih količin, ki nas zanimajo.
- Sledi evaluacija modela na podlagi aposteriornih porazdelitev in po potrebi ponovitev postopka.

Medtem ko Bayesov statistik verjame, da se v Bayesovem intervalu za neznano količino nahaja ta količina z veliko verjetnostjo, pa frekventistični statistik verjame le to, da je "njegov" interval zaupanja nekaj, kar je v zvezi z velikim številom ponavljanja nekega poskusa v podobnih poskusih.

Bayesovo statistično sklepanje

Delovanje Bayesove statistike lahko opišemo v treh korakih:

- Postavimo verjetnostni model s *skupno verjetnostno porazdelitvijo* vseh *opaznih* in *neopaznih* količin na podlagi našega vedenja o problemu ter vseh doslej zbranih podatkov.
- Na podlagi *opaženih podatkov* in iz modela dobljenih *pogojnih verjetnosti* izračunamo in interpretiramo *aposteriorne porazdelitve* neopaznih količin, ki nas zanimajo.
- Sledi evaluacija modela na podlagi aposteriornih porazdelitev in po potrebi ponovitev postopka.

Medtem ko Bayesov statistik verjame, da se v Bayesovem intervalu za neznano količino nahaja ta količina z veliko verjetnostjo, pa frekventistični statistik verjame le to, da je "njegov" interval zaupanja nekaj, kar je v zvezi z velikim številom ponavljanja nekega poskusa v podobnih poskusih.

Bayesovo statistično sklepanje — nadaljevanje

Generični primer je novo zdravilo za raka, ki ga testiramo v nekem poskusu v primerjavi z obstoječim zdravilom na podlagi verjetnosti, da bo pacient, ki jemlje zdravilo, živel vsaj še 5 let. Seveda je potrebno izbrati slučajni vzorec.

Pomembno je razlikovati med parametri in podatki. Količine, ki jih ne opazujemo, pa nas zanimajo, delimo na *opazne* (to so tiste, ki jih lahko opazujemo, kot npr rezultati bodočih opazovanj) ter *neopazne* (to so tiste, ki so sestavni del našega modela, pa jih ni mogoče opazovati) - navadno pravimo tem tudi *parametri modela*. Te neopazne količine postavimo v vektor θ (če rabimo več črk, uporabimo še druge grške črke). Latinske črke uporabljamo za (vektor) opaznih in "opaženih" količin (kot npr y bi lahko pomenilo v generičnem primeru število preživelih, ali pa umrlih, v preskušani skupini pacientov). Latinske črke z vijugo (kot npr $\widetilde{y}$) pa pomenijo neznane a opazne količine. Oglejmo si zdaj "statistično" Bayesovo pravilo.

Bayesovo statistično sklepanje — nadaljevanje

Generični primer je novo zdravilo za raka, ki ga testiramo v nekem poskusu v primerjavi z obstoječim zdravilom na podlagi verjetnosti, da bo pacient, ki jemlje zdravilo, živel vsaj še 5 let. Seveda je potrebno izbrati slučajni vzorec.

Pomembno je razlikovati med parametri in podatki. Količine, ki jih ne opazujemo, pa nas zanimajo, delimo na *opazne* (to so tiste, ki jih lahko opazujemo, kot npr rezultati bodočih opazovanj) ter *neopazne* (to so tiste, ki so sestavni del našega modela, pa jih ni mogoče opazovati) - navadno pravimo tem tudi *parametri modela*. Te neopazne količine postavimo v vektor θ (če rabimo več črk, uporabimo še druge grške črke). Latinske črke uporabljamo za (vektor) opaznih in "opaženih" količin (kot npr y bi lahko pomenilo v generičnem primeru število preživelih, ali pa umrlih, v preskušani skupini pacientov). Latinske črke z vijugo (kot npr $\widetilde{y}$) pa pomenijo neznane a opazne količine. Oglejmo si zdaj "statistično" Bayesovo pravilo.

Bayesovo pravilo — 1

Imejmo verjetnostni model za vektor podatkov (tj opazovanih vrednosti) y in vektor parametrov θ. Model predstavlja skupna verjetnostna gostota $p(\theta, y)$ teh dveh vektorjev. Gostota je vselej označena z isto generično črko p neodvisno od tega, za katero konkretno gostoto gre. Marginalno porazdelitev parametrov privzamemo za znano in njeno gostoto verjetnosti $p(\theta)$ poimenujemo *apriorna porazdelitev*. Poleg te imejmo še pogojno gostoto verjetnosti podatkov glede na parametre $p(y|\theta)$, ki jo poimenujemo *porazdelitev podatkov*. Skupna gostota je podana s posplošenim pravilom množenja

$$p(\theta, y) = p(\theta)p(y|\theta).$$

Aposteriorno gostoto dobimo zdaj po Bayesovem pravilu

$$p(\theta|y) = \frac{p(\theta, y)}{p(y)} = \frac{p(\theta)p(y|\theta)}{p(y)}.$$

Bayesovo pravilo — 1

Imejmo verjetnostni model za vektor podatkov (tj opazovanih vrednosti) y in vektor parametrov θ. Model predstavlja skupna verjetnostna gostota $p(\theta, y)$ teh dveh vektorjev. Gostota je vselej označena z isto generično črko p neodvisno od tega, za katero konkretno gostoto gre. Marginalno porazdelitev parametrov privzamemo za znano in njeno gostoto verjetnosti $p(\theta)$ poimenujemo *apriorna porazdelitev*. Poleg te imejmo še pogojno gostoto verjetnosti podatkov glede na parametre $p(y|\theta)$, ki jo poimenujemo *porazdelitev podatkov*. Skupna gostota je podana s posplošenim pravilom množenja

$$p(\theta, y) = p(\theta)p(y|\theta).$$

Aposteriorno gostoto dobimo zdaj po Bayesovem pravilu

$$p(\theta|y) = \frac{p(\theta, y)}{p(y)} = \frac{p(\theta)p(y|\theta)}{p(y)}.$$

Bayesovo pravilo — 2

Ostane nam določiti marginalno gostoto $p(y)$. To dobimo iz normalizacijskega pogoja. Bayesovih statistikov ne zanimajo mere, temveč poznajo samo dve vrsti slučajnih spremenljivk, to so diskretne in zvezne. Prav tako jih zanimajo samo modeli, v katerih so bodisi vse spremenljivke diskretne bodisi so vse zvezne. V prvem primeru nam normalizacijski pogoj da

$$p(y) = \sum_{\theta} p(\theta)p(y|\theta),$$

v drugem primeru pa

$$p(y) = \int_{\theta} p(\theta)p(y|\theta)\, d\theta.$$

Bayesovo pravilo zapišimo še brez normalizacijske konstante v sorazmernostni obliki

$$p(\theta|y) \propto p(\theta)p(y|\theta). \tag{3}$$

Bayesovo pravilo — 2

Ostane nam določiti marginalno gostoto $p(y)$. To dobimo iz normalizacijskega pogoja. Bayesovih statistikov ne zanimajo mere, temveč poznajo samo dve vrsti slučajnih spremenljivk, to so diskretne in zvezne. Prav tako jih zanimajo samo modeli, v katerih so bodisi vse spremenljivke diskretne bodisi so vse zvezne. V prvem primeru nam normalizacijski pogoj da

$$p(y) = \sum_{\theta} p(\theta)p(y|\theta),$$

v drugem primeru pa

$$p(y) = \int_{\theta} p(\theta)p(y|\theta)\, d\theta.$$

Bayesovo pravilo zapišimo še brez normalizacijske konstante v sorazmernostni obliki

$$p(\theta|y) \propto p(\theta)p(y|\theta). \tag{3}$$

Bayesovo pravilo — 3

Opazimo, da so si Bayesovi statistiki svoj matematični del življenja zelo olajšali. Tudi diskretne verjetnostne funkcije imenujejo gostote in označujejo enako. Vse označujejo z isto sugestivno črko p. Vendar pa tak telegrafski zapis v resnici pomaga k bolj jasno predstavljeni ideji.

Povejmo še nekaj o napovedovanju. Pivzamemo, da so napovedovane količine $\widetilde{y}$ pogojno neodvisne od opazovanih količin y pri danem θ. Od tod sledi

$$p(\widetilde{y}|y) = \int p(\widetilde{y},\theta|y)\,d\theta = \int p(\widetilde{y}|\theta)p(\theta|y)\,d\theta$$

Bayesovo pravilo — 3

Opazimo, da so si Bayesovi statistiki svoj matematični del življenja zelo olajšali. Tudi diskretne verjetnostne funkcije imenujejo gostote in označujejo enako. Vse označujejo z isto sugestivno črko p. Vendar pa tak telegrafski zapis v resnici pomaga k bolj jasno predstavljeni ideji.

Povejmo še nekaj o napovedovanju. Pivzamemo, da so napovedovane količine $\widetilde{y}$ pogojno neodvisne od opazovanih količin y pri danem θ. Od tod sledi

$$p(\widetilde{y}|y) = \int p(\widetilde{y}, \theta|y)\, d\theta = \int p(\widetilde{y}|\theta) p(\theta|y)\, d\theta$$

Za konec tega minimalističnega uvoda v Bayesovo statistiko naj poudarimo, da v pravilu (3) vplivajo podatki na aposteriorno porazdelitev samo preko gostote $p(y|\theta)$. Kadar jo gledamo pri fiksnem podatku y kot funkcijo θ, jo zato poimenujemo *funkcija verjetja.*

Bayesovo pravilo — 3

Opazimo, da so si Bayesovi statistiki svoj matematični del življenja zelo olajšali. Tudi diskretne verjetnostne funkcije imenujejo gostote in označujejo enako. Vse označujejo z isto sugestivno črko p. Vendar pa tak telegrafski zapis v resnici pomaga k bolj jasno predstavljeni ideji.

Povejmo še nekaj o napovedovanju. Pivzamemo, da so napovedovane količine $\widetilde{y}$ pogojno neodvisne od opazovanih količin y pri danem θ. Od tod sledi

$$p(\widetilde{y}|y) = \int p(\widetilde{y},\theta|y)\,d\theta = \int p(\widetilde{y}|\theta)p(\theta|y)\,d\theta$$

Za konec tega minimalističnega uvoda v Bayesovo statistiko naj poudarimo, da v pravilu (3) vplivajo podatki na aposteriorno porazdelitev samo preko gostote $p(y|\theta)$. Kadar jo gledamo pri fiksnem podatku y kot funkcijo θ, jo zato poimenujemo *funkcija verjetja*.

Funkcija verjetja

Na dani množici podatkov pripeljejo modeli z isto funkcijo verjetja do istih aposteriornih porazdelitev. Ta princip je smiseln le takrat, kadar smo za obravnavo problema poiskali pravi model. Uporabni Bayesov statistik mora znati svoje modele preverjati in jih po potrebi tudi dopolniti ali spremeniti. Oglejmo si kvocient med aposteriornima gostotama $p(\theta|y)$ v dveh točkah θ_1 in θ_2, pa tudi kvocient med funkcijama verjetja v teh dveh točkah θ_1 in θ_2. Med obema kvocientoma obstaja preprosta zveza, ki jo dobimo po pravilu (3):

$$\frac{p(\theta_1|y)}{p(\theta_2|y)} = \frac{p(\theta_1)}{p(\theta_2)} \frac{p(y|\theta_1)}{p(y|\theta_2)}.$$

Funkcija verjetja

Na dani množici podatkov pripeljejo modeli z isto funkcijo verjetja do istih aposteriornih porazdelitev. Ta princip je smiseln le takrat, kadar smo za obravnavo problema poiskali pravi model. Uporabni Bayesov statistik mora znati svoje modele preverjati in jih po potrebi tudi dopolniti ali spremeniti. Oglejmo si kvocient med aposteriornima gostotama $p(\theta|y)$ v dveh točkah θ_1 in θ_2, pa tudi kvocient med funkcijama verjetja v teh dveh točkah θ_1 in θ_2. Med obema kvocientoma obstaja preprosta zveza, ki jo dobimo po pravilu (3):

$$\frac{p(\theta_1|y)}{p(\theta_2|y)} = \frac{p(\theta_1)}{p(\theta_2)} \frac{p(y|\theta_1)}{p(y|\theta_2)}.$$

Kvocient med dvema alternativnima verjetnostma poimenujemo tudi kvocient priložnosti in predstavlja alternativni prikaz verjetnosti. Zgornjo enačbo lahko povemo drugače tako, da je kvocient aposteriornih priložnosti enak kvocientu apriornih priložnosti pomnoženem s kvocientom verjetij.

Funkcija verjetja

Na dani množici podatkov pripeljejo modeli z isto funkcijo verjetja do istih aposteriornih porazdelitev. Ta princip je smiseln le takrat, kadar smo za obravnavo problema poiskali pravi model. Uporabni Bayesov statistik mora znati svoje modele preverjati in jih po potrebi tudi dopolniti ali spremeniti. Oglejmo si kvocient med aposteriornima gostotama $p(\theta|y)$ v dveh točkah θ_1 in θ_2, pa tudi kvocient med funkcijama verjetja v teh dveh točkah θ_1 in θ_2. Med obema kvocientoma obstaja preprosta zveza, ki jo dobimo po pravilu (3):

$$\frac{p(\theta_1|y)}{p(\theta_2|y)} = \frac{p(\theta_1)}{p(\theta_2)} \frac{p(y|\theta_1)}{p(y|\theta_2)}.$$

Kvocient med dvema alternativnima verjetnostma poimenujemo tudi kvocient priložnosti in predstavlja alternativni prikaz verjetnosti. Zgornjo enačbo lahko povemo drugače tako, da je kvocient aposteriornih priložnosti enak kvocientu apriornih priložnosti pomnoženem s kvocientom verjetij.

Izrek Clifford-Hammersley

Pri vrednotenju finančnih sredstev v zveznem času naletimo pogosto na izjemno zapletene večrazsežne porazdelitve, ki otežujejo možnost za neposredno slučajno vzorčenje. Finančniki govorijo o *prekletstvu mnogih razsežnosti*. V metodah MCMV se tega problema lotimo tako, da razbijemo skupno gostoto verjetnosti na pogojne verjetnosti vse manjših razsežnosti. Teoretično podlago za to predstavlja izjemni izrek Clifforda in Hammersleya, ki pa ga nista baje nikoli objavila. V mnogih aplikacijah je potrebno uporabiti ta izrek po večkrat, dokler ne razbijemo prekletstva mnogih razsežnosti.

Izrek (Clifford-Hammersley)

Pogojne gostote $p(\theta_1|\theta_2,\ldots,\theta_k,y)$, $p(\theta_2|\theta_1,\ldots,\theta_k,y)$, pa vse do $p(\theta_k|\theta_2,\ldots,\theta_{k-1},y)$, enolično določajo (pod nekimi tehničnimi predpostavkami) gostoto $p(\theta_1,\theta_2,\ldots,\theta_k|y)$.

Idejo dokaza zapišimo za dvorazsežni brezpogojni primer.

Izrek Clifford-Hammersley

Pri vrednotenju finančnih sredstev v zveznem času naletimo pogosto na izjemno zapletene večrazsežne porazdelitve, ki otežujejo možnost za neposredno slučajno vzorčenje. Finančniki govorijo o *prekletstvu mnogih razsežnosti*. V metodah MCMV se tega problema lotimo tako, da razbijemo skupno gostoto verjetnosti na pogojne verjetnosti vse manjših razsežnosti. Teoretično podlago za to predstavlja izjemni izrek Clifforda in Hammersleya, ki pa ga nista baje nikoli objavila. V mnogih aplikacijah je potrebno uporabiti ta izrek po večkrat, dokler ne razbijemo prekletstva mnogih razsežnosti.

Izrek (Clifford-Hammersley)

Pogojne gostote $p(\theta_1|\theta_2,\dots,\theta_k,y)$, $p(\theta_2|\theta_1,\dots,\theta_k,y)$, pa vse do $p(\theta_k|\theta_2,\dots,\theta_{k-1},y)$, enolično določajo (pod nekimi tehničnimi predpostavkami) gostoto $p(\theta_1,\theta_2,\dots,\theta_k|y)$.

Idejo dokaza zapišimo za dvorazsežni brezpogojni primer.

Izrek Clifford-Hammersley — nadaljevanje

Uporabimo obe varianti Bayesovega pravila, da dobimo

$$p(x,y) = p(x)p(y|x) = p(y)p(x|y).$$

To nas pripelje do enačbe

$$p(x) = p(y)\frac{p(x|y)}{p(y|x)}.$$

To integriramo na obeh straneh po spremenljivki x, da dobimo

$$\frac{1}{p(y)} = \int \frac{p(x|y)}{p(y|x)}\,dx$$

in končno

$$p(x,y) = p(y|x)\left[\int \frac{p(y|x)}{p(x|y)}\,dy\right]^{-1} = p(x|y)\left[\int \frac{p(x|y)}{p(y|x)}\,dx\right]^{-1}.$$

Izrek Clifford-Hammersley – nadaljevanje

Uporabimo obe varianti Bayesovega pravila, da dobimo

$$p(x, y) = p(x)p(y|x) = p(y)p(x|y).$$

To nas pripelje do enačbe

$$p(x) = p(y)\frac{p(x|y)}{p(y|x)}.$$

To integriramo na obeh straneh po spremenljivki x, da dobimo

$$\frac{1}{p(y)} = \int \frac{p(x|y)}{p(y|x)}\, dx$$

in končno

$$p(x, y) = p(y|x)\left[\int \frac{p(y|x)}{p(x|y)}\, dy\right]^{-1} = p(x|y)\left[\int \frac{p(x|y)}{p(y|x)}\, dx\right]^{-1}.$$

Zgled: Gibbsov vzorčevalnik

V tem zgledu začnimo postopoma uvajati oznake, ki so bolj v navadi pri uporabi metod MCMV v finančni matematiki. Medtem ko rezerviramo oznako y predvsem za cene finančnih sredstev, vpeljemo še oznako x, ki je namenjena "stanjem sistema", to so predvsem tiste spremenljivke, ki niso opazne, pa jih ne moremo šteti med parametre. Pri Gibbsovem algoritmu predpostavimo, da poznamo robni verjetji $p(\theta|x, y)$ in $p(x|\theta, y)$. Cilj je, da pri danih podatkih y konstruiramo slučajni vzorec parov (θ, x), porazdeljenih z gostoto $p(\theta, x|y)$. Algoritem je preprost. Najprej izberemo (poljubna) začetna približka θ_0 in x_0. Nato slučajno "izvlečemo" prvi naslednji par približkov θ_1 in x_1 po pravilu:

$$(1)\ \text{izberi} \quad \theta_1 \sim p(\theta|x_0, y) \qquad (2)\ \text{izberi} \quad x_1 \sim p(x|\theta_0, y)$$

Postopek nadaljujemo in prvih nekaj parov odvržemo, preostali pa tvorijo iskani slučajni vzorec.

Zgled: Gibbsov vzorčevalnik

V tem zgledu začnimo postopoma uvajati oznake, ki so bolj v navadi pri uporabi metod MCMV v finančni matematiki. Medtem ko rezerviramo oznako y predvsem za cene finančnih sredstev, vpeljemo še oznako x, ki je namenjena "stanjem sistema", to so predvsem tiste spremenljivke, ki niso opazne, pa jih ne moremo šteti med parametre. Pri Gibbsovem algoritmu predpostavimo, da poznamo robni verjetji $p(\theta|x, y)$ in $p(x|\theta, y)$. Cilj je, da pri danih podatkih y konstruiramo slučajni vzorec parov (θ, x), porazdeljenih z gostoto $p(\theta, x|y)$. Algoritem je preprost. Najprej izberemo (poljubna) začetna približka θ_0 in x_0. Nato slučajno "izvlečemo" prvi naslednji par približkov θ_1 in x_1 po pravilu:

$$(1)\ \text{izberi}\quad \theta_1 \sim p(\theta|x_0, y) \qquad (2)\ \text{izberi}\quad x_1 \sim p(x|\theta_0, y)$$

Postopek nadaljujemo in prvih nekaj parov odvržemo, preostali pa tvorijo iskani slučajni vzorec.

Zgled: Gibbsov vzorčevalnik — nadaljevanje

Če ne poznamo porazdelitev, ker so v parametru ostale še premnoge razsežnosti, lomimo to prekletstvo z nadaljnjim drobljenjem po Clifford-Hammersleyju, npr pri iskanju skupne porazdelitve $(\eta, \theta, x|y)$ "popravimo algoritem" po izbiri začetnih približkov η_0, θ_0 in x_0 v

$$\begin{aligned}
&(1)\ \text{izberi} \quad \eta_1 \sim p(\eta|\theta_0, x_0, y)\\
&(2)\ \text{izberi} \quad \theta_1 \sim p(\theta|\eta_0, x_0, y)\\
&(3)\ \text{izberi} \quad x_1 \sim p(x|\eta_0, \theta_0, y)
\end{aligned}$$

Kadar so tudi porazdelitve nižjih razsežnosti nestandardne, ne moremo izbirati približkov avtomatično in uporabimo kakšno numerično metodo, npr ti *požrešno Gibbsovo*. Nestandardno porazdelitev (npr $p(\eta|\theta_0, x_0, y)$) aproksimiramo z diskretno in izračunamo obrat porazdelitvene funkcije te aproksimacije. Slednjo uporabimo na slučajni izbiri, porazdeljeni enakomerno zvezno na intervalu $[0, 1]$.

Zgled: Gibbsov vzorčevalnik — nadaljevanje

Če ne poznamo porazdelitev, ker so v parametru ostale še premnoge razsežnosti, lomimo to prekletstvo z nadaljnjim drobljenjem po Clifford-Hammersleyju, npr pri iskanju skupne porazdelitve $(\eta, \theta, x|y)$ "popravimo algoritem" po izbiri začetnih približkov η_0, θ_0 in x_0 v

$$\begin{aligned}
&(1) \text{ izberi} \quad \eta_1 \sim p(\eta|\theta_0, x_0, y)\\
&(2) \text{ izberi} \quad \theta_1 \sim p(\theta|\eta_0, x_0, y)\\
&(3) \text{ izberi} \quad x_1 \sim p(x|\eta_0, \theta_0, y)
\end{aligned}$$

Kadar so tudi porazdelitve nižjih razsežnosti nestandardne, ne moremo izbirati približkov avtomatično in uporabimo kakšno numerično metodo, npr ti *požrešno Gibbsovo*. Nestandardno porazdelitev (npr $p(\eta|\theta_0, x_0, y)$) aproksimiramo z diskretno in izračunamo obrat porazdelitvene funkcije te aproksimacije. Slednjo uporabimo na slučajni izbiri, porazdeljeni enakomerno zvezno na intervalu $[0, 1]$.

Osnovna varianta

Pogosto se dogaja, da pogojnih porazdelitev ne znamo dobro vzorčiti, kar nam oži uporabo Gibbsove metode. Včasih porazdelitev ni na nobenem seznamu znanih gostot ali pa je in enostavno ni zakodirana v paketih, ki jih imamo na voljo. Dokaj splošen pristop je v takih primerih algoritem Metropolis-Hastings.

Naj bo pogojna porazdelitev $p(\theta|\eta, x, y)$, ki jo preprosto označimo s $\pi(\theta)$, taka, da znamo gostoto izračunati kot funkcijo θ, ne znamo pa po njej slučajno izbirati. Od raziskovalca pričakujemo, da zna predlagati neko pogojno gostoto $q(\theta_{n+1}|\theta_n)$. Ta naj bo preprosta za računanje, lahko upošteva odvisnost od vseh znanih parametrov in drugih vrednosti, preprosto pa bi moral biti izračunljiv kvocient $\pi(\theta_{n+1})/\pi(\theta_n)$. Algoritem je podoben Gibbsovemu, le da vsako izbiro, ki je ne znamo narediti avtomatično, zamenjamo z dvema korakoma algoritma.

Osnovna varianta

Pogosto se dogaja, da pogojnih porazdelitev ne znamo dobro vzorčiti, kar nam oži uporabo Gibbsove metode. Včasih porazdelitev ni na nobenem seznamu znanih gostot ali pa je in enostavno ni zakodirana v paketih, ki jih imamo na voljo. Dokaj splošen pristop je v takih primerih algoritem Metropolis-Hastings.

Naj bo pogojna porazdelitev $p(\theta|\eta, x, y)$, ki jo preprosto označimo s $\pi(\theta)$, taka, da znamo gostoto izračunati kot funkcijo θ, ne znamo pa po njej slučajno izbirati. Od raziskovalca pričakujemo, da zna predlagati neko pogojno gostoto $q(\theta_{n+1}|\theta_n)$. Ta naj bo preprosta za računanje, lahko upošteva odvisnost od vseh znanih parametrov in drugih vrednosti, preprosto pa bi moral biti izračunljiv kvocient $\pi(\theta_{n+1})/\pi(\theta_n)$. Algoritem je podoben Gibbsovemu, le da vsako izbiro, ki je ne znamo narediti avtomatično, zamenjamo z dvema korakoma algoritma.

Osnovna varianta — nadaljevanje

- Na prvem koraku izberemo θ_{n+1} na podlagi predlagane pogojne gostote porazdelitve q in θ_n.
- Tako določeni približek θ_{n+1} sprejmemo z verjetnostjo $\alpha(\theta_n, \theta_{n+1})$, kjer je

$$\alpha(\theta_n, \theta_{n+1}) = \min\left(\frac{\pi(\theta_{n+1})/q(\theta_{n+1}|\theta_n)}{\pi(\theta_n)/q(\theta_n|\theta_{n+1})}, 1\right)$$

Algoritem razdeli obravnavo težko prepoznavne pogojne porazdelitve v dva dela. Najprej generiramo kandidata za novo točko z neko znano porazdelitvijo. Potem uporabimo kombinacijo težko prepoznavne in znane porazdelitve za to, da določimo sprejemni pogoj. Ta pogoj nam zagotavlja, da ima algoritem pravo ravnotežno porazdelitev. Drugače povedano, zaporedje θ_n ima limitno porazdelitev enako $\pi(\theta)$.

Osnovna varianta — nadaljevanje

- Na prvem koraku izberemo θ_{n+1} na podlagi predlagane pogojne gostote porazdelitve q in θ_n.
- Tako določeni približek θ_{n+1} sprejmemo z verjetnostjo $\alpha(\theta_n, \theta_{n+1})$, kjer je
$$\alpha(\theta_n, \theta_{n+1}) = \min\left(\frac{\pi(\theta_{n+1})/q(\theta_{n+1}|\theta_n)}{\pi(\theta_n)/q(\theta_n|\theta_{n+1})}, 1\right)$$

Algoritem razdeli obravnavo težko prepoznavne pogojne porazdelitve v dva dela. Najprej generiramo kandidata za novo točko z neko znano porazdelitvijo. Potem uporabimo kombinacijo težko prepoznavne in znane porazdelitve za to, da določimo sprejemni pogoj. Ta pogoj nam zagotavlja, da ima algoritem pravo ravnotežno porazdelitev. Drugače povedano, zaporedje θ_n ima limitno porazdelitev enako $\pi(\theta)$.

Algoritem Metropolis-Hastings je posplošitev Gibbsovega algoritma, ki ga dobimo nazaj, če v njem vzamemo $q(\theta_{n+1}|\theta_n) \propto \pi(\theta_{n+1})$ in $\alpha = 1$.

Osnovna varianta — nadaljevanje

- Na prvem koraku izberemo θ_{n+1} na podlagi predlagane pogojne gostote porazdelitve q in θ_n.
- Tako določeni približek θ_{n+1} sprejmemo z verjetnostjo $\alpha(\theta_n, \theta_{n+1})$, kjer je

$$\alpha(\theta_n, \theta_{n+1}) = \min\left(\frac{\pi(\theta_{n+1})/q(\theta_{n+1}|\theta_n)}{\pi(\theta_n)/q(\theta_n|\theta_{n+1})}, 1\right)$$

Algoritem razdeli obravnavo težko prepoznavne pogojne porazdelitve v dva dela. Najprej generiramo kandidata za novo točko z neko znano porazdelitvijo. Potem uporabimo kombinacijo težko prepoznavne in znane porazdelitve za to, da določimo sprejemni pogoj. Ta pogoj nam zagotavlja, da ima algoritem pravo ravnotežno porazdelitev. Drugače povedano, zaporedje θ_n ima limitno porazdelitev enako $\pi(\theta)$. Algoritem Metropolis-Hastings je posplošitev Gibbsovega algoritma, ki ga dobimo nazaj, če v njem vzamemo $q(\theta_{n+1}|\theta_n) \propto \pi(\theta_{n+1})$ in $\alpha = 1$.

Neodvisna varianta

Teoretično sicer ni omejitev na izbiro predlagane gostote q, vendar pa v praksi, kadar npr predlagamo gostoto, katere repi so pretanki v primerjavi s ciljno gostoto π, se lahko konvergenca algoritma tako upočasni, da postane njegova uporaba vprašljiva.

V neodvisni varianti algoritma pa izbiramo θ_{n+1} po neki brezpogojni gostoti $q(\theta)$, ki pa je lahko odvisna še od drugih parametrov. Odvisnost od θ_n pa pripeljemo v zgodbo preko sprejemnega pogoja:

- Na prvem koraku izberemo θ_{n+1} na podlagi predlagane pogojne gostote porazdelitve q.
- Tako določeni približek θ_{n+1} sprejmemo z verjetnostjo $\alpha(\theta_n, \theta_{n+1})$, kjer je

$$\alpha(\theta_n, \theta_{n+1}) = \min\left(\frac{\pi(\theta_{n+1})q(\theta_n)}{\pi(\theta_n)q(\theta_{n+1})}, 1\right)$$

Neodvisna varianta

Teoretično sicer ni omejitev na izbiro predlagane gostote q, vendar pa v praksi, kadar npr predlagamo gostoto, katere repi so pretanki v primerjavi s ciljno gostoto π, se lahko konvergenca algoritma tako upočasni, da postane njegova uporaba vprašljiva.

V neodvisni varianti algoritma pa izbiramo θ_{n+1} po neki brezpogojni gostoti $q(\theta)$, ki pa je lahko odvisna še od drugih parametrov. Odvisnost od θ_n pa pripeljemo v zgodbo preko sprejemnega pogoja:

- Na prvem koraku izberemo θ_{n+1} na podlagi predlagane pogojne gostote porazdelitve q.
- Tako določeni približek θ_{n+1} sprejmemo z verjetnostjo $\alpha(\theta_n, \theta_{n+1})$, kjer je
$$\alpha(\theta_n, \theta_{n+1}) = \min\left(\frac{\pi(\theta_{n+1})q(\theta_n)}{\pi(\theta_n)q(\theta_{n+1})}, 1\right)$$

Varianta s slučajnim sprehodom

V tej varianti se slučajne izbire pokoravajo modelu slučajnega sprehoda $\theta_{n+1} = \theta_n + e_t$. Tu je "napaka" e_t neodvisna izbira s povprečjem 0, ki naj ima debele repe, kot npr Studentova porazdelitev T. Pri izbiri te porazdelitve se ne oziramo toliko ne ciljno pogojno porazdelitev. Zaradi simetrije v pogojni porazdelitvi se algoritem poenostavi v:

- Na prvem koraku izberemo θ_{n+1} na podlagi predlagane pogojne gostote porazdelitve.
- Tako določeni približek θ_{n+1} sprejmemo z verjetnostjo $\alpha(\theta_n, \theta_{n+1})$, kjer je

$$\alpha(\theta_n, \theta_{n+1}) = \min\left(\frac{\pi(\theta_{n+1})}{\pi(\theta_n)}, 1\right)$$

Na voljo imamo še izbiro porazdelitve člena napake. Za konvergenco je še posebej pomembno prav uglasiti disperzijo te porazdelitve.

Varianta s slučajnim sprehodom

V tej varianti se slučajne izbire pokoravajo modelu slučajnega sprehoda $\theta_{n+1} = \theta_n + e_t$. Tu je “napaka” e_t neodvisna izbira s povprečjem 0, ki naj ima debele repe, kot npr Studentova porazdelitev T. Pri izbiri te porazdelitve se ne oziramo toliko ne ciljno pogojno porazdelitev. Zaradi simetrije v pogojni porazdelitvi se algoritem poenostavi v:

- Na prvem koraku izberemo θ_{n+1} na podlagi predlagane pogojne gostote porazdelitve.
- Tako določeni približek θ_{n+1} sprejmemo z verjetnostjo $\alpha(\theta_n, \theta_{n+1})$, kjer je

$$\alpha(\theta_n, \theta_{n+1}) = \min\left(\frac{\pi(\theta_{n+1})}{\pi(\theta_n)}, 1\right)$$

Na voljo imamo še izbiro porazdelitve člena napake. Za konvergenco je še posebej pomembno prav uglasiti disperzijo te porazdelitve.

Splošna pravila konvergence

Pri študiju konvergence se opremo na ergodijske lastnosti markovskih verig. Ugotoviti želimo, da metoda res konvergira, kako dobro konvergira in da konvergira k ciljni verjetnostni porazdelitvi. Pri tem rabimo teorijo markovskih verig z diskretnim časom, vendar v splošnem niso homogene in nimajo diskretnih stanj. Za poljubno Borelovo podmnožico A stanj na koraku n in poljubno stanje x na začetku, poznamo prehodno verjetnost po n korakih

$$P_n(x, A) = P\left[\theta_n \in A | \theta_0 = x\right].$$

Da bi imela veriga ranotežno stanje oziroma stacionarno porazdelitev, mora biti nerazcepna in aperiodična. To pomeni, da je vsako stanje dosegljivo iz vsakega stanja s pozitivno verjetnostjo in da noben del stanj veriga ne obiskuje ciklično. Če ima taka veriga stacionarno porazdelitev π, potem je ta enolična in k njej veriga tudi konvergira. Velja namreč:

Splošna pravila konvergence

Pri študiju konvergence se opremo na ergodijske lastnosti markovskih verig. Ugotoviti želimo, da metoda res konvergira, kako dobro konvergira in da konvergira k ciljni verjetnostni porazdelitvi. Pri tem rabimo teorijo markovskih verig z diskretnim časom, vendar v splošnem niso homogene in nimajo diskretnih stanj. Za poljubno Borelovo podmnožico A stanj na koraku n in poljubno stanje x na začetku, poznamo prehodno verjetnost po n korakih

$$P_n(x, A) = P\left[\theta_n \in A | \theta_0 = x\right].$$

Da bi imela veriga ranotežno stanje oziroma stacionarno porazdelitev, mora biti nerazcepna in aperiodična. To pomeni, da je vsako stanje dosegljivo iz vsakega stanja s pozitivno verjetnostjo in da noben del stanj veriga ne obiskuje ciklično. Če ima taka veriga stacionarno porazdelitev π, potem je ta enolična in k njej veriga tudi konvergira. Velja namreč:

Splošna pravila konvergence – nadaljevanje

$$\lim_{n \to \infty} P_n(x, A) = \pi(A).$$

Tudi kadar je teoretično konvergenca geometrijska, je lahko realno počasna.

Omejimo se le na vprašanje konvergence algoritmov tipa Metropolis-Hastings, saj je Gibbsov algoritem le njegov poseben primer. Ti algoritmi pa imajo nekatere posebne lastnosti, ki nam omogočajo preverjanje konvergenčnih pogojev v precejšnji splošnosti.

Splošna pravila konvergence – nadaljevanje

$$\lim_{n \to \infty} P_n(x, A) = \pi(A).$$

Tudi kadar je teoretično konvergenca geometrijska, je lahko realno počasna.

Omejimo se le na vprašanje konvergence algoritmov tipa Metropolis-Hastings, saj je Gibbsov algoritem le njegov poseben primer. Ti algoritmi pa imajo nekatere posebne lastnosti, ki nam omogočajo preverjanje konvergenčnih pogojev v precejšnji splošnosti.

Prvi od teh pogojev je *pogoj simetričnega ravnotežja* oziroma *pogoj obrnljivega časa*. Ta pogoj je izpolnjen, če za prehodno funkcijo $P(x, y)$ obstaja taka funkcija $\pi(x)$, da velja

$$P(x, y)\pi(x) = P(y, x)\pi(y) \tag{4}$$

za vsa stanja x in y. Če taka funkcija obstaja, ima avtomatično lastnost stacionarne gostote verjetnosti.

Splošna pravila konvergence — nadaljevanje

$$\lim_{n\to\infty} P_n(x, A) = \pi(A).$$

Tudi kadar je teoretično konvergenca geometrijska, je lahko realno počasna.

Omejimo se le na vprašanje konvergence algoritmov tipa Metropolis-Hastings, saj je Gibbsov algoritem le njegov poseben primer. Ti algoritmi pa imajo nekatere posebne lastnosti, ki nam omogočajo preverjanje konvergenčnih pogojev v precejšnji splošnosti.

Prvi od teh pogojev je *pogoj simetričnega ravnotežja* oziroma *pogoj obrnljivega časa*. Ta pogoj je izpolnjen, če za prehodno funkcijo $P(x, y)$ obstaja taka funkcija $\pi(x)$, da velja

$$P(x, y)\pi(x) = P(y, x)\pi(y) \tag{4}$$

za vsa stanja x in y. Če taka funkcija obstaja, ima avtomatično lastnost stacionarne gostote verjetnosti.

Pogoj obrnljivega časa

Lahko je preveriti, da velja

$$\int P(x,y)\pi(y)\,dy = \pi(x)\int P(y,x)\,dy = \pi(x).$$

Iz pogoja (4) pa potem sledi, da ima v ravnotežnem stanju dogodek, da veriga iz stanja x doseže stanje y isto verjetnost kot dogodek, da iz stanja y doseže stanje x. Pogoj (4) ima potem še eno posledico, namreč to, da je gostota $\pi(x)$ enolična povsod, kjer je pogojna gostota $P(x,y)$ neničelna.

Preverimo veljavnost pogoja obrnljivega časa za osnovno varianto algoritma Metropolis-Hastings. V tem primeru je

$$P(x,y) = \alpha(x,y)q(y|x) + (1-r(x))\delta_x(y),$$

kjer je

$$r(x) = \int \alpha(x,y)q(y|x)\,dy \quad \text{in} \quad \alpha(x,y) = \min\left(\frac{\pi(y)/q(y|x)}{\pi(x)/q(x|y)}, 1\right).$$

Pogoj obrnljivega časa

Lahko je preveriti, da velja

$$\int P(x,y)\pi(y)\,dy = \pi(x)\int P(y,x)\,dy = \pi(x).$$

Iz pogoja (4) pa potem sledi, da ima v ravnotežnem stanju dogodek, da veriga iz stanja x doseže stanje y isto verjetnost kot dogodek, da iz stanja y doseže stanje x. Pogoj (4) ima potem še eno posledico, namreč to, da je gostota $\pi(x)$ enolična povsod, kjer je pogojna gostota $P(x,y)$ neničelna.
Preverimo veljavnost pogoja obrnljivega časa za osnovno varianto algoritma Metropolis-Hastings. V tem primeru je

$$P(x,y) = \alpha(x,y)q(y|x) + (1-r(x))\delta_x(y),$$

kjer je

$$r(x) = \int \alpha(x,y)q(y|x)\,dy \quad \text{in} \quad \alpha(x,y) = \min\left(\frac{\pi(y)/q(y|x)}{\pi(x)/q(x|y)}, 1\right).$$

Pogoj obrnljivega časa — nadaljevanje

Vstavimo vse to v (4), da dobimo:

$$P(x,y)\pi(x) = \alpha(x,y)q(y|x)\pi(x) + (1 - r(x))\delta_x(y)\pi(x) =$$

$$= \min\left(\frac{\pi(y)/q(y|x)}{\pi(x)/q(x|y)}, 1\right) q(y|x)\pi(x) + (1 - r(x))\delta_x(y)\pi(x) =$$

$$= \min\left(\pi(y)q(x|y), q(y|x)\pi(x)\right) + (1 - r(y))\delta_y(x)\pi(y) =$$

$$= P(y,x)\pi(y).$$

Tako Gibbsov vzorčevalnik kot vse obravnavane variante algoritma Metropolis-Hastings generirajo Monte Carlo markovske verige, ki imajo obrnljiv čas z ravnotežno gostoto verjetnosti, ki je enaka ciljni gostoti verjetnosti.

Pogoj obrnljivega časa — nadaljevanje

Vstavimo vse to v (4), da dobimo:

$$P(x,y)\pi(x) = \alpha(x,y)q(y|x)\pi(x) + (1-r(x))\delta_x(y)\pi(x) =$$

$$= \min\left(\frac{\pi(y)/q(y|x)}{\pi(x)/q(x|y)}, 1\right) q(y|x)\pi(x) + (1-r(x))\delta_x(y)\pi(x) =$$

$$= \min\left(\pi(y)q(x|y), q(y|x)\pi(x)\right) + (1-r(y))\delta_y(x)\pi(y) =$$

$$= P(y,x)\pi(y).$$

Tako Gibbsov vzorčevalnik kot vse obravnavane variante algoritma Metropolis-Hastings generirajo Monte Carlo markovske verige, ki imajo obrnljiv čas z ravnotežno gostoto verjetnosti, ki je enaka ciljni gostoti verjetnosti.

Drugi pomembni pogoj, ki ga z lahkoto preverimo, je π-nerazcepnost.

Pogoj obrnljivega časa — nadaljevanje

Vstavimo vse to v (4), da dobimo:

$$P(x,y)\pi(x) = \alpha(x,y)q(y|x)\pi(x) + (1 - r(x))\delta_x(y)\pi(x) =$$

$$= \min\left(\frac{\pi(y)/q(y|x)}{\pi(x)/q(x|y)}, 1\right) q(y|x)\pi(x) + (1 - r(x))\delta_x(y)\pi(x) =$$

$$= \min\left(\pi(y)q(x|y), q(y|x)\pi(x)\right) + (1 - r(y))\delta_y(x)\pi(y) =$$

$$= P(y,x)\pi(y).$$

Tako Gibbsov vzorčevalnik kot vse obravnavane variante algoritma Metropolis-Hastings generirajo Monte Carlo markovske verige, ki imajo obrnljiv čas z ravnotežno gostoto verjetnosti, ki je enaka ciljni gostoti verjetnosti.
Drugi pomembni pogoj, ki ga z lahkoto preverimo, je π-nerazcepnost.

Pogoj π-nerazcepnosti

Za pogoj nerazcepnosti (glede na ciljno porazdelitev $\pi(x)$) lahko v literaturi najdemo celo vrsto zadostnih pogojev, preko katerih lahko dobimo nerazcepnost markovske verige, ki jo generira algoritem. Eden takih zadostnih pogojev je na primer, da iz pozitivnosti gostote $\pi(x)$ pri nekem $x \in \mathbb{R}$ sledi, da je $Q(x, y) > 0$. Obstaja tudi posplošitev pojma povezanosti poljubnih stanj x in y, ki v bistvu pomeni, da je možno iz kateregakoli stanja x slejkoprej priti v katerokoli stanje y; tudi ta pogoj je zadosten za nerazcepnost te verige.

Nadalje obstaja trditev, ki pove, da je vsaka nerazcepna markovska veriga, ki je porojena z algoritmom tipa Metropolis-Hastings, tudi aperiodična.

Pogoj π-nerazcepnosti

Za pogoj nerazcepnosti (glede na ciljno porazdelitev $\pi(x)$) lahko v literaturi najdemo celo vrsto zadostnih pogojev, preko katerih lahko dobimo nerazcepnost markovske verige, ki jo generira algoritem. Eden takih zadostnih pogojev je na primer, da iz pozitivnosti gostote $\pi(x)$ pri nekem $x \in \mathbb{R}$ sledi, da je $Q(x, y) > 0$. Obstaja tudi posplošitev pojma povezanosti poljubnih stanj x in y, ki v bistvu pomeni, da je možno iz kateregakoli stanja x slejkoprej priti v katerokoli stanje y; tudi ta pogoj je zadosten za nerazcepnost te verige.
Nadalje obstaja trditev, ki pove, da je vsaka nerazcepna markovska veriga, ki je porojena z algoritmom tipa Metropolis-Hastings, tudi aperiodična.
Zdaj pa je čas, da povemo, kakšne vrste konvergenca nas pravzaprav zanima pri metodah MCMV. Le redkokdaj je to taka ali drugačna konvergenca samega zaporedja porazdelitev k ciljni gostoti.

Pogoj π-nerazcepnosti

Za pogoj nerazcepnosti (glede na ciljno porazdelitev $\pi(x)$) lahko v literaturi najdemo celo vrsto zadostnih pogojev, preko katerih lahko dobimo nerazcepnost markovske verige, ki jo generira algoritem. Eden takih zadostnih pogojev je na primer, da iz pozitivnosti gostote $\pi(x)$ pri nekem $x \in \mathbb{R}$ sledi, da je $Q(x, y) > 0$. Obstaja tudi posplošitev pojma povezanosti poljubnih stanj x in y, ki v bistvu pomeni, da je možno iz kateregakoli stanja x slejkoprej priti v katerokoli stanje y; tudi ta pogoj je zadosten za nerazcepnost te verige.
Nadalje obstaja trditev, ki pove, da je vsaka nerazcepna markovska veriga, ki je porojena z algoritmom tipa Metropolis-Hastings, tudi aperiodična.
Zdaj pa je čas, da povemo, kakšne vrste konvergenca nas pravzaprav zanima pri metodah MCMV. Le redkokdaj je to taka ali drugačna konvergenca samega zaporedja porazdelitev k ciljni gostoti.

Vrste konvergence

Bolj pogosto gre za konvergenco nekega vzorčnega funkcionala, izračunanega na verigi. Kot primer navedimo izračun aposteriornega povprečja za dani parameter. Vprašamo se, kako konvergira zaporedje funkcionalov

$$\frac{1}{N}\sum_{n=1}^{N} f(\theta_n).$$

Tole je primer konvergenčnega izreka:

Izrek (Ergodijsko povprečenje)

Naj bo θ_n ergodijska veriga s stacionarno porazdelitvijo $\pi(x)$ in f funkcija z lastnostjo $\int |f(x)|\pi(x)\,dx < \infty$. Potem je neodvisno od začetnega stanja θ_0

$$\lim_{N\to\infty} \frac{1}{N}\sum_{n=1}^{N} f(\theta_n) = \int f(\theta)\pi(\theta)\,d\theta$$

Vrste konvergence

Bolj pogosto gre za konvergenco nekega vzorčnega funkcionala, izračunanega na verigi. Kot primer navedimo izračun aposteriornega povprečja za dani parameter. Vprašamo se, kako konvergira zaporedje funkcionalov

$$\frac{1}{N}\sum_{n=1}^{N} f(\theta_n).$$

Tole je primer konvergenčnega izreka:

Izrek (Ergodijsko povprečenje)

Naj bo θ_n ergodijska veriga s stacionarno porazdelitvijo $\pi(x)$ in f funkcija z lastnostjo $\int |f(x)|\pi(x)\,dx < \infty$. Potem je neodvisno od začetnega stanja θ_0

$$\lim_{N\to\infty} \frac{1}{N}\sum_{n=1}^{N} f(\theta_n) = \int f(\theta)\pi(\theta)\,d\theta$$

Centralni limitni izrek

Včasih potrebujemo še več, tj izrek tipa CLI:

Izrek (Centralni limitni izrek)

Naj bo θ_n ergodijska veriga s stacionarno porazdelitvijo $\pi(x)$ in f funkcija z lastnostjo $\int |f(x)|\pi(x)\,dx < \infty$. Potem velja pri nekem pozitivnem številu $\sigma(f)$, da konvergira pri poljubnem začetnem stanju θ_0

$$\sqrt{N}\left(\frac{1}{N}\sum_{n=1}^{N} f(\theta_n) - \int f(\theta)\pi(\theta)\,d\theta\right)$$

po zakonu proti normalni porazdelitvi z matematičnim upanjem 0 *in standardnim odklonom $\sigma(f)$.*

Nekateri matematiki opozarjajo, da so taki limitni izreki sami sebi namen, če ne povedo česa tudi o hitrosti konvergence. V tem primeru pravzaprav ne vemo, kdaj iteracijo ustaviti.

Centralni limitni izrek

Včasih potrebujemo še več, tj izrek tipa CLI:

Izrek (Centralni limitni izrek)

Naj bo θ_n ergodijska veriga s stacionarno porazdelitvijo $\pi(x)$ in f funkcija z lastnostjo $\int |f(x)|\pi(x)\,dx < \infty$. Potem velja pri nekem pozitivnem številu $\sigma(f)$, da konvergira pri poljubnem začetnem stanju θ_0

$$\sqrt{N}\left(\frac{1}{N}\sum_{n=1}^{N} f(\theta_n) - \int f(\theta)\pi(\theta)\,d\theta\right)$$

po zakonu proti normalni porazdelitvi z matematičnim upanjem 0 in standardnim odklonom $\sigma(f)$.

Nekateri matematiki opozarjajo, da so taki limitni izreki sami sebi namen, če ne povedo česa tudi o hitrosti konvergence. V tem primeru pravzaprav ne vemo, kdaj iteracijo ustaviti.

Hitrost konvergence

Ena od možnih ocen za hitrost konvergence je *geometrijska konvergenca*. Ta zagotavlja obstoj takega $0 < \lambda < 1$ in take konstante $K > 0$, da je

$$\|p_n(\cdot, \theta_0) - \pi(\cdot)\| \leqslant K\lambda^{-n},$$

kjer $\|\cdot\|$ pomeni to ali ono normo. Za algoritme tipa Metropolis-Hastings obstaja vrsta rezultatov v odvisnosti od ciljnih in predlaganih porazdelitev ter v odvisnosti od same metode. Pri neodvisni varianti te metode je za geomterijsko konvergenco dovolj, da repi predlagane gostote dominirajo repe ciljne gostote. Kljub tem obetavnim rezultatom pa je včasih občutek varnosti lahko lažen. Popularen je primer čarovničinega klobuka, porazdelitve, na kateri ima denimo Gibbsov vzočevalnik geometrijsko konvergenco, a λ je že tako blizu 1, da algoritem praktično nikoli ne konvergira. Možnosti, da se pomaknemo z roba klobuka na vrh, so eksponentno majhne.

Obdobje zažiganja

Kar nekaj je tudi literature o tem, kako na hitrost konvergence sklepamo iz samih simuliranih podatkov. Ker so simulirane vrednosti med seboj odvisne zaporedoma v času, je ena od metod analiza korelacijske strukture izbir. V ta namen izračunamo avtokorelacijsko funkcijo te časovne vrste. Vendar pa moramo biti pri interpretaciji spet previdni. Morebitne nizke avtokorelacije nas jasno opozarjajo na počasno konvergenco. Neki podatek o hitrosti konvergence je tudi izračun standardnega odklona simuliranih količin, kot npr iskanega funkcionala

$$\frac{1}{N}\sum_{n=1}^{N} f(\theta_n).$$

Tudi to nam lahko pomaga pri ocenjeganju konvergence. Prav tako nam lahko pomagajo izreki tipa centralni limitni izrek.

Obdobje zažiganja

Kar nekaj je tudi literature o tem, kako na hitrost konvergence sklepamo iz samih simuliranih podatkov. Ker so simulirane vrednosti med seboj odvisne zaporedoma v času, je ena od metod analiza korelacijske strukture izbir. V ta namen izračunamo avtokorelacijsko funkcijo te časovne vrste. Vendar pa moramo biti pri interpretaciji spet previdni. Morebitne nizke avtokorelacije nas jasno opozarjajo na počasno konvergenco. Neki podatek o hitrosti konvergence je tudi izračun standardnega odklona simuliranih količin, kot npr iskanega funkcionala

$$\frac{1}{N}\sum_{n=1}^{N} f(\theta_n).$$

Tudi to nam lahko pomaga pri ocenjeganju konvergence. Prav tako nam lahko pomagajo izreki tipa centralni limitni izrek.

Obdobje zažiganja

Ena od metod, s katero je možno nekoliko pospešiti konvergenco, je vpeljava *obdobja zažiganja*. Odločimo se, da bomo prvih K iteracij algoritma odvrgli oziroma "sežgali". Spomnimo se, da smo to metodo uporabili v našem prvem zgledu, ki smo ga tudi zakodirali. Predvidevamo, naj bi bil po tem času vpliv začetne porazdelitve že zelo majhen in naj bi se že približali ciljni porazdelitvi. Nadaljnjih N izbir pa vzamemo v celoti.